CAR: REDUCING DECODER OUTPUT SHIFT IN VISUAL TOKEN PRUNING VIA COVERAGE SELECTION AND ATTENTION REALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token pruning is an effective way to accelerate multimodal large language models (MLLMs). However, existing methods degrade substantially under aggressive pruning, especially on visual grounding (VG). We analyze the sources of this degradation from the perspective of the output shift that pruning causes in decoder layers, and decompose this shift in closed form into two terms, an image summary error and an attention allocation error. The summary error stems from the loss of visual information, and the allocation error from the drop in the share of attention assigned to visual tokens. To reduce both errors, we propose CAR (Coverage selection and Attention Reallocation), a token pruning framework that restores attention share while maintaining information. Specifically, to minimize the summary error, whose upper bound tightens as the retained tokens better cover all visual tokens, our coverage selection retains tokens that are both salient and non-redundant, with saliency from encoder heads chosen by a label-free probe. To minimize the allocation error, our attention reallocation reassigns the attention of the discarded tokens to the retained ones through a closed-form logit bias. On Qwen2.5-VL-7B with 90% of the visual tokens pruned, CAR retains 93.0% of the unpruned accuracy on visual question answering (VQA) and 77.9% on VG, against 26.5% for the best existing method on VG; the logit bias alone raises the VG accuracy of existing methods by up to 24 points. Experiments on multiple MLLMs across VQA, VG and video understanding show that CAR generalizes well and consistently outperforms existing methods.

1 INTRODUCTION

Multimodal large language models (MLLMs) (Liu et al., 2024a; Bai et al., 2025b; Zhu et al., 2025) have achieved strong performance on vision-language tasks such as visual question answering (Singh et al., 2019; Liu et al., 2024d), visual grounding (Yu et al., 2016; Mao et al., 2016) and video understanding (Fu et al., 2025b; Li et al., 2024). However, modern vision encoders represent a single image as hundreds to thousands of patch tokens (Liu et al., 2024b; Chen et al., 2024b), which dominate prefilling at a quadratic attention cost (Chen et al., 2024a; Xing et al., 2025) and hinder deployment under real-time or resource constraints. Visual token pruning therefore reduces this cost by retaining only a subset of the visual tokens while maximally preserving performance.

We categorize mainstream visual token pruning methods into three types based on their operational stage: Vision Encoder-Side Pruning (Yang et al., 2025; Alvar et al., 2025; Zhang et al., 2025a; Liu et al., 2026a; Oh & Kim, 2026; Wang et al., 2026b), LLM Decoder-Side Pruning (Chen et al., 2024a; Zhang et al., 2025b; Xing et al., 2025; Wen et al., 2025; Zhu et al., 2026), and Two-Stage Pruning (Zhang et al., 2026a; Huang et al., 2026). These methods maintain their performance on Visual Question Answering (VQA) but degrade sharply on Visual Grounding (VG) under aggressive pruning (Huang et al., 2026). Figure 1(a) compares these pruning methods with two simple baselines, random sampling and average pooling, on the eight splits of RefCOCO, RefCOCO+ and RefCOCOg (Yu et al., 2016; Mao et al., 2016). Counterintuitively, these methods hold only a small advantage over the simple baselines. This observation raises a question: *Why do existing pruning methods exhibit such significant performance degradation?*

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.

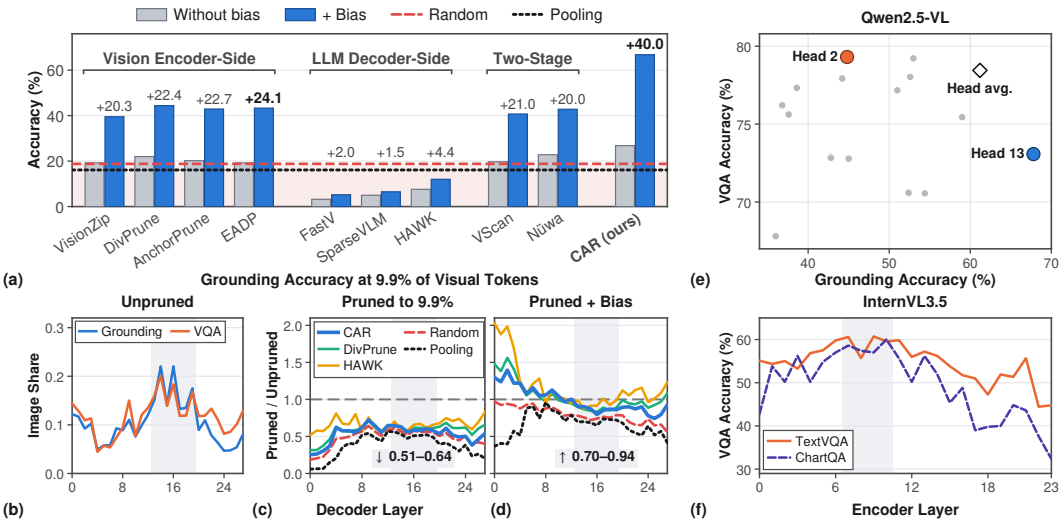


Figure 1: **(a)** Grounding accuracy of existing methods and two baselines (random sampling, pooling) on Qwen2.5-VL-7B at 9.9% retention, averaged over eight RefCOCO/+/g splits, without and with the uniform bias $\log(N/k)$. **(b)** Visual attention share, the share of the text tokens' attention allocated to the image, at each decoder layer in the unpruned model. **(c, d)** Visual attention share after pruning to 9.9%, divided by its unpruned value on the same image, without (c) and with (d) the bias. The grey band in (b)–(d) marks layers 13 to 19. **(e)** Grounding and VQA accuracy with saliency from each head of the last encoder layer of Qwen2.5-VL-7B. **(f)** VQA accuracy on TextVQA and ChartQA with saliency from each encoder layer of InternVL3.5-8B. Setup in Appendix C.13.

We find that this performance degradation stems not only from the loss of information carried by discarded tokens, but also from a drop in the visual attention share, the share of attention that the text allocates to the image. In the unpruned model, the visual attention share is highest in the intermediate layers (Figure 1(b)), consistent with the finding that these layers transfer question-relevant visual information into the text tokens (Zhang et al., 2025c). Since the softmax normalizes over all tokens, pruning removes the attention that the discarded visual tokens received, while the text tokens keep theirs. As a result, the visual attention share in these layers falls to only 51–64% of its unpruned value regardless of the selection method, even under random sampling (Figure 1(c)). As the share shrinks, the attention output relies less on the image and more on the text, weakening the visual evidence available for the answer. However, most existing methods focus on token selection while overlooking the loss of visual attention share.

To quantify these two sources of performance degradation, we formulate pruning as the problem of minimizing the shift in decoder layer outputs and derive a closed-form decomposition of this shift (Section 2.1). This shift is the sum of two terms, an image summary error caused by the information loss and an attention allocation error caused by the drop in the visual attention share. We further derive an upper bound on the image summary error, which decreases as the retained tokens better cover all visual tokens, especially those that receive more attention from the query. We minimize the attention allocation error with an additive logit bias on the retained visual tokens to restore the visual attention share, which requires the k retained tokens to collectively carry the total attention originally held by all N visual tokens. This condition is underdetermined with respect to the k bias terms, resulting in multiple possible solutions; we present two approximate solutions in closed form: a uniform bias that assigns an identical bias to each retained token, determined solely by the retention ratio; and a per-cell bias that allocates to each retained token the portion of attention originally held by the group of tokens it represents.

In this paper, we propose CAR, a novel framework that reduces the decoder output shift in visual token pruning through Coverage selection and Attention Reallocation. Specifically, as shown in Figure 2, to minimize the image summary error, we introduce a weighted coverage selection method with a spatial kernel, which selects tokens that are both important and non-redundant so that they represent the entire set of visual tokens. Since the head or layer of the vision encoder that best supplies the token importance varies across tasks and backbones (Figure 1(e, f)), we further introduce

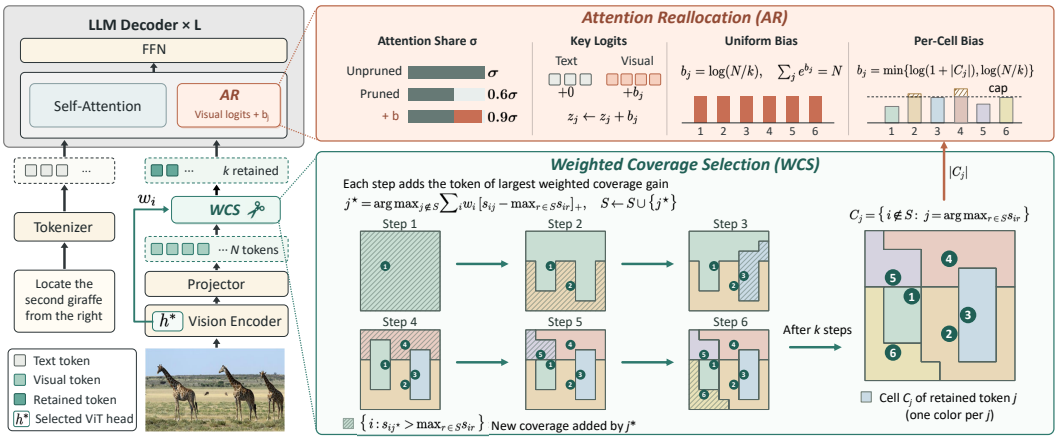


Figure 2: **Overview. Left:** WCS reduces the N visual tokens to k before the LLM, weighting the coverage by the saliency w_i of the ViT head h^* that the probe selects; AR adds a logit bias b_j to the retained tokens in each decoder layer. **Bottom right:** greedy selection adds in each step the token with the largest new coverage; after k steps, each discarded token is assigned to the most similar retained token, which forms the cells C_j. **Top right:** pruning lowers the image's attention share σ, and the bias restores it, uniformly for grounding and capped per cell for VQA.

a head probe that selects it by the consistency between the outputs of the pruned and the unpruned model. To minimize the attention allocation error, for grounding we apply the uniform bias, which scales the attention of all retained tokens by the same factor and thus keeps its spatial distribution, and for question answering the per-cell bias, which returns to each retained token the attention of the tokens it represents. The uniform bias depends only on the retention ratio and leaves the retained tokens unchanged, so it can be attached to any existing pruning method. Applying it alone to existing pruning methods brings their visual attention share closer to that of the unpruned model (Figure 1(d)) and improves their grounding accuracy by up to 24 points (Figure 1(a)). Our main contributions are summarized as follows:

❶ We present a closed-form decomposition of the output shift caused by visual token pruning, together with a practical form for each of its two terms: an upper bound that turns the image summary error into a coverage criterion, and an exact condition that turns the attention allocation error into a constraint on the biases of the retained tokens.

❷ We propose Coverage selection and Attention Reallocation (CAR) for training-free visual token pruning, which minimizes the two errors via a saliency-weighted coverage objective with a label-free head probe and an additive logit bias on the retained tokens, yielding a retained set that represents the image and carries the attention share the image held before pruning.

❸ Extensive experiments show that CAR outperforms prior state-of-the-art methods across multiple MLLMs and benchmarks, leading the strongest baseline by 44.0 points in grounding accuracy at a retention ratio of 9.9%, and generalizes well across VQA, grounding, and video understanding.

2 Method

2.1 Formulating Token Pruning as Output Shift Minimization

To measure the effect of pruning, we consider the output of one attention head of a decoder layer at a text query, through which visual information is aggregated into the text positions where the answer is generated. Let T index the text tokens and V the N visual tokens, let k_u and v_u be the key and the value of token $u \in T \cup V$, and let $z_u = q^\top k_u / \sqrt{d_h}$ be the logit with which a text query q attends to it, where d_h is the head dimension. The unpruned attention output is

$$O = \sum_{u \in T \cup V} p_u \, v_u, \qquad p_u = \mathrm{softmax}_{T \cup V}(z)_u = \frac{e^{z_u}}{\sum_{w \in T \cup V} e^{z_w}}. \tag{1}$$

Token pruning retains a subset $S \subseteq V$ of size k and discards the remainder, leaving the text tokens intact, so that the softmax loses terms on its visual side only. To let the retained tokens compensate for the lost terms, we allow an additive logit bias b_u on each retained visual token, which scales its softmax term by e^{b_u}, with $b_t = 0$ for $t \in T$, so that the pruned output is

$$O' \;=\; \sum_{u \in T \cup S} p'_u \, v_u, \qquad p'_u \;=\; \mathrm{softmax}_{T \cup S}(z + b)_u \;=\; \frac{e^{z_u + b_u}}{\sum_{w \in T \cup S} e^{z_w + b_w}}. \tag{2}$$

Our objective is to minimize the expected output shift of the layer,

$$\min_{|S|=k,\; b} \;\; \mathcal{E}(S, b) \;=\; \mathbb{E}\big[\|O' - O\|^2\big], \tag{3}$$

where the expectation is over queries and images. Two factors control this error: the retained set S, and the attention the decoder pays to it through b. Evaluating Eq. 3 requires the unpruned output O, however, grouping the attention sum by modality writes each output as a mixture of an image summary and a text summary (Appendix B.1),

$$O \;=\; \sigma \, \bar{v}_{\mathrm{img}} + (1 - \sigma) \, \bar{v}_{\mathrm{txt}}, \qquad O' \;=\; \sigma' \, \bar{v}_S + (1 - \sigma') \, \bar{v}_{\mathrm{txt}}. \tag{4}$$

Here $\sigma = \sum_{u \in V} p_u$ is the share of its attention that the query spends on the image, and $\bar{v}_{\mathrm{img}}$ and $\bar{v}_{\mathrm{txt}}$ are the attention-weighted mean values of the two groups; after pruning, σ' and $\bar{v}_S$ are the same two quantities on the retained set, while the text summary is unchanged. Subtracting the two outputs and splitting $\sigma' \bar{v}_S$ into $\sigma \bar{v}_S + (\sigma' - \sigma)\bar{v}_S$ separates the change in the summary from the change in the share:

$$O' - O \;=\; \underbrace{\sigma\big(\bar{v}_S - \bar{v}_{\mathrm{img}}\big)}_{\mathcal{E}_{\mathrm{sum}}\ \text{(image summary error)}} \;+\; \underbrace{(\sigma' - \sigma)\big(\bar{v}_S - \bar{v}_{\mathrm{txt}}\big)}_{\mathcal{E}_{\mathrm{alloc}}\ \text{(attention allocation error)}}. \tag{5}$$

By the triangle inequality, $\|O' - O\|^2 \le 2\|\mathcal{E}_{\mathrm{sum}}\|^2 + 2\|\mathcal{E}_{\mathrm{alloc}}\|^2$, so we minimize Eq. 3 through this upper bound, reducing the two terms separately. The image summary error captures the gap between the summaries formed from S and from V. The attention allocation error reflects attention shifting from the image to the text; it vanishes exactly when $\sigma' = \sigma$.

2.2 Minimizing the Image Summary Error

Minimizing $\mathcal{E}_{\mathrm{sum}} = \sigma(\bar{v}_S - \bar{v}_{\mathrm{img}})$ directly is infeasible: we cannot obtain $\bar{v}_{\mathrm{img}}$, and choosing S is a search over $\binom{N}{k}$ subsets. It is, however, bounded by how well the retained set covers the image.

Proposition 1 (Coverage bounds the image summary error) *Let $a_i = e^{z_i}/Z_V$, with $Z_V = \sum_{u \in V} e^{z_u}$, be the share of the image's attention that visual token i receives from the query, so that $\sum_{i \in V} a_i = 1$, and let $\rho = \sum_{j \in S} a_j$ be the share held by the retained set. If the value vectors have equal norm R, and if the attention of each discarded token is carried by the retained token with the most similar value, then*

$$\big\|\bar{v}_S - \bar{v}_{\mathrm{img}}\big\| \;\le\; \sqrt{2}\, R \sqrt{1 - \rho}\, \sqrt{1 - F_v(S)}, \qquad F_v(S) = \sum_{i \in V} a_i \max_{j \in S} \cos(v_i, v_j). \tag{6}$$

The proof is in Appendix B.2. To minimize the image summary error, the bound therefore suggests maximizing $F_v(S)$, the attention-weighted coverage of the image by the retained set. The bound asks that *each visual token be covered by a retained token similar to it, and that the coverage of a token contribute in proportion to the attention it receives*. The shares a_i are unknown at selection time, since they depend on how later layers attend to the visual tokens and vary with the layer and the query. We therefore maximize a computable surrogate of F_v.

Weighted coverage selection. To have the retained set well represent the original visual tokens, we formulate the token selection as a weighted facility-location objective (Iyer et al., 2021),

$$F(S) = \sum_{i=1}^{N} w_i \max_{j \in S} s_{ij}, \qquad s_{ij} = \big[\cos(x_i, x_j)\big]_+, \tag{7}$$

where x_i is the encoder feature of token i, w_i its saliency, the attention it receives from all the tokens in the vision encoder, and s_{ij} the cosine similarity between two features with negative values clipped to zero. The inner maximum is how well token i is represented in S, and the sum asks the retained set to represent the whole image, giving more weight to the salient tokens. The similarity of Eq. 7 does not account for spatial structure, so a token can be covered by a distant one of similar appearance, and its region may keep no token at all. Since grounding depends on the coordinates of the target region, for grounding we multiply the feature similarity by a spatial kernel on the token grid,

$$s_{ij} = \big[\cos(x_i, x_j)\big]_+ \cdot \exp\Big(-\frac{\|g_i - g_j\|^2}{2\,(N/k)}\Big), \tag{8}$$

where g_i is the grid position of token i. The kernel discounts the feature similarity by the distance on the grid. The scale of the kernel is the area each of the k representatives covers, about N/k cells, so it widens as the budget tightens. Question answering, which concerns semantic information, uses the similarity of Eq. 7, without the spatial kernel.

Head probe. The heads of the vision encoder attend to different aspects of the image, and they are therefore unequal as estimates of a_i. A head that disagrees with the decoder weights the wrong tokens, and the retained set then covers parts of the image that the query does not attend to. Accuracy varies widely with the head, and the best source differs between backbones and task families (Figure 1(e, f)). Since a_i is not observable, we select the source with a probe: each candidate h, the heads of the final block and their average, is scored by the agreement between the pruned model's output and the unpruned model's,

$$\mathrm{score}(h) \;=\; \frac{1}{n}\sum_{i=1}^{n} \mathrm{sim}\big(y_i^{(h)},\, y_i^{\mathrm{full}}\big), \tag{9}$$

where sim is the task's own evaluation metric, exact match for question answering and the IoU between the predicted boxes for grounding. The probe is run once offline for each backbone and task family, and the best-scoring candidate h^* is then fixed (Appendix C.3). On InternVL3.5-8B and LLaVA-1.5-7B, whose late layers exhibit attention sinks (Darcet et al., 2024), the probe selects the encoder layer as well (Appendix C.3).

Greedy selection. F is monotone and submodular, so although maximizing it is NP-hard, greedy selection reaches at least $1 - 1/e$ of the optimum (Nemhauser et al., 1978) (Appendix B.3). It builds S one token at a time until $|S| = k$, each step adding the token of largest marginal coverage,

$$\arg\max_{j \notin S} \sum_{i=1}^{N} w_i \Big[\, s_{ij} - \max_{r \in S} s_{ir} \,\Big]_+, \tag{10}$$

where $\max_{r\in S} s_{ir}$ is how well token i is already covered and the bracket is the coverage that j adds for it, weighted by w_i. Once S is fixed, each discarded token is assigned to the retained token most similar to it, and the tokens assigned to j form its cell C_j.

2.3 Minimizing the Attention Allocation Error

Pruning lowers the share of the attention that the image receives from the query (Figure 1(c)). The allocation error $\mathcal{E}_{\mathrm{alloc}} = (\sigma' - \sigma)(\bar{v}_S - \bar{v}_{\mathrm{txt}})$ of Eq. 5 is the product of the change in the share σ and the gap between the image summary and the text summary. The gap does not vanish in general, so the error vanishes only when the share is restored, $\sigma' = \sigma$. We restore it with the logit bias b_j of Eq. 2, which raises the logits of the retained tokens.

The two shares, σ before pruning and σ' after it, differ only in their visual part: both have the form $x/(Z_T + x)$ with the same text part $Z_T = \sum_{t\in T} e^{z_t}$, while x is $Z_V = \sum_{i\in V} e^{z_i}$ before pruning and $\sum_{j\in S} e^{z_j + b_j}$ after (Appendix B.1). Since $x \mapsto x/(Z_T + x)$ is strictly increasing, the image keeps its share exactly when the biased visual part again equals Z_V. With $m_j = e^{b_j}$,

$$\sum_{j\in S} e^{z_j}\, m_j \;=\; Z_V. \tag{11}$$

In words, the bias lets the k retained tokens carry the attention mass that all N visual tokens carried before pruning. This is an equation involving k unknowns, meaning multiple solutions exist, and the specific choice of solution is crucial. Since $\bar{v}_S$ is constructed using biased weights, if the m_j values corresponding to the chosen solution vary across the retained tokens, the image summary will change, thereby affecting $\mathcal{E}_{\text{sum}}$.

Uniform bias. The simplest solution gives each retained token the same $m_j = m$. Equation 11 then reads $m \sum_{j \in S} e^{z_j} = Z_V$, and since $\sum_{j \in S} e^{z_j} = \rho Z_V$, where ρ is the share of the visual attention held by the retained set (Proposition 1), this is $m\rho = 1$ and $m = 1/\rho$. Computing ρ needs the attention of all N visual tokens at a decoder layer, and pruning has already discarded $N - k$ of them. We therefore replace ρ by its expectation over queries, treating the visual tokens as exchangeable under the query distribution, so that each receives the same attention in expectation. Then $\mathbb{E}[\rho] = k/N$ whichever k are retained, and

$$m_j \;=\; \frac{N}{k}, \qquad b_j \;=\; \log \frac{N}{k}. \tag{12}$$

The substitution solves Eq. 11 in expectation and not per query, and the gap grows with the variance of ρ. Equation 12 depends only on the retention ratio k/N, and it applies a uniform bias to the retained tokens, which leaves $\bar{v}_S$ and therefore $\mathcal{E}_{\text{sum}}$ unchanged (Appendix B.4).

Per-cell bias. The selector assigns each discarded token to the retained token most similar to it, and those assigned to j form its cell C_j (Section 2.2). A second solution gives each retained token the mass of the tokens it represents, $e^{z_j} m_j = e^{z_j} + \sum_{i \in C_j} e^{z_i}$. Each discarded token belongs to exactly one cell, so summing over $j \in S$ returns Z_V and Eq. 11 holds. The weight of a retained token in $\bar{v}_S$ is then the attention of the tokens it represents, which is what Proposition 1 assumes. The sum $\sum_{i \in C_j} e^{z_i}$ is not observable, and setting $e^{z_i} \approx e^{z_j}$ inside a cell gives

$$m_j \;=\; 1 + |C_j|, \qquad b_j \;=\; \log\big(1 + |C_j|\big). \tag{13}$$

The assumption is exchangeability inside a cell, which the selector built by similarity. Since m_j grows with the size of the cell, a token that stands for many discarded ones is weighted up, and $\bar{v}_S$ changes with it. Inside a cell $e^{z_i} < e^{z_j}$ is the common case, so the count overstates what the cell carried, the more so the larger the cell. To limit it we cap the count at its mean N/k,

$$m_j \;=\; \min\Big\{1 + |C_j|,\; \frac{N}{k}\Big\}, \qquad b_j \;=\; \min\Big\{\log\big(1 + |C_j|\big),\; \log \frac{N}{k}\Big\}, \tag{14}$$

which restores less share than pruning removed and perturbs $\bar{v}_S$ less (Appendix B.5).

Choice of the bias form. The uniform bias assigns the same value to each retained token, while the per-cell forms give more to the tokens that stand for more discarded ones, so what they restore follows the redundancy of each region. Grounding asks *where*, so the answer depends on which part of the image the attention falls on, and the uniform bias scales each retained token by the same factor and leaves that unchanged. Question answering asks *what*, and the answer depends on the content of the image, which the per-cell forms recover more faithfully by returning to each retained token the attention of the tokens it represents. We therefore use the uniform bias for grounding and the capped per-cell bias for question answering (Section 3.2).

3 Experiments

Experimental setup. To verify the generality and effectiveness of our method, we evaluate it on image understanding, visual grounding and video understanding across several MLLMs and benchmarks. The backbones are Qwen2.5-VL-7B (Bai et al., 2025b), Qwen3-VL-8B (Bai et al., 2025a), InternVL3.5-8B (Wang et al., 2025) and LLaVA-1.5-7B (Liu et al., 2024a). We use eight VQA benchmarks (e.g., TextVQA (Singh et al., 2019) and MME (Fu et al., 2025a)), the eight splits of RefCOCO, RefCOCO+ (Yu et al., 2016) and RefCOCOg (Mao et al., 2016) for visual grounding, and three video benchmarks on Qwen3-VL-8B (e.g., Video-MME (Fu et al., 2025b)). All experiments run on NVIDIA RTX 3090 GPUs. Detailed configurations are given in Appendix C.

Table 1: VQA performance comparison on Qwen2.5-VL-7B. **Bold**: Best. Underline: Second best.

Method	TextVQA	ChartQA	AI2D	OCRBench	HallBench	MME	MMB-EN	MMB-CN	Acc.	Rel.
Upper Bound, All 1,296 Tokens (100%)										
Qwen2.5-VL-7B	84.3	87.1	81.6	867	56.2	2342.3	83.1	82.9	84.9	100.0%
256 Tokens on Average (↓80.2%)										
VisionZip (CVPR25)	76.1	77.2	79.3	722	51.3	2277.2	80.6	81.3	79.0	92.9%
DivPrune (CVPR25)	75.3	69.2	77.1	666	46.8	2182.9	79.7	79.2	75.4	88.5%
CDPruner (NeurIPS25)	77.4	60.5	79.6	669	50.5	2258.7	79.9	80.1	76.0	89.4%
HiPrune (ACL26)	77.9	80.4	79.7	738	51.0	2274.5	81.0	**82.0**	79.9	94.0%
AnchorPrune (ECCV26)	80.5	82.7	**80.4**	804	51.4	2290.4	81.1	81.8	81.6	95.9%
EADP (ECCV26)	68.7	68.8	76.6	702	48.7	2217.3	80.7	79.1	75.5	88.6%
FastV (ECCV24)	78.8	66.0	75.0	631	48.2	2130.7	77.1	76.9	74.0	87.0%
SparseVLM (ICML25)	75.7	52.3	77.8	587	48.5	2255.0	80.5	79.9	73.3	86.1%
DART (EMNLP25)	70.5	63.6	74.8	623	44.6	2197.2	79.3	78.4	72.9	85.4%
HAWK (CVPR26)	**82.7**	81.7	79.7	810	51.1	2317.6	81.7	81.4	81.9	96.2%
VScan (TMLR26)	81.2	79.8	74.6	740	49.5	2235.5	77.0	77.9	78.2	91.9%
Nüwa (ICLR26)	79.5	81.6	79.9	751	50.4	2286.6	81.0	**82.0**	80.5	94.5%
CAR (Ours)	81.8	**83.9**	80.2	**831**	**53.4**	**2335.7**	**82.3**	**82.0**	**82.9**	**97.5%**
128 Tokens on Average (↓90.1%)										
VisionZip (CVPR25)	66.4	63.4	76.3	592	44.1	2184.1	77.8	77.9	71.8	84.1%
DivPrune (CVPR25)	65.9	51.4	72.7	535	43.2	2044.6	77.3	76.7	67.9	79.7%
CDPruner (NeurIPS25)	72.5	46.9	77.3	581	47.0	2181.6	79.2	77.6	71.0	83.4%
HiPrune (ACL26)	68.6	66.9	77.1	629	47.1	2192.7	79.1	78.5	73.7	86.6%
AnchorPrune (ECCV26)	75.9	71.1	78.2	692	48.2	2230.9	79.8	80.3	76.8	90.2%
EADP (ECCV26)	60.1	54.5	73.2	597	42.8	2152.9	78.8	77.3	69.3	81.1%
FastV (ECCV24)	39.3	24.1	67.7	310	37.8	1395.8	41.3	40.7	44.0	52.3%
SparseVLM (ICML25)	60.4	28.0	72.0	363	41.9	1969.7	72.0	71.2	60.0	70.6%
DART (EMNLP25)	56.5	47.0	69.8	512	40.5	2046.3	77.2	76.0	65.1	76.2%
HAWK (CVPR26)	**79.9**	71.3	77.5	705	48.8	2225.6	79.0	78.6	77.1	90.6%
VScan (TMLR26)	74.4	65.6	72.9	605	46.3	2138.3	75.5	75.9	72.3	84.8%
Nüwa (ICLR26)	71.3	70.6	77.4	611	**49.2**	2247.8	78.2	79.4	74.9	88.0%
CAR (Ours)	78.8	**76.1**	**79.4**	**731**	49.0	**2314.3**	**81.1**	**81.3**	**79.3**	**93.0%**

Table 2: VG performance comparison on Qwen2.5-VL-7B. **Bold**: Best. Underline: Second best.

Method	RefCOCO			RefCOCO+			RefCOCOg		Avg.	Rel.
	testA	testB	val	testA	testB	val	test	val		
Upper Bound, All Visual Tokens (100%)										
Qwen2.5-VL-7B	92.2	84.8	89.4	87.6	74.1	82.8	86.7	86.7	85.5	100.0%
Retain 25% of Visual Tokens										
Random	63.9	52.1	58.3	58.1	44.5	52.5	52.7	52.7	54.4	63.4%
Pooling	64.1	49.1	57.5	58.8	42.0	51.6	50.7	50.3	53.0	61.8%
VisionZip (CVPR25)	71.0	59.0	66.7	65.0	50.5	59.1	61.2	61.2	61.7	72.0%
DivPrune (CVPR25)	74.1	59.8	67.5	68.6	51.9	60.9	63.3	63.7	63.7	74.3%
AnchorPrune (ECCV26)	75.6	62.4	68.3	68.4	53.2	61.5	64.7	65.5	64.9	75.8%
EADP (ECCV26)	71.9	53.1	63.3	68.0	44.4	57.2	56.9	56.3	58.9	68.6%
FastV (ECCV24)	23.7	30.7	28.4	21.6	25.6	24.1	29.1	29.2	26.6	31.2%
SparseVLM (ICML25)	30.9	31.7	31.3	26.4	25.3	24.9	29.3	29.9	28.7	33.6%
HAWK (CVPR26)	40.4	45.7	42.7	36.0	40.0	38.0	44.9	45.7	41.7	48.9%
VScan (TMLR26)	78.7	66.9	71.7	72.3	55.5	64.0	67.9	68.2	68.2	79.5%
Nüwa (ICLR26)	77.5	72.5	75.6	69.7	60.3	65.8	72.8	72.3	70.8	82.7%
CAR (Ours)	**87.0**	**80.1**	**84.3**	**80.2**	**68.3**	**75.9**	**79.4**	**79.6**	**79.4**	**92.7%**
Retain 9.9% of Visual Tokens										
Random	23.1	17.2	21.5	20.3	14.5	18.6	17.3	18.1	18.8	21.9%
Pooling	20.0	15.6	17.5	18.2	13.0	15.5	14.8	14.2	16.1	18.8%
VisionZip (CVPR25)	23.3	19.2	20.5	18.8	16.0	17.7	18.4	19.5	19.2	22.4%
DivPrune (CVPR25)	27.8	19.6	24.2	24.7	15.8	20.9	21.3	21.5	22.0	25.6%
AnchorPrune (ECCV26)	24.4	18.0	22.0	21.4	16.2	19.4	19.7	20.2	20.2	23.5%
EADP (ECCV26)	24.1	16.5	21.2	23.4	14.6	18.6	17.4	18.2	19.2	22.4%
FastV (ECCV24)	3.2	2.7	3.2	3.3	2.9	3.5	3.6	3.3	3.2	3.8%
SparseVLM (ICML25)	5.7	5.7	5.8	3.8	4.2	3.8	5.4	5.3	5.0	5.8%
HAWK (CVPR26)	8.1	8.6	7.5	6.9	7.0	7.0	8.5	7.5	7.6	8.9%
VScan (TMLR26)	23.8	19.8	21.4	20.4	15.5	18.0	18.5	20.5	19.7	23.0%
Nüwa (ICLR26)	27.5	23.7	26.2	21.8	17.4	20.5	23.0	22.0	22.8	26.5%
CAR (Ours)	**75.4**	**66.7**	**72.2**	**67.2**	**55.9**	**62.3**	**67.8**	**66.5**	**66.8**	**77.9%**

3.1 Main Results

Performance on VQA tasks. Table 1 compares CAR with representative baselines on Qwen2.5-VL-7B. CAR achieves the best average accuracy at both retention ratios and leads on most benchmarks. With 80% and 90% of the visual tokens pruned, it retains 97.5% and 93.0% of the unpruned performance, against 96.2% and 90.6% for the strongest baseline, HAWK. The best-performing baseline varies across benchmarks (HAWK on TextVQA, AnchorPrune on AI2D and Nüwa on HallusionBench), whereas CAR ranks first or second on every benchmark at both ratios.

Performance on visual grounding tasks. Visual grounding relies on fine-grained spatial information and suffers from severe degradation under aggressive pruning. As shown in Table 2, with 75% of the visual tokens pruned, CAR retains 92.7% of the unpruned performance, against 82.7% for the strongest baseline, Nüwa. With 90% pruned, all baselines collapse to at most 26.5%, close to random sampling (21.9%). At both ratios the decoder-side methods (FastV, SparseVLM and HAWK) even fall below random sampling: they keep the top-k tokens ranked by the attention from the text, so the

Table 3: Average performance comparison across backbones and tasks.

	VQA				Visual Grounding					Video	
	Qwen3-VL-8B		InternVL3.5-8B		Qwen3-VL-8B		InternVL3.5-8B		LLaVA-1.5-7B	Qwen3-VL-8B	
Method	22.2%	11.1%	22.2%	11.1%	25%	9.9%	25%	9.9%	11.1%	25%	9.9%
Unpruned	85.9	85.9	83.0	83.0	85.0	85.0	89.4	89.4	51.3	65.5	65.5
VisionZip (CVPR25)	75.1	59.9	66.1	62.8	64.0	25.5	78.5	70.8	8.0	63.2	59.5
DivPrune (CVPR25)	72.5	62.0	73.3	69.1	58.7	25.4	80.1	73.3	17.3	61.7	58.1
AnchorPrune (ECCV26)	76.5	62.2	71.0	66.5	65.2	26.0	78.9	71.9	15.1	63.4	60.4
EADP (ECCV26)	75.1	66.6	73.6	69.5	62.0	22.4	82.0	75.0	20.0	63.3	59.8
GSTEP (MM26)	–	–	–	–	–	–	–	–	–	62.7	60.9
Nüwa (ICLR26)	75.9	64.7	63.4	58.2	64.5	28.9	72.9	62.7	28.2	63.5	61.6
CAR (Ours)	**79.7**	**70.6**	**79.1**	**74.4**	**72.2**	**46.6**	**83.9**	**76.2**	**30.4**	**64.0**	**62.0**

Table 4: Efficiency on Qwen2.5-VL-7B at 128 of 1,296 visual tokens.

Method	**Prefill (ms)**	**Decode (ms/tok)**	**Latency (ms)**	**FLOPs (T)**	**KV (MB)**	**Mem. (GB)**
Qwen2.5-VL-7B	484.0	27.1	543.3	10.28	73.2	16.02
128 Tokens on Average (↓90.1%)						
AnchorPrune (ECCV26)	262.0 (1.85×)	27.3	319.8 (1.70×)	3.81 (2.70×)	9.4	15.78
EADP (ECCV26)	254.8 (1.90×)	27.1	312.1 (1.74×)	3.81 (2.70×)	9.4	16.34
FastV (ECCV24)	222.6 (2.17×)	27.1	280.1 (1.94×)	3.83 (2.69×)	9.3	15.73
SparseVLM (ICML25)	241.5 (2.00×)	27.1	300.0 (1.81×)	3.85 (2.67×)	9.6	16.44
HAWK (CVPR26)	231.8 (2.09×)	27.1	288.1 (1.89×)	3.81 (2.70×)	9.4	16.34
VScan (TMLR26)	284.5 (1.70×)	27.3	358.0 (1.52×)	3.80 (2.70×)	9.3	19.81
Nüwa (ICLR26)	232.1 (2.09×)	27.0	287.6 (1.89×)	3.81 (2.70×)	9.4	16.36
CAR (Ours)	246.6 (1.96×)	27.3	304.7 (1.78×)	3.81 (2.70×)	9.4	15.82

retained tokens cluster on a few regions and leave much of the image uncovered, which breaks the spatial structure that grounding relies on. Our CAR significantly outperforms other methods, retaining 77.9% of the original model's performance. This lead does not come from the bias alone: with the same bias added to every method, CAR still leads the strongest one by 17.7 points (Table 30).

Generalization across backbones and tasks. To verify the generality and effectiveness of CAR, we evaluate it on three additional MLLMs, Qwen3-VL-8B with DeepStack, InternVL3.5-8B with dynamic tiling, and LLaVA-1.5-7B, across VQA, VG, and video understanding (Table 3). CAR performs best across varying backbones, tasks and retention ratios. On VQA it leads the strongest baseline at both retention ratios on Qwen3-VL and InternVL3.5. On grounding its advantage is largest at the aggressive ratio on Qwen3-VL, the setting where pruning costs the most. On video it also leads every baseline, including GSTEP, which is designed for video. Per-benchmark results are in Appendix C.

Efficiency analysis. Table 4 evaluates efficiency on TextVQA with Qwen2.5-VL-7B, keeping 128 of the 1,296 visual tokens, with batch size 1. Compared with the unpruned model, CAR reduces the prefill time by 1.96× and the end-to-end latency by 1.78×, and shrinks the KV cache by 7.8×. CAR reaches the highest accuracy at a cost comparable to the other pruning methods.

3.2 Ablation

Ablation on components. To assess the contribution of each component, Table 5 starts from the unweighted coverage selection ($w_i = 1$) and adds one component at a time, with random selection as a reference. On grounding the bias is decisive, raising accuracy by 29.9 points, while on VQA it is worth 1.5. Selection behaves the other way: unweighted coverage alone gains 8.0 points compared with random selection on VQA, and weighting it by the encoder saliency averaged over the heads adds 4.8. Selecting the head with the probe improves both tasks, and the spatial kernel adds 4.3 points on grounding but costs 0.2 on VQA, so we apply it to grounding only. The two tasks therefore rely on different components, one for each error term of Eq. 5. VG averages the eight RefCOCO splits and VQA the eight benchmarks; the per-split and per-benchmark results are in Appendix C.9.

Ablation on selection rule. Table 6 compares ten representative selection rules that share the saliency, retention ratio, bias and features of CAR and differ only in how tokens are selected (Appendix C.11); VG is averaged over three of the eight RefCOCO splits and VQA over the eight benchmarks. Averaged over the two tasks, the coverage-based rules, CAR ⑩ and SCOPE ⑨, outperform the other rules, and by also accounting for spatial distance in the similarity, CAR performs best on grounding. Combining the saliency with the diversity is better than either alone: diversity alone ② drops task-relevant tokens, and saliency alone ① keeps redundant ones. The per-split and per-benchmark results are given in Appendix C.11.

Table 5: Component ablation.

Configuration	VG	VQA
Unpruned	85.5	84.9
Random	18.8	64.2
+ bias	38.6	64.5
Coverage, w_i=1	20.1	72.2
+ bias	50.0	73.7
+ saliency w (head avg.)	54.8	78.5
+ head h^*	62.5	**79.3**
+ spatial kernel	**66.8**	79.1

Table 6: Selection-rule ablation.

	Score or objective	Representative	VG	VQA
①	w_i	FastV	58.6	78.1
②	$\min_{j\in S}(1-s_{ij})$, w unused	DivPrune	44.4	67.8
③	$\lambda w_i - (1-\lambda)\max_{j\in S} s_{ij}$	IDPruner	59.1	78.4
④	$w_i^{\alpha}\min_{j\in S}(1-s_{ij})$	AnchorPrune	59.6	78.6
⑤	$\log\det K_S$, $K_{ij}=w_i w_j s_{ij}$	CDPruner	59.5	78.1
⑥	$(1-\alpha)w_i + \alpha\,\Delta_i \log\det(X_S X_S^\top + \lambda I)$	CoIn	52.5	74.3
⑦	w_i subject to $\max_{j\in S} s_{ij} < \tau$	AgilePruner	60.5	76.1
⑧	Pivots by w_i, then graph suppression	D^2Pruner	61.4	77.1
⑨	$w_i \sum_{l\in V}[s_{il} - \max_{j\in S} s_{jl}]_+$	SCOPE	62.1	78.4
⑩	$\sum_{i\in V} w_i \max_{j\in S} s_{ij}$ (Ours)	—	**66.4**	**78.7**

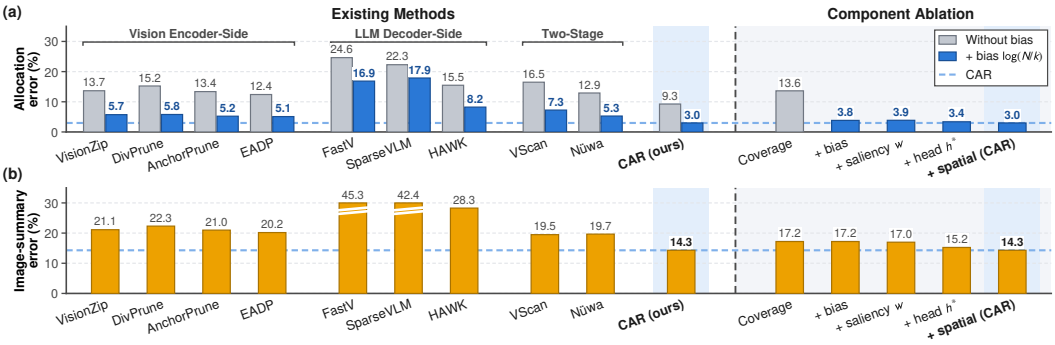


Figure 3: **The two errors of Eq. 5 on grounding** for existing methods and the ablation of Table 5 (Qwen2.5-VL-7B, 9.9%). Each bar is $\sum\|\mathcal{E}\|^2 / \sum\|O\|^2$ (%) over layers 13–19; lower is better.

Table 7: Bias-form ablation.

Bias b_j	VG	VQA
None	26.8	78.7
$\log(N/k)$	**66.8**	78.3
$\log(1+\lvert C_j\rvert)$	65.5	79.2
$\min\{\log(1+\lvert C_j\rvert), \log(N/k)\}$	66.3	**79.3**

Ablation on bias form. Table 7 fixes the retained tokens and compares the three bias forms of Section 2.3. On grounding, all forms lift accuracy from 26.8 to above 65, and the uniform bias, which preserves the attention across image regions, performs best. On VQA, the per-cell and capped forms perform better. This matches the location-sensitive nature of grounding and the content-centric nature of VQA, so we use the uniform bias for grounding and the capped bias for VQA. The per-split and per-benchmark results are in Appendix C.10.

Error decomposition. Figure 3 measures the two errors of Eq. 5 on grounding; the measurement protocol is described in Appendix C.12. Compared with existing methods, CAR has both the lowest image summary error and the lowest allocation error. The bias lowers the allocation error of each method by restoring the visual attention share. In the component ablation, the bias lowers the allocation error from 13.6 to 3.8 and leaves the summary error unchanged, whereas the selection components lower the summary error from 17.2 to 14.3 with little change in the allocation error, so each component acts on the error term it targets.

4 Conclusion

Under aggressive pruning, existing visual token pruning methods degrade substantially, especially on the grounding task. We analyze pruning through the shift it causes in the decoder output, and decompose this shift exactly into an image summary error and an attention allocation error, which correspond to the information loss and the attention distortion caused by pruning. Based on this decomposition, we propose CAR, which reduces the image summary error with weighted coverage selection guided by a head probe, and the attention allocation error with a logit bias on the retained visual tokens that restores the attention share the image held before pruning. Since the bias does not change which tokens are retained, it can also be applied directly to existing methods. Across four backbones and VQA, grounding, and video understanding, CAR consistently achieves state-of-the-art performance.

Reproducibility statement

The code that produces the results of this paper is provided as anonymous supplementary material. The proofs and their assumptions are given in Appendix B, the packages, weights and settings shared by all experiments in Appendix C.2, the protocol of the head probe in Appendix C.3, and the configuration of each backbone in Appendices C.4 to C.8.

AI use statement

In this work, we used generative AI tools to assist in proving the mathematical claims in Appendix. Additionally, we used generative AI tools to aid and polish the writing of the paper, and for retrieval and discovery, such as finding related work. We reviewed all AI-assisted content: every proof was checked by the authors, the text was reviewed and edited by the authors, and every retrieved reference was verified against its original publication. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

References

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

Changwoo Baek, Jouwon Song, Sohyeon Kim, and Kyeongbo Kong. AgilePruner: An empirical study of attention and diversity for adaptive visual token pruning in large vision-language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

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

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 335–336, 1998.

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.

Zengjie Chen, Yuxiang Cai, Jingcai Guo, Taotao Cai, Jianwei Yin, and Zhi Chen. Accelerating multimodal large language models with prior-corrected token reduction. In *European Conference on Computer Vision*, 2026.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b.

Hyeonwoo Cho, Donghyeon Baek, Yewon Kim, and Bumsub Ham. Improving visual token reduction via rectifying distortions for efficient multimodal LLM inference. In *International Conference on Machine Learning*, 2026.

Yunkai Dang, Yizhu Jiang, Yifan Jiang, Qi Fan, Yinghuan Shi, Wenbin Li, and Yang Gao. CLASP: Class-adaptive layer fusion and dual-stage pruning for multimodal large language models. *arXiv preprint arXiv:2604.12767*, 2026.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, 2024.

Jinhong Deng, Wen Li, Joey Tianyi Zhou, and Yang He. SCOPE: Saliency-coverage oriented token pruning for efficient multimodal LLMs. In *Advances in Neural Information Processing Systems*, 2025.

Chenxi Du, Yongheng Deng, Jiani Liu, Yujia Zhang, Xi Chen, and Ju Ren. CoIn: Coverage and informativeness-guided token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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. In *Advances in Neural Information Processing Systems*, volume 38, pp. 162549–162567, 2025a.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025b.

Tianxiao Gao, Shanwei Zhao, Shuo Fang, Shiai Zhu, and Chenguang Ma. QuietPrune: Query-guided early token pruning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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

Yihong Huang, Fei Ma, Yihua Shao, Jingcai Guo, Zitong Yu, Laizhong Cui, and Qi Tian. Nüwa: Mending the spatial integrity torn by VLM token pruning. In *International Conference on Learning Representations*, 2026.

Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pp. 722–754. PMLR, 2021.

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

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *International Conference on Learning Representations*, 2025.

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

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024.

Yuxuan Liang, Xu Li, Xiaolei Chen, Haotian Chen, Yi Zheng, Zhe Liu, Rui Zhu, Bin Li, and Xiangyang Xue. Pyramid token pruning for high-resolution large vision-language models via region, token, and instruction-guided importance. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(6):8372–8384, 2026.

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.

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 2024b. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

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, 2026a.

Shi Liu, Kecheng Zheng, and Wei Chen. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision*, pp. 125–140, 2024c.

Xuyang Liu, Ziming Wang, Junjie Chen, Yuhang Han, Yingyao Wang, Jiale Yuan, Jun Song, Siteng Huang, and Honggang Chen. Global compression commander: Plug-and-play inference acceleration for high-resolution large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026b.

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, 2024d.

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, 2024e.

Yuan-Hsi Lo, Hsi-Ren Hung, and Huei-Fang Yang. HSSP: Training-free visual token pruning with head selection and spatial constraints. In *Proceedings of the IEEE International Conference on Image Processing*, 2026.

Jiayun Luo, Wan-Cyuan Chris Fan, Lyuyang Wang, Xiangteng He, Tanzila Rahman, Purang Abolmaesumi, and Leonid Sigal. To sink or not to sink: Visual information pathways in large vision-language models. In *International Conference on Learning Representations*, 2026.

Haifeng Ma, Mingyue Guo, Linhui Xiao, Qingfang Zheng, and Qingming Huang. SHARP: Semantic head-aware representation pruning for efficient MLLMs. In *Proceedings of the International Conference on Multimedia Retrieval*, 2026.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

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

Yu Meng, Kaiyuan Li, Chenran Huang, Chen Gao, Xinlei Chen, Yong Li, and Xiaoping Zhang. PLPHP: Per-layer per-head vision token pruning for efficient large vision-language models. *arXiv preprint arXiv:2502.14504*, 2025.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

Kyuan Oh and Bumsoo Kim. AnchorPrune: Relevance-anchored contextual expansion for visual token pruning. In *European Conference on Computer Vision*, 2026.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 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.

Guohao Sun, Yufei Wang, Sizhuo Ma, Yuege Xie, Yuting Cheng, Zhiqiang Tao, and Jian Wang. IF-Prune: Information-flow guided token pruning for efficient vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Zhichao Sun, Yidong Ma, Gang Liu, Yibo Chen, Xu Tang, Yao Hu, and Yongchao Xu. IVC-Prune: Revealing the implicit visual coordinates in LVLMs for vision token pruning. In *International Conference on Learning Representations*, 2026b.

Yifan Tan, Yifu Sun, Shirui Huang, Hong Liu, Guanghua Yu, Jianchen Zhu, and Yangdong Deng. IDPruner: Harmonizing importance and diversity in visual token pruning for MLLMs. *arXiv preprint arXiv:2602.13315*, 2026.

Hengyi Wang, Weiying Xie, Hui Jiang, Yaotao Wei, Kai Jiang, Mingxiang Cao, Chenhe Hao, and Leyuan Fang. TOP-RL: Task-optimized progressive token pruning with reinforcement learning for vision language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15824–15832, 2026a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Xuehui Wang, Xuankun Yang, and Wei Shen. Combating textual noise and redundancy: Entropy-aware dense visual token pruning. In *European Conference on Computer Vision*, 2026b.

Yuhao Wang, Mu Qiao, Haiwen Diao, Yunzhi Zhuge, Pingping Zhang, Xindong Zhang, Lei Zhang, and Huchuan Lu. ERA: Entropy-guided visual token pruning with rectified attention for efficient MLLMs. *arXiv preprint arXiv:2606.31982*, 2026c.

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. 9961–9980, 2025.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14593–14603, 2025.

Tong Xu, Hailong Shi, and Xingyu Gao. SCoRe: Salience-coverage reduction for vision token pruning in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Cheng-Yu Yang, Shao-Yuan Lo, and Yu-Lun Liu. Reroute, don't remove: Recoverable visual token routing for vision-language models. *arXiv preprint arXiv:2606.12412*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. FlashInfer: Efficient and customizable attention engine for LLM inference serving. In *Proceedings of Machine Learning and Systems*, 2025.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pp. 69–85. Springer, 2016.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. VScan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research*, 2026a.

Evelyn Zhang, Fufu Yu, Aoqi Wu, Zichen Wen, Ke Yan, Shouhong Ding, Biqing Qi, and Linfeng Zhang. D^2Pruner: Debiased importance and structural diversity for MLLM token pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12412–12420, 2026b.

Mengjie Zhang, Qihui Zhu, Tao Zhang, Shuangwu Chen, Huihuang Qin, Yu Guo, Shenghao Ye, Zijian Wen, Yunpeng Hou, Dong Jin, et al. GSTEP: Global spatio-temporal density-driven visual token pruning for efficient video large language models. In *Proceedings of the ACM International Conference on Multimedia*, 2026c.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

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. In *International Conference on Machine Learning*, 2025b.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19781–19791, 2025c.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Qihui Zhu, Tao Zhang, Yuchen Wang, Shuangwu Chen, Xiaobin Tan, Jian Yang, Yang Liu, and Yinfei Pan. HAWK: Head importance-aware visual token pruning in multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39583–39592, 2026.

APPENDIX

A RELATED WORK

A.1 VISUAL TOKEN PRUNING

Visual token pruning shortens the visual sequence of a multimodal model, and existing approaches can be classified along several dimensions. Based on their training requirements, token reduction methods fall into *training-based* and *training-free* approaches. Training-based methods, such as TOP-RL (Wang et al., 2026a), IF-Prune (Sun et al., 2026a) and QuietPrune (Gao et al., 2026), adapt the model to the shortened sequence through gradient updates. Training-free methods operate at inference time without modifying model parameters, which makes them directly applicable to off-the-shelf models; our comparison focuses on training-free methods. Regarding the stage at which pruning is applied (or at which the pruning signal is computed), they are *encoder-side*, such as VisionZip (Yang et al., 2025), DivPrune (Alvar et al., 2025), CDPruner (Zhang et al., 2025a), HiPrune (Liu et al., 2026a), AnchorPrune (Oh & Kim, 2026), EADP (Wang et al., 2026b), SCOPE (Deng et al., 2025), CoIn (Du et al., 2026), SCoRe (Xu et al., 2026) and SHARP (Ma et al., 2026); *decoder-side*, such as FastV (Chen et al., 2024a), SparseVLM (Zhang et al., 2025b), PDrop (Xing et al., 2025), HAWK (Zhu et al., 2026), PriorTR (Chen et al., 2026), PLPHP (Meng et al., 2025) and HSSP (Lo et al., 2026), D^2Pruner (Zhang et al., 2026b); or *two-stage*, such as VScan (Zhang et al., 2026a), Nüwa (Huang et al., 2026) and CLASP (Dang et al., 2026). Each method further involves two decisions: (1) *identifying which tokens to retain*, based on their *importance* (FastV, SparseVLM, VisionZip, HAWK), their *mutual similarity* (DivPrune, DART (Wen et al., 2025)), their *spatial position* (Nüwa, IVC-Prune (Sun et al., 2026b)), or a combination of these signals (CDPruner, AnchorPrune, D^2Pruner, IDPruner (Tan et al., 2026), EADP, SCOPE, CoIn, SCoRe); and (2) *handling the discarded tokens*, whose useful features can be aggregated into the retained ones, as in ToMe (Bolya et al., 2023), PruMerge+ (Shang et al., 2025), RESTORE (Cho et al., 2026) and ERA (Wang et al., 2026c), or which can be reused at a later stage, as in Reroute (Yang et al., 2026).

Encoder-side pruning. These methods select tokens before the language model. DivPrune selects a maximally diverse subset based on the similarity between visual features. VisionZip and HiPrune use the visual attention from the vision encoder as the importance signal, VisionZip keeping the dominant tokens and merging the remaining ones, and HiPrune organizing the attention hierarchically across encoder stages. CDPruner, AnchorPrune and EADP score tokens by their relevance to the instruction and balance it against diversity: CDPruner maximizes conditional diversity through a determinantal point process, AnchorPrune protects a relevance anchor and expands it with complementary context, and EADP denoises the relevance score with statistical entropy. The methods that rely on the encoder attention, however, largely overlook the varying importance of the attention heads, assuming that all heads contribute equally to interpreting visual information.

Decoder-side pruning. FastV pioneered the approach of performing token pruning within language models based on text–visual attention scores, providing guidance for subsequent studies such as SparseVLM and PyramidDrop. These methods progressively remove less informative tokens according to the attention scores. Since the text–visual attention is biased, PriorTR and D^2Pruner focus on rectifying the attention signal, PriorTR by separating the task-conditioned attention from a model-induced prior and D^2Pruner by debiasing the positional bias of the attention scores. DART relies purely on feature similarity: it selects a small set of pivot tokens and then removes the tokens that duplicate them. HAWK weights the text-guided attention by head importance estimated offline, by ablating each head and measuring the resulting accuracy drop on benchmark datasets.

Two-stage pruning. These methods combine a selection in the vision encoder with a second pruning step inside the language model. VScan merges tokens during visual encoding through complementary global and local scans and then prunes at intermediate decoder layers. Nüwa partitions the token grid and aggregates the discarded tokens into their spatial neighbours, so that the spatial reference frame on which grounding relies is preserved, and refines the selection with a text-guided step inside the decoder.

Table 8: Summary of notation used in the derivations of Section 2.

Symbol	Description	Symbol	Description
T, V, S	Text, visual and retained tokens	w_i	Saliency of token i
N, k	Number of visual tokens; number retained	s_{ij}	Similarity between tokens i and j
$z_u,\ e^{z_u}$	Attention logit of token u and its unnormalized mass	C_j	Tokens assigned to the retained token j
p_u, p'_u	Attention weight on u, before and after pruning	$b_j,\ m_j$	Logit bias on j; its mass factor $m_j = e^{b_j}$
σ, σ'	Share of attention on the image, before and after pruning	Z_T, Z_V	Total attention mass of the text and of the image
$\bar{v}_{\text{img}}, \bar{v}_S, \bar{v}_{\text{txt}}$	Summaries of V, of S and of the text	ρ	Fraction of the visual mass on the retained tokens

A.2 Attention Calibration

Compensating for the attention of removed tokens originates in token merging. ToMe (Bolya et al., 2023) merges similar tokens inside a vision transformer and averages their features; it then adds $\log s$ to the logit of the merged token, where s is the number of original tokens it represents, so that the merged token receives attention in proportion to them. RESTORE and ERA transfer this idea to multimodal models. RESTORE (Cho et al., 2026) is applied on top of an existing pruning method and splits its budget in half: one half is the tokens that method selects, and the other half is formed by merging the tokens it discards. The logit of each merged token is raised by the logarithm of the number of tokens it has absorbed, scaled by a distance factor that counteracts the decay of attention induced by RoPE. ERA (Wang et al., 2026c) selects its own anchors, recycles the pruned tokens into them by moving each anchor towards the centroid of its cluster, and raises its logit by the logarithm of the saliency-weighted size of that cluster. In all three the bias is attached to the tokens that absorbed the discarded ones, and its magnitude follows the number of tokens absorbed.

These calibrations are developed within a merging framework, so the features of the retained tokens are modified. Our bias instead only restores the visual attention share and does not change the features of the retained tokens. Our uniform form depends only on the retention ratio, so it can be applied on top of any pruning method. Our per-cell form recovers the same $\log s$ correction, with the assignment from the discarded tokens to the retained ones following naturally from our coverage selection.

Adjusting the attention that text pays to the image has also been studied without pruning. PAI (Liu et al., 2024c) amplifies the attention weights on image tokens at inference to counter text inertia, and VAR (Kang et al., 2025) redistributes the attention absorbed by visual sink tokens to the remaining visual tokens in image-centric heads. Unlike PAI and VAR, our bias has a defined target, the share the image held before pruning, and its magnitude follows from the retention ratio rather than a tuned factor.

B Derivations and Proofs

B.1 Notation and the Two-Group Form of the Attention Output

This section derives Eq. 4. The attention output O is an attention-weighted average of the value vectors of all visual and text tokens. We rewrite this average in two stages. The first stage averages the values inside each modality, under the attention renormalized within that modality, and returns one summary vector per modality. The second stage mixes the two summaries, weighting each by the attention mass its modality holds.

Notation. Throughout we fix one decoder layer, one attention head and one text query, and suppress the corresponding indices. Let T index the text tokens that precede the query and V the N visual tokens, with attention logits z_u and values v_u, and let $S \subseteq V$ with $|S| = k$ be the visual tokens retained by pruning. Write

$$Z_T = \sum_{t \in T} e^{z_t}, \qquad Z_V = \sum_{i \in V} e^{z_i}, \qquad Z_S = \sum_{j \in S} e^{z_j}, \qquad \rho = \frac{Z_S}{Z_V}, \tag{15}$$

so that $\rho \in (0, 1]$ is the fraction of the visual softmax mass carried by S.

Step 1 (split the sum at the modality boundary). Writing $Z = Z_T + Z_V$ for the full softmax denominator, Eq. 1 splits as

$$O = \sum_{u \in T \cup V} \frac{e^{z_u}}{Z} v_u = \sum_{i \in V} \frac{e^{z_i}}{Z} v_i \; + \; \sum_{t \in T} \frac{e^{z_t}}{Z} v_t. \tag{16}$$

Step 2 (write each weight as a group share times a within-group weight). A token's weight in the whole sequence factors into the share held by its group and its weight inside that group:

$$\frac{e^{z_i}}{Z} = \frac{Z_V}{Z} \cdot \frac{e^{z_i}}{Z_V}, \qquad \frac{e^{z_t}}{Z} = \frac{Z_T}{Z} \cdot \frac{e^{z_t}}{Z_T}. \tag{17}$$

The group mass does not depend on the summation index, so it leaves the sum:

$$O = \frac{Z_V}{Z} \underbrace{\sum_{i \in V} \frac{e^{z_i}}{Z_V} v_i}_{\bar{v}_{\text{img}}} \; + \; \frac{Z_T}{Z} \underbrace{\sum_{t \in T} \frac{e^{z_t}}{Z_T} v_t}_{\bar{v}_{\text{txt}}}. \tag{18}$$

Each brace is a convex combination of that group's values: its weights are non-negative and sum to one by construction ($\sum_{i \in V} e^{z_i}/Z_V = 1$, and likewise for T), so $\bar{v}_{\text{img}}$ and $\bar{v}_{\text{txt}}$ are exactly the attention-weighted mean values of the two groups.

Step 3 (identify the two prefactors). Since $Z = Z_T + Z_V$,

$$\frac{Z_V}{Z} = \sum_{i \in V} \frac{e^{z_i}}{Z} = \sum_{u \in V} p_u = \sigma, \qquad \frac{Z_T}{Z} = \frac{Z - Z_V}{Z} = 1 - \sigma, \tag{19}$$

and substituting into Step 2 gives the first line of Eq. 4,

$$O = \sigma \, \bar{v}_{\text{img}} + (1 - \sigma) \, \bar{v}_{\text{txt}}. \tag{20}$$

The same rewriting after pruning. Pruning restricts the sum to $T \cup S$ and adds the bias b_j to the visual logits, with $b_t = 0$ on the text side. The visual part of the denominator becomes $Z_S^b = \sum_{j \in S} e^{z_j + b_j}$, and the text part remains Z_T. The three steps go through unchanged and give the second line of Eq. 4,

$$O' = \sigma' \, \bar{v}_S + (1 - \sigma') \, \bar{v}_{\text{txt}}, \qquad \sigma' = \frac{Z_S^b}{Z_T + Z_S^b}, \qquad \bar{v}_S = \frac{1}{Z_S^b} \sum_{j \in S} e^{z_j + b_j} v_j. \tag{21}$$

The text summary $\bar{v}_{\text{txt}}$ is the same vector as in the first line, since its weights are e^{z_t}/Z_T and pruning changes neither Z_T nor the text logits.

B.2 The Coverage Bound

This section proves the bound quoted in Section 2.2: the image summary error is controlled by how well the retained set covers the whole token set, so that maximizing coverage minimizes an upper bound on the error.

Setup. Write $a_i = e^{z_i}/Z_V$ for the attention on visual token i renormalized within the image, as in Appendix B.1, so that $\sum_{i \in V} a_i = 1$ and $\sum_{j \in S} a_j = \rho$. Let $\pi(i) = \arg\max_{j \in S} \cos(v_i, v_j)$ for $i \notin S$ and $\pi(j) = j$ for $j \in S$, let $A_j = \sum_{i : \pi(i) = j} a_i$ be the attention mass of the tokens that j represents, so that $\sum_{j \in S} A_j = 1$, and write $c_i = \cos(v_i, v_{\pi(i)})$. The objective

$$F_v(S) = \sum_{i \in V} a_i \max_{j \in S} \cos(v_i, v_j) \tag{22}$$

is a facility-location objective: the retained tokens are the facilities, all N visual tokens are the clients, and each client is weighted by the attention it receives.

We restate Proposition 1: if the value vectors have equal norm R, and if each retained token carries the attention mass of the tokens it represents, that is, $\bar{v}_S = \sum_{j \in S} A_j v_j$, then

$$\left\| \bar{v}_S - \bar{v}_{\text{img}} \right\| \; \le \; \sqrt{2} \, R \, \sqrt{1 - \rho} \, \sqrt{1 - F_v(S)}. \tag{23}$$

Step 1 (replacement view). Splitting $\bar{v}_{\text{img}}$ into its retained and discarded parts and using the assumption on $\bar{v}_S$,

$$\bar{v}_S - \bar{v}_{\text{img}} = \sum_{j\in S} A_j v_j - \sum_{j\in S} a_j v_j - \sum_{i\notin S} a_i v_i \tag{24}$$

$$= \sum_{j\in S} \sum_{i\notin S:\, \pi(i)=j} a_i v_j - \sum_{i\notin S} a_i v_i \;=\; \sum_{i\notin S} a_i \big(v_{\pi(i)} - v_i\big), \tag{25}$$

so the error is the weighted sum of the replacements, one per discarded token.

Step 2 (triangle inequality). Since the weights are non-negative,

$$\big\|\bar{v}_S - \bar{v}_{\text{img}}\big\| \le \sum_{i\notin S} a_i \|v_{\pi(i)} - v_i\|. \tag{26}$$

Step 3 (equal norms turn distance into cosine). With $\|v_i\| = \|v_{\pi(i)}\| = R$,

$$\|v_i - v_{\pi(i)}\|^2 = 2R^2 - 2\langle v_i, v_{\pi(i)}\rangle = 2R^2(1-c_i), \qquad \text{hence} \qquad \|v_i - v_{\pi(i)}\| = \sqrt{2}\,R\sqrt{1-c_i}. \tag{27}$$

Step 4 (Cauchy–Schwarz). Splitting each weight as $a_i = \sqrt{a_i}\cdot\sqrt{a_i}$ and using $\sum_{i\notin S} a_i = 1-\rho$,

$$\sum_{i\notin S} a_i\sqrt{1-c_i} = \sum_{i\notin S} \sqrt{a_i}\cdot\sqrt{a_i(1-c_i)} \le \sqrt{1-\rho}\,\Big(\sum_{i\notin S} a_i(1-c_i)\Big)^{1/2}. \tag{28}$$

Step 5 (identify the coverage objective). A retained token covers itself with $\cos = 1$, and a discarded token is covered by $\pi(i)$, which is the $\arg\max$ of the same cosine that defines F_v, so

$$F_v(S) = \rho + \sum_{i\notin S} a_i c_i, \qquad \text{and therefore} \qquad \sum_{i\notin S} a_i(1-c_i) = (1-\rho) - \big(F_v(S) - \rho\big) = 1 - F_v(S). \tag{29}$$

Combining Steps 1 to 5 gives Eq. 23.

Tightness. The bound is loose: Step 2 is an equality only when the replacement vectors all point the same way, and Step 4 only when every discarded token is represented equally well. It therefore separates a good retained set from a bad one, but does not rank two sets that are close.

Scope. Since $\rho \ge 0$, the right-hand side is at most $\sqrt{2}R\sqrt{1-F_v(S)}$; the form quoted in Section 2.2 keeps the factor $\sqrt{1-\rho}$, which shows that a set keeping more attention mass tightens the bound as well. Equal norms enter only in Step 3. Without them, Step 3 gains the term $(\|v_i\| - \|v_{\pi(i)}\|)^2$, and the same steps give the bound with R replaced by $R_{\max} = \max_u \|v_u\|$ plus an additive term $(1-\rho)(R_{\max} - R_{\min})$, which vanishes when the norms are equal. In practice, the selector computes this objective from encoder features and encoder saliency, since the decoder values and attention shares are not available before the tokens enter the decoder. The assumption on $\bar{v}_S$ is exactly what the per-cell bias realizes; with the uniform bias, it holds only approximately.

B.3 Submodularity of the Selection Objective

This section proves that the objective of Eq. 7 is monotone and submodular, so that the greedy selection of Eq. 10 comes with the classical $1 - 1/e$ guarantee.

Setup. Let $\phi(i,j) \ge 0$ be any non-negative similarity between tokens, let $w_i \ge 0$ be any weights, and set

$$F(S) = \sum_{i=1}^{N} w_i\, m_i(S), \qquad m_i(S) = \max_{j\in S} \phi(i,j), \qquad m_i(\emptyset) = 0. \tag{30}$$

Monotonicity. A set function is monotone if $F(A) \le F(B)$ whenever $A \subseteq B$. For any such A and B, the set $\{\phi(i,j) : j \in A\}$ is contained in $\{\phi(i,j) : j \in B\}$, and a maximum over a larger set is no smaller, so $m_i(A) \le m_i(B)$ for every i. Summing with non-negative weights preserves the inequality, hence $F(A) \le F(B)$.

Submodularity. A set function is submodular if it has diminishing returns, that is, $F(A \cup \{u\}) - F(A) \ge F(B \cup \{u\}) - F(B)$ for every $A \subseteq B$ and every $u \notin B$. The gain of a single client is

$$m_i(S \cup \{u\}) - m_i(S) = \big(\phi(i,u) - m_i(S)\big)_+. \tag{31}$$

Monotonicity gives $m_i(A) \le m_i(B)$, and $(\cdot)_+$ is non-decreasing, so $\big(\phi(i,u) - m_i(A)\big)_+ \ge \big(\phi(i,u) - m_i(B)\big)_+$: the more a client is already covered, the less a new facility can add. Summing over i with non-negative weights gives the claim.

Our objective. Both similarities we use are non-negative: $[\cos(x_i, x_j)]_+$ of Eq. 7, and its product with the positive Gaussian kernel of Eq. 8. With w_i the encoder saliency, both objectives are instances of Eq. 30.

Approximation guarantee. Maximizing a monotone submodular function under a cardinality constraint is NP-hard, but greedy selection is near-optimal: writing S_{greedy} for the set it returns and S_{opt} for the optimum of the same size,

$$F(S_{\text{greedy}}) \;\ge\; \Big(1 - \tfrac{1}{e}\Big) F(S_{\text{opt}}) \;\approx\; 0.632\, F(S_{\text{opt}}) \tag{32}$$

(Nemhauser et al., 1978). The step of Eq. 10 adds the token with the largest $\sum_i w_i \big(\phi(i,j) - m_i(S)\big)_+$, which by Eq. 31 is exactly the marginal gain $F(S \cup \{j\}) - F(S)$ the theorem requires, so the guarantee applies to the selection we run.

B.4 Properties of the Uniform Bias

This section establishes the properties of the uniform bias of Section 2.3.

A uniform bias does not move the visual summary. By Appendix B.1, the image summary after pruning $\bar{v}_S$ is the average of the retained values under the weights renormalized inside the image,

$$\bar{v}_S = \sum_{j \in S} \frac{e^{z_j + b_j}}{\sum_{j' \in S} e^{z_{j'} + b_{j'}}}\, v_j. \tag{33}$$

With the same $b_j = b$ on every $j \in S$, the factor e^b cancels between the numerator and the denominator,

$$\frac{e^{z_j + b}}{\sum_{j' \in S} e^{z_{j'} + b}} = \frac{e^b e^{z_j}}{e^b Z_S} = \frac{e^{z_j}}{Z_S}, \tag{34}$$

so every weight, and therefore $\bar{v}_S$, is independent of b. A uniform bias changes σ' alone and acts on $\mathcal{E}_{\text{alloc}}$ alone.

Remark on the substitution $\mathbb{E}[\rho] = k/N$. For a given query the exact constant is $b^\star = -\log \rho$, and ρ varies from query to query. We cannot evaluate it, so we substitute $\mathbb{E}[\rho] = k/N$ and use $b = \log(N/k)$ for every query. The restored mass is $e^b \rho = \rho N/k$, which equals the exact value 1 in expectation. The share, however, is a concave function of this mass,

$$\sigma' = \frac{x}{Z_T/Z_V + x}, \qquad x = e^b \rho, \tag{35}$$

so by Jensen's inequality $\mathbb{E}[\sigma'] \le \sigma'(\mathbb{E}[x]) = \sigma$: the bias restores the share less than the exact solution does on average. The gap is zero when ρ is the same for every query and grows with its variance.

Remark on the identity $\mathbb{E}[\rho] = k/N$. The identity assumes that the visual tokens are exchangeable, which the selector does not respect, since it retains the tokens that carry the most saliency. Where the retained set holds more than k/N of the visual attention, the exact constant $-\log \rho$ is smaller than the $\log(N/k)$ we apply, so the bias restores more share than pruning removed. The convexity of $-\log$ and the failure of exchangeability act in opposite directions, and which one dominates changes with the layer. The share measured in Appendix C.10 exceeds the unpruned level in the early layers and falls below it in the later ones.

B.5 The Per-Cell Bias

This section derives the per-cell bias of Section 2.3.

Setup. We keep the notation of Appendix B.2: $a_i = e^{z_i}/Z_V$, $\pi(i) = \arg\max_{j \in S} \cos(v_i, v_j)$ and $A_j = \sum_{i:\pi(i)=j} a_i$. The selector assigns each discarded token to the retained token most similar to it, and those assigned to j form its cell C_j. The bound defines π by the cosine between value vectors, while the selector assigns by the cosine between encoder features, and we use C_j as an approximation of $\pi^{-1}(j)$ throughout.

The per-cell constraint. The mass condition of Eq. 11 is one equation in k unknowns. We impose the k constraints that give each retained token the mass of the tokens it represents,

$$e^{z_j} m_j \;=\; e^{z_j} + \sum_{i \in C_j} e^{z_i}, \qquad j \in S, \tag{36}$$

where the left-hand side is the mass of j after the bias and the right-hand side is the mass of j together with that of its cell. The k constraints determine every unknown, $m_j = 1 + \big(\sum_{i \in C_j} e^{z_i}\big)/e^{z_j}$.

The constraint solves the mass condition. Summing Eq. 36 over $j \in S$ gives

$$\sum_{j \in S} e^{z_j} m_j = \sum_{j \in S} e^{z_j} + \sum_{j \in S} \sum_{i \in C_j} e^{z_i} = Z_V, \tag{37}$$

where the first sum is the mass of the retained tokens and the second is the mass of the discarded ones, since the cells partition them. Together they are the mass of all N visual tokens, so Eq. 11 holds.

The weights are the ones the bound assumes. Proposition 1 holds under the hypothesis that each retained token carries the attention of the tokens it represents, that is, $\bar{v}_S = \sum_{j \in S} A_j v_j$. The per-cell constraint produces exactly these weights. The biased masses sum to Z_V, so the visual weight of a retained token is

$$\frac{e^{z_j} m_j}{\sum_{j' \in S} e^{z_{j'}} m_{j'}} = \frac{e^{z_j} + \sum_{i \in C_j} e^{z_i}}{Z_V} = a_j + \sum_{i \in C_j} a_i = A_j. \tag{38}$$

The per-cell bias therefore makes the hypothesis of the bound hold.

Counting inside a cell. The sum $\sum_{i \in C_j} e^{z_i}$ on the right-hand side of Eq. 36 is not available at run time. Setting $e^{z_i} \approx e^{z_j}$ for $i \in C_j$ turns the ratio into a count,

$$m_j = 1 + \frac{\sum_{i \in C_j} e^{z_i}}{e^{z_j}} \;\approx\; 1 + |C_j|. \tag{39}$$

The substitution assumes exchangeability inside a cell, whose members the selector grouped by similarity to their representative.

The capped bias. The count overstates the mass a cell carried. The representative is a token the selector retained for its saliency and the members are tokens it discarded, so $e^{z_i} < e^{z_j}$ is the common case and a large cell is the one that overstates the most. We therefore cap the count at its mean, which is N/k since $\sum_{j \in S}(1 + |C_j|) = N$,

$$m_j \;=\; \min\Big\{1 + |C_j|,\; \frac{N}{k}\Big\}, \qquad b_j \;=\; \min\Big\{\log\big(1 + |C_j|\big),\; \log\frac{N}{k}\Big\}. \tag{40}$$

The cap leaves a below-average cell at its per-cell value and returns an above-average one to the mean. The biased masses then sum to less than Z_V, so the capped form restores less share than pruning removed. The capped m_j are flatter than those of the uncapped form, so the cap perturbs $\bar{v}_S$ less, which is the safer choice when the count is unreliable.

C Experimental Details and Results

C.1 Algorithm

To more clearly illustrate the proposed method, particularly the greedy coverage selection and the form of the bias it produces, we provide the pseudocode of CAR in Algorithm 1.

Algorithm 1 CAR with Weighted Coverage Selection and Attention Reallocation

Require:
Encoder features $X \in \mathbb{R}^{N \times d}$ (visual tokens entering the language model)
Saliency $w \in [0,1]^N$ from the probed head h^*
Grid positions $g_1, \ldots, g_N$ of the visual tokens
Target number of retained tokens k
Task family (grounding or question answering)
Ensure:
Retained indices $S \subset \{1, \ldots, N\}$, $|S| = k$ (positions of the selected tokens)
Logit bias $b \in \mathbb{R}^k$ (one value per retained token)
Similarity between visual tokens
$X \leftarrow \mathrm{rownormalise}(X)$ ▷ Unit-norm features
$M \leftarrow [\, XX^\top]_+$ ▷ Clipped cosine similarity, Eq. 7
if grounding **then**
  $K_{ij} \leftarrow \exp(-\|g_i - g_j\|^2 / 2(N/k))$ ▷ Eq. 8
  $M \leftarrow M \odot K$ ▷ Coverage becomes local
end if
Greedy coverage selection
$S \leftarrow \emptyset,\ c \leftarrow \mathbf{0}_N$ ▷ c_i: how well token i is covered
for $t = 1, \ldots, k$ **do**
  $j^* \leftarrow \arg\max_{j \notin S} w^\top [\, M_{:,j} - c]_+$ ▷ Largest gain, Eq. 10
  $S \leftarrow S \cup \{j^*\},\ c \leftarrow \max(c,\ M_{:,j^*})$ ▷ Update coverage
end for
Cell of each retained token
$C_j \leftarrow \{\, i \notin S :\ j = \arg\max_{j' \in S} M_{ij'} \,\}$ ▷ Closest survivor of each discarded token
Attention reallocation
if grounding **then**
  $b_j \leftarrow \log(N/k)$ ▷ Uniform bias, Eq. 12
else
  $b_j \leftarrow \min\{\log(1 + |C_j|), \log(N/k)\}$ ▷ Capped per-cell bias, Eq. 14
end if
Add b_j to the logits of $j \in S$ in every decoder layer during prefill ▷ Restores the visual attention share, Eq. 2
return S, b

C.2 Common Setup

Environment. To ensure the reproducibility of the results presented in this paper, Table 9 lists the main packages. Model weights are `Qwen/Qwen2.5-VL-7B-Instruct`, `Qwen/Qwen3-VL-8B-Instruct`, `OpenGVLab/InternVL3_5-8B-HF` and `liuhaotian/llava-v1.5-7b` from Hugging Face.

Benchmarks. Grounding uses the eight splits of RefCOCO, RefCOCO+ (Yu et al., 2016) and RefCOCOg (Mao et al., 2016); on LLaVA-1.5-7B it uses the seven splits evaluated by Nüwa (Huang et al., 2026) through lmms-eval (Appendix C.8). Question answering uses TextVQA (Singh et al., 2019), ChartQA (Masry et al., 2022), AI2D (Kembhavi et al., 2016), OCRBench (Liu et al., 2024e), HallusionBench (Guan et al., 2024), MME (Fu et al., 2025a) and MMBench-EN/CN (Liu et al., 2024d). Video understanding uses Video-MME (Fu et al., 2025b), LongVideoBench (Wu et al.,

Table 9: Main packages of the three environments.

Package	Qwen2.5-VL-7B	Qwen3-VL-8B / InternVL3.5-8B	LLaVA-1.5-7B
python	3.10.4	3.10.4	3.10.4
torch	2.3.0+cu118	2.8.0+cu126	2.6.0+cu124
transformers	4.51.3	4.57.6	4.54.0
flash-attn	2.5.8	2.8.3	–
accelerate	1.12.0	1.12.0	1.8.1
qwen-vl-utils	0.0.14	0.0.14	–
llava	–	–	1.2.2.post1
VLMEvalKit	0.1.0	0.1.0	–
lmms-eval	–	0.5.0 (video)	0.3.0

2024) and MVBench (Li et al., 2024) on Qwen3-VL-8B at 32 frames. In the question answering tables, Acc. is the mean of the eight benchmarks, with OCRBench divided by 10 and the MME total score by 20, and Rel. is the mean over benchmarks of the ratio to the unpruned score.

Implementation Details. For a fair comparison, we reproduce all baselines from their released code and evaluate them in the same pipeline as our method. We align the token budget of all methods by the average number of visual tokens per decoder layer,

$$\bar{k} = \frac{1}{L} \sum_{l=1}^{L} k_l, \tag{41}$$

where L is the number of decoder layers and k_l is the number of visual tokens entering layer l.

C.3 Saliency Source and Head Probe

Probe Protocol. We detail below how the probe of Eq. 9 is run. Table 10 lists the source it selects for each backbone and task family.

Head probe on VQA. We take the VQA task of Qwen2.5-VL-7B as an example, where 128 visual tokens are retained. The probe set consists of 500 samples from each of ChartQA, OCRBench, TextVQA and MMBench-EN, and the unpruned model is run over it once to give the reference answers. The candidates are the 16 heads of the final encoder block and their average. We then run CAR once per candidate, with w read from it, and score it by the exact-match rate of its answers against the references. Since these rates are not on a common scale across benchmarks, we rank the candidates within each benchmark and keep the one with the best mean rank as h^*. The VQA and video probes on Qwen3-VL-8B proceed in the same way.

Head probe on grounding. We take the grounding task of Qwen2.5-VL-7B as an example, where 9.9% of the visual tokens are retained. The probe set consists of 500 images of RefCOCO val, over which the unpruned model gives the reference boxes. We then run CAR once per candidate, again the 16 heads of the final encoder block and their average, and score it by the mean IoU of its boxes against the references; the highest-scoring candidate is kept as h^*. The grounding probe on Qwen3-VL-8B proceeds in the same way.

Layer and head probe on InternVL3.5-8B and LLaVA-1.5-7B. We take the VQA task of InternVL3.5-8B as an example, where 11.1% of the visual tokens are retained. The probe set again consists of 500 samples from each of ChartQA, OCRBench, TextVQA and MMBench-EN, over which the unpruned model gives the reference answers. The late layers of this encoder exhibit attention sinks (Darcet et al., 2024; Luo et al., 2026), where a few tokens absorb most of the attention, so the final layer is not the best source and the probe covers the layers as well, in two steps. It first runs CAR once per encoder layer, with w read from that layer and averaged over its heads, and keeps the layer with the best mean rank; it then repeats this over the heads of that layer and their average, which gives h^*. The grounding probe on InternVL3.5-8B proceeds in the same way. On LLaVA-1.5-7B, which we evaluate only on grounding, the probe likewise selects the layer and then the head, with the probe set and score of the grounding probe above.

The probe is run once for each backbone and task family and needs one evaluation of the probe set per candidate, together with one run of the unpruned model. On Qwen2.5-VL-7B this takes about

1.4 hours on grounding and 6 hours on VQA, measured on seven RTX 3090 GPUs. The source it selects is then fixed for all later experiments.

Table 10: Saliency source selected by the probe on each backbone and task family. The candidates are the 16 heads of the final encoder block and their average; on InternVL3.5-8B and LLaVA-1.5-7B the probe selects the encoder layer as well.

Backbone	Task	Encoder layer	Head
Qwen2.5-VL-7B	VQA	final	Head 2
	Grounding	final	Head 13
Qwen3-VL-8B	VQA	final	Head 13
	Grounding	final	Head 8
	Video	final	Head avg.
InternVL3.5-8B	VQA	Layer 10	Head 10
	Grounding	Layer 12	Head avg.
LLaVA-1.5-7B	Grounding	Layer 9	Head 15

Attention Concentration in the InternVL3.5-8B Vision Encoder. Table 11 quantifies the attention sinks of the late layers of the InternVL3.5-8B vision encoder. Per image tile, we average the attention each token receives over the heads, sum it over the queries, and report the share taken by the class token together with the share of patch attention taken by the top 1% of patches. Up to layer 11 both quantities stay at about 7% or below; from layer 12 on they grow steadily, and over the last eight layers the top 1% of patches take up to a quarter of the patch attention. This is exactly where the accuracy in Figure 1(f) begins to decline, since a few tokens absorb most of the attention and the tokens the task needs are then hard to cover.

Table 11: Attention concentration across layers of the InternVL3.5-8B vision encoder.

Layers	Class-token share (%)	Top-1% patch share (%)
0–11	3.3–7.1	2.4–6.3
12	13.8	6.7
13–15	17.4–22.8	12.2–13.5
16–23	31.2–38.8	17.9–26.8

Reliability of the Probe. The probe selects the source whose outputs agree most with the unpruned model. To check its reliability, Figure 4 compares this ranking with the ranking by labeled accuracy. On grounding the two agree almost perfectly (Spearman 0.986), and the probe's first choice, head 13, is also the most accurate. On VQA they correlate at 0.929, and the two candidates the probe ties at the top, heads 2 and 9, differ by 0.15 points in accuracy. On both tasks the label-free criterion therefore identifies the important head.

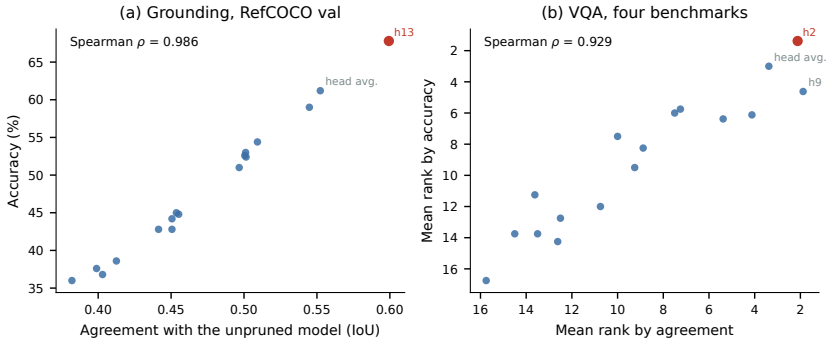


Figure 4: **Agreement with the unpruned model tracks accuracy across the candidate sources** (Qwen2.5-VL-7B, 16 heads and their average; the selected source is in red). (a) Grounding at 9.9%, on RefCOCO val. (b) VQA at 128 tokens, mean ranks over the four probe benchmarks.

Table 12: **VQA performance comparison on Qwen2.5-VL-7B with 32 retained tokens.**

Method	TextVQA	ChartQA	AI2D	OCRBench	HallBench	MME	MMB-EN	MMB-CN	Acc.	Rel.
Upper Bound, All 1,296 Tokens (100%)										
Qwen2.5-VL-7B	84.3	87.1	81.6	867	56.2	2342.3	83.1	82.9	84.9	100.0%
Retain 32 Tokens (↓97.5%)										
AnchorPrune (ECCV26)	53.5	35.4	69.8	435	39.0	1879.5	73.2	72.1	60.0	70.6%
HAWK (CVPR26)	**66.1**	**38.8**	70.8	431	43.1	1791.2	64.2	63.9	59.9	70.9%
Nüwa (ICLR26)	48.7	38.1	69.2	381	38.8	1820.6	70.4	70.0	58.0	68.3%
CAR (Ours)	60.6	37.5	**73.1**	**523**	**44.0**	**2120.6**	**76.5**	**74.5**	**65.6**	**76.9%**

Probe on Held-Out Data. To test whether the selected source depends on the probe data, we run the probe on Qwen2.5-VL-7B again on data disjoint from every evaluation set. For VQA, the probe set takes 500 samples from each of DocVQA val, InfographicVQA val, MMStar and RealWorldQA, none of which is among the eight VQA benchmarks. For grounding, it takes 500 referring expressions from the RefCOCO train split, one per image, excluding every image used by the evaluation splits of RefCOCO, RefCOCO+ and RefCOCOg. All other settings follow the probe protocol above (Appendix C.3). On both tasks the held-out probe selects the same source as before, head 2 on VQA and head 13 on grounding, and its ranking of the 17 candidates agrees with that of the original probe (Spearman 0.931 and 0.956). The source therefore transfers across probe data.

C.4 QWEN2.5-VL-7B

Setup. We describe how our method is implemented on Qwen2.5-VL-7B. The saliency of visual token i is

$$w_i = \frac{1}{4} \sum_{p \in \mathcal{P}_i} \sum_{q} A_{qp}^{(h^*)}, \tag{42}$$

rescaled to $[0, 1]$ within each image, where $A^{(h^*)}$ is the attention map of head h^* in the last vision-encoder block, which uses full attention, and $\mathcal{P}_i$ is the set of four patches merged into token i. The features x_i are the merger outputs, which are the vectors passed to the language model. For VQA, we pad each image to a square and resize it to 1008×1008, giving 1,296 visual tokens. For grounding, images keep Qwen's native dynamic resolution, and budgets are given as fractions of the visual tokens. Retained tokens keep their original M-RoPE positions.

VQA under an Extreme Budget. When about 90% of the visual tokens are pruned, existing methods degrade severely on grounding but largely preserve their performance on VQA. To examine VQA under more aggressive pruning, Table 12 compares CAR with the three strongest baselines when only 32 tokens are retained, a 97.5% reduction. Even under this budget, the best baseline retains 70.9% of the unpruned performance, which indicates that grounding is more sensitive to pruning than VQA. CAR retains 76.9%, leading the best baseline by 6.0 percentage points.

C.5 QWEN3-VL-8B

Setup. We describe how our method is implemented on Qwen3-VL-8B. The saliency is computed as in Eq. 42 from head h^* of the last vision-encoder block, and the features x_i are the merger outputs. Images are processed at native dynamic resolution, so budgets are given as fractions of the visual tokens. Qwen3-VL adopts DeepStack, which injects features from three intermediate vision-encoder blocks into the first three decoder layers. We keep DeepStack enabled for all methods and prune these intermediate features with the same indices retained by token pruning. Retained tokens keep their original M-RoPE positions.

VQA and Visual Grounding Results on Qwen3-VL-8B. Tables 13 and 14 compare CAR with five competitive baselines on Qwen3-VL-8B. On VQA, CAR retains 92.5% of the unpruned performance at a retention ratio of 22.2% and 81.7% at 11.1%, leading the best baseline by 3.6 and 4.8 percentage points. On grounding, it retains 84.9% at a retention ratio of 25% and 54.7% at 9.9%, leading the best baseline by 8.3 and 20.7 percentage points. CAR consistently demonstrates superior performance across different retention ratios for both tasks, leading in the vast majority of benchmarks. Its advantage becomes more pronounced as pruning becomes more aggressive on both tasks, and is also more prominent on visual grounding, which is more sensitive to pruning.

Table 13: **VQA performance comparison on Qwen3-VL-8B.**

Method	TextVQA	ChartQA	AI2D	OCRBench	HallBench	MME	MMB-EN	MMB-CN	Acc.	Rel.
Upper Bound, All Visual Tokens (100%)										
Qwen3-VL-8B	83.5	83.5	84.1	874	57.9	2417.2	85.4	84.9	85.9	100.0%
Retain **22.2**% *of Visual Tokens*										
VisionZip (CVPR25)	73.9	65.9	79.0	610	52.0	2162.9	80.7	79.9	75.1	87.4%
DivPrune (CVPR25)	72.9	52.9	77.5	630	48.4	2150.0	80.2	77.5	72.5	84.1%
AnchorPrune (ECCV26)	77.0	64.1	**80.5**	651	51.9	2250.9	81.4	79.8	76.5	88.9%
EADP (ECCV26)	75.3	58.6	78.7	675	49.5	2245.0	80.3	78.9	75.1	87.1%
Nüwa (ICLR26)	76.0	60.9	79.0	628	**53.9**	2260.0	81.6	79.9	75.9	88.2%
CAR (Ours)	**79.5**	**70.6**	80.4	**739**	53.0	**2340.0**	**82.1**	**81.2**	**79.7**	**92.5%**
Retain **11.1**% *of Visual Tokens*										
VisionZip (CVPR25)	55.6	43.0	71.5	347	41.9	1806.6	71.1	71.1	59.9	69.6%
DivPrune (CVPR25)	61.6	34.2	70.6	477	43.4	1897.3	72.1	71.7	62.0	72.0%
AnchorPrune (ECCV26)	61.8	40.5	73.4	405	44.3	1884.8	72.0	71.1	62.2	72.3%
EADP (ECCV26)	67.8	40.0	73.4	539	42.7	2074.2	75.8	75.3	66.6	76.9%
Nüwa (ICLR26)	64.4	45.8	72.9	403	44.9	2004.1	74.5	74.7	64.7	75.1%
CAR (Ours)	**72.2**	**46.8**	**75.7**	**564**	**48.1**	**2183.7**	**78.2**	**77.9**	**70.6**	**81.7%**

Table 14: **Visual grounding performance comparison on Qwen3-VL-8B.**

Method	RefCOCO			RefCOCO+			RefCOCOg		Avg.	Rel.
	testA	testB	val	testA	testB	val	test	val		
Upper Bound, All Visual Tokens (100%)										
Qwen3-VL-8B	90.4	84.7	88.5	85.8	75.5	81.4	87.1	86.4	85.0	100.0%
Moderate Budget, Retain **25**% *of Visual Tokens*										
VisionZip (CVPR25)	70.4	65.4	67.8	61.6	55.8	59.5	66.1	65.6	64.0	75.3%
DivPrune (CVPR25)	66.3	56.1	62.4	60.6	47.5	55.1	61.7	59.6	58.7	68.9%
AnchorPrune (ECCV26)	72.6	61.9	69.0	66.0	53.9	61.4	68.4	68.4	65.2	76.6%
EADP (ECCV26)	66.7	63.3	65.7	58.3	54.9	57.5	65.0	64.4	62.0	72.9%
Nüwa (ICLR26)	67.7	68.3	67.7	60.3	58.1	58.6	67.3	67.6	64.5	75.8%
CAR (Ours)	**78.3**	**73.8**	**76.8**	**69.6**	**63.1**	**66.9**	**74.5**	**74.7**	**72.2**	**84.9%**
Aggressive Budget, Retain **9.9**% *of Visual Tokens*										
VisionZip (CVPR25)	30.4	26.4	28.2	24.2	21.8	23.9	24.6	24.4	25.5	29.9%
DivPrune (CVPR25)	31.9	25.2	27.9	26.3	19.0	22.8	24.7	25.4	25.4	29.8%
AnchorPrune (ECCV26)	31.3	25.3	28.3	25.1	21.4	23.7	26.4	26.5	26.0	30.5%
EADP (ECCV26)	26.5	23.8	24.7	20.9	19.0	20.6	22.3	21.6	22.4	26.3%
Nüwa (ICLR26)	32.3	32.5	30.4	26.8	24.4	25.2	30.2	29.4	28.9	34.0%
CAR (Ours)	**53.7**	**48.7**	**51.4**	**44.3**	**38.9**	**41.9**	**47.0**	**46.8**	**46.6**	**54.7%**

C.6 VIDEO UNDERSTANDING ON QWEN3-VL-8B

Setup. We describe how our method is implemented for video understanding on Qwen3-VL-8B. Each video is sampled to 32 frames, and the visual tokens of all frames are selected jointly. The saliency is computed as in Eq. 42 from the last vision-encoder block, with the attention averaged over all heads (Appendix C.3), and the features x_i are the merger outputs. We use the capped bias of Eq. 14 without the spatial kernel. We keep DeepStack enabled for all methods and prune its intermediate features with the same indices retained by token pruning. Retained tokens keep their original M-RoPE positions.

Video Understanding Results on Qwen3-VL-8B. Table 15 compares CAR with six competitive baselines, including the video-specific GSTEP (Zhang et al., 2026c). CAR retains 97.9% of the unpruned performance at a retention ratio of 25% and 94.7% at 9.9%, leading the best baseline by 0.9 and 0.5 percentage points. It performs best on Video-MME and LongVideoBench at both retention ratios. Videos in these benchmarks are longer and repeat the same content across many frames, and the coverage objective removes this redundancy and spreads the retained tokens over the distinct content of the video. On MVBench, CAR trails the best baseline by about 1 point, mainly on the five subtasks built on CLEVRER (Yi et al., 2020), which ask about the motion of small objects: such an object looks nearly identical across frames, so CAR keeps few of its copies and loses the positions that reveal its motion.

C.7 INTERNVL3.5-8B

Setup. We describe how our method is implemented on InternVL3.5-8B. Following the InternVL evaluation setting in VLMEvalKit, each image is split into at most 12 tiles of 448×448 pixels for ChartQA, 24 for OCRBench and 6 for the other benchmarks, plus a thumbnail of the whole image

Table 15: **Video understanding performance comparison on Qwen3-VL-8B.**

Method	Video-MME				Long VideoBench	MVBench	Avg.	Rel.
	Short	Medium	Long	Overall				
Upper Bound, All Visual Tokens (100%)								
Qwen3-VL-8B	78.1	63.4	55.7	65.7	60.2	69.5	65.5	100.0%
Retain **25**% *of Visual Tokens*								
VisionZip (CVPR25)	74.4	60.9	54.2	63.2	57.6	**68.9**	63.2	96.6%
DivPrune (CVPR25)	72.9	59.1	53.8	61.9	57.1	65.6	61.7	94.4%
AnchorPrune (ECCV26)	75.3	61.1	54.3	63.6	57.8	68.1	63.4	96.8%
EADP (ECCV26)	74.1	61.9	54.3	63.4	57.5	68.7	63.3	96.8%
GSTEP (MM26)	74.9	58.2	53.7	62.3	58.6	68.8	62.7	95.8%
Nüwa (ICLR26)	75.9	61.0	54.1	63.7	57.8	68.5	63.5	97.0%
CAR (Ours)	**76.1**	**62.4**	**54.8**	**64.4**	**58.9**	67.7	**64.0**	**97.9%**
Retain **9.9**% *of Visual Tokens*								
VisionZip (CVPR25)	69.9	57.2	53.1	60.1	55.0	61.5	59.5	91.1%
DivPrune (CVPR25)	68.7	56.7	50.3	58.6	54.4	60.1	58.1	88.9%
AnchorPrune (ECCV26)	71.4	57.7	53.1	60.7	55.3	64.1	60.4	92.4%
EADP (ECCV26)	70.4	57.6	53.8	60.6	53.9	62.8	59.8	91.6%
GSTEP (MM26)	71.9	56.8	52.6	60.4	57.4	**66.1**	60.9	93.0%
Nüwa (ICLR26)	72.9	**59.6**	53.6	62.0	55.7	66.0	61.6	94.2%
CAR (Ours)	**73.7**	59.1	**53.9**	**62.2**	**57.8**	65.0	**62.0**	**94.7%**

when more than one tile is used. We find that retaining thumbnail tokens is important, since the thumbnail is the only view of the whole image. The thumbnail has 256 tokens. For all methods, we therefore reserve 128 of the k retained tokens for the thumbnail, or all k when $k < 128$, and assign the rest to the other tiles. The tokens of the thumbnail and of the other tiles are selected separately. The saliency is computed per tile as in Eq. 42, from the vision-encoder layer and head selected in Appendix C.3, and the features x_i are the projector outputs. The spatial kernel on InternVL3.5-8B is

$$s_{ij} = \big[\cos(x_i, x_j)\big]_+ \cdot \exp\Big(-\frac{\|g_i - g_j\|^2}{2\gamma}\Big), \tag{43}$$

with the grid position g_i and the scale γ given by

$$\begin{aligned} &\text{tiles:} && g_i = (16r + u,\ 16c + v), && \gamma = 256\, n_{\text{tile}}/k, \\ &\text{thumbnail:} && g_i = (u,\ v), && \gamma = 256/k_{\text{th}}, \end{aligned}$$

where (u, v) is the row and column of the token in the 16×16 token grid of its tile or of the thumbnail, (r, c) is the row and column of the tile in the tile layout, n_{tile} is the number of tiles, and k_{th} is the number of tokens reserved for the thumbnail. We disable the bias on this backbone, where it does not improve accuracy (Table 25). Retained tokens keep their original positions.

VQA and Visual Grounding Results on InternVL3.5-8B. Tables 16 and 17 compare CAR with five competitive baselines on InternVL3.5-8B. On VQA, CAR retains 94.9% of the unpruned performance at a retention ratio of 22.2% and 89.3% at 11.1%, leading the best baseline by 6.6 and 5.9 percentage points. On grounding, it retains 93.8% at a retention ratio of 25% and 85.2% at 9.9%, leading the best baseline by 2.1 and 1.4 percentage points. This is attributed to the saliency-weighted coverage: since the late layers of the InternVL3.5 vision encoder suffer from attention sinks (Appendix C.3), the probe selects a suitable intermediate layer to extract the saliency, which helps retain important tokens.

Why the Bias Is Disabled on InternVL3.5-8B. InternVL3.5-8B splits an image into tiles and appends a thumbnail of the whole image, and it relies on the thumbnail much more than on the other tiles: in the unpruned model the thumbnail holds 16% of the visual tokens but receives 36% of the attention the text pays to the image. Prior work likewise treats the thumbnail as the global view of high-resolution LVLMs and uses it to guide the compression of the tiles (Liu et al., 2026b; Sun et al., 2026b; Liang et al., 2026). Following IVC-Prune (Sun et al., 2026b), we keep half of the thumbnail tokens and assign the rest of the budget to the other tiles.

Table 18 gives the visual attention share under this protocol before and after pruning, separately for the thumbnail and the other tiles. Pruning lowers the visual share from 25.8% to 15.8%, and the drop comes entirely from the other tiles: their share falls to 0.25 of its unpruned value, while that of the thumbnail does not drop and even rises to 1.26, as it now competes with far fewer tile

Table 16: **VQA performance comparison on InternVL3.5-8B.**

Method	TextVQA	ChartQA	AI2D	OCRBench	HallBench	MME	MMB-EN	MMB-CN	Acc.	Rel.
Upper Bound, All Visual Tokens (100%)										
InternVL3.5-8B	76.5	86.3	83.4	839	52.7	2333.1	83.0	81.3	83.0	100.0%
Retain **22.2**% *of Visual Tokens*										
VisionZip (CVPR25)	56.6	54.5	78.5	369	42.0	2143.8	77.2	75.7	66.1	79.1%
DivPrune (CVPR25)	67.7	68.6	78.8	591	44.4	2244.2	79.1	76.8	73.3	87.9%
AnchorPrune (ECCV26)	64.0	64.4	79.0	508	44.6	2159.4	**79.7**	77.7	71.0	85.3%
EADP (ECCV26)	71.0	65.4	77.5	653	43.9	2220.0	77.6	77.1	73.6	88.3%
Nüwa (ICLR26)	56.7	45.8	76.1	296	42.9	2055.0	77.0	76.2	63.4	76.2%
CAR (Ours)	**72.7**	**80.9**	**81.3**	**759**	**47.7**	**2336.7**	**79.7**	**77.9**	**79.1**	**94.9%**
Retain **11.1**% *of Visual Tokens*										
VisionZip (CVPR25)	52.2	52.8	76.4	343	38.3	2066.2	73.2	71.9	62.8	75.0%
DivPrune (CVPR25)	61.4	60.9	76.3	505	42.8	2186.2	76.5	74.8	69.1	82.7%
AnchorPrune (ECCV26)	57.0	58.0	77.2	435	42.2	2096.3	75.7	73.5	66.5	79.7%
EADP (ECCV26)	64.0	56.3	75.2	576	44.2	2173.8	75.7	74.4	69.5	83.4%
Nüwa (ICLR26)	48.1	37.6	73.6	252	39.4	1991.3	71.9	70.2	58.2	69.7%
CAR (Ours)	**66.1**	**68.7**	**79.5**	**666**	**47.5**	**2280.4**	**77.7**	**75.0**	**74.4**	**89.3%**

Table 17: **Visual grounding performance comparison on InternVL3.5-8B.**

Method	RefCOCO			RefCOCO+			RefCOCOg		Avg.	Rel.
	testA	testB	val	testA	testB	val	test	val		
Upper Bound, All Visual Tokens (100%)										
InternVL3.5-8B	94.3	88.9	92.0	92.2	82.5	87.7	89.2	88.6	89.4	100.0%
Moderate Budget, Retain **25**% *of Visual Tokens*										
VisionZip (CVPR25)	83.6	79.0	81.1	79.5	70.5	75.3	79.5	79.6	78.5	87.8%
DivPrune (CVPR25)	87.4	78.0	81.8	84.8	71.1	76.9	80.5	80.2	80.1	89.5%
AnchorPrune (ECCV26)	85.3	78.3	80.6	81.7	70.8	75.3	80.1	79.2	78.9	88.2%
EADP (ECCV26)	88.8	80.5	84.6	85.9	73.5	79.4	81.7	81.7	82.0	91.7%
Nüwa (ICLR26)	78.1	72.6	74.7	73.4	65.6	69.3	74.7	75.0	72.9	81.5%
CAR (Ours)	**89.9**	**82.5**	**86.5**	**86.8**	**76.2**	**81.3**	**84.3**	**83.5**	**83.9**	**93.8%**
Aggressive Budget, Retain **9.9**% *of Visual Tokens*										
VisionZip (CVPR25)	76.9	71.2	72.3	71.3	63.7	66.3	72.4	72.6	70.8	79.2%
DivPrune (CVPR25)	80.9	70.9	76.1	77.7	63.7	70.1	73.2	73.4	73.3	81.8%
AnchorPrune (ECCV26)	77.9	71.0	73.8	74.3	63.8	67.8	73.0	73.3	71.9	80.3%
EADP (ECCV26)	82.0	73.8	77.2	78.9	67.0	71.8	74.3	75.0	75.0	83.8%
Nüwa (ICLR26)	67.8	64.3	64.4	61.6	56.2	57.6	64.0	65.7	62.7	70.1%
CAR (Ours)	**82.7**	**75.5**	**78.7**	**79.3**	**67.8**	**73.7**	**76.0**	**76.1**	**76.2**	**85.2%**

tokens. The global view the model relies on most is thus not weakened, and the missing attention lies on the high-resolution tiles, whose content the thumbnail already represents at a coarser scale. Restoring attention to these tiles therefore adds little information: computing the bias separately for the thumbnail and the other tiles, so that almost all of the compensation goes to the tiles, restores their share to 0.81 but does not improve accuracy. We therefore disable the bias on this backbone.

Table 18: **Visual attention share on InternVL3.5-8B** (11.1% of the visual tokens retained, half of the thumbnail kept). Shares (%) are fractions of the attention the text pays to all tokens, averaged over layers and 32 images; parentheses give the ratio to the unpruned value. The per-pool bias is computed separately for the thumbnail and the other tiles.

	Thumbnail	Other tiles	All visual
Unpruned	9.2	16.6	25.8
Pruned, no bias	11.6 (1.26×)	4.1 (0.25×)	15.8 (0.61×)
Pruned, per-pool bias	11.1 (1.20×)	13.4 (0.81×)	24.5 (0.95×)

C.8 LLaVA-1.5-7B

Setup. We describe how our method is implemented on LLaVA-1.5-7B. Each image is padded to a square and resized to 336×336, giving 576 visual tokens on a 24×24 grid. The saliency is computed as in Eq. 42 from the layer and head selected by the probe (Appendix C.3). The features x_i are the projector outputs, and g_i in Eq. 8 is the grid position of token i. Following Nüwa (Huang et al., 2026), we evaluate grounding with lmms-eval on seven splits of RefCOCO, RefCOCO+ and RefCOCOg, retaining 64 visual tokens on average. Since grounding depends on where the retained tokens lie, retained tokens keep their original positions for all methods.

Table 19: **Visual grounding performance comparison on LLaVA-1.5-7B.**

Method	RefCOCO		RefCOCO+			RefCOCOg		Avg.	Rel.
	test	val	testA	testB	val	test	val		
Upper Bound, All Visual Tokens (**576**)									
LLaVA-1.5-7B	57.8	56.2	59.3	38.9	49.9	48.4	48.7	51.3	100.0%
Average Tokens **64** (**11.1%** *of Visual Tokens*)									
VisionZip (CVPR25)	9.4	9.4	7.4	8.1	7.3	7.2	7.2	8.0	15.8%
DivPrune (CVPR25)	20.3	19.6	19.2	13.8	16.7	15.6	16.1	17.3	33.8%
AnchorPrune (ECCV26)	17.3	16.1	15.5	12.6	14.3	15.2	15.0	15.1	29.7%
EADP (ECCV26)	23.1	22.2	22.6	15.0	19.3	19.2	18.8	20.0	39.0%
Nüwa (ICLR26)	33.2	32.7	34.0	20.6	26.5	24.5	25.6	28.2	54.6%
CAR (Ours)	**36.4**	**35.5**	**35.2**	**22.3**	**29.0**	**27.2**	**27.3**	**30.4**	**59.0%**

Table 20: **Component ablation on visual grounding with Qwen2.5-VL-7B** (9.9% of the visual tokens retained). Δ is the gain over the row above.

Component					RefCOCO			RefCOCO+			RefCOCOg		Avg.	Δ
Facility loc.	Bias	Saliency w	Head sel.	Spatial	testA	testB	val	testA	testB	val	test	val		
Upper Bound, All Visual Tokens (**100%**)														
		–			92.2	84.8	89.4	87.6	74.1	82.8	86.7	86.7	85.5	–
No Selection Signal (Random Selection)														
×	×	–	–	–	23.1	17.2	21.5	20.3	14.5	18.6	17.3	18.1	18.8	–
×	✓	–	–	–	44.0	37.6	42.9	38.3	31.7	37.0	39.0	38.4	38.6	+19.8
✓	×	–	–	×	25.3	17.6	21.5	22.9	16.0	19.2	18.8	19.4	20.1	–
✓	✓	–	–	×	57.9	46.7	55.0	53.2	38.6	48.7	49.8	50.1	50.0	+29.9
✓	✓	✓	×	×	62.9	53.4	58.6	56.5	44.6	50.8	55.6	56.3	54.8	+4.8
✓	✓	✓	✓	×	71.8	60.6	67.7	64.2	50.8	59.5	63.6	61.7	62.5	+7.7
✓	✓	✓	✓	✓	**75.4**	**66.7**	**72.2**	**67.2**	**55.9**	**62.3**	**67.8**	**66.5**	**66.8**	+4.3

Visual Grounding Results on LLaVA-1.5-7B. Table 19 compares CAR with five competitive baselines on LLaVA-1.5-7B. Retaining 64 visual tokens on average, CAR retains 59.0% of the unpruned performance, performing best on every split and leading the best baseline, Nüwa, by 4.4 percentage points. Nüwa and CAR both preserve the spatial structure that grounding relies on, Nüwa by partitioning the token grid and aggregating the discarded tokens into their spatial neighbours, and CAR by weighted coverage selection with a spatial kernel. Two factors contribute to the further lead of CAR. The late layers of the CLIP encoder exhibit attention sinks, so the probe selects the intermediate layer 9 as the saliency source, which helps retain important tokens. The bias restores the attention share lost to pruning and further improves accuracy on average (Table 25).

C.9 Full Results of the Component Ablation

Tables 20 and 21 give the per-split and per-benchmark results of the component ablation on Qwen2.5-VL-7B, of which Table 5 reports the averages. To separate the contribution of the selection from that of the attention reallocation, the tables first list two references without any selection signal: random selection, and random selection with our bias alone. The remaining rows add the components cumulatively: (i) the unweighted facility-location objective; (ii) the logit bias; (iii) the saliency w weighting each token in that objective, with w averaged over the heads; (iv) the probed head h^* in place of the head average; and (v) the spatial kernel on the similarity, which completes the method.

The two tasks depend on different components. On grounding the attention reallocation dominates: adding the bias raises the average accuracy from 18.8 to 38.6 under random selection and from 20.1 to 50.0 under coverage selection, whereas without the bias, replacing random selection by coverage selection is worth only 1.3 points. Restoring the visual attention share is therefore a precondition at this budget, and the selection pays off only once the share is restored. On top of it, the saliency weighting and the probed head add 4.8 and 7.7 points and the spatial kernel a further 4.3, consistently over the eight splits. On VQA the components rank the other way round: the bias is worth 0.3 and 1.5 points under the two selections, while coverage selection over random selection is worth 8.0 points, and the saliency weighting and the probed head add 4.8 and 0.8. The spatial kernel costs 0.2 points there, so we enable it on grounding only.

Table 21: **Component ablation on VQA with Qwen2.5-VL-7B** (128 of 1296 visual tokens retained). The shaded row is the deployed configuration. Δ is the gain over the row above.

Component					TextVQA	ChartQA	AI2D	OCRBench	HallB	MME	MMB_{EN}	MMB_{CN}	Acc.	Δ
Facility loc.	Bias	Saliency w	Head sel.	Spatial										
Upper Bound, All Visual Tokens (100%)														
		–			84.3	87.1	81.6	867	56.2	2342.3	83.1	82.9	84.9	–
No Selection Signal (Random Selection)														
×	×	–	–	–	56.4	38.0	72.5	497	41.2	2066.8	76.2	76.4	64.2	–
×	✓	–	–	–	55.0	37.0	73.9	489	42.5	2100.4	75.8	77.6	64.5	+0.3
✓	×	–	–	×	69.7	54.5	76.7	630	45.1	2201.6	79.8	79.0	72.2	–
✓	✓	–	–	×	70.1	58.1	77.2	645	48.3	2217.6	80.4	80.1	73.7	+1.5
✓	✓	✓	×	×	78.3	71.8	79.3	708	49.5	2330.7	80.9	80.5	78.5	+4.8
✓	✓	✓	✓	×	78.8	76.1	79.4	731	49.0	2314.3	81.1	81.3	79.3	+0.8
✓	✓	✓	✓	✓	77.6	76.5	79.5	735	49.3	2302.5	80.3	80.8	79.1	-0.2

Table 22: **Bias form ablation on visual grounding with Qwen2.5-VL-7B** (9.9% of the visual tokens retained).

Bias b_j	Distribution	Mass	RefCOCO			RefCOCO+			RefCOCOg		Avg.
			testA	testB	val	testA	testB	val	test	val	
None	—	k	33.9	24.5	29.2	30.2	20.9	25.4	25.2	25.0	26.8
$\log(N/k)$	constant	N	75.4	66.7	72.2	67.2	55.9	62.3	67.8	66.5	**66.8**
$\log(1+\lvert C_j\rvert)$	per-group	N	74.5	66.2	70.7	66.1	54.7	61.0	65.9	65.3	65.5
$\min\big(\log(1+\lvert C_j\rvert), \log(N/k)\big)$	per-group	$< N$	74.5	66.8	71.1	66.6	55.9	61.5	67.2	66.6	66.3

C.10 Attention Bias

Full Results of the Bias-Form Ablation. Tables 22 and 23 give the per-split and per-benchmark results of the bias-form ablation, of which Table 7 reports the averages. All rows share the same selector and therefore the same retained tokens, and differ only in the bias added to the retained visual logits: no bias, the uniform bias $\log(N/k)$ of Eq. 12, the per-cell bias $\log(1+|C_j|)$ of Eq. 13 and the capped bias of Eq. 14. The tables separate the two factors that distinguish these forms: *Mass* is the total $\sum_{j\in S} e^{b_j}$ that the bias hands to the survivors, which the condition $\sigma' = \sigma$ fixes at N, and *Distribution* is how that total is spread over them, which the condition leaves free. Rows two and three restore the same mass with different distributions, rows three and four share a distribution and differ in mass, so each factor is read off an adjacent pair.

Restoring the attention share helps on both tasks, but the two factors do not act alike. On grounding all three biases raise the average accuracy from 26.8 to between 65.5 and 66.8, a spread of at most 1.3 points, so the gain comes from the mass that is restored; at equal mass, the constant distribution is 1.3 points above the per-cell one. On VQA the bias matters far less and the order reverses: the uniform bias is 0.4 points below no bias, while the per-cell and capped forms add 0.5 and 0.6. At equal distribution, the capped bias is no worse than the per-cell one on either task (66.3 against 65.5, 79.3 against 79.2). We therefore use the uniform bias for grounding and the capped bias for VQA.

Visual Attention Share after Pruning. We check how much of the visual attention share each bias restores across the decoder layers. At every layer we compute the softmax mass that a text query assigns to the visual tokens, divide it by the value the unpruned model gives on the same image, and average over images and heads; a ratio of 1 means the retained tokens receive as much attention as the whole image did. Figure 5 shows that without a bias the ratio stays below 1 at every layer, reaching 0.60 on grounding and 0.74 on VQA over layers 13 to 19, where text attends to the image most. The three biases raise it to 0.87, 0.77 and 0.76 on grounding and to 0.84, 0.81 and 0.81 on VQA. The uniform bias restores the most and overshoots in the early layers, where it exceeds 1 on grounding, as expected from a value that satisfies $\sigma' = \sigma$ only in expectation. How much is restored does not order the accuracies, however: on VQA the uniform bias restores the most share and is the least accurate of the three (Table 23). Restoring the share is therefore necessary but not sufficient, and accuracy also depends on how the restored share is spread over the retained tokens.

Where the Bias Is Applied. The bias enters the prefill only: the queries of the prefill attend to the retained tokens through the biased logits, while each decoding step runs the unmodified kernel on the cached keys and values, so decoding costs the same as in the pruned model (Table 31). Table 24 compares adding the bias at the prefill only with adding it at every decoding step as well. The two settings share the selector and the retained tokens and differ only in where the bias is added. They are equally accurate on VQA and within 0.2 points on VG, so adding the bias at the decoding steps

Table 23: **Bias form ablation on VQA with Qwen2.5-VL-7B** (128 of 1296 visual tokens retained).

Bias b_j	Distribution	Mass	TextVQA	ChartQA	AI2D	OCRBench	HallB	MME	MMB_{EN}	MMB_{CN}	Acc.
None	—	k	78.5	74.6	79.1	732	48.8	2270.0	80.7	81.0	78.7
$\log(N/k)$	constant	N	77.8	74.2	78.3	727	47.8	2298.5	80.6	80.0	78.3
$\log(1+\lvert C_j\rvert)$	per-group	N	79.0	76.3	79.3	732	48.8	2302.3	80.5	81.5	79.2
$\min\big(\log(1+\lvert C_j\rvert), \log(N/k)\big)$	per-group	$< N$	78.8	76.1	79.4	731	49.0	2314.3	81.1	81.3	**79.3**

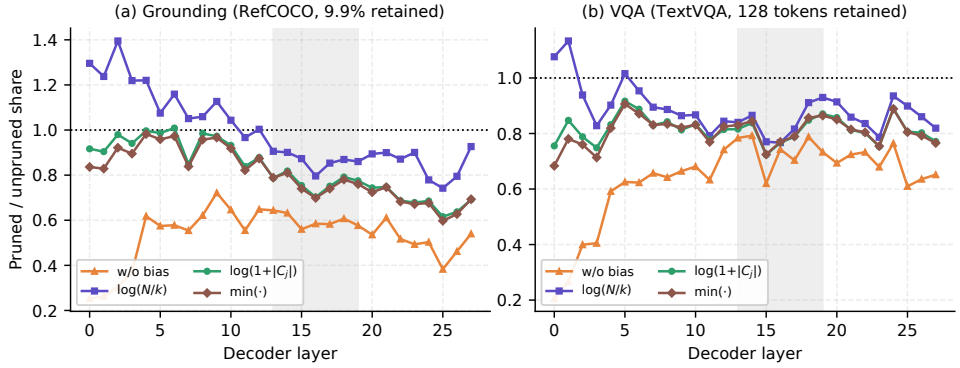


Figure 5: **Visual attention share restored by each bias, across decoder layers** (Qwen2.5-VL-7B, 100 images per task). The curves share the selector of the deployed configuration and differ only in the bias; the dotted line marks the unpruned level and the band marks layers 13 to 19.

brings no further gain. The correction still reaches the decoding steps, through the prompt tokens. At the prefill, the bias raises the attention that every prompt token pays to the retained visual tokens, so their hidden states, and the keys and values computed from them and stored in the cache, carry the corrected image information. Each generated token attends to these cached prompt tokens, and so reads the corrected information without a bias of its own. Reading the image through the prompt tokens is how MLLMs normally work. In the middle layers the question tokens gather the visual information relevant to the question, which is then passed to the last prompt position that predicts the answer (Zhang et al., 2025c), and a model can still describe an image from the question tokens alone when the generated tokens are not allowed to attend to the image tokens (Kaduri et al., 2025).

Table 24: **Bias applied at the prefill only or at the prefill and every decoding step** (Qwen2.5-VL-7B). VG: average over $\text{RefCOCO}_{\text{testA}}$, $\text{RefCOCO+}_{\text{testB}}$ and $\text{RefCOCOg}_{\text{test}}$ at 9.9% of the visual tokens. VQA: average over eight benchmarks at 128 of 1,296 tokens.

Bias applied at	VG	VQA
None (no bias)	26.7	78.7
Prefill only (ours)	66.4	79.3
Prefill + Decode	66.2	79.3

Effect of the Bias across Backbones. Table 25 compares CAR with and without the bias on the four backbones, at the lowest retention ratio of each task; the two settings differ only in the bias. On Qwen2.5-VL-7B, Qwen3-VL-8B and LLaVA-1.5-7B, the bias improves accuracy on every task, and the gain is larger on grounding than on VQA. It is smaller on Qwen3-VL-8B than on Qwen2.5-VL-7B because the pruned Qwen3-VL-8B degrades less without the bias (43.5 against 26.8 on grounding at 9.9%), which leaves less for the bias to recover; this may relate to DeepStack, through which Qwen3-VL-8B injects visual features from several encoder layers into the language model (Appendix C.5). On InternVL3.5-8B the bias slightly lowers the accuracy (-0.4 on grounding and -1.0 on VQA). This backbone splits an image into tiles and appends a thumbnail, and the attention that pruning withdraws falls on the tiles, whose content the thumbnail already provides (Appendix C.7); we therefore disable the bias on this backbone. On LLaVA-1.5-7B the bias adds 7.7 points on grounding (22.7 to 30.4). Like Qwen2.5-VL-7B, it has no other visual pathway such as DeepStack or a thumbnail, so the pruned model degrades more without the bias and leaves more for it to recover.

Table 25: **CAR with and without the bias across backbones**, at the lowest retention ratio of each task. On InternVL3.5-8B the bias is computed separately for the thumbnail and the other tiles.

Backbone	Task	Retained	w/o bias	w/ bias
Qwen2.5-VL-7B	VG	9.9%	26.8	66.8 (+40.0)
	VQA	9.9%	78.7	79.3 (+0.6)
Qwen3-VL-8B	VG	9.9%	43.5	46.6 (+3.1)
	VQA	11.1%	69.8	70.6 (+0.8)
	Video	9.9%	60.7	62.0 (+1.3)
InternVL3.5-8B	VG	9.9%	76.2	75.8 (−0.4)
	VQA	11.1%	74.4	73.4 (−1.0)
LLaVA-1.5-7B	VG	11.1%	22.7	30.4 (+7.7)

Table 26: **Comparison with RESTORE (Cho et al., 2026) on VQA with Qwen2.5-VL-7B.** Each calibration, RESTORE's and ours, is applied on top of both selectors.

Method	TextVQA	ChartQA	AI2D	OCRBench	HallBench	MME	MMB-EN	MMB-CN	Acc.	Rel.
Upper Bound, All 1,296 Tokens (100%)										
Qwen2.5-VL-7B	84.3	87.1	81.6	867	56.2	2342.3	83.1	82.9	84.9	100.0%
Retain 128 Tokens (↓90.1%)										
DivPrune (CVPR25), no calibration	65.9	51.4	72.7	535	43.2	2044.6	77.3	76.7	67.9	79.7%
+ RESTORE merging	65.8	51.2	73.9	582	43.6	2156.4	79.1	78.6	69.8	81.8%
+ RESTORE merging and bias	65.3	51.2	73.8	578	43.3	2183.3	79.6	79.6	70.0	81.9%
+ our bias	66.2	53.2	73.7	549	44.2	2119.2	78.5	77.9	69.3	81.4%
Ours, no calibration	78.5	74.6	79.1	732	48.8	2270.0	80.7	81.0	78.7	92.3%
+ RESTORE merging	74.0	64.6	78.2	665	47.3	2242.4	79.8	80.1	75.3	88.4%
+ RESTORE merging and bias	73.6	63.5	78.2	654	46.8	2255.4	80.8	80.3	75.2	88.1%
+ our bias (**CAR**)	**78.8**	**76.1**	**79.4**	**731**	**49.0**	**2314.3**	**81.1**	**81.3**	**79.3**	**93.0%**

Comparison with RESTORE. Here we compare with RESTORE, which also calibrates attention after pruning. RESTORE splits the token budget in half: one half is selected by the underlying pruning method and the other half is formed by merging the tokens it discards, each merged token receiving on its logit the logarithm of the number of tokens it absorbs. Since RESTORE ties this merging to its calibration and applies to different selectors, Tables 26 and 27 compare, on DivPrune and on our selector, its merging alone, its merging together with its bias, and our bias (on VQA the capped per-cell bias, whose magnitude follows the number of discarded tokens each retained token stands for under the similarity, an assignment used only for counting; on grounding the uniform bias).

From Tables 26 and 27 we observe the following. (1) On question answering, the merging of RESTORE gains 1.9 points on DivPrune and costs 3.4 points on our selector. The effect of the merging depends on the selector: DivPrune selects by diversity alone and does not consider importance, so the discarded tokens still hold information worth recycling; our selection already weights tokens by saliency, and what can be recycled does not repay the half of the budget that is given up. On grounding the merging costs accuracy on both selectors. The merging groups tokens by feature similarity, so the members of a merged token may lie far apart on the grid, which breaks the spatial structure that grounding depends on. (2) The bias of RESTORE behaves very differently on the two tasks. On question answering it has almost no effect, whereas on grounding it gains 19.5 and 30.6 points. Grounding tasks are more sensitive to pruning, and restoring the visual attention share helps the model locate answers within the image. (3) Our bias gains on both tasks and on both selectors, and on three of the four combinations it is clearly better than the full form of RESTORE. On question answering with DivPrune, our bias comes close to the gain that their merging brings.

C.11 Selection Rule

Selection Mechanisms. To assess the weighted coverage selection, we compare it with other selection rules under the same setting: every rule receives the same saliency w from the probed head, the same budget, the same bias and the same feature space, and every rule builds the retained set greedily, one token at a time, so that they differ only in what each step selects on. The groups follow Tables 28 and 29. *No diversity signal*: ① the k tokens of largest saliency are kept, and the redundancy among them is not considered (FastV, SparseVLM (Chen et al., 2024a; Zhang et al., 2025b)). *Marginal penalty relative to the selected set*: these rules compute at each step the largest similarity

Table 27: **Comparison with RESTORE (Cho et al., 2026) on visual grounding with Qwen2.5-VL-7B.** Each calibration, RESTORE's and ours, is applied on top of both selectors.

Method	RefCOCO			RefCOCO+			RefCOCOg		Avg.	Rel.
	testA	testB	val	testA	testB	val	test	val		
Upper Bound, All Visual Tokens (100%)										
Qwen2.5-VL-7B	92.2	84.8	89.4	87.6	74.1	82.8	86.7	86.7	85.5	100.0%
Retain 9.9% of Visual Tokens										
DivPrune (CVPR25), no calibration	27.8	19.6	24.2	24.7	15.8	20.9	21.3	21.5	22.0	25.6%
+ RESTORE merging	15.7	13.4	13.9	13.9	11.4	12.1	12.3	12.4	13.1	15.3%
+ RESTORE merging and bias	37.1	31.8	35.7	33.4	26.7	30.6	33.4	31.8	32.6	38.0%
+ our bias	51.5	42.5	46.9	46.1	35.5	40.8	46.3	45.4	44.4	51.8%
Ours, no calibration	33.9	24.5	29.2	30.2	20.9	25.4	25.2	25.0	26.8	31.2%
+ RESTORE merging	16.8	13.8	16.2	15.1	12.0	14.5	13.2	13.0	14.3	16.7%
+ RESTORE merging and bias	52.1	45.1	48.1	46.7	37.3	41.9	45.2	43.1	44.9	52.4%
+ our bias (**CAR**)	**75.4**	**66.7**	**72.2**	**67.2**	**55.9**	**62.3**	**67.8**	**66.5**	**66.8**	**77.9%**

between a candidate and the selected set. ② Taking the candidate whose largest similarity is smallest maximizes the diversity among the retained tokens, and the saliency is not used (DivPrune (Alvar et al., 2025)). ③ Subtracting that similarity from the saliency, with λ setting the weight of the two terms, takes the candidate that is salient and at the same time unlike the tokens already selected (IDPruner (Carbonell & Goldstein, 1998; Tan et al., 2026)). ④ Multiplying the saliency by the dissimilarity to the selected set keeps a token only when both are high (AnchorPrune (Oh & Kim, 2026; Xu et al., 2026)). *Set-function maximization*: ⑤ a kernel is built from the saliency and the similarity, and its log-determinant measures the volume the retained tokens span, so it scores importance and redundancy at once. Each step takes the largest gain in that volume, which keeps the retained set salient and non-redundant (CDPruner (Zhang et al., 2025a)). ⑥ The saliency of the retained tokens is instead added to the log-determinant of their feature Gram matrix, with α weighting the two terms (CoIn (Du et al., 2026)). *Hard suppression*: ⑦ tokens are scanned in order of saliency and, after each selection, every candidate whose similarity to it exceeds a threshold is removed (AgilePruner (Baek et al., 2026)); ⑧ a part of the budget first goes to pivots ranked by saliency. A graph is then built from feature similarity fused with spatial adjacency, and every time a token is selected its neighbours on the graph are removed, until k tokens have been chosen (D^2Pruner (Zhang et al., 2026b)). *Weighted coverage selection*: these rules take at each step the largest gain in how well the retained set covers the whole image. ⑨ The gain counts every covered token equally and is multiplied by the saliency of the candidate (SCOPE (Deng et al., 2025)); ⑩ ours weights each covered token by its saliency instead, so a candidate scores high when it covers many salient tokens, and multiplies the similarity by a spatial kernel (Eq. 8) so that tokens far apart on the grid do not cover each other.

Full Results of the Selection-Rule Ablation. Tables 28 and 29 give the detailed results of the selection-rule ablation on grounding and on VQA, of which Table 6 reports the averages. Three observations follow. (1) The coverage-based rules, ⑨ and ⑩, give the best average over the two tasks. Their objective is how well the retained set covers all the visual tokens, while the other rules examine only the similarity among the retained tokens. Ours is the best on every grounding split, where the spatial kernel adds to the coverage, and attains the highest VQA average. (2) Using the saliency together with the diversity is in general better than using either alone. The rule that uses the diversity alone, ②, is clearly below the rest, and when the importance of a token is not considered the tokens the task needs may not be retained. The rule that uses the saliency alone, ①, is not the best on either task either, since the tokens it selects are often redundant and the retained set tends to concentrate on a few regions. (3) The relative order of the rules is not the same on the two tasks. On grounding the rules that suppress by a threshold, ⑦ and ⑧, stand above ③, ④ and ⑤, as grounding depends on the spatial structure of the image and removing similar tokens entirely helps spread the retained tokens over different regions. On VQA the rules other than ② and ⑥ lie close together, and the suppression rules fall even below ①, as text and charts consist of many mutually similar tokens and a threshold may remove the salient ones among them as a group.

C.12 Error Decomposition

This section describes how the two errors in Figure 3 are measured. The model itself is not pruned during the measurement. We run one full forward pass of the unpruned model and, at every layer,

Table 28: **Selection rule ablation on visual grounding with Qwen2.5-VL-7B** (three test splits, 9.9% of the visual tokens retained). Every row uses the uniform bias $\log(N/k)$ of Eq. 12. For our row, s_{ij} is the similarity of Eq. 8, with the spatial kernel, as deployed; among the other rules only D^{2}Pruner carries a geometric term.

	Score or objective	Representative	RefCOCO$_{\text{testA}}$	RefCOCO+$_{\text{testB}}$	RefCOCOg$_{\text{test}}$
	No Diversity Signal				
①	w_i	FastV	65.1	50.5	60.1
	Marginal Penalty Relative to the Selected Set				
②	$\min_{j\in S}(1-s_{ij})$, w unused	DivPrune	51.5	35.5	46.3
③	$\lambda w_i-(1-\lambda)\max_{j\in S} s_{ij}$	IDPruner	65.5	51.1	60.7
④	$w_i^{\alpha}\min_{j\in S}(1-s_{ij})$	AnchorPrune	66.9	50.0	61.8
	Set-Function Maximization				
⑤	$\log\det K_S$, $K_{ij}=w_i w_j s_{ij}$	CDPruner	65.8	51.0	61.8
⑥	$(1-\alpha)w_i+\alpha\,\Delta_i\log\det(X_S X_S^{\top}+\lambda I)$	CoIn	58.3	44.0	55.2
	Hard Suppression				
⑦	w_i subject to $\max_{j\in S} s_{ij}<\tau$	AgilePruner	68.1	51.0	62.5
⑧	Pivots by w_i, then graph suppression	D^{2}Pruner	68.9	53.3	62.1
	Weighted Coverage Selection				
⑨	$w_i\sum_{l\in V}\left[s_{il}-\max_{j\in S} s_{jl}\right]_+$	SCOPE	70.7	52.3	63.2
⑩	$\sum_{i\in V} w_i\max_{j\in S} s_{ij}$ **(ours)**	—	**75.4**	**55.9**	**67.8**

Table 29: **Selection rule ablation on VQA with Qwen2.5-VL-7B** (128 of 1296 visual tokens retained). No row uses the bias. For our row, s_{ij} is the similarity of Eq. 7, without the spatial kernel, as deployed.

	Score or objective	Representative	TextVQA	ChartQA	AI2D	OCRBench	HallBench	MME	MMB-EN	MMB-CN	Avg
	No Diversity Signal										
①	w_i	FastV	76.8	73.0	78.6	724	49.9	2278.1	80.1	80.4	78.1
	Marginal Penalty Relative to the Selected Set										
②	$\min_{j\in S}(1-s_{ij})$, w unused	DivPrune	65.8	51.4	72.7	533	42.7	2041.2	77.4	76.7	67.8
③	$\lambda w_i-(1-\lambda)\max_{j\in S} s_{ij}$	IDPruner	77.3	73.4	78.6	733	49.3	2278.1	80.7	80.4	78.4
④	$w_i^{\alpha}\min_{j\in S}(1-s_{ij})$	AnchorPrune	77.3	75.4	79.1	736	48.2	2282.7	80.6	80.8	78.6
	Set-Function Maximization										
⑤	$\log\det K_S$, $K_{ij}=w_i w_j s_{ij}$	CDPruner	77.8	73.3	79.2	725	48.4	2250.6	80.3	80.7	78.1
⑥	$(1-\alpha)w_i+\alpha\,\Delta_i\log\det(X_S X_S^{\top}+\lambda I)$	CoIn	76.0	64.6	77.0	650	46.7	2170.3	78.4	78.3	74.3
	Hard Suppression										
⑦	w_i subject to $\max_{j\in S} s_{ij}<\tau$	AgilePruner	76.2	69.7	77.7	692	46.4	2211.0	79.8	79.3	76.1
⑧	Pivots by w_i, then graph suppression	D^{2}Pruner	75.6	69.5	78.9	685	49.4	2271.9	80.2	80.9	77.1
	Weighted Coverage Selection										
⑨	$w_i\sum_{l\in V}\left[s_{il}-\max_{j\in S} s_{jl}\right]_+$	SCOPE	77.8	73.0	78.4	734	48.7	2276.6	80.7	81.4	78.4
⑩	$\sum_{i\in V} w_i\max_{j\in S} s_{ij}$ **(ours)**	—	78.5	74.6	79.1	732	48.8	2270.0	80.7	81.0	78.7

compute two outputs from that layer's input: the output O of the full attention, and the output O' obtained when only the tokens S selected by the method are retained and the bias b is added. For methods that prune inside the decoder, S and the bias $\log(N/k_l)$ both follow the tokens the method actually retains at layer l. We then split $O'-O$ into the two errors of Eq. 5. Since the input of every layer comes from the unpruned model, the measurement captures the error introduced by pruning at that layer alone, without the drift accumulated in earlier layers. Each error is aggregated over layers 13 to 19, the intermediate layers marked by the band in Figure 1, and reported as its squared norm relative to that of O, in per cent. The samples are drawn from subsets of the RefCOCO, RefCOCO+ and RefCOCOg validation sets, 100 from each, 300 in total.

C.13 Details of the Motivating Observations

This section gives the setup behind the three groups of panels of Figure 1.

Grounding across retention ratios (panel a). We evaluate Qwen2.5-VL-7B (Bai et al., 2025b) on the eight splits of RefCOCO, RefCOCO+ and RefCOCOg (Yu et al., 2016; Mao et al., 2016). The panel reports the aggressive retention ratio of 9.9%, without and with the bias. The nine published methods are grouped by where the signal that decides the retained set is read: encoder-side (VisionZip (Yang et al., 2025), DivPrune (Alvar et al., 2025), AnchorPrune (Oh & Kim, 2026), EADP (Wang et al., 2026b)); decoder-side (FastV (Chen et al., 2024a), SparseVLM (Zhang et al., 2025b), HAWK (Zhu et al., 2026)); and two-stage (VScan (Zhang et al., 2026a), Nüwa (Huang et al., 2026)). We add two simple baselines that carry no selection signal, uniform random sampling and average pooling, which merges neighbouring visual tokens into k averaged ones. The bias is the

Table 30: **Grounding on Qwen2.5-VL-7B at a moderate and an aggressive retention ratio.** Δ is the gain of the uniform bias on the eight-split average.

Method	RefCOCO			RefCOCO+			RefCOCOg		Avg.	Δ
	testA	testB	val	testA	testB	val	test	val		
Upper Bound, All Visual Tokens (**100**%)										
Qwen2.5-VL-7B	92.2	84.8	89.4	87.6	74.1	82.8	86.7	86.7	85.5	–
Moderate Budget, Retain **25**% *of Visual Tokens*										
Random	63.9	52.1	58.3	58.1	44.5	52.5	52.7	52.7	54.4	–
+bias	70.1	63.4	67.8	63.6	53.9	60.6	62.9	63.0	63.2	+8.8
Pooling	64.1	49.1	57.5	58.8	42.0	51.6	50.7	50.3	53.0	–
+bias	76.9	66.8	73.4	70.7	56.7	66.2	68.7	68.1	68.4	+15.4
VisionZip (CVPR25)	71.0	59.0	66.7	65.0	50.5	59.1	61.2	61.2	61.7	–
+bias	77.4	71.4	75.5	70.8	60.8	66.8	72.8	72.7	71.0	+9.3
DivPrune (CVPR25)	74.1	59.8	67.5	68.6	51.9	60.9	63.3	63.7	63.7	–
+bias	81.2	70.6	76.9	76.1	60.8	69.0	74.3	74.5	72.9	+9.2
AnchorPrune (ECCV26)	75.6	62.4	68.3	68.4	53.2	61.5	64.7	65.5	64.9	–
+bias	81.6	72.9	77.1	75.1	62.9	69.0	74.8	76.2	73.7	+8.8
EADP (ECCV26)	71.9	53.1	63.3	68.0	44.4	57.2	56.9	56.3	58.9	–
+bias	79.6	66.3	74.4	74.9	55.9	67.3	69.7	69.8	69.7	+10.8
FastV (ECCV24)	23.7	30.7	28.4	21.6	25.6	24.1	29.1	29.2	26.6	–
+bias	27.7	37.8	33.9	25.5	32.7	29.5	36.8	37.5	32.7	+6.1
SparseVLM (ICML25)	30.9	31.7	31.3	26.4	25.3	24.9	29.3	29.9	28.7	–
+bias	39.4	40.7	40.7	34.3	32.4	34.3	39.5	39.1	37.6	+8.9
HAWK (CVPR26)	40.4	45.7	42.7	36.0	40.0	38.0	44.9	45.7	41.7	–
+bias	46.4	53.1	49.3	42.3	45.7	44.2	53.8	54.6	48.7	+7.0
VScan (TMLR26)	78.7	66.9	71.7	72.3	55.5	64.0	67.9	68.2	68.2	–
+bias	84.3	74.8	80.4	78.4	63.5	72.2	77.5	77.5	76.1	+7.9
Nüwa (ICLR26)	77.5	72.5	75.6	69.7	60.3	65.8	72.8	72.3	70.8	–
+bias	81.5	74.3	77.5	73.7	61.2	68.2	74.7	75.0	73.3	+2.5
Ours w/o bias	79.8	66.6	75.3	73.0	58.5	68.1	67.5	67.9	69.6	–
CAR (Ours)	87.0	80.1	84.3	80.2	68.3	75.9	79.4	79.6	79.4	+9.8
Aggressive Budget, Retain **9.9**% *of Visual Tokens*										
Random	23.1	17.2	21.5	20.3	14.5	18.6	17.3	18.1	18.8	–
+bias	44.0	37.6	42.9	38.3	31.7	37.0	39.0	38.4	38.6	+19.8
Pooling	20.0	15.6	17.5	18.2	13.0	15.5	14.8	14.2	16.1	–
+bias	58.7	46.7	53.6	51.5	38.8	46.7	48.0	48.9	49.1	+33.0
VisionZip (CVPR25)	23.3	19.2	20.5	18.8	16.0	17.7	18.4	19.5	19.2	–
+bias	41.3	42.4	42.4	35.6	35.5	36.1	40.8	41.8	39.5	+20.3
DivPrune (CVPR25)	27.8	19.6	24.2	24.7	15.8	20.9	21.3	21.5	22.0	–
+bias	51.5	42.5	46.9	46.1	35.5	40.8	46.3	45.4	44.4	+22.4
AnchorPrune (ECCV26)	24.4	18.0	22.0	21.4	16.2	19.4	19.7	20.2	20.2	–
+bias	46.8	44.2	44.2	40.8	37.3	39.2	44.8	45.7	42.9	+22.7
EADP (ECCV26)	24.1	16.5	21.2	23.4	14.6	18.6	17.4	18.2	19.2	–
+bias	51.0	39.3	47.8	46.2	34.6	41.7	42.3	43.3	43.3	+24.1
FastV (ECCV24)	3.2	2.7	3.2	3.3	2.9	3.5	3.6	3.3	3.2	–
+bias	5.7	3.9	5.6	6.1	4.3	4.9	5.8	5.5	5.2	+2.0
SparseVLM (ICML25)	5.7	5.7	5.8	3.8	4.2	3.8	5.4	5.3	5.0	–
+bias	7.8	7.4	7.7	4.8	5.1	5.1	7.0	6.8	6.5	+1.5
HAWK (CVPR26)	8.1	8.6	7.5	6.9	7.0	7.0	8.5	7.5	7.6	–
+bias	9.9	13.1	12.4	9.0	11.4	11.0	14.9	14.4	12.0	+4.4
VScan (TMLR26)	23.8	19.8	21.4	20.4	15.5	18.0	18.5	20.5	19.7	–
+bias	48.0	40.5	43.8	42.0	31.6	36.4	41.6	42.0	40.7	+21.0
Nüwa (ICLR26)	27.5	23.7	26.2	21.8	17.4	20.5	23.0	22.0	22.8	–
+bias	48.2	43.9	47.6	40.5	35.5	38.7	43.6	44.1	42.8	+20.0
Ours w/o bias	33.9	24.5	29.2	30.2	20.9	25.4	25.2	25.0	26.8	–
CAR (Ours)	75.4	66.7	72.2	67.2	55.9	62.3	67.8	66.5	66.8	+40.0

uniform $\log(N/k)$ of Eq. 12, added to the retained visual tokens only and leaving the set that the selector retains unchanged.

Table 30 gives the per-split numbers at a moderate retention ratio of 25% and at 9.9%, each without and with the bias. At 25% the encoder-side and two-stage methods are clearly ahead of the two baselines, while the decoder-side ones stay below them at either ratio; at 9.9% the lead of the best method, Nüwa, over random sampling falls from 16.4 points to 4.0. The bias improves every method at both ratios; at 9.9% it raises the published methods by up to 24.1 points and the pooling baseline by 33.0. With our bias, the simple pooling baseline reaches 49.1 at 9.9% and so outperforms existing complex pruning designs, the best of them being DivPrune at 44.4. CAR is the best at both ratios, 79.4 and 66.8, ahead of the strongest biased baseline by 3.3 and 17.7 points.

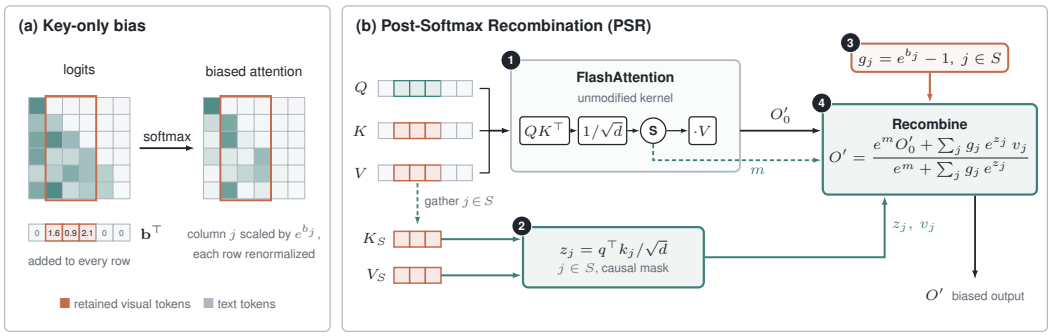


Figure 6: (a) A key-only bias rescales the retained columns of the attention. (b) The four steps of PSR around an unmodified FlashAttention.

Visual attention share across decoder layers (panels b to d). At every decoder layer of Qwen2.5-VL-7B we measure σ, the share of the attention of text tokens that falls on visual tokens, averaged over heads and over text queries. Panel (b) gives σ in the unpruned model on 100 RefCOCO val images and 100 TextVQA images: on both tasks it peaks in layers 13 to 19, the layers where text tokens attend to the image most. Panels (c) and (d) prune the 100 RefCOCO val images to a retention ratio of 9.9% and report the ratio σ'/σ of Eq. 4, the share after pruning over the share the unpruned model gives on the same image. They compare CAR with DivPrune, HAWK, uniform random sampling and average pooling, without any bias in (c) and, in (d), with the uniform bias of Eq. 12 added to the tokens each method retains. In layers 13 to 19, pruning lowers the share to between 0.51 and 0.64 of its unpruned value, and the bias raises it to between 0.70 and 0.94.

Saliency source (panels e and f). In panels (e) and (f) we hold the token selection fixed to our weighted coverage selection of Section 2.2 and vary only the source that supplies the saliency w. For Qwen2.5-VL-7B the candidates are the 16 attention heads of the final encoder layer and their average. We report grounding accuracy on 500 RefCOCO val images at a 9.9% retention ratio and question answering accuracy averaged over the eight benchmarks at 128 tokens. There is significant variation in importance among different attention heads, with accuracy differences exceeding 30 percentage points on the grounding task. Furthermore, attention heads exhibit task dependency; the head performing best on one task does not necessarily excel on another: head 13 leads in the grounding task, while head 2 leads in the question-answering task. For InternVL3.5-8B, the late layers develop attention sinks (Darcet et al., 2024; Luo et al., 2026), where a few sink tokens absorb most of the attention (Appendix C.3), which leaves the final layer the worst source. We therefore compare the layers that supply w_i, averaging over the heads of each, and scan them at an 11.1% retention ratio on 500 samples of each of the two question answering benchmarks. The results show that performance on both TextVQA and ChartQA peaks at the intermediate layers, which the band marks.

D Efficiency Analysis

D.1 Compatibility with FlashAttention

To maintain compatibility with optimized attention kernels such as FlashAttention (Dao, 2024), which never materialize the attention matrix, we propose Post-Softmax Recombination (PSR) to apply the logit bias of Eq. 2 without modifying the kernel. PSR builds on a standard practice of attention kernels: besides the attention output, the kernel returns the log-sum-exp of the logits, which suffices to add further terms to the softmax outside the kernel (Dao, 2024; Ye et al., 2025). PSR therefore runs the unmodified kernel and adds the effect of the bias in such a recombination.

This recombination applies because our bias is key-only: b_j depends on the retained token $j \in S$ but not on the query, and text tokens receive $b_t = 0$. As shown in Figure 6(a), the bias thus scales each retained column of the attention by e^{b_j} and renormalizes each row, which only adds extra terms for the k retained tokens to the softmax.

Table 31: **Cost of the bias in isolation** (Qwen2.5-VL-7B, 128 of 1,296 visual tokens). The two configurations retain the same tokens and differ only in the bias.

	w/o bias	+ bias (ours)	Difference
Prefill (ms)	241.6	246.6	+5.0
Decode (ms/token)	27.2	27.3	+0.1
Latency (ms)	299.1	304.7	+5.6
Peak memory (GB)	15.82	15.82	<0.01

Figure 6(b) shows the four steps of PSR for a query q with logits z_u. In step 1, we call the unmodified FlashAttention, which returns the unbiased output O'_0, that is, Eq. 2 with $b = 0$, together with the log-sum-exp m of the logits:

$$O'_0,\ m = \mathrm{FlashAttn}(Q, K, V), \qquad m = \log \sum_{u \in T \cup S} e^{z_u}, \tag{44}$$

so that e^m and $e^m O'_0$ are the denominator and numerator of the unbiased softmax. In step 2, we gather the keys K_S and values V_S of the retained tokens and compute their logits $z_j = q^\top k_j / \sqrt{d_h}$ under the kernel's causal mask. In step 3, we write $e^{b_j} = 1 + g_j$ for $j \in S$, so that the bias adds only the extra weight $g_j e^{z_j}$ of each retained token to the softmax. In step 4, we recombine these quantities into the biased output:

$$O' = \frac{e^m\, O'_0 + \sum_{j \in S} g_j\, e^{z_j}\, v_j}{e^m + \sum_{j \in S} g_j\, e^{z_j}}. \tag{45}$$

Since the extra terms involve only the retained tokens, PSR adds computation only on these k columns.

D.2 Complexity Analysis

Our method adds computation in two places. (1) The first is the token selection that runs after the vision encoder and before the language model. It computes the similarity matrix with a matrix multiplication of complexity $O(N^2 d)$, which dominates this stage. The greedy pass performs k steps and, since the gain of a candidate is a sum over all the visual tokens, every step rescans the matrix, $O(kN^2)$. The saliency requires one explicit attention of the head h^* over the patches, $O(P^2 d_h)$. For the per-cell bias, the values of the bias require the assignment of the discarded tokens to the retained ones, $O((N-k)kd)$. (2) The second is the prefill, where the bias is added to the logits at every decoder layer: the recombination of Eq. 45 acts on the retained columns of each layer, $O(Lnkd)$.

However, this overhead is negligible compared to the total computational cost of the model. As shown in Table 32, these steps require 24.2 GFLOPs for Qwen2.5-VL-7B, only 0.6% of the 3.8 TFLOPs the pruned model spends, and the decode time is unchanged, 27.3 against 27.1 ms in Table 4. The greedy pass is a sequence of k steps, so the selection takes 13.9 ms against 1.6 ms for top-k selection by the saliency, which is why our prefill is longer than that of the methods built on it.

We further isolate the cost of the bias itself. The two configurations of Table 31 use the same selection and retain the same tokens, so they differ only in whether the bias is applied. Applying it adds 5.0 ms to the prefill and 5.6 ms to the end-to-end latency, and leaves the peak memory unchanged. The decode time differs by 0.1 ms per token, less than the variation between repeated runs, because the bias enters the prefill only and the decoding steps run the unmodified kernel. Of the 5.0 ms, the assignment of the discarded tokens takes 0.5 ms and the recombination of Eq. 45 across the 28 layers takes the remaining 4.5 ms. The bias therefore adds 1.9% to the end-to-end latency, all of it in the prefill.

We measure the efficiency of Table 4 on the first 200 questions of TextVQA using one RTX 3090 with batch size 1 in `bfloat16`. The language model and the vision encoder run FlashAttention-2 for every method, and our bias is injected through PSR (Appendix D.1). Its FLOPs column is theoretical, obtained by applying $4nd^2 + 2n^2 d + 2ndm$ to the token count measured at each layer of the vision encoder and the language model.

Table 32: **Cost of the steps our method adds** (Qwen2.5-VL-7B, 128 of 1,296 visual tokens). N visual tokens, k retained, d the hidden size, $P = 4N$ patches before the merger, d_h the encoder head dimension, n the prompt length, L the decoder layers.

Step	Complexity	FLOPs
Saliency from the probed head	$O(P^2 d_h)$	4.3 G
Similarity matrix	$O(N^2 d)$	12.0 G
Spatial kernel (grounding only)	$O(N^2)$	<0.1 G
Greedy coverage, k steps	$O(kN^2)$	0.6 G
Assignment for the bias	$O((N-k)kd)$	1.1 G
Bias in prefill, L layers	$O(Lnkd)$	6.2 G
Total		**24.2 G**

Selection Cost at Large N. The greedy pass rescans the $N \times N$ similarity matrix at every step, so it costs $O(kN^2)$, which becomes noticeable when N is large, as for Qwen3-VL-8B selecting jointly over 32 video frames ($N = 11{,}520$) or InternVL3.5-8B with up to 24 tiles ($N = 6{,}144$). We change only how the gains are computed. When a token is selected, only the few tokens that are similar to it and not yet well covered, about a dozen at the median in our measurements, gain coverage, so each step updates the gains by their contribution alone instead of rescanning the matrix. This lowers the selection time on video from 5.07 s to 0.61 s and the extra peak memory from 1.52 to 1.19 GB, and on InternVL3.5-8B with 24 tiles from 814 to 279 ms. For $N < 3{,}000$ the fixed cost of each incremental step dominates, so there we recompute all gains at every step.

E Qualitative Results

In this section we supplement the results with four groups of visualizations: the tokens different selection rules retain, the effect of the attention bias, the saliency read from different encoder layers of InternVL3.5-8B, and failure cases of the deployed configuration.

Retained Tokens and Cells. Figure 7 compares three selection rules: ranking tokens by w (the importance-ranking rule of Table 6), facility location with $w = 1$, and our saliency-weighted facility location. Ranking by w concentrates the retained tokens on the regions of highest saliency, where the tokens a task requires may be left unrepresented, while facility location with $w = 1$ distributes them uniformly over the image and disregards the saliency. Our rule covers the image and at the same time allocates more tokens to the salient regions, and it answers both questions correctly. The last column shows the assignment of each discarded token to its representative, which induces a partition of the image into cells. Every region of the image is thereby represented, and the cells are larger in the uniform regions and smaller where the saliency concentrates. The number of tokens each retained token represents is the count $|C_j|$ on which the per-cell bias of Section 2.3 is defined.

Effect of the Attention Bias. With the retained set held fixed, Figure 8 compares the attention that each retained token receives in the pruned model without and with the bias. Without bias, the retained tokens receive 0.57 of the share they receive in the unpruned model, and the predicted box covers the wrong sheep. The bias restores the share to 0.85, the retained tokens on the referred sheep are attended again, and the predicted box returns to the target.

Saliency Layers on InternVL3.5-8B. Figure 9 shows the attention of three encoder layers of InternVL3.5-8B and the tokens retained when the saliency is read from each of them. The attention sinks intensify with depth: at layer 23 a few tokens absorb most of the attention, so the saliency carries little information about the content and the retained tokens no longer cover what the task needs. They spread over the background, the letters keep few of them and the model misreads the word. At layer 10, the layer the probe selects on VQA, the saliency follows the strokes of the letters, the retained tokens sit on them and the model reads the word. Layer 12, where the concentration sets in, falls in between.

Failure Cases. Figure 10 presents one case of each failure mode. The first is the loss of fine detail under aggressive pruning: the target region is retained and attended, yet the detail on which the answer depends is no longer resolved. In the VQA case the clock is retained, while the position of

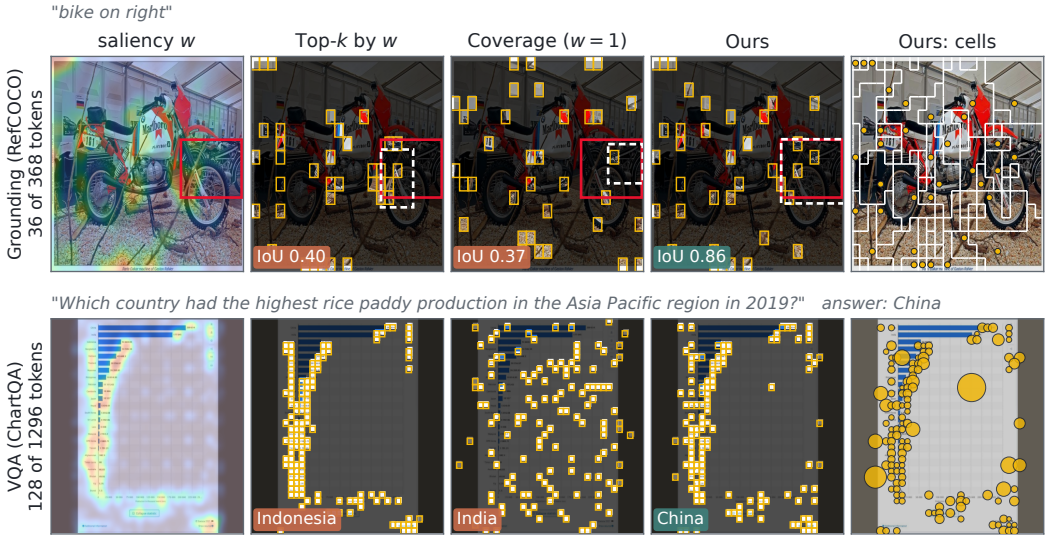


Figure 7: **Retained tokens under three selection rules.** Qwen2.5-VL-7B on RefCOCO at 9.9% of the tokens and on ChartQA at 128 tokens. Discarded tokens are darkened and retained ones outlined; red is the ground truth, dashed the prediction, and the label gives the IoU or the answer. The last column shows which retained token represents each discarded one: cell boundaries on grounding, and on VQA each representative with an area proportional to the tokens it represents.

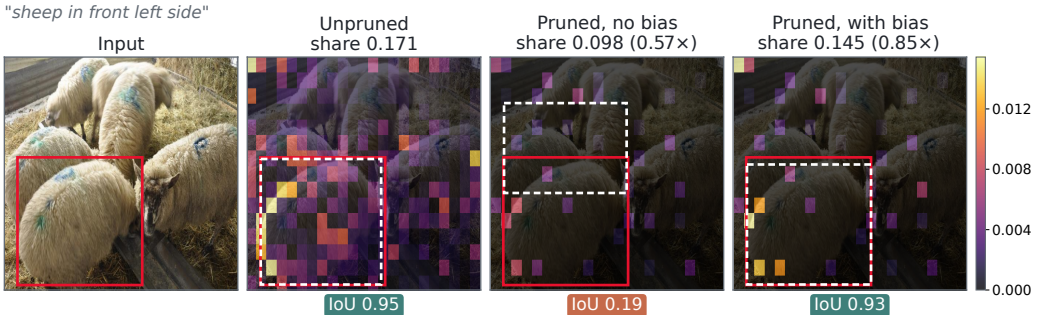


Figure 8: **Effect of the attention bias under a fixed retained set** (Qwen2.5-VL-7B, RefCOCO at 9.9%). Color encodes the attention a visual token receives from the text, averaged over heads, text queries and decoder layers 13 to 19. Discarded tokens are darkened, the ground truth is in red and the prediction is dashed. *Share* denotes the fraction of the text attention falling on the image, with its ratio to the unpruned model in brackets.

its hands is not, and the model reports the wrong time. The second is the saliency source, which does not always respond to the region the task requires. In the grounding case the saliency is concentrated on the game boxes and assigns little weight to the referred book, so a single token is retained there and the prediction falls elsewhere.



Figure 9: **Saliency from three encoder layers of InternVL3.5-8B and the tokens it retains.** OCRBench, 142 of 1,280 tokens kept; the image is cut into 4×1 tiles (white lines) plus a thumbnail tile, shown in the bottom row as saliency and retained tokens. The label above the second row is the answer. The attention sinks grow with depth: for this image, the class token receives 3%, 19% and 36% of the encoder attention at layers 10, 12 and 23, and the top 1% of patches receive 4%, 8% and 25%.

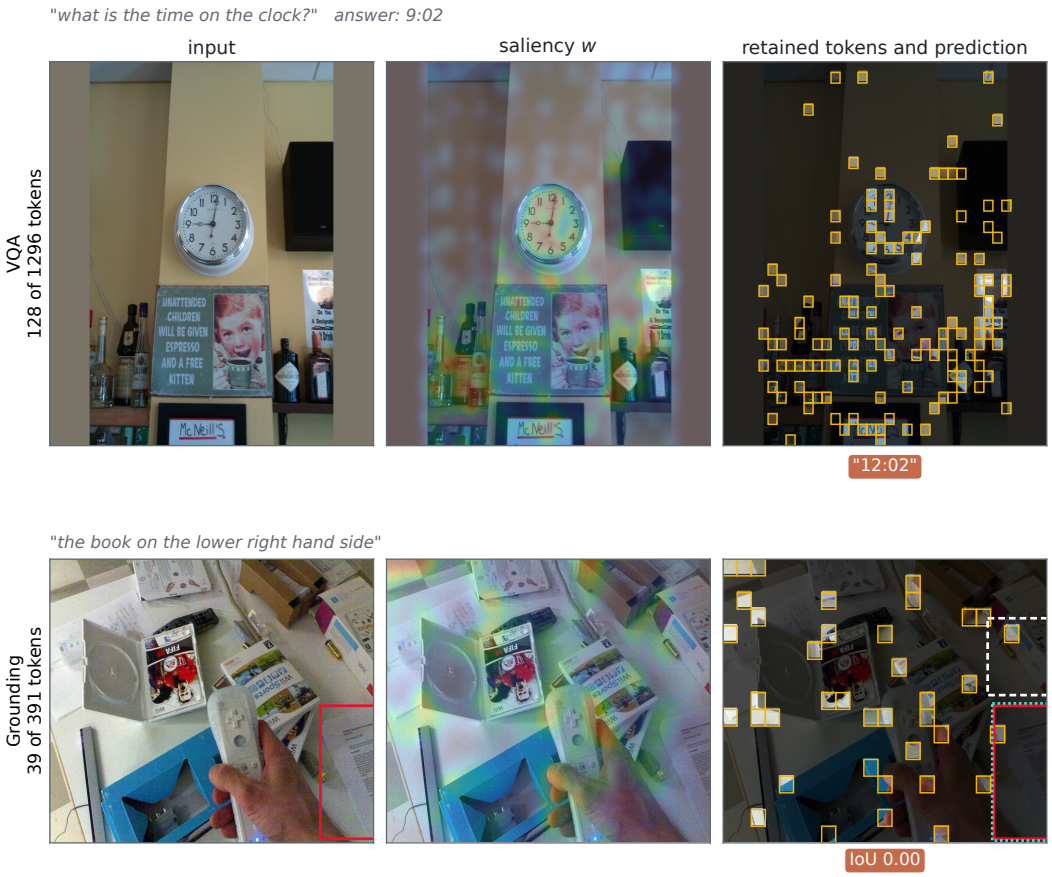


Figure 10: **Two failure cases** (Qwen2.5-VL-7B; the unpruned model answers both correctly). Left to right: the input, the saliency w, and the retained tokens. Top, VQA: the clock is retained and salient, yet the position of the hands is not resolved. Bottom, grounding: the ground truth is in red, our prediction is the white dashed box and the unpruned prediction the dotted one; the saliency misses the referred book, which keeps a single token.