RobustPruner: Decoupled Relevance and Uncertainty for Efficient Visual Token Pruning in MLLMs

Hanwei Zhu[*1] **Junhan Fu**[*2] **Xi Zhang**[1✉] **Jiamang Wang**[3] **Weisi Lin**[1✉]
[1]Nanyang Technological University [2]The Chinese University of Hong Kong (Shenzhen)
[3]Alibaba Group
h4nwei.zhu@gmail.com, junhanfu@link.cuhk.edu.cn

https://oscakywalker.github.io/RobustPruner_2026/

Abstract

Multimodal large language models (MLLMs) represent images as long sequences of visual tokens, making inference costly and often redundant. The central challenge is to prune these tokens aggressively without removing evidence required for reliable reasoning. Existing pruning methods typically rank tokens by estimated importance or redundancy, implicitly assuming that low-scored tokens are safe to discard. This assumption is fragile when small, occluded, or fine-grained visual cues are decisive and stable, and it becomes especially brittle under semantics-preserving prompt perturbations. We present **RobustPruner**, a decoder-integrated framework for prompt-robust and uncertainty-aware visual token pruning. RobustPruner predicts query-conditioned relevance, uses a determinantal point process (DPP) to construct a diverse candidate subset, and then applies uncertainty-guided refinement within that subset. By decoupling relevance and uncertainty, the method preserves ambiguous but potentially decisive evidence while remaining computationally efficient. Inserted into intermediate decoder layers, RobustPruner performs one-shot pruning and reduces KV cache and memory costs. Across three open-source MLLMs and diverse vision-language benchmarks, RobustPruner consistently achieves a stronger accuracy–efficiency–robustness trade-off than prior pruning baselines, with especially clear gains on fine-grained and text-rich tasks. On Qwen2.5-VL-7B, with only 10% of the original graph visual tokens remained, RobustPruner retains 91.13% of the original performance while reducing GPU memory, prefill FLOPs, and KV cache size to 91.43%, 50.91%, and 20.57% of the baseline.

1 Introduction

Multimodal large language models (MLLMs), such as Flamingo [1], BLIP-2 [2], and LLaVA [3], have significantly advanced general-purpose vision-language reasoning. Their effectiveness relies on representing an image as a sequence of visual tokens that are jointly processed with textual tokens. While expressive, this design is computationally expensive: a single image can introduce hundreds or thousands of visual tokens, increasing prefilling latency, attention computation, and key–value (KV) cache memory during decoding [4, 5]. These costs become increasingly prohibitive in high-resolution, multi-image, and latency-sensitive applications. Therefore, visual token pruning has become a central problem for efficient MLLM deployment.

Prior work has shown that transformer representations contain substantial token redundancy. In vision backbones, DynamicViT [6] and ToMe [7] prune or merge tokens with limited accuracy degradation.

[*]Equal contribution
✉ Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

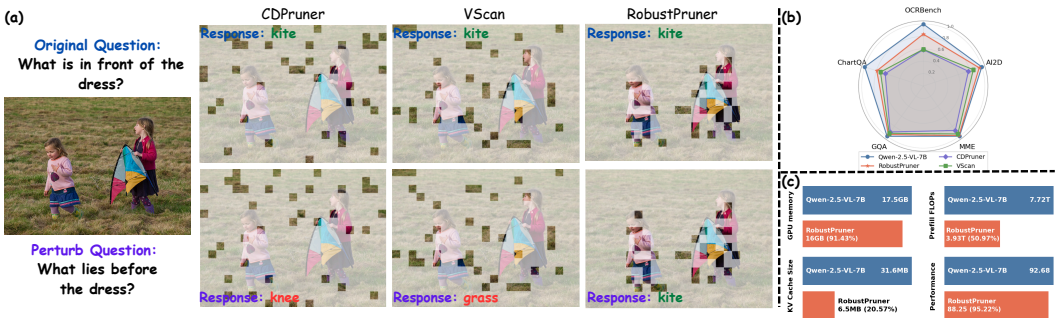

Figure 1: **Motivation and overview of RobustPruner. (a)** Under *semantics-preserving prompt perturbations*, prior *attention-based* and *similarity-based* pruning methods may retain inconsistent visual evidence for semantically equivalent prompts, leading to failures when rephrasings remove task-critical tokens. **RobustPruner** preserves relevant and diverse tokens with low prompt-induced uncertainty across prompt formulations, yielding more stable token selection and predictions. **(b)** Robustness-aware accuracy comparison. For each question, accuracy is averaged over the first four semantically equivalent prompt perturbations aggregated across multiple vision-language benchmarks. **(c)** Efficiency comparison on Qwen2.5-VL-7B.

In MLLMs, FastV [4], SparseVLM [8], and CoViPAL [5] extend this idea using attention patterns, similarity measures, text-guided saliency, or learned contextual scores. Although effective in reducing computation, existing methods remain brittle under careful evaluation and can even underperform simple baselines in challenging settings [9]. We argue that this brittleness stems from a common assumption: most pruning methods formulate token selection as an *importance-ranking* problem, implicitly treating low-scoring or trivial tokens as dispensable.

This assumption is often inadequate for vision-language reasoning. Attention-based pruning tends to overemphasize dominant responses and may therefore retain redundant visual evidence, whereas similarity-based pruning can fail to adapt reliably to the fine-grained semantics of the current instruction [10]. More importantly, tokens associated with small objects, subtle attributes, or contextual background cues may receive uncertain scores even though they are decisive for answering the query. This limitation becomes particularly pronounced under semantics-preserving prompt perturbations: prompts with the same meaning can induce different pruning masks, causing the model to preserve task-critical evidence under one phrasing but discard it under another. As shown in Fig. 1(a), both attention-based [11] and similarity-based [10] pruning may answer the original prompt correctly when the relevant visual token is retained, yet fail under an equivalent rephrasing when that token is removed. These observations reveal a central gap in existing methods: token importance is not equivalent to token dispensability. Effective pruning should therefore preserve not only highly salient tokens, but also *relevant and potentially decisive evidence with low prompt-induced uncertainty*.

These observations motivate uncertainty-aware visual token pruning. Although uncertainty estimation provides a principled framework for characterizing ambiguous or weakly supported predictions [12, 13], incorporating it into MLLM token pruning remains challenging. Effective pruning requires a retained subset that is *query-conditioned*, because token relevance depends on the instruction, while remaining stable under semantics-preserving prompt perturbations, since equivalent phrasings can otherwise induce inconsistent pruning masks and eliminate subtle yet task-critical evidence.

To address this gap, we propose **RobustPruner**, a plug-and-play framework for **prompt-robust and uncertainty-aware visual token pruning** that decouples *relevance* and *uncertainty*. RobustPruner first predicts query-conditioned relevance scores and then applies a Determinantal Point Process (DPP) to construct an informative and diverse candidate subset, suppressing redundant visual evidence while preserving alignment with the instruction. It subsequently performs uncertainty-guided refinement to retain *prompt-stable and potentially decisive* tokens that confidence-driven pruning would otherwise discard. Inserted into intermediate decoder layers, RobustPruner performs one-shot token pruning, reducing attention and memory costs throughout subsequent decoding. As shown in Figs. 1(b)&(c), RobustPruner achieves a stronger accuracy–efficiency–robustness trade-off than prior methods and, on Qwen2.5-VL-7B, reduces GPU memory, prefill FLOPs, and KV cache size to 91.43%, 50.97%, and 20.57% of the baseline, respectively, while retaining 95.22% of the original performance.

Our contributions are summarized as follows:

- We identify a fundamental failure mode of existing MLLM token pruning methods: they are often uncertainty-blind and sensitive to semantics-preserving prompt perturbations, which leads to inconsistent visual evidence retention and unstable predictions.
- We introduce RobustPruner, a query-conditioned and plug-and-play pruning module that decouples relevance estimation, DPP-based diversity selection, and uncertainty-guided refinement, thereby preserving task-critical visual evidence with low prompt-induced uncertainty under aggressive compression.
- Across three open-source MLLMs and twelve benchmarks, RobustPruner consistently achieves a stronger accuracy–efficiency–robustness trade-off than strong pruning baselines, while substantially reducing inference cost and preserving high task performance.

2 Empirical Analysis

Herein, we examine whether the pruning signals commonly used in MLLMs remain stable under *semantics-preserving textual perturbations*, an invariance-style evaluation protocol closely related to behavioral testing and prompt-sensitivity analysis [14, 15, 16]. Existing methods typically rely on attention, similarity, or learned importance scores, and recent work has improved token reduction by refining pruning stages, layer selection, and query-conditioned relevance estimation [11, 17, 18]. However, these methods primarily optimize importance and efficiency while largely overlooking the stability of pruning evidence across semantically equivalent prompt formulations. This perspective exposes uncertainty as a distinct and underexplored dimension of token pruning: a volatile relevance score need not indicate irrelevance; it may instead reflect ambiguous yet task-critical visual evidence.

Preliminaries. Given an image I and a query Q, let $V_\ell = [v_{\ell,1}, \dots, v_{\ell,N}] \in \mathbb{R}^{N\times d}$ and $T_\ell = [t_{\ell,1}, \dots, t_{\ell,M}] \in \mathbb{R}^{M\times d}$ denote the visual and textual hidden states at decoder layer ℓ. Visual token pruning seeks a subset $Y_\ell \subseteq \{1, \dots, N\}$ with $|Y_\ell| = m \ll N$ such that decoding with $\widetilde{V}_\ell = \{v_{\ell,i} \mid i \in Y_\ell\}$ preserves downstream utility under a fixed token budget. Formally, the generic objective is formulated as:

$$Y_\ell^\star = \underset{Y\subseteq\{1,\dots,N\},|Y|=m}{\arg\max} \mathcal{F}_\ell(Y; I, Q), \tag{1}$$

where $\mathcal{F}_\ell$ balances task informativeness and computational efficiency. In practice, most methods instantiate this objective through a query-conditioned importance vector $\phi_\ell(I, Q) = [\phi_{\ell,1}, \dots, \phi_{\ell,N}] \in \mathbb{R}^N$ and retain the top-m tokens induced by ϕ_ℓ. To assess the stability of this pruning evidence, we construct five semantics-preserving textual perturbations $\mathcal{P}(Q) = \{Q^{(1)}, Q^{(2)}, Q^{(3)}, Q^{(4)}, Q^{(5)}\}$ using Qwen3-VL-Plus [19]. Appendix B details the construction procedure and lists the five prompt templates. Let $\phi_{\ell,i}^{(p)} = \phi_{\ell,i}(I, Q^{(p)})$ denote the importance score of token i under perturbation $Q^{(p)}, p \in \{1, 2, 3, 4\}$. The fifth perturbation is used only in robustness validation and does not participate in uncertainty construction. We define the token-level pruning uncertainty as the variance of the perturbed scores:

$$u_{\ell,i} = \mathrm{Var}_{Q^{(p)}\in\mathcal{P}(Q)}\big[\phi_{\ell,i}^{(p)}\big], \qquad i = 1, \dots, N, \tag{2}$$

so that smaller $u_{\ell,i}$ values correspond to more stable importance estimates under semantically equivalent prompt formulations, making token i a more reliable candidate for retention under pruning. Accordingly, lower-uncertainty tokens provide a more dependable basis for downstream prediction.

Attention-Based Uncertainty in MLLMs. Attention-based pruning uses aggregated text-to-vision cross-attention as a query-conditioned importance signal. In Fig. 2(a), the attention-based importance map is spatially diffuse and biased toward visually salient regions, whereas the corresponding uncertainty map is concentrated on thin boundaries, small structures, and partially occluded content. Fig. 2(b) further shows that uncertainty is not a monotonic function of attention magnitude: only a subset of attended tokens exhibits large instability across semantically equivalent prompt rephrasings. This separation explains why attention alone is insufficient for reliable pruning. Tokens with high relevance but small uncertainty provide stable evidence for retention, whereas large-uncertainty tokens expose the prompt sensitivity that makes attention-only pruning brittle.

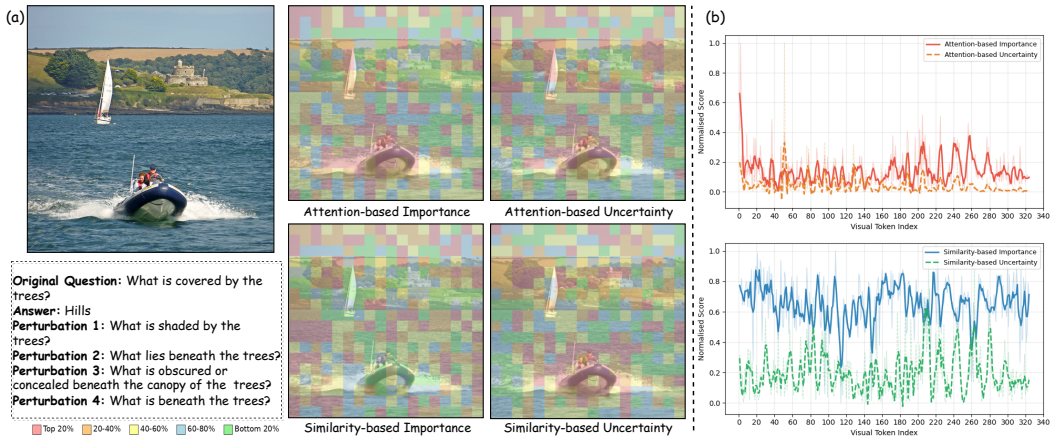

Figure 2: **Importance and uncertainty under semantics-preserving textual perturbations. (a)** Example image with attention-based and similarity-based token importance and uncertainty maps. While importance highlights query-relevant regions, uncertainty localizes tokens whose estimated relevance is unstable under equivalent rephrasings. High-importance and low-uncertainty regions therefore correspond to stable core evidence, whereas high-uncertainty regions mark visually ambiguous cues whose relevance is less reliable. **(b)** Token-wise normalized importance and uncertainty scores. Uncertainty does not collapse to importance: many highly relevant tokens remain low-uncertainty, while uncertainty peaks only on a sparse subset of tokens.

Similarity-Based Uncertainty in MLLMs. Similarity-based pruning replaces raw attention with an explicit query–visual relevance score and therefore yields a sharper, more semantically aligned importance map in Fig. 2(a). However, Fig. 2(b) shows that sparse uncertainty peaks persist even when relevance estimation improves. These peaks are concentrated on secondary objects, contextual regions, and visually ambiguous cues whose estimated usefulness varies across semantically equivalent prompts. The key implication is that uncertainty is not another measure of importance, but a measure of reliability. This observation directly motivates RobustPruner: relevance and DPP-based diversity first define an informative candidate pool, and uncertainty then selects the more stable, low-uncertainty tokens for final retention.

3 Method

Given an image–query pair (I, Q), let $V_K = [v_{K,1}, \ldots, v_{K,N}] \in \mathbb{R}^{N\times d}$ and $T_K = [t_{K,1}, \ldots, t_{K,M}] \in \mathbb{R}^{M\times d}$ denote the visual and textual hidden states at decoder layer K. As illustrated in Fig. 3, RobustPruner performs one-shot pruning at layer K. It first constructs a candidate set that preserves query-conditioned relevance and diversity, and then refines this set using predictive uncertainty to satisfy the final token budget $m \ll N$.

3.1 RobustPruner

Query Anchor Token. We introduce a *Query Anchor Token* (QAT), denoted by q, to provide a compact query representation for visual token pruning. Unlike a randomly initialized auxiliary token, the QAT is initialized by the average hidden state of the instruction tokens, $q_0 = \frac{1}{M}\sum_{j=1}^{M} t_{0,j}$, and is appended after the instruction tokens during prefilling. Owing to causal self-attention, the QAT can attend to visual contexts while leaving the original token representations unchanged. After the first K decoder layers, the resulting hidden state $q_K \in \mathbb{R}^d$ serves as the compact query summary blended with visual information that is used for pruning. The QAT is discarded immediately after pruning and does not participate in answer generation.

Query-Conditioned Relevance Prediction. We model the relation between the QAT and each visual token through multi-head query–visual relevance scores at decoder layer K. For attention head h, the

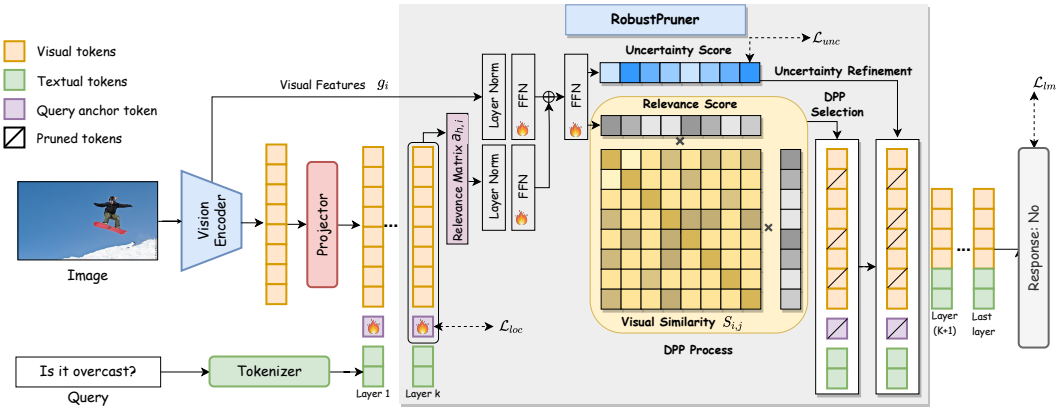

Figure 3: **Overview of RobustPruner.** At pruning layer K, a *Query Anchor Token* summarizes the query and interacts with visual tokens to produce query-conditioned relevance features. A lightweight prediction head outputs token-level relevance and uncertainty scores. RobustPruner then applies DPP-based candidate selection to preserve relevance and conditional diversity, followed by uncertainty-guided refinement to satisfy the final token budget.

relevance logit of visual token i is:

$$a_{h,i} = \frac{(W_q^h q_K)^\top (W_k^h v_{K,i})}{\sqrt{d_h}}, \qquad h = 1, \ldots, H, \tag{3}$$

where $d_h = d/H$ and $W_q^h, W_k^h \in \mathbb{R}^{d_h \times d}$ are learnable projections. We then apply LayerNorm (LN) over the head dimension for each visual token i:

$$\hat{a}_{h,i} = \mathrm{LN}(a_{h,i}) = \gamma_h \frac{a_{h,i} - \frac{1}{H}\sum_{k=1}^{H} a_{k,i}}{\sqrt{\frac{1}{H}\sum_{k=1}^{H}\left(a_{k,i} - \frac{1}{H}\sum_{m=1}^{H} a_{m,i}\right)^2 + \epsilon}} + \beta_h, \qquad h = 1, \ldots, H, \tag{4}$$

where γ_h and β_h are the learned scale and shift, and ϵ is a small positive constant added for stability.

Let $g_i \in \mathbb{R}^{d_g}$ denote the projected multi-layer visual summary feature of token i, computed from selected visual encoder layers, and let $\hat{a}_i = [\hat{a}_{1,i}, \ldots, \hat{a}_{H,i}] \in \mathbb{R}^H$ denote the normalized head-wise relevance vector. We first project the relevance vector and visual features into a shared fusion space with size d_f, and then construct the token evidence vector as:

$$h_i^{\mathrm{rel}} = \mathrm{SiLU}\left(W_a \hat{a}_i + b_a\right); \quad h_i^{\mathrm{vis}} = \mathrm{SiLU}\left(W_v\ \mathrm{LN}(g_i) + b_v\right); \quad x_i = \left[h_i^{\mathrm{rel}};\ h_i^{\mathrm{vis}};\ h_i^{\mathrm{rel}} \odot h_i^{\mathrm{vis}}\right] \tag{5}$$

where $W_a \in \mathbb{R}^{d_f \times H}$, $W_v \in \mathbb{R}^{d_f \times d_g}$, $h_i^{\mathrm{rel}}, h_i^{\mathrm{vis}} \in \mathbb{R}^{d_f}$, $x_i \in \mathbb{R}^{3d_f}$. The $\odot$ denotes element-wise multiplication between h_i^{rel} and h_i^{vis}, and x_i denotes the concatenated input among hidden states. A lightweight token-wise MLP maps x_i to an relevance logit before converted into a strictly positive DPP quality score, which adds the sigmoid function to serve as the token quality term in the DPP kernel,

$$\rho_i = \mathrm{MLP}_\rho(x_i), \qquad r_i = \frac{1}{1 + \exp(-\rho_i)}, \qquad r_i \in (0, 1), \tag{6}$$

where ρ_i is the relevance logit, and r_i is the output DPP quality score. The architectural details of MLP_ρ are deferred to Appendix C.1.

DPP Candidate Selection. To suppress redundant yet individually relevant tokens, we define the similarity matrix from the pairwise cosine similarity of the visual features at layer K:

$$S_{ij} = \frac{v_{K,i}^\top v_{K,j}}{\|v_{K,i}\|_2 \, \|v_{K,j}\|_2}, \tag{7}$$

where $S \in \mathbb{R}^{N \times N}$ is the visual-feature similarity matrix. Let $\mathbf{r} \in \mathbb{R}^{N \times 1}$ denote the vector of query-conditioned quality scores r_i. We combine $\mathbf{r}$ with S to construct the DPP kernel:

$$L = \mathrm{diag}(\mathbf{r})\, S\, \mathrm{diag}(\mathbf{r}). \tag{8}$$

Given the target budget m, RobustPruner first forms an over-complete candidate set of size $k = \lceil 1.1m \rceil$ and selects the DPP candidate set via the k-DPP MAP objective:

$$Y_{\mathrm{DPP}} = \underset{Y \subseteq \{1,\dots,N\},\, |Y|=k}{\arg\max} \log\det(L_Y), \tag{9}$$

where L_Y is the principal submatrix of L indexed by Y. In practice, we adopt greedy maximum a posteriori (MAP) inference [10], to efficiently construct a candidate pool that balances query-conditioned quality and set-wise diversity. We also provide a Projected Determinantal Leverage-score Selection alternative as an option to reduce the relatively high time and memory costs of constructing the full similarity matrix S and performing sequential greedy MAP inference.

Uncertainty-Guided Refinement. The DPP stage intentionally retains more tokens than the final budget, leaving room for reliability-oriented refinement. Similar to the relevance logit ρ_i, we predict a token uncertainty score under an MLP:

$$u_i = \mathrm{Softplus}(\mathrm{MLP}_u(x_i)), \qquad u_i > 0, \tag{10}$$

Since Y_{DPP} already enforces the relevance and diversity criteria, the refinement stage uses uncertainty only as a reliability signal within the candidate pool and retains the m least uncertain tokens:

$$Y^\star = \underset{Y \subseteq Y_{\mathrm{DPP}},\, |Y|=m}{\arg\min} \sum_{i \in Y} u_i. \tag{11}$$

Equivalently, $Y^\star$ contains the m tokens in Y_{DPP} with the smallest uncertainty scores. This design cleanly decouples relevance, diversity, and uncertainty. Relevance determines token quality in the DPP kernel, diversity is enforced by the determinant objective, and uncertainty is used only as the final reliability criterion. The retained visual tokens $\widetilde{V}_K = \{v_{K,i} \mid i \in Y^\star\}$ are passed to the remaining decoder layers together with the textual states, while all pruned visual tokens and their corresponding KV cache entries are removed.

3.2 Training Objective

We freeze the backbone MLLM and optimize only the QAT with its transformation matrix W, the projection layers, and the lightweight prediction heads. Training uses grounded GQA supervision with answer-relevant bounding boxes [20]. Each bounding box is rasterized onto the visual-token grid to produce binary token labels $b_i \in \{0, 1\}$, where $b_i = 1$ indicates that token i overlaps an answer-relevant region. Let $\mathbf{b}$ denote the label vector of b_i. The relevance head is trained with the localization objective:

$$\mathcal{L}_{\mathrm{loc}} = \mathcal{L}_{\mathrm{Dice}}(\mathbf{r}, \mathbf{b}) + 0.1\mathcal{L}_{\mathrm{BCE}}(\mathbf{r}, \mathbf{b}), \tag{12}$$

where $\mathbf{r}$ is the vector of predicted relevance probabilities. In addition, the language-modeling loss $\mathcal{L}_{\mathrm{lm}}$ encourages the QAT and pruning heads to preserve answer-relevant evidence for downstream generation.

For uncertainty supervision, we adopt the token-level uncertainty measure introduced in Section 2. For each training query Q selected in the GQA training set, we construct the first four semantics-preserving perturbations $\mathcal{P}(Q) = \{Q^{(1)}, \dots, Q^{(4)}\}$ using the Qwen3-VL-Plus prompt templates described in Appendix B. Let $\phi_{K,i}^{(p)}$ denote the similarity-based importance score of visual token i at layer K under perturbation $Q^{(p)}$. We compute the target uncertainty $\bar{u}_i$ by evaluating the variance in Eqn. (2) on $\{\phi_{K,i}^{(p)}\}_{p=1}^{4}$ and then normalizing the resulting scores to $[0, 1]$ within each sample. Consequently, large $\bar{u}_i$ values correspond to tokens whose estimated relevance is unstable across semantically equivalent prompt formulations. The uncertainty head is trained with an L_1 loss:

$$\mathcal{L}_{\mathrm{unc}} = \frac{1}{N} \sum_{i=1}^{N} |u_i - \bar{u}_i|, \tag{13}$$

and the full training loss is defined as:

$$\mathcal{L} = \lambda_{\mathrm{lm}}\mathcal{L}_{\mathrm{lm}} + \lambda_{\mathrm{loc}}\mathcal{L}_{\mathrm{loc}} + \lambda_{\mathrm{unc}}\mathcal{L}_{\mathrm{unc}}, \tag{14}$$

where λ_{lm}, λ_{loc} and λ_{unc} control the relationship among language-modeling, localization, and uncertainty supervision terms. The prompt perturbations are required only for constructing uncertainty targets during training. At inference time, RobustPruner receives a single input query without generating additional paraphrases, predicts r_i and u_i in a single forward pass, and then performs DPP candidate selection followed by uncertainty-guided refinement.

Table 1: **Performance comparison of different pruning methods on Qwen2.5-VL-7B. Acc.** is the average score across benchmarks, excluding MME. **Rel.** is the mean per-benchmark retention ratio relative to the full model across benchmarks, including MME. We divided popular token pruning methods into attention-based methods, similarity-based methods, and attention & similarity-based methods. The tested benchmarks are divided into low-resolution-tolerant (GQA, MME, ScienceQA, RealworldQA) and the remaining high-resolution-sensitive ones.

Method	GQA	MME	SciQA	RWQA	AI2D	ChartQA	DocVQA	TextVQA	MMMU	OCR	V*Bench	Acc.	Rel.
Qwen2.5-VL-7B	60.46	2323.62	87.46	69.80	93.75	83.08	94.38	82.88	50.67	83.8	70.68	77.7	100.0%
Retain 10% Tokens (↓90%)													
FastV (ECCV24)	47.32	1941.67	81.80	59.21	75.53	44.56	58.71	**70.93**	46.00	43.6	49.21	57.7	75.9%
CDPruner (NeurIPS25)	51.69	1950.13	76.95	53.33	67.07	42.84	30.79	42.55	45.56	36.5	51.83	49.9	68.0%
DART (EMNLP25)	51.72	2011.58	81.80	57.25	66.97	33.48	33.21	53.40	46.11	41.5	54.97	52.0	70.7%
DivPrune (CVPR25)	55.34	2032.39	83.29	60.52	72.05	40.80	54.32	62.96	45.44	47.3	65.97	58.8	78.2%
PruMerge+ (ICCV25)	52.64	1952.94	81.26	62.74	71.05	41.12	41.86	56.61	44.67	41.0	62.83	55.6	74.3%
VisionZip (CVPR25)	54.11	1933.56	83.29	63.40	72.38	43.88	48.44	55.84	45.11	39.5	63.87	57.0	75.8%
VScan (TMLR26)	54.17	2126.76	81.56	60.00	74.16	41.64	54.68	59.02	48.22	38.1	58.64	57.0	76.6%
RobustPruner (Ours)	**57.11**	**2213.69**	**84.33**	**66.27**	**85.43**	**59.00**	**78.41**	69.81	**49.67**	**59.6**	**67.02**	**67.7**	**88.6%**
Retain 20% Tokens (↓80%)													
FastV (ECCV24)	51.44	2131.93	82.70	64.31	82.42	60.56	75.86	**77.92**	49.67	59.7	56.54	66.1	86.2%
CDPruner (NeurIPS25)	55.68	2071.28	80.32	58.56	72.50	53.12	42.51	53.78	47.45	49.6	60.73	57.4	77.0%
DART (EMNLP25)	56.11	2166.96	83.49	63.53	70.44	48.60	50.36	67.48	46.00	54.0	59.69	60.0	80.1%
DivPrune (CVPR25)	58.38	2208.03	84.38	64.05	81.12	58.12	72.82	73.81	48.11	63.5	68.06	67.2	88.1%
PruMerge+ (ICCV25)	56.55	2215.23	**87.46**	64.97	83.06	52.44	60.72	68.65	45.89	56.2	67.54	64.3	84.9%
VisionZip (CVPR25)	57.00	2169.39	83.84	66.66	87.99	61.64	75.42	71.85	47.56	59.3	67.01	67.8	88.5%
VScan (TMLR26)	57.12	2258.84	84.53	65.49	80.38	59.60	77.61	73.91	**50.33**	51.3	61.78	66.2	87.2%
RobustPruner (Ours)	**58.54**	**2291.17**	85.72	**67.97**	**89.51**	**65.72**	**88.01**	76.91	49.89	**71.1**	**70.16**	**72.4**	**94.0%**
Retain 30% Tokens (↓70%)													
FastV (ECCV24)	54.15	2267.16	83.19	66.14	86.95	68.04	84.10	**80.33**	49.00	68.5	62.30	70.3	91.3%
CDPruner (NeurIPS25)	57.47	2161.50	81.41	60.13	76.17	60.64	53.37	60.70	48.45	57.2	67.02	62.3	82.7%
DART (EMNLP25)	57.85	2207.58	83.84	66.41	75.52	57.08	62.98	74.34	47.22	63.4	58.64	64.7	85.4%
DivPrune (CVPR25)	59.14	2245.39	84.88	65.62	87.21	67.08	81.79	77.66	47.89	72.1	69.11	71.2	92.5%
PruMerge+ (ICCV25)	58.21	2276.19	85.57	65.62	83.06	62.60	73.99	73.40	46.67	64.3	66.49	68.0	89.1%
VisionZip (CVPR25)	58.61	2278.95	86.37	66.14	87.99	**72.16**	86.26	77.00	48.89	70.6	65.97	72.0	93.4%
VScan (TMLR26)	58.39	2301.62	86.12	66.93	85.43	71.08	88.22	79.32	49.78	63.0	69.11	71.7	93.4%
RobustPruner (Ours)	**59.51**	**2336.41**	**86.52**	**68.24**	**91.71**	69.52	**90.74**	79.87	**49.89**	**77.7**	**71.20**	**74.5**	**96.5%**

4 Experiments

4.1 Experimental Settings

Model Architectures. We evaluate RobustPruner on the three open-source MLLMs: Qwen2.5-VL-7B [21], LLaVA-OneVision-1.5-8B [22], and LLaVA-v1.5-7B [23]. These backbones span two strong MLLM families and provide complementary evaluation regimes, allowing us to assess whether the proposed pruning module transfers across different visual encoders, token budgets, and generation backbones. Additional architectural details are summarized in Appendix C.2.

Training Setups. We use the grounded GQA training set with answer-relevant bounding-box supervision to train the pruning module [20]. The backbone MLLM is kept frozen, and we optimize only the query anchor token, the projection layers, and the lightweight relevance and uncertainty heads. To construct uncertainty supervision, we apply the semantics-preserving perturbation pipeline described in Appendix B to 30% of the training queries, generating four equivalent rewrites for each selected query; the remaining training samples are used with the original instruction only. More details on the training data and supervision protocol are provided in Appendix C.3.

Evaluation Benchmarks. We evaluate on a diverse suite of image-language benchmarks, including GQA [24], MME [25], ScienceQA [26], RealWorldQA [27], AI2D [28], ChartQA [29], DocVQA [30], TextVQA [31], MMMU [32], OCRBench [33], POPE [34], and V*Bench [35]. We use the default evaluation protocol of each benchmark. Additional benchmark descriptions are provided in Appendix C.4.

Comparison Methods. We compare RobustPruner with recent visual token reduction baselines spanning attention-based, diversity-driven, and hybrid pruning strategies, including FastV [4], CDPruner [10], DART [36], DivPrune [37], PruMerge+ [38], VisionZip [39], and VScan [11]. Because not every baseline is available for every backbone and some settings lead to out-of-memory failures, each table reports the strongest applicable subset of comparison methods for the corresponding model and retain ratio. Further baseline details are listed in Appendix C.5.

Table 2: **Performance comparison of different pruning methods on LLaVA-OneVision-1.5-8B. Acc.** and **Rel.** are computed only on benchmarks with complete results across all methods. **Rel.** is the mean per-benchmark retention ratio relative to the full model across these benchmarks.

Method	GQA	MME	SciQA	RWQA	AI2D	MMMU	OCR	POPE	V*Bench	Acc.	Rel.
LLaVA-OneVision-1.5-8B	62.07	2255.17	98.51	67.97	94.01	56.33	82.7	88.45	76.96	83.0	100.0%
Retain 10% Tokens (↓90%)											
FastV (ECCV24)	48.87	1807.58	81.60	49.41	73.28	49.00	37.2	62.95	37.17	64.5	77.6%
CDPruner (NeurIPS25)	50.99	1645.79	81.46	53.73	72.67	49.78	18.0	73.28	72.25	68.6	82.6%
DivPrune (CVPR25)	56.75	1896.01	87.95	59.87	77.07	51.22	41.9	84.21	72.77	76.3	92.0%
VisionZip (CVPR25)	55.48	–	88.80	**61.17**	71.37	–	–	79.05	–	74.4	89.6%
VScan (TMLR26)	52.90	1919.46	86.27	–	73.48	–	24.8	75.77	–	71.6	86.2%
RobustPruner (Ours)	**59.94**	**2175.94**	**95.59**	58.82	**88.70**	**52.44**	**51.7**	**88.64**	**76.96**	**81.4**	**97.9%**

Table 3: **Performance comparison of different pruning methods on LLaVA-v1.5-7B. Acc.** is computed over the available non-MME benchmarks only. **Rel.** is the mean per-benchmark retention ratio relative to the full model across the available benchmarks.

Method	GQA	MME	SciQA	RWQA	AI2D	MMMU	POPE	V*Bench	Acc.	Rel.
LLaVA-v1.5-7B	61.92	1864.84	69.51	55.82	58.81	35.89	85.85	50.26	59.7	100.0%
Retain 10% Tokens (↓90%)										
FastV (ECCV24)	46.88	1272.40	**69.06**	44.71	52.85	34.11	40.15	37.17	46.4	78.6%
CDPruner (NeurIPS25)	58.64	1712.25	68.37	50.85	55.60	35.56	**87.37**	47.12	57.6	95.6%
DivPrune (CVPR25)	57.41	1662.78	68.62	49.54	54.18	35.00	85.45	46.60	56.7	93.9%
VisionZip (CVPR25)	55.61	1672.70	68.77	49.80	53.59	35.00	76.98	36.65	53.8	89.9%
VScan (TMLR26)	56.99	1713.64	68.96	50.85	54.63	**36.22**	81.99	41.88	55.9	93.4%
RobustPruner (Ours)	**59.99**	**1862.69**	67.23	**53.33**	**58.36**	35.67	85.83	**49.74**	**58.6**	**98.3%**

4.2 Main Results

Results on Qwen2.5-VL-7B. Table 1 shows that RobustPruner consistently achieves the best accuracy–efficiency trade-off on Qwen2.5-VL-7B across all token budgets. At 10%, 20%, and 30% token retention, it attains the highest average accuracy and the strongest performance retention, outperforming strong baselines such as DivPrune, VisionZip, and VScan. The gains are largest under the most aggressive compression and remain clear as the token budget increases, with especially strong improvements on fine-grained and text-rich benchmarks such as AI2D, ChartQA, DocVQA, OCRBench, and V*Bench. These results indicate that uncertainty-guided refinement provides information that is complementary to relevance and diversity, rather than redundant with them.

Results on LLaVA-OneVision-1.5-8B. Table 2 shows that RobustPruner remains strongest on the high-resolution LLaVA-OneVision-1.5-8B even under a 90% token reduction. It achieves the best average score (81.4) and the highest performance retention (97.9%), and ranks first on eight of the nine benchmarks with complete results. The gains are particularly notable on high-resolution-sensitive tasks, while the exact preservation of the full-model V*Bench score further suggests that RobustPruner retains the fine-grained evidence required by dense visual inputs. These results underscore both the effectiveness and the practicality of the method on advanced high-resolution MLLMs. Additional results under 20% and 30% token retention ratios are reported in Table 8 of Appendix D.

Results on LLaVA-v1.5-7B. As shown in Table 3, on the widely used LLaVA-v1.5-7B, RobustPruner again yields the best trade-off under 10% token retention, achieving the highest average accuracy (58.6) and the strongest performance retention (98.3%). Its improvements are concentrated on GQA, RealWorldQA, AI2D, and V*Bench, indicating better preservation of spatially localized and fine-grained evidence, while MME remains nearly unchanged from the full model. This result shows that the proposed pruning module transfers cleanly from stronger high-resolution backbones to a widely adopted baseline without losing stability under severe compression. Additional results under 20% and 30% token retention ratios are reported in Table 9 of Appendix D.

Robustness Validation. Fig. 1(b) displays robustness to four semantics-preserving prompt perturbations utilized in uncertainty construction. For each image–question pair, we evaluate the original question and all paraphrases that preserve its meaning and visual grounding. The first four paraphrase styles are applied to the GQA training split to construct token-level uncertainty targets, whereas

Table 4: **Perturbation5 performance and mean flip rate.** P5 denotes the score on the held-out Perturbation5. Flip denotes the mean flip rate over perturbations 1–5 relative to the original questions.

	AI2D		ChartQA		GQA		MME		OCRBench	
Method	P5	Flip	P5	Flip	P5	Flip	P5	Flip	P5	Flip
Qwen2.5-VL-7B	93.36	2.01%	64.48	14.29%	59.19	15.30%	2120.76	8.42%	84.20	3.94%
FastV (ECCV24)	82.41	3.85%	49.44	13.37%	50.19	16.57%	1901.81	9.65%	59.20	**5.12%**
CDPruner (NeurIPS25)	72.89	5.49%	45.12	14.37%	53.10	16.42%	1872.82	10.12%	48.70	6.83%
DivPrune (CVPR25)	81.41	3.94%	45.96	14.63%	56.16	16.64%	1988.10	9.49%	62.80	6.68%
VisionZip (CVPR25)	82.19	3.65%	46.32	14.86%	55.05	16.45%	1999.88	9.16%	59.50	5.98%
VScan (TMLR26)	79.48	4.25%	48.24	14.33%	54.48	16.24%	1900.35	9.22%	48.46	6.78%
RobustPruner (Ours)	**89.99**	**2.67%**	**52.01**	**14.30%**	**56.86**	**16.05%**	**2201.16**	**8.48%**	**70.40**	5.15%

Table 5: **Efficiency analysis of RobustPruner** on the AI2D test set with only 10% visual tokens retained. Compared with the original LLaVA-OneVision-1.5-8B and Qwen2.5-VL-7B models, RobustPruner substantially reduces the number of visual tokens, prefill FLOPs, KV cache size, and GPU memory usage, while maintaining competitive task performance.

Method	Visual Tokens	Prefill FLOPs	KV Cache	GPU Memory	Avg.Time per Question	Score
LLaVA-OneVision-1.5-8B	523.3	8.53T	83.1MB	19.0GB	265.09ms	94.01
+RobustPruner (Ours)	**52.0 (10.06×)**	**4.41T (1.93×)**	**17.0MB (4.89×)**	**16.5GB (1.15×)**	**218.84ms (1.21×)**	**88.7**
Qwen2.5-VL-7B	510.2	7.72T	31.6MB	17.5GB	154.87ms	93.75
+RobustPruner (Ours)	**50.6 (10.08×)**	**3.93T (2.0×)**	**6.5MB (4.86×)**	**16.0GB (1.09×)**	**145.67ms (1.08×)**	**85.43**

we add a fifth style which is held out from this process to test out-of-distribution robustness to an unseen rewrite style. For evaluation, we average performance over the five paraphrases within AI2D, ChartQA, GQA, MME, and OCRBench test sets and then aggregate the benchmark-level results; performance on the original questions is reported separately as a reference. Under this protocol, RobustPruner preserves a larger fraction of the full-model performance than prior pruning baselines, suggesting that its gains extend beyond the rewrite styles used for uncertainty training. We further validate prediction stability using answer consistency, measured by the mean flip rate (percentage of originally correct answers that become incorrect in perturbations). Across five perturbations, RobustPruner achieves the lowest mean flip rate among all pruning methods except for OCRBench. Table 4 indicates the performance of RobustPruner on out-of-distribution (Perturbation5) and the mean flip rate. Additional results for all five perturbations are provided in Table 10 of Appendix D.

Efficiency Analysis. Table 5 reports the efficiency gains on AI2D at 10% token retention. RobustPruner reduces the visual tokens by about 10×, which leads to about 2× lower prefill FLOPs and nearly 5× smaller KV cache on both LLaVA-OneVision-1.5-8B and Qwen2.5-VL-7B. GPU memory and average time per question are reduced more moderately, since a large part of the memory comes from the frozen backbone parameters and the pruning layer K is inside the decoding stage. These results show that RobustPruner brings concrete inference savings while maintaining competitive AI2D performance. RobustPruner is not intended to maximize raw speed in isolation, but to provide a substantially stronger accuracy–efficiency trade-off under aggressive pruning.

4.3 Visualization Analysis

Fig. 1(a) highlights the main qualitative advantage of RobustPruner under semantics-preserving prompt perturbations. Prior pruning methods can shift the retained evidence across equivalent rephrasings, making the prediction sensitive to surface-form variation and occasionally removing visually decisive regions. In contrast, RobustPruner preserves more consistent answer-critical evidence across prompt variants, leading to more stable predictions. This behavior is consistent with the robustness–efficiency trends in Fig. 1(b)–(c): by retaining evidence that is both query-relevant and prompt-stable, the method improves robustness without sacrificing the benefits of aggressive compression. Additional examples are provided in Appendix E.

4.4 Ablation and Sensitivity Experiments

Ablation Experiments. We conduct ablation studies on RobustPruner. As shown in Fig. 4, we evaluate two variants of the proposed method on Qwen2.5-VL-7B. The first variant (*i.e.* , RobustPruner w/o DPP) replaces the DPP-inspired token selection in RobustPruner with a simpler Top-K rule that

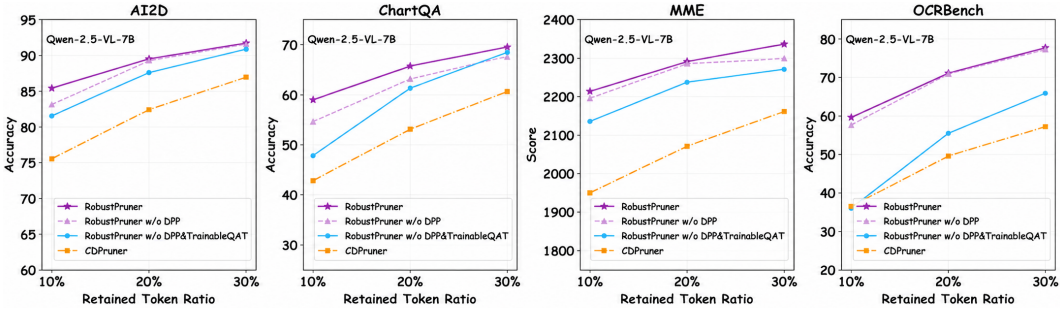

Figure 4: **Ablation study of RobustPruner on Qwen2.5-VL-7B.** We compare RobustPruner with two simplified variants. The consistent performance drops validate the effectiveness of our diversity-aware and learnable token selection design.

Table 6: **Sensitivity analysis.** Left: Accuracy (%)/latency (ms) across pruning layers. Right: Accuracy (%) across candidate-pool ratios (k/m) at 20% retention. * denotes the default setting.

Dataset	Ret.	$K=8$	$K=12^*$	$K=14$	$K=18$
AI2D	10%	76.55/129.79	84.90/140.48	90.70/148.34	93.39/157.95
	20%	82.67/132.31	89.64/141.99	92.45/148.88	93.49/157.59
	30%	87.01/135.43	91.68/144.13	93.16/149.82	93.43/159.70
OCRBench	10%	29.8/745.26	59.2/680.93	72.0/682.46	71.9/698.87
	20%	49.7/729.84	71.4/672.39	78.8/671.59	79.2/703.22
	30%	62.0/688.76	77.5/672.06	80.3/672.96	80.7/692.46

k/m	1.00	1.05	1.10^*	1.15	1.20
AI2D	89.28	89.86	89.54	89.44	88.96
OCRBench	70.9	71.4	71.4	71.0	70.3
k/m	1.25	1.30	1.40	1.50	2.00
AI2D	89.38	89.42	88.89	88.67	87.47
OCRBench	70.1	68.9	69.0	67.1	61.9

directly keeps the tokens with the highest scores . The second variant (*i.e.* , RobustPruner w/o DPP & TrainableQAT) further degrades this design by keeping the QAT component fixed. As a result, the language loss only trains the transformation matrix that controls how the query attends to visual tokens. These two variants reduce the algorithmic complexity of RobustPruner, but they also degrade performance. Nevertheless, both ablated variants still outperform the training-free CDPruner baseline overall, indicating that RobustPruner benefits substantially from learning an intelligent visual-token selector rather than relying on a fixed training-free pruning rule. The localization loss supervision is beneficial for text-intensive recognition while the pruning framework remains effective without it, as removing Grounded-GQA localization supervision reduces AI2D accuracy only slightly (89.51 to 88.99) but lowers OCRBench accuracy from 71.1 to 64.1 at 20% token retention.

Sensitivity Experiments. According to the choice of decoder layer K and candidate set size k, we prune at decoder layer $K = 12$ and use a candidate pool of approximately 1.1 times the final budget m. We evaluate $K \in \{8, 12, 14, 18\}$ at 10%, 20%, and 30% token retention, pruning at $K = 8$ can reduce accuracy because visual tokens have not accumulated sufficient query-conditioned information; $K = 12$ best balances quality and early token removal, whereas later pruning limits acceleration in spite of the accuracy improvement in larger token budgets. The candidate size is fixed in all experiments rather than tuned separately for each benchmark. Performance remains stable over the moderate range from $1.05m$ to $1.15m$. Setting $k = m$ leaves no capacity for uncertainty refinement, whereas excessively large candidate pools admit less relevant tokens and weaken final selection. Matched-layer and candidate-size sensitivity experiments are provided in Table 6. These settings define our chosen operating point rather than an optimum for every benchmark.

5 Conclusion

We presented RobustPruner, a plug-and-play visual token pruning framework that decouples query-conditioned relevance, DPP-based diversity, and uncertainty-guided refinement to preserve task-critical visual evidence with low prompt-induced uncertainty under aggressive compression. Across three open-source MLLMs and diverse vision-language benchmarks, RobustPruner consistently achieves a stronger accuracy–efficiency–robustness trade-off than prior pruning baselines, with stable behavior under semantics-preserving prompt perturbations and clear gains in prefill FLOPs, KV cache, GPU memory, and inference speed per question. Overall, the results show that uncertainty-aware pruning is a practical and effective basis for reliable, efficient, and robust multimodal inference.

Acknowledgements

This research is partially supported by the Ministry of Education, Singapore, under the funding of MOE-T2EP20123-0006, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL).

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 23716–23736, 2022.

[2] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742, 2023.

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, pages 34892–34916, 2023.

[4] 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*, pages 19–35, 2024.

[5] Zicong Tang, Ziyang Ma, Suqing Wang, Zuchao Li, Lefei Zhang, Hai Zhao, Yun Li, and Qianren Wang. CoViPAL: Layer-wise contextualized visual token pruning for large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20701–20714, 2025.

[6] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, pages 13937–13949, 2021.

[7] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, pages 1–10, 2023.

[8] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, pages 74840–74857, 2025.

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

[10] 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.

[11] Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. VScan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research*, 2025.

[12] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, 2016.

[13] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[14] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, 2020.

[15] Alina Leidinger, Robert van Rooij, and Ekaterina Shutova. The language of prompting: What linguistic properties make a prompt successful? In *Findings of the Association for Computational Linguistics: EMNLP*, pages 9210–9232, 2023.

[16] Mohamed Insaf Ismithdeen, Muhammad Uzair Khattak, and Salman Khan. Promptception: How sensitive are large multimodal models to prompts? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23950–23985, 2025.

[17] Kexin Ma, Jing Xiao, Chaofeng Chen, Geyong Min, Guibo Zhu, Jinqiao Wang, and Liang Liao. Decoupled similarity for task-aware token pruning in large vision-language models. *arXiv preprint arXiv:2604.11240*, 2026.

[18] Rinyoichi Takezoe, Yaqian Li, Zihao Bo, Anzhou Hou, Mo Guang, and Kaiwen Long. Learnpruner: Rethinking attention-based token pruning in vision language models. In *International Conference on Learning Representations*, 2026.

[19] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[20] Quan-Sheng Zeng, Yunheng Li, Qilong Wang, Peng-Tao Jiang, Zuxuan Wu, Ming-Ming Cheng, and Qibin Hou. A glimpse to compress: Dynamic visual token pruning for large vision-language models. *arXiv preprint arXiv:2508.01548*, 2025.

[21] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[22] 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*, 2025.

[23] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[24] Drew A Hudson and Christopher D Manning. GQA: a new dataset for real-world visual reasoning and compositional question answering. In *IEEEF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019.

[25] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models, 2023.

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

[27] xAI. Grok-1.5 vision preview. xAI Blog, 2024. Introduces the RealWorldQA benchmark. https://x.ai/blog/grok-1.5v.

[28] 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*, pages 235–251, 2016.

[29] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

[30] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. DocVQA: A dataset for vqa on document images. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2200–2209, 2021.

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

[32] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[33] 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, 2024.

[34] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 292–305, 2023.

[35] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024.

[36] Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Conference on Empirical Methods in Natural Language Processing*, pages 9972–9991, 2025.

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

[38] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *IEEE International Conference on Computer Vision*, pages 22857–22867, 2025.

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

[40] Jiaying Zhu, Yurui Zhu, Xin Lu, Wenrui Yan, Dong Li, Kunlin Liu, Xueyang Fu, and Zheng-Jun Zha. VisionSelector: End-to-end learnable visual token compression for efficient multimodal llms. *arXiv preprint arXiv:2510.16598*, 2025.

[41] Zijun Chen, Wenbo Hu, Guande He, Zhijie Deng, Zheng Zhang, and Richang Hong. Unveiling uncertainty: A deep dive into calibration and performance of multimodal large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3095–3109, 2025.

[42] Tuo Wang, Adithya Kulkarni, Tyler Cody, Peter A. Beling, Yujun Yan, and Dawei Zhou. GENUINE: Graph enhanced multi-level uncertainty estimation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 20522–20541, 2025.

[43] Rui Li, Jing Long, Muge Qi, Heming Xia, Lei Sha, Peiyi Wang, and Zhifang Sui. Towards harmonized uncertainty estimation for large language models. In *Annual Meeting of the Association for Computational Linguistics*, pages 22938–22953, 2025.

[44] Roman Vashurin, Maiya Goloburda, Preslav Nakov, and Maxim Panov. UNCERTAINTY-LINE: Length-invariant estimation of uncertainty for large language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 7882–7919, 2025.

[45] Debarun Bhattacharjya, Balaji Ganesan, Junkyu Lee, Radu Marinescu, Katsiaryna Mirylenka, Michael Glass, and Xiao Shou. SIMBA UQ: Similarity-based aggregation for uncertainty quantification in large language models. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 15880–15894, 2025.

[46] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

[47] Alex Kulesza and Ben Taskar. k-DPPs: Fixed-size determinantal point processes. In *International Conference on Machine Learning*, pages 1193–1200, 2011.

[48] Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy MAP inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*, pages 5627 – 5638, 2018.

[49] Zelda E. Mariet, Yaniv Ovadia, and Jasper Snoek. DppNet: Approximating determinantal point processes with deep networks. In *Advances in Neural Information Processing Systems*, pages 3218 – 3229, 2019.

Appendix

A Related Work **15**

B Construction of Semantics-Preserving Textual Perturbations **16**

C Additional Experimental Setups **19**

C.1 Details on the Lightweight MLP 19

C.2 Details on MLLMs 19

C.3 Details on the Model Training 20

C.4 Details on the Evaluation Benchmarks 20

C.5 Details on the Comparison Methods 21

D Additional Quantitative Comparisons **22**

E Additional Qualitative Comparisons **24**

F Limitations **26**

G Broader Impacts **26**

A Related Work

Visual Token Reduction in MLLMs. Building on token reduction in vision transformers, such as DynamicViT [6] and ToMe [7], visual pruning in MLLMs has evolved from heuristic filtering toward task-aware selection. Representative methods include FastV [4], which prunes with decoder attention, SparseVLM [8], which uses text-aware sparsification, and CoViPAL [5], which introduces lightweight layer-wise contextual pruning. Recent advances sharpen the selection signal. VScan redesigns the pipeline with encoder-side global–local scanning and intermediate-layer pruning [11]. VisionSelector learns an end-to-end scorer for adaptive compression [40]. CDPruner enforces instruction-conditioned diversity through a DPP objective [10]. LearnPruner revisits attention and identifies middle-layer text-to-vision attention as a stronger pruning cue [18]. DeSAP combines decoupled cross-modal similarity with visual saliency [17]. Compared with these methods, **RobustPruner** explicitly models predictive uncertainty, complementing relevance and diversity with a reliability signal for retaining ambiguous but task-critical tokens.

Uncertainty Estimation for Reliable Prediction. Classical uncertainty estimation in deep learning is grounded in Bayesian approximations such as MC dropout [12] and in decomposing predictive uncertainty into epistemic and aleatoric components [13]. For MLLMs, Chen *et al*. show that current systems remain miscalibrated under multimodal reasoning [41], whereas GENUINE introduces graph-enhanced multi-level estimation beyond token-level confidence [42]. For LLMs, recent post-hoc methods refine uncertainty from complementary perspectives: CUE harmonizes heterogeneous uncertainty signals [43], UNCERTAINTY-LINE corrects length bias [44], and SIMBA UQ aggregates semantic similarity for more robust estimates [45]. Together, these results indicate that reliable uncertainty estimation requires structured signals beyond raw confidence. Our method brings this perspective to visual token pruning, using predictive uncertainty to preserve visually ambiguous yet task-critical tokens.

Determinantal Point Processes. Determinantal point processes (DPPs) are a canonical model for diverse subset selection, balancing item quality and set-wise diversity through determinant-based repulsion [46]. Classical extensions include k-DPPs, which impose fixed-cardinality control [47], and fast greedy MAP inference, which makes DPP-based selection more practical at scale [48]. More recent work studies scalable approximations such as DppNet [49]. In MLLMs, CDPruner adapts this line by using instruction-conditioned similarity to maximize conditional diversity [10]. Our method likewise uses a DPP to build a diverse candidate pool, but decouples diversity from uncertainty by applying uncertainty-aware refinement after DPP selection, allowing relevance, diversity, and reliability to be modeled separately.

Projected Determinantal Leverage-score Selection. Projected Determinantal Leverage-score Selection is a projected DPP-marginal alternative to reduce the relatively high time and memory costs of constructing the full similarity matrix S and performing sequential greedy MAP inference [46]. Let $x_i \in \mathbb{R}^{d_v}$ be the normalized visual feature supplied to the selector, and let $R \in \mathbb{R}^{d_v \times s}$ be a fixed Gaussian projection, where $s = \min\{k, N, d_v\}$. Using the same query-conditioned quality score r_i, we define

$$z_i = r_i \frac{R^\top x_i}{\|R^\top x_i\|_2}, \qquad Z = [z_1^\top; \ldots; z_N^\top]. \tag{15}$$

Score each token by its projected DPP-style marginal and select the k highest-scoring candidates:

$$\ell_i = z_i^\top (I_s + Z^\top Z)^{-1} z_i, \qquad Y_{\text{marg}} = \text{TopK}_k\{\ell_i\}_{i=1}^N. \tag{16}$$

This alternative avoids materializing the $N \times N$ similarity matrix and replaces the k sequential greedy selection steps with low-dimensional matrix operations. In RobustPruner, the resulting scores retain query-conditioned quality and account for redundancy across the projected tokens and have verifiable performances.

B Construction of Semantics-Preserving Textual Perturbations

For each original visual query Q, we use Qwen3-VL-Plus to generate semantics-preserving rewrites, with one rewrite produced from each of the prompt templates below. Each template is designed to preserve the original question meaning, answer equivalence, and visual grounding. In particular, the rewrites are required to retain all objects, attributes, quantities, spatial relations, comparisons, negations, temporal constraints, and other visually grounded conditions in the original query. We denote the resulting set of rewrites by

$$\mathcal{P}(Q) = \{Q^{(1)}, Q^{(2)}, Q^{(3)}, Q^{(4)}, Q^{(5)}\},$$

where $Q^{(k)}$ corresponds to the output generated by the k-th prompt template. The first four perturbation sets are used in the empirical analysis in Section 2 and in the uncertainty-supervision procedure described in Section 3, while all five sets are used in the robustness evaluation.

1. **Prompt 1 (lexical paraphrase).** Rewrite the following visual question as a single natural paraphrase. Preserve its meaning exactly. Do not change the answer, objects, attributes, counts, spatial relations, negations, or temporal constraints. Output only the rewritten question. Query: `<QUERY>`.
2. **Prompt 2 (syntactic reformulation).** Rephrase the following visual question using a different sentence structure while keeping the same visual grounding and answer. Do not add, remove, or weaken any condition. Output only the rewritten question. Query: `<QUERY>`.
3. **Prompt 3 (semantic-equivalence rewrite).** Produce one semantically equivalent rewrite of the following visual question. Preserve all entities, attributes, quantities, comparisons, and relations exactly. Keep the wording natural and concise. Output only the rewritten question. Query: `<QUERY>`.
4. **Prompt 4 (academic-style rewrite).** Rewrite the following image question so that it is phrased differently while asking exactly the same thing. Use academic, sophisticated vocabulary while maintain answer equivalence and the same relevant visual evidence. Output only the rewritten question. Query: `<QUERY>`.
5. **Prompt 5 (imperative sentence rewrite).** Rewrite the following image question so that it is phrased differently while asking exactly the same thing. Use instruction-style imperative sentences while maintain sentence expressions equivalent. Output only the rewritten question. Query: `<QUERY>`.

Prompt 1: Lexical Paraphrase

System Message:
You are a visual question rewriting assistant. Rewrite the given visual question as a single natural paraphrase. Preserve the meaning exactly. Do not change the answer, objects,

attributes, counts, spatial relations, negations, or temporal constraints. Output only the rewritten question.

User Message:
Query:

1. What cooking utensil is round?
2. What is the person to the right of the ball holding?
3. Which kind of vehicle is the skyscraper behind?
4. What animal is in front of the bench that is made of wood?

Example Rewrites:

1. Which kitchen utensil is circular?
2. What is the individual to the right of the ball holding?
3. What type of vehicle is the skyscraper behind?
4. Which animal is in front of the wooden bench?

Prompt 2: Syntactic Reformulation

System Message:
You are a visual question rewriting assistant. Rephrase the given visual question using a different sentence structure while preserving the same visual grounding and answer. Do not add, remove, or weaken any condition stated in the original query. Output only the rewritten question.

User Message:
Query:

1. What cooking utensil is round?
2. What is the person to the right of the ball holding?
3. Which kind of vehicle is the skyscraper behind?
4. What animal is in front of the bench that is made of wood?

Example Rewrites:

1. Which cooking utensil has a round shape?
2. What is being held by the person positioned to the right of the ball?
3. Behind which kind of vehicle is the skyscraper located?
4. In front of the wooden bench, what animal is present?

Prompt 3: Semantic-Equivalence Rewrite

System Message:
You are a visual question rewriting assistant. Produce one semantically equivalent rewrite of the given visual question. Preserve all entities, attributes, quantities, comparisons, relations, negations, and visually grounded constraints exactly. Keep the wording natural and concise. Output only the rewritten question.

User Message:
Query:

1. What cooking utensil is round?
2. What is the person to the right of the ball holding?

3. Which kind of vehicle is the skyscraper behind?
4. What animal is in front of the bench that is made of wood?

Example Rewrites:

1. Which round cooking utensil is shown?
2. What object is held by the person to the right of the ball?
3. What kind of vehicle is in front of the skyscraper?
4. Which animal is located in front of the wooden bench?

Prompt 4: Academic-Style Rewrite

System Message:
You are a visual question rewriting assistant. Rewrite the given image-based question so that it is phrased on precise, sophisticated vocabulary and more elaborate syntax while asking exactly the same thing. Maintain answer equivalence and rely on the same relevant visual evidence. Do not change the intended meaning, required reasoning, or any visually grounded condition. Output only the rewritten question.

User Message:
Query:

1. What cooking utensil is round?
2. What is the person to the right of the ball holding?
3. Which kind of vehicle is the skyscraper behind?
4. What animal is in front of the bench that is made of wood?

Example Rewrites:

1. Which cooking utensil exhibits a round configuration?
2. What is being held by the individual situated to the right of the ball?
3. Which category of vehicle is the skyscraper situated behind?
4. Which animal occupies a position in front of the bench constructed from wood?

Prompt 5: Imperative Sentence Rewrite

System Message:
You are a visual question rewriting assistant. Rewrite the given image-based question as instruction-style imperative sentences so that it is phrased differently while stating exactly the same thing. Maintain answer equivalence and rely on the same relevant visual evidence. Do not change the intended meaning, required reasoning, or any visually grounded condition. Output only the rewritten question.

User Message:
Query:

1. What cooking utensil is round?
2. What is the person to the right of the ball holding?
3. Which kind of vehicle is the skyscraper behind?
4. What animal is in front of the bench that is made of wood?

Example Rewrites:

1. Identify the cooking utensil that is round.

2. State what the person to the right of the ball is holding.
3. Identify the type of vehicle that the skyscraper is behind.
4. Name the animal in front of the wooden bench.

C Additional Experimental Setups

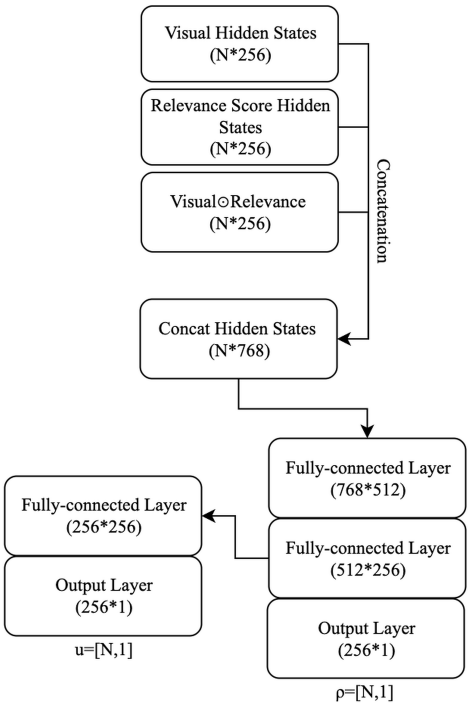

Figure 5: Architecture of the relevance and uncertainty prediction networks.

Table 7: Training hyperparameters.

Hyperparameter	**Value**
Number of GPUs	4
Optimizer	AdamW
Batch size	1
Gradient accumulation steps	8
Effective batch size	32
Number of training epochs	1
Learning rate	1×10^{-4}
Warmup ratio	0.1
Training data size	20,000
Perturbed question size	6,000
Data type	bfloat16
Attention implementation	flash_attention_2
λ_{lm}	1.0
λ_{loc}	1.0
λ_{unc}	0.5

C.1 Details on the Lightweight MLP

As shown in Fig. 5, the scorer MLP_ρ in Eq. (6) is applied independently to each token evidence vector x_i and shares parameters across all visual tokens. Its input is the normalized concatenation $x_i \in \mathbb{R}^{3d_f}$ of the current-layer relevance feature h_i^{rel}, the projected multi-layer summary h_i^{vis}, and their element-wise product. The module outputs the relevance logit ρ_i, which is subsequently reused by the DPP quality score in Eq. (6), and the embedding is linked to the uncertainty head in Eq. (10) simultaneously. The DPP stage then combines the resulting quality score r_i with the pairwise cosine similarity matrix in Eq. (7) to build the kernel in Eq. (8). We intentionally keep MLP_ρ lightweight so that query-conditioned scoring adds negligible overhead relative to the frozen MLLM backbone.

C.2 Details on MLLMs

We evaluate RobustPruner on three representative open-source MLLMs.

1. **Qwen2.5-VL-7B** [21] is used as our primary strong open-source baseline. It is a recent vision-language model from the Qwen-VL series, equipped with a native dynamic-resolution visual encoder and a Qwen2.5 language backbone. The model supports diverse visual inputs, including natural images, documents, charts, and videos, and shows strong capabilities in visual grounding, OCR, chart understanding, and multimodal reasoning. We adopt Qwen2.5-VL-7B to evaluate our method on a competitive and widely used modern MLLM architecture.

2. **LLaVA-OneVision-1.5-8B** [22] is a recent high-capacity LLaVA-style multimodal model designed for strong cross-scenario visual understanding. It is developed under the LLaVA-OneVision framework and trained with large-scale curated multimodal data, covering image, document, chart, and general visual reasoning tasks. Compared with earlier LLaVA variants, it provides stronger overall performance and better generalization across different evaluation settings. We use this model to test whether our method transfers to a newer and more capable LLaVA-family backbone.
3. **LLaVA-v1.5-7B** [23] is a widely adopted open-source multimodal model that connects a CLIP ViT-L/14 visual encoder with a Vicuna-based language model through a lightweight multimodal projector. Due to its simple architecture and broad use in prior visual token pruning studies, LLaVA-v1.5-7B serves as a standard backbone for comparison with existing methods. We include it to ensure fair evaluation against previous pruning baselines and to analyze the behavior of our method on a classical LLaVA architecture.

C.3 Details on the Model Training

Training uses the grounded GQA data with answer-relevant bounding boxes. Each box is rasterized onto the visual-token grid to supervise the relevance head, while the language-modeling objective preserves answer-critical evidence in the pruned model. For uncertainty supervision, we perturb only 30% of the training queries using the first four semantics-preserving prompt templates in Appendix B; for each selected query, these rewrites are used to compute the variance-based uncertainty target defined in Section 3. The remaining 70% of training samples use the original query only, which keeps the training pipeline efficient while still providing substantial perturbation-based supervision. The training hyperparameters are listed in Table 7. The total training process can be completed in around 20 minutes conducted on four NVIDIA H20 GPUs.

C.4 Details on the Evaluation Benchmarks

Our benchmark suite is designed to cover complementary multimodal capabilities.

- **GQA** [24] is a compositional question answering dataset designed for real-world visual reasoning. The images are derived from Visual Genome, where each image is paired with a scene graph describing objects, attributes, and relations. Based on these structured annotations, GQA provides about 22M automatically generated yet semantically grounded questions, covering skills such as object recognition, attribute identification, spatial reasoning, relation understanding, and multi-hop inference. The dataset further balances answer distributions to reduce reliance on language priors. We use it to evaluate the fine-grained visual grounding and compositional reasoning ability of multimodal models.
- **ScienceQA** [26] is a large-scale multimodal multiple-choice benchmark for science reasoning. It contains 21,208 questions from elementary and high-school science curricula, spanning natural science, language science, and social science, with 26 topics, 127 categories, and 379 skills. The questions may include image context, text context, or both, with 48.7% containing images and 30.8% containing both modalities. Most samples are also paired with lectures and explanations that provide supporting knowledge and reasoning chains. We evaluate on the image-context test split, commonly known as ScienceQA-IMG.
- **MME** [25] is a comprehensive evaluation benchmark for multimodal large language models, covering both perceptual and cognitive abilities. It consists of 14 subtasks with a unified yes/no question format, enabling a more controlled comparison across models. The perception-related tasks evaluate coarse-grained recognition, fine-grained recognition, and OCR, including object existence, counting, position, color, celebrity, landmark, scene, artwork, and text recognition. The cognition-related tasks further examine commonsense reasoning, numerical calculation, text translation, and code reasoning. We use MME to evaluate whether multimodal models can reliably ground their answers in visual evidence while handling diverse recognition and reasoning scenarios.
- **RealWorldQA** [27] is a visual question answering benchmark introduced by xAI to evaluate real-world spatial and physical understanding in multimodal models. It contains over 700 image-question pairs collected from practical everyday scenarios, including images captured from vehicles and other real-world environments. The questions focus on abilities such as

object counting, size comparison, relative position reasoning, scene understanding, and judging physical affordances in the environment. Each question is paired with a verifiable answer, making the benchmark suitable for measuring whether models can ground their responses in concrete visual evidence rather than relying on generic priors. We use RealWorldQA to assess the model's capability in practical spatial reasoning and real-world visual understanding.

- **AI2D** [28] is a diagram-based visual question answering benchmark built from elementary-school science diagrams. It contains over 5K diagrams and more than 15K multiple-choice questions, covering diverse diagram types such as life cycles, food webs, flow charts, and system illustrations. The benchmark mainly evaluates a model's ability to understand diagram structures, recognize visual components, and reason over relations between textual labels and graphical elements.
- **ChartQA** [29] is a chart-based visual question answering benchmark for evaluating chart understanding and numerical reasoning. It contains 9.6K human-written questions and 23.1K machine-generated questions over real-world charts collected from online sources. The questions require models to interpret visual elements, read values, compare quantities, and perform arithmetic reasoning based on chart data.
- **DocVQA** [30] is a document-based visual question answering benchmark built from scanned document images. It contains 50K questions over more than 12K document pages from the Industry Documents Library, covering diverse layouts such as forms, tables, figures, and text-heavy pages. The benchmark evaluates a model's ability to read document content, understand layout structure, and answer questions grounded in visual and textual evidence.
- **TextVQA** [31] is a text-oriented visual question answering benchmark built from images containing scene text. It includes 45,336 questions over 28,408 images from Open Images, where answering often requires recognizing and reasoning over textual content in the image. The benchmark evaluates a model's ability to combine OCR-based text understanding with visual grounding and question reasoning.
- **OCRBench** [33] is a comprehensive OCR-oriented benchmark for evaluating text perception and reasoning in multimodal models. It contains 1,000 manually curated question-answer pairs covering five OCR-related tasks, including text recognition, scene-text VQA, document-oriented VQA, key information extraction, and handwritten mathematical expression recognition. The benchmark measures whether models can accurately read, localize, and reason over textual information in diverse visual scenarios.
- **MMMU** [32] is a massive multi-discipline multimodal understanding benchmark designed for expert-level reasoning. It contains 11.5K questions collected from college exams, quizzes, and textbooks, covering six core disciplines and 30 subject areas. The questions require models to interpret diverse visual inputs such as charts, diagrams, tables, maps, and scientific figures, while performing subject-specific reasoning across domains including science, engineering, medicine, business, and humanities.
- **POPE** [34] is a polling-based benchmark for evaluating object hallucination in multimodal models. It formulates hallucination detection as a series of binary questions about object existence, covering three settings with different negative-sample construction strategies: random, popular, and adversarial. The benchmark is commonly used to measure whether a model can faithfully judge the presence of objects in an image rather than generating answers biased by language priors.
- **V*Bench** [35] is a visual question answering benchmark designed to evaluate fine-grained visual search in high-resolution and visually crowded images. Unlike standard VQA benchmarks that often focus on salient objects, V*Bench requires models to locate small or subtle visual details and answer questions based on the identified evidence. It mainly tests whether multimodal models can perform targeted visual grounding, detail recognition, and spatial understanding under challenging image conditions.

C.5 Details on the Comparison Methods

The compared baselines cover several complementary pruning philosophies.

- **FastV** [4] is a visual token pruning method for accelerating multimodal large language models. It estimates token importance from attention patterns in the language model and removes less

informative image tokens in deeper layers. We include FastV as a representative attention-based pruning baseline.

- **CDPruner** [10] formulates visual token selection as a determinantal point process to preserve a conditionally diverse token subset. It jointly considers visual feature similarity and instruction relevance, allowing the retained tokens to better match the current user query. We include CDPruner as a representative diversity-aware pruning baseline.
- **DART** [36] reduces visual redundancy by pruning image tokens according to feature similarity before they are passed into the language model. Unlike attention-based methods, it does not rely on internal LLM attention scores for token importance estimation. We include DART as a representative similarity-based pruning baseline.
- **DivPrune** [37] preserves a compact yet diverse subset of visual tokens by formulating token selection as a diversity maximization problem. It focuses on reducing redundancy among image tokens based on their visual feature similarity. We include DivPrune as a representative diversity-based pruning baseline.
- **PruMerge+** [38] is a visual token reduction method that combines token pruning and token merging for efficient MLLM inference. It first identifies informative visual tokens according to their relevance to the text instruction, and then merges redundant tokens to further compress the visual sequence while preserving contextual information. We include PruMerge+ as a representative pruning-and-merging baseline.
- **VisionZip** [39] is a visual token compression method that reduces image tokens from both spatial and contextual perspectives. It keeps a small set of dominant tokens to preserve global visual information and further selects informative contextual tokens to retain fine-grained details. We include VisionZip as a representative token selection and compression baseline for efficient MLLM inference.
- **VScan** [11] is a two-stage visual token reduction method for efficient LVLM inference. It first applies complementary global and local scans during visual encoding to retain both semantically important and spatially diverse tokens, followed by token merging to preserve visual information. It then performs middle-layer pruning in the language model to remove query-irrelevant visual tokens while maintaining cross-modal interactions. We include VScan as a representative progressive token reduction baseline.

D Additional Quantitative Comparisons

Full results on LLaVA-OneVision-1.5-8B. Table 8 reports the complete results on LLaVA-OneVision-1.5-8B under 10%, 20%, and 30% retained token ratios, complementing the main paper where only the most challenging 10% setting is presented due to space limits. Across all retention ratios, RobustPruner consistently achieves the best average accuracy and highest performance retention among all pruning methods. At 10% retention, it obtains an average score of 81.4 and preserves 97.9% of the full-model performance, substantially outperforming training-free baselines and recent token-pruning methods. As the retained token ratio increases, RobustPruner further reaches 81.8 average score and 98.5% relative performance at 20% retention, and maintains 81.8 average score and 98.4% relative performance at 30% retention, suggesting that the selected tokens already preserve most task-relevant visual information under aggressive pruning. The gains are especially clear on high-resolution and fine-grained benchmarks such as AI2D, OCRBench, POPE, and V*Bench; notably, RobustPruner exactly preserves the full-model V*Bench score at 10% retention and remains highly competitive at higher retention ratios. These results indicate that the proposed query-aware and diversity-aware token selection strategy effectively retains visual evidence important for dense visual understanding, rather than merely keeping tokens with high individual saliency.

Full results on LLaVA-v1.5-7B. Table 9 presents the complete results on LLaVA-v1.5-7B under 10%, 20%, and 30% retained token ratios. RobustPruner consistently achieves the best average accuracy and relative performance across all three pruning settings. Even with only 10% visual tokens retained, it preserves 98.3% of the full-model performance, outperforming existing training-free and recent token-pruning baselines. As the retained token ratio increases, RobustPruner further improves the relative performance to 98.7% at 20% retention and 99.4% at 30% retention. It also performs strongly on benchmarks such as GQA, MME, RWQA, AI2D, and V*Bench, indicating that

the learned query-aware token selector effectively preserves task-relevant visual evidence. These results show that RobustPruner generalizes well to the classic LLaVA-v1.5 backbone and provides a consistently better efficiency–performance trade-off across different token budgets.

Table 8: **Performance comparison of different pruning methods on LLaVA-OneVision-1.5-8B. Acc.** and **Rel.** are computed only on benchmarks with complete results across all methods. **Rel.** is the mean per-benchmark retention ratio relative to the full model across these benchmarks.

Method	GQA	MME	SciQA	RWQA	AI2D	MMMU	OCR	POPE	V*Bench	Acc.	Rel.
LLaVA-OneVision-1.5-8B	62.07	2255.17	98.51	67.97	94.01	56.33	82.7	88.45	76.96	83.0	100.0%
Retain 10% Tokens (↓90%)											
FastV (ECCV24)	48.87	1807.58	81.60	49.41	73.28	49.00	37.2	62.95	37.17	64.5	77.6%
CDPruner (NeurIPS25)	50.99	1645.79	81.46	53.73	72.67	49.78	18.0	73.28	72.25	68.6	82.6%
DivPrune (CVPR25)	56.75	1896.01	87.95	59.87	77.07	51.22	41.9	84.21	72.77	76.3	92.0%
VisionZip (CVPR25)	55.48	–	88.80	**61.17**	71.37	–	–	79.05	–	74.4	89.6%
VScan (TMLR26)	52.90	1919.46	86.27	–	73.48	–	24.8	75.77	–	71.6	86.2%
RobustPruner (Ours)	**59.94**	**2175.94**	**95.59**	58.82	**88.70**	**52.44**	**51.7**	**88.64**	**76.96**	**81.4**	**97.9%**
Retain 20% Tokens (↓80%)											
FastV (ECCV24)	51.60	1918.60	88.10	52.94	78.79	51.44	49.2	67.12	43.98	68.9	82.8%
CDPruner (NeurIPS25)	56.06	1820.58	87.90	58.82	79.21	51.22	30.1	82.99	**79.05**	75.6	91.1%
DivPrune (CVPR25)	59.44	2063.25	92.31	63.00	84.52	53.00	**57.0**	87.22	75.96	79.7	96.0%
VisionZip (CVPR25)	58.37	–	91.82	**66.14**	–	–	–	84.83	–	78.3	94.4%
VScan (TMLR26)	57.43	2096.05	92.56	–	79.34	–	43.7	84.12	–	78.0	93.9%
RobustPruner (Ours)	**60.80**	**2272.84**	**96.08**	63.01	**89.54**	**53.67**	56.6	**88.38**	76.44	**81.8**	**98.5%**
Retain 30% Tokens (↓70%)											
FastV (ECCV24)	53.05	2017.06	91.12	54.90	83.52	53.67	**59.3**	70.55	48.17	71.6	85.9%
CDPruner (NeurIPS25)	58.81	1937.24	90.23	59.74	83.23	51.89	40.5	86.14	**78.01**	78.4	94.6%
DivPrune (CVPR25)	60.29	2110.49	94.45	64.71	89.09	**54.44**	–	88.10	77.49	80.9	97.5%
VisionZip (CVPR25)	60.12	–	96.22	**67.06**	–	–	–	86.85	–	81.1	97.6%
VScan (TMLR26)	59.53	2197.49	95.58	–	84.68	–	57.9	86.64	–	80.6	97.0%
RobustPruner (Ours)	**60.46**	**2268.15**	**96.43**	65.49	**89.64**	52.33	56.8	**88.41**	77.49	**81.8**	**98.4%**

Table 9: **Performance comparison of different pruning methods on LLaVA-v1.5-7B. Acc.** is computed over the available non-MME benchmarks only. **Rel.** is the mean per-benchmark retention ratio relative to the full model across the available benchmarks.

Method	GQA	MME	SciQA	RWQA	AI2D	MMMU	POPE	V*Bench	Acc.	Rel.
LLaVA-v1.5-7B	61.92	1864.84	69.51	55.82	58.81	35.89	85.85	50.26	59.7	100.0%
Retain 10% Tokens (↓90%)										
FastV (ECCV24)	46.88	1272.40	**69.06**	44.71	52.85	34.11	40.15	37.17	46.4	78.6%
CDPruner (NeurIPS25)	58.64	1712.25	68.37	50.85	55.60	35.56	**87.37**	47.12	57.6	95.6%
DivPrune (CVPR25)	57.41	1662.78	68.62	49.54	54.18	35.00	85.45	46.60	56.7	93.9%
VisionZip (CVPR25)	55.61	1672.70	68.77	49.80	53.59	35.00	76.98	36.65	53.8	89.9%
VScan (TMLR26)	56.99	1713.64	68.96	50.85	54.63	**36.22**	81.99	41.88	55.9	93.4%
RobustPruner (Ours)	**59.99**	**1862.69**	67.23	**53.33**	**58.36**	35.67	85.83	**49.74**	**58.6**	**98.3%**
Retain 20% Tokens (↓80%)										
FastV (ECCV24)	49.71	1466.40	68.42	47.71	53.82	35.00	53.14	37.43	49.3	83.5%
CDPruner (NeurIPS25)	59.66	1744.93	68.42	53.07	55.83	35.89	**87.12**	47.12	58.2	96.7%
DivPrune (CVPR25)	59.36	1716.80	68.47	49.67	56.19	35.78	86.65	48.17	57.8	95.9%
VisionZip (CVPR25)	57.61	1742.65	68.82	52.29	55.41	**36.22**	82.04	45.03	56.8	94.9%
VScan (TMLR26)	59.45	1767.78	**68.96**	53.07	55.96	**36.22**	85.66	45.02	57.8	96.3%
RobustPruner (Ours)	**60.43**	**1854.23**	68.77	**54.12**	**58.71**	36.00	85.85	**48.69**	**58.9**	**98.7%**
Retain 30% Tokens (↓70%)										
FastV (ECCV24)	52.72	1571.13	68.47	47.71	54.73	35.22	63.19	38.17	51.5	86.8%
CDPruner (NeurIPS25)	60.30	1762.18	68.96	52.68	55.83	35.67	**86.82**	49.21	58.5	97.4%
DivPrune (CVPR25)	59.85	1744.81	**69.26**	51.50	56.09	36.44	86.62	45.55	57.9	96.3%
VisionZip (CVPR25)	58.90	1778.06	69.06	51.37	56.12	**36.89**	85.29	43.98	57.4	95.9%
VScan (TMLR26)	60.30	1778.79	68.96	52.68	57.19	36.56	86.40	46.07	58.3	97.2%
RobustPruner (Ours)	**61.07**	**1860.76**	68.96	**55.69**	**58.55**	36.11	85.87	**49.22**	**59.4**	**99.4%**

Robustness under perturbations. Table 10 reports the robustness evaluation on Qwen2.5-VL-7B when retaining 20% visual tokens. For each benchmark, we evaluate the original test set and five perturbed variants, and report the average score as well as the relative retention ratio with respect to the full model. This setting tests whether a pruning method can preserve task-relevant visual evidence when the input distribution is affected by common perturbations.

RobustPruner consistently achieves the best average performance and relative retention across all five benchmarks. On AI2D, ChartQA, MME, and OCRBench, RobustPruner clearly outperforms all competing pruning methods under both the original and perturbed settings. On GQA, RobustPruner obtains the highest average score and relative retention, while remaining competitive on each individual perturbation. These results show that RobustPruner is less sensitive to input perturbations than prior training-free and recent token-pruning baselines.

The robustness gain is especially clear on fine-grained and high-resolution-sensitive tasks such as AI2D, ChartQA, and OCRBench, where preserving the correct visual evidence is crucial. For example, RobustPruner retains 95.5%, 79.4%, 98.6%, and 83.7% of the full-model performance on AI2D, ChartQA, MME, and OCRBench, respectively, outperforming the strongest competing baselines by clear margins. This confirms that the proposed query-aware and diversity-aware token selection strategy improves not only clean accuracy but also robustness under distribution shifts.

Table 10: **Robustness under perturbations on Qwen2.5-VL-7B across five benchmarks (Retain 20% tokens).** Each benchmark reports the original score, five perturbed variants, the mean of the original and five perturbed scores (**Avg.**), and the relative retention ratio (**Rel.**) with respect to the full model.

Method	Original	Perturbation1	Perturbation2	Perturbation3	Perturbation4	Perturbation5	Avg.	Rel.
AI2D								
Qwen2.5-VL-7B	93.75	92.68	93.10	92.20	93.01	93.36	93.02	100.0%
FastV (ECCV24)	83.06	81.83	82.80	81.64	82.35	82.41	82.35	88.5%
CDPruner (NeurIPS25)	72.51	70.95	72.11	71.28	71.50	72.89	71.87	77.3%
DivPrune (CVPR25)	81.86	80.86	81.63	80.67	80.76	81.41	81.20	87.3%
VisionZip (CVPR25)	87.99	81.02	81.76	81.41	81.70	82.19	82.68	88.9%
VScan (TMLR26)	80.38	79.14	80.21	78.40	79.27	79.48	79.48	85.4%
RobustPruner (Ours)	**89.51**	**88.63**	**89.35**	**88.08**	**88.47**	**89.99**	**89.00**	**95.7%**
ChartQA								
Qwen2.5-VL-7B	83.08	77.12	79.68	76.44	68.36	64.48	74.86	100.0%
FastV (ECCV24)	62.52	59.60	61.40	59.80	52.64	49.44	57.57	76.9%
CDPruner (NeurIPS25)	53.12	50.60	51.44	50.80	44.12	45.12	49.20	65.7%
DivPrune (CVPR25)	59.28	56.16	57.40	56.08	49.84	45.96	54.12	72.3%
VisionZip (CVPR25)	61.64	56.88	57.96	57.04	51.04	46.32	55.15	73.7%
VScan (TMLR26)	59.60	56.52	57.76	56.92	50.52	48.24	54.93	73.4%
RobustPruner (Ours)	**65.72**	**61.80**	**63.60**	**61.40**	**52.88**	**52.01**	**59.57**	**79.6%**
GQA								
Qwen2.5-VL-7B	60.46	54.59	53.95	54.91	51.73	59.19	55.80	100.0%
FastV (ECCV24)	51.46	46.54	46.35	46.94	44.40	50.19	47.65	85.4%
CDPruner (NeurIPS25)	55.68	50.13	49.48	50.10	46.49	53.10	50.83	91.1%
DivPrune (CVPR25)	58.46	52.33	**52.13**	53.12	**49.67**	56.16	53.64	96.1%
VisionZip (CVPR25)	56.78	51.05	50.77	51.56	48.35	55.05	52.26	93.6%
VScan (TMLR26)	57.12	51.34	50.94	51.60	48.34	54.48	52.30	93.7%
RobustPruner (Ours)	**58.54**	**52.72**	52.11	**53.14**	49.44	**56.86**	**53.80**	**96.4%**
MME								
Qwen2.5-VL-7B	2323.62	2249.76	2327.96	1879.38	2303.97	2120.76	2200.91	100.0%
FastV (ECCV24)	2155.85	2086.32	2128.24	1722.57	2117.19	1901.81	2018.66	91.7%
CDPruner (NeurIPS25)	2071.28	1988.46	2066.65	1686.62	1993.84	1872.82	1946.61	88.4%
DivPrune (CVPR25)	2194.63	2131.34	2156.13	1747.64	2133.30	1988.10	2058.52	93.5%
VisionZip (CVPR25)	2200.18	2124.87	2178.78	1732.12	2153.43	1999.88	2064.88	93.8%
VScan (TMLR26)	2258.84	2175.52	2206.16	1773.13	2161.16	1900.35	2079.19	94.5%
RobustPruner (Ours)	**2276.22**	**2235.81**	**2263.97**	**1905.92**	**2250.25**	**2201.16**	**2188.89**	**99.5%**
OCRBench								
Qwen2.5-VL-7B	83.80	79.70	82.00	81.50	82.10	84.20	82.22	100.0%
FastV (ECCV24)	59.70	57.50	58.00	58.40	58.70	59.20	58.58	71.3%
CDPruner (NeurIPS25)	49.60	45.90	48.60	48.30	47.40	48.70	48.08	58.5%
DivPrune (CVPR25)	63.50	59.50	62.40	61.30	62.00	62.80	61.92	75.3%
VisionZip (CVPR25)	59.30	56.10	58.80	57.40	59.10	59.50	58.37	71.0%
VScan (TMLR26)	51.30	48.60	48.10	44.50	49.80	48.46	48.46	58.9%
RobustPruner (Ours)	**71.10**	**65.20**	**68.70**	**67.10**	**70.30**	**70.40**	**68.80**	**83.7%**

E Additional Qualitative Comparisons

Fig. 6 provides qualitative comparisons under semantics-preserving textual perturbations. Each example includes an original visual question and a perturbed question with the same meaning and answer, where red boxes indicate answer-critical regions and the overlaid patches show the retained visual tokens after pruning. The examples cover attribute recognition, fine-grained part localization, and spatial relation reasoning. Across these settings, RobustPruner preserves more stable and semantically aligned evidence than CDPruner and VScan. For instance, when the query

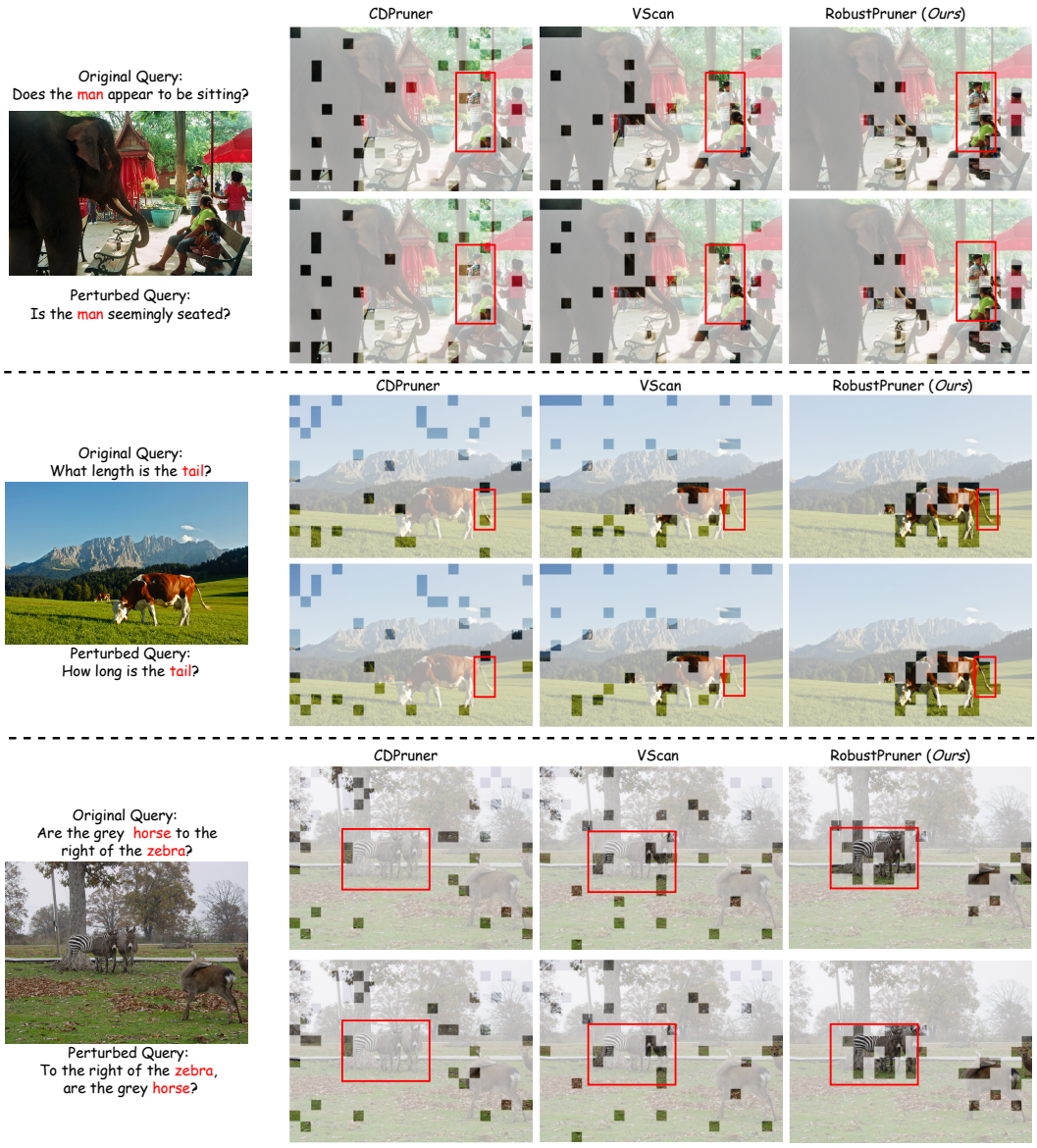

Figure 6: Qualitative comparison of visual-token pruning under semantics-preserving query perturbations. Each example shows an original query and a perturbed query that preserve the same meaning and answer. Red boxes mark answer-critical visual regions, while the overlaid retained patches indicate the evidence preserved after pruning. Compared with CDPruner and VScan, RobustPruner maintains more stable and semantically aligned evidence across equivalent query formulations, especially around the queried object or relation.

asks whether the man is sitting, the baselines exhibit noticeable shifts in retained evidence across paraphrases, whereas RobustPruner consistently retains tokens around the relevant person. Similarly, for the tail-length query, RobustPruner maintains evidence near the small queried region, while the baselines retain more dispersed or less consistently localized patches. In the spatial-relation example involving the grey horse and the zebra, RobustPruner again preserves tokens around the visually decisive relational region under both query formulations. These results indicate that effective pruning should retain not only visually informative tokens but also evidence that remains stable under meaning-preserving linguistic variation. By jointly modeling query relevance and uncertainty, RobustPruner better preserves answer-critical regions across paraphrases.

F Limitations

RobustPruner still has several limitations. First, our experiments are mainly conducted on representative open-source MLLMs and standard vision-language benchmarks. Although the results show consistent efficiency–performance gains, further evaluation on broader model families, proprietary systems, and more diverse real-world scenarios would better validate its generality.

Second, RobustPruner requires an additional training stage to learn the query-aware visual-token selector. This cost is incurred only once before deployment and does not affect inference-time efficiency, but it still increases the preparation cost compared with purely training-free pruning methods. Future work could explore more lightweight training strategies or stronger transferability across models and domains.

G Broader Impacts

RobustPruner aims to improve the inference efficiency of MLLMs by reducing redundant visual tokens. Its positive impacts include lower computational cost, reduced memory usage, and potentially lower energy consumption, which can make MLLMs more accessible for resource-constrained deployment scenarios.

At the same time, more efficient MLLMs may also lower the barrier to deploying vision-language systems at scale. If such systems are used in sensitive or harmful applications, efficiency improvements could amplify existing risks related to hallucination, biased predictions, surveillance, or misuse. Therefore, RobustPruner should be deployed together with appropriate evaluation, safety checks, and application-specific safeguards.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the scope and contributions of the paper, including the proposed robust visual-token pruning framework, its motivation, and the main empirical findings. The claims are supported by the main experimental results and ablation studies reported in the paper and appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of RobustPruner in the appendix, including its current evaluation scope, dependence on representative calibration/training data, and the potential overhead introduced during the pruning-module training stage.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper does not present formal theoretical results such as theorems or lemmas. The proposed method is evaluated empirically through benchmark comparisons, efficiency analysis, and ablation studies.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the key details needed to reproduce the main experimental results, including the evaluated MLLM backbones, benchmarks, pruning ratios, baselines, evaluation metrics, and implementation details in the paper and appendix. We also report ablation studies and efficiency measurements to support the main claims and conclusions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the source code of implementation of RobustPruner and instructions for reproducing the main results in the supplementary material. The code also includes scripts for constructing our perturbed dataset from existing public datasets, together with evaluation scripts for the reported benchmarks and pruning ratios.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify the experimental settings needed to understand the results, including the evaluated model backbones, datasets, pruning ratios, baselines, evaluation metrics, and training/evaluation protocols. Additional implementation details, including hyperparameters and data construction procedures for the perturbed dataset, are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: We do not report error bars or statistical significance tests due to the high computational cost of repeatedly training and evaluating large MLLMs. To support the robustness of the conclusions, we report results across multiple model backbones, benchmarks, pruning ratios, and ablation settings under fixed evaluation protocols.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the compute resources used for the experiments, including the GPU type, memory requirements, and approximate training and evaluation costs for reproducing the reported results. These details are included in the implementation details and the source code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms to the NeurIPS Code of Ethics. We use public datasets and open-source model backbones, preserve anonymity in the submission materials and code, and do not involve human-subject data collection or sensitive personal information.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of RobustPruner, including its potential positive impact in reducing the computational cost, memory usage, and energy consumption of MLLM inference. We also acknowledge that making MLLMs more efficient may lower the barrier to deploying such systems, which could amplify existing risks of MLLMs if used in inappropriate or harmful applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release high-risk pretrained models, image generators, or newly scraped datasets. The released source code focuses on the RobustPruner implementation, evaluation scripts, and scripts for constructing the perturbed dataset from existing public datasets, following their original usage terms.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit the creators of the existing model backbones, datasets, and baseline methods used in this paper through citations. We also follow the corresponding licenses and terms of use, and provide additional information on the assets and the construction of our perturbed dataset in the appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce a perturbed dataset constructed from existing public datasets and provide documentation alongside the source code. The source code describes the dataset construction procedure, usage instructions, and evaluation protocol, while respecting the licenses and terms of the original datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: This paper does not involve crowdsourcing experiments or research with human subjects. All experiments are conducted using existing public datasets and model backbones.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing experiments or research with human subjects. Therefore, IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core algorithm and method development in this paper does not involve LLMs as an important, original, or non-standard component. The MLLMs used in this work are the target model backbones for pruning and evaluation, and semantics-preserving textual perturbation datasets used for uncertainty measurement are constructed by Qwen3-VL-Plus. No additional LLM-based system is used for annotation or automated scientific decision-making.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.