GatePrune: Layer-Aware Visual Token Pruning via Adaptive Gates to Achieve Efficient VLMs

Anonymous ACL submission

Abstract

Vision-language models (VLMs) excel across diverse multimodal tasks, yet their efficiency is severely hindered by the quadratic complexity of self-attention over large numbers of visual tokens. Existing token pruning methods either compress tokens before the language model, ignoring redundancy in deep layers, or prune inside the language model under the inaccurate assumption that early layers are the most critical. In this paper, we first conduct a layer-wise redundancy analysis, revealing that the middle layers of the LLM are the primary region of visual processing, while both early and deep layers exhibit substantial redundancy. Motivated by this finding, we propose GatePrune, a two-stage token pruning framework with two lightweight learnable gates: an Early Pruning Gate that removes tokens before the LLM, and a Deep Pruning Gate that performs early exit on tokens whose processing is complete. Experiments on LLaVA-1.5, Qwen2.5-VL, and Qwen3-VL demonstrate that GatePrune significantly outperforms state-of-the-art methods. Notably, on LLaVA-1.5-7B, GatePrune retains 99.2% of the base model's accuracy while pruning 66.7% of visual tokens. Our code and models will be released to the public upon acceptance of the paper.

1 Introduction

Vision-language models (VLMs) have demonstrated remarkable capability across a wide range of tasks, from image classification to complex visual reasoning on charts and documents (Bai et al., 2025a).

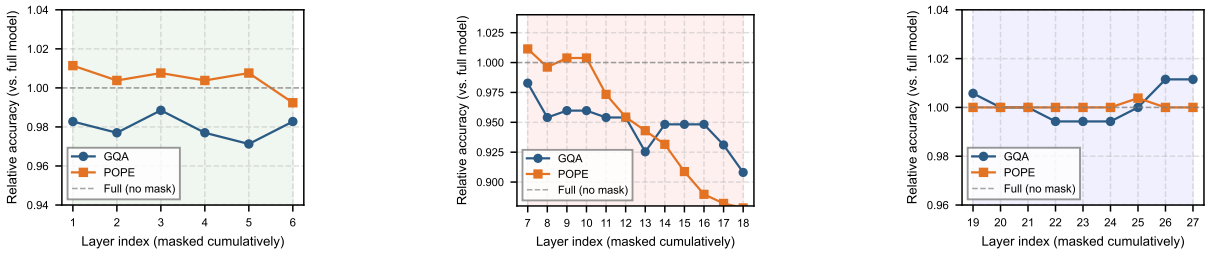
However, previous research has shown that one of the fundamental challenges of VLMs, despite their high capability, is the large number of visual tokens (Chen et al., 2024). The computational cost of self-attention grows quadratically with the number of tokens, which leads to a significant increase in computational cost (FLOPs) and memory consumption,ultimately limiting the inference speed and scalability of the model.

Nevertheless, it has been shown in VLMs that a significant portion of these tokens are redundant, and by intelligently pruning them, one can significantly reduce FLOPs, KV-cache memory, and inference latency while maintaining performance or dropping negligible accuracy (Huang et al., 2025; Zhang et al., 2025). In this regard, two main approaches have emerged: (1) pruning tokens before entering the language model, such as VisionZip (Yang et al., 2025) and HiPrune (Liu et al., 2026), and (2) pruning inside the language model, such as FastV (Chen et al., 2024) and PyramidDrop (Xing et al., 2025).

Despite these advances, existing methods often determine the pruning location without a precise analysis of the layer-wise behavior of the LLM. For instance, studies such as FastV (Chen et al., 2024), DynamicLLaVA (Huang et al., 2025), and PyramidDrop (Xing et al., 2025), assuming that the early layers play the primary role in visual processing, leave these layers untouched and initiate pruning from subsequent layers. In this paper, through a layer-by-layer analysis, we challenge this assumption and show that, contrary to common belief, the primary processing occurs in the middle layers, while the early and deep layers exhibit high redundancy.

Based on this observation, we propose a two-stage pruning framework that targets redundancy precisely in these two regions: the *Early Pruning Gate* (EPG), placed before the LLM, filters out visual tokens that are entirely irrelevant to the task at the very beginning, and the *Deep Pruning Gate* (DPG), applied at the deep layers, performs early exit by removing visual tokens whose processing is complete.

Our main contributions are: (1) Revealed through a layer-wise analysis that visual pro-



(a) Early layers. (b) Middle layers. (c) Deep layers.

Figure 1: Layer-wise redundancy analysis on Qwen 2.5 VL 7B.

cessing concentrates in the middle layers, motivating a two-stage pruning approach. (2) Proposed the **GatePrune** framework, which uses two lightweight learnable gates (EPG and DPG) to adaptively prune redundant visual tokens. (3) Retained 96.6% accuracy while keeping only 11.1% of the visual tokens on LLaVA-1.5-7B.

2 Related Works

Vision-Language Models Despite their impressive capabilities (Liu et al., 2023; Bai et al., 2025b,a), VLMs are computationally expensive due to the large number of visual tokens (Huang et al., 2025). For instance, LLaVA-1.5 maps each image to 576 tokens at a fixed resolution (Liu et al., 2023), while newer models such as Qwen2.5-VL and Qwen3-VL with dynamic resolution can produce thousands of visual tokens per image (Bai et al., 2025b,a). This escalating growth highlights the need for effective token pruning.

Pruning Location: Before or Inside the LLM? A typical VLM consists of a vision encoder (e.g., ViT) that converts an image into visual features, a projector that maps these features to the LLM input space, and an LLM that processes the resulting visual tokens (Liu et al., 2023). Token reduction methods can be divided into two categories based on the pruning location:

1. Pre-LLM Pruning: Current approaches reduce computational overhead prior to the Large Language Model (LLM) stage. For instance, VisionZip (Yang et al., 2025) merges redundant visual tokens post-Vision Transformer (ViT), while HiPrune (Liu et al., 2026) hierarchically selects key tokens within the ViT. However, these methods fail to address token redundancy within the deeper layers of the LLM.

2. Intra-LLM Pruning: Alternatively, methods like FastV (Chen et al., 2024) and SparseVLM (Zhang et al., 2025) defer pruning to deeper layers based on the assumption that early layers are uniformly critical. Similarly, PyramidDrop (Xing et al., 2025) employs a multi-stage pruning schedule under the premise that early-layer tokens are entirely indispensable. As our subsequent analysis demonstrates, these underlying assumptions regarding redundancy distribution are flawed.

Selection Mechanism: Heuristic or Learning-based? From the mechanism perspective, methods fall into two groups: training-free (relying on attention weights and heuristic criteria, e.g., FastV (Chen et al., 2024), PyramidDrop (Xing et al., 2025)) and learning-based (e.g., DynamicLLaVA (Huang et al., 2025) with a learnable gate and Gumbel-Softmax (Jang et al., 2017)). Even learning-based approaches employ only a single gate in the early layers and do not address pruning in deeper layers. The GatePrune framework bridges this gap with two learnable gates: EPG (before the LLM) and DPG (before deeper layers), targeting redundancy in both regions.

3 Methodology

3.1 Layer Behavior Analysis

To understand where visual token redundancy is concentrated, we conduct a controlled experiment that systematically masks the attention mechanism at various depths of the LLM and measures the resulting accuracy drop.

3.1.1 Early Layers: A Safe Region for Conditional Pruning

In prior work such as FastV (Chen et al., 2024), a common assumption is that the core of visual processing resides in the early layers of the LLM; as a result, pruning is either left entirely untouched for these layers, or performed from layer 2 onward. To test this assumption, we completely masked the attention mechanism in the early layers of the LLM.

The results, summarized in Figure 1a, show that this intervention leads to only a negligible drop in

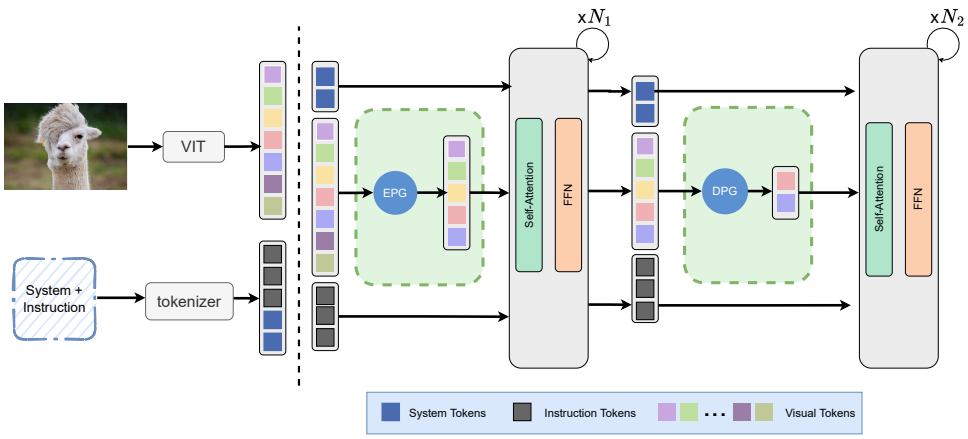


Figure 2: Overview of the proposed GatePrune framework.

accuracy across various tasks. This observation reveals that the early layers are not the primary region of visual processing and exhibit a high degree of redundancy. However, the small yet non-zero accuracy drop confirms that these layers are not entirely irrelevant and do perform some initial processing.

Consequently, the early layers represent a suitable region for significant token reduction, conditional on the pruning being performed with awareness of the visual tokens from the early layers that are important for the middle layers.

3.1.2 Middle Layers: The Core of Visual Processing

In contrast to the early layers, applying attention masking to the middle layers causes a severe drop, in the model's performance across all tasks (Figure 1b). This result indicates that fundamental visual processing and multimodal integration primarily occur in this region, and any disruption severely impairs the model's reasoning ability.

Therefore, pruning in this region is not only unhelpful, but can be catastrophic. This finding reveals a clear design requirement: the tokens that pass through the early layers must be delivered to this critical region with minimal damage. In other words, the success of early-stage pruning depends entirely on the pruning gate's ability to predict the impact of removing each token on the processing in the middle layers. This insight justifies the need for a learnable —as opposed to simple heuristic-based methods—and directly motivates the design of the Early Pruning Gate (EPG).

3.1.3 Deep Layers: An Opportunity for Early Exit

Unlike the middle layers, where even minor interference can be disastrous, masking the attention mechanism in the deep layers has very little impact on the final accuracy (Figure 1c). This behavior clearly indicates that the majority of visual processing is already complete before this stage. In other words, similar to the early layers, we again encounter high redundancy, although of a different nature: the tokens are not redundant because they are unimportant, but because their processing has been completed. This finding provides a golden opportunity for a second pruning stage. We therefore place the Deep Pruning Gate (DPG), our second learnable gate, at the entrance to the deep layers. The DPG identifies tokens whose processing is already complete and removes them from the computation path.

3.2 GatePrune

Based on the above findings, we design the *GatePrune* framework, which targets redundancy in two stages: the EPG, placed before the LLM, removes visual tokens that are unnecessary for answering the question, and the DPG, placed at the beginning of the deep layers, performs early exit by removing visual tokens whose processing is complete. Both gates are extremely lightweight, adding less than **0.2%** to the parameters of the base model.

Training and Inference. All parameters of the LLM, ViT and the projector remain **frozen** throughout training, and only two gates are trained in a two-stage schedule: in the first stage, only the EPG is added to the model and trained to remove unnec-

essary tokens; then, with EPG and the entire base model frozen, the DPG is added to the model and trained in the second stage to identify redundant tokens in the deep layers.

To learn binary decisions (remove/keep), we employ Gumbel-Softmax (Jang et al., 2017) with the Straight-Through Estimator (Bengio et al., 2013), which produces 0/1 masks in the forward pass and propagates gradients through the soft version in the backward pass. The final mask is applied to the LLM layers via MaskedSoftmax (Vaswani et al., 2017; Rao et al., 2021), ensuring that pruned tokens play no role in attention computations.

The binary mask generated by DPG is multiplied element-wise with the EPG mask, so that tokens already removed by EPG remain pruned; the sparsity loss then forces DPG to prune tokens that were not already removed by EPG in order to meet the desired ratio. Each gate is trained by minimizing the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda(\rho - \hat{\rho})^2, \tag{1}$$

where $\mathcal{L}_{\text{LM}}$ is the language modeling loss, ρ is the target keep rate, and $\hat{\rho}$ is the actual keep rate for the current sample.

At inference, pruning becomes deterministic. Let $\alpha, \beta \in (0, 1]$ denote the keep ratios of EPG and DPG, respectively, satisfying $\alpha \geq \rho_{\text{EPG}}$ and $\beta \geq \rho_{\text{DPG}}$, where ρ_{EPG} and ρ_{DPG} are the target keep rates used during training for each gate. EPG first keeps $K_1 = \lceil \alpha V \rceil$ visual tokens with the highest importance scores from the initial V tokens. DPG then selects $K_2 = \lceil \beta K_1 \rceil$ tokens from the K_1 remaining ones. Consequently, the N_1 layers before DPG (i.e., early and middle layers) process K_1 tokens, while the N_2 deep layers after DPG process only K_2 tokens. The average number of visual tokens across all LLM layers is, therefore,

$$\overline{V}_{\text{retained}} = \frac{K_1 N_1 + K_2 N_2}{N_{\text{total}}}, \tag{2}$$

where $N_{\text{total}} = N_1 + N_2$ is the total number of LLM layers. This directly leads to reductions in FLOPs and KV-cache memory.

4 Experiments

4.1 Experimental Setup

We conduct experiments on three models: LLaVA-1.5-7B (Liu et al., 2023), Qwen2.5-VL-7B (Bai et al., 2025b), and Qwen3-VL-4B (Bai et al., 2025a). All gates are trained with identical settings on the LLaVA-Instruct-150K dataset (Liu et al., 2023). The learning rate is 2×10^{-4}, batch size is 1, and gradient accumulation steps are 4, using two NVIDIA RTX 3090 GPUs. Because the gates are very lightweight, training is extremely fast; for example, training both gates on Qwen2.5-VL-7B takes approximately 4 hours. The values of the keep ratios α and β for each model are given in the Appendix C.

Method	GQA	MMB	MME	POPE	SQA^{IMG}
Vanilla, 576 Tokens (100%)					
LLaVA-1.5-7B	61.9	64.7	1862	85.9	69.5
Retain 64 Tokens (11.1%)					
FastV	46.1	48.0	1256	48.0	51.1
PyramidDrop	47.5	58.8	1561	55.9	69.0
VisionZip	55.1	60.1	1690	77.0	69.0
SparseVLM	53.7	60.1	1559	77.5	**69.7**
HiPrune	53.6	59.5	1646	73.0	68.9
Ours	**57.5**	**63.2**	**1772**	**85.2**	68.2

Table 1: Main results on LLaVA-1.5-7B at 11.1% visual token retention, along with the vanilla baseline (100%).

4.2 Results

Table 1 reports the evaluation results on LLaVA-1.5-7B at the retention ratio of 11.1% (64 visual tokens). Even under such aggressive pruning, our GatePrune framework outperforms all competing methods on four of the five benchmarks, achieving the best scores on GQA (Hudson and Manning, 2019), MMB (Liu et al., 2025), MME (Fu et al., 2024), and POPE (Li et al., 2023), while remaining competitive on SQA^{IMG} (Lu et al., 2022). Full results at additional retention ratios (33% and 22%) and on Qwen2.5-VL-7B and Qwen3-VL-4B are provided in Appendix B. Additional analysis of efficiency is also provided in Appendix A.

5 Conclusion

We conducted a layer-wise analysis of visual token redundancy in VLMs and found that, contrary to common assumptions, the middle layers are the primary region of visual processing while early and deep layers are highly redundant. Based on this insight, we proposed GatePrune, a two-stage framework with lightweight learnable gates that prune irrelevant and redundant tokens. Experiments on LLaVA-1.5-7B, Qwen2.5-VL-7B, and Qwen3-VL-4B show consistent improvements over existing methods. On LLaVA-1.5-7B, GatePrune retains 96.6% accuracy while keeping only 11.1% of visual tokens, reducing FLOPs by 78.3%.

6 Limitations

A limitation of this work is that, due to computational resource constraints, we were unable to evaluate GatePrune on models larger than 7B parameters. All experiments were conducted on LLaVA-1.5-7B, Qwen2.5-VL-7B, and Qwen3-VL-4B using two RTX 3090 GPUs. Scaling the framework to larger VLMs (e.g., 13B, 34B, or beyond) would require access to higher-memory GPUs or multi-node setups, which were not available for this study. We leave the investigation of GatePrune on larger-scale models to future work.

Acknowledgements

The model DeepSeek-V4 was used exclusively for writing assistance and editing purposes in this study.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. 2025a. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*.

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

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. 2024. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22363–22372.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, Yao Hu, and Shaohui Lin. 2025. Dynamic-LLaVA: Efficient multimodal large language models via dynamic vision-language context sparsification. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations (ICLR)*.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*. Oral.

Jizhihui Liu, Feiyi Du, Guangdao Zhu, Niu Lian, Jun Li, Bin Chen, Weili Guan, and Yaowei Wang. 2026. HiPrune: Hierarchical attention for efficient token pruning in vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2026*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2025. MMBench: Is your multi-modal model an all-around player? *International Journal of Computer Vision*, 133(3):1197–1213.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2025. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

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. 2025. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 74840–74857. PMLR.

A Appendix A: Efficiency Analysis

Reducing the number of visual tokens directly impacts computational cost and memory consumption. Table 2 reports this impact for LLaVA-1.5-7B at various retention ratios. Moving from the full model (100%) to 22.2% retention, FLOPs drop by 69% (from 8.65 TFLOPs to 2.70 TFLOPs), and prefill latency decreases by 53% (from 187ms to 87ms). The KV cache size shrinks correspondingly by 78% (from 317MB to 110MB). These results demonstrate that GatePrune maintains accuracy (cf. Table 1) while delivering substantial reductions in computational cost and memory footprint.

Ratio	FLOPs (T)	Prefill (ms)	Peak Mem (GB)	KV Cache (MB)
LLaVA-1.5-7B Efficiency Analysis				
100% (Full)	8.65	186.87	14.98	317.00
33.3% (↓ 66.7%)	3.53	106.14	14.42	142.03
22.2% (↓ 77.8%)	2.70	87.45	14.32	110.50
11.1% (↓ 88.9%)	1.87	86.35	14.21	78.75

Table 2: Efficiency analysis of LLaVA-1.5-7B.

B Appendix B: Supplementary Evaluation Results

Additional experimental results that could not be included in the main paper due to space constraints are presented here. We report the full evaluation on LLaVA-1.5-7B across all retention ratios in Table 4, the results on Qwen2.5-VL-7B with multiple baselines in Table 5, and the performance on Qwen3-VL-4B in Table 3.

Method	GQA	MMB	POPE	SQA	MMB_CN
Qwen 3 VL 4B - Vanilla, 576 Tokens (100%)					
Baseline	62.06	83.33	89.77	92.76	80.84
Retain 64 Tokens (11.1%)					
Ours	56.70	79.98	88.54	88.70	76.90
FastV	50.60	56.52	74.40	77.74	56.52

Table 3: Performance evaluation on Qwen 3 VL 4B. Comparison of our proposed method against FastV at 11.1% token retention ratio across multiple visual reasoning benchmarks.

C Appendix C: Pruning Ratio Configurations

The keep ratios α (for the Early Pruning Gate, EPG) and β (for the Deep Pruning Gate, DPG) used in our experiments are listed in Table 6.

Method	Venue	GQA	MMB	MME	POPE	SQA^{IMG}
Vanilla, 576 Tokens (100%)						
LLaVA-1.5-7B	*CVPR'24*	61.9	64.7	1862	85.9	69.5
Token retention ratio 33.3% (↓ **66.7%**)						
FastV	*ECCV'24*	52.7	61.2	1612	64.8	67.3
PyramidDrop	*CVPR'25*	57.3	63.3	1797	82.3	**69.2**
VisionZip	*CVPR'25*	59.3	63.0	1783	85.3	68.9
SparseVLM	*ICML'25*	59.5	**64.1**	1780	85.4	68.8
HiPrune	*ACL'26 Findings*	59.2	62.8	1814	86.1	68.9
Ours	-	**60.50**	63.83	**1868**	**86.89**	68.91
Token retention ratio 22.2% (↓ **77.8%**)						
FastV	*ECCV'24*	49.6	56.1	1490	59.6	60.2
PyramidDrop	*CVPR'25*	57.1	61.6	1761	82.3	68.4
VisionZip	*CVPR'25*	57.6	62.0	1762	83.2	**68.9**
SparseVLM	*ICML'25*	53.8	**64.4**	1761	85.0	68.5
HiPrune	*ACL'26 Findings*	57.3	62.2	1782	82.8	68.3
Ours	-	**59.64**	63.83	**1817**	**86.71**	67.97
Token retention ratio 11.1% (↓ **88.9%**)						
FastV	*ECCV'24*	46.1	48.0	1256	48.0	51.1
PyramidDrop	*CVPR'25*	47.5	58.8	1561	55.9	69.0
VisionZip	*CVPR'25*	55.1	60.1	1690	77.0	69.0
SparseVLM	*ICML'25*	53.7	**60.1**	1559	**77.5**	**69.7**
HiPrune	*ACL'26 Findings*	53.6	59.5	1646	73.0	68.9
Ours	-	**57.53**	63.23	**1772**	85.21	68.22

Table 4: Full evaluation results on LLaVA-1.5-7B across different visual token retention budgets. The results of the baseline methods are sourced from HiPrune (Liu et al., 2026)

Method	GQA	MMB	MME	POPE	SQA	MMB^{CN}	VizWiz
Vanilla, 576 Tokens (100%)							
Qwen2.5-VL-7B	60.86	84.27	2303	87.60	88.20	80.75	70.84
Token retention ratio 33.3% (↓ **66.7%**)							
HiPrune	58.90	82.60	2297	85.10	87.00	79.50	**69.20**
Ours	**58.96**	**83.07**	**2335**	**87.30**	**87.50**	**80.40**	68.97
Token retention ratio 22.2% (↓ **77.8%**)							
HiPrune	57.20	80.20	2168	84.00	85.70	77.60	**68.60**
Ours	**57.62**	**82.30**	**2327**	**86.69**	**87.11**	**79.81**	68.57
Token retention ratio 11.1% (↓ **88.9%**)							
PDrop	41.80	42.90	1642	68.40	77.00	38.70	53.30
VisionZip	54.90	75.60	2032	82.90	82.70	72.60	66.40
VScan	**57.10**	74.10	2130	**86.10**	82.20	71.50	**67.50**
HiPrune	52.50	76.10	1998	80.20	82.80	73.50	66.40
Ours	52.89	**78.78**	**2196**	84.70	**86.51**	**76.46**	66.71

Table 5: Main evaluation results on Qwen2.5-VL-7B across different visual token retention budgets. We compare our proposed method (Ours) against state-of-the-art visual token reduction algorithms under three fixed token constraints.

Model	**Target Retention**	α (EPG)	β (DPG)
LLaVA-1.5-7B	33.3%	0.635	0.151
	22.2%	0.425	0.150
	11.1%	0.212	0.150
Qwen2.5-VL-7B	33.3%	0.465	0.350
	22.2%	0.310	0.350
	11.1%	0.145	0.350
Qwen3-VL-4B	11.1%	0.195	0.100

Table 6: Keep ratios α (EPG) and β (DPG) for all models and retention targets.