SCOUT: VISUAL TOKEN PRUNING FOR MULTIMODAL LARGE LANGUAGE MODELS VIA SEMANTIC AND CROSS-MODAL GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) have achieved strong performance in vision–language understanding by integrating vision encoders with large language models. However, the quadratic complexity of attention with respect to token length makes inference expensive, with visual tokens dominating the input sequence. Recent studies have explored visual token pruning to address this issue, but most methods estimate token importance solely based on attention scores. In certain cases, such scores may not reliably reflect token importance for two reasons: attention in the vision encoder arises from unimodal self-attention, while attention concentration can assign high weights to regions with limited semantic relevance. To address these limitations, we propose **SCOUT** (**S**emantic and **C**ross-m**o**dal guided vis**u**al **T**oken pruning), a novel two-stage visual token pruning framework. In Stage 1, SCOUT combines encoder attention with value norms to mitigate the limitations of relying solely on attention scores and performs semantic region-aware coarse filtering. In Stage 2, it uses text-to-visual attention for fine-grained pruning within the language model, with entropy-guided gating to reduce the influence of attention sink-like behavior. Experiments on multiple MLLM architectures across image and video benchmarks show that SCOUT consistently improves the trade-off between efficiency and performance.

1 INTRODUCTION

Recent advances in Large Language Models (LLMs) have significantly accelerated progress in multimodal learning Chiang et al. (2023); Meta (2025); Yang et al. (2025a). By integrating powerful language models with vision encoders, Multimodal Large Language Models (MLLMs) have achieved remarkable progress in various vision-language tasks, including visual question answering (VQA) and image captioning Liu et al. (2023); Li et al. (2023b). A key factor behind these capabilities is the attention mechanism in Transformer architectures Vaswani et al. (2017). In MLLMs, attention determines how visual information is routed to the language model during decoding Liu et al. (2023); Wang et al. (2024). Visual tokens interact with textual tokens through attention layers, and the resulting attention patterns determine which visual regions contribute to decoding. Therefore, in terms of model performance, more accurately directing attention to question-relevant objects plays an important role in multimodal reasoning.

However, attention introduces a major efficiency bottleneck, as its computational cost grows quadratically with the token sequence length. In MLLMs, visual tokens often dominate the decoding sequence, especially for high-resolution images and long videos. Since images are represented as discrete patch embeddings Dosovitskiy et al. (2020), visual tokens inevitably contain redundancy, such as background regions or similar patches Ryoo et al. (2021); Liang et al. (2022). This redundancy amplifies the inference cost, making visual token reduction essential for efficient MLLMs.

Visual token pruning has recently emerged as an effective solution Chen et al. (2024); Zhang et al. (2025); Yang et al. (2025c); Tong et al. (2025); Ye et al. (2025); Yang et al. (2025b). The key is to more accurately identify visual tokens that are important for multimodal reasoning. Most existing methods estimate token importance using attention scores Chen et al. (2024); Zhang et al. (2025);

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

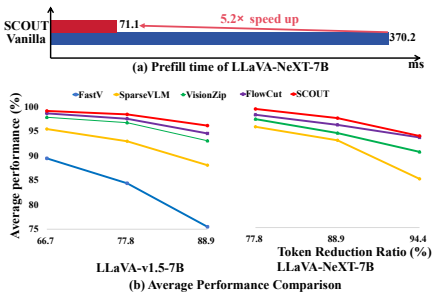

Figure 1: Efficiency and performance of SCOUT. (a) Prefill acceleration on LLaVA-NeXT-7B. (b) Comparison with previous token-pruning methods.

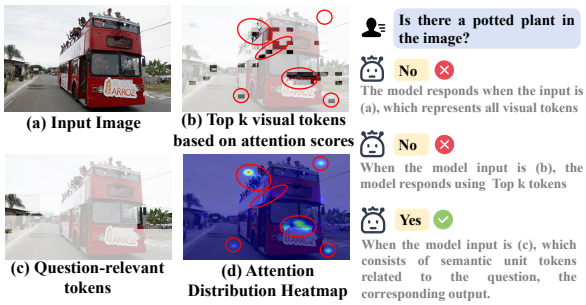

Figure 2: Visualization of attention-based token selection on POPE. (a) Input image. (b) Top-k tokens selected by attention scores ($k = 53$). (c) Question-relevant semantic tokens. (d) [CLS] attention heatmap.

Yang et al. (2025c); Tong et al. (2025), implicitly assuming that such scores reflect the importance of visual tokens for downstream language generation.

However, this assumption does not always hold. In vision encoders such as CLIP Radford et al. (2021), attention is computed through intra-modal self-attention among visual tokens. These scores mainly capture structural relationships within visual representations, rather than cross-modal interactions with textual tokens during language decoding. Therefore, vision-encoder attention may not accurately indicate token importance for multimodal reasoning. Moreover, attention-based pruning is susceptible to the attention sink phenomenon Sun et al. (2023); Gu et al. (2024); Qiu et al. (2025), where certain tokens, such as the "[CLS]" token or specific visual tokens, receive disproportionately attention weights despite limited semantic relevance. This behavior has been widely observed in Transformer-based models and arises from architectural properties such as softmax normalization Sun et al. (2024); Gu et al. (2024); Qiu et al. (2026), making precise pruning based on vanilla attention scores unreliable in certain scenarios.

Motivated by these observations, we revisit the role of attention in visual token pruning. To achieve more precise pruning, visual saliency and query-conditioned cross-modal interactions provide complementary cues for estimating token importance. Moreover, the bias caused by attention concentrating on semantically less relevant regions should also be mitigated.

Based on these insights, we propose a two-stage visual token pruning framework. In Stage 1, we perform coarse-grained filtering within the vision encoder. By combining attention scores with value norms, we estimate token importance beyond attention alone and preserve tokens from semantically informative regions. At this stage, to improve efficiency, we perform only conservative preliminary pruning to retain visual evidence for subsequent text-guided refinement. In Stage 2, we use text-to-visual attention within the LLM to identify tokens relevant to the current question. An entropy-guided gate adjusts their importance scores, reducing the influence of tokens that receive high aggregate attention but have diffuse associations across text tokens. The resulting scores guide fine-grained pruning.

Our contributions are summarized as follows: (i) We propose **SCOUT**, a novel two-stage visual token pruning framework for MLLMs. The first stage performs semantic region-aware filtering in the vision encoder, leveraging complementary attention and value information to preserve informative regions while reducing token redundancy. (ii) In the second stage, we introduce a cross-modal refinement strategy in the language model, where entropy-guided gating dynamically identifies query-relevant visual tokens and mitigates the effects of attention sink-like behaviors. (iii) Extensive experiments across various MLLM architectures and diverse image and video understanding benchmarks demonstrate that SCOUT consistently achieves a superior accuracy-efficiency trade-off and outperforms existing state-of-the-art token pruning methods, as shown in Figure 1.

2 Related Work

Visual Token Pruning in MLLMs. Visual tokens dominate the input sequence and contain substantial redundancy during MLLM inference. Visual token pruning accelerates inference by remov-

ing redundant visual tokens. Among training-free methods, FastV Chen et al. (2024) and SparseVLM Zhang et al. (2025) perform pruning in the LLM decoder. FastV removes visual tokens in early decoder layers based on aggregated attention scores, while SparseVLM uses text-to-vision attention and adaptively determines the pruning ratio from the rank of the attention matrix. In contrast, encoder-side methods prune tokens within the vision encoder. VisionZip Yang et al. (2025c) estimates token importance using attention scores from the penultimate encoder layer. More recently, FlowCut Tong et al. (2026) exploits cross-layer information flow and uses the CLS token as an information relay to guide progressive pruning. Beyond training-free approaches, training-based methods have also been explored Shao et al. (2025); Ye et al. (2025), with ATP-LLaVA predicting instance-specific pruning thresholds from global image attention distributions. Despite their differences, most existing methods rely primarily on attention scores to estimate token importance.

Attention-based Token Importance Estimation. Attention scores are widely used as proxies for token importance in visual token pruning Chen et al. (2024); Zhang et al. (2025); Yang et al. (2025c); Tong et al. (2025). Although empirically effective, these methods implicitly assume that attention scores faithfully reflect visual token contributions to multimodal reasoning. However, the validity of this assumption remains insufficiently understood, particularly due to the mismatch between vision encoder attention patterns and language-guided token relevance, as well as structural biases in Transformer attention Gu et al. (2024); Qin et al. (2024); Lin et al. (2025).

3 METHODOLOGY

Motivated by the shortcomings of existing attention-based pruning strategies, we investigate the reliability of attention-driven token selection and propose **SCOUT**, a two-stage framework for effective visual token pruning, as illustrated in Figure 3. We first revisit attention-based importance estimation and examine its limitations. These observations motivate SCOUT to perform more reliable token selection through semantic-aware filtering and language-guided refinement.

3.1 REVISITING ATTENTION SCORES

In this section, we revisit the role of attention in visual token pruning through empirical analyses and case studies, focusing on two key limitations: the unimodal nature of intra-vision attention and the structural bias introduced by the attention sink-like behavior. Specifically, attention scores from vision encoders are derived from unimodal self-attention and capture only intra-visual relationships, failing to reflect cross-modal interactions between visual tokens and textual queries during language decoding Radford et al. (2021); Yang et al. (2025c). Moreover, Transformer attention distributions often exhibit the attention sink-like behavior Xiao et al. (2023); Gu et al. (2024); Kang et al. (2025); Zuhri et al. (2025), where certain tokens receive excessive attention regardless of semantic relevance. These factors lead to biased estimation of token importance and suboptimal pruning.

3.1.1 INTRA-VISION SELF-ATTENTION ANALYSIS.

To investigate the reliability of attention-based pruning in MLLMs, we analyze the attention distributions of the visual encoder, with a representative example from the POPE dataset shown in Figure 2(a). Since the visual encoder relies solely on intra-modal self-attention, its attention scores capture visual relationships without considering textual queries. Therefore, pruning based only on vanilla attention may fail to identify question-relevant tokens and potentially remove critical visual evidence. Meanwhile, retaining all visual tokens is not always optimal, as redundant tokens introduce noise and increase attention competition during cross-modal interaction.

Given the textual question **"Is there a potted plant in the image?"**, the model without token pruning shown in Figure 2(a) incorrectly predicts "No". This observation suggests that excessive visual tokens may introduce redundancy and interfere with multimodal reasoning. We then select the top-k tokens according to vanilla attention scores. However, the model still produces an incorrect answer. As shown in Figure 2(b), these tokens contain few regions related to the textual concept ***potted plant***, demonstrating that unimodal attention may discard question-relevant evidence. For comparison, we construct a subset consisting of 53 question-relevant visual tokens and one CLS token, as shown in Figure 2(c). Under this setting, the model correctly answers "Yes". This comparison highlights the importance of text-guided token selection in preserving visual evidence relevant to reasoning.

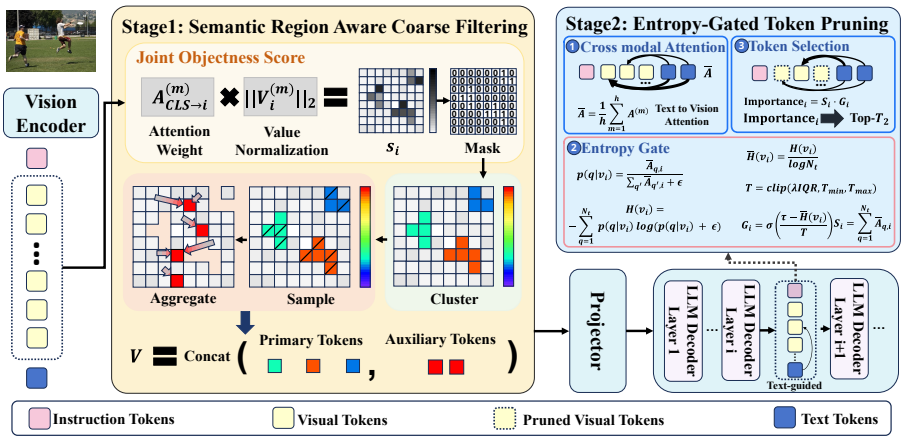

Figure 3: Overview of SCOUT, a two-stage visual token pruning framework. Stage 1 combines vision-encoder attention scores with value norms for coarse pruning while preserving semantically informative regions. Stage 2 uses text-guided, entropy-gated refinement to reduce the influence of attention sink-like patterns and select query-relevant visual tokens.

3.1.2 Attention Sink.

Attention sink is a common phenomenon in Transformer architectures that also affects MLLMs. We visualize the CLS attention map from the penultimate layer of the visual encoder in LLaVA-1.5-7B, as shown in Figure 2(d). Although several regions receive disproportionately high attention weights, these regions are not consistently aligned with meaningful objects or the textual query. Comparing Figure 2(b) and Figure 2(d), we observe that attention-based pruning tends to preserve tokens from sink regions. Therefore, effectively mitigating attention sink-like behavior is crucial for reliable visual token pruning Yu et al. (2024).

3.2 Proposed Approach: SCOUT

SCOUT is a token pruning framework that adopts a two-stage strategy, as shown in Figure 3. In the first stage, coarse-grained visual token filtering is performed with semantic regions as the central focus, aiming to preserve as much informative content as possible. In the second stage, inspired by gating mechanisms, we reweight the attention scores to mitigate the influence of attention sinks. By leveraging joint interactions between textual and visual information, the framework further prunes unimportant tokens to achieve more effective and efficient token selection.

3.2.1 Semantic Region Aware Coarse Filtering.

The first stage aims to reduce visual redundancy while preserving diverse semantic information as much as possible. Motivated by the coarse-to-fine processing behavior of human vision, we divide the token budget into two complementary groups: Primary Tokens that preserve salient semantic regions and Auxiliary Tokens that maintain global scene context.

Formally, the token budget is defined as $T_1 = T_p + T_a$, where T_p and T_a denote the numbers of primary and auxiliary tokens. Let $\mathbf{H} \in \mathbb{R}^{B\times N\times C}$ denote the patch representations from the penultimate layer of the vision encoder, and $\mathbf{A} \in \mathbb{R}^{B\times h\times N\times N}$ denote the corresponding attention maps, where B, h, N, and C denote the batch size, number of attention heads, sequence length, and embedding dimensionality, respectively. Although CLS-to-patch attention provides a natural saliency cue, attention concentration may bias token selection.

To obtain more reliable token importance estimation Zhao et al. (2026); Ishibashi & Meng (2025); Fayyaz et al. (2022), we incorporate value norms as complementary information. Specifically, the value representations are computed as:

$$\mathbf{V} = W_V LN(\mathbf{H}), \tag{1}$$

where W_V denotes the value projection. The semantic saliency score of the i-th patch is defined as:

$$s_i = \sum_{m=1}^{h} A^{(m)}_{\mathrm{CLS}\to i} \|\mathbf{v}_i^{(m)}\|_2, \tag{2}$$

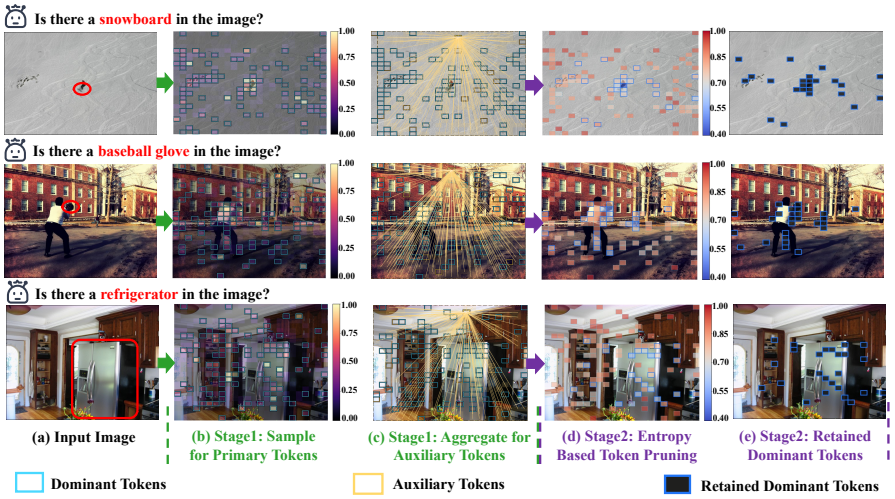

Figure 4: Visualization of the progressive token selection process of SCOUT. Stage 1 performs coarse-grained token selection by sampling Primary Tokens and aggregating Auxiliary Tokens, while Stage 2 refines the selection through entropy-guided pruning to obtain the Dominant Tokens.

where $A^{(m)}_{\text{CLS}\to i}$ and $\mathbf{v}_i^{(m)}$ denote the CLS-to-patch attention weight and value vector of the m-th head, respectively. Different from attention-only selection, the proposed score jointly considers whether a patch is attended by the CLS token and whether it carries informative visual representations. This design reduces the dependence on attention distributions alone and provides a more robust criterion for identifying informative visual regions.

Based on the semantic saliency scores, we further identify spatially coherent semantic regions to guide token selection. Instead of independently selecting patches based on their scores, we exploit the spatial structure of visual tokens to maintain semantic consistency. We convert the semantic saliency scores into a binary mask:

$$m_i = 1[s_i > \tau], \quad \tau = Q_\alpha(s), \tag{3}$$

where Q_α denotes the α-quantile threshold. Connected component labeling is then applied on the patch grid to obtain spatially coherent candidate regions $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$. For each spatially coherent region $\mathcal{C}_k$, we employ a sampling strategy $\mathcal{S}$ to select representative patches, ensuring that each region retains a compact yet informative token representation. If the number of selected tokens from all regions is smaller than the primary token budget T_p, the remaining slots are filled with the highest-scoring unselected patches. The resulting primary token set, denoted as $\mathbf{H}_p$, is formulated as follows, with qualitative examples shown in Figure 4(b):

$$\mathbf{H}_p = \{\mathbf{h}_{\text{CLS}}\} \cup \{\mathbf{h}_{\mathcal{C}_i}\}_{i=1}^{K}. \tag{4}$$

The remaining patches mainly contain background and contextual information. Instead of discarding them completely, we compress them into T_a auxiliary tokens. Specifically, we use the key representations from the same attention layer as the merging metric, since key states capture token affinity before attention aggregation Qiu et al. (2025). Anchor tokens are uniformly sampled from the remaining patches, and other tokens are merged according to cosine similarity. The aggregation mechanism and the corresponding results are shown in Figure 4(c). Finally, the visual tokens fed into the language model are obtained by concatenating primary and auxiliary tokens:

$$\mathbf{V} = \text{Concat}(\mathbf{H}_p, \mathbf{H}_a). \tag{5}$$

The quantitative results are reported in Table 6. More details are provided in Further Studies.

3.2.2 Entropy-Gated Visual Token Pruning.

Based on the above observation, Stage 2 performs fine-grained pruning on the tokens preserved by Stage 1. Unlike the vision encoder, the LLM decoder provides text-conditioned visual representations through cross-modal attention. Therefore, we exploit text-to-visual attention distributions to identify query-relevant tokens while mitigating attention-sink bias.

Formally, at layer k, the cross-attention weights between N_t text tokens and N_v visual tokens are represented as $\mathbf{A} \in \mathbb{R}^{h \times N_t \times N_v}$. The attention maps are averaged across heads as $\bar{\mathbf{A}} = \frac{1}{h}\sum_{m=1}^{h} \mathbf{A}^{(m)}$. For each visual token v_i, the attention weights assigned by text tokens are normalized into a probability distribution, where ϵ is a small constant for numerical stability:

$$p(q|v_i) = \frac{\bar{A}_{q,i}}{\sum_{q'} \bar{A}_{q',i} + \epsilon}, \tag{6}$$

and the corresponding attention entropy is computed as:

$$H(v_i) = -\sum_{q=1}^{N_t} p(q \mid v_i) \log(p(q \mid v_i) + \epsilon). \tag{7}$$

Low entropy indicates that the token has a selective association with a subset of text tokens, while high entropy reflects a more diffuse text-vision interaction. Therefore, entropy provides a measure of query selectivity complementary to attention magnitude. To eliminate the effect of varying text lengths, the entropy is normalized as $\bar{H}(v_i) = \frac{H(v_i)}{\log N_t}$. We then construct an adaptive entropy gate, where the threshold is obtained from the sample-wise entropy distribution as $\tau = Q_\rho(\{\bar{H}(v_i)\}_{i=1}^{N_v})$, with Q_ρ denoting the ρ-quantile function. We use $\rho = 0.5$ by default. To adapt the gating behavior to different samples, we estimate the entropy dispersion using the interquartile range:

$$\mathrm{IQR} = Q_{0.75}\left(\{\bar{H}(v_i)\}_{i=1}^{N_v}\right) - Q_{0.25}\left(\{\bar{H}(v_i)\}_{i=1}^{N_v}\right). \tag{8}$$

The sample-specific temperature is defined as:

$$T = \mathrm{clip}\left(\lambda\, \mathrm{IQR}, T_{\min}, T_{\max}\right), \tag{9}$$

where λ controls the gate sensitivity. A larger entropy dispersion produces a smoother gate, while a concentrated entropy distribution results in a sharper discrimination. Based on the adaptive threshold and temperature, the entropy gate for each visual token is formulated as:

$$G_i = \sigma\left(\frac{\tau - \bar{H}(v_i)}{T}\right). \tag{10}$$

When the entropy of a visual token is lower than the adaptive threshold, indicating a more selective text-vision association, the gate value approaches 1 and preserves its importance. In contrast, high-entropy tokens are suppressed to reduce the influence of diffuse attention patterns. The qualitative results in Figure 4(d) show that entropy-guided gating preserves more low-entropy, text-relevant tokens. Although entropy measures the selectivity of text-vision interaction, it does not capture the overall attention magnitude. Therefore, we combine the entropy gate with cross-modal attention strength, where $S_i = \sum_{q=1}^{N_t} \bar{A}_{q,i}$, and define the final importance score as:

$$\mathrm{Importance}_i = S_i \cdot G_i. \tag{11}$$

This entropy-calibrated score prioritizes visual tokens with strong cross-modal relevance, while suppressing tokens whose importance mainly arises from diffuse or biased attention distributions. Finally, we retain the top-T_2 tokens according to the importance scores and preserve their original order. The dominant tokens are visualized in Figure 4(e), and quantitative results are reported in Table 6. For more details on the method parameters, please refer to Appendix A.2.3.

4 Experiments

In this section, we evaluate the generalizability of SCOUT across different MLLM architectures and diverse image and video benchmarks against state-of-the-art methods. Furthermore, we analyze the efficiency gains brought by SCOUT. Finally, we conduct comprehensive ablation studies to assess the effectiveness of the proposed method.

Table 1: Performance comparison of different token pruning methods on LLaVA-v1.5-7B under varying visual token budgets. The best and second-best results are highlighted in bold and underlined, respectively. Avg. denotes the average performance relative to the vanilla model.

Method	GQA	SQA	POPE	MME	MMB	VQA^{V2}	VQA^{T}	MMMU	SEED	Avg.
Upper Bound, 576 Tokens (100%)										
Vanilla	61.9	69.5	85.9	1862	64.7	78.5	58.2	36.3	58.6	100.0%
Retain 192 Tokens (↓ 66.7%)										
FastV	52.7	67.3	64.8	1612	61.2	67.1	52.5	34.3	57.1	89.6%
SparseVLM	57.6	**69.1**	83.6	1721	62.5	75.6	56.1	33.8	55.8	95.5%
VisionZip	59.3	68.9	85.3	1782.6	63.0	76.8	57.3	**36.6**	56.4	97.9%
FlowCut	60.1	68.6	86.1	**1836**	63.2	77.1	57.5	**36.6**	57.5	98.7%
Ours	**60.6**	68.5	**86.7**	1800	**63.7**	**78.1**	**57.8**	36.0	**59.8**	**99.2%**
Retain 128 Tokens (↓ 77.8%)										
FastV	49.6	60.2	59.6	1490	56.1	61.8	50.6	34.9	55.9	84.4%
SparseVLM	56.0	67.1	80.5	1696	60.0	73.8	54.9	33.8	53.4	93.0%
VisionZip	57.6	**68.9**	83.2	1761.7	62.0	75.6	56.8	**37.9**	54.9	96.8%
FlowCut	58.5	68.6	85.2	**1792**	62.1	76.0	57.3	37.2	56.2	97.6%
Ours	**59.7**	**68.9**	**86.1**	1781	**63.5**	**77.4**	**57.5**	37.1	**58.8**	**98.8%**
Retain 64 Tokens (↓ 88.9%)										
FastV	46.1	51.1	48.0	1256	48.0	55.0	47.8	34.0	51.9	75.6%
SparseVLM	52.7	62.2	75.1	1505	56.2	68.2	51.8	32.7	51.1	87.0%
VisionZip	55.1	69.0	77.0	1690	60.1	72.4	55.5	36.2	52.2	93.1%
FlowCut	55.6	**69.1**	80.2	1744	60.8	72.8	55.6	**36.8**	53.5	94.6%
Ours	**57.7**	68.8	**82.8**	**1764**	**61.4**	**75.4**	**56.1**	36.3	**56.2**	**96.3%**

4.1 Experimental Setting

For image understanding tasks, we evaluate SCOUT on nine widely used benchmarks, including GQA Hudson & Manning (2019), SQA Lu et al. (2022), POPE Li et al. (2023c), MME Fu et al. (2023), MMBench Liu et al. (2024b), VQA^{V2} Goyal et al. (2017), VQA^{T} Singh et al. (2019), MMMU Yue et al. (2024), and SEED-Bench Li et al. (2023a), which cover diverse image understanding capabilities. We compare our method with SOTA visual token pruning approaches, including FastV, SparseVLM, VisionZip, and FlowCut, across diverse model architectures from the LLaVA and Qwen families, such as LLaVA-v1.5-7B and Qwen3-VL. Further results and details are included in the supplementary material. For video understanding tasks, we evaluate our method on Video-LLaVA across three video question answering benchmarks: TGIF-QA Jang et al. (2017), MSVD-QA Xu et al. (2017), and MSRVTT-QA Xu et al. (2017). We use GPT-5-nano to evaluate the generated responses. All experiments are conducted on NVIDIA A6000 GPUs. The remaining hyperparameters and further implementation details are provided in the supplementary material.

4.2 Performance and Efficiency Comparison

In this section, we compare SCOUT with state-of-the-art methods on image-based tasks using LLaVA-v1.5-7B and LLaVA-NeXT-7B, as well as on video-based tasks using Video-LLaVA. We further evaluate its inference efficiency and KV cache reduction.

Results on LLaVA-v1.5-7B. For LLaVA-v1.5-7B, each image is represented by 576 visual tokens, which are pruned to 192, 128, and 64 tokens. As shown in Table 1, SCOUT retains 99.2% average performance with 192 tokens and 96.3% with only 64 tokens, outperforming previous SOTA methods by 1.7%. These results show that SCOUT better preserves query-relevant visual tokens, enabling the model to achieve superior performance under the same pruning ratio.

Results on LLaVA-NeXT-7B. Compared with LLaVA-v1.5-7B, LLaVA-NeXT-7B adopts AnyRes with 2880 visual tokens, which are pruned to 640, 320, and 160 tokens. As shown in Table 2, SCOUT preserves 99.6% average performance with 640 tokens and 94.6% with only 5.6% tokens retained, demonstrating its robustness under extreme pruning and high-resolution inputs.

Results on Video-LLaVA. Video-LLaVA samples eight frames, each encoded into 256 visual tokens by LanguageBind, resulting in 2048 tokens per video. Following prior work, we evaluate pruning methods on the first 1,000 samples of each dataset while retaining 256 tokens. As shown in Table 3, SCOUT achieves the best performance, with an average score of 96.1%.

Table 2: Performance comparison of different pruning methods on LLaVA-NeXT-7B under varying visual token budgets. The best and second-best results are highlighted in bold and underlined, respectively. Avg. denotes the average performance relative to the vanilla model.

Method	GQA	SQA	POPE	MME	MMB	VQA^{V2}	VQA^{T}	MMMU	SEED	Avg.
Upper Bound, 2,880 Tokens (100%)										
Vanilla	64.2	70.2	86.4	1849	67.9	80.1	61.3	35.1	63.6	100.0%
Retain 640 Tokens (↓ 77.8%)										
SparseVLM	60.3	67.7	84.2	1772	65.7	77.1	57.8	34.6	59.3	95.9%
VisionZip	61.3	68.1	85.9	1787	66.3	79.1	60.2	34.7	60.9	97.5%
FlowCut	61.9	68.2	86.1	**1837**	**66.7**	79.8	60.6	34.6	61.9	98.4%
Ours	**62.0**	**68.7**	**87.4**	1823	65.8	**80.6**	**61.0**	**37.1**	**63.4**	**99.6%**
Retain 320 Tokens (↓ 88.9%)										
SparseVLM	57.7	67.3	82.2	1685	63.2	73.4	55.9	34.4	56.9	93.1%
VisionZip	59.3	67.3	82.1	1702	63.1	76.2	58.9	35.3	58.2	94.6%
FlowCut	59.8	67.3	83.4	**1791**	**65.3**	77.8	60.1	35.0	59.5	96.3%
Ours	**61.1**	**67.5**	**85.3**	1748	64.9	**78.9**	**60.5**	**36.7**	**61.5**	**97.6%**
Retain 160 Tokens (↓ 94.4%)										
SparseVLM	51.2	67.5	72.7	1542	52.1	66.3	46.4	32.8	49.2	85.1%
VisionZip	55.5	**68.3**	74.8	1628	60.1	71.4	56.2	36.1	54.2	90.7%
FlowCut	57.6	66.7	79.9	**1746**	62.8	74.6	**57.6**	36.0	56.8	93.7%
Ours	**59.6**	67.8	**81.2**	1704	**63.0**	**75.7**	56.2	**36.3**	**59.2**	**94.6%**

Table 3: Comparison of pruning methods on Video-LLaVA. Best and second-best results are highlighted in bold and underlined, respectively.

Method	TGIF		MSVD		MSRVTT		Avg.	
	Acc.	Score	Acc.	Score	Acc.	Score	Acc.	Score
Upper Bound, 2,048 Tokens (100%)								
Vanilla	13.7	0.84	57.9	3.00	51.7	2.59	100.0%	100.0%
Retain an Average of 256 Tokens								
SparseVLM	8.1	0.53	49.8	2.59	45.9	2.29	78.0%	79.3%
VisionZip	12.2	0.72	**57.9**	**2.97**	48.7	2.44	94.4%	93.0%
FlowCut	12.5	0.75	57.0	2.91	**50.6**	**2.55**	95.8%	94.9%
Ours	**12.8**	**0.81**	57.5	2.94	49.5	2.48	**96.1%**	**96.7%**

Table 4: Efficiency analysis of SCOUT. Total Time ↓, Prefill Time ↓, and KV Cache ↓ are reported in s, ms, and MB, respectively.

Method	Token	Total Time	Prefill Time	KV Cache	TFLOPs
LLaVA-1.5-7B	576	1,506	105.6	318.9	8.47
+SCOUT	192	$1{,}257_{1.2\times}$	$64.2_{1.6\times}$	$131.9_{2.4\times}$	$3.46_{2.4\times}$
	64	$1{,}065_{1.4\times}$	$42.5_{2.5\times}$	$64.5_{4.9\times}$	$1.68_{5.0\times}$
LLaVA-NeXT-7B	2,880	4,515	370.2	1,156.1	29.95
+SCOUT	640	$3{,}174_{1.4\times}$	$188.0_{2.0\times}$	$580.1_{2.0\times}$	$15.03_{2.0\times}$
	160	$2{,}089_{2.2\times}$	$71.1_{5.2\times}$	$168.1_{6.9\times}$	$4.36_{6.9\times}$

Efficiency Analysis. In addition to preserving model performance under visual token pruning, we further evaluate inference efficiency and KV cache reduction on POPE using LLaVA-v1.5-7B and LLaVA-NeXT-7B. As summarized in Table 4, SCOUT achieves a 2.2× end-to-end speedup, 5.2× prefill acceleration, and 6.9× KV cache compression on LLaVA-NeXT-7B.

4.3 Further Studies

To evaluate the effectiveness of the components in the proposed two-stage SCOUT framework and analyze the impact of hyperparameters, we conduct ablation studies on multiple datasets using the LLaVA-v1.5-7B model with 64 retained visual tokens. Additionally, we conduct experiments on models with different parameter scales, such as LLaVA-v1.5-13B, and under smaller token budgets, such as 32 visual tokens. More results are provided in the Appendix B.1.

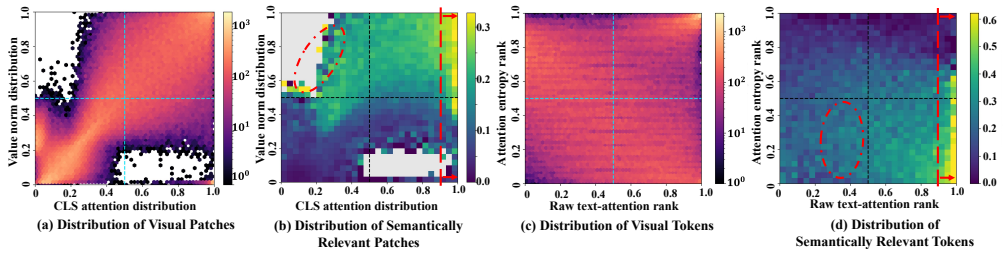

Figure 5: Analysis of SCOUT token selection on the POPE dataset. (a-b) Stage 1 analysis of the relationship between CLS attention and Value Norm. (c-d) Stage 2 analysis of the relationship between raw text-attention rank and entropy.

Table 5: Effectiveness of token pruning strategies in Stages 1 and 2. ✓ denotes the proposed strategy; ✓† and ✓‡ denote variants without value normalization and Stage 2 entropy gating, respectively. Top-k selects the highest-attention tokens, while Random selects tokens randomly.

Stage 1	Stage 2	GQA	POPE	MME
✓	✓	**57.7**	82.8	**1764**
✓†	✓	57.5	82.6	1761
Top-k	✓	57.0	82.1	1736
✓	Random	56.5	82.7	1730
✓	✓‡	57.2	**83.0**	1729

Table 6: Token selection quality on POPE at the two pruning stages. *Contamination* measures the proportion of selected tokens deemed undesirable. *Recall* measures coverage of object instances in Stage 1 and semantic regions in Stage 2. Metric definitions are provided in the Appendix A.2.2.

Stage	Strategy	Contamination ↓	Recall ↑
Stage 1	Attention Top-k	15.8	87.1
	Atten$\times$Value Top-k	13.4	87.6
	Ours	**12.6**	**88.0**
Stage 2	Raw Attention Top-k	50.5	31.1
	Entropy-Gated Top-k	**5.6**	**42.8**

4.3.1 Ablation Studies: Impact of pruning strategies in different stages.

Table 5 evaluates the token pruning strategies in the two stages of SCOUT. The full model achieves the best overall performance. Removing value normalization in Stage 1 causes a slight degradation, while replacing the proposed strategy with attention-based Top-k selection further reduces performance, validating the effectiveness of the remaining Stage 1 design choices. Similarly, removing entropy-guided gating in Stage 2 and relying solely on attention also degrades performance. We additionally compare against random token selection, which yields inferior results.

4.3.2 Token Selection Reliability Analysis.

Although SCOUT improves performance across benchmarks, we further analyze whether its two stages achieve their intended goals. Stage 1 preserves semantic regions during coarse-grained token pruning, while Stage 2 mitigates attention sink-like tokens via text-guided entropy gating. We perform quantitative and qualitative analyses as follows on POPE for validation.

Is Stage 1 Effective in Preserving Semantically Meaningful Tokens? Stage 1 jointly considers attention scores and Value representations for more reliable token importance estimation. As shown in Table 6, incorporating Value information reduces contamination and improves recall, indicating that Value-aware selection mitigates over-reliance on attention scores and preserves more semantic regions. We further provide qualitative analysis in Figure 5(a-b). Figure 5(a) shows the distribution of visual patches in the CLS attention–Value Norm space, while Figure 5(b) presents semantically relevant patches in the same space. The results show that meaningful patches are mainly distributed in the high-attention and high-value norm region, with some located in the low-attention but high-value norm region. Therefore, attention-only selection (right of the vertical dashed line) may discard valuable regions, whereas incorporating value information (within the dashed ellipse) enables Stage 1 to retain complementary semantic regions for effective coarse-grained filtering.

Is Stage 2 Effective in Mitigating Attention Sink-like Behaviors? Stage 2 refines Stage 1 tokens through entropy-guided gating. As shown in Table 6, entropy-guided selection reduces contamination and improves semantic-region recall compared with attention alone, demonstrating the complementary role of entropy in identifying less discriminative high-attention tokens. Figure 5(c-d) shows that semantically relevant tokens generally exhibit lower entropy, while high-attention and high-entropy tokens tend to be less task-relevant and exhibit attention sink-like behaviors. Thus, entropy-guided gating effectively mitigates attention sink-like tokens and enables more precise text-aware token pruning.

5 Conclusion

In this work, we revisited the role of attention in visual token pruning for MLLMs. We showed that vanilla attention scores are not always reliable indicators of token importance, due to the unimodal nature of intra-vision self-attention and the attention sink phenomenon in Transformers. To address this challenge, we proposed SCOUT, a two-stage visual token pruning framework that combines coarse-grained semantic region filtering in the vision encoder with cross-modal attention refinement in the language model. Experiments on multiple MLLM architectures across various benchmarks show that our method consistently improves the trade-off between efficiency and performance.

AI USE STATEMENT

We used large language models to correct grammar and occasionally improve the clarity of sentence-level expression in this manuscript. We reviewed all AI-suggested edits to ensure that they preserved the intended technical meaning. All research ideas, experimental designs, and analyses presented in this work were developed and carried out by the authors. We take full responsibility for the final text, claims, and results of this work.

ETHICS STATEMENT

We propose a visual token pruning method to improve the inference efficiency of multimodal large language models (MLLMs). All models and datasets used in this study are publicly available. We are not aware of any ethical concerns specific to the proposed pruning method beyond those associated with the underlying models and datasets.

REPRODUCIBILITY STATEMENT

All models and datasets used in this work are publicly available and are identified with citations in the paper. The main text describes the two stages of SCOUT, including visual token scoring and selection, region aggregation, and entropy-guided pruning. The experimental section and appendix document the model configurations, token budgets, parameter settings, datasets, and evaluation protocols used to obtain the reported results. Upon acceptance, we will release the source code along with the configurations and instructions needed to reproduce our experiments.

REFERENCES

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

David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 190–200, 2011.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European conference on computer vision*, pp. 396–414. Springer, 2022.

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

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*, 2024.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Ryuto Ishibashi and Lin Meng. Automatic pruning rate adjustment for dynamic token reduction in vision transformer: R. ishibashi and l. meng. *Applied Intelligence*, 55(5):342, 2025.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2758–2766, 2017.

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

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

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*, pp. 19730–19742. PMLR, 2023b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023c.

Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4641–4650, 2016.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Zhixuan Lin, Evgenii Nikishin, Xu Owen He, and Aaron Courville. Forgetting transformer: Softmax attention with a forget gate. *arXiv preprint arXiv:2503.02130*, 2025.

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

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

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

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*, 35:2507–2521, 2022.

AI Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, 2025.

Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. Lightning attention-2: A free lunch for handling unlimited sequence lengths in large language models. *arXiv preprint arXiv:2401.04658*, 2024.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025.

Zihan Qiu, Zeyu Huang, Kaiyue Wen, Peng Jin, Bo Zheng, Yuxin Zhou, Haofeng Huang, Zekun Wang, Xiao Li, Huaqing Zhang, et al. A unified view of attention and residual sinks: Outlier-driven rescaling is essential for transformer training. *arXiv preprint arXiv:2601.22966*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021.

Zhenwei Shao, Mingyang Wang, Zhou Yu, Wenwen Pan, Yan Yang, Tao Wei, Hongyuan Zhang, Ning Mao, Wei Chen, and Jun Yu. Growing a twig to accelerate large vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20064–20074, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Jintao Tong, Wenwei Jin, Pengda Qin, Anqi Li, Yixiong Zou, Yuhong Li, Yuhua Li, and Ruixuan Li. Flowcut: Rethinking redundancy via information flow for efficient vision-language models. *arXiv preprint arXiv:2505.19536*, 2025.

Jintao Tong, Wenwei Jin, Pengda Qin, Anqi Li, Yixiong Zou, Yuhong Li, Yuhua Li, and Ruixuan Li. Flowcut: Rethinking redundancy via information flow for efficient vision-language models. *Advances in Neural Information Processing Systems*, 38:94946–94973, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1645–1653, 2017.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296, 2016.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, et al. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19803–19813, 2025b.

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

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24972–24982, 2025.

Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan Celine Lin. Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. *arXiv preprint arXiv:2406.15765*, 2024.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, pp. 74840–74857. PMLR, 2025.

Kai Zhao, Wubang Yuan, Yuchen Lin, Liting Ruan, Xiaofeng Lu, Deng-Ping Fan, Ming-Ming Cheng, and Dan Zeng. Attention debiasing for token pruning in vision language models, 2026. URL https://arxiv.org/abs/2508.17807.

Zayd MK Zuhri, Erland Hilman Fuadi, and Alham Fikri Aji. Softpick: No attention sink, no massive activations with rectified softmax. *arXiv preprint arXiv:2504.20966*, 2025.

APPENDIX

In this Appendix, we provide additional details to complement the experimental analysis and further explain the related components presented in the main paper. Section A Experiment and Implementation Details elaborates on more experimental and implementation details of our framework. Section B Additional Results and Deeper Analysis provides additional quantitative comparisons, extended analyses, qualitative results, to demonstrate the effectiveness of SCOUT.

A EXPERIMENT AND IMPLEMENTATION DETAILS

In this section, we firstly provide the details of the benchmark datasets used in this work. Secondly, we describe the implementation details of the key modules in SCOUT.

A.1 EVALUATION BENCHMARKS

We employ nine image-based benchmarks, including GQA Hudson & Manning (2019), SQA Lu et al. (2022), POPE Li et al. (2023c), MME Fu et al. (2023), MMBench Liu et al. (2024b), VQA^{V2} Goyal et al. (2017), VQA^{T} Singh et al. (2019), MMMU Yue et al. (2024), and SEED-Bench Li et al. (2023a), to comprehensively evaluate the model's capabilities in visual understanding, reasoning, and other aspects of multimodal perception. To further assess the effectiveness of the pruning algorithm on video-related tasks, we conduct additional evaluations on three video question answering datasets, including TGIF-QA Jang et al. (2017), MSVD-QA Xu et al. (2017), and MSRVTT-QA Xu et al. (2017).

GQA Hudson & Manning (2019) is a large-scale visual question answering benchmark that provides images annotated with detailed scene graphs describing objects, attributes, and relationships. It contains compositional questions designed to evaluate a model's ability to perform structured visual reasoning over complex scenes. The benchmark emphasizes spatial relations, object attributes, and multi-step reasoning within real-world images.

ScienceQA Lu et al. (2022) is a multimodal question-answering benchmark covering a wide range of scientific disciplines, including natural, language, and social science. Questions are hierarchically organized into topics, categories, and skills, enabling evaluation of multimodal understanding and multi-step reasoning abilities. Since some questions are text-only, we report results only on the subset with visual inputs.

POPE Li et al. (2023c) is designed to evaluate object hallucination in vision-language models. It formulates binary questions regarding the presence of objects within images and assesses model predictions using accuracy, recall, precision, and F1-score. The reported results are typically computed as the mean F1-score across three sampling strategies: adversarial, random, and popular.

MME Fu et al. (2023) evaluates both perceptual and cognitive abilities of multimodal models through 14 subtasks. Perception tasks cover coarse and fine grained object recognition, including object existence, count, position, color, and recognition of scenes, landmarks, artworks, and celebrities. Cognitive tasks assess abilities such as commonsense reasoning, numerical calculation, translation, and code reasoning.

MMBench Liu et al. (2024b) is a comprehensive benchmark for evaluating multimodal large language models across multiple capability dimensions. It organizes evaluation into three hierarchical levels: Level-1 focuses on fundamental perception and reasoning abilities, Level-2 expands these into six sub-abilities, and Level-3 further decomposes them into twenty fine-grained tasks.

VQA-V2 Goyal et al. (2017) is a large-scale visual question answering benchmark built upon images from the MSCOCO dataset Lin et al. (2014). It contains 265,016 real-world images paired with open-ended questions, each annotated with ten human-provided ground truth answers. The dataset evaluates models' abilities to understand visual content and generate accurate responses to queries.

TextVQA Singh et al. (2019) focuses on the understanding of textual information embedded within images. It evaluates a model's ability to interpret visual elements and understand text embedded within images. During the experimental process, optical character recognition (OCR) results are provided to assist the model in recognizing text in the images.

MMMU Yue et al. (2024) evaluates multimodal models on tasks requiring college-level knowledge and reasoning. It contains 11.5K curated questions from exams, quizzes, and textbooks across six disciplines: Art & Design, Business, Science, Health & Medicine, Humanities & Social Science, and Tech & Engineering. Covering 30 subjects and 183 subfields, the benchmark includes about 30 image types such as charts, diagrams, and chemical structures, challenging models with advanced perception and domain-specific reasoning.

SeedBench Li et al. (2023a) is a large-scale benchmark consisting of 19,000 human-annotated multiple-choice questions. It evaluates multimodal models across 12 capability dimensions, including spatial and temporal reasoning over both images and videos.

TGIF-QA Jang et al. (2017) extends visual question answering to the video domain using animated GIFs. The benchmark contains approximately 165K question–answer pairs derived from the Tumblr GIF dataset Li et al. (2016). It introduces four task types, including repetition count, repeating action, state transition, and frame-based QA, requiring spatial-temporal reasoning.

MSVD-QA Xu et al. (2017) is built upon the MSVD video dataset Chen & Dolan (2011) and consists of 1,970 short video clips paired with around 50.5K question–answer pairs. Questions are open-ended and span five categories: what, who, how, when, and where, evaluating models' ability to understand video content.

MSRVTT-QA Xu et al. (2017) is constructed from the MSRVTT dataset Xu et al. (2016) and contains 10K video clips with 243K question–answer pairs. Compared with MSVD-QA, it includes more complex scenes and activities, requiring models to integrate both visual and temporal information for accurate video understanding.

A.2 Implementation Details

In this section, we provide the implementation details of our experiments, including the token calculation strategy, detailed explanations of relevant definitions, and hyperparameter settings.

A.2.1 Strategy for Retaining Tokens.

Since visual token pruning primarily reduces the number of tokens involved in the prefill stage, where self-attention exhibits quadratic complexity with respect to the token length, we focus on comparing the TFLOPs of the prefill stage to evaluate the efficiency of different pruning strategies. In this stage, the decoder processes system, text, and visual tokens, while output tokens are generated in the subsequent autoregressive decoding stage and are therefore excluded from the FLOPs calculation. Specifically, the TFLOPs can be estimated as follows:

$$\text{TFLOPs} = T \times (4nd^2 + 2n^2d + 2ndm) \tag{12}$$

where T denotes the number of Transformer layers, n represents the sequence length, d is the hidden state dimension, and m denotes the intermediate dimension of the FFN. The sequence length n is defined as the total number of tokens in the input sequence, including system tokens, visual tokens, and textual tokens.

Since the proposed SCOUT is a two-stage visual token pruning method, the number of tokens retained at each stage is determined according to the theoretical TFLOPs. Specifically, during the experiments, we adjust the number of retained tokens such that the theoretical TFLOPs of the LLM decoding process remain approximately the same as those of other methods Yang et al. (2025c).

A.2.2 Metrics and Visualization Details for Token Selection Analysis.

In the main paper, we introduce the Token Selection Reliability Analysis to quantitatively and qualitatively verify that SCOUT achieves semantic region-aware filtering in Stage 1 and entropy-guided

cross-modal fine-grained pruning in Stage 2. In this section, we provide a more detailed description of this analysis, including the definitions of contamination and recall metrics for assessing the reliability of token selection in both stages.

Stage 1. Stage 1 performs image-side coarse filtering before language-conditioned reasoning. We define undesirable tokens as sink-like visual patches, i.e., patches with high CLS-to-patch attention but low value norm. Specifically, given the CLS attention score a_i and value norm n_i of patch i, a patch is considered sink-like if

$$a_i > Q_{0.9}(\{a_i\}) \quad \text{and} \quad n_i < Q_{0.5}(\{n_i\}),$$

where $Q_\rho(\cdot)$ denotes the ρ-quantile operator. The Stage-1 contamination is then defined as the fraction of selected tokens that fall into this sink-like set:

$$\text{Contamination}_{\text{S1}} = \frac{|\mathcal{S}_{\text{S1}} \cap \mathcal{U}_{\text{S1}}|}{|\mathcal{S}_{\text{S1}}|},$$

where $\mathcal{S}_{\text{S1}}$ is the selected token set and $\mathcal{U}_{\text{S1}}$ is the sink-like token set.

To measure semantic coverage, we use COCO instance annotations associated with POPE images as a proxy. A patch is considered relevant if its overlap with the target-object bounding box exceeds a fixed threshold, which is set to 0.25. Stage-1 recall measures object-instance coverage: an object instance is counted as covered if at least one selected token overlaps with its relevant patch region. Therefore,

$$\text{Recall}_{\text{S1}} = \frac{|\mathcal{I}_{\text{covered}}|}{|\mathcal{I}_{\text{target}}|},$$

where $\mathcal{I}_{\text{target}}$ denotes the set of target object instances, and $\mathcal{I}_{\text{covered}}$ denotes the subset of target instances covered by at least one selected token.

Stage 2. Stage 2 performs language-conditioned pruning inside the LLM using text-to-vision attention. In this stage, undesirable tokens are defined as high-attention but high-entropy visual tokens. These tokens receive large total attention from text tokens, but the attention is diffusely distributed across many text positions, indicating weak token-specific alignment. Therefore, the selected tokens may not necessarily correspond to semantic regions relevant to the specific textual query.

We rank visual tokens by raw text-to-vision attention and attention entropy within each prompt. A token is treated as undesirable if it lies in the high-attention and high-entropy region:

$$r_i^{\text{attn}} \geq 0.5 \quad \text{and} \quad r_i^{\text{ent}} \geq 0.5,$$

where r_i^{attn} and r_i^{ent} denote the within-prompt rank of raw attention and normalized entropy, respectively. The Stage-2 contamination is defined as

$$\text{Contamination}_{\text{S2}} = \frac{|\mathcal{S}_{\text{S2}} \cap \mathcal{U}_{\text{S2}}|}{|\mathcal{S}_{\text{S2}}|}.$$

Compared with Stage-1 recall, which focuses on instance-level semantic coverage and evaluates whether the selected tokens preserve diverse target object instances, Stage-2 recall measures semantic evidence retention by quantifying how much target-related semantic information is preserved after fine-grained pruning. Therefore, the two recall metrics evaluate token selection reliability from complementary perspectives.

For Stage 2 recall, we propagate COCO patch-level relevance to the Stage-1 visual tokens. Dominant tokens inherit binary patch relevance, while auxiliary tokens receive aggregation-weighted relevance according to the patches merged into them. We then compute semantic-region coverage as relevance mass recall:

$$\text{Recall}_{\text{S2}} = \frac{\sum_{i \in \mathcal{S}_{\text{S2}}} r_i}{\sum_i r_i},$$

where r_i denotes the semantic relevance score of token i. This metric measures how much target-related semantic evidence is retained after Stage-2 pruning.

We provide additional details for the heatmap visualizations in Further Studies. Stage 1 is analyzed at the image-level patch space, while Stage 2 is analyzed at the prompt-level token space. Semantic

Table 7: Two-Stage Token Retention in LLaVA-v1.5-7B. 49_7 indicates that 49 primary tokens and 7 auxiliary tokens are retained in Stage 1, totaling 56 tokens.

	32 tokens		64 tokens		128 tokens		192 tokens	
	S_1	S_2	S_1	S_2	S_1	S_2	S_1	S_2
LLaVA-v1.5-7B	49_7	14	98_{14}	28	196_{28}	56	294_{42}	84

Table 8: Two-Stage Token Retention in LLaVA-NeXT-7B.

	160 tokens		320 tokens		640 tokens	
	S_1	S_2	S_1	S_2	S_1	S_2
LLaVA-NeXT-7B	245_{35}	70	490_{70}	140	980_{140}	280

relevance is estimated using COCO instance annotations associated with POPE images, where object bounding boxes are mapped to the corresponding patch/token regions.

For Stage 1, the horizontal and vertical axes correspond to the normalized CLS-to-patch attention score and the normalized value norm, respectively, following the definitions in the Method section. For Stage 2, the horizontal axis denotes the sample-wise rank of the raw text-to-visual attention score,

$$S_i = \sum_{q=1}^{N_t} \bar{A}_{q,i},$$

and the vertical axis denotes the sample-wise rank of the normalized attention entropy $\bar{H}(v_i)$. A larger rank indicates a larger attention magnitude or entropy within the same prompt.

Subfigures (a) and (c) show the token density distribution, where the color intensity corresponds to the number of tokens in each bin, displayed in log scale. Subfigures (b) and (d) show the semantic relevance intensity, where the color indicates the average COCO-based semantic relevance of tokens falling into each bin.

Each heatmap is divided into four regions for interpretation. In Stage 1, the high-attention/low-value-norm region is likely to contain sink-like patches, while the high-attention/high-value-norm region is more likely to contain object-relevant patches. The remaining two regions correspond to low-attention/low-value-norm and low-attention/high-value-norm patches. In Stage 2, the high-attention/high-entropy region tends to contain undesirable diffuse-attention tokens, whereas the high-attention/low-entropy region is more likely to contain text-relevant visual tokens with concentrated cross-modal alignment.

The red dashed annotations provide qualitative guidance for interpreting the selection behavior. The vertical dashed regions highlight tokens that would be favored by attention-only selection due to their high attention scores, while the dashed elliptical regions highlight the regions emphasized after incorporating the proposed correction terms, i.e., value-norm awareness in Stage 1 and entropy-guided gating in Stage 2. These annotations are intended as visual aids rather than strict decision boundaries.

A.2.3 Hyperparameter Settings.

Regarding the placement of the two-stage pruning strategy, the first-stage pruning is applied to the output of the vision encoder. For the second-stage pruning within the LLM, an ablation study on LLaVA-v1.5-7B shows that the best performance is achieved when the pruning layer index is set to k=14. Therefore, we adopt this setting for LLaVA-v1.5-7B. For the remaining models, the value of k is determined by maintaining the same relative position within the decoder stack as that of LLaVA-v1.5-7B. Specifically, the LLM decoder of LLaVA-v1.5-7B contains 32 layers, with k=14, corresponding to approximately 43.75% of the total depth. Accordingly, k is set to 18 for LLaVA-v1.5-13B, which contains 40 decoder layers, and to 16 for Qwen3-VL-8B, whose LLM decoder consists of 36 layers.



Figure 6: The principle of selecting patches from clusters

Table 9: Two-Stage Token Retention in LLaVA-v1.5-13B.

	64 tokens		128 tokens		192 tokens	
	S_1	S_2	S_1	S_2	S_1	S_2
LLaVA-v1.5-13B	94_{14}	28	188_{28}	56	282_{42}	84

Table 10: Two-Stage Token Retention in Qwen3-VL-8B-Instruct. $0.5_{0.06}$ indicates that 50% of the primary tokens and 6% of the auxiliary tokens, measured relative to their original sequence lengths, are retained in Stage 1.

	Reduction (↓ 66.7%)		Reduction (↓ 77.8%)		Reduction (↓ 88.9%)	
	S_1	S_2	S_1	S_2	S_1	S_2
Qwen3-VL-8B-Instruct	$0.5_{0.06}$	0.28	$0.34_{0.04}$	0.28	$0.17_{0.02}$	0.28

During the token pruning process on the LLaVA-v1.5-7B model, the token configurations for the two pruning stages under different token budgets are shown in Table 7. In addition, for LLaVA-NeXT-7B, where each image is divided into five sub-images as inputs, the token configurations for the two pruning stages are shown in Table 8. The token retention settings adopted in the two-stage pruning strategy for LLaVA-v1.5-13B and Qwen3-VL-8B-Instruct are summarized in Tables 9 and 10, respectively. Note that, owing to the dynamic-resolution mechanism employed by Qwen3-VL-8B-Instruct, the retained tokens are specified as a proportion of the original sequence length rather than as absolute token counts.

The remaining hyperparameter settings used in the experiments are described as follows. In Stage 1, namely semantic region-centered coarse-grained filtering, we set α to 0.7 for all models except LLaVA-NeXT and LLaVA-v1.5-7B with 32 retained tokens, for which α is set to 0.8. For the representative patch selection strategy $\mathcal{S}$ from each cluster $\mathcal{C}_k$, the patch with the highest objectness score within each cluster is selected.

After computing the objectness score for each visual patch, we threshold the score map to obtain a binary foreground mask. Since visual tokens preserve the original 2D patch order, the 1D mask is reshaped into an $H \times W$ grid. We then apply connected component labeling with 4-neighborhood connectivity to group spatially adjacent foreground patches into connected clusters $\{\mathcal{C}_k\}$. Specifically, two foreground patches are assigned to the same cluster if they are connected through a path of foreground patches using horizontal or vertical adjacency on the patch grid. A representative example of the token processing procedure is illustrated in Figure 6.

For the representative patch selection strategy $\mathcal{S}$, we select the patch with the highest objectness score from each cluster $\mathcal{C}_k$. If the number of clusters exceeds the token budget, we retain the clusters with the highest average objectness scores; otherwise, the remaining token slots are filled with the

Table 11: Performance comparison of different token pruning methods on Qwen3-VL-8B-Instruct.

Method	SQA	GQA	POPE	MME	MMB	MMB^{CN}	Avg.
Upper Bound, All Tokens (100%)							
Vanilla	94.4	61.6	88.5	2373	84.8	82.6	100%
Token Reduction (↓ 66.7%)							
FastV	87.5	55.5	85.6	2196	80.5	79.6	93.9%
SparseVLM	88.6	56.4	83.7	2144	80.4	78.9	93.5%
VisionZip	**92.3**	59.1	87.5	**2278**	**82.4**	79.6	**97.0%**
FlowCut	81.0	55.0	81.8	1876	73.3	73.2	86.9%
Ours	88.3	**59.9**	**88.0**	2206	82.0	**79.7**	96.1%
Token Reduction (↓ 77.8%)							
FastV	84.4	51.8	81.3	1954	75.9	73.7	87.7%
SparseVLM	85.3	53.9	80.7	2055	78.4	75.2	89.9%
VisionZip	**88.8**	56.4	85.9	2084	79.1	76.0	92.7%
FlowCut	80.3	54.0	80.8	1809	72.9	72.1	85.6%
Ours	87.0	**58.9**	**87.4**	**2173**	**79.8**	**77.8**	**94.4%**
Token Reduction (↓ 88.9%)							
FastV	78.8	45.8	69.3	1594	63.0	63.6	75.8%
SparseVLM	80.2	49.2	71.3	1773	70.4	69.1	81.2%
VisionZip	83.3	51.5	78.3	1790	71.4	68.7	83.9%
FlowCut	80.1	49.6	76.5	1713	69.2	68.7	81.5%
Ours	**84.5**	**54.4**	**84.5**	**2022**	**75.8**	**72.7**	**89.3%**

highest-scoring unselected patches. This spatial clustering step reduces redundant selections from the same local region and encourages the retained representative tokens to cover distinct salient regions.

As for auxiliary tokens, we perform token merging on the remaining patches to further reduce redundancy. Specifically, we adopt the key representations from the same attention layer as the similarity metric for token merging, since key states capture token affinity prior to attention aggregation. Anchor tokens are uniformly sampled from the remaining patches. Each non-anchor token is assigned to its nearest anchor based on cosine similarity in the normalized key space, and tokens associated with the same anchor are merged through average aggregation. The aggregated feature is subsequently added to the corresponding anchor token, producing the final auxiliary token representation.

For Stage 2, we investigate two hyperparameters in the entropy-guided gating mechanism, including the quantile parameter ρ and the temperature scaling factor λ. We set $\rho = 0.5$ by default, which corresponds to the median of the sample-wise entropy distribution and provides a robust adaptive threshold for different samples. For λ, which controls the sensitivity of the entropy gate through the sample-specific temperature, we adopt $\lambda = 1/4$ in all experiments, achieving a favorable balance between preserving informative visual tokens and suppressing noisy attention patterns.

In Section B Additional Results and Deeper Analysis, we provide further empirical analysis on the sensitivity of these hyperparameters, validating the effectiveness of the selected default values for Stage1 and Stage2.

B Additional Results and Deeper Analysis

In this section, we conduct additional experiments to further validate the effectiveness and generalizability of SCOUT across various settings, including different model families (Qwen3-VL-8B-Instruct), larger-scale models (LLaVA-v1.5-13B), and extreme token compression with only 32 retained tokens on LLaVA-v1.5-7B. We also perform ablation studies to examine the effectiveness of different design choices and the sensitivity of key hyperparameters.

Table 12: Performance comparison of different token pruning methods on LLaVA-v1.5-13B.

Method	GQA	SQA	POPE	MME	MMB	MMMU	SEED-I	Avg.
Upper Bound, All Tokens (100%)								
Vanilla	63.2	72.8	85.9	1818	67.7	36.4	66.9	100%
Retain 192 Tokens								
VisionZip	59.1	73.5	85.1	1754	66.9	**36.4**	65.2	98.1%
FlowCut	59.8	**73.7**	**85.9**	**1819**	66.9	36.3	65.5	98.9 %
Ours	**60.8**	73.3	85.8	1780	**67.9**	36.1	**66.9**	**99.2%**
Retain 128 Tokens								
VisionZip	57.9	74.0	**85.2**	1743	66.7	36.1	63.8	97.3%
FlowCut	58.6	73.6	84.5	1772	66.8	**37.0**	64.5	98.1%
Ours	**59.3**	**74.7**	85.1	**1777**	**67.5**	36.4	**65.8**	**98.8%**
Retain 64 Tokens								
VisionZip	56.2	**74.4**	76.0	1676	64.9	**36.4**	60.4	94.0%
FlowCut	56.5	73.4	79.1	1730	64.6	35.2	60.9	94.4%
Ours	**57.6**	73.9	**81.2**	**1751**	**66.7**	36.1	**63.5**	**96.6%**

B.1 Evaluation with More Models

In addition to conducting experiments on LLaVA series models to validate the effectiveness of SCOUT, we also performed experiments using the Qwen3-VL-8B Bai et al. (2025) model. The Qwen3-VL-8B demonstrates strong capabilities across a variety of tasks and differs from LLaVA in terms of model architecture. As shown in Table 11, our method consistently outperforms FastV under different token budgets while maintaining strong performance. Furthermore, the experimental results indicate that the advantages of SCOUT become increasingly evident under more aggressive pruning settings. Specifically, as the pruning ratio increases and the number of retained visual tokens decreases, the performance improvement achieved by SCOUT becomes larger. For instance, when 88.9% of the tokens are removed, SCOUT surpasses the previous state-of-the-art approach by 5.4%. Therefore, these experiments further validate the generalization ability of SCOUT.

B.2 Evaluation on Larger Models

In addition to evaluating SCOUT across different model families, we further assess its effectiveness on models of different scales. Specifically, we conduct experiments on LLaVA-v1.5-13B Liu et al. (2024a) by retaining different numbers of visual tokens and compare our method with existing approaches. As shown in Table 12, SCOUT consistently outperforms the previous state-of-the-art method when retaining 64, 128, and 192 visual tokens, demonstrating its strong generalization ability across models with different parameter scales. In particular, when only 64 visual tokens are retained, SCOUT achieves a 2.2% improvement over the previous best-performing method.

B.3 Retain Less Visual Tokens

To further evaluate the effectiveness of SCOUT under a more constrained token budget, we conducted experiments retaining only 32 tokens. As shown in Table 13, our method significantly outperforms other approaches under such an extreme token budget, achieving an average performance of 91.6%. These findings confirm the strong performance of SCOUT even under highly compressed conditions.

Table 13: Performance comparison of different pruning methods on LLaVA-v1.5-7B with a 32-token budget.

Method	SQA	GQA	POPE	VQA^{V2}	VQA^{Text}	Avg.
Upper Bound, 576 Tokens (100%)						
Vanilla	69.5	61.9	85.9	78.5	58.2	100%
Retain 32 Tokens (↓94.4%)						
FastV	68.3 98.3%	46.2 74.6%	35.7 41.6%	56.0 71.3%	51.5 88.5%	74.9%
VisionZip	68.2 98.1%	51.5 83.2%	68.3 79.5%	65.1 82.9%	51.7 88.8%	86.5%
FlowCut	**68.7** **98.8%**	52.1 84.2%	69.6 81.0%	66.9 85.2%	**53.0** **91.1%**	88.1%
Ours	**68.7** **98.8%**	**55.3** **89.3%**	**77.4** **90.1%**	**70.1** **89.3%**	52.8 90.7%	**91.6%**

Table 14: Analysis of Stage 1 semantic saliency score and Stage 2 entropy-guided gating.

Method	Setting	GQA	POPE	MME	Avg.
Upper Bound, 576 Tokens (100%)					
Vanilla		61.9	85.9	1818	100%
Retain 64 Tokens (↓88.9%)					
SCOUT		**57.7** **93.2%**	82.8 96.4%	**1764** **94.7%**	**94.77%**
Stage 1	w/o value norm	57.5 92.9%	82.6 96.2%	1761 94.6%	94.57%
	w/o attention	57.4 92.7%	**83.0** **96.6%**	1732 93.0%	94.10%
Stage 2	w/o entropy	57.2 92.4%	**83.0** **96.6%**	1729 92.9%	93.97%
	w/o attention	57.2 92.4%	82.7 96.3%	1757 94.4%	94.37%

B.4 More Ablations

B.4.1 Analysis of Token Selection Strategies

To further validate the effectiveness of the proposed token selection strategies in SCOUT, we conduct additional ablation studies on both stages. Specifically, we analyze whether the semantic saliency score in Stage 1 effectively captures informative visual regions and whether the entropy-guided gating mechanism in Stage 2 improves cross-modal token selection.

Stage 1 aims to preserve semantically informative regions by jointly considering CLS attention and value representations. We compare three different scoring strategies: attention-only, value norm-only, and the proposed attention-value joint score. As shown in Table 14, relying solely on attention or value norm leads to inferior performance, while jointly considering both complementary signals achieves the best results. This demonstrates that attention captures the visual importance perceived by the vision encoder, whereas value norm provides additional information about the intrinsic representation quality of visual tokens. Their combination enables more reliable selection of informative regions.

For Stage 2, we compare the proposed entropy-guided gating strategy with alternative token selection criteria, including attention-based selection and entropy-only selection. As shown in Table 14, entropy-guided gating consistently outperforms these alternatives. Although attention scores pro-

Table 15: Effect of the threshold α in Stage 1.

α	GQA	SQA	POPE	MME	VQA^{T}	MMMU	Avg.
Upper Bound, 576 Tokens (100%)							
Vanilla	61.9	69.5	85.9	1862	58.2	36.3	100%
Retain 64 Tokens (↓88.9%)							
α=0.6	57.3	**69.0**	82.5	1720	55.9	36.3	96.20%
α=0.7	57.3	**69.0**	**83.0**	**1737**	56.1	36.3	**96.62%**
α=0.8	**57.4**	68.5	82.7	1724	56.2	**36.6**	96.38%
α=0.9	57.3	68.5	82.9	1720	**56.3**	36.0	96.20%

Table 16: Effect of the hyperparameters ρ in Stage 2.

ρ	GQA	SQA	POPE	MME	MMB	VQA^{T}	MMMU	SEED	Avg.
Upper Bound, 576 Tokens (100%)									
Vanilla	61.9	69.5	85.9	1862	64.7	58.2	36.3	58.6	100%
Retain 64 Tokens (↓88.9%)									
ρ=0.7	57.5	68.9	82.8	1754	61.3	56.1	36.6	56.2	96.30%
ρ=0.5	57.7	68.8	82.8	1764	61.4	56.1	36.3	56.2	**96.31%**
ρ=0.3	57.7	68.9	82.8	1760	61.4	56.1	36.1	56.2	96.23%

vide useful cross-modal relevance cues, they may be affected by attention concentration and attention sink effects. By calibrating attention with entropy-based uncertainty estimation, our method better identifies semantically reliable tokens and achieves more effective fine-grained pruning.

B.4.2 Sensitivity Analysis of the Quantile Parameter α in Stage 1.

In this section, we mainly investigate the impact of the hyperparameter α on the pruning performance. The quantile parameter α determines the binary mask generation process from the semantic saliency scores, which subsequently affects the semantic regions obtained by Connected Component Labeling (CCL). These semantic regions further influence the visual token pruning results in Stage 1. The pruning results with different α values are presented in Table 15. According to the experimental results, setting $\alpha = 0.7$ achieves the best average performance across multiple datasets. Therefore, we adopt $\alpha = 0.7$ as the default setting in our experiments.

For the LLaVA-NeXT-7B and LLaVA-v1.5-7B models with only 32 retained tokens, we observe that $\alpha = 0.8$ achieves better performance than $\alpha = 0.7$. This may be attributed to the fact that, under such an aggressive token compression setting, a smaller α value initially preserves more tokens, which may introduce redundant regions and negatively affect the subsequent token clustering and sampling process.

B.4.3 Sensitivity Analysis of hyperparameters ρ in Stage 2.

In this section, we investigate the impact of the hyperparameter ρ on the entropy-guided gating mechanism. Specifically, ρ determines the adaptive threshold τ by controlling the position of the entropy distribution quantile, thereby affecting the selection of low-entropy, text-relevant visual tokens. A smaller ρ results in a lower threshold, making the gate more conservative and potentially suppressing informative tokens, while a larger ρ increases the threshold and may preserve excessive high-entropy tokens, reducing the effectiveness of entropy-based filtering. The performance comparison with different ρ values is presented in Table 16. According to the experimental results, the performance is relatively insensitive to different ρ values, indicating that the entropy-guided gating mechanism is robust to the choice of quantile threshold. Among all evaluated settings, $\rho = 0.5$ achieves slightly better overall performance across multiple datasets. Therefore, we adopt $\rho = 0.5$ as the default setting in our experiments.

Table 17: Effect of the hyperparameters λ in Stage 2.

λ	GQA	SQA	POPE	MME	MMB	VQA^T	MMMU	SEED	Avg.
Upper Bound, 576 Tokens (100%)									
Vanilla	61.9	69.5	85.9	1862	64.7	58.2	36.3	58.6	100%
Retain 64 Tokens (↓88.9%)									
λ=1/2	57.6	68.7	82.8	1744	61.4	56.1	36.2	56.2	96.11%
λ=1/4	57.7	68.8	82.8	1764	61.4	56.1	36.3	56.2	**96.31%**
λ=1/6	57.7	68.8	82.8	1765	61.4	56.1	36.2	56.2	96.29%

Table 18: Performance comparison under different S_1:S_2 ratios.

S_1:S_2	S_1	S_2	SQA	GQA	POPE	MME	VQA^T	MMMU	Avg.
Upper Bound, 576 Tokens (100%)									
Vanilla			69.5	61.9	85.9	1862	58.2	36.3	100%
Retain 64 Tokens (↓88.9%)									
1:1	56_8	64	**69.4** **99.9%**	55.6 89.8%	76.7 89.3%	1592 85.5%	53.7 92.3%	35.3 97.2%	92.33%
2:1	78_{12}	44	69.3 99.7%	56.9 91.9%	81.1 94.4%	1697 91.1%	55.5 95.4%	**37.3** **102.8%**	95.88%
3:1	88_{14}	34	68.9 99.1%	57.5 92.9%	82.2 95.7%	1747 93.8%	55.9 96.0%	36.9 101.7%	96.53%
4:1	98_{14}	28	68.8 99.0%	**57.7** 93.2%	82.8 96.4%	**1764** 94.7%	**56.1** 96.4%	36.3 100%	**96.62%**
5:1	100_{16}	24	69.0 99.3%	57.5 92.9%	83.1 96.7%	1750 94.0%	56.0 96.2%	36.2 99.7%	96.47%
6:1	104_{16}	20	68.7 98.8%	57.5 92.9%	**83.3** **97.0%**	1756 94.3%	**56.1** **96.4%**	36.3 100%	96.57%

B.4.4 Sensitivity Analysis of the Temperature Scaling Factor λ in Stage 2.

In this section, we investigate the impact of the temperature scaling factor λ on the entropy-guided gating mechanism. Specifically, λ controls the sample-specific temperature T, which determines the smoothness of the gating function. A smaller λ produces a sharper gate with stronger discrimination between low- and high-entropy tokens, while an excessively small value may lead to over-suppression of informative visual tokens. In contrast, a larger λ results in a smoother gate, which preserves more tokens but may weaken the filtering capability for noisy visual representations. The performance comparison with different λ values is shown in Table 17. The results demonstrate that the proposed entropy-guided gating mechanism is relatively robust to different λ settings. Among the evaluated configurations, setting $\lambda = 1/4$ achieves slightly better overall performance across multiple datasets, providing a favorable balance between token preservation and noise suppression. Therefore, we adopt $\lambda = 1/4$ as the default setting in all experiments.

B.4.5 Proportion of Tokens Retained Across Two Stages.

For SCOUT, a two-stage token pruning algorithm, we conducted experiments by varying the ratio of tokens retained in Stage 1 and Stage 2. The results are presented in Table 18. When the ratio of tokens retained in Stage 1 to Stage 2 was set to 4:1, the best performance was achieved. Accordingly, a default ratio of 4:1 is used in all experiments.

B.4.6 Token Pruning Layers.

Considering that the interaction between textual and visual information during the decoding process of a language model is crucial for evaluating the importance of visual tokens, we conducted

Table 19: Performance comparison under different pruning layers in Stage 2.

Layer	S_1	S_2	SQA	GQA	POPE	MME	VQA^{Text}	MMMU	Avg.
Upper Bound, 576 Tokens (100%)									
Vanilla			69.5	61.9	85.9	1862	58.2	36.3	100%
Retain 64 Tokens (↓88.9%)									
6	142_{22}	40	65.0 93.5%	57.2 92.4%	83.6 97.3%	1667 89.5%	53.9 92.6%	34.4 94.8%	93.35%
12	104_{16}	30	68.4 98.4%	57.2 92.4%	83.6 97.3%	1746 93.8%	56.4 96.9%	36.2 99.7%	96.42%
13	100_{15}	29	68.4 98.4%	57.4 92.7%	**83.2** **96.9%**	1731 93.0%	55.8 95.9%	35.0 96.4%	95.55%
14	98_{14}	28	68.8 99.0%	**57.7** **93.2%**	82.8 96.4%	**1764** 94.7%	**56.1** 96.4%	36.3 100%	**96.62%**
15	92_{14}	26	68.9 99.1%	57.3 92.6%	82.4 95.9%	1747 93.8%	55.7 95.7%	35.1 96.7%	95.63%
16	90_{14}	26	68.4 98.4%	57.4 92.7%	82.5 96.0%	1731 93.0%	**56.1** **96.4%**	**37.3** **102.8%**	96.55%
20	76_{12}	22	**69.4** **99.9%**	56.9 91.9%	80.8 94.1%	1662 89.3%	55.5 95.4%	37.1 102.2%	95.47%

experiments to compare the effects of pruning at different layers in stage 2. As shown in Table 19, the LLaVA-v1.5-7B model consists of 32 LLM layers, and pruning at the 14th layer achieves the best performance. Therefore, we set the 14th layer as the default pruning layer in all subsequent experiments. For other models with different numbers of LLM layers, we select the pruning layer according to the same relative depth ratio as that of LLaVA-v1.5-7B.

The suboptimal performance observed when pruning at shallower layers may be attributed to insufficient interaction between textual and visual information at these stages, resulting in weaker guidance from the textual modality. Conversely, pruning at deeper layers may result in degraded performance due to the attention sink phenomenon.

B.5 VISUALIZATIONS

To more intuitively illustrate the effect of SCOUT during the token pruning process, we conducted a visualization using the LLaVA-v1.5-7B model on the POPE dataset, with a token budget of 64. For Auxiliary Tokens, which encode aggregated information from multiple visual tokens, we do not provide visualization results because they lack a one-to-one correspondence with any individual token in the original image. As shown in Figure 7, the results demonstrate the two-stage token pruning algorithm of SCOUT, where Stage 1 preserves as many semantic units as possible and Stage 2 performs pruning with textual interaction information to mitigate the attention sink problem. The results indicate that SCOUT preserves more semantic tokens relevant to the question, leading to superior performance on specific task.

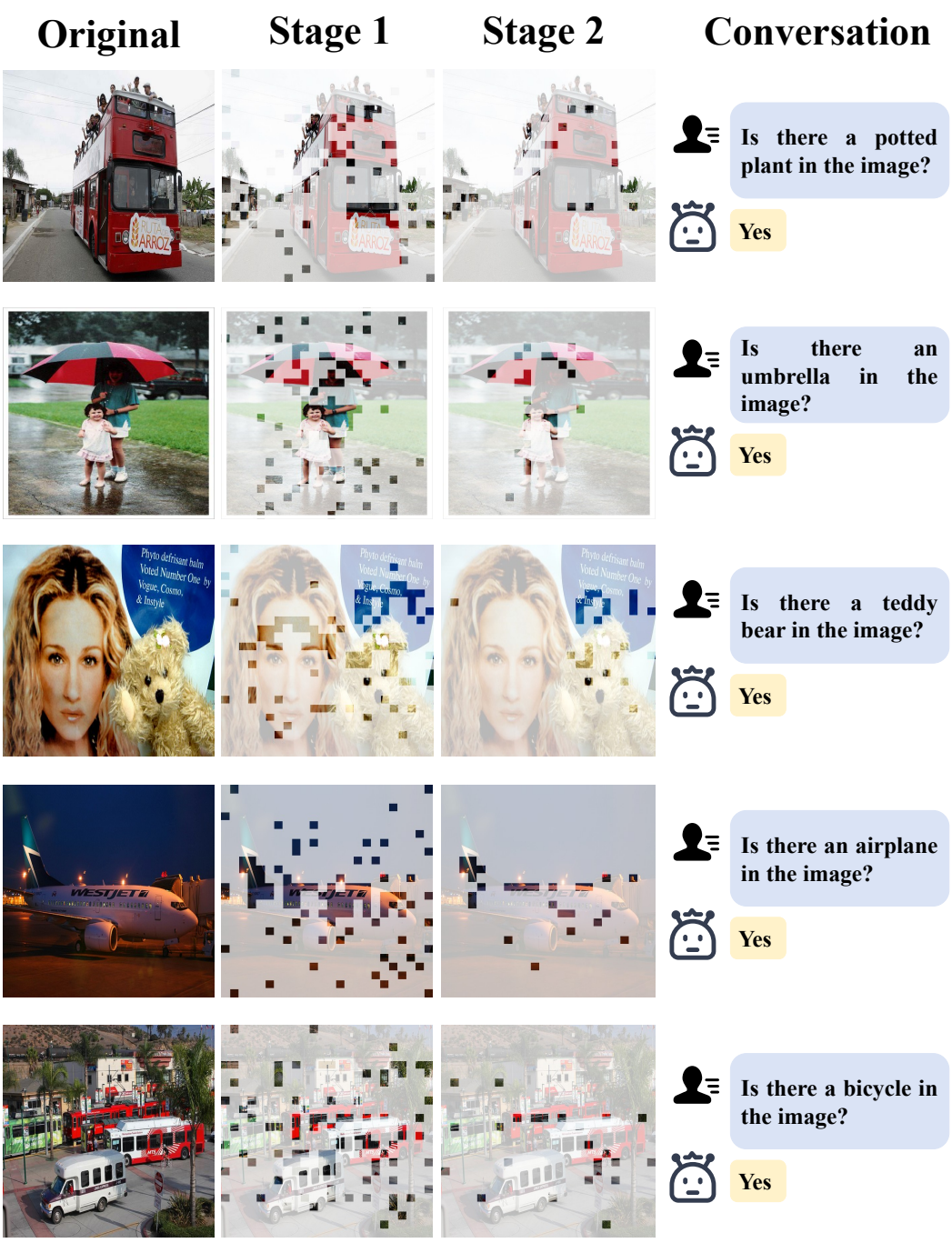

Figure 7: Additional token pruning examples of SCOUT on LLaVA-v1.5-7B.