Hi-Lo Prune: Look at What You'll Lose before Pruning with Hierarchical Token Selection

Supplementary Material

6. Detailed Experiment Settings

6.1. Implementation Details

In this section, we provide additional details on the experimental setup for evaluating pruning methods on both Qwen and LLaVA models. Following the official Qwen-VL evaluation code, we conduct comprehensive experiments across nine benchmark datasets. To ensure fair comparison, all methods are evaluated under identical pruning configurations. Specifically, for single-layer pruning, we prune the 2nd vision layer with token retain ratios of 0.2 and 0.1. For multi-layer pruning, we prune layers 4, 8, and 16 with token retain schedules of [0.8, 0.4, 0.2] and [0.4, 0.2, 0.1]. For the efficiency analysis, we align the usage of FlashAttention across baselines. CDPruner [59] and DivPrune [3] natively support FlashAttention, and thus are accelerated accordingly. FastV [10] and DART [50] are evaluated strictly following their official implementations, without applying FlashAttention-based optimization.

Additionally, our ablation studies reveal that different models exhibit varying dependencies on shallow visual features. Accordingly, we adopt model-specific settings in the first-stage pruning process. For Qwen2-VL-2B, we retain 60% of candidate tokens, while for Qwen2.5-VL-7B and Qwen3-VL-8B, we keep 30%. For Llava-1.5-7B, which rely more heavily on early-layer representations, we retain 90% of candidate tokens. For the Pre-Aware Fusion module, we set the top-k ratio to 10% and λ to 0.3. τ is set to 0.1 for Qwen2 and Qwen2.5 models, and 0.05 for Qwen3 and Llava models.

6.2. Datasets and Metrics

We evaluate our method on a broad set of vision-language benchmarks spanning both general and domain-specific tasks. General benchmarks include SQA [35] for scientific reasoning, POPE [25] for object hallucination, MMBench and its Chinese variant [32] for hierarchical perception–reasoning evaluation, and MME [61] with 14 subtasks covering visual perception and cognition. Domain-specific benchmarks include TextVQA [43] and DocVQA [38] for scene-text and document understanding, ChartQA [37] for chart interpretation and numerical reasoning, and VizWiz [16] for evaluating robustness on real-world, imperfect images.

Additionally, following FastV [10], we use inference FLOPs to compare the computational costs of different methods. Specifically, we estimate FLOPs by considering the multi-head attention (MHA) and feed-forward network (FFN) computations in each transformer layer. For a layer processing n tokens with hidden size d and FFN intermediate size m, the FLOPs are approximately $4nd^2 + 2n^2d + 2ndm$.

When our method applies pruning at layer L with ratio r, reducing tokens from n to $\tilde{n} = (1-r) \cdot n$ for a T-layer model, the overall FLOPs reduction ratio is calculated as:

$$1 - \frac{\sum_{i=1}^{L}(4nd^2 + 2n^2d + 2ndm)}{T(4nd^2 + 2n^2d + 2ndm)} - \frac{\sum_{i=L+1}^{T}(4\tilde{n}d^2 + 2\tilde{n}^2d + 2\tilde{n}dm)}{T(4nd^2 + 2n^2d + 2ndm)} \quad (3)$$

This formulation accounts for the full computational cost in the early layers and the reduced cost in the later layers after token pruning.

7. Evaluation on Video Understanding Tasks

To comprehensively assess the generalization capabilities of Hi-Lo Prune in the video domain, we extend our evaluation to video understanding tasks using the Video-LLaVA-7B [28] and Qwen3-VL-8B models. We conduct experiments on three standard video QA benchmarks: MSVD-QA [52], MSRVTT-QA [53], and TGIF [24], comparing our approach against early pruning baselines. As shown in Table 7, Hi-Lo Prune consistently outperforms baselines across both model architectures while retaining only 10% of visual tokens. Notably, on the MSRVTT-QA [53] benchmark, our method not only surpasses other pruning techniques but also achieves an accuracy 0.4% higher than the vanilla (unpruned) baseline. This validates that our "look at what you'll lose" philosophy effectively captures temporal redundancy, proving its value beyond static images.

8. Additional Ablation Analysis

As shown in Table 9, we conduct an ablation study on the fusion strength λ using a fixed 20% pruning ratio. We evaluate $\lambda \in 0.1, 0.3, 0.5, 1.0$ and observe a trade-off across task types. Larger values (e.g., $\lambda = 1.0$) enhance reasoning performance, notably on SQA, but reduce accuracy on perception-oriented benchmarks such as ChartQA, MME, and MMB. Smaller values (e.g., $\lambda = 0.1$) show the opposite trend, favoring visual understanding at the cost of reasoning ability. Balancing these effects, we adopt $\lambda = 0.3$ as the

Methods	MSRVT		MSVD		TGIF	
	Acc	Score	Acc	Score	Acc	Score
Video-LLaVA-7B						
Vanilla	54.2	3.39	69.6	3.92	48.2	3.40
+FastV	51.4	3.31	65.3	3.79	44.5	3.28
+DART	52.9	3.35	66.6	3.83	45.0	3.30
+DivPrune	51.7	3.30	66.0	3.80	45.2	3.31
+CDPruner	51.6	3.29	66.9	3.83	45.5	3.32
+Hi-Lo Prune	**53.8**	**3.37**	**67.4**	**3.87**	**46.0**	**3.34**
Qwen3-VL-8B						
Vanilla	59.6	3.52	74.5	4.02	56.1	3.54
+FastV	53.3	3.37	66.0	3.78	42.3	3.15
+DART	53.7	3.37	67.9	3.86	44.6	3.25
+DivPrune	56.5	3.44	71.1	3.94	46.6	3.35
+CDPruner	57.3	3.45	72.2	**3.98**	49.8	**3.50**
+Hi-Lo Prune	**60.0**	**3.54**	**72.6**	**3.98**	**51.2**	**3.50**

Table 7. Performance comparison on video understanding tasks with 10% token retention. Our method consistently outperforms baselines across both Video-LLaVA-7B and Qwen3-VL-8B architectures.

Methods	Resolution (Prefilling / Total Time)		
	16×16	64×64	80×80
Vanilla	*0.0660 / 0.6090*	*0.1746 / 0.7183*	*0.2494 / 0.7941*
DART (eager)	0.0645 / 0.6299	0.4123 / 0.9833	0.8810 / 1.4520
FastV (eager)	0.0639 / 0.6312	0.4081 / 0.9792	0.8755 / 1.4496
DivPrune	0.0652 / 0.5995	0.1105 / 0.6437	0.1524 / 0.6895
CDPruner	0.0674 / 0.6137	0.1270 / 0.6637	0.1784 / 0.7144
Hi-Lo Prune (Ours)	0.0665 / 0.6095	0.1210 / 0.6560	0.1690 / 0.7050

Table 8. Efficiency Analysis on Qwen3-VL-8B. P/T denotes Prefilling time / Total time(s), and headers like 16×16 indicate the number of image tokens. All methods utilize Flash Attention support, except for those explicitly marked as '(eager)'.

default setting, which provides strong overall performance across diverse benchmarks and demonstrates the robustness of our fusion mechanism.

To further evaluate the stability and generalizability of our method, we conducted an in-depth sensitivity analysis regarding two critical hyperparameters: the relaxation factor α and the threshold τ. All experiments were repeated over 4 independent runs to ensure statistical reliability. As illustrated in Figure 6, the empirical results confirm that our method is not strictly dependent on precise hyperparameter tuning and maintains robustness across a reasonable range of settings.

9. Analysis of Latency and Complexity

In this section, we provide a detailed analysis of the computational complexity and real-world inference latency of our method. As shown in Table 8, our approach maintains high efficiency comparable to baselines with a different number

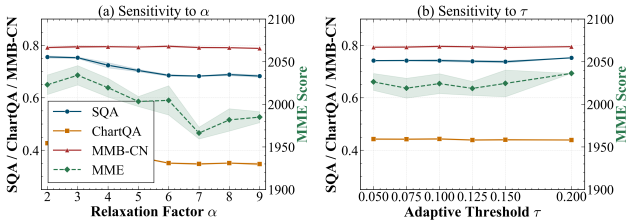

Figure 6. Sensitivity Analysis. (a) Factor α $[2, 9]$. (b) Threshold τ $[0.05, 0.2]$. Shaded regions denote variance across 4 runs.

of tokens.

9.1. Theoretical Complexity Analysis

Our method maintains a theoretical complexity of $(O(N^2D))$, consistent with early approaches. Crucially, we employ a relaxation factor α to optimize the token selection process. By shrinking the search space from N to M ($M = \alpha K \ll N$), we avoid the heavy computational burden of a full-pool search. Table 5 further confirms that across resolutions (512^2–2560^2), the overhead is negligible as PA-Fusion is restricted to early layers. Total inference time remains comparable to simple pruning.

9.2. Empirical Latency Analysis

As shown in Table 8, our method achieves inference speeds that match simple pruning baselines (e.g., DivPrune). This is primarily because, in modern GPU architectures employing Flash Attention, latency is often bound by memory bandwidth and CUDA kernel launch overhead rather than by minor floating-point operations. The algorithmic cost of our "High-Low" selection strategy is effectively "hidden" by the massive parallelization of the GPU.

10. Qualitative Visualizations

As shown in Figure 7, we present qualitative comparisons on representative MME and POPE samples. To better illustrate the effectiveness of our approach, we select challenging cases in which only our method produces correct predictions. In the visualizations, the red boxes highlight the regions that contain the information required to answer the question. By hierarchically selecting tokens and explicitly transferring their information to retained ones, Hi-Lo Prune enables aggressive pruning without significant performance loss. Together, these qualitative observations align closely with our quantitative results, further demonstrating that our fusion-based pruning pipeline maintains high fidelity while substantially reducing visual tokens.

	fusion strength λ	SQA	ChartQA	POPE	TextVQA	MME	MMB	MMB-CN^{Val}
	Vanilla	68.22%	84.64%	88.53%	76.85%	2390.83	89.78%	89.26%
20%	$\lambda = 0.1$	71.59%	53.28%	86.61%	70.43%	2147.48	84.64%	84.59%
	$\lambda = 0.3$	71.29%	52.88%	86.74%	70.60%	2156.45	85.82%	84.28%
	$\lambda = 0.5$	73.13%	53.16%	86.59%	70.40%	2130.08	85.22%	84.28%
	$\lambda = 1.0$	77.49%	52.72%	86.59%	70.40%	2134.13	85.57%	84.02%

Table 9. Ablation study on fusion strength λ. We evaluate different fusion strength values with 20% pruning ratio across multiple benchmarks on Qwen3-VL-8B. Results are compared against the vanilla unpruned model (shown in gray).

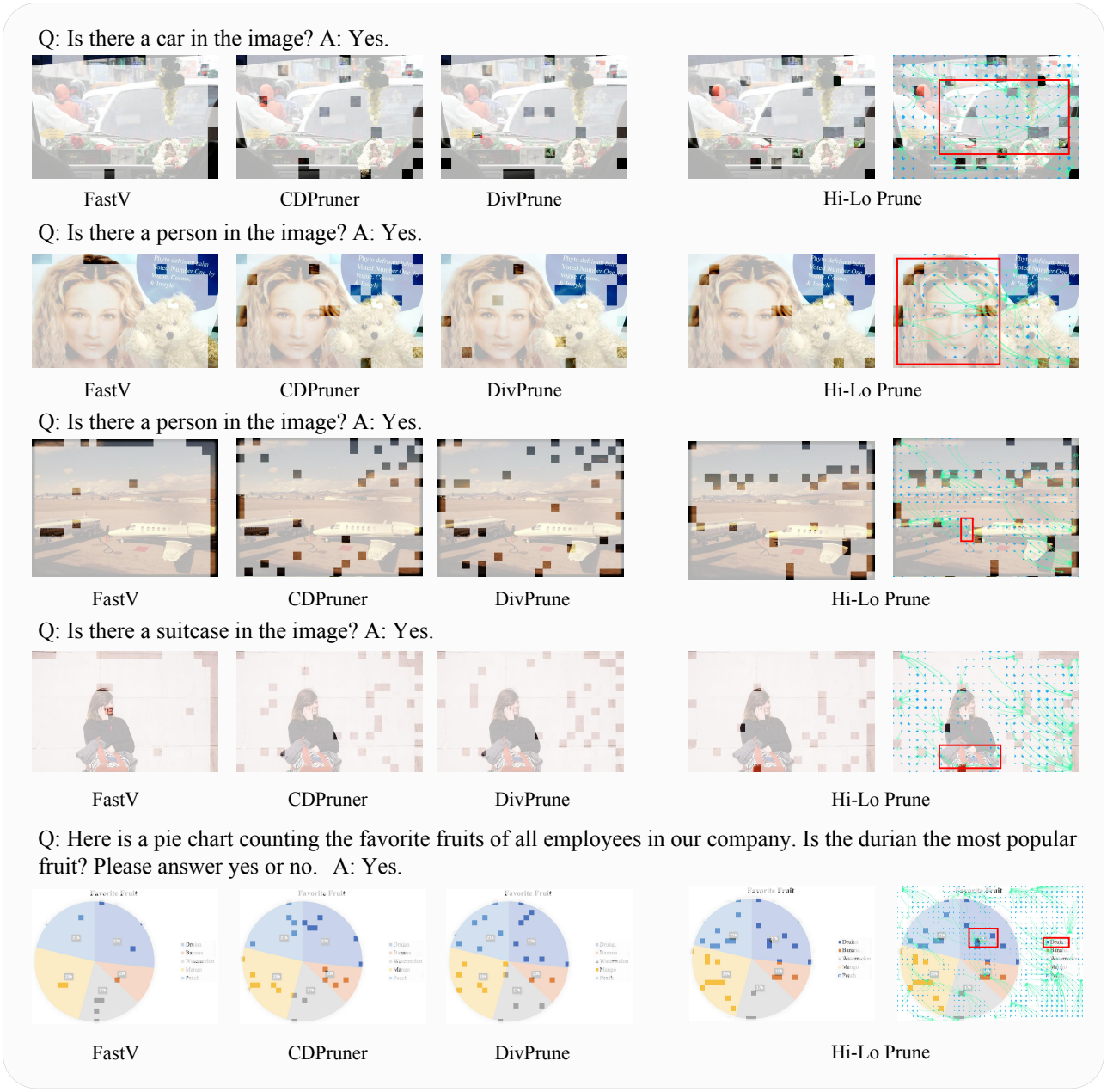

Figure 7. Visualization examples. Representative samples from MME and POPE benchmarks demonstrate that our approach preserves accurate visual understanding and reduces hallucination through hierarchical selection and pre-aware fusion.