SQ-PRUNER: SPECTRAL SALIENCY AND QUERY-GUIDED VISUAL TOKEN PRUNING IN LARGE VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Redundant visual tokens impose substantial computational overhead on Large Vision-Language Models (LVLMs). Prior studies typically estimate token importance from visual representations after patchification or spatial aggregation, potentially weakening fine-grained structural information. Moreover, existing query-guided pruning methods often rely on a single relevance signal, which may not fully capture the correspondence between visual tokens and the input query. To address these issues, we propose **Spectral and Query-Guided Pruner (SQ-Pruner)**, a training-free visual token pruning framework that adopts a two-stage pipeline: 1) **Spectral Saliency Anchoring (SSA)** estimates pixel-level spectral saliency before visual encoding to preserve structurally informative tokens as anchors. 2) **Query-Guided Relevance Selection (QRS)** adaptively fuses query-related signals to select visual tokens from the remaining candidates. Extensive experiments demonstrate the effectiveness of our framework across diverse LVLMs and tasks in both image and video modalities. On Qwen2.5-VL, SQ-Pruner retains 81.6% of the original performance while pruning 95.1% of visual tokens, outperforming the strongest prior method by 4.3%. SQ-Pruner also achieves a 1.42× prefilling speedup over the original model, demonstrating a favorable accuracy-efficiency trade-off for LVLM inference.

1 INTRODUCTION

Large Vision-Language Models (LVLMs) (Bai et al., 2025; Chen et al., 2025; Li et al., 2024; Lin et al., 2024; Liu et al., 2024b;a; Chen et al., 2024b) integrate visual encoders with Large Language Models (LLMs), enabling powerful reasoning capabilities for visual understanding. However, these advances come with an increasing number of visual tokens (Yang et al., 2025b; Li et al., 2025), imposing substantial computational and memory overhead during inference, particularly for high-resolution images and long video sequences.

Since many visual tokens are redundant for generating responses (Xu et al., 2026; Chen et al., 2024a), training-free visual token pruning has emerged as an effective approach for improving LVLM inference. However, existing approaches still suffer from several limitations. First, prior studies often estimate token importance from representations obtained after patchification or spatial aggregation (Chen et al., 2024a; Bolya et al., 2022; Alvar et al., 2025; Zhang et al., 2025). Fine-grained visual information from the original image may not be effectively utilized during this process. Second, query relevance is typically estimated from a single query-guided signal, such as attention or similarity, which may not fully capture the relevance between visual tokens and the input query (Wen et al., 2025b; Zhang et al., 2024b). As illustrated in Fig. 1, relying on a single signal may overlook query-relevant visual information, leading to the removal of important tokens and the retention of tokens that are less relevant to the current query (Wen et al., 2025a;b).

To address these limitations, we propose Spectral and Query-Guided Pruner (SQ-Pruner), a training-free visual token pruning framework that adopts a two-stage pipeline consisting of Spectral Saliency Anchoring (SSA) and Query-Guided Relevance Selection (QRS). First, SSA estimates pixel-level spectral saliency directly from the input image before visual encoding to preserve structurally informative tokens as anchors. Second, QRS adaptively fuses query-related signals to retain visual

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.

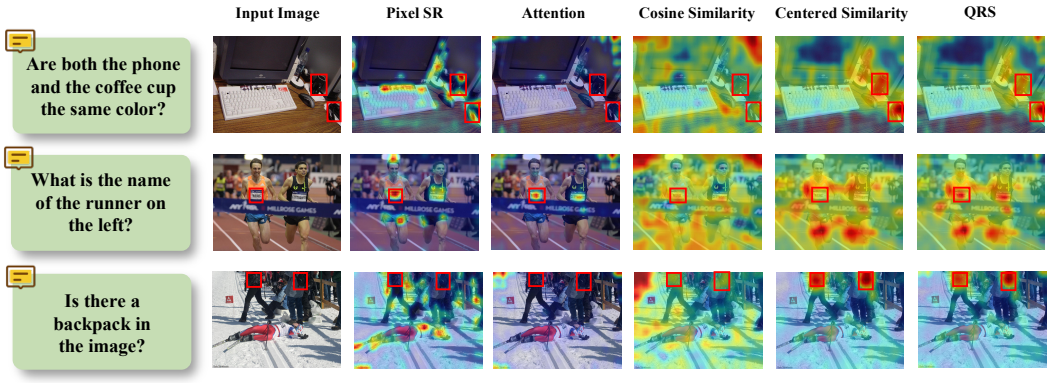

Figure 1: Visualization of scoring maps from Pixel SR, Attention, Cosine Similarity, Centered Similarity and QRS. Red boxes indicate query-relevant regions.

tokens from the remaining candidates. By combining structure-aware anchoring with multi-signal query relevance, SQ-Pruner effectively removes redundant visual tokens while preserving query-relevant information. In summary, the contributions of our work are three-fold:

- We identify the limitations of prior training-free visual token pruning methods: insufficient exploitation of fine-grained visual information in importance estimation and incomplete characterization of query relevance.
- We propose SQ-Pruner, a training-free two-stage visual token pruning framework. SQ-Pruner combines SSA to preserve structurally informative tokens and QRS to adaptively fuse complementary query-related signals.
- We validate SQ-Pruner across various LVLMs and tasks in both image and video modalities. Extensive experiments show that our method consistently outperforms state-of-the-art approaches, achieving a better accuracy-efficiency trade-off.

2 Related Work

2.1 Visual Token Pruning in LVLMs

Training-free visual token pruning has become an effective approach for reducing the overhead of LVLM inference. Existing methods estimate token importance from visual or multimodal representations. Attention-based methods exploit attention patterns to estimate visual token importance. FastV (Chen et al., 2024a) selects visual tokens based on attention scores from shallow layers of the LLM, while HiRED (Arif et al., 2025), FasterVLM (Zhang et al., 2024a) and HiPrune (Liu et al., 2026) further leverage attention signals from the visual encoder or different layers to identify informative visual tokens. However, these methods mainly rely on attention, which may exhibit positional bias and provide an incomplete characterization of token importance (Wen et al., 2025a;b). Methods based on diversity or redundancy seek to preserve diverse information. DeCo (Yao et al., 2024), LLaVA-PruMerge (Shang et al., 2025), DivPrune (Alvar et al., 2025) and DART (Wen et al., 2025b) explore pooling, diversity-aware sampling and pivot selection. While improving visual coverage, these methods may favor visually diverse tokens that are less relevant to the query. Other query-guided approaches exploit feature similarity or query-visual signals to estimate relevance. CDPruner (Zhang et al., 2026) incorporates query relevance as a prior in diversity-aware token selection, while AnchorPrune (Oh & Kim, 2026) identifies query-relevant anchors and supplements them with non-redundant visual context. Existing pruning methods typically estimate token importance from visual token representations and rely on individual signals for query-visual relevance.

2.2 Frequency-Domain Analysis

Frequency-domain analysis has been explored for efficient visual representation and token pruning, as spectral cues provide a lightweight way to characterize visual structures. Prior works (Lee-Thorp et al., 2022; Jiang et al., 2024; Wang et al., 2022) exploit Fourier or spectral representations for efficient token interaction, feature decomposition and token pruning. Recent studies further introduce frequency-based methods into LVLMs for visual token pruning and cross-layer analysis (Wang et al., 2025; Chen et al., 2026). Prior studies mainly apply frequency-domain operations to encoded visual representations, such as frequency filtering or feature transformation, rather than directly analyzing the input image before visual encoding.

3 Motivation

3.1 Spectral Saliency Provides Effective Structural Priors

We first investigate whether spectral saliency can provide an effective prior for visual token pruning. Existing methods typically estimate token importance after patchification or spatial aggregation, where fine-grained visual information from the original image may not be fully utilized. To examine this, we compute Spectral Residual (SR) (Hou & Zhang, 2007) directly from the original image and use its pixel-level saliency to rank visual tokens. As shown in Table 1, Pixel SR provides an effective selection prior under aggressive pruning. Across the evaluated benchmarks, Pixel SR retains 78.2% of the original performance with only 64 tokens, outperforming FastV (Chen et al., 2024a) (70.8%) and DART (Wen et al., 2025b) (71.2%). On the fine-grained ChartQA (Masry et al., 2022), Pixel SR achieves 49.6, substantially higher than FastV (22.8) and DART (27.7). We further compare with Token SR, which is computed after visual tokenization. Under the same token budget, Pixel SR achieves higher performance than Token SR, retaining 93.2% of the original performance at 256 tokens, compared with 81.4% for Token SR. At 128 tokens, Pixel SR retains 86.3%, compared with 72.4% for Token SR. This performance gap indicates that computing spectral saliency at the pixel level can provide more effective guidance for visual token selection. As illustrated in Fig. 1, Pixel SR captures major contours and fine-grained structural regions, providing an informative spatial prior for token importance estimation. Detailed implementation and comparisons are provided in Appendix A.2.3 and A.5.1.

Table 1: Comparison of SR with visual token pruning methods on Qwen2.5-VL-7B.

Method	GQA	POPE	SQA	VQA^{Text}	AI2D	ChartQA	Avg.
Upper Bound, 1,296 Tokens (**100%**)							
Vanilla	59.8	85.9	87.8	82.5	83.0	80.5	100.0%
Retain 256 Tokens (↓ **80.2%**)							
$FastV_{ECCV24}$	54.7	82.6	84.0	78.5	78.6	58.7	91.0%
$DART_{EMNLP25}$	57.3	82.8	84.5	68.6	73.6	55.9	88.3%
Token SR	58.5	83.9	84.3	38.4	74.1	49.1	81.4%
Pixel SR	55.7	82.3	85.9	70.9	80.4	72.0	**93.2%**
Retain 128 Tokens (↓ **90.1%**)							
$FastV_{ECCV24}$	49.9	76.5	79.9	71.2	73.2	35.6	80.4%
$DART_{EMNLP25}$	54.4	78.7	84.4	55.7	69.1	39.4	79.7%
Token SR	56.2	81.0	81.8	25.0	68.0	32.6	72.4%
Pixel SR	52.7	77.4	82.3	60.8	77.2	63.7	**86.3%**
Retain 64 Tokens (↓ **95.1%**)							
$FastV_{ECCV24}$	45.4	62.7	79.2	60.2	69.8	22.8	70.8%
$DART_{EMNLP25}$	50.5	70.7	81.6	43.8	66.6	27.7	71.2%
Token SR	53.0	77.2	77.6	18.8	65.4	21.6	65.9%
Pixel SR	49.6	73.0	80.4	49.7	72.9	49.6	**78.2%**

3.2 Single Query-Relevance Signal is Insufficient

Although spectral saliency provides structurally informative anchors, it is inherently query-agnostic. A visually prominent object may be irrelevant to the input query, while a less salient local region may contain the critical information needed to answer it. Therefore, an effective pruning strategy should further identify query-relevant tokens from the remaining candidates. Existing pruning methods typically estimate query-visual relevance using a single signal, such as attention or feature similarity. However, each signal has inherent limitations. Attention can be diffuse and affected by attention sinks or other attention biases (Xiao et al., 2024; Wen et al., 2025b), while cosine similarity can be less discriminative due to anisotropic embedding distributions (Mu et al., 2017). Therefore, a single signal may not reliably characterize query-visual relevance. As illustrated in Fig. 1, attention and cosine similarity may respond to irrelevant background or visually salient but query-irrelevant content. Moreover, the highlighted regions of signals are not fully consistent, motivating the fusion of multiple query-related signals for more reliable relevance estimation.

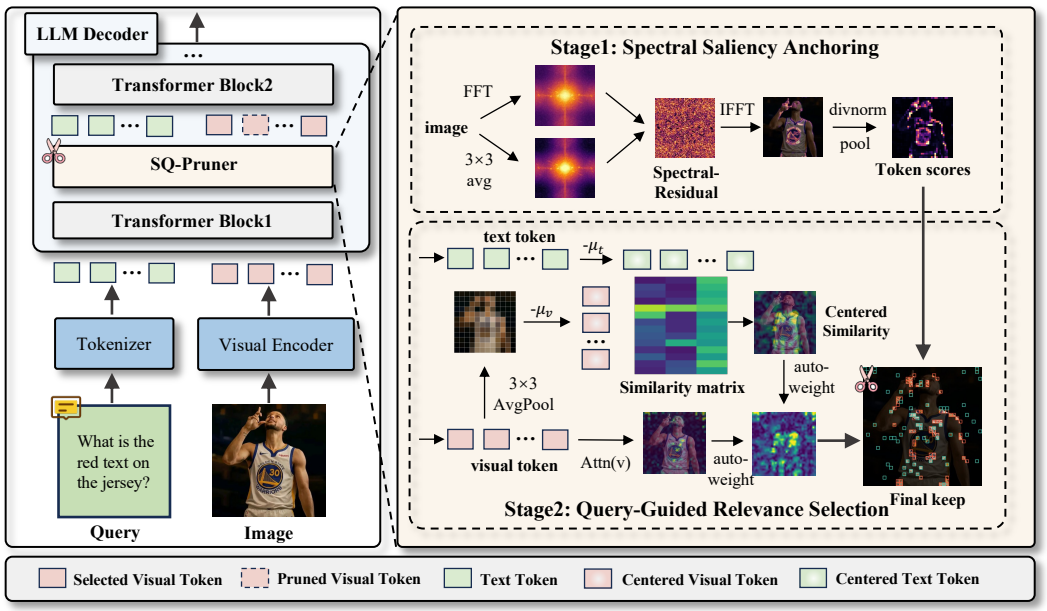

Figure 2: **Overview of SQ-Pruner**. Stage 1: SSA estimates pixel-level spectral saliency before visual encoding to preserve structurally informative tokens as anchors. Stage 2: QRS then adaptively fuses query-related signals to select query-relevant tokens from the remaining candidates.

4 METHOD

4.1 PRELIMINARY

Large Vision-Language Models. LVLMs are designed to jointly process visual and query inputs. Visual features are extracted by a vision encoder and projected into the hidden space of the language model, yielding $\mathbf{X}_V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N] \in \mathbb{R}^{N \times D}$, where N denotes the number of visual tokens and D denotes the hidden dimension. Given a query containing L_q tokens, the embedding layer maps them to a matrix $\mathbf{X}_T = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{L_q}] \in \mathbb{R}^{L_q \times D}$. Typically, $N \gg L_q$. The visual and query tokens are concatenated as $\mathbf{X} = [\mathbf{X}_V; \mathbf{X}_T] \in \mathbb{R}^{(L_q+N) \times D}$ and processed by the language model for multimodal understanding and autoregressive generation.

Visual Token Pruning in LVLMs. Visual tokens exhibit substantial redundancy (Xu et al., 2026; Chen et al., 2024a), making token pruning an effective way to reduce the visual sequence length. Given a target budget $K \ll N$, we select a subset of K visual tokens from the original N tokens:

$$\widetilde{\mathbf{X}}_V = \mathbf{X}_V[\mathcal{S}], \qquad \widetilde{\mathbf{X}}_V \in \mathbb{R}^{K \times D}, \tag{1}$$

where $\mathcal{S} \subseteq \{1, \ldots, N\}$ denotes the selected token indices, with $|\mathcal{S}| = K$.

4.2 OVERVIEW OF SQ-PRUNER

Motivated by the above observations, we propose a training-free visual token pruning framework named SQ-Pruner, as illustrated in Fig. 2. At the pruning layer ℓ, SQ-Pruner reduces the original N visual tokens to a target budget of K through two stages. SSA estimates pixel-level spectral saliency before visual encoding to preserve structurally informative tokens as anchors, while QRS adaptively fuses query-related signals to select query-relevant tokens.

4.3 Stage 1: Spectral Saliency Anchoring

Pixel-Level Spectral Residual. Existing pruning methods typically estimate saliency after patchification and spatial aggregation, where fine-grained visual information from the original image may not be fully utilized. SSA instead performs saliency estimation before visual encoding.

Given an input image $\mathbf{I}$, we convert it to grayscale and resample it to the ViT input resolution, obtaining $\mathbf{G}_{\text{work}} \in \mathbb{R}^{(H_0P)\times(W_0P)}$, with P as the patch size and (H_0, W_0) as the patch grid. We compute the 2D Fourier transform $\mathcal{F} = \text{FFT2}(\mathbf{G}_{\text{work}})$, obtain the logarithmic amplitude spectrum $\mathbf{A}_{\text{amp}} = \log\big(|\mathcal{F}| + \epsilon\big)$ and phase spectrum $\boldsymbol{\phi} = \angle\mathcal{F}$, where $\epsilon = 10^{-8}$ is a small constant used for numerical stability. Following spectral residual saliency (Hou & Zhang, 2007), we remove locally predictable spectral components by computing $\mathcal{R} = \mathbf{A}_{\text{amp}} - h_3(\mathbf{A}_{\text{amp}})$, with $h_3(\cdot)$ as a 3×3 mean filter. We then combine the residual amplitude with the original phase and transform the resulting spectrum back to the spatial domain to obtain the dense saliency map, where i denotes the imaginary unit:

$$\mathbf{S}_{\text{dense}} = \left|\text{IFFT2}\left(\exp(\mathcal{R} + \mathrm{i}\boldsymbol{\phi})\right)\right|^2 \in \mathbb{R}^{(H_0P)\times(W_0P)}. \tag{2}$$

Local Divisive Normalization. Ranking spectral residual responses may favor spatially extensive repetitive textures over localized salient structures. We therefore apply local divisive normalization to enhance locally salient responses and suppress spatially distributed high-response patterns:

$$\mathbf{S}'_{\text{dense}}[p] = \frac{\mathbf{S}_{\text{dense}}[p]}{h_{k_s}\big(\mathbf{S}_{\text{dense}}\big)[p] + \alpha_d\overline{\mathbf{S}_{\text{dense}}} + \epsilon}, \tag{3}$$

where p denotes a pixel location, $h_{k_s}(\cdot)$ is a local mean operator with window size k_s, $\overline{(\cdot)}$ represents the global spatial mean and $\alpha_d = 0.5$ controls the contribution of the global mean. The spatial window size is adaptively determined as $k_s = \text{oddceil}\big(\frac{1}{4}\min(H_0P, W_0P)\big)$, where $\text{oddceil}(\cdot)$ rounds its input upward to the nearest odd integer.

Token Alignment. Let m denote the spatial merging ratio, where each $m \times m$ patch group forms one visual token. For Qwen2.5-VL, we use the default merging ratio $m = 2$, yielding a visual token grid of $(H_g, W_g) = (H_0/m, W_0/m)$, with $N = H_gW_g$. For token i, we average the normalized saliency responses within its corresponding $(mP) \times (mP)$ pixel region Ω_i, i.e., $s_{\text{ssa}}[i] = \frac{1}{|\Omega_i|}\sum_{p\in\Omega_i}\mathbf{S}'_{\text{dense}}[p]$, where $|\Omega_i| = (mP)^2$. These scores are flattened according to the model scanning order to obtain $\mathbf{s}_{\text{ssa}} \in \mathbb{R}^N$, serving as the structural prior for subsequent query-guided selection.

4.4 Stage 2: Query-Guided Relevance Selection

Centered Similarity. Let $\mathbf{V} \in \mathbb{R}^{N\times D}$ and $\mathbf{T} \in \mathbb{R}^{L_q\times D}$ denote the visual and query features at the pruning layer ℓ. We first apply local mean pooling to the visual features $\mathbf{V}$ over a 3×3 local neighborhood. Using the grid metadata provided by the vision encoder, we reshape each spatially contiguous block into its native 2D grid, perform pooling independently within each block and then flatten the pooled features in the original scanning order. The pooled feature is given by $\widetilde{\mathbf{V}}[i] = \frac{1}{|\mathcal{N}_3(i)|}\sum_{n\in\mathcal{N}_3(i)}\mathbf{V}[n]$. To mitigate modality-specific common directions (Mu et al., 2017), we independently center and normalize the two modalities. Specifically, we first compute $\bar{\mathbf{V}} = \frac{1}{N}\sum_{i=1}^{N}\widetilde{\mathbf{V}}[i]$ and $\bar{\mathbf{T}} = \frac{1}{L_q}\sum_{j=1}^{L_q}\mathbf{T}[j]$ and obtain $\mathbf{V}_c = \widetilde{\mathbf{V}} - \bar{\mathbf{V}}$ and $\mathbf{T}_c = \mathbf{T} - \bar{\mathbf{T}}$. We then apply row-wise L_2 normalization to obtain $\mathbf{V}_n$ and $\mathbf{T}_n$. The resulting centered cosine similarity matrix is:

$$\mathbf{C} = \mathbf{V}_n\mathbf{T}_n^\top \in \mathbb{R}^{N\times L_q}. \tag{4}$$

To reduce the influence of irrelevant query tokens, we aggregate the query-token similarities using a temperature-scaled softmax:

$$w_{ij} = \frac{\exp\big(\mathbf{C}[i,j]/\tau\big)}{\sum_{n=1}^{L_q}\exp\big(\mathbf{C}[i,n]/\tau\big)}, \qquad r_{\text{sim}}[i] = \sum_{j=1}^{L_q} w_{ij}\mathbf{C}[i,j], \tag{5}$$

where τ is the softmax temperature, set to 0.05 in all experiments. A smaller τ makes the aggregation approach max pooling over query tokens while retaining a smooth relevance score. The resulting scores form the similarity vector $\mathbf{r}_{\text{sim}} = [r_{\text{sim}}[1], \ldots, r_{\text{sim}}[N]] \in \mathbb{R}^N$.

Query-Guided Attention Score. At the pruning layer, we compute attention using Query/Key projections with RoPE and softmax normalization over all causally visible keys for each query token. Let $A_{j,i}^{(h)}$ denote the attention weight from query position j to visual token i in head h. Averaging across all H heads and L_q query positions yields $r_{\text{attn}}[i] = \frac{1}{HL_q} \sum_{h=1}^{H} \sum_{j=1}^{L_q} A_{j,i}^{(h)}$. The resulting vector $\mathbf{r}_{\text{attn}} \in \mathbb{R}^N$ measures query relevance through the attention assigned to each visual token.

Concentration-Adaptive Fusion. The two signals may exhibit different levels of discriminability across inputs. We use response concentration to adapt their relative contributions. For $\mathbf{r} \in \mathbb{R}^N$, we normalize its positive responses to obtain $\tilde{\mathbf{r}}$, where $\tilde{r}_i = [r_i]_+ / \sum_{n=1}^{N} [r_n]_+$; if the sum of positive responses is below ϵ, we use a uniform distribution. The response concentration is defined as:

$$c(\mathbf{r}) = \exp\big(-\mathcal{H}(\mathbf{r})\big), \qquad \mathcal{H}(\mathbf{r}) = -\sum_{i=1}^{N} \tilde{r}_i \log\big(\max(\tilde{r}_i, \epsilon)\big). \tag{6}$$

Since lower entropy $\mathcal{H}(\mathbf{r})$ indicates a more concentrated response distribution, higher $c(\mathbf{r})$ corresponds to a sharper response pattern. We consider a signal valid when its standard deviation exceeds 10^{-4} and define $\mathcal{U} = \big\{u \in \{\text{sim}, \text{attn}\} : \text{std}(\mathbf{r}_u) > 10^{-4}\big\}$. For each valid signal, we determine the fusion weight using its relative concentration $\lambda_u = c(\mathbf{r}_u)/\bar{c}$, where $\bar{c} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} c(\mathbf{r}_u)$. We combine the valid signals to obtain the final QRS relevance score:

$$\mathbf{r}_{\text{QRS}} = \sum_{u \in \mathcal{U}} \lambda_u z(\mathbf{r}_u), \tag{7}$$

where $z(\cdot)$ denotes standardization to zero mean and unit variance.

4.5 Two-Stage Token Selection

We design a two-stage token selection strategy to preserve structurally salient anchors and allocate the remaining budget to query-relevant tokens, balancing structural saliency and query relevance.

Given a target token budget K, SQ-Pruner adopts a two-stage selection strategy. We introduce a budget allocation coefficient α to control the relative allocation between the two stages:

$$K_1 = \text{round}\left(\frac{\alpha}{1+\alpha} K\right), \qquad K_2 = K - K_1. \tag{8}$$

We first select K_1 structural anchors based on $\mathbf{s}_{\text{ssa}}$ as $\mathcal{P} = \text{TopK}(\mathbf{s}_{\text{ssa}}, K_1)$ and define the remaining candidates as $\mathcal{M} = \{1, \ldots, N\} \setminus \mathcal{P}$. We then select K_2 query-relevant tokens from $\mathcal{M}$ based on their QRS scores $\mathcal{C} = \text{TopK}_{i \in \mathcal{M}}(r_{\text{QRS}}[i], K_2)$. The final retained set is $\mathcal{S} = \mathcal{P} \cup \mathcal{C}$ with $|\mathcal{S}| = K$.

5 Experiments

We evaluate SQ-Pruner on representative LVLMs, including Qwen2.5-VL-7B/32B (Bai et al., 2025), LLaVA-1.5-7B (Liu et al., 2024a), LLaVA-NeXT-7B (Liu et al., 2024b) and Video-LLaVA (Lin et al., 2024). We compare SQ-Pruner with state-of-the-art training-free pruning methods, including FastV (Chen et al., 2024a), DivPrune (Alvar et al., 2025), VisionZip (Yang et al., 2025a), SparseVLM (Zhang et al., 2024b), DART (Wen et al., 2025b), HiPrune (Liu et al., 2026) and AnchorPrune (Oh & Kim, 2026). For all pruning methods, we follow the configurations specified in their respective works. For image understanding, we evaluate on MME (Fu et al., 2026), GQA (Hudson & Manning, 2019), POPE (Li et al., 2023), ScienceQA (Lu et al., 2022), MMBench, MMBench-CN (Liu et al., 2024c), TextVQA (Singh et al., 2019), DocVQA (Mathew et al., 2021), OCRBench (Liu et al., 2024d), AI2D (Kembhavi et al., 2016), MMMU (Yue et al., 2024), ChartQA (Masry et al., 2022), VQAv2 (Goyal et al., 2017) and VizWiz (Gurari et al., 2018). For

Table 2: Performance of SQ-Pruner on Qwen2.5-VL-7B. Vanilla uses all 1,296 visual tokens. The best and second-best Avg. results in each setting are **bolded** and underlined, respectively.

Method	MME	GQA	POPE	SQA	MMB^{EN}	MMB^{CN}	VQA^{Text}	DocVQA	OCRBench	AI2D	ChartQA	MMMU	Avg.
Upper Bound, 1,296 Tokens (**100%**)													
Vanilla	2326.9	59.8	85.9	87.8	83.8	80.2	82.5	94.1	745	83.0	80.5	50.7	100.0%
Retain 256 Tokens (↓ **80.2%**)													
$FastV_{ECCV24}$	2234.9	54.7	82.6	84.0	80.2	75.7	78.5	50.6	498	78.6	58.7	48.6	87.4%
$DART_{EMNLP25}$	2259.5	57.3	82.8	84.5	80.6	76.2	68.6	43.8	505	73.6	55.9	48.9	85.7%
$DivPrune_{CVPR25}$	2227.9	59.4	84.9	85.7	80.6	77.2	74.2	68.3	559	80.0	65.9	46.8	91.0%
$SparseVLM_{ICML25}$	2330.9	55.5	85.6	85.3	81.2	77.1	77.2	55.5	511	79.3	55.0	50.2	88.9%
$VisionZip_{CVPR25}$	2304.2	58.1	84.8	86.5	81.3	78.4	74.0	67.6	581	81.4	71.5	49.1	92.6%
$HiPrune_{ACL26}$	2311.9	58.2	84.9	85.8	81.3	78.4	74.0	67.3	586	81.2	71.9	48.7	92.6%
$AnchorPrune_{ECCV26}$	2201.0	59.6	85.0	83.5	80.9	77.9	76.6	72.9	604	78.7	73.3	48.9	93.0%
Ours	2294.1	58.4	86.3	85.9	80.8	77.4	79.5	75.1	646	81.6	73.6	49.1	**94.7%**
Retain 128 Tokens (↓ **90.1%**)													
$FastV_{ECCV24}$	2104.9	49.9	76.5	79.9	74.8	71.7	71.2	31.8	398	73.2	35.6	46.4	77.5%
$DART_{EMNLP25}$	2135.3	54.4	78.7	84.4	78.4	73.2	55.7	28.9	394	69.1	39.4	46.6	77.5%
$DivPrune_{CVPR25}$	2100.5	57.6	82.6	84.3	77.0	74.0	64.6	47.8	401	75.1	50.7	45.1	82.4%
$SparseVLM_{ICML25}$	2086.1	50.9	78.4	80.7	75.3	72.2	57.7	27.4	327	72.3	26.7	47.8	74.6%
$VisionZip_{CVPR25}$	2166.9	55.1	82.0	82.9	79.0	74.9	63.1	44.6	464	77.7	58.4	48.1	84.2%
$HiPrune_{ACL26}$	2194.0	55.1	81.9	83.1	79.3	75.6	61.8	44.7	465	77.6	59.0	46.9	84.1%
$AnchorPrune_{ECCV26}$	2167.0	58.6	84.2	82.3	79.3	75.9	68.5	50.9	481	77.6	60.4	47.1	86.3%
Ours	2246.4	57.2	85.5	82.7	77.8	74.9	73.5	56.7	555	79.0	63.8	47.8	**88.7%**
Retain 64 Tokens (↓ **95.1%**)													
$FastV_{ECCV24}$	1859.5	45.4	62.7	79.2	66.8	63.7	60.2	20.9	305	69.8	22.8	44.2	67.8%
$DART_{EMNLP25}$	2001.4	50.5	70.7	81.6	73.9	70.9	43.8	20.4	313	66.6	27.7	46.6	70.5%
$DivPrune_{CVPR25}$	1950.9	54.3	79.3	82.1	72.5	69.5	51.1	30.2	243	70.6	34.8	42.8	72.8%
$SparseVLM_{ICML25}$	1870.5	49.6	71.7	78.7	70.6	67.4	30.7	16.0	151	66.6	17.5	43.9	64.0%
$VisionZip_{CVPR25}$	1938.4	49.9	76.6	81.5	74.7	72.9	49.2	27.2	343	72.9	43.2	46.4	74.7%
$HiPrune_{ACL26}$	1899.4	50.2	76.6	81.4	75.5	72.7	49.0	27.3	341	72.7	44.4	46.6	74.8%
$AnchorPrune_{ECCV26}$	2043.6	55.4	80.6	81.0	76.0	73.2	57.3	31.3	356	74.2	40.9	44.7	77.3%
Ours	2165.8	55.3	84.0	81.1	76.5	71.5	64.5	38.9	424	75.8	52.9	45.4	**81.6%**

video understanding, we evaluate Video-LLaVA on TGIF-QA (Jang et al., 2017), MSVD-QA (Xu et al., 2017), MSRVTT-QA (Xu et al., 2017) and ActivityNet-QA (Yu et al., 2019). We uniformly sample 8 frames per video and retain 128 or 64 visual tokens per frame, reporting both accuracy (%) and score (0–5). GLM-4-Flash serves as the evaluator for video question answering. All experiments are conducted on NVIDIA H20 GPUs. Detailed benchmark descriptions, additional experiments and visualizations are provided in Appendix A.

5.1 Image Understanding Tasks

Results on Qwen2.5-VL-7B/32B. To ensure a fair comparison, we conduct controlled experiments under three token budgets. As illustrated in Table 2, SQ-Pruner consistently achieves the best average performance, retaining 94.7%, 88.7% and 81.6% of the full-token performance at 256, 128 and 64 tokens. At 256 tokens, SQ-Pruner surpasses the strongest method AnchorPrune (Oh & Kim, 2026) by 1.7%, with the margin increasing to 2.4% and 4.3% at 128 and 64 tokens. Results on Qwen2.5-VL-32B are provided in Appendix A.3.1, where SQ-Pruner also maintains competitive performance across different token budgets.

Results on LLaVA-1.5-7B. We further evaluate SQ-Pruner on LLaVA-1.5-7B, with results shown in Table 3. Following prior work (Chen et al., 2026; Fang et al., 2026), we conduct experiments with token budgets of 192, 128 and 64. SQ-Pruner achieves the highest average performance across all three budgets, retaining 98.7% and 98.1% of the vanilla performance at 192 and 128 tokens. SQ-Pruner outperforms the strongest pruning methods by 0.4% and 0.8%. Even at 64 tokens, our method maintains 95.6% of the vanilla performance with an 88.9% token reduction.

Results on LLaVA-NeXT-7B. Table 4 presents results on LLaVA-NeXT-7B with dynamic high-resolution inputs. With only 160 tokens, SQ-Pruner retains 90.2% of the vanilla performance, outperforming DivPrune (Alvar et al., 2025), DART (Wen et al., 2025b) and HiPrune (Liu et al., 2026) by 2.0%, 2.7% and 7.4% in average performance retention. Our SQ-Pruner achieves 87.0 on POPE and 75.9 on VQAv2, exceeding DART by 9.8 and 2.7.

Table 3: Performance on LLaVA-1.5-7B.

Method	MME	GQA	POPE	SQA	MMB^{EN}	MMB^{CN}	VQA^{Text}	VQA^{v2}	OCRBench	VizWiz	Avg.
Upper Bound, 576 Tokens (**100%**)											
Vanilla	1862	62.0	85.9	69.5	64.1	55.8	58.1	78.5	314	54.4	100%
Retain 192 Tokens (↓ **66.7%**)											
FastV$_{ECCV24}$	1613	52.8	65.1	67.7	60.7	53.6	52.9	67.1	295	50.9	90.0%
DivPrune$_{CVPR25}$	1753	59.2	87.0	68.7	62.4	53.5	56.5	75.1	296	55.6	97.2%
VisionZip$_{CVPR25}$	1784	59.3	85.3	68.5	63.8	54.0	57.2	76.8	313	54.1	98.1%
SparseVLM$_{ICML25}$	1724	57.6	83.6	69.6	62.4	52.9	56.6	75.6	306	50.6	95.9%
DART$_{EMNLP25}$	1853	60.2	83.7	69.4	63.9	53.8	57.5	76.9	298	55.0	98.3%
HiPrune$_{ACL26}$	1803	59.2	86.1	68.6	63.0	53.1	56.1	76.2	302	54.3	97.4%
AnchorPrune$_{ECCV26}$	1776	60.0	86.6	69.0	63.6	53.3	57.4	77.2	307	54.6	98.2%
Ours	1824	61.1	85.8	69.2	63.2	53.1	58.2	76.2	309	55.3	**98.7%**
Retain 128 Tokens (↓ **77.8%**)											
FastV$_{ECCV24}$	1492	49.7	59.3	60.6	56.1	53.2	51.3	61.3	281	49.9	84.7%
DivPrune$_{CVPR25}$	1753	57.8	85.1	68.2	61.5	51.5	55.3	74.2	285	54.8	95.4%
VisionZip$_{CVPR25}$	1763	57.6	84.4	69.0	61.5	53.4	57.1	75.5	301	54.5	96.7%
SparseVLM$_{ICML25}$	1696	57.4	83.5	69.4	62.3	52.6	56.1	74.9	284	49.5	94.6%
DART$_{EMNLP25}$	1831	58.9	81.2	69.3	63.1	53.4	56.9	76.3	294	55.5	97.2%
HiPrune$_{ACL26}$	1787	57.3	83.0	68.3	62.3	52.8	56.3	74.9	296	54.4	96.2%
AnchorPrune$_{ECCV26}$	1727	58.9	85.8	68.9	62.9	52.5	56.9	76.4	302	56.0	97.3%
Ours	1794	60.9	85.7	68.9	62.7	52.8	57.6	75.5	308	55.5	**98.1%**
Retain 64 Tokens (↓ **88.9%**)											
FastV$_{ECCV24}$	1257	46.0	50.2	51.8	48.2	41.2	47.8	57.2	249	49.1	74.8%
DivPrune$_{CVPR25}$	1644	54.8	82.5	68.0	60.2	49.4	54.5	74.1	270	54.3	92.7%
VisionZip$_{CVPR25}$	1690	56.2	77.2	69.1	60.1	51.2	55.1	72.2	281	55.1	93.4%
SparseVLM$_{ICML25}$	1506	51.4	70.3	69.0	58.1	49.7	52.5	67.4	208	48.9	85.7%
DART$_{EMNLP25}$	1751	57.1	76.2	69.0	62.4	51.4	55.1	74.4	272	55.4	94.2%
HiPrune$_{ACL26}$	1646	53.4	73.4	69.0	59.6	50.3	54.3	69.2	259	54.3	90.6%
AnchorPrune$_{ECCV26}$	1648	56.8	83.1	68.9	60.8	51.9	56.3	73.7	285	57.0	95.0%
Ours	1768	59.5	83.6	68.7	60.5	50.0	56.4	73.7	289	55.7	**95.6%**

Table 4: Performance on LLaVA-NeXT-7B.

Method	MME	GQA	POPE	SQA	MMB^{EN}	MMB^{CN}	VQA^{Text}	VQA^{v2}	OCRBench	VizWiz	Avg.
Upper Bound, 2,880 Tokens (**100%**)											
Vanilla	1852	64.3	86.5	70.4	67.3	57.3	61.4	80.1	521	60.7	100%
Retain 160 Tokens (↓ **94.4%**)											
FastV$_{ECCV24}$	1409	51.3	56.3	63.4	56.5	46.2	48.6	68.5	138	49.8	74.9%
DivPrune$_{CVPR25}$	1703	58.9	81.4	67.9	64.1	54.8	46.1	72.6	296	57.6	88.2%
SparseVLM$_{ICML25}$	1545	53.3	80.7	65.2	60.1	49.3	51.0	68.2	179	50.6	81.3%
DART$_{EMNLP25}$	1675	57.5	77.2	67.4	61.5	51.0	50.0	73.2	324	57.5	87.5%
HiPrune$_{ACL26}$	1568	51.9	67.7	68.8	60.4	51.6	43.3	71.8	270	57.4	82.8%
Ours	1634	60.7	87.0	68.4	62.3	50.8	55.1	75.9	324	56.9	**90.2%**

Table 5: Performance on Video-LLaVA.

Method	TGIF	MSVD	MSRVTT	ActivityNet	Avg.
Upper Bound, 256 Tokens (**100%**)					
Vanilla	54.4 / 2.91	75.5 / 3.58	61.5 / 3.04	50.2 / 2.77	60.4 / 3.08
Retain 128 Tokens (↓ **50%**)					
FastV$_{ECCV24}$	51.4 / 2.78	73.4 / 3.50	60.0 / 2.99	44.7 / 2.54	57.4 / 2.95
DivPrune$_{CVPR25}$	51.3 / 2.79	74.1 / 3.55	59.8 / 2.99	45.6 / 2.60	57.7 / 2.98
DART$_{EMNLP25}$	52.3 / 2.82	73.7 / 3.51	59.9 / 2.98	44.9 / 2.56	57.7 / 2.97
Ours	54.2 / 2.92	75.3 / 3.57	61.3 / 3.03	49.7 / 2.75	**60.1 / 3.07**
Retain 64 Tokens (↓ **75%**)					
FastV$_{ECCV24}$	49.9 / 2.74	72.7 / 3.49	58.9 / 2.95	45.0 / 2.55	56.6 / 2.93
DivPrune$_{CVPR25}$	50.2 / 2.75	73.4 / 3.53	58.8 / 2.96	45.6 / 2.61	57.0 / 2.96
DART$_{EMNLP25}$	52.3 / 2.82	73.4 / 3.52	59.1 / 2.96	45.1 / 2.57	57.5 / 2.97
Ours	53.8 / 2.87	73.3 / 3.48	60.4 / 2.99	48.3 / 2.68	**59.0 / 3.01**

5.2 Video Understanding Tasks

Table 5 demonstrates the effectiveness of our SQ-Pruner on video understanding tasks. With 128 retained tokens, SQ-Pruner achieves 60.1% average accuracy and a 3.07 average score, outperforming DART (Wen et al., 2025b) by 2.4%. On ActivityNet-QA, our method achieves 49.7, close to the 50.2 of the vanilla. Even at 64 tokens, SQ-Pruner retains 59.0% average accuracy and a 3.01 average score, with only a 1.1% drop in average accuracy from 128 tokens.

5.3 Ablation Study

Component-Wise Analysis. Table 6 evaluates the two-stage design and QRS alignment. Combining SSA and QRS achieves the best average performance across the 12 benchmarks, reaching 94.7%, 88.7% and 81.6% at 256, 128 and 64 tokens. Compared with stage 1 alone, adding stage 2 improves performance by 1.5%, 3.1% and 4.1%, with larger gains under tighter budgets. Moreover, QRS achieves the best average performance among the attention-, cosine similarity- and centered similarity-based methods in stage 2, indicating the effectiveness of the proposed alignment strategy.

Hyperparameter Sensitivity Analysis. Table 7 evaluates the stage balancing hyperparameter α in Eq. 8 over $\{0.6, 0.8, 1.0, 1.2, 1.4\}$. $\alpha = 1.0$ achieves the best performance at all three budgets, reaching 93.6%, 86.1% and 77.0% with 256, 128 and 64 tokens. The performance varies only slightly across different values of α, indicating that the method is relatively insensitive to this hyperparameter. We therefore set $\alpha = 1.0$ in all our experiments.

Robustness under Extreme Token Pruning. Fig. 3 evaluates SQ-Pruner across token budgets ranging from 512 to 13. Our method achieves the highest performance on DocVQA and OCRBench across all budgets, while outperforming all pruning methods on POPE and TextVQA with 256 tokens or fewer. Even with only 13 visual tokens, SQ-Pruner retains 76.1 on POPE, 15.2 on DocVQA, 215 on OCRBench and 32.2 on TextVQA, outperforming all pruning methods.

Effectiveness of Adaptive Fusion. To validate the effectiveness of adaptive fusion, we compare it with fixed fusion coefficients using only 64 visual tokens on Qwen2.5-VL-7B. Specifically, we replace the adaptive weights λ_u with a fixed coefficient β, where $\beta = 1.0$ denotes equal weighting. As shown in Table 8, adaptive fusion achieves the highest average performance of 81.6%, outperforming the best fixed coefficient setting by 1.0%. These results demonstrate that input-dependent signal weighting is more effective than using fixed coefficients.

Table 6: Ablation study of the two-stage strategy.

Stage 1	Stage 2	Avg.		
		256	128	64
SSA	–	93.2%	85.6%	77.5%
QRS	–	93.6%	87.8%	80.7%
SSA	Attention	93.5%	87.6%	80.2%
	Cosine	93.2%	86.0%	77.6%
	Centered	94.3%	88.0%	80.3%
	QRS	**94.7%**	**88.7%**	**81.6%**

Table 7: Ablation study of the token allocation ratio α between SSA and QRS on Qwen2.5-VL-7B.

Method	MME (Total)			ScienceQA			OCRBench			ChartQA			Avg.		
	256	128	64	256	128	64	256	128	64	256	128	64	256	128	64
Vanilla	2326.9			87.8			745			80.5			100.0%		
$\alpha = 0.6$	2280.8	2235.8	2140.6	85.7	82.4	80.5	638	550	417	73.0	63.5	52.9	93.0%	85.7%	76.3%
$\alpha = 0.8$	2288.1	2241.6	2141.5	85.8	82.6	80.9	644	548	421	73.4	63.7	52.9	93.4%	85.8%	76.6%
$\boldsymbol{\alpha = 1.0}$	2294.1	2246.4	2165.8	85.9	82.7	81.1	646	555	424	73.6	63.8	52.9	**93.6%**	**86.1%**	**77.0%**
$\alpha = 1.2$	2293.0	2243.6	2157.0	85.8	82.7	81.0	647	553	422	73.2	63.3	52.8	93.5%	85.9%	76.8%
$\alpha = 1.4$	2287.1	2239.4	2154.7	85.4	82.4	80.6	642	549	419	73.3	62.1	52.6	93.2%	85.2%	76.5%

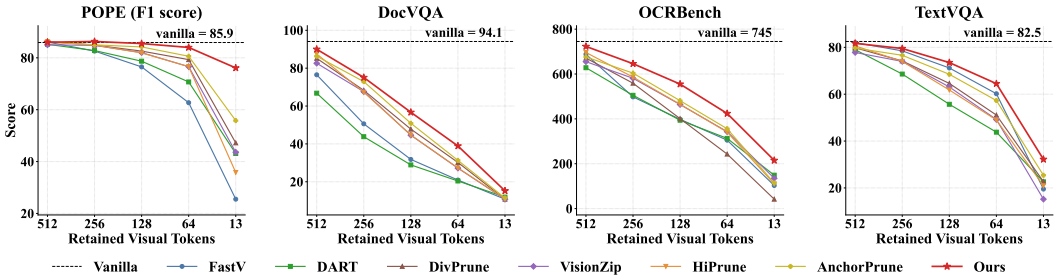

Figure 3: Comparison of visual token pruning methods under extreme token reduction.

Table 8: Ablation study of the fusion coefficient β on Qwen2.5-VL-7B.

Method	MME	GQA	POPE	SQA	MMB^{EN}	MMB^{CN}	VQA^{Text}	DocVQA	OCRBench	AI2D	ChartQA	MMMU	Avg.
Upper Bound, 1,296 Tokens (**100%**)													
Vanilla	2326.9	59.8	85.9	87.8	83.8	80.2	82.5	94.1	745	83.0	80.5	50.7	100.0%
Retain 64 Tokens (↓ **95.1%**)													
$\beta = 0.8$	2118.2	54.4	83.7	81.1	75.9	70.6	61.5	37.1	398	73.9	51.8	45.8	80.1%
$\beta = 1.0$	2148.4	54.6	83.4	81.4	75.9	70.9	62.5	37.6	409	74.4	52.6	45.2	80.6%
$\beta = 1.2$	2144.9	54.3	83.0	81.6	75.4	71.0	63.3	37.3	410	74.5	52.5	45.0	80.5%
Ours	2165.8	55.3	84.0	81.1	76.5	71.5	64.5	38.9	424	75.8	52.9	45.4	**81.6%**

Table 9: Efficiency analysis and comparison. Prefilling and inference times represent the average per-sample latency.

Method	Performance		FLOPs		Prefill	Inference
	POPE	% of Vanilla	(T)	Reduction	(ms)	(ms)
Vanilla	85.9	100.0%	26.87	–	192.56	227.43
FastV$_{ECCV24}$	62.7	73.0%	10.57	60.7%	130.13	162.25
DART$_{EMNLP25}$	70.7	82.3%	9.99	62.8%	130.20	162.92
DivPrune$_{CVPR25}$	79.3	92.3%	8.73	67.5%	127.45	158.36
VisionZip$_{CVPR25}$	76.6	89.2%	8.80	67.2%	138.30	169.66
HiPrune$_{ACL26}$	76.6	89.2%	8.87	67.0%	153.26	183.50
AnchorPrune$_{ECCV26}$	80.6	93.8%	8.80	67.2%	144.13	175.07
Ours	84.0	97.8%	9.93	63.0%	135.63	165.06

5.4 Efficiency Analysis

Table 9 compares the efficiency of SQ-Pruner on Qwen2.5-VL-7B. All experiments use a fixed image resolution of 1008×1008 and retain 64 visual tokens. The reported FLOPs include the full computational cost of SQ-Pruner, including all operations required for token pruning. SQ-Pruner reduces FLOPs by 63.0%, from 26.87T to 9.93T and reduces prefill latency from 192.56 ms to 135.63 ms, yielding a 1.42× speedup. Despite the computational savings, SQ-Pruner achieves 84.0 on POPE, retaining 97.8% of the Vanilla performance. Our method outperforms AnchorPrune (Oh & Kim, 2026), VisionZip (Yang et al., 2025a) and DivPrune (Alvar et al., 2025) by 3.4, 7.4 and 4.7, demonstrating a favorable performance-efficiency trade-off.

6 Conclusion

In this work, we present SQ-Pruner, a training-free visual token pruning framework for efficient LVLM inference. SQ-Pruner adopts a two-stage pipeline that uses pixel-level spectral saliency to preserve structurally informative tokens and query-guided relevance to select query-relevant tokens. Extensive experiments on image and video benchmarks demonstrate that SQ-Pruner consistently achieves state-of-the-art performance across different LVLM and pruning settings. These results highlight the potential of spectral saliency and query-guided token selection for scaling LVLMs to increasingly demanding and resource-constrained settings.

AI USE STATEMENT

In this work, we used large language models (LLMs) to assist with the writing and refinement of the manuscript. Specifically, LLMs were used to improve the clarity, readability and overall coherence of the paper, including tasks such as sentence rewriting, grammar checking and refinement of academic language.

LLMs were not involved in the conception of the research, the development of the research methodology, or the design of the experiments. All research ideas, methodological decisions, experimental designs, analyses and scientific conclusions were independently developed and conducted by the authors. The use of LLMs was limited to improving the linguistic quality and presentation of the manuscript and did not involve the generation or analysis of scientific data.

The authors reviewed all LLM-assisted text and take full responsibility for the content of the manuscript, including any text generated or refined with the assistance of LLMs. We also verified that the final manuscript complies with academic integrity standards and contains no plagiarism or other forms of academic misconduct.

ETHICS STATEMENT

This work does not involve human subjects, private or sensitive data, animal experiments, or the collection of new personal data. We use publicly available datasets and pretrained vision-language models for evaluation. We are not aware of any additional ethical concerns specific to the methodology or experiments presented in this work.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure the reproducibility of this work.

Method. SQ-Pruner is training-free and introduces no learnable parameters. The complete procedure is specified in Section 4, with all equations stated explicitly. Step-by-step pseudocode for spectral saliency anchoring, query-guided relevance selection and two-stage token selection is provided in Appendix A.1.

Hyperparameters. Section 4 specifies the numerical settings used in the scoring formulation. The choice of the allocation coefficient α is discussed in the sensitivity analysis in Table 7. The default pruning layer ℓ is specified in Appendix A.3.4, together with its performance and computational cost under different pruning layers. The ablation of fusion parameter β is also discussed in Appendix A.3.5.

Experimental setup. Appendix A.2 summarizes the evaluated LVLMs, datasets and implementation details. Token budgets are reported in the experimental tables, while Section 5.4 presents the efficiency comparison and specifies its image resolution and retained token budget.

Code. An anonymized implementation of SQ-Pruner will be publicly released upon acceptance.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1773–1781, 2025.

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.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Bin Chen, Yuxiang Cai, Yadan Luo, Yi Zhang, Jianwei Yin, and Zhi Chen. Spectral evolution-guided token pruning in multimodal large language models. *arXiv preprint arXiv:2606.24165*, 2026.

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

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, volume 2025, pp. 18227–18246, 2025.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024b.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in vlms via synergistic importance-diversity. *arXiv preprint arXiv:2603.09480*, 2026.

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. *Advances in Neural Information Processing Systems*, 38, 2026.

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.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617. IEEE, 2018.

Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, pp. 1–8. Ieee, 2007.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1359–1367. IEEE, 2017.

Xingyu Jiang, Xiuhui Zhang, Ning Gao, and Yue Deng. When fast fourier transform meets transformer for image restoration. In *European conference on computer vision*, pp. 381–402. Springer, 2024.

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

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, pp. 4296–4313, 2022.

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*, 2024.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, 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, 2023.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 5971–5984, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, january 2024. *URL https://llava-vl. github. io/blog/2024-01-30-llava-next*, 1(8), 2024b.

Jizhihui Liu, Guangdao Zhu, Feiyi Du, Niu Lian, Jun Li, Bin Chen, Weili Guan, and Yaowei Wang. Hiprune: Hierarchical attention for efficient token pruning in vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3274–3291, 2026.

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

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024d.

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.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.

Kyuan Oh and Bumsoo Kim. Anchorprune: Relevance-anchored contextual expansion for visual token pruning. *arXiv preprint arXiv:2607.07033*, 2026.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867. IEEE, 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8309–8318. IEEE, 2019.

Huanyu Wang, Jushi Kai, Haoli Bai, Lu Hou, Bo Jiang, Ziwei He, and Zhouhan Lin. Fourier compressor: Frequency-domain visual token compression for vision-language models. *arXiv preprint arXiv:2508.06038*, 2025.

Zhenyu Wang, Hao Luo, Pichao Wang, Feng Ding, Fan Wang, and Hao Li. Vtc-lfc: Vision transformer compression with low-frequency components. *Advances in neural information processing systems*, 35:13974–13988, 2022.

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*, pp. 15537–15549, 2025a.

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

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.

Rui Xu, Yunke Wang, Yong Luo, and Bo Du. Rethinking visual token reduction in lvlms under cross-modal misalignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27323–27331, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025a.

Senqiao Yang, Jiaming Liu, Renrui Zhang, Mingjie Pan, Ziyu Guo, Xiaoqi Li, Zehui Chen, Peng Gao, Hongsheng Li, Yandong Guo, et al. Lidar-llm: Exploring the potential of large language models for 3d lidar understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9247–9255, 2025b.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. Deco: Decoupling token compression from semantic abstraction in multimodal large language models. *arXiv preprint arXiv:2405.20985*, 2024.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 9127–9134, 2019.

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.

Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv preprint arXiv:2412.01818*, 2(3):4, 2024a.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20857–20867. IEEE, 2025.

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. *Advances in Neural Information Processing Systems*, 38:25438–25468, 2026.

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

A Appendix

A.1 Pseudocode Algorithm

To clarify the overall procedure, we provide the pseudocode of SQ-Pruner in Alg. 1, Alg. 2 and Alg. 3. Given a pruning layer ℓ, SQ-Pruner reduces the original N visual tokens to a target budget K without additional training.

Alg. 1 implements Stage 1, Spectral Saliency Anchoring (SSA). The input image is first converted to grayscale and resampled to the ViT input resolution. We then compute the spectral residual from the log-amplitude spectrum using FFT and reconstruct the dense saliency map using inverse FFT (Eq. 2), followed by local divisive normalization to enhance locally distinctive structures (Eq. 3). The resulting dense saliency map is aligned with the visual-token grid through region-wise averaging, yielding the structural prior $\mathbf{s}_{\mathrm{ssa}}$.

Algorithm 1 Spectral Saliency Anchoring

Require: Input image $\mathbf{I}$; patch grid size $H_0 \times W_0$ with patch size P;

1: Convert $\mathbf{I}$ to grayscale and resample to ViT input resolution, obtaining $\mathbf{G}_{\mathrm{work}} \in \mathbb{R}^{(H_0P)\times(W_0P)}$
2: $\mathcal{F} \leftarrow \mathrm{FFT2}(\mathbf{G}_{\mathrm{work}})$; $\mathbf{A}_{\mathrm{amp}} \leftarrow \log(|\mathcal{F}| + \epsilon)$; $\boldsymbol{\phi} \leftarrow \angle\mathcal{F}$
3: $\mathcal{R} \leftarrow \mathbf{A}_{\mathrm{amp}} - h_3(\mathbf{A}_{\mathrm{amp}})$ ▷ remove locally predictable spectrum, h_3: 3×3 mean filter
4: $\mathbf{S}_{\mathrm{dense}} \leftarrow \left|\mathrm{IFFT2}\big(\exp(\mathcal{R} + \mathrm{i}\boldsymbol{\phi})\big)\right|^2$ ▷ Eq. (2)
5: $k_s \leftarrow \mathrm{oddceil}\left(\frac{1}{4}\min(H_0P, W_0P)\right)$
6: $\mathbf{S}'_{\mathrm{dense}}[p] \leftarrow \dfrac{\mathbf{S}_{\mathrm{dense}}[p]}{h_{k_s}(\mathbf{S}_{\mathrm{dense}})[p] + \alpha_d\overline{\mathbf{S}_{\mathrm{dense}}} + \epsilon}$ ▷ $\alpha_d = 0.5$, local divisive normalization, Eq. (3)
7: For each token i, compute $s_{\mathrm{ssa}}[i] \leftarrow \frac{1}{|\Omega_i|}\sum_{p\in\Omega_i}\mathbf{S}'_{\mathrm{dense}}[p]$ over its corresponding region Ω_i, then flatten in scanning order
8: $\mathbf{s}_{\mathrm{ssa}} \leftarrow [s_{\mathrm{ssa}}[1], \ldots, s_{\mathrm{ssa}}[N]]$
9: **return** $\mathbf{s}_{\mathrm{ssa}}$

Alg. 2 implements Stage 2, Query-Guided Relevance Selection (QRS). Given the visual and query features at layer ℓ, we first apply local mean pooling to obtain region-level visual features. The visual and query features are independently centered and L_2-normalized to compute the centered similarity matrix, followed by query-wise aggregation to obtain the similarity scores $\mathbf{r}_{\mathrm{sim}}$ (Eq. 4–5). In parallel, query-to-visual attention scores are computed across all attention heads and query tokens to obtain $\mathbf{r}_{\mathrm{attn}}$. The two signals are adaptively fused according to their entropy-based response concentrations, yielding the final QRS scores $\mathbf{r}_{\mathrm{QRS}}$ (Eq. 6–7). Signals with insufficient variation are excluded using a standard-deviation threshold. If neither signal passes the threshold, the QRS stage is skipped.

Alg. 3 implements the final two-stage token selection. When QRS provides valid relevance scores, the token budget K is divided into K_1 and K_2 according to the allocation coefficient α (Eq. 8). The top-K_1 tokens ranked by $\mathbf{s}_{\mathrm{ssa}}$ are selected as structural anchors. The top-K_2 tokens ranked by $\mathbf{r}_{\mathrm{QRS}}$ are then selected from the remaining candidates $\mathcal{M}$, ensuring that the two stages select disjoint token sets. Their union forms the final retained set $\mathcal{S}$ with $|\mathcal{S}| = K$. If QRS does not provide valid relevance scores, the selection falls back to SSA by directly selecting the top-K tokens according to $\mathbf{s}_{\mathrm{ssa}}$. The selected visual tokens are arranged in their original scanning order, while their original position IDs are preserved without re-indexing, and concatenated with the query tokens to construct the compact input sequence $\widetilde{\mathbf{X}}$.

A.2 Experimental Details

A.2.1 Models

We evaluate SQ-Pruner on five representative open-source LVLMs spanning two model families and multiple model scales. **Qwen2.5-VL-7B/32B** (Bai et al., 2025) adopts a native dynamic-resolution vision encoder with window attention and a 2×2 spatial patch-merging scheme. Following prior work (Oh & Kim, 2026), all input images are resized to 1008×1008 for both variants, resulting in a fixed number of visual tokens and a uniform pruning ratio across samples. We evaluate

Algorithm 2 Query-Guided Relevance Selection

Require: Visual and query features $\mathbf{V} \in \mathbb{R}^{N\times D}$, $\mathbf{T} \in \mathbb{R}^{L_q\times D}$ at pruning layer ℓ; hyperparameter τ

$\widetilde{\mathbf{V}}[i] \leftarrow \frac{1}{|\mathcal{N}_3(i)|}\sum_{n\in\mathcal{N}_3(i)} \mathbf{V}[n]$ ▷ 3×3 local mean pooling within each spatial block

Center and L_2-normalize $\widetilde{\mathbf{V}}$ and $\mathbf{T}$ independently $\Rightarrow \mathbf{V}_n, \mathbf{T}_n$; $\mathbf{C} \leftarrow \mathbf{V}_n\mathbf{T}_n^\top$ ▷ Eq. (4)

$w_{ij} \leftarrow \mathrm{Softmax}_j(\mathbf{C}[i,j]/\tau)$; $r_{\mathrm{sim}}[i] \leftarrow \sum_{j=1}^{L_q} w_{ij}\mathbf{C}[i,j]$

$\mathbf{r}_{\mathrm{sim}} \leftarrow [r_{\mathrm{sim}}[1], \ldots, r_{\mathrm{sim}}[N]]$ ▷ Eq. (5)

Compute query-to-key attention scores $A_{j,k}^{(h)}$ with RoPE over all causally visible keys k for each query position j and head h

$r_{\mathrm{attn}}[i] \leftarrow \frac{1}{HL_q}\sum_{h=1}^{H}\sum_{j=1}^{L_q} A_{j,i}^{(h)}$

$\mathbf{r}_{\mathrm{attn}} \leftarrow [r_{\mathrm{attn}}[1], \ldots, r_{\mathrm{attn}}[N]]$

$\mathcal{U} \leftarrow \{u \in \{\mathrm{sim}, \mathrm{attn}\} : \mathrm{std}(\mathbf{r}_u) > 10^{-4}\}$

if $|\mathcal{U}| = 0$ **then**

  return $\varnothing$

end if

$$\tilde{r}_u[i] \leftarrow \begin{cases} [r_u[i]]_+ / \sum_{n=1}^{N}[r_u[n]]_+, & \sum_{n=1}^{N}[r_u[n]]_+ > \epsilon, \\ 1/N, & \text{otherwise} \end{cases}$$ ▷ for each $u \in \mathcal{U}$

$\mathcal{H}(\mathbf{r}_u) \leftarrow -\sum_{i=1}^{N}\tilde{r}_u[i]\log\big(\max(\tilde{r}_u[i], \epsilon)\big)$; $c(\mathbf{r}_u) \leftarrow \exp\big(-\mathcal{H}(\mathbf{r}_u)\big)$ ▷ Eq. (6)

$\bar{c} \leftarrow \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} c(\mathbf{r}_u)$

$\lambda_u \leftarrow c(\mathbf{r}_u)/\bar{c}$ for each $u \in \mathcal{U}$

$\mathbf{r}_{\mathrm{QRS}} \leftarrow \sum_{u\in\mathcal{U}} \lambda_u z(\mathbf{r}_u)$ ▷ Eq. (7)

return $\mathbf{r}_{\mathrm{QRS}}$

Algorithm 3 Two-Stage Token Selection

Require: Saliency scores $\mathbf{s}_{\mathrm{ssa}}$ from Alg. 1; relevance scores $\mathbf{r}_{\mathrm{QRS}}$ from Alg. 2; visual tokens $\mathbf{X}_V$ and query tokens $\mathbf{X}_T$; token budget K; allocation coefficient α

$K_1 \leftarrow \mathrm{round}\big(\frac{\alpha}{1+\alpha}K\big)$; $\quad K_2 \leftarrow K - K_1$ ▷ Eq. (8)

if $\mathbf{r}_{\mathrm{QRS}} = \varnothing$ **then**

  $\mathcal{S} \leftarrow \mathrm{TopK}(\mathbf{s}_{\mathrm{ssa}}, K)$ ▷ SSA fallback

else

  $\mathcal{P} \leftarrow \mathrm{TopK}(\mathbf{s}_{\mathrm{ssa}}, K_1)$ ▷ structural anchors

  $\mathcal{M} \leftarrow \{1, \ldots, N\} \setminus \mathcal{P}$

  $\mathcal{C} \leftarrow \mathrm{TopK}_{i\in\mathcal{M}}(\mathbf{r}_{\mathrm{QRS}}[i], K_2)$ ▷ query-relevant tokens

  $\mathcal{S} \leftarrow \mathcal{P} \cup \mathcal{C}$

end if

$\widetilde{\mathbf{X}}_V \leftarrow \mathbf{X}_V[\mathcal{S}]$; $\quad \widetilde{\mathbf{X}} \leftarrow [\widetilde{\mathbf{X}}_V; \mathbf{X}_T]$

return $\mathcal{S}$, $\widetilde{\mathbf{X}}$

SQ-Pruner at both 7B and 32B scales to assess its consistency across model sizes. **LLaVA-1.5-7B** (Liu et al., 2024a) connects a CLIP ViT-L/14 vision encoder to a Vicuna-7B language model through a two-layer MLP projector and processes images into a fixed-length visual token sequence. **LLaVA-NeXT-7B** (Liu et al., 2024b) extends LLaVA-1.5 with the AnyRes strategy, which decomposes a high-resolution image into multiple sub-images. Each image crop is independently encoded into a fixed-length visual token sequence, resulting in a longer multi-crop visual token sequence. **Video-LLaVA** (Lin et al., 2024) aligns visual representations into a unified feature space. In our experiments, we uniformly sample 8 frames from each video, resulting in a long multi-frame visual token sequence. SQ-Pruner can be directly applied to these models without fine-tuning or additional learnable parameters.

A.2.2 DATASETS

Image Understanding. We evaluate SQ-Pruner on widely used benchmarks covering general multimodal understanding, visual question answering, text-rich and document-centric understanding and hallucination evaluation. MME (Fu et al., 2026) evaluates both perception and cogni-

tion through 14 subtasks with manually designed instruction–answer pairs. MMBench (Liu et al., 2024c) and its Chinese version MMBench-CN provide hierarchical evaluation across three levels of ability. ScienceQA (Lu et al., 2022) and MMMU (Yue et al., 2024) evaluate multimodal reasoning across diverse scientific and academic disciplines. AI2D (Kembhavi et al., 2016) focuses on the understanding of scientific textbook diagrams. GQA (Hudson & Manning, 2019) evaluates compositional scene-graph reasoning. VQAv2 (Goyal et al., 2017) measures open-ended visual question answering on real-world images with multiple human-annotated answers per question. For text-rich scenarios, TextVQA (Singh et al., 2019) evaluates reading and reasoning over scene text, while DocVQA (Mathew et al., 2021) focuses on document understanding. ChartQA (Masry et al., 2022) evaluates logical and arithmetic reasoning over charts, while OCRBench (Liu et al., 2024d) covers fine-grained OCR-related tasks ranging from text recognition to handwritten formula parsing. POPE (Li et al., 2023) evaluates object hallucination through binary presence questions. VizWiz (Gurari et al., 2018) assesses visual question answering in realistic assistive scenarios with images and questions collected from blind users.

Video Understanding. We evaluate Video-LLaVA on four standard video question answering benchmarks. TGIF-QA (Jang et al., 2017) extends image question answering to short GIF videos and includes repetition counting, repeating action and state transition tasks that require spatio-temporal reasoning. MSVD-QA (Xu et al., 2017) and MSRVTT-QA (Xu et al., 2017) provide open-ended questions of five types over 1,970 and 10,000 video clips. ActivityNet-QA (Yu et al., 2019) contains human-annotated QA pairs collected from 5,800 videos, covering motion, spatial and temporal relationships for evaluating long-term spatio-temporal reasoning. We follow the evaluation protocols of the respective benchmarks. In our experiments, we uniformly sample 8 frames per video and use GLM-4-Flash as the evaluator, reporting both accuracy (%) and score (0–5). For token pruning, selection is performed independently within each frame, with no cross-frame token competition.

A.2.3 Implementation Details

Environments. All experiments are conducted on NVIDIA H20 GPUs using PyTorch 2.7.0+cu128. We use the official checkpoints and default decoding configurations for each LVLM. The same settings are used for the vanilla model and all pruning methods. We follow the official evaluation scripts provided by each benchmark. SQ-Pruner is applied at the same pruning layer ℓ for all inputs of each model, while the token budget K is specified in the main text.

Definition of Token SR. To assess whether spectral residual saliency remains effective when computed on visual tokens, we define Token SR using the same SR operator as Pixel SR. The main difference is that SR is applied to the token grid at the pruning layer rather than to the input image. Since SR operates on a single-channel spatial field, we reduce each visual token $\mathbf{z}_i \in \mathbb{R}^D$ to its ℓ_2 magnitude $\|\mathbf{z}_i\|_2$ and arrange the resulting scalars on the corresponding $H \times W$ token grid. The ℓ_2 norm provides a simple way to quantify the overall activation magnitude of each token without introducing additional dimensionality-reduction choices such as PCA or channel averaging. We apply the same spectral residual computation as in Pixel SR and retain the top-K tokens according to the resulting saliency scores.

Query Representation. For QRS, we extract the instance-specific query from the post-image text sequence while excluding fixed task instructions. Query extraction is performed at the token level. We tokenize the instance-specific query using the model tokenizer and identify its corresponding contiguous token span within the post-image text sequence. The hidden states of the identified query tokens are then used as the textual representation for query–image relevance estimation.

A.3 Extended Experiments

A.3.1 Results on Qwen2.5-VL-32B

To evaluate SQ-Pruner at a larger model scale, we conduct experiments on Qwen2.5-VL-32B, which uses 1,296 visual tokens per image under our evaluation setting. As shown in Fig. 4, SQ-Pruner achieves the best average performance among the compared pruning methods across the eight benchmarks at all three token budgets, retaining 98.1%, 94.8% and 89.8% of the Vanilla performance with

256, 128 and 64 tokens, respectively. Compared with the strongest baseline AnchorPrune (Oh & Kim, 2026), SQ-Pruner achieves higher performance retention by 1.56%, 1.29% and 3.19%.

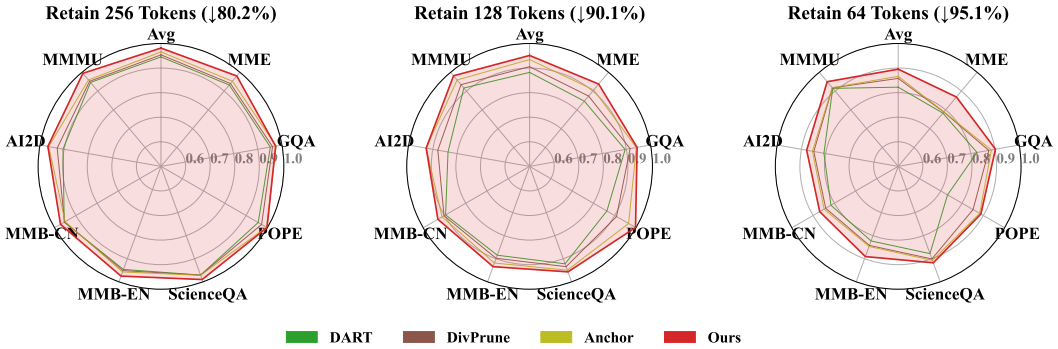

Figure 4: Performance on Qwen2.5-VL-32B.

A.3.2 Results on LLaVA-NeXT-7B

Following the token-budget convention of HiPrune (Liu et al., 2026), we evaluate SQ-Pruner on LLaVA-NeXT-7B with budgets of 640, 320 and 160 visual tokens, reporting retention ratios relative to a 2,880-token reference. As shown in Table 10, SQ-Pruner achieves the best average performance across all three token budgets, retaining 97.5%, 94.0% and 90.2% of the Vanilla performance with 640, 320 and 160 tokens. It surpasses DART (Wen et al., 2025b) by 0.4% at 640 tokens and 0.5% at 320 tokens, while achieving a 2.0% margin over DivPrune (Alvar et al., 2025) at the most aggressive pruning level. The gains are also reflected in several fine-grained visual understanding tasks, where SQ-Pruner achieves 60.7 on GQA, 87.0 on POPE, 55.1 on TextVQA and 324 on OCRBench.

Table 10: Performance on LLaVA-NeXT-7B.

Method	MME	GQA	POPE	SQA	MMB^{EN}	MMB^{CN}	VQA^{Text}	VQA^{v2}	OCRBench	VizWiz	Avg.
Upper Bound, 2,880 Tokens (**100%**)											
Vanilla	1852	64.3	86.5	70.4	67.3	57.3	61.4	80.1	521	60.7	100%
Retain 640 Tokens (↓ **77.8%**)											
$FastV_{ECCV24}$	1631	58.1	80.5	68.1	64.4	55.2	56.9	76.5	285	58.4	89.9%
$DART_{EMNLP25}$	1783	63.3	85.9	68.9	65.7	55.2	59.7	78.7	472	59.9	97.1%
$DivPrune_{CVPR25}$	1798	62.0	86.2	68.9	66.0	56.4	56.3	78.2	435	58.6	95.7%
$SparseVLM_{ICML25}$	1727	63.2	85.2	69.3	64.8	56.8	59.7	78.9	381	59.6	95.1%
$HiPrune_{ACL26}$	1833	60.6	85.0	68.0	66.8	56.9	60.6	77.8	491	59.8	**97.5%**
Ours	1792	62.2	87.8	69.4	65.6	55.4	59.0	78.9	497	59.2	**97.5%**
Retain 320 Tokens (↓ **88.9%**)											
$FastV_{ECCV24}$	1561	54.3	61.6	67.1	61.5	51.6	54.8	71.2	215	51.4	82.1%
$DART_{EMNLP25}$	1716	61.1	83.5	68.1	63.9	54.5	57.7	75.6	408	58.9	93.5%
$DivPrune_{CVPR25}$	1769	60.4	84.0	67.7	64.6	55.7	51.2	74.8	343	58.0	91.4%
$SparseVLM_{ICML25}$	1634	56.2	82.7	66.1	64.5	56.6	57.4	73.5	271	52.1	88.3%
$HiPrune_{ACL26}$	1705	57.4	78.9	67.3	64.9	56.4	56.8	74.2	394	59.8	92.2%
Ours	1719	61.2	87.1	68.8	63.9	52.8	57.0	77.8	421	57.8	**94.0%**
Retain 160 Tokens (↓ **94.4%**)											
$FastV_{ECCV24}$	1409	51.3	56.3	63.4	56.5	46.2	48.6	68.5	138	49.8	74.9%
$DART_{EMNLP25}$	1675	57.5	77.2	67.4	61.5	51.0	50.0	73.2	324	57.5	87.5%
$DivPrune_{CVPR25}$	1703	58.9	81.4	67.9	64.1	54.8	46.1	72.6	296	57.6	88.2%
$SparseVLM_{ICML25}$	1545	53.3	80.7	65.2	60.1	49.3	51.0	68.2	179	50.6	81.3%
$HiPrune_{ACL26}$	1568	51.9	67.7	68.8	60.4	51.6	43.3	71.8	270	57.4	82.8%
Ours	1634	60.7	87.0	68.4	62.3	50.8	55.1	75.9	324	56.9	**90.2%**

A.3.3 Robustness under Extreme Token Pruning

We further evaluate the robustness of SQ-Pruner on Qwen2.5-VL-7B under extreme token pruning. The visual token budget is progressively reduced from 512 to 13 tokens. As shown in Table 11, SQ-Pruner maintains strong performance under increasingly aggressive token pruning. At 256 tokens and below, SQ-Pruner achieves the best performance across all four benchmarks. Its advantage over competing methods is particularly evident at lower token budgets, demonstrating robust performance under aggressive token pruning. With 512 tokens, SQ-Pruner achieves 86.0 on POPE, 90.0 on DocVQA, 723 on OCRBench and 81.7 on TextVQA, remaining close to the Vanilla performance. At 64 tokens, SQ-Pruner still achieves 84.0 on POPE, 38.9 on DocVQA, 424 on OCRBench and 64.5 on TextVQA. Under the most aggressive setting of 13 tokens, SQ-Pruner maintains 76.1 on POPE, 15.2 on DocVQA, 215 on OCRBench and 32.2 on TextVQA. Compared with AnchorPrune (Oh & Kim, 2026), SQ-Pruner achieves margins of 20.3 on POPE, 3.8 on DocVQA, 101 on OCRBench and 6.8 on TextVQA. These results demonstrate the robustness of SQ-Pruner under extreme visual token pruning.

Table 11: Comparison of visual token pruning methods under extreme token reduction.

Method	POPE					DocVQA					OCRBench					TextVQA				
Vanilla	85.9					94.1					745					82.5				
	512	256	128	64	13	512	256	128	64	13	512	256	128	64	13	512	256	128	64	13
FastV$_{\text{ECCV24}}$	86.1	82.6	76.5	62.7	25.5	76.5	50.6	31.8	20.9	10.8	688	498	398	305	103	82.0	78.5	71.2	60.2	19.5
DivPrune$_{\text{CVPR25}}$	86.3	84.9	82.6	79.3	47.2	85.2	68.3	47.8	30.2	10.8	671	559	401	243	42	79.8	74.2	64.6	51.1	22.7
VisionZip$_{\text{CVPR25}}$	85.0	84.8	82.0	76.6	43.6	82.6	67.6	44.6	27.2	10.8	655	581	464	343	135	77.8	74.0	63.1	49.2	15.2
HiPrune$_{\text{ACL26}}$	86.2	84.9	81.9	76.6	35.8	87.5	67.3	44.7	27.3	11.0	696	586	465	341	110	80.6	74.0	61.8	49.0	20.9
AnchorPrune$_{\text{ECCV26}}$	86.1	85.0	84.2	80.6	55.8	86.3	72.9	50.9	31.3	11.4	676	604	481	356	114	79.7	76.6	68.5	57.3	25.4
Ours	86.0	86.3	85.5	84.0	76.1	90.0	75.1	56.7	38.9	15.2	723	646	555	424	215	81.7	79.5	73.5	64.5	32.2

A.3.4 Efficiency Analysis of Pruning Layer

We further explore the effect of the pruning layer under different token budgets. We index the model layers starting from 0, and pruning at layer ℓ uses the hidden states output by that layer. For the input stage (Layer -1), where no layer-wise hidden states are available, we compute Centered Similarity directly from the visual and query embeddings and use it as the sole signal in QRS. As shown in Fig. 5, moving the pruning layer from the input stage (Layer -1) to Layer 1 yields a substantial performance improvement across all token budgets. With 64 retained tokens, SQ-Pruner improves from 75.4% at the input stage to 81.6% at Layer 1, corresponding to a 6.2% gain at the cost of only 1.2 additional TFLOPs. Further delaying pruning to deeper layers continues to improve performance, but the gains become smaller relative to the additional computational cost. Layer 1 therefore provides a favorable trade-off between performance and computational cost. We adopt Layer 1 as the default pruning layer throughout our experiments.

A.3.5 Ablation of Adaptive Fusion

To validate the adaptive fusion in Eq. (7), we replace the adaptive weights λ_u with a fixed coefficient β, resulting in $\mathbf{r}_{\text{QRS}} = z(\mathbf{r}_{\text{sim}}) + \beta z(\mathbf{r}_{\text{attn}})$. Table 12 compares fixed coefficients $\beta \in \{0.8, 1.0, 1.2\}$ with our adaptive fusion on Qwen2.5-VL-7B.

The best-performing fixed β varies across token budgets. $\beta = 1.2$ achieves the highest average performance among fixed settings at 256 and 128 tokens, while $\beta = 1.0$ performs best at 64 tokens. In contrast, adaptive fusion achieves the highest average performance across the three token budgets, reaching 94.75%, 88.73% and 81.60% at 256, 128 and 64 tokens. The improvement over the best fixed setting increases from 0.33% to 0.47% and 1.00% as the token budget decreases. At 64 tokens, adaptive fusion outperforms the best fixed setting by 1.3 on DocVQA, 15 on OCRBench and 2.0 on TextVQA. These results suggest that the relative importance of the two signals varies across inputs, favoring adaptive weighting over a fixed coefficient. We therefore use adaptive fusion throughout all experiments.

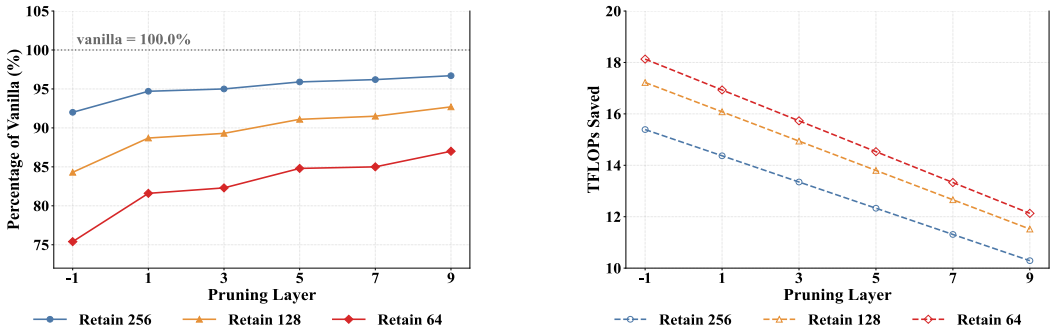

Figure 5: Effect of the pruning layer on performance and TFLOPs savings. (a) Performance relative to the Vanilla model under different token budgets. (b) TFLOPs saved at different pruning layers.

Table 12: Ablation study of adaptive fusion on Qwen2.5-VL-7B.

Method	MME	GQA	POPE	SQA	MMB^{EN}	MMB^{CN}	VQA^{Text}	DocVQA	OCRBench	AI2D	ChartQA	MMMU	Avg.
Upper Bound, 1,296 Tokens (**100%**)													
Vanilla	2326.9	59.8	85.9	87.8	83.8	80.2	82.5	94.1	745	83.0	80.5	50.7	100.00%
Retain 256 Tokens (↓ **80.2%**)													
$\beta = 0.8$	2291.0	57.8	86.1	86.3	81.1	77.3	79.0	73.8	645	81.6	73.5	48.2	94.35%
$\beta = 1.0$	2305.4	57.8	86.4	86.2	81.0	77.5	79.3	73.6	642	81.6	73.1	48.4	94.41%
$\beta = 1.2$	2283.8	57.7	86.0	86.1	81.0	77.7	79.4	74.1	648	81.7	73.1	48.4	94.42%
Ours	2294.1	58.4	86.3	85.9	80.8	77.4	79.5	75.1	646	81.6	73.6	49.1	**94.75%**
Retain 128 Tokens (↓ **90.1%**)													
$\beta = 0.8$	2257.1	56.7	85.5	82.6	78.0	74.0	72.0	55.9	540	78.3	62.9	47.5	88.01%
$\beta = 1.0$	2254.2	56.5	85.2	82.4	78.1	74.8	72.7	56.0	543	78.0	63.9	47.3	88.17%
$\beta = 1.2$	2248.8	56.4	85.1	82.4	77.9	75.2	73.6	56.2	541	78.3	64.0	47.2	88.26%
Ours	2246.4	57.2	85.5	82.7	77.8	74.9	73.5	56.7	555	79.0	63.8	47.8	**88.73%**
Retain 64 Tokens (↓ **95.1%**)													
$\beta = 0.8$	2118.2	54.4	83.7	81.1	75.9	70.6	61.5	37.1	398	73.9	51.8	45.8	80.13%
$\beta = 1.0$	2148.4	54.6	83.4	81.4	75.9	70.9	62.5	37.6	409	74.4	52.6	45.2	80.60%
$\beta = 1.2$	2144.9	54.3	83.0	81.6	75.4	71.0	63.3	37.3	410	74.5	52.5	45.0	80.52%
Ours	2165.8	55.3	84.0	81.1	76.5	71.5	64.5	38.9	424	75.8	52.9	45.4	**81.60%**

A.3.6 Ablation of Divisive Normalization and Pixel SR

We evaluate the contribution of divisive normalization and Pixel SR on Qwen2.5-VL-7B under different token budgets. As shown in Table 13, Pixel SR with divisive normalization consistently outperforms Pixel SR, improving the average performance from 92.2%, 84.0% and 75.5% to 93.2%, 85.6% and 77.5% at 256, 128 and 64 tokens. The gain from divisive normalization becomes larger as the token budget decreases, reaching 2.0% at 64 tokens. Meanwhile, Token SR performs substantially worse across all token budgets, particularly on text-rich benchmarks such as TextVQA, DocVQA and OCRBench. These results demonstrate the effectiveness of pixel-level SR and the benefit of divisive normalization for preserving informative visual tokens.

A.4 Limitations and Future Work

Although SQ-Pruner performs well across a range of settings, several limitations remain. First, SSA can provide weaker structural cues for images with low contrast, weak texture, or heavy blur, potentially affecting token selection. Second, QRS may be less effective for highly abstract queries, such as "What does this image make you feel?" where relevant visual information may be distributed across multiple regions of the image rather than concentrated in a specific area. In future

Table 13: Ablation study of divisive normalization and pixel-level spectral residual estimation.

Method	MME	GQA	POPE	SQA	MMB^{EN}	MMB^{CN}	VQA^{Text}	DocVQA	OCRBench	AI2D	ChartQA	MMMU	Avg.
Upper Bound, 1,296 Tokens (**100%**)													
Vanilla	2326.9	59.8	85.9	87.8	83.8	80.2	82.5	94.1	745.0	83.0	80.5	50.7	100.0%
Retain 256 Tokens (↓ **80.2%**)													
Token SR	1975.0	58.5	83.9	84.3	77.7	74.4	38.4	43.5	305	74.1	49.1	47.6	78.3%
Pixel SR	2283.8	55.7	82.3	85.9	80.0	77.1	70.9	72.2	632	80.4	72.0	48.8	92.2%
Pixel SR Div	2283.6	56.7	83.9	86.2	81.4	77.5	72.7	73.6	651	80.6	72.1	48.2	**93.2%**
Retain 128 Tokens (↓ **90.1%**)													
Token SR	1825.5	56.2	81.0	81.8	72.7	69.6	25.0	22.7	183	68.0	32.6	45.3	68.7%
Pixel SR	2147.8	52.7	77.4	82.3	76.6	73.3	60.8	51.0	518	77.2	63.7	46.4	84.0%
Pixel SR Div	2178.3	54.1	80.3	82.5	77.9	73.6	63.2	52.0	521	78.0	64.7	48.1	**85.6%**
Retain 64 Tokens (↓ **95.1%**)													
Token SR	1736.2	53.0	77.2	77.6	67.9	64.7	18.8	13.5	111	65.4	21.6	44.9	62.5%
Pixel SR	1951.8	49.6	73.0	80.4	72.4	70.2	49.7	32.8	395	72.9	49.6	46.3	75.5%
Pixel SR Div	2026.5	50.7	75.8	79.7	74.8	71.5	50.1	35.3	402	74.8	54.2	46.3	**77.5%**

work, we will explore adaptive pruning strategies that jointly consider image complexity and query characteristics to achieve more robust token allocation.

A.5 Visualization

A.5.1 Visualization of Token-Level and Pixel-Level SR

To further illustrate the effectiveness of Pixel SR, we visualize the intermediate saliency maps and the corresponding 128 retained visual tokens in Fig. 6. As shown in the third and fifth columns, Pixel SR assigns high saliency to fine-grained text, chart components and visual contours. The corresponding retained tokens show that Pixel SR can effectively leverage fine-grained visual information for token pruning.

A.5.2 Visualization of Query-Guided Relevance Estimation

To illustrate the responses of different query-guided relevance signals, we visualize their heatmaps in Fig. 7. Centered Similarity highlights concentrated regions relevant to the query, while Attention Score captures regions receiving stronger attention from the model. Their adaptive fusion in QRS produces more focused relevance maps, highlighting query-related visual information such as specific text on airplanes, license plates, and target objects in cluttered scenes.

A.5.3 Visualization of SQ-Pruner

To illustrate the token pruning behavior of SQ-Pruner, we compare its retained visual token masks with those of competing pruning methods in Fig. 8. SQ-Pruner more selectively retains visual tokens from query-relevant regions. By combining the query-independent structural prior from SSA with the query-guided relevance from QRS, SQ-Pruner more effectively retains visual tokens corresponding to query-relevant content, such as “refrigerator,” “jeans,” and “glove” in cluttered scenes.

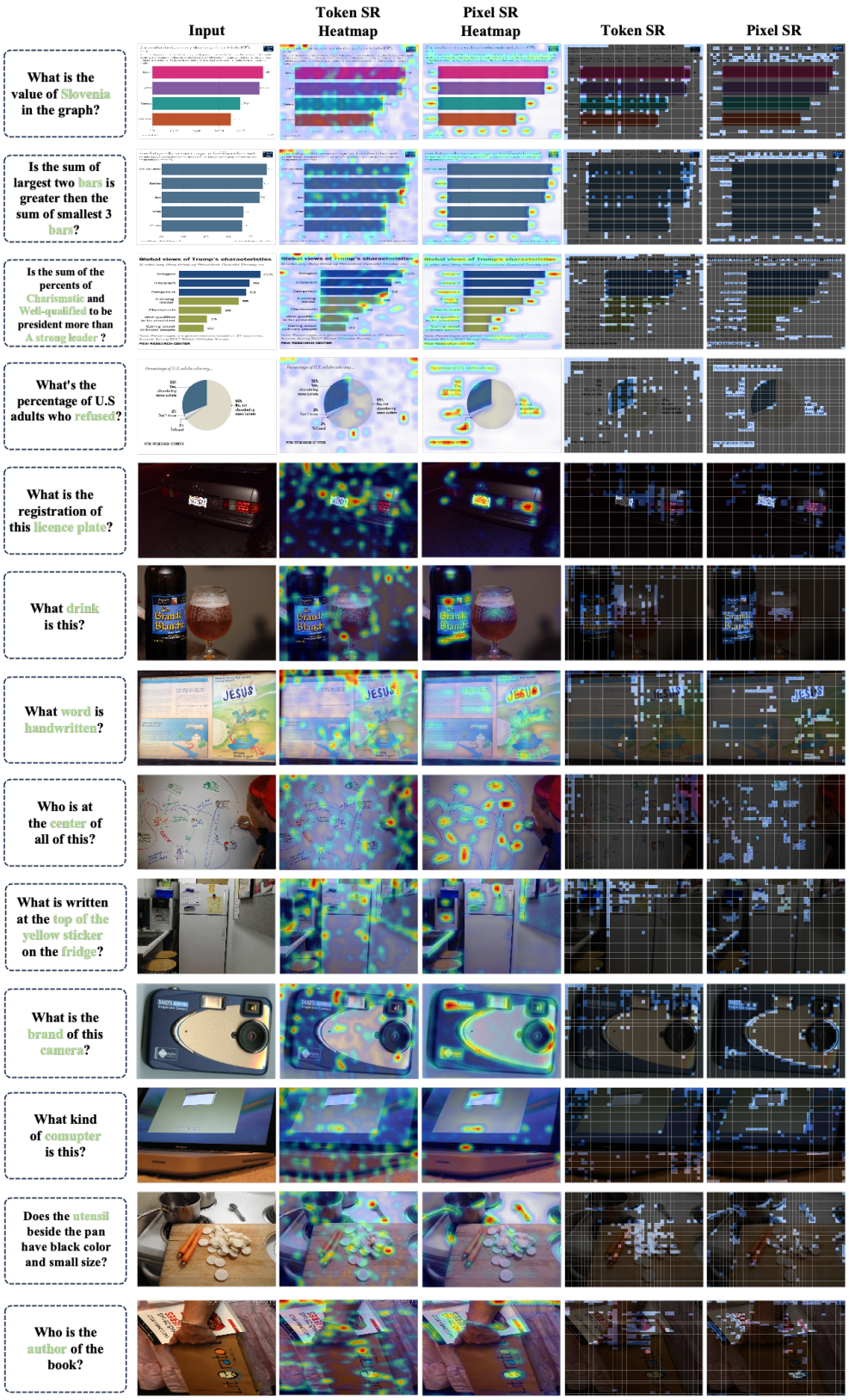

Figure 6: Visualization of Token SR and Pixel SR saliency maps and the corresponding 128 retained visual tokens.

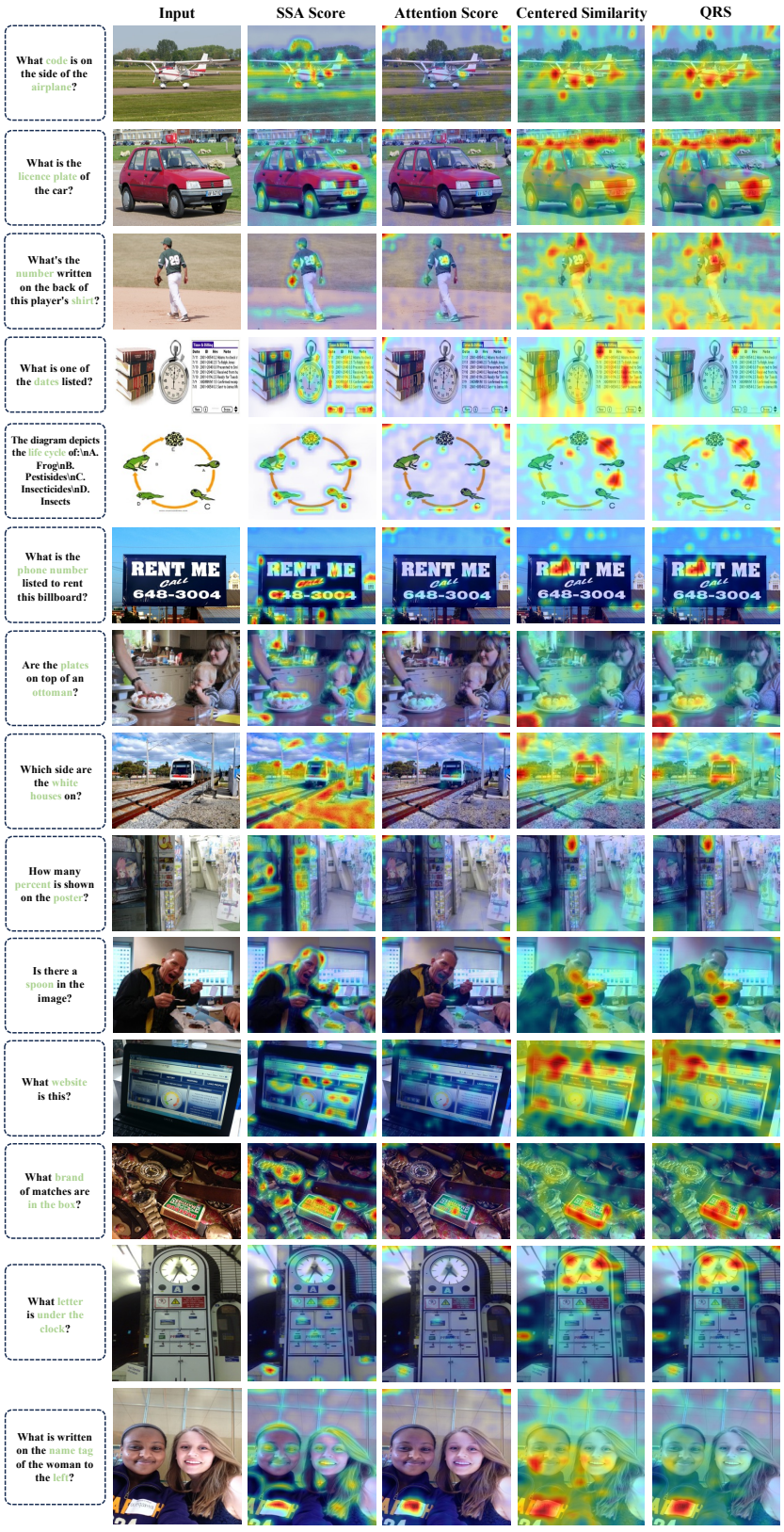

Figure 7: Visualization of query-guided relevance signals.

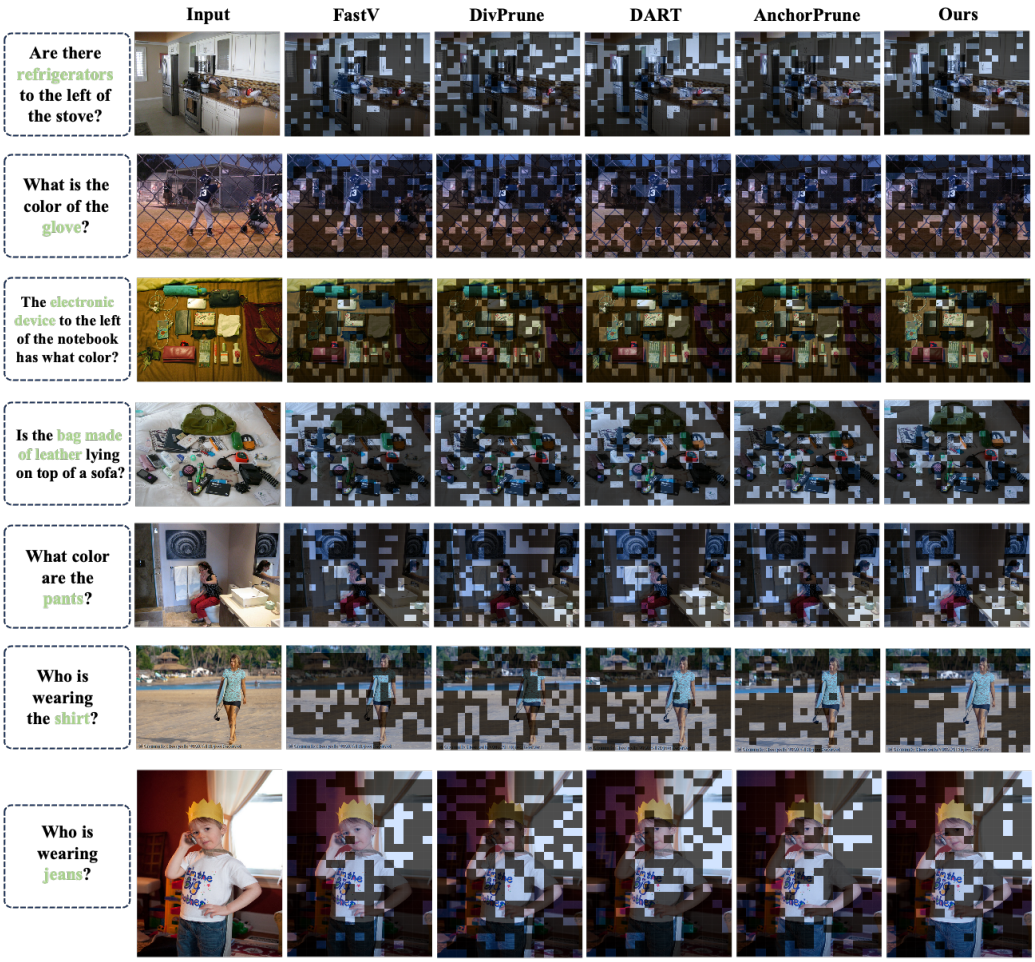

Figure 8: Visualization of token pruning results. Black patches indicate pruned tokens, with 128 visual tokens retained.