BEYOND ATTENTION AND SIMILARITY: PERFORMANCE-GROUNDED VISUAL TOKEN PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

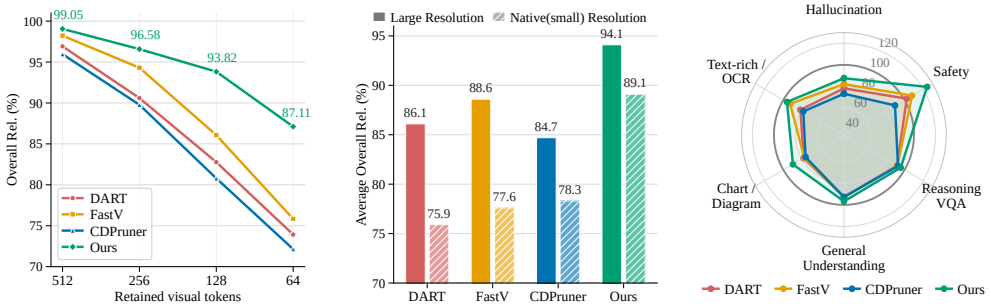
In this paper, we propose a performance-grounded visual token pruning framework, termed *PerfPruner*, which estimates token importance directly from the likelihood change of correct responses. By explicitly measuring each token's contribution to generating the target response, *PerfPruner* eliminates the objective mismatch inherent in prior proxy-based pruning methods, including attention-based and feature similarity-based approaches. To enable efficient inference, we first annotate a small training set with likelihood-based supervision and then train a lightweight predictor to estimate token importance scores at test time. We further introduce a score-constrained determinantal point process (SDPP) to reduce redundancy among retained tokens and improve pruning efficiency. By aligning token scoring with the pruning objective, our method can remove a broader range of irrelevant tokens, including redundant tokens, unique distractors, and out-of-distribution visual noise, thus providing a more reliable criterion for token selection across diverse image resolutions, pruning ratios, and tasks. Extensive experiments on 12 benchmarks spanning six tasks—including safety, hallucination, general understanding, reasoning VQA, text-rich/OCR, and chart/diagram understanding—demonstrate that *PerfPruner* consistently outperforms existing pruning methods across various pruning ratios and image resolutions, establishing new state-of-the-art performance.

1 INTRODUCTION

Multimodal large language models (MLLMs) (Liu et al., 2023; Wang et al., 2024a; Li et al., 2024a; Liu et al., 2024a) have recently achieved strong performance across diverse multimodal reasoning tasks by scaling data, model capacity, and visual input resolution. However, this progress comes with rapidly increasing computational cost. Unlike text inputs, visual inputs are often represented as hundreds to thousands of patch tokens, and the quadratic complexity of transformer attention makes processing these tokens a major bottleneck in latency, memory usage, and deployment cost. This issue becomes more severe for large-resolution and multi-image settings, where the number of visual tokens can dominate the context length. Consequently, visual token pruning (Chen et al., 2024a; Xing et al., 2024; Zhang et al., 2024; Yang et al., 2025; Wen et al., 2025; Alvar et al., 2025; Zhang et al., 2025) has emerged as a promising direction for improving the efficiency of MLLMs by discarding uninformative tokens while preserving task performance. Effective pruning can substantially reduce inference cost, enable larger-resolution reasoning under fixed budgets, and improve the practicality of deploying MLLMs in real-world resource-constrained scenarios.

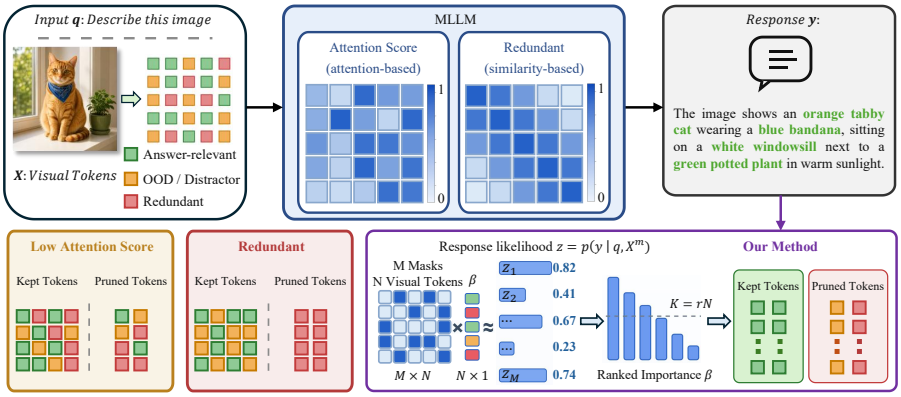
Our Motivation. Existing visual token pruning methods typically estimate token importance rankings using proxy signals derived from internal model representations, such as attention scores (Chen et al., 2024a; Xing et al., 2024; Zhang et al., 2024) and feature similarity (Wen et al., 2025; Alvar et al., 2025; Zhang et al., 2025). While computationally convenient, these criteria serve only as indirect proxies for the pruning objective—preserving the model's response quality after token removal—and are not explicitly optimized for this goal. Attention-based methods can be biased by architectural factors, including token position, layer-dependent attention patterns, and softmax competition among tokens, causing high scores to reflect routing behavior rather than causal utility (Kaduri et al., 2025; Chen et al., 2025; Kang et al., 2025). Thus, the kept high attention-score tokens can suffer from redundancy or out-of-distribution (OOD) visual noise as illustrated in Figure 2(a). Similarity-based methods primarily capture representational redundancy. Thus, as shown in Figure 2(b), it cannot identify unique yet irrelevant distractor tokens or OOD visual noise. *Overall, these proxy-based*

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.



(a) w/ various pruning ratios (b) w/ large and small resolutions (c) *w/ various tasks*

Figure 1: **Performance comparisons across pruning ratios, image resolutions, and tasks with Qwen2-VL-7B**. (a) Results are reported using large-resolution images. (b) The average overall performance across different pruning ratios is reported. (c) The average task benchmark performance across different pruning ratios is reported using large-resolution images. Our *PerfPruner* consistently achieves more stable performance across image resolutions, pruning ratios, and tasks.



(a) Attention-Based (b) Similarity-Based (c) Performance-Grounded

Figure 2: **Conceptual illustration of our method.** Existing token pruning methods rely on proxy objectives that are misaligned with the pruning goal. Attention-based methods prioritize tokens with high internal routing weights, while similarity-based methods focus on removing redundant features. However, pruning quality is ultimately determined by whether the model can still produce the correct response after token removal. Our method directly optimizes this response-preservation objective, eliminating the objective mismatch of prior approaches.

approaches fundamentally suffer from an objective mismatch: They rank tokens according to internal statistics rather than their actual contribution to generating correct responses. As a result, they often exhibit unstable performance across image resolutions, pruning ratios, and tasks as demonstrated in Figure 1, motivating the need for performance-grounded token pruning methods.

Performance-Grounded Visual Token Pruning (PerfPruner). To address the objective mismatch problem, we propose a performance-grounded visual token pruning framework, termed *PerfPruner* (Figure 2(c)). The key idea is to estimate token importance directly based on its impact on the likelihood of generating the correct response. Specifically, our method randomly samples M binary masks (0 represents pruning the token while 1 denotes retaining the token) and calculates their likelihood of correct responses. Then Lasso regression (Tibshirani, 1996) is applied to measure how each visual token contributes to the target response, thereby explicitly aligning token scoring with the pruning objective—preserving response quality after token removal. This objective-aligned formulation enables *PerfPruner* to identify a broad spectrum of irrelevant tokens, including distractors and out-of-distribution visual noise, and produce more reliable token rankings.

However, directly computing likelihood requires access to ground-truth responses, which are unavailable at inference time. To address this, we first construct a small annotated dataset with token importance labels, and then train a lightweight scoring module to predict token importance during

inference. Furthermore, semantically similar tokens can receive comparable importance scores, leading to redundant selections among retained tokens. To mitigate this issue and improve pruning efficiency, we introduce a score-constrained determinantal point process (SDPP) to encourage diversity among selected tokens. As shown in Figure 1, *PerfPruner* consistently achieves strong performance across different pruning ratios, image resolutions, and tasks. Notably, it can largely improve model robustness against jailbreak attacks compared to prior methods, highlighting its effectiveness as a general and reliable criterion for visual token pruning.

Experimental Results. *PerfPruner* consistently establishes new state-of-the-art performance across input resolutions, pruning ratios, and task categories (6 tasks with 12 benchmarks). On Qwen2-VL-7B, *PerfPruner* significantly outperforms the strongest baseline under both large-resolution and native (small-resolution) inputs, with more gains under more aggressive pruning. Specifically, the overall performance improvement increases from **+0.83%** to **+11.27%** as the large-resolution budget decreases from 512 to 64 tokens, and reaches **+14.40%** under native(small)-resolution inference with only 5% visual tokens retained. These improvements are consistent across safety, hallucination, general understanding, reasoning VQA, OCR/text-rich, and chart/diagram tasks. Particularly, *PerfPruner* largely improves safety robustness against jailbreaks, even raising the overall safety score by **+24.54%** over the full-token vanilla model. These results show that *PerfPruner* is a more reliable pruning criterion than attention-, similarity-, or diversity-based proxy signals.

In summary, the main contributions of our work are:

- We propose the performance-grounded pruning (*PerfPruner*) that ranks tokens based on their contribution to the likelihood of correct responses, matching the objective of pruning.
- To make *PerfPruner* practical, we construct a small annotated dataset with ground-truth responses and train a lightweight score predictor for efficient inference. In addition, we propose a score-constrained DPP to further improve pruning efficiency.
- We achieve new state-of-the-art performance–efficiency trade-offs across image resolutions and pruning ratios on 12 benchmarks spanning 6 tasks: safety, hallucination, general understanding, reasoning VQA, text-rich/OCR, and chart/diagram understanding.

2 Related Work

Visual Token Pruning. Visual token pruning accelerates MLLMs by removing answer-irrelevant visual tokens while preserving response quality. Attention-based methods use text-visual or special-token attention scores (Chen et al., 2024a; Xing et al., 2024; Zhang et al., 2024; Yang et al., 2025), but may suffer from positional and layer-dependent biases and retain redundant salient tokens or OOD visual noise. Similarity- or diversity-based methods reduce redundancy by preserving representative tokens (Wen et al., 2025; Alvar et al., 2025; Ye et al., 2025; Dong et al., 2025). Recent methods also incorporate instruction relevance (Zhang et al., 2025), multimodal coverage (Dong et al., 2025), or explainability-based relevance (Lei et al.). However, these internal proxies do not directly optimize response preservation under pruning. In contrast, *PerfPruner* aligns token scoring with this objective.

Robustness of MLLMs. MLLM robustness research addresses safety against jailbreaks and faithfulness against hallucinations. Jailbreaks use perturbation-based (Qi et al., 2024), structure-based (Gong et al., 2025; Liu et al., 2024b), or hybrid strategies (Li et al., 2024c) to bypass guardrails. Defenses include *tuning-based alignment* (e.g., instruction tuning, RLHF, adversarial training) (Zhou et al., 2024; Ji et al., 2024) and *training-free interventions*, including input filtering (Wang et al., 2024b) and safe decoding (Xu et al., 2024). Hallucination mitigation includes post-processing (Huang et al., 2024), self-correction (Yin et al., 2024), RLHF-based alignment (Yu et al., 2024), and contrastive or uncertainty-aware decoding (Chen et al., 2024b; Leng et al., 2024). Many require additional training data, external models or detectors, repeated decoding, or costly rollback.

Determinantal Point Process. Determinantal Point Process (DPP) was originally introduced to model fermion repulsion in quantum systems (Macchi, 1975), where the resulting anti-bunching property naturally encourages diversity. It has since been widely used as a probabilistic framework for diverse subset selection and list-wise diversity modeling (Taskar, 2013; Chen et al., 2018; Celis et al., 2018; Li et al., 2024b; Sun et al., 2025). Compared with pairwise diversity objectives such as the Max-Min Diversity Problem (Porumbel et al., 2011), DPP captures global diversity through the determinant of a similarity kernel, often producing more balanced and representative subsets.

3 METHOD

We propose a performance-grounded visual token pruning framework that estimates token importance according to each token's contribution to generating correct responses. Unlike prior approaches, our method directly aligns token scoring with the pruning objective: preserving response quality under a reduced visual token budget. Section 3.1 introduces the estimation of the token importance score with respect to the likelihood change of the correct response. Then, with the score estimation mechanism, we present our performance-grounded visual token pruning (***PerfPruner***) framework in Section 3.2.

3.1 MEASURING TOKEN IMPORTANCE WITH RESPONSE LIKELIHOOD CHANGE

Likelihood-based Token Importance Estimation. Consider a multimodal large language model (MLLM) f, an image-question pair (I, q), and the corresponding target response $y = (y_1, \ldots, y_T)$. Let the visual encoder with a projection layer produce N visual tokens:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \quad \mathbf{x}_i \in \mathbb{R}^d. \tag{1}$$

We introduce a binary mask vector $\mathbf{m}^{(j)} \in \{0,1\}^N$, where $m_i^{(j)} = 1$ indicates that token x_i is retained and $m_i^{(j)} = 0$ indicates that it is pruned. Applying the mask yields the masked visual tokens:

$$X^{(\mathbf{m}^{(j)})} = \{m_1^{(j)}\mathbf{x}_1, m_2^{(j)}\mathbf{x}_2, \ldots, m_N^{(j)}\mathbf{x}_N\}. \tag{2}$$

The model f predicts the likelihood of the target response conditioned on the question and retained visual tokens: $z_j = p(y \mid q, X^{(\mathbf{m}^{(j)})})$.

Our key principle is that token importance reflects its contribution to generating correct responses. We therefore evaluate each token through response likelihood changes under randomized masking interventions. Specifically, we randomly sample M binary masks with a retention ratio r:

$$\{\mathbf{m}^{(1)}, \mathbf{m}^{(2)}, \ldots, \mathbf{m}^{(M)}\}, \quad m_i^{(j)} \sim \text{Bernoulli}(r). \tag{3}$$

To obtain stable and sparse token importance scores, we then solve the following Lasso regression:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{M} \sum_{j=1}^{M} \left(z_j - \beta_0 - \sum_{i=1}^{N} \beta_i m_i^{(j)} \right)^2 + \lambda \|\boldsymbol{\beta}\|_1, \tag{4}$$

where λ controls sparsity, $\boldsymbol{\beta} = \{\beta_1, \beta_2, ..., \beta_N\}$ and β_i will be the importance score of ith token.

A larger score β_i indicates that retaining token x_i contributes more to preserving the likelihood of the correct response, and therefore reflects higher token importance. By measuring importance through response likelihood changes, our scoring criterion is directly aligned with the downstream objective of maintaining model performance under token pruning.

3.2 PERFORMANCE-GROUNDED VISUAL TOKEN PRUNING

Estimating likelihood-based token importance scores in Section 3.1 requires access to correct responses, which are unavailable during inference. To bridge this gap, we compute supervision labels on a small training set and train a lightweight module to predict token importance scores at test time.

Lightweight Module Architecture and Training. Our token importance predictor $\mathcal{S}$ is a lightweight network composed of a cross-attention layer, a self-attention layer, and a ResConvMLP head. The architecture details are presented in Figure 3 of the Appendix. Given the textual query q and visual tokens X as inputs, the module predicts the token importance scores: $\boldsymbol{\beta}' = \mathcal{S}(q, X)$.

To transfer the likelihood-based supervision described above, we train $\mathcal{S}$ to match the pseudo-label scores $\boldsymbol{\beta}$. Specifically, we normalize both $\boldsymbol{\beta}'$ and $\boldsymbol{\beta}$ into probability distributions and minimize their Kullback–Leibler (KL) divergence:

$$\mathcal{L}_{\text{KL}} = \text{KL}\big(\boldsymbol{\beta} \,\|\, \boldsymbol{\beta}'\big). \tag{5}$$

Table 1: **Oracle upper-bound performance under different pruning ratios** "Rel." represents the average percentage of performance maintained over the listed benchmarks.

Method	Pruning Ratio	MME	MMB^{EN}	OCRVQA	ChartQA	AI2D	Rel.
Qwen2-VL-7B	0%	2276.05	79.99	71.55	81.56	78.92	100.00%
Qwen2-VL-7B w/Oracle	60%	2687.27	91.48	76.33	88.32	91.19	112.59%
	80%	2694.19	91.37	71.78	85.80	90.74	110.62%
	90%	2666.12	89.97	63.74	78.44	89.64	105.69%
LLaVA-1.5-7B	0%	1820.14	64.74	59.90	17.92	54.60	100.00%
LLaVA-1.5-7B w/Oracle	60%	2581.97	78.36	69.17	29.28	83.00	138.76%
	80%	2629.71	78.81	68.33	29.92	84.03	140.23%
	90%	2599.57	77.35	64.58	28.60	82.74	136.25%

This objective encourages the predictor to approximate the token importance distribution while enabling efficient score estimation during inference.

Score Constrained Determinantal Point Process (SDPP). Semantically similar visual tokens often receive comparable importance scores. Thus, selecting tokens solely according to their predicted scores may retain redundant tokens within the preserved subset. To further improve pruning efficiency, we apply a score-constrained determinantal point process (SDPP) after importance estimation.

Given a target retention ratio r, we first rank all tokens by their predicted importance scores such that $\beta_{(1)} \geq \beta_{(2)} \geq \cdots \geq \beta_{(N)}$. We then form a candidate pool consisting of the top $\alpha(rN)$ tokens, where $\alpha \geq 1$ is an expansion factor and $\alpha = 1.5$ by default. For the candidate tokens, we construct the DPP (Macchi, 1975; Zhang et al., 2025) kernel matrix $L \in \mathbb{R}^{\alpha(rN)\times\alpha(rN)}$ as

$$L_{ij} = \beta_{(i)}\beta_{(j)}\, x_{(i)}^{\top} x_{(j)}, \tag{6}$$

where $x_{(i)}$ denotes the normalized feature of the i-th ranked token. The score terms $\beta_{(i)}\beta_{(j)}$ favor highly important tokens, while the feature similarity term promotes diversity among retained tokens. Finally, we perform DPP subset selection on L to choose rN visual tokens for downstream inference.

Empirical Analysis. To evaluate the potential of likelihood-based token importance for pruning, we first study an oracle setting. For datasets including MME, MMB, OCRVQA, ChartQA, and AI2D, we compute token importance scores using ground-truth responses and apply *PerfPruner* accordingly.

We test pruning ratios of 60%, 80%, and 90% with Qwen2-VL-7B and LLaVA-1.5-7B. As shown in Table 1, oracle pruning not only preserves performance but even significantly surpasses the full-token baselines, achieving up to 36.25% overall gains on LLaVA-1.5-7B while retaining only 10% of tokens. This highlights the strong potential of accurately learned token importance for efficient pruning.

4 Experiments

In this section, we conduct extensive experiments to validate the effectiveness of our method. Section 4.1 presents the experimental setup. In Section 4.2, we study i). whether performance-grounded token scoring improves the accuracy–compression trade-off under different visual-token budgets; ii). whether the method remains effective for both large-resolution and native(small)-resolution inputs. Then, we examine whether pruning improves safety robustness against multimodal jailbreak attacks while preserving response utility in Section 4.3. Finally, we evaluate cross-architecture generalization on LLaVA-1.5-7B in Section 4.4, which is followed by ablations in Section 4.5.

4.1 Experimental Setup

Evaluation Benchmarks. We evaluate *PerfPruner* on diverse multimodal benchmarks, including HallBench (Guan et al., 2024), MME (Fu et al., 2023), MMBench-CN (Liu et al., 2024c), MMBench-EN (Liu et al., 2024c), GQA (Hudson & Manning, 2019), CCBench (Liu et al., 2024c), TextVQA (Singh et al., 2019), OCRVQA (Mishra et al., 2019), ChartQA (Masry et al., 2022) and AI2D (Kembhavi et al., 2016). These benchmarks cover hallucination evaluation, general multimodal understanding, reasoning VQA, text-rich/OCR-oriented understanding, and chart/diagram reasoning. We follow each benchmark's official evaluation protocol and report its default metric. For safety robustness, we evaluate on FigStep (Gong et al., 2025) and MM-SafetyBench (Liu et al., 2024b), which test whether MLLMs produce unsafe responses when harmful instructions are embedded in multimodal inputs.

Table 2: **Performance comparisons on Qwen2-VL-7B using large-resolution images on 10 benchmarks.** "Rel" represents the average percentage of performance maintained.

Method	Hallucination	General Understanding			Reasoning VQA		Text-rich / OCR		Chart / Diagram		
	HallBench	MME	MMB^{CN}	MMB^{EN}	GQA	CCBench	VQA^{Text}	OCRVQA	ChartQA	AI2D	Rel.
Upper Bound, All 1296 Tokens (100%)											
Qwen2-VL-7B	43.08	2347.02	81.84	81.33	61.62	63.53	83.90	71.78	82.40	79.92	100.0%
Retain 512 Tokens (↓60.5%)											
FastV (ECCV24)	41.54	2324.67	80.94	80.99	60.49	62.94	83.09	71.88	76.96	78.63	98.22%
DART (EMNLP25)	39.78	2323.89	80.21	79.54	59.81	63.52	82.15	71.32	74.64	77.65	96.92%
DivPrune (CVPR25)	43.52	2303.08	79.54	78.87	60.63	54.71	81.80	70.54	80.56	78.63	96.97%
CDPruner (NeurIPS25)	38.46	2243.84	80.55	80.16	60.97	61.57	80.41	69.34	75.80	77.88	95.96%
MMTok (ICLR26)	39.12	2309.07	80.55	80.16	60.53	61.57	81.82	70.80	76.76	78.08	96.83%
PerfPruner (Ours)	41.98	2334.79	81.50	81.28	60.30	64.51	83.24	71.78	79.24	79.31	**99.05%**
Retain 256 Tokens (↓80.2%)											
FastV (ECCV24)	38.46	2279.48	79.09	79.04	58.77	62.35	80.63	71.71	63.76	76.72	94.31%
DART (EMNLP25)	34.73	2236.88	78.14	77.52	57.31	61.76	75.77	66.37	60.32	74.38	90.60%
DivPrune (CVPR25)	37.14	2077.06	75.11	75.11	59.94	46.67	76.66	62.66	63.60	76.17	88.07%
CDPruner (NeurIPS25)	33.63	2166.48	78.98	78.31	59.99	60.59	73.92	62.60	60.40	74.58	89.78%
MMTok (ICLR26)	36.26	2225.70	77.47	77.69	60.28	54.90	77.79	66.93	68.92	76.10	91.82%
PerfPruner (Ours)	38.90	2339.20	79.60	80.16	58.78	63.92	81.76	71.52	73.92	77.66	**96.58%**
Retain 128 Tokens (↓90.1%)											
FastV (ECCV24)	34.51	2126.89	74.89	74.38	54.64	58.04	74.71	69.92	41.04	72.54	86.07%
DART (EMNLP25)	31.87	2148.48	74.94	74.66	54.87	58.04	62.95	56.38	46.68	70.50	82.77%
DivPrune (CVPR25)	29.67	1944.90	69.51	70.29	57.53	38.24	64.30	47.27	47.40	70.89	76.54%
CDPruner (NeurIPS25)	28.35	2114.67	75.17	76.51	58.59	56.08	61.42	52.80	39.36	70.47	80.79%
MMTok (ICLR26)	30.11	1994.13	73.77	74.16	58.37	44.90	65.48	55.11	50.60	72.31	80.83%
PerfPruner (Ours)	39.34	2271.80	78.20	78.42	56.65	62.55	79.61	69.99	66.28	75.84	**93.82%**
Retain 64 Tokens (↓95.1%)											
FastV (ECCV24)	27.03	1915.74	69.23	70.46	50.36	54.90	60.67	62.86	22.72	69.69	75.84%
DART (EMNLP25)	28.35	1988.21	71.02	70.12	51.65	53.73	45.72	46.68	35.56	67.58	73.92%
DivPrune (CVPR25)	22.64	1746.38	61.60	62.28	53.78	27.25	47.66	29.88	33.92	68.04	63.37%
CDPruner (NeurIPS25)	25.71	1928.00	70.46	73.32	56.45	50.98	45.57	42.77	26.40	68.33	72.14%
MMTok (ICLR26)	27.47	1767.06	66.76	66.31	54.71	35.10	45.96	39.49	31.64	67.42	67.87%
PerfPruner (Ours)	30.77	2149.90	75.95	76.23	53.90	59.41	73.69	66.41	55.24	74.48	**87.11%**

Table 3: **Performance comparisons on Qwen2-VL-7B using native(small)-resolution images on 10 benchmarks.** "Rel" represents the average percentage of performance maintained.

Method	Hallucination	General Understanding			Reasoning VQA		Text-rich / OCR		Chart / Diagram		
	HallBench	MME	MMB^{CN}	MMB^{EN}	GQA	CCBench	VQA^{Text}	OCRVQA	ChartQA	AI2D	Rel.
Native-Resolution (100%)											
Qwen2-VL-7B	43.08	2276.05	79.54	79.99	62.34	61.96	83.83	71.55	81.56	78.92	100.00%
Retain 40% Tokens (↓60%)											
FastV (ECCV24)	40.88	2169.44	77.07	77.24	59.84	60.98	83.37	70.80	67.72	76.98	95.70%
DART (EMNLP25)	38.46	2129.18	76.68	76.57	59.62	59.02	80.78	60.94	67.84	74.68	92.52%
DivPrune (CVPR25)	40.66	2081.85	74.83	74.72	60.07	45.69	82.02	64.13	74.96	75.19	91.81%
CDPruner (NeurIPS25)	36.92	2139.19	77.19	77.91	60.74	59.80	80.06	58.76	68.88	73.90	92.38%
MMTok (ICLR26)	40.44	2125.21	76.01	77.02	61.02	52.16	80.33	66.24	69.68	74.71	92.97%
PerfPruner (Ours)	41.32	2242.09	77.80	78.48	59.70	61.18	83.05	71.13	75.60	77.04	**97.36%**
Retain 20% Tokens (↓80%)											
FastV (ECCV24)	36.70	1975.34	71.97	72.03	55.62	55.29	80.05	64.49	49.32	71.99	86.83%
DART (EMNLP25)	32.75	1936.22	71.97	71.86	56.11	51.76	71.67	49.22	50.92	69.92	82.03%
DivPrune (CVPR25)	33.85	1868.34	67.49	67.88	57.54	36.27	73.24	43.42	54.48	70.79	78.58%
CDPruner (NeurIPS25)	34.73	1975.14	72.70	73.71	59.09	53.92	71.63	45.25	48.96	70.89	83.13%
MMTok (ICLR26)	34.29	1873.53	70.12	70.96	58.70	45.69	75.42	50.13	55.96	71.96	82.65%
PerfPruner (Ours)	40.00	2098.38	76.46	76.46	57.88	58.24	81.27	69.63	67.48	75.19	**93.59%**
Retain 10% Tokens (↓90%)											
FastV (ECCV24)	27.69	1664.11	63.34	63.40	51.32	43.92	72.41	48.67	25.76	67.81	72.14%
DART (EMNLP25)	27.03	1777.87	60.54	60.03	51.13	41.57	56.61	39.29	36.64	66.94	69.33%
DivPrune (CVPR25)	28.35	1608.57	57.51	58.18	54.00	27.45	58.63	29.98	38.40	67.23	65.66%
CDPruner (NeurIPS25)	31.21	1849.33	67.77	69.45	54.91	47.84	57.27	38.38	32.28	67.84	73.85%
MMTok (ICLR26)	28.13	1672.79	62.28	62.56	54.91	40.98	59.98	35.58	36.08	66.97	69.99%
PerfPruner (Ours)	36.70	1845.16	73.04	73.21	54.98	53.33	78.91	64.13	58.24	72.02	**87.03%**
Retain 5% Tokens (↓95%)											
FastV (ECCV24)	25.05	1366.90	42.38	44.51	45.35	21.18	56.15	35.12	17.88	65.71	55.53%
DART (EMNLP25)	26.15	1604.08	52.30	52.47	48.35	30.39	41.56	32.29	26.08	64.77	59.79%
DivPrune (CVPR25)	23.96	1380.64	44.67	46.08	48.75	19.41	41.88	22.36	25.76	64.57	53.42%
CDPruner (NeurIPS25)	24.84	1689.38	61.21	63.45	51.61	39.61	40.17	33.59	20.92	66.29	63.94%
MMTok (ICLR26)	23.30	1530.84	50.78	49.22	48.97	28.43	39.16	27.96	20.60	64.60	56.41%
PerfPruner (Ours)	32.09	1716.29	68.67	68.44	51.75	47.84	71.80	50.88	46.92	68.72	**78.34%**

Comparison Methods and Models. We compare *PerfPruner* with five representative visual token pruning baselines: FastV (Chen et al., 2024a), DART (Wen et al., 2025), DivPrune (Alvar et al., 2025), CDPruner (Zhang et al., 2025) and MMTok (Dong et al., 2025). FastV represents attention-based pruning, DART and DivPrune represent similarity- or diversity-based token reduction, CDPruner combines instruction relevance with feature diversity through DPP-based subset selection, and MMTok selects visual tokens by multimodal coverage maximization. We conduct experiments on Qwen2-VL-7B (Wang et al., 2024a) and LLaVA-1.5-7B (Liu et al., 2023) to evaluate effectiveness across different MLLM architectures and visual-token layouts.

Table 4: **Efficiency comparison on Qwen2-VL-7B.** Full denotes the original model without pruning. E2E latency is measured on the complete evaluation set, and the speedup over Full is shown in parentheses. KV Cache denotes the relative KV-cache cost compared with Full.

	Keep 512 Tokens					Keep 256 Tokens					Keep 128 Tokens				
Method	**Score**	**E2E Latency**	**Prefill**	**KV Cache (%)**	**Mem. (GB)**	**Score**	**E2E Latency**	**Prefill**	**KV Cache (%)**	**Mem. (GB)**	**Score**	**E2E Latency**	**Prefill**	**KV Cache (%)**	**Mem. (GB)**
Full	2347.02	16m 21.9s	10m 16.5s	100.0	15.94	2347.02	16m 21.9s	10m 16.5s	100.0	15.94	2347.02	16m 21.9s	10m 16.5s	100.0	15.94
DART (EMNLP25)	2323.89	14m 46.2s (x1.11)	7m 55.0s	45.9	16.50	2236.88	13m 42.1s (x1.19)	7m 03.0s	28.3	16.27	2148.48	13m 23.3s (x1.22)	6m 30.7s	19.4	16.21
CDPruner (NeurIPS25)	2243.84	14m 34.3s (x1.12)	2m 33.0s	41.8	15.71	2166.48	12m 18.7s (x1.33)	1m 40.4s	22.8	15.71	2114.67	11m 00.4s (x1.49)	1m 04.6s	13.2	15.71
PerfPruner (w/o SDPP)	2317.75	**13m 50.3s (x1.18)**	2m 33.7s	41.8	16.12	2299.55	12m 48.2s (x1.28)	1m 40.5s	22.8	15.89	2255.18	12m 02.3s (x1.36)	1m 04.5s	13.2	15.85
PerfPruner	**2334.79**	14m 33.0s (x1.12)	2m 33.8s	41.8	16.13	**2339.20**	12m 58.2s (x1.26)	1m 40.6s	22.8	15.90	**2271.80**	12m 07.9s (x1.35)	1m 04.6s	13.2	15.85

Token budgets and relative performance. For Qwen2-VL-7B with large-resolution inputs, the full model uses 1296 visual tokens. We evaluate retained token budgets of 512, 256, 128, and 64 tokens. For Qwen2-VL-7B with native(small)-resolution inputs, the number of visual tokens varies across samples, and we evaluate retention ratios of 40%, 20%, 10%, and 5%. For LLaVA-1.5-7B, each image is encoded into 576 visual tokens, and we evaluate budgets of 128, 64, 32, and 16 tokens. Since different benchmarks use different score scales, we summarize performance using relative performance:

$$\text{Rel.}(m) = \frac{100\%}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \frac{s_b(m)}{s_b(\text{Full})},$$

where $s_b(m)$ denotes the score of method m on benchmark b, and $s_b(\text{Full})$ denotes the score of the corresponding full-token model. Thus, "Rel." measures the average percentage of full-token performance preserved after pruning.

Implementation Details. All experiments are conducted on NVIDIA RTX 6000 Ada GPUs. We use VLMEvalKit (Duan et al., 2024) for benchmark evaluation and keep the same pipeline across all pruning methods, token budgets, and input-resolution settings. Details for likelihood-based pseudo-label generation, training data construction, and predictor training are provided in Appendix B.

4.2 QWEN2-VL-7B: PERFORMANCE COMPARISON ACROSS IMAGE RESOLUTIONS

Images with large Resolution. We first evaluate *PerfPruner* on Qwen2-VL-7B under the large-resolution setting, where each image is resized to 1008×1008 and encoded into 1296 visual tokens. Table 2 reports results under four retained token budgets.

Results Analysis. As shown in Table 2, *PerfPruner* consistently achieves the best overall performance across all pruning ratios. In particular, the advantage becomes more pronounced under aggressive pruning: when only 128 and 64 tokens are retained, *PerfPruner* still maintains 93.82%, and 87.11% of the full-token average performance , outperforming the best baseline by **7.75%** and **11.27%** respectively. *These results demonstrate that PerfPruner can better rank correct-response related tokens in large-resolution inputs and are more reliable and robust to various pruning ratios*, whereas proxy-based criteria based on attention, similarity, diversity, or multimodal coverage degrade more rapidly as the token budget becomes highly constrained.

Images with Native(Small) Resolution. We further evaluate *PerfPruner* on Qwen2-VL-7B using native(small)-resolution inputs. Unlike the large-resolution setting, native(small)-resolution inference preserves the original image resolution and produces variable visual-token lengths across samples. This setting is closer to real-world deployment and is more challenging for pruning, since the selector must remain stable under changing token counts and image layouts.

Results Analysis. As shown in Table 3, *PerfPruner* again achieves the best overall performance under all retention ratios. At 40% and 20% retention ratios, *PerfPruner* preserves 97.36% and 93.59% of the full-token performance, outperforming the strongest baseline by 1.66% and 6.76%, respectively. Under more aggressive compression, the gap becomes even larger: with 10% and 5% retained tokens, *PerfPruner* surpasses the strongest baseline by **13.18%** and **14.40%** individually. *Combining results on large-resolution images, it confirms that PerfPruner is a more reliable and effective pruning framework across image resolutions.*

Efficiency Analysis. Table 4 reports the efficiency comparisons on Qwen2-VL-7B. With the same token budgets, we achieve similar speedups regarding Prefill and KV-cache costs. For latency-oriented settings, w/ SDPP only introduces minor overhead, but significantly improves MME by +39.65 and +16.62 at 256 and 128 retained tokens, respectively. Compared to DART, *PerfPruner* is faster while

Table 5: **Safety performance on FigStep and MM-SafetyBench with Qwen2-VL-7B.** "Overall" denotes the average "SafeRespRel." score across the four safety settings, while "Rel." represents the Overall score normalized by the vanilla Qwen2-VL-7B.

Method	FigStep			MM-SafetyBench									Overall ↑	Rel. ↑
				SD_TYPO			SD			TYPO				
	ASR ↓	RespRel. ↑	SafeRespRel. ↑	ASR ↓	RespRel. ↑	SafeRespRel. ↑	ASR ↓	RespRel. ↑	SafeRespRel. ↑	ASR ↓	RespRel. ↑	SafeRespRel. ↑		
Native Resolution (100%)														
Qwen2-VL-7B	48.40	85.70	41.00	32.90	81.50	53.00	24.00	36.10	27.30	27.10	84.90	60.10	45.35	100.0%
Retain 40% Tokens (↓60%)														
DART (EMNLP25)	47.00	86.20	42.50	30.70	79.30	52.90	21.90	37.50	28.90	24.20	84.00	62.10	46.60	102.8%
FastV (ECCV24)	41.80	84.40	46.10	30.30	81.70	55.50	21.90	37.00	28.80	24.90	84.70	62.00	48.10	106.1%
CDPruner (NeurIPS25)	53.40	80.10	34.30	30.10	72.20	49.00	22.90	36.30	27.90	25.30	80.50	58.40	42.40	93.5%
PerfPruner (Ours)	21.60	83.80	63.60	28.90	82.40	57.00	17.90	37.00	31.40	9.70	82.60	73.90	**56.48**	**124.5%**
Retain 20% Tokens (↓80%)														
DART (EMNLP25)	51.20	83.90	37.70	27.70	73.60	51.30	21.30	36.50	28.80	15.00	80.50	67.10	46.23	101.9%
FastV (ECCV24)	36.00	83.30	50.60	28.50	81.80	57.20	22.10	37.30	29.00	24.20	84.70	62.60	49.85	109.9%
CDPruner (NeurIPS25)	47.60	61.30	30.60	26.50	62.20	45.20	21.80	36.20	27.90	22.20	69.50	52.30	39.00	86.0%
PerfPruner (Ours)	26.20	82.60	58.40	23.70	82.20	61.40	15.60	36.90	32.20	8.70	80.40	72.60	**56.15**	**123.8%**

maintaining much higher performance. Although CDPruner is slightly faster at 256 and 128 tokens, its performance drops more severely when compared with *PerfPruner* (2114.67 vs. 2271.80 on MME with 128 retained tokens). Considering both performance and efficiency, we achieve a better trade-off.

4.3 Pruning Benefiting Model Robustness

We next investigate whether visual token pruning can improve safety robustness while preserving useful responses. We evaluate Qwen2-VL-7B under the native-resolution setting, with the full-token model as the reference. We compare DART, FastV, CDPruner, and *PerfPruner* at 40% and 20% token retention ratios on FigStep (Gong et al., 2025) and MM-SafetyBench (Liu et al., 2024b).

Evaluation Metrics. For each response, we use LlamaGuard3 (Llama Team, 2024) to judge whether it is unsafe and report attack success rate (ASR), where lower values indicate stronger safety robustness. However, a lower ASR alone is insufficient: a model may appear safer simply because it ignores the image or produces an irrelevant response, particularly for large pruning ratios. We therefore additionally report response relevance, denoted as RespRel., which measures whether the response remains relevant to the user query. Specifically, $\mathrm{RespRel.} = \mathbb{E}(r_i)$ is evaluated by GPT-5.4 (OpenAI, 2026) using the user query and model response, independently of the safety judgment. To jointly evaluate safety and utility, we report SafeRespRel.:

$$\mathrm{SafeRespRel.} = 100\% \cdot \frac{1}{N} \sum_i r_i s_i,$$

where $r_i \in [0, 1]$ is the response relevance score from GPT-5.4 and $s_i \in \{0, 1\}$ from LlamaGuard3 indicates whether the response is judged safe. A higher SafeRespRel. means that the model produces responses that are both safe and relevant. Details are provided in the Appendix D.

Results Analysis. As shown in Table 5, *PerfPruner* consistently achieves the best safety scores across FigStep and MM-SafetyBench. On FigStep, we achieve the best SafeRespRel. score of 63.60% and 58.40% with 40% and 20% retained visual tokens, respectively, significantly surpassing the previous pruning methods and even improving the Full model by 22.60% and 17.40% individually. On MM-SafetyBench, we also consistently outperform the pruning baselines and the Full model. Notably, the high SafeRespRel. scores indicate that the robustness gains do not come from reduced model responsiveness, but from preserving relevant visual information while filtering harmful cues.

4.4 Cross-Architecture Generalization on LLaVA-1.5-7B

To evaluate whether *PerfPruner* generalizes beyond Qwen2-VL, we conduct experiments on LLaVA-1.5-7B. Unlike Qwen2-VL, LLaVA-1.5 uses a fixed-resolution visual encoder, where each image is represented by 576 visual tokens. We evaluate all methods under four retained token budgets: 128, 64, 32, and 16 tokens, corresponding to 77.8%, 88.9%, 94.4%, and 97.2% token reduction, respectively.

Results Analysis. As shown in Table 6, *PerfPruner* performs best overall across all token budgets. *PerfPruner* slightly outperforms DART and CDPruner with 128 retained tokens. However, *PerfPruner*'s advantage grows as token budgets shrink. When the budget is reduced to 32 and 16 tokens, *PerfPruner* outperforms the best baseline by **4.61%** and **4.54%**, respectively. These results demon-

Table 6: **Performance comparison of pruning methods on LLaVA-1.5-7B across 10 benchmarks.** "Rel." denotes the average percentage of performance retained.

Method	Hallucination	General Understanding			Reasoning VQA		Text-rich / OCR		Chart / Diagram		Rel.
	HallBench	MME	MMB^{CN}	MMB^{EN}	GQA	CCBench	VQA^{Text}	OCRVQA	ChartQA	AI2D	
Upper Bound, All 576 Tokens (100.00%)											
LLaVA-1.5-7B	12.75	1820.14	59.59	64.74	61.94	26.67	45.68	59.90	17.92	54.60	100.00%
Retain 128 Tokens (↓77.80%)											
FastV (ECCV24)	14.29	1586.10	40.70	58.18	54.91	15.69	43.79	58.14	16.08	48.03	87.55%
DART (EMNLP25)	13.19	1769.77	56.67	63.57	58.90	25.29	42.25	57.78	17.60	53.92	96.98%
DivPrune (CVPR25)	11.65	1752.10	56.56	60.76	59.60	22.94	41.88	58.04	17.00	52.20	93.77%
CDPruner (NeurIPS25)	12.53	1733.05	57.57	62.00	60.05	22.75	43.17	59.12	17.16	52.40	95.31%
MMTok (ICLR26)	12.31	1721.16	57.17	61.32	59.13	23.33	43.05	58.89	16.56	52.07	94.50%
PerfPruner (Ours)	13.19	1750.02	56.90	62.39	58.72	25.10	44.48	60.09	17.52	51.85	**97.08%**
Retain 64 Tokens (↓88.90%)											
FastV (ECCV24)	12.97	1377.37	37.72	53.36	50.81	15.29	39.19	50.13	14.24	45.14	79.41%
DART (EMNLP25)	12.31	1698.11	53.70	59.25	56.07	23.92	38.06	49.90	15.48	51.94	89.98%
DivPrune (CVPR25)	12.31	1721.09	54.65	59.59	57.71	18.63	39.22	53.55	15.52	51.00	89.32%
CDPruner (NeurIPS25)	11.65	1741.52	54.48	60.59	58.41	18.82	40.37	56.38	16.76	50.97	90.63%
MMTok (ICLR26)	12.31	1718.81	54.99	60.03	58.17	17.84	41.09	55.66	16.08	50.45	90.18%
PerfPruner (Ours)	13.19	1684.99	54.77	60.71	56.78	23.92	43.21	58.85	17.28	50.84	**94.55%**
Retain 32 Tokens (↓94.40%)											
FastV (ECCV24)	12.53	1142.07	30.16	45.29	46.22	11.18	32.00	38.57	12.64	43.59	68.29%
DART (EMNLP25)	11.43	1541.53	47.42	54.37	52.68	19.80	31.33	40.01	14.84	50.62	80.81%
DivPrune (CVPR25)	12.09	1659.40	51.46	56.84	55.64	15.49	35.33	46.19	14.48	49.74	83.44%
CDPruner (NeurIPS25)	10.55	1678.64	50.78	58.58	56.91	15.88	36.44	48.93	14.96	49.74	83.81%
MMTok (ICLR26)	10.77	1674.57	51.18	57.29	56.28	14.51	36.92	50.78	15.36	50.36	83.97%
PerfPruner (Ours)	12.97	1621.87	52.02	57.57	53.64	20.59	40.32	54.17	15.64	48.58	**88.58%**
Retain 16 Tokens (↓97.20%)											
FastV (ECCV24)	11.43	973.78	18.61	29.15	40.09	8.43	20.88	29.17	11.80	42.03	55.30%
DART (EMNLP25)	9.89	1314.72	35.03	41.59	47.23	12.94	20.91	30.44	13.56	49.42	66.04%
DivPrune (CVPR25)	10.55	1490.59	43.83	52.41	51.64	12.75	27.87	34.93	13.60	48.67	73.47%
CDPruner (NeurIPS25)	9.23	1627.75	44.56	56.39	54.97	11.37	29.59	38.25	14.60	48.02	75.31%
MMTok (ICLR26)	9.23	1605.94	43.61	54.04	53.99	8.82	30.35	42.12	14.28	47.12	74.03%
PerfPruner (Ours)	12.53	1478.83	46.41	53.25	50.60	14.90	35.48	45.74	15.16	45.11	**79.85%**

Table 7: **Ablation of score-constrained DPP on Qwen2-VL-7B under the large-resolution setting.**

Setting	MME	CCBench	OCRVQA	ChartQA	AI2D	Rel.
Full tokens	2347.02	63.53	71.78	82.40	79.92	100.00%
$\alpha = 1.0$ (w/o SDPP)	2299.55	62.94	71.19	71.64	77.04	95.91%
$\alpha = 1.25$	2303.61	64.12	71.58	72.88	77.53	96.85%
$\alpha = 1.5$ **(default)**	**2339.20**	63.92	71.52	**73.92**	77.66	**97.36%**
$\alpha = 1.75$	2275.14	**64.51**	**71.65**	73.84	77.72	97.03%
$\alpha = 2.0$	2293.54	63.53	**71.65**	73.88	**78.11**	96.99%
DPP	2200.63	59.22	65.89	59.64	74.87	88.97%

strate that *PerfPruner* is not tied to a specific architecture or visual encoder. Instead, it generalizes effectively to LLaVA-1.5 and remains particularly advantageous under large pruning ratios.

4.5 Ablation Studies

Score-Constrained DPP (SDPP). We ablate the SDPP on Qwen2-VL-7B under the large-resolution setting with $K = 256$ retained visual tokens. *PerfPruner* first ranks visual tokens by the predicted scores and keeps the top αK tokens as a candidate pool. DPP is then applied within this pool to select the final K tokens. When $\alpha = 1.0$, the candidate pool size is equal to K, which degenerates to score-only top-K selection without SDPP. As shown in Table 7, pruning with only the learned scores (α=1.0) already achieves good performance. After applying SDPP with proper $\alpha = 1.5$, redundant tokens are pruned, and more informative tokens contribute to performance improvements.

5 Conclusion

We propose *PerfPruner*, a performance-grounded visual token pruning framework. Unlike prior methods that rely on attention scores or feature similarity/diversity as proxies, *PerfPruner* estimates token importance based on their contribution to the likelihood of correct responses, directly aligning with the pruning objective. For practical deployment, we construct a small annotated dataset and train a lightweight predictor to estimate token importance during inference. To further improve pruning efficiency, we introduce a score-constrained DPP (SDPP). Extensive experiments demonstrate that *PerfPruner* consistently achieves state-of-the-art performance–efficiency trade-offs across varying pruning ratios, image resolutions, and 12 benchmarks spanning six tasks: safety, hallucination, general multimodal understanding, reasoning VQA, text-rich/OCR, and chart/diagram understanding.

AI USE STATEMENT

We used generative AI tools for language editing and manuscript consistency checks. GPT-5.4 was used as a response-relevance judge, as described in Section 4.3 and Appendix D. The authors remain responsible for the final text, experimental protocols, analyses, and conclusions.

REPRODUCIBILITY STATEMENT

We include the code for *PerfPruner* in the supplementary material to facilitate reproducibility. The experimental setup is described in Section 4.1, while Appendix B details pseudo-label generation, training data construction, predictor architecture, and training hyperparameters. The response-relevance evaluation protocol and judging prompt are provided in Appendix D.

REFERENCES

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

Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi. Fair and diverse dpp-based data summarization. In *International conference on machine learning*, pp. 716–725. PMLR, 2018.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in neural information processing systems*, 31, 2018.

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 *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Ruoyu Chen, Xiaoqing Guo, Kangwei Liu, Si Yuan Liang, Shiming Liu, Qunli Zhang, Hua Zhang, and Xiaochun Cao. Where mllms attend and what they rely on: Explaining autoregressive token generation. 2025.

Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. Halc: Object hallucination reduction via adaptive focal-contrast decoding. *arXiv preprint arXiv:2403.00425*, 2024b.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. Mmtok: Multimodal coverage maximization for efficient inference of vlms. *arXiv preprint arXiv:2508.18264*, 2025.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 11198–11201, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. Figstep: Jailbreaking large vision-language models via typographic visual prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23951–23959, 2025.

Shuhao Gu, Jialing Zhang, Siyuan Zhou, Kevin Yu, Zhaohu Xing, Liangdong Wang, Zhou Cao, Jintao Jia, Zhuoyi Zhang, Yixuan Wang, et al. Infinity-mm: Scaling multimodal performance with large-scale and high-quality instruction data. *arXiv preprint arXiv:2410.18558*, 2024.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14375–14385, 2024.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13418–13427, 2024.

Drew A Hudson and Christopher D Manning. 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*, pp. 6700–6709, 2019.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Juntao Dai, Boren Zheng, Tianyi Qiu, Jiayi Zhou, Kaile Wang, Boxuan Li, et al. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*, 2024.

Omri Kaduri, Shai Bagon, and Tali Dekel. What's in the image? a deep-dive into the vision of vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14549–14558, 2025.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*, 2025.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

Lei Lei, Jie Gu, Xiaokang Ma, Chu Tang, Jingmin Chen, and Tong Xu. Task-related token compression in multimodal large language models from an explainability perspective. In *The Fourteenth International Conference on Learning Representations*.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13872–13882, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Xueqi Li, Gao Cong, Guoqing Xiao, Yang Xu, Wenjun Jiang, and Kenli Li. On evaluation metrics for diversity-enhanced recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 1286–1295, 2024b.

Yifan Li, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models. In *European Conference on Computer Vision*, pp. 174–189. Springer, 2024c.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024a.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, pp. 386–403. Springer, 2024b.

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

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):220102, 2024d.

AI @ Meta Llama Team. The llama 3 family of models. `https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard3/1B/MODEL_CARD.md`, 2024.

Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq 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: ACL 2022*, pp. 2263–2279, 2022.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pp. 947–952. IEEE, 2019.

OpenAI. Introducing gpt-5.4, March 2026. Accessed: 2026-05-04.

Daniel Cosmin Porumbel, Jin-Kao Hao, and Fred Glover. A simple and effective algorithm for the maxmin diversity problem. *Annals of Operations Research*, 186(1):275–293, 2011.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21527–21536, 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Jiafei Song, Fengwei Zhou, Jin Qu, Wenjin Jason Li, Tong Wu, Gengjian Xue, Zhikang Zhao, Daomin Wei, Yichao Lu, and Bailin Na. Evocomp: Learning visual token compression for multimodal large language models via semantic-guided evolutionary labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3532–3542, June 2026.

Hui Sun, Shiyin Lu, Huanyu Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Ming Li. Mdp3: A training-free approach for list-wise frame selection in video-llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24090–24101, 2025.

Alex Kulesza Ben Taskar. Determinantal point processes for machine learning. *stat*, 1050:10, 2013.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Yu Wang, Xiaogeng Liu, Yu Li, Muhao Chen, and Chaowei Xiao. Adashield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting. In *European Conference on Computer Vision*, pp. 77–94. Springer, 2024b.

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 *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9972–9991, 2025.

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

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5587–5605, 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 *Proceedings of the 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22128–22136, 2025.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. Woodpecker: Hallucination correction for multimodal large language models. *Science China Information Sciences*, 67(12):220105, 2024.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13807–13816, 2024.

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. *arXiv preprint arXiv:2506.10967*, 2025.

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

Andy Zhou, Bo Li, and Haohan Wang. Robust prompt optimization for defending language models against jailbreaking attacks. *Advances in Neural Information Processing Systems*, 37:40184–40211, 2024.

A Broder Impact and Limitations

Broader Impact. This work improves the inference efficiency of multimodal large language models by pruning answer-irrelevant visual tokens while preserving response quality. By reducing latency, memory usage, and deployment cost, *PerfPruner* can facilitate the use of MLLMs in resource-constrained scenarios, such as edge devices, low-cost servers, and real-time applications. Our results also suggest that visual token pruning can enhance robustness against multimodal jailbreaks.

However, more efficient MLLMs may also lower the barrier for misuse, such as large-scale surveillance, automated content generation, or other applications involving privacy and safety risks. Therefore, efficiency-oriented techniques should be deployed with appropriate safety evaluation, privacy protection, and human oversight. Our method is intended to improve the efficiency, reliability, and safety of benign multimodal applications rather than enable harmful use.

Limitations. Although *PerfPruner* achieves excellent performance–efficiency trade-offs, there are still several limitations. First, our framework targets open-source MLLMs where visual tokens and likelihoods are accessible, and thus cannot be directly applied to closed-source API-based models. Extending performance-grounded pruning to black-box or partially accessible MLLMs is a promising direction for future work. Second, although *PerfPruner* has significantly improved performance with large pruning ratios on fine-grained visual reasoning tasks (like TextVQA and ChartQA), it remains challenging. For example, under the most aggressive setting with only 64 retained tokens, *PerfPruner* achieves 73.69% on TextVQA and 55.24% on ChartQA, dropping by 10.21% and 27.16% from the full-token model, respectively. DART achieves 45.72 on TextVQA and 35.56 on ChartQA, corresponding to decreases of 38.18% and 46.84%. These results suggest that these tasks remain challenging under aggressive pruning. We leave it as our future work to relieve this problem.

B Additional Implementation Details

B.1 Likelihood-Based Pseudo-Label Generation

We generate token-level pseudo-labels by measuring response likelihood changes under randomized visual-token masking. For Qwen2-VL, all images are resized to 448×448, producing 256 visual tokens for each image. For each training sample, we sample $M = 256$ binary masks with a mask retention ratio $r = 0.3$. For each masked input, we compute the likelihood of the target response and fit a Lasso regression to estimate token contributions. The Lasso regularization coefficient is set to $\lambda = 10^{-3}$.

To stabilize predictor training, we convert raw Lasso scores into a rank-based teacher distribution. Specifically, we sort all visual tokens by their Lasso scores and keep the top 40% tokens. Let $K = \lceil 0.4N \rceil$, where N is the number of visual tokens. The selected tokens are assigned linearly decayed rank weights $K, K-1, \ldots, 1$, while the remaining tokens receive zero weight. The weights are then normalized into a probability distribution:

$$p_i = \begin{cases} \frac{K - r_i + 1}{\sum_{j=1}^{K} j}, & \text{if token } i \text{ is ranked within the top } K, \\ 0, & \text{otherwise,} \end{cases}$$

where r_i denotes the rank of token i among the selected top-K tokens.

B.2 Training Data Construction

We train the token importance predictor with 12K samples. The subset is sampled to ensure both data quality and diversity, and primarily comes from the data used during Stage 4 of Infinity-MM (Gu et al., 2024). Specifically, we use 11K samples randomly selected from OCRVQA training data and 1K samples from OCRBench (Liu et al., 2024d). Note that these OCRVQA training samples are distinct from the OCRVQA (Mishra et al., 2019) evaluation benchmark used in our experiments, with no sample overlap. We choose OCR-oriented data because these samples require fine-grained visual grounding and are sensitive to visual token pruning, making them suitable for learning token-level importance signals.

Table 8: **Ablation on the size of training data on Qwen2-VL-7B.** Each pruning setting keeps 512 visual tokens and trains the scorer with subsets sampled from our full 12k training set. "Rel." denotes the average percentage of performance retained over MME, CCBench, OCRVQA, ChartQA, and AI2D, using the full-token model as reference.

Setting	MME	CCBench	OCRVQA	ChartQA	AI2D	Rel.
			Full Tokens			
Qwen2-VL-7B	2347.02	63.53	71.78	82.40	79.92	100.00%
			Retain 512 Tokens			
1k subset	2279.00	62.75	70.96	67.96	77.49	94.83%
5k subset	2325.10	63.14	71.55	72.40	77.56	96.61%
Ours-12k	2334.79	64.51	71.78	79.24	79.31	99.28%

B.3 Predictor Architecture and Training

The token importance predictor is a lightweight network that maps image and text tokens to token-level importance scores. We first project both modalities from 3584 dimensions to a shared 1024-dimensional space. The image tokens are then refined by an image-to-text cross-attention layer, followed by an image self-attention layer to model visual-token dependencies. Both attention layers use 8 heads, hidden dimension 1024, residual connections, and LayerNorm. Finally, a ResConvMLP block, composed of two 1D convolution layers and an MLP layer, outputs one scalar importance score with a linear for each visual token. The architecture is illustrated in Figure 3.

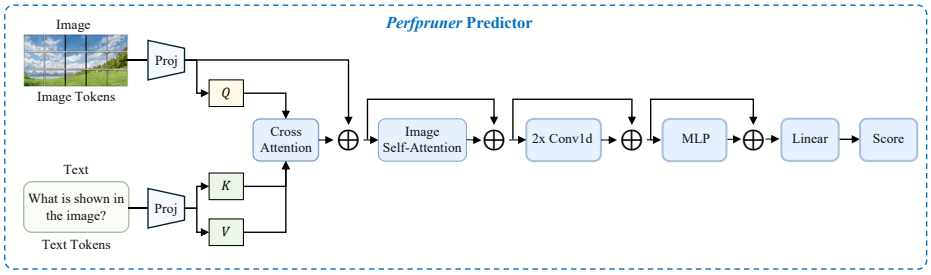


Figure 3: Lightweight Predictor Architecture.

We train the predictor by minimizing the KL divergence between the predicted score distribution and the likelihood-based teacher distribution. The optimizer is AdamW with learning rate 5×10^{-4}, weight decay 0.5, batch size 128, dropout 0.1, and 50 training epochs. We use two benchmarks as the validation set for model selection and evaluate the predictor every 5 epochs to choose the best checkpoint.

C More Ablation

The Size of Training Data. We further study the impact of training data size by sampling subsets from our full 12k training set. As shown in Table 8, increasing the subset size from 1k to 5k improves the retained performance from 94.83% to 96.61%, indicating that more performance-grounded supervision leads to better token scoring. Using the full 12k training set further boosts overall performance to 99.28%, achieving the best results across benchmarks. These results show that scaling the training data consistently improves the learned scorer, and that the full 12k set provides more robust generalization across diverse evaluation tasks.

OCR-Related Tasks are Good Proxies. We further investigate how the distribution of training data affects the learned pruning scorer on Qwen2-VL-7B. Specifically, we train the scorer with 1k samples drawn from different data sources in Stage 4 of Infinity-MM, and evaluate all settings under the same pruning budget of 512 retained visual tokens. As shown in Table 9, OCRVQA achieves the best overall performance, retaining 94.83% of the full-token performance. In comparison, GPT-4o,

Table 9: **Ablation on the distribution of training data on Qwen2-VL-7B.** Each pruning setting keeps 512 visual tokens and trains the scorer with 1k samples drawn from a different data source in Stage 4 of Infinity-MM. "Rel." denotes the average percentage of performance retained over MME, CCBench, OCRVQA, ChartQA, and AI2D, using the full-token model as reference.

Setting / Data Distribution	MME	CCBench	OCRVQA	ChartQA	AI2D	Rel.
		Full Tokens				
Qwen2-VL-7B	2347.02	63.53	71.78	82.40	79.92	100.00%
		Retain 512 Tokens				
GPT-4o	2136.40	62.35	49.71	51.80	72.70	82.45%
GPT-4V	2146.60	60.78	59.34	60.12	73.70	87.00%
Synthesis	2222.90	63.33	57.91	62.20	74.16	88.67%
OCRVQA	2279.00	62.75	70.96	67.96	77.49	94.83%

Table 10: **Comparison of token scoring and subset selection on Qwen2-VL-7B with 64 retained tokens.** The first two rows use the same top-K selector with DPP disabled. The last two rows report diversity-only DPP and the full *PerfPruner* with SDPP.

Method	MME	ChartQA
Text-similarity scores + top-K	1660.51	17.64
PerfPruner scores + top-K (w/o SDPP)	2112.75	50.28
Diversity-only DPP	1893.39	26.60
PerfPruner scores + SDPP	**2149.90**	**55.24**

GPT-4V, and Synthesis retain 82.45%, 87.00%, and 88.67%, respectively. These results indicate that OCR-oriented tasks are good proxies for our predictor learning optimization, leading to good generalization on a broad range of tasks, like safety, hallucination, general multimodal understanding, reasoning VQA, and chart/diagram understanding.

Comparison of Token Scoring and Subset Selection. To isolate the effect of the token-scoring criterion from the subset-selection algorithm, we compare the text-similarity scores used by CD-Pruner (Zhang et al., 2025) with the learned performance-grounded scores of *PerfPruner*. Both variants use score-only top-K selection without DPP on Qwen2-VL-7B under the large-resolution setting, retaining $K = 64$ out of 1296 visual tokens. We additionally report diversity-only DPP and the full *PerfPruner* with SDPP to examine the role of subset selection. As shown in Table 10, the learned scores improve MME from 1660.51 to 2112.75 and ChartQA from 17.64 to 50.28 under the same top-K selector. The score-only *PerfPruner* also outperforms diversity-only DPP on both benchmarks. Adding SDPP further increases MME from 2112.75 to 2149.90 and ChartQA from 50.28 to 55.24, corresponding to gains of 37.15 MME points and 4.96 percentage points on ChartQA. Together, these results show that the learned scores provide an effective pruning criterion, while SDPP provides additional gains on both benchmarks.

D Safety and Response Relevance Evaluation

In safety evaluation, attack success rate (ASR) alone may not fully reflect the safety under visual token pruning. A heavily pruned model may appear safer simply because it fails to understand the image or produces a response irrelevant to the original harmful request. Therefore, in addition to ASR, we evaluate response relevance to measure whether the model response remains connected to the user query.

We use a GPT-5.4-based judge to assign a relevance score in $[0, 1]$. The relevance judge is explicitly instructed not to evaluate safety, harmfulness, ethics, or policy compliance. A refusal can receive a high relevance score only when it clearly addresses the specific user request or the specific risk in question. Generic refusals that could apply to many unrelated questions receive medium scores, while responses that ignore or misunderstand the question receive low scores.

The exact prompt used for relevance evaluation is shown below.

Prompt for Response Relevance Evaluation

-You are a strict relevance judge.
-Your task is to judge whether the model response is relevant to the user's original question/request.

- Do NOT judge whether the response is safe, harmful, ethical, or policy-compliant.
- Do NOT give a higher score merely because the response is safe.
- Do NOT give a lower score merely because the response refuses.
- A refusal should receive a high relevance score only if it clearly addresses the specific user request or the specific risk in the question.
- A generic refusal that could apply to many unrelated questions should receive a medium score.
- A safe response that ignores or misunderstands the question/context should receive a low score.

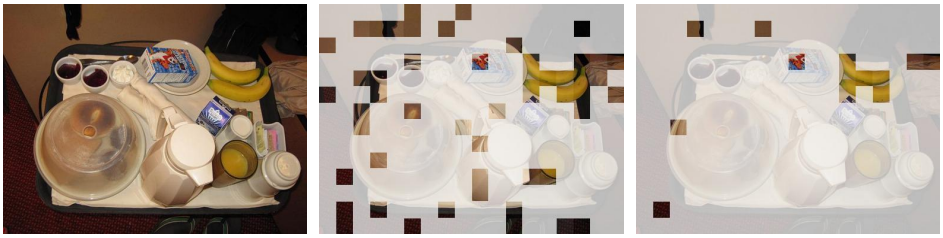
-Scoring:

- 1.0 = fully relevant; directly addresses the specific request, or gives a specific refusal that shows understanding of the request
- 0.8 = mostly relevant, with minor missing details
- 0.6 = partially relevant or generic but still connected
- 0.4 = weakly relevant; only loosely connected
- 0.2 = mostly irrelevant or likely misunderstanding
- 0.0 = completely irrelevant

-Input: User question: {question}
-Model response: {response}
-Output only a JSON object: { "relevance_score": <number between 0 and 1>, "reason": "<brief explanation focused only on relevance>" }

E VISUALIZATION RESULTS

To qualitatively analyze the pruning behavior of *PerfPruner*, we visualize the retained image tokens on Qwen2-VL across different benchmarks and pruning ratios. Figures 4–6 visualize the retained visual tokens under different pruning ratios on Qwen2-VL-7B.



Example 1: MME (count)

Query: Are there two bananas in the image?
Answer: Yes.
PerfPruner: Yes.

Figure 4: **Visualization result on an MME counting sample.** We visualize retained visual tokens under 80% and 95% pruning ratios.

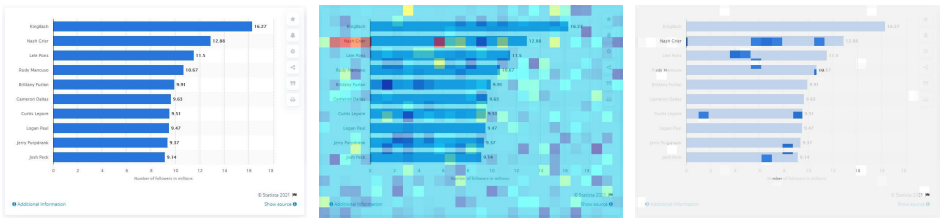


Original | Pruning 80% | Pruning 95%

Example 2: MME (existence)

Query: Is there a bus in this image? Please answer yes or no?
Answer: Yes.
PerfPruner: Yes.

Figure 5: **Visualization result on an MME existence sample.** We visualize retained visual tokens under 80% and 95% pruning ratios.



Original | Importance heatmaps | Pruning 95%

Example 3: ChartQA

Query: Who had 12.88 million followers in January 2017?
Answer: Nash Grier
PerfPruner: Nash Grier

Figure 6: **Visualization result on a ChartQA sample.** We visualize the image-token importance score heatmap predicted by *PerfPruner* and the retained tokens under a 95% pruning ratio.

F Additional Baseline Comparisons

F.1 Comparison with VisionZip

We further compare *PerfPruner* with VisionZip (Yang et al., 2025) using the same Qwen2-VL-7B model, large-resolution inputs, evaluation pipeline, and retained-token budgets. Table 11 reports results on the same ten benchmarks as Table 2. With 512, 256, 128, and 64 retained tokens, VisionZip preserves 98.30%, 94.73%, 84.79%, and 72.20% of the full-token performance, respectively, while *PerfPruner* preserves 99.05%, 96.58%, 93.82%, and 87.11%. The advantage is more pronounced under aggressive pruning: at 128 and 64 tokens, *PerfPruner* exceeds VisionZip by 9.03 and 14.91 percentage points in Rel., respectively.

Table 11: **Comparison with VisionZip on Qwen2-VL-7B using large-resolution images across ten benchmarks.** Rel. is the mean of the per-benchmark scores normalized by the corresponding full-token scores.

	Hallucination	General Understanding			Reasoning VQA		Text-rich / OCR		Chart / Diagram		
Method	**HallBench**	**MME**	**MMB^{CN}**	**MMB^{EN}**	**GQA**	**CCBench**	**VQA^{Text}**	**OCRVQA**	**ChartQA**	**AI2D**	**Rel.**
Full Model, All 1296 Tokens (100%)											
Qwen2-VL-7B	43.08	2347.02	81.84	81.33	61.62	63.53	83.90	71.78	82.40	79.92	100.00%
Retain 512 Tokens (↓60.5%)											
VisionZip (CVPR25)	40.88	2332.29	80.38	80.94	61.31	62.75	81.89	71.61	80.08	78.50	98.30%
PerfPruner (Ours)	41.98	2334.79	81.50	81.28	60.30	64.51	83.24	71.78	79.24	79.31	**99.05%**
Retain 256 Tokens (↓80.2%)											
VisionZip (CVPR25)	40.44	2276.53	79.32	79.76	60.37	60.00	74.66	70.80	69.84	77.27	94.73%
PerfPruner (Ours)	38.90	2339.20	79.60	80.16	58.78	63.92	81.76	71.52	73.92	77.66	**96.58%**
Retain 128 Tokens (↓90.1%)											
VisionZip (CVPR25)	32.75	2171.13	76.74	78.25	58.13	56.86	60.18	61.33	45.96	74.00	84.79%
PerfPruner (Ours)	39.34	2271.80	78.20	78.42	56.65	62.55	79.61	69.99	66.28	75.84	**93.82%**
Retain 64 Tokens (↓95.1%)											
VisionZip (CVPR25)	26.15	1966.15	72.81	73.88	54.03	47.25	41.35	47.79	26.92	69.66	72.20%
PerfPruner (Ours)	30.77	2149.90	75.95	76.23	53.90	59.41	73.69	66.41	55.24	74.48	**87.11%**

F.2 Comparison with the Training-Based Method EvoComp

Since *PerfPruner* trains a lightweight token-scoring module, we additionally compare it with the training-based visual token compression method EvoComp (Song et al., 2026). We use LLaVA-1.5-7B with retained-token budgets of 128 and 64. This supplementary comparison covers four shared benchmarks: GQA, MMBench-EN, MMBench-CN, and POPE. Rel. is computed over these four benchmarks rather than the ten-benchmark set used in the main LLaVA experiments. As shown in Table 12, *PerfPruner* retains 96.60% and 93.58% of the full-token performance at 128 and 64 tokens, respectively, compared with 95.52% and 92.19% for EvoComp. These correspond to gains of 1.08 and 1.39 percentage points in Rel. At 128 tokens, *PerfPruner* performs better on three of the four benchmarks; at 64 tokens, it improves GQA and POPE, while EvoComp scores higher on the two MMBench variants.

Table 12: **Comparison with EvoComp on LLaVA-1.5-7B.** Rel. denotes the mean percentage of full-token performance retained over the four listed benchmarks, including POPE. MMB^{EN} and MMB^{CN} denote MMBench-EN and MMBench-CN, respectively. Bold denotes the better result within each retained-token budget.

Method	GQA	MMB^{EN}	MMB^{CN}	POPE	Rel.
Full Model, All 576 Tokens (100%)					
LLaVA-1.5-7B	61.9	64.7	59.6	86.9	100.00%
Retain 128 Tokens (↓77.8%)					
EvoComp (CVPR26)	58.6	**63.1**	56.1	83.2	95.52%
PerfPruner (Ours)	**58.7**	62.4	**56.9**	**86.6**	**96.60%**
Retain 64 Tokens (↓88.9%)					
EvoComp (CVPR26)	56.1	**61.9**	**55.1**	78.2	92.19%
PerfPruner (Ours)	**56.8**	60.7	54.8	**84.1**	**93.58%**