WHERE TO LOOK, WHAT TO KEEP: TWO-PHASE LEARNING FOR GROUNDING-AWARE VISUAL TOKEN SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) encode images into hundreds or thousands of visual tokens, incurring substantial inference overhead. Although visual token pruning mitigates this cost, aggressive pruning can impair visual grounding, which requires both recognizing the referred target and predicting its spatial coordinates. We diagnose this degradation with budget-matched token-addition experiments and find that existing pruners either lose focus on the target or discard the spatial context needed for grounding. These findings motivate two complementary learning objectives: learning query-conditioned target relevance (*where to look*) and optimizing the retained subset for grounding accuracy (*what to keep*). We implement these objectives through a two-phase framework that trains a lightweight, query-conditioned token scorer on frozen MLLM backbones. In Phase One, **Localization Prior Learning (LPL)** uses bounding-box supervision to teach the scorer *where to look*. In Phase Two, **Grounding-Guided Selection Refinement (GSR)** optimizes token selections through direct grounding feedback, teaching the scorer *what to keep* by rewarding subsets that yield accurate bounding-box predictions. At an 11.1% visual-token retention ratio, our method retains 96.6% of unpruned grounding accuracy on Qwen2.5-VL-7B and exceeds the unpruned LLaVA-1.5-7B (117.6%), while retaining 98.6% and 98.2% of general VQA performance across ten benchmarks, respectively. Furthermore, our approach achieves up to $1.54\times$ end-to-end speedup and $1.42\times$ prefill speedup, offering a practical solution for efficient MLLM inference. Code and model checkpoints will be publicly released.

1 INTRODUCTION

Visual token pruning reduces the inference cost of multimodal large language models (MLLMs), whose high-resolution inputs can contain hundreds or thousands of visual tokens. Recent methods remove a substantial fraction of these tokens while preserving performance on general multimodal question answering benchmarks (Chen et al., 2024a; Yang et al., 2025; Zhang et al., 2025b). Preserving visual grounding under aggressive pruning remains more challenging (Chien et al., 2025; Huang et al., 2026b). Given an image and a referring expression, grounding requires the model to identify the described object and predict its bounding box. The retained visual tokens must therefore provide sufficient evidence for both target recognition and precise localization.

Existing methods provide several criteria for selecting visual evidence. General pruning approaches use attention, visual salience, or feature diversity to estimate token importance or redundancy (Chen et al., 2024a; Yang et al., 2025; Alvar et al., 2025). GlimpsePrune learns instruction-conditioned importance through language and bounding-box supervision (Zeng et al., 2026). To preserve spatial information, Nüwa combines spatially structured token aggregation with text-guided pruning (Huang et al., 2026b), while IVC-Prune retains implicit coordinate tokens alongside semantically relevant foreground tokens (Sun et al., 2026). These advances highlight the value of target-relevant content and spatial structure. Yet identifying the target region does not fully determine which visual tokens best support grounding under a tight budget. We investigate this through controlled token-addition experiments (Section 3, Figure 1). For query-agnostic selectors, adding tokens that cover the ground-truth box (target additions) yields substantially larger gains than expanding the native se-

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.

lection to the same total budget, highlighting the importance of target-relevant evidence. However, even Oracle Target, which fills its budget from the ground-truth box outward, benefits more from tokens at uniformly spaced grid anchors (context additions) than from extending its own selection. Together, these results show that target relevance is a useful selection criterion, but target guidance alone is insufficient for accurate grounding under a fixed budget.

These observations motivate two complementary learning objectives. The first learns query-conditioned target relevance from bounding-box supervision, establishing a prior for *where to look*. The second learns *what to keep* by optimizing fixed-budget token subsets using the frozen MLLM's grounding outcomes. Region labels provide direct supervision for target relevance, while downstream feedback evaluates how well the selected tokens jointly support coordinate prediction. This motivates learning a target prior first, then refining the selection through grounding feedback.

We implement this approach through a two-phase framework that trains a lightweight, query-conditioned token scorer while keeping the MLLM backbone frozen. In Phase One, **Localization Prior Learning (LPL)** trains the scorer with bounding-box supervision, teaching it which visual tokens correspond to the query-relevant region (*where to look*). In Phase Two, **Grounding-Guided Selection Refinement (GSR)** optimizes the discrete token selection under a fixed token budget. The scorer samples candidate subsets by perturbing its logits with Gumbel noise; the frozen MLLM generates a bounding box for each candidate, and its IoU with the ground-truth box serves as a reward signal. A group-relative policy gradient then updates the scorer to favor selections that yield better grounding predictions, teaching it *what to keep* to support accurate grounding under a fixed token budget. At inference, sampling is removed and the scorer applies a single deterministic top-K selection at an intermediate decoder layer.

Our main contributions are summarized as follows:

- **Controlled analysis of token selection for grounding.** Budget-matched experiments show that target additions outperform native budget expansion for query-agnostic selectors, while context additions further improve target-guided selectors. These findings motivate learning both a target-relevance prior and a selection policy optimized for grounding accuracy.
- **Two-phase learning for grounding-aware token selection.** LPL uses bounding-box supervision to initialize the scorer with a localization prior; GSR then refines the token selection through grounding IoU rewards, training only the lightweight scorer while the MLLM backbone remains frozen throughout.
- **Strong grounding performance with preserved general capability.** Under an aggressive 11.1% token retention ratio, our method retains up to 117.6% of unpruned grounding accuracy while preserving over 98% of general VQA performance, together with up to $1.54\times$ end-to-end speedup and $1.42\times$ prefill speedup.

2 Related Work

2.1 Visual Token Pruning for MLLMs

Visual token pruning reduces the inference and memory cost of multimodal large language models by discarding redundant visual tokens. Heuristic methods exploit text-to-vision attention, vision-encoder importance proxies, feature diversity, or text–vision feature coverage to select or merge tokens (Chen et al., 2024a; Yang et al., 2025; Zhang et al., 2025b; 2026; Alvar et al., 2025; Dong et al., 2026). While effective on standard VQA, pruning based on such heuristic criteria may discard referred objects or spatial context required for accurate visual localization. Learned selectors differ in the signal that defines a useful token. LVPruning trains language-guided cross-attention modules for mid-decoder selection with the language-modeling loss (Sun et al., 2025). GlimpsePrune supervises token retention with bounding-box overlap, and GlimpsePrune+ further fine-tunes the language decoder with reinforcement learning (Zeng et al., 2026). Recent methods train selectors with reinforcement learning, rewarding task performance on general multimodal benchmarks (Cao et al., 2026; Huang et al., 2026a). These signals define token usefulness through region supervision, language-modeling loss, or general-benchmark rewards. For grounding, region supervision identifies the target (*where to look*), and the retained subset additionally needs context that supports

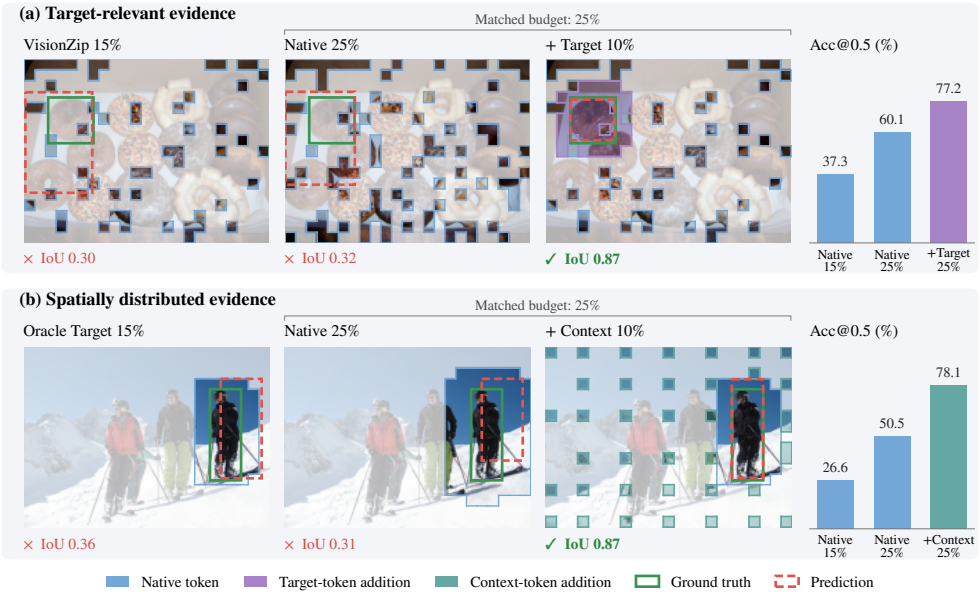

Figure 1: **Token-addition experiments motivate two complementary learning objectives** (Qwen2.5-VL, RefCOCO+ val). Starting from a 15% selection, we compare native budget expansion with target or context additions at the same 25% total budget. **(a)** For VisionZip, target additions outperform native expansion. **(b)** For Oracle Target, context additions outperform native expansion. These observations motivate learning target relevance (*where to look*) and refining fixed-budget selections through grounding feedback (*what to keep*).

coordinate prediction (*what to keep*; Section 3). We therefore refine the selection with the frozen MLLM's grounding outcome on the same top-K subset used at inference.

2.2 Preserving Spatial Grounding under Token Pruning

Visual grounding exposes severe spatial degradation under aggressive token pruning that often remains concealed on aggregate VQA benchmarks. Grounding-Aware Token Pruning attributes this degradation to the altered values and order of position IDs after pruning and restores accuracy by preserving the original position IDs of the selected tokens (Chien et al., 2025). Nüwa partitions the visual token grid into non-overlapping local regions and selects aggregation centers within each region using attention and key-norm salience. It aggregates neighboring features with spatial constraints before applying text-guided selection in the language decoder (Huang et al., 2026b). IVC-Prune exploits the mathematical properties of multimodal rotary position embeddings (RoPE) to identify and retain implicit visual coordinate tokens (Sun et al., 2026).

These methods prescribe spatial structure through grid partitions, constrained aggregation, or positional rules. Our pilot study shows that the missing evidence depends on the selector: query-agnostic selections benefit from added target tokens, whereas target-guided selections benefit from added context tokens (Section 3). We therefore learn both components with the MLLM backbone frozen: region supervision provides the target prior (*where to look*), and grounding feedback determines the supporting context within the budget (*what to keep*).

3 Diagnosing Grounding Degradation: Target Evidence vs. Spatial Context

To examine how target and context additions affect grounding under a fixed total budget, we run budget-matched token-addition experiments on RefCOCO+ validation. Starting from each selector's native 15% subset, we add 10% more tokens, either ranked by the ground-truth box (target additions,

+Target) or placed at uniformly spaced grid anchors (context additions, +Context; Appendix C.4), and compare against the same selector run natively at the matched 25% budget. Each selector receives the addition that its criterion omits: query-agnostic selectors receive target additions, and target-guided selectors receive context additions. We test query-agnostic VisionZip (Yang et al., 2025) and DivPrune (Alvar et al., 2025), together with target-guided GlimpsePrune (Zeng et al., 2026) and *Oracle Target*, which ranks tokens by distance to the ground-truth box (Appendix C.4).

Table 1: **Adding target evidence.** Adding target tokens to the native 15% selection gives a total 25% budget; ↑: gain over Native 25%.

Backbone	Method	Acc@0.5 (%)		
		Native 15%	Native 25%	+Target 10%
Qwen2.5-VL	VisionZip	37.3	60.1	**77.2** (↑17.1)
	DivPrune	38.6	63.2	**76.5** (↑13.3)
LLaVA-1.5	VisionZip	13.0	25.5	**36.7** (↑11.2)
	DivPrune	24.4	39.0	**45.2** (↑6.2)

Table 2: **Adding spatial context.** Adding context tokens to target-guided selections gives a total 25% budget; ↑: gain over Native 25%.

Backbone	Method	Acc@0.5 (%)		
		Native 15%	Native 25%	+Context 10%
Qwen2.5-VL	Oracle Target	26.6	50.5	**78.1** (↑27.6)
	GlimpsePrune	70.9	73.4	**81.4** (↑8.0)
LLaVA-1.5	Oracle Target	23.1	36.2	**45.3** (↑9.1)
	GlimpsePrune	49.9	52.0	**54.0** (↑2.0)

Target evidence is critical for query-agnostic pruning. At the matched 25% budget, +Target improves VisionZip and DivPrune by 17.1 and 13.3 points on Qwen2.5-VL, and by 11.2 and 6.2 points on LLaVA-1.5 (Table 1). Figure 1a illustrates this comparison for VisionZip: target additions achieve 77.2%, compared with 60.1% for native budget expansion. These results highlight the value of explicitly prioritizing target-relevant evidence.

Target guidance alone is insufficient. For target-guided selectors, +Context improves Oracle Target by 27.6 points on Qwen2.5-VL and 9.1 points on LLaVA-1.5, and improves GlimpsePrune by 8.0 and 2.0 points, respectively (Table 2). At the same total budget, context additions are thus more effective than extending a target-guided selection, even when that selection follows the ground-truth box. Target guidance alone is therefore insufficient for accurate grounding; tokens outside the target likely provide spatial references for coordinate prediction.

Together, these experiments reveal the two failure modes in Figure 1 and motivate two complementary learning objectives: learning query-conditioned target relevance (*where to look*) and optimizing the retained subset for downstream grounding (*what to keep*). Region labels specify the target but leave open which additional tokens support coordinate prediction, so we learn the second objective from the MLLM's own grounding outcome through IoU-guided GSR, following region-supervised LPL (Section 4).

4 METHOD

Figure 2 shows our framework. Given an image I and query q, a lightweight scorer assigns logits $\mathbf{s} = [s_1, \ldots, s_N]$ to N visual tokens after decoder layer ℓ. The top-K tokens retain their original order and positional indices for layers $\ell + 1, \ldots, L$. Motivated by Section 3, **Localization Prior Learning (LPL)** learns target relevance from region supervision, and **Grounding-Guided Selection Refinement (GSR)** refines fixed-budget subsets using grounding feedback. Only the scorer is trained; the backbone remains frozen.

4.1 QUERY-CONDITIONED TOKEN SCORER

Let $\mathbf{H}^\ell = [\mathbf{h}_1^\ell, \ldots, \mathbf{h}_N^\ell] \in \mathbb{R}^{N \times D}$ denote visual hidden states after decoder layer ℓ, and let $\mathbf{q}^\ell$ be the hidden state of the final query token. As shown in Figure 2a, the scorer first fuses $\mathbf{h}_i^\ell$ with spatial descriptor $\mathbf{c}_i$ and attention cue $\bar{\alpha}_i$:

$$\mathbf{z}_i = W_v \, \mathrm{LN}(\mathbf{h}_i^\ell) + \phi_c(\mathbf{c}_i) + \phi_a(\bar{\alpha}_i). \tag{1}$$

Here W_v is a learned projection matrix, ϕ_c is a two-layer MLP that encodes the token's spatial position and scale on the visual grid, and ϕ_a is a two-layer MLP that embeds the scalar attention cue $\bar{\alpha}_i$, the head-averaged log-attention weight from a designated text position to visual token i in decoder layer ℓ. The spatial parameterization and backbone-specific text-position choices are provided in Appendix B.

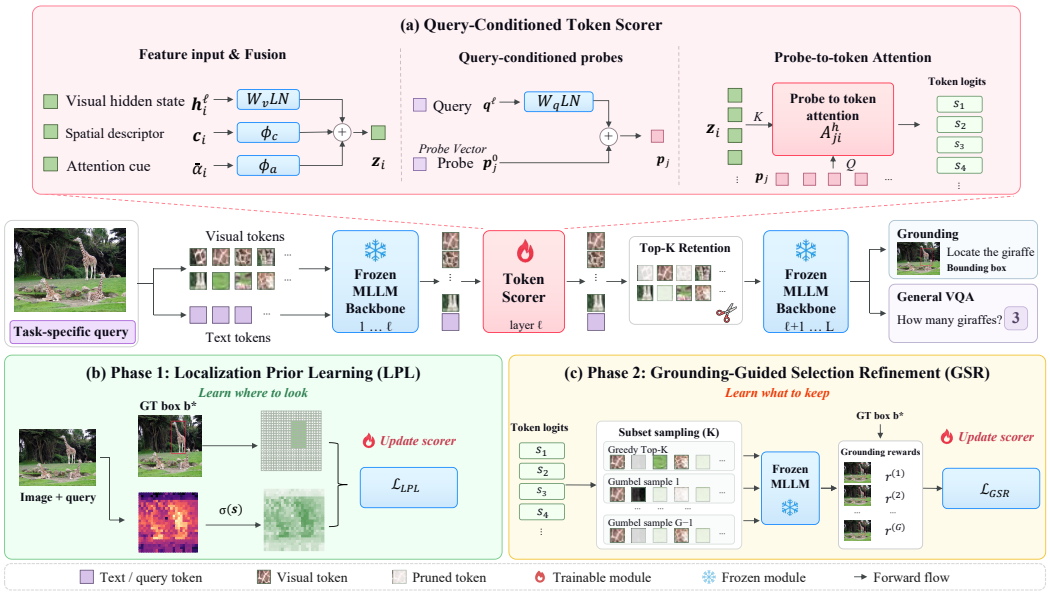

Figure 2: **Framework overview.** The central pipeline inserts the scorer after layer ℓ and passes the top-K tokens to layers $\ell+1, \ldots, L$. **(a)** Query-conditioned probes score tokens using visual, spatial, and attention cues. **(b)** LPL learns *where to look* from box-derived masks. **(c)** GSR learns *what to keep* from grounding IoU; all candidates enter the group mean, and only stochastic candidates contribute policy gradients. Both phases train the same scorer with the backbone frozen.

It then conditions P learnable probes $\{\mathbf{p}_j^0\}_{j=1}^P$ on the query representation:

$$\mathbf{p}_j = \mathbf{p}_j^0 + W_q \,\mathrm{LN}(\mathbf{q}^\ell), \qquad j = 1, \ldots, P, \tag{2}$$

where W_q is a learned projection matrix. The conditioned probes attend to the fused visual representations through multi-head attention, with probes as queries and visual representations as keys. The resulting softmax weights A_{ji}^h measure probe j's attention to visual token i in head h (Appendix B). We aggregate these weights across probes and heads to obtain token logits:

$$s_i = \log\left(\frac{N}{PH}\sum_{h=1}^{H}\sum_{j=1}^{P} A_{ji}^h + \epsilon\right). \tag{3}$$

At inference, we retain the K highest-scoring visual tokens while preserving their original order and positional indices. By combining visual content, spatial position, and query-conditioned attention into a single score, the scorer provides a common interface for both LPL supervision and GSR policy optimization.

4.2 Localization Prior Learning

LPL learns target relevance from region supervision (Figure 2b). Given ground-truth box $\mathbf{b}^*$, token i is labeled positive when its visual patch overlaps the box:

$$y_i = \mathbb{I}(\mathrm{patch}_i \cap \mathbf{b}^* \neq \emptyset). \tag{4}$$

With $p_i = \sigma(s_i)$, Dice and BCE losses align the relevance map $\mathbf{p}$ with the target mask $\mathbf{y}$:

$$\mathcal{L}_{\mathrm{LPL}} = \mathcal{L}_{\mathrm{Dice}}(\mathbf{p}, \mathbf{y}) + \lambda_{\mathrm{BCE}}\mathcal{L}_{\mathrm{BCE}}(\mathbf{p}, \mathbf{y}), \tag{5}$$

following region-supervised token selection (Zeng et al., 2026). The standard Dice and BCE definitions are provided in Appendix B.

4.3 GROUNDING-GUIDED SELECTION REFINEMENT

LPL provides target relevance. GSR refines the fixed-budget subset using the frozen MLLM's grounding outcome on each candidate subset (Figure 2c).

For each input, we generate G candidate subsets: one greedy top-K candidate ($g = 1$) and $G - 1$ stochastic candidates ($g = 2, \ldots, G$) sampled by perturbing the scorer logits with Gumbel noise at temperature τ:

$$\mathbf{m}^{(g)} = \mathrm{TopKMask}(\mathbf{s} + \tau\boldsymbol{\epsilon}^{(g)}, K), \qquad \epsilon_i^{(g)} \sim \mathrm{Gumbel}(0, 1), \quad g = 2, \ldots, G. \tag{6}$$

Gumbel perturbations induce a Plackett–Luce distribution over ordered token selections (Kool et al., 2019), allowing score-function optimization of the discrete choices. Applying candidate $\mathbf{m}^{(g)}$ after layer ℓ produces a predicted box $\widehat{\mathbf{b}}^{(g)}$, which is scored by

$$r^{(g)} = \mathrm{IoU}(\widehat{\mathbf{b}}^{(g)}, \mathbf{b}^*), \tag{7}$$

with invalid or unparseable predictions receiving zero reward. We compute the group-relative advantage using all G rewards, $A^{(g)} = r^{(g)} - \frac{1}{G}\sum_{j=1}^{G} r^{(j)}$, and optimize the following surrogate:

$$\mathcal{L}_{\mathrm{GSR}} = -\frac{1}{G-1}\sum_{g=2}^{G} A^{(g)} \frac{1}{K} \log p_\theta(\pi_{1:K}^{(g)} \mid I, q). \tag{8}$$

Here $\pi_{1:K}^{(g)}$ denotes the sampled selection order, and $1/K$ averages its log-likelihood over the selected tokens. Advantages are treated as constants during differentiation. The greedy candidate enters the group mean, and the stochastic candidates carry the policy-gradient terms through their selection log-probabilities. The explicit Plackett–Luce likelihood and implementation details are deferred to Appendix B.

Throughout LPL and GSR, the vision encoder, projector, and language model remain frozen. At inference, sampling is removed and the scorer applies a single deterministic top-K selection. Additional scorer and optimization details are provided in Appendix B, with hyperparameters summarized in Appendix C.2.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Backbones and pruning configurations. We evaluate Qwen2.5-VL-7B (Bai et al., 2025) and LLaVA-1.5-7B (Liu et al., 2024a) in the main experiments. For each backbone, a lightweight query-conditioned scorer prunes visual tokens at an intermediate decoder layer. Appendix Table 11 summarizes the selected pruning layers and training configurations; all pruned methods are evaluated at an 11.1% visual-token retention ratio. Appendices A.1 and A.2 extend the evaluation to 22.2% and 33.3% retention and to LLaVA-1.5-13B.

Baselines. We compare with DivPrune (Alvar et al., 2025), VisionZip (Yang et al., 2025), GlimpsePrune (Zeng et al., 2026), Nüwa (Huang et al., 2026b), and IVC-Prune (Sun et al., 2026). DivPrune and VisionZip are query-agnostic, GlimpsePrune is query-conditioned, while Nüwa and IVC-Prune are designed to better preserve grounding, with IVC-Prune also conditioning selection on the query. We match the final visual-token retention ratio to 11.1%; when an official operating point is unavailable, we proportionally scale the method's token allocation. Retained tokens preserve their original positional indices. Additional evaluation settings are provided in Appendix C.3.

Training data. LPL uses 8,000 Visual-CoT examples (Shao et al., 2024), balanced across GQA, InfoGraphicsVQA, DocVQA, and TextVQA. For GSR, we use an 8,000-example pool from RefCOCO, RefCOCO+ (Yu et al., 2016), and RefCOCOg (Mao et al., 2016), with no COCO image-ID overlap with the eight grounding evaluation splits. Both backbones train on the same 4,000 examples from this pool.

Table 3: **Visual grounding at 11.1% token retention.** Avg. Ret. is the unweighted mean of per-split score ratios to the unpruned model, computed from unrounded results. Bold and underline mark the best and second-best pruned results per backbone and split; † marks grounding-oriented methods.

Model	Method	RefCOCO			RefCOCOg		RefCOCO+			**Avg. Ret.**
		Val	TestA	TestB	Val	Test	Val	TestA	TestB	
	Token Retain Ratio = 11.1%									
Qwen2.5-VL	Unpruned	89.3	92.2	84.2	86.4	86.0	82.1	87.7	74.4	100.0%
	VisionZip	23.7	26.2	22.2	22.9	21.8	21.3	22.6	18.2	26.2%
	DivPrune	30.0	36.0	24.2	27.3	26.4	26.5	33.3	19.9	32.8%
	GlimpsePrune	74.8	75.0	71.8	71.3	71.3	67.4	70.5	<u>63.0</u>	82.8%
	Nüwa†	69.4	73.0	66.1	67.1	67.4	59.2	63.9	55.2	76.4%
	IVC-Prune†	<u>80.3</u>	<u>82.7</u>	<u>75.0</u>	<u>73.7</u>	<u>72.2</u>	<u>69.6</u>	<u>75.3</u>	60.5	<u>86.4%</u>
	Ours†	**86.5**	**90.0**	**81.3**	**83.5**	**83.4**	**78.6**	**85.2**	**70.3**	**96.6%**
LLaVA-1.5	Unpruned	56.2	64.6	47.4	48.8	48.4	49.9	59.5	39.1	100.0%
	VisionZip	10.2	10.4	9.2	8.4	8.6	8.1	8.2	8.7	17.4%
	DivPrune	19.0	23.3	18.0	16.1	14.9	15.8	18.9	14.4	33.9%
	GlimpsePrune	55.2	59.9	48.3	47.2	46.7	47.7	54.5	39.6	96.5%
	Nüwa†	<u>60.3</u>	<u>68.2</u>	<u>51.1</u>	<u>53.2</u>	<u>52.2</u>	<u>51.8</u>	<u>62.6</u>	<u>41.0</u>	<u>106.4%</u>
	IVC-Prune†	43.8	48.8	36.4	45.2	44.6	37.5	44.3	28.6	79.5%
	Ours†	**66.4**	**74.9**	**57.0**	**59.5**	**57.6**	**57.1**	**68.9**	**45.1**	**117.6%**

Evaluation benchmarks. For visual grounding, we report $\mathrm{Acc@0.5}$ on all eight evaluation splits of RefCOCO, RefCOCO+ (Yu et al., 2016), and RefCOCOg (Mao et al., 2016). For general visual understanding, we evaluate on VQAv2-Lite (Goyal et al., 2017), GQA (Hudson & Manning, 2019), OCRBench (Liu et al., 2024c), ChartQA (Masry et al., 2022), TextVQA (Singh et al., 2019), MMBench (Liu et al., 2024b), MMStar (Chen et al., 2024b), MME-P (Fu et al., 2025), CV-Bench 2D (Tong et al., 2024), and AI2D (Kembhavi et al., 2016). All evaluations use the `lmms-eval` framework (Zhang et al., 2025a).

5.2 Main Results

5.2.1 Visual Grounding Performance

Table 3 reports results on the eight grounding splits. At 11.1% retention, our method reaches an average $\mathrm{Acc@0.5}$ of 82.35 on Qwen2.5-VL and 60.82 on LLaVA-1.5, corresponding to 96.6% and 117.6% of their unpruned performance, respectively. LLaVA-1.5 has comparatively weak unpruned grounding, and both our method and the training-free Nüwa (106.4%) surpass it, suggesting that reducing visual redundancy can benefit grounding. The grounding advantage holds beyond this operating point: at 22.2% and 33.3% retention, our method achieves the highest $\mathrm{Acc@0.5}$ among Nüwa, IVC-Prune, and ours on every RefCOCO+ split for both backbones (Appendix A.1), and on LLaVA-1.5-13B it exceeds the unpruned model on all three RefCOCO+ splits (Appendix A.2).

5.2.2 General Multimodal Understanding

To evaluate how grounding-oriented token selection transfers to broader visual capabilities, we benchmark each method on the 10-task full suite. As reported in Table 4, our method retains 98.6% and 98.2% of unpruned performance on Qwen2.5-VL and LLaVA-1.5, respectively, giving the highest reported average retention among the pruned methods for both backbones.

5.3 Efficiency Analysis

Table 5 compares inference efficiency against IVC-Prune and GlimpsePrune on an RTX 4090 (batch 32). Our method achieves $\mathbf{1.54}\times$ and $\mathbf{1.44}\times$ E2E speedups on LLaVA-1.5 and Qwen2.5-VL while attaining the highest grounding accuracy among all pruned arms.

Table 4: **General visual understanding.** Accuracy or scores accross ten benchmarks. Avg. Ret. is the unweighted mean of per-task score ratios to the unpruned model. Bold and underline mark the best and second-best pruned results per backbone and task.

Model	Method	VQAv2lite	GQA	OCRB	Chart	VQA^T	MMB	MMstar	MME-P	CV2D	AI2D	**Avg. Ret.**
Qwen2.5-VL	Unpruned	79.3	60.9	85.5	84.0	82.8	83.5	62.4	1685.7	75.2	82.5	100.0
	VisionZip	67.0	52.8	40.7	47.0	63.6	76.7	49.7	1544.1	67.7	72.8	79.3
	DivPrune	66.3	55.6	41.1	43.0	57.4	74.0	46.1	1540.4	63.9	70.8	76.8
	GlimpsePrune	78.2	60.2	77.4	**80.1**	82.1	**83.4**	**62.3**	1680.6	75.0	82.5	98.2
	Nüwa	76.1	58.1	61.7	69.2	76.0	82.0	56.5	1693.5	72.4	79.3	91.9
	IVC-Prune	76.9	59.5	43.3	60.0	71.7	83.2	60.3	1692.5	**75.5**	82.0	90.0
	Ours	**79.6**	**60.4**	**79.2**	80.0	**82.7**	83.0	61.2	**1695.2**	**75.5**	**82.8**	**98.6**
LLaVA-1.5	Unpruned	71.0	61.7	31.1	17.8	45.8	64.2	33.7	1476.6	60.2	54.4	100.0
	VisionZip	64.6	55.3	27.4	16.2	42.0	59.5	30.8	1256.9	53.5	52.0	90.5
	DivPrune	68.5	57.9	26.8	15.5	39.1	59.0	30.8	1229.1	54.5	50.4	89.9
	GlimpsePrune	**69.9**	**61.5**	21.2	16.9	43.8	64.2	33.6	1476.1	60.2	**54.5**	95.7
	Nüwa	69.4	60.5	24.4	**18.0**	45.0	63.2	**34.1**	1436.1	51.0	53.8	95.4
	IVC-Prune	66.6	60.9	13.4	15.1	30.8	**64.3**	33.4	**1481.1**	60.4	**54.5**	88.8
	Ours	69.4	59.6	**28.3**	17.3	**45.5**	64.0	33.5	1477.5	**60.9**	**54.5**	**98.2**

Table 5: **Inference efficiency** on an RTX 4090 under batch 32. Theory = decoder-prefill TFLOPs/req (scorer overhead included); KV = KV-cache footprint per request (MiB, analytic); speedups are relative to Dense. VG is the average performance on eight visual grounding benchmarks. Bold/underline: best/second-best pruned per backbone and column.

	Qwen2.5-VL					LLaVA-1.5				
Method	KV↓	Theory↓	Pref. spd↑	E2E spd↑	VG↑	KV↓	Theory↓	Pref. spd↑	E2E spd↑	VG↑
Unpruned	21.1	5.10	1.00×	1.00×	85.27	340.3	9.06	1.00×	1.00×	51.72
IVC-Prune	**13.8**	**3.34**	**1.29**×	**1.46**×	73.66	212.3	5.63	1.32×	1.45×	41.13
GlimpsePrune	15.7	3.78	1.14×	1.37×	70.64	260.3	6.92	1.21×	1.35×	49.89
Ours	14.5	3.49	1.23×	1.44×	**82.35**	**188.3**	**4.99**	**1.42**×	**1.54**×	**60.82**

5.4 Ablation Studies

We ablate training phases, pruning depth, and scorer design at an 11.1% visual-token budget ($K = 64$ for LLaVA-1.5). An analysis of what GSR learns is provided in Appendix A.3.

Training phases. Table 6 shows the complementary roles of LPL and GSR. LPL-only preserves general VQA performance but falls short on grounding; GSR-only achieves strong grounding but incurs larger losses on general tasks, notably OCRBench. LPL→GSR combines both: LPL initialization keeps retention high while subsequent GSR training recovers grounding accuracy, consistent with the pilot finding that target focus alone is insufficient.

Pruning depth. To select the best layers for different MLLM backbones, we sweep different pruning layers and evaluate on RefCOCO+ Val using the same settings as the main results. Both backbones show clear performance saturation (Table 7), and we select layer 16 for Qwen2.5-VL and layer 12 for LLaVA-1.5 as the optimal efficiency-performance trade-off.

Table 7: **Pruning depth.** Performance and efficiency on RefCOCO+ validation datasets. Layer with † is finally selected.

Qwen2.5-VL				*LLaVA-1.5*			
ℓ	FLOPs↓	Acc.↑	IoU↑	ℓ	FLOPs↓	Acc.↑	IoU↑
12	2.90	67.21	59.94	8	4.13	38.72	38.37
14	3.19	74.69	65.68	12†	4.99	57.12	48.75
16†	3.49	78.57	70.53	16	5.84	**58.33**	**49.83**
18	3.78	**79.18**	**71.50**	20	6.70	58.22	49.44

Scorer design. Panel (b) of Table 6 isolates the coordinate cue, text-to-vision attention cue, and probe-based scoring mechanism. The MLP variant retains both cues and query conditioning, concatenating each augmented visual-token representation with the projected query to predict its score.

Table 6: **Ablation studies.** **(a)** Training-phase ablation: RefCOCO+ Acc@0.5 (%) and five representative General VQA tasks. **(b)** Scorer design ablation: RefCOCO+ Acc@0.5 (%) per variant. Bold and underline mark the best and second-best per backbone and metric.

(a) Training phases

		Grounding (RefCOCO+)				General VQA					
Model	Phase	Val	TestA	TestB	Avg.	OCR	Chart	MMStar	MMB	VQAv2	Avg.
Qwen2.5-VL	LPL only	65.02	69.33	55.49	63.28	**79.7**	**80.1**	61.6	82.9	79.3	**76.7**
	GSR only	78.56	84.54	**71.20**	**78.10**	50.9	79.0	**61.7**	**83.0**	75.9	70.1
	LPL→GSR	**78.57**	**85.24**	70.34	78.05	79.2	80.0	61.2	**83.0**	**79.6**	76.6
LLaVA-1.5	LPL only	46.91	58.68	33.46	46.35	27.9	**17.3**	**33.7**	63.7	68.9	42.3
	GSR only	55.36	66.85	43.49	55.23	10.7	17.0	**33.7**	63.8	66.4	38.3
	LPL→GSR	**57.12**	**68.91**	**45.10**	**57.05**	**28.3**	**17.3**	33.5	**64.0**	**69.4**	**42.5**

(b) Scorer design (RefCOCO+ Acc@0.5 %)

	Scorer components			LLaVA-1.5				Qwen2.5-VL			
	$\mathbf{c}_i$	$\bar{\alpha}_i$	$\mathbf{p}_j$	Val	TestA	TestB	Avg.	Val	TestA	TestB	Avg.
	×	✓	✓	**57.16**	67.85	44.45	56.48	78.25	84.56	69.30	77.37
	✓	×	✓	56.20	66.92	43.77	55.63	71.68	79.10	63.00	71.26
	✓	✓	×	55.01	66.31	43.22	54.85	74.08	78.64	65.76	72.83
Ours	✓	✓	✓	57.12	**68.91**	**45.10**	**57.05**	**78.57**	**85.24**	**70.34**	**78.05**

$\mathbf{c}_i$: coordinate cue; $\bar{\alpha}_i$: text-to-vision attention cue; $\mathbf{p}_j$: query-conditioned probes. ✓/×: enabled/disabled; we replace probe attention with a query-conditioned MLP when probe is disabled.

Replacing probe attention with this MLP reduces average grounding accuracy by 5.22 points on Qwen2.5-VL and 2.20 points on LLaVA-1.5. Removing the attention cue causes a larger average drop than removing the coordinate cue on both backbones (6.79 vs. 0.68 points and 1.42 vs. 0.57 points, respectively).

6 Conclusion

We identified two complementary challenges in visual token pruning for grounding: retaining target-relevant content and preserving the spatial context needed for accurate localization. Our two-phase framework addresses these challenges by learning a localization prior from region supervision and refining token selection through grounding feedback, while keeping the MLLM backbone frozen. At 11.1% token retention, our method outperforms the evaluated pruning baselines on grounding across both backbones, preserves over 98% of unpruned general VQA performance, and achieves up to $1.54\times$ end-to-end speedup. These results highlight the value of combining spatial supervision with downstream task feedback for visual token selection.

7 Limitations and Discussion

Our method applies a fixed retention ratio and pruning layer to each backbone across inputs and tasks; adapting both to input complexity and task demands is a promising direction. In addition, GSR derives its reward from referring expression grounding. Extending this reward to feedback from other visual tasks, such as answer correctness in VQA, may further improve the generality of learned token selection.

AI use statement

We used generative AI tools as assistants in several stages of this work, including literature discovery, translation and language editing, and code implementation and debugging. All AI-assisted content and code were reviewed and, where applicable, tested by the authors.

For literature discovery, we used AI-assisted tools to help search for and identify potentially relevant papers and related work. The authors independently examined the relevant source materials and determined which works to cite and how they relate to this study. For writing assistance, we used AI tools for translation, grammatical correction, and language polishing to improve the clarity

and readability of the manuscript. The technical content, scientific claims, and conclusions were reviewed and determined by the authors.

For coding assistance, we used AI coding tools to assist with debugging and the integration or adaptation of existing methods into our experimental framework, including implementations of baseline token-pruning methods (e.g. IVC-Prune, VisionZip). AI tools were also used to help diagnose implementation issues during early-stage training. All AI-assisted code relevant to the reported experiments was reviewed and tested by the authors, and the reported experimental results were obtained on mentioned datasets and evaluation benchmarks.

We did not use generative AI to generate synthetic experimental data or fabricate experimental results. We also did not use generative AI to formulate or prove mathematical claims in this work. The core research questions, the proposed two-stage learning framework, the experimental methodology, the model architecture, and the interpretation and presentation of the final scientific conclusions were determined by the authors. We take responsibility for the final content of this work, including all text, claims, experimental results, and artifacts produced with the assistance of generative AI tools.

Ethics statement

This work studies visual token pruning for multimodal large language models using publicly available datasets (RefCOCO, RefCOCO+, RefCOCOg) and pre-trained model weights. No human subjects were involved. The datasets used are standard vision-language benchmarks released for research purposes. We do not foresee direct harms from this work; efficiency improvements for inference-time MLLM computation are broadly beneficial. We follow the terms of use for all datasets and model weights used in our experiments.

Reproducibility statement

To reproduce our results, we provide a detailed description of our methods in Section 4 and Appendix B. The detailed evaluation settings are described in Appendix C.3.

The source code, model checkpoints, and raw experiment logs are submitted as supplementary materials and will be released publicly after the paper is accepted.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, June 2025.

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, 2025. URL https://arxiv.org/abs/2502.13923.

Sihan Cao, Jianwei Zhang, Pengcheng Zheng, Jiaxin Yan, Caiyan Qin, Yalan Ye, Wei Dong, Peng Wang, Yang Yang, and Chaoning Zhang. Language-guided token compression with reinforcement learning in large vision-language models, 2026. URL https://arxiv.org/abs/2603.13394.

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 Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 19–35, Cham, 2024a. Springer Nature Switzerland. ISBN 978-3-031-73004-7.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=evP9mxNNxJ.

Tzu-Chun Chien, Chieh-Kai Lin, Shiang-Feng Tsai, Ruei-Chi Lai, Hung-Jen Chen, and Min Sun. Grounding-aware token pruning: Recovering from drastic performance drops in visual grounding caused by pruning, 2025. URL https://arxiv.org/abs/2506.21873.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. Mm-tok: Multimodal coverage maximization for efficient inference of vlms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 48075–48099, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/4f8745c0ca5c951a92844ad7668cfca5-Paper-Conference.pdf.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025. URL https://arxiv.org/abs/2306.13394.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Mingzhe Huang, Weijun Wang, Xin Ding, Liang Mi, Hao Wen, Yuanchun Li, Lichen Pang, Shansong Yang, Yunxin Liu, and Ting Cao. Grip-vlm: Group-relative importance pruning for efficient vision-language models, 2026a. URL https://arxiv.org/abs/2605.13375.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 100825–100856, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a46156bd3579c3b268108ea6aca71d13-Paper-Conference.pdf.

Drew A. Hudson and Christopher D. Manning. GQA: a new dataset for compositional question answering over real-world images. *CoRR*, abs/1902.09506, 2019. URL http://arxiv.org/abs/1902.09506.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images, 2016. URL https://arxiv.org/abs/1603.07396.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The Gumbel-top-k trick for sampling sequences without replacement. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3499–3508. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296, 2024a. doi: 10.1109/CVPR52733.2024.02484.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VI*, pp. 216–233, Berlin, Heidelberg, 2024b. Springer-Verlag. ISBN 978-3-031-72657-6. doi: 10.1007/978-3-031-72658-3_13. URL https://doi.org/10.1007/978-3-031-72658-3_13.

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), 2024c. ISSN 1869-1919. doi: 10.1007/s11432-024-4235-6. URL http://dx.doi.org/10.1007/s11432-024-4235-6.

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 (CVPR)*, June 2016.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 8612–8642. Curran Associates, Inc., 2024. doi: 10.52202/079017-0275. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0ff38d72a2e0aa6dbe42de83a17b2223-Paper-Datasets_and_Benchmarks_Track.pdf.

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 (CVPR)*, June 2019.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Batista-Navarro. LVPruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4299–4308, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.242. URL https://aclanthology.org/2025.findings-naacl.242/.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 111471–111492, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/b51b3565a6775c557093cb99457fc7fc-Paper-Conference.pdf.

Shengbang Tong, Ellis L Brown II, Penghao Wu, Sanghyun Woo, ADITHYA JAIRAM IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, Xichen Pan, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Vi8AepAXGy.

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 (CVPR)*, pp. 19792–19802, June 2025.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 69–85, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46475-6.

Quan-Sheng Zeng, Yunheng Li, Qilong Wang, Peng-Tao Jiang, Zuxuan Wu, Ming-Ming Cheng, and Qibin Hou. A glimpse to compress: Dynamic visual token pruning for large vision-language models. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2026. doi: 10.1109/TCSVT.2026.3702147.

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*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=KZYhyilFnt.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-eval: Reality check on the evaluation of large multimodal models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, Albuquerque, New Mexico, April 2025a. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.51. URL https://aclanthology.org/2025.findings-naacl.51/.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 13–19 Jul 2025b. URL https://proceedings.mlr.press/v267/zhang25s.html.

A Additional Experiments and Analyses

We first present additional experiments and analyses, followed by method details in Appendix B and reproducibility settings in Appendix C.

A.1 Additional Token Budgets

We test whether the grounding and general VQA results extend to larger visual-token budgets. We compare our method with Nüwa and IVC-Prune at retention ratios of 22.22% and 33.33% ($2/9$ and $1/3$ of the visual tokens) on both 7B backbones. Table 8 reports full-split RefCOCO+ Acc@0.5 and five representative VQA tasks; VQA retention averages the five task-score ratios to the same unpruned backbone.

Table 8: **Grounding and general VQA at additional token budgets.** Grounding reports full RefCOCO+ Acc@0.5 (%) and its three-split mean. General VQA reports five task scores (%); Ret. is the mean of their score ratios to the unpruned backbone (%). Ours rows are shaded.

		Grounding (RefCOCO+)				General VQA					
Model	Method	Val	TestA	TestB	Avg.	OCR	Chart	MMStar	MMB	VQAv2	Ret.
Retained visual tokens: 22.22%											
Qwen2.5-VL	Nüwa	73.49	78.19	66.80	72.83	73.6	75.5	59.2	82.7	78.1	93.7
	IVC-Prune	78.18	83.79	70.12	77.36	61.3	70.2	61.2	82.7	79.4	90.5
	Ours	81.33	87.29	74.02	80.88	81.4	80.5	61.8	83.4	79.5	98.0
LLaVA-1.5	Nüwa	51.64	62.56	41.01	51.73	29.7	18.0	33.8	63.7	70.6	99.2
	IVC-Prune	41.01	48.15	32.81	40.66	23.2	16.1	33.2	64.0	69.6	92.2
	Ours	59.43	71.43	48.80	59.89	31.0	17.1	33.4	64.3	70.0	98.7
Retained visual tokens: 33.33%											
Qwen2.5-VL	Nüwa	77.50	83.01	70.51	77.00	78.8	78.0	60.4	83.1	78.0	95.9
	IVC-Prune	80.42	86.54	72.94	79.97	69.4	76.2	60.7	83.0	79.4	93.7
	Ours	81.79	87.41	74.08	81.09	83.2	80.9	62.1	83.1	79.1	98.5
LLaVA-1.5	Nüwa	51.69	62.56	41.09	51.78	30.6	17.9	33.8	63.7	70.7	99.6
	IVC-Prune	44.58	52.48	36.10	44.39	26.1	16.8	33.3	64.0	71.5	95.5
	Ours	60.15	71.31	48.93	60.13	31.2	16.9	33.5	63.9	70.2	98.6

Our method achieves the highest grounding accuracy among the three methods on every evaluated split at both budgets. For general VQA, it also achieves the highest mean retention on Qwen2.5-VL-7B (98.0%/98.5%). On LLaVA-1.5-7B, Nüwa achieves higher mean retention (99.2%/99.6% versus our 98.7%/98.6%), with stronger ChartQA scores, while our method performs better on OCRBench. Thus, the grounding advantage persists across these budgets, while the VQA comparison depends on the backbone and task.

A.2 Extension to LLaVA-1.5-13B

To examine applicability to a larger language backbone, we separately train a scorer for frozen LLaVA-1.5-13B, retaining 64 of 576 visual tokens (11.1%) after zero-based decoder block 15. Table 9 reports full-split RefCOCO+ results and the same five VQA tasks used above. Training and inference settings are given in Appendix C.2.

Table 9: **LLaVA-1.5-13B grounding and general VQA.** Ours retains 64/576 visual tokens. Grounding reports full-split Acc@0.5 (%); Avg. averages the three RefCOCO+ splits. VQA scores use the units of Table 4. Ret. averages score ratios over the five reported VQA tasks (%).

		Grounding (RefCOCO+)				General VQA					
Model	Method	Val	TestA	TestB	Avg.	OCR	Chart	MMStar	MMB	VQAv2	Ret.
LLaVA-1.5-13B	Unpruned	59.86	67.69	48.54	58.70	33.3	18.2	35.7	68.2	74.4	100.0
	Ours	66.80	74.89	54.06	65.25	32.9	18.0	36.1	68.7	72.0	99.3

Our method improves Acc@0.5 over the unpruned model on all three RefCOCO+ splits and preserves 99.3% of its mean five-task VQA performance. These results support applicability to the 13B backbone; comparisons against other pruning methods at this scale remain to be evaluated.

A.3 What Does GSR Learn?

To examine whether GSR changes token selection in a meaningful direction, let S_{LPL} and S_{GSR} denote the Top-K sets before and after GSR. We define GSR-promoted tokens as $\Delta S^{+} = S_{\mathrm{GSR}} \setminus S_{\mathrm{LPL}}$ and GSR-discarded tokens as $\Delta S^{-} = S_{\mathrm{LPL}} \setminus S_{\mathrm{GSR}}$. We add these tokens to GlimpsePrune (GP) via set union, $S^{\pm}_{\mathrm{GP}} = S_{\mathrm{GP}} \cup \Delta S^{\pm}$, and evaluate GP at the same per-example token count as S^{+}_{GP} for a fair comparison.

Table 10: **Grounding utility of GSR-promoted and discarded tokens.** Entries report RefCOCO+ validation Acc@0.5 (%). We define $\Delta S^{+} = S_{\mathrm{GSR}} \setminus S_{\mathrm{LPL}}$ and $\Delta S^{-} = S_{\mathrm{LPL}} \setminus S_{\mathrm{GSR}}$, and transfer either set to GlimpsePrune (GP) by simple union. The matched-budget GP row uses the same per-example token count as GP $\cup\, \Delta S^{+}$. Retention is the average fraction of visual tokens retained.

Qwen2.5-VL

Retention	Selection	Acc@0.5 ↑
11.1%	GP	67.38
14.29%	GP (matched budget)	70.56
13.68%	GP $\cup\, \Delta S^{-}$	70.09
14.29%	GP $\cup\, \Delta S^{+}$	**79.91**

LLaVA-1.5

Retention	Selection	Acc@0.5 ↑
11.1%	GP	47.71
15.76%	GP (matched budget)	50.53
15.96%	GP $\cup\, \Delta S^{-}$	49.59
15.76%	GP $\cup\, \Delta S^{+}$	**53.82**

Table 10 shows that adding GSR-promoted tokens (ΔS^{+}) improves grounding performance compared with adding GSR-discarded tokens or matching the GP selection budget. This indicates that GSR preferentially selects visual tokens with greater downstream grounding utility, supporting its role in refining *what to keep* after LPL establishes *where to look*.

B Additional Method Details

B.1 Scorer Formulation

The normalized spatial descriptor used in Eq. 1 is

$$\mathbf{c}_i = [x_i, y_i, a_i, r_i], \tag{9}$$

where (x_i, y_i) are normalized grid coordinates, a_i is the token's normalized area, and r_i is the grid aspect ratio. For probe j, head h, and token i, cross-attention uses

$$A^{h}_{ji} = \frac{\exp\big((W_Q \mathbf{p}_j)^{\top}_h (W_K \mathbf{z}_i)_h / \sqrt{d_h}\big)}{\sum_{t=1}^{N} \exp\big((W_Q \mathbf{p}_j)^{\top}_h (W_K \mathbf{z}_t)_h / \sqrt{d_h}\big)}. \tag{10}$$

In Eq. 3, the factor $N/(PH)$ divides the aggregate attention by its value under uniform attention, so $s_i > 0$ when token i receives above-uniform attention for any token count N; we set $\epsilon = 10^{-6}$.

Implementation conventions. Paper layer indices are one-based; configuration indices are zero-based. Qwen2.5-VL uses the last valid user-message text token for $\mathbf{q}^{\ell}$ and the last full-prompt token for the attention cue. LLaVA-1.5 uses the position immediately before the assistant prefix for both.

B.2 LPL Objective

For $p_i = \sigma(s_i)$ and binary region labels y_i, the losses in Eq. 5 are

$$\mathcal{L}_{\mathrm{Dice}} = 1 - \frac{2\sum_i p_i y_i + \epsilon}{\sum_i p_i + \sum_i y_i + \epsilon}, \qquad \mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N}\sum_i \left[y_i \log p_i + (1 - y_i) \log(1 - p_i) \right]. \tag{11}$$

B.3 GSR Objective and Sampling

Gumbel Top-K samples an ordered subset without replacement, inducing a Plackett–Luce distribution. Let $\pi^{(g)}$ denote the ordering induced by the perturbed scores for stochastic candidate

$g \in \{2, \ldots, G\}$. Its top-K log-likelihood is

$$\log p_\theta(\pi^{(g)}_{1:K} \mid I, q) = \sum_{t=1}^{K} \left[\frac{s_{\pi^{(g)}_t}}{\tau} - \log \sum_{j=t}^{N} \exp\left(\frac{s_{\pi^{(g)}_j}}{\tau} \right) \right]. \tag{12}$$

Equation 8 averages these log-likelihood terms over the $G - 1$ stochastic candidates after normalization by K, weighted by detached advantages. The group mean averages the rewards of all G candidates, including the greedy candidate. Because this group mean includes each stochastic candidate's own reward, we describe the update as a group-relative policy-gradient surrogate without claiming an unbiased gradient estimator.

Training and inference. The two phases train only the lightweight scorer while keeping the MLLM frozen. LPL initializes the scorer from region supervision, and GSR continues from the LPL checkpoint using sampled fixed-budget subsets. Inference removes Gumbel perturbations and rollout evaluation, applying a single deterministic Top-K selection.

C Experimental Settings and Reproducibility

C.1 Training Data and Deduplication

Both backbones and all pruning-depth variants use the same two data pools and seed 42. The exact annotations, images, source identifiers, file hashes, and consumed row IDs are archived in `release/data`.

- **LPL (Localization Prior Learning):** We use 8,000 bounding-box-annotated Visual-CoT examples (Shao et al., 2024), comprising 2,000 examples each from GQA, InfoGraphicsVQA, DocVQA, and TextVQA. We retain the original questions and region annotations for token-level box-overlap supervision with Dice and BCE losses. Training performs 1,000 optimizer updates with eight examples accumulated per update, consuming each example once.
- **GSR (Grounding-Guided Selection Refinement):** We use 8,000 image-unique examples from RefCOCO (UNC; 1,604), RefCOCO+ (UNC; 1,614) (Yu et al., 2016), and RefCOCOg (UMD; 4,782) (Mao et al., 2016). Their COCO image IDs do not overlap with the eight grounding evaluation splits. GSR performs 4,000 update steps with 24 candidate subsets per instance, using the same 4,000 examples for both backbones.

C.2 Model and Training Configurations

Table 11 lists settings and training schedules for the two 7B backbones. Both scorers use $d = 256$, $P = 4$ probes, $H = 4$ heads, and dropout 0.05. AdamW learning rates are 10^{-4} for LPL and 5×10^{-6} for GSR; LPL loss weights are $\lambda_{\text{Dice}} = 1.0$ and $\lambda_{\text{BCE}} = 0.1$. GSR uses one greedy and 23 Gumbel-perturbed candidates ($\tau = 0.4$).

Table 11: **Model and training configurations.** Decoder block indices are zero-based; trainable parameters include the full lightweight scorer ($< 0.1\%$ of either backbone).

Backbone	Block ℓ	Budget	Scorer params	Updates LPL / GSR	G
LLaVA-1.5-7B	12	64 / 576	≈ 2.38M	1,000 / 4,000	24
Qwen2.5-VL-7B	16	11.1%	≈ 2.12M	1,000 / 4,000	24

LLaVA-1.5-13B extension. The separately trained 13B scorer uses seed 42 and the checkpoint after 4,000 GSR updates. It retains 64 of 576 visual tokens after zero-based block 15. Evaluation uses the stock LLaVA prompt configuration, FP16 backbone inference, and SDPA attention, with batch size 32 for grounding and 16 for VQA.

C.3 Evaluation Protocols

Coordinate normalization and postprocessing. Different multimodal backbones use distinct coordinate output conventions. The evaluation protocol converts all predictions to canonical $[0, 1]^4$ normalized coordinates:

- **Qwen2.5-VL-7B:** Generates bounding boxes as absolute pixel coordinates within a smart-resized canvas. During postprocessing, predictions exceeding the image canvas are clamped to the canvas boundary, and the wrapper normalizes them by dividing by the canvas dimensions.
- **LLaVA-1.5-7B:** Generates coordinates directly normalized relative to the original image canvas.

Each wrapper converts predictions to canonical $[0, 1]$ coordinates exactly once. Outputs that cannot be parsed as valid bounding boxes or contain reversed corner orders are assigned an accuracy of 0.

Prompt standardization. For Qwen2.5-VL grounding evaluations, all methods use the prompt:

```
\nLocate {expression} and output the bbox coordinates in JSON format.
```

with the system message *"You are a helpful assistant."* to replace the default lmms-eval prompts. LLaVA-1.5 uses the default prompt settings from `lmms-eval`.

General-VQA evaluation settings. The general multimodal evaluations in Table 4 cover the 10-task full suite: VQAv2-Lite, GQA, OCRBench, ChartQA, TextVQA, MMBench, MMStar, MME-P, CV-Bench 2D, and AI2D. Retention is computed as the arithmetic mean of the ten task-score ratios relative to the unpruned model. The released Qwen and LLaVA Ours runs use the system prompt *"You are a helpful assistant."*. Task-specific question templates and answer formatting follow the recorded `lmms-eval` task configurations. For TextVQA, the scores use the validation split.

The supplementary comparisons in Tables 8 and 9 report OCRBench, ChartQA, MMStar, MMBench, and VQAv2-Lite. Their VQA retention averages these five task-score ratios, whereas Table 4 averages all ten.

C.4 Pilot Study Protocol

The token-addition experiments in Section 3 are evaluated on RefCOCO+ validation (10,758 referring expressions). Evaluations use identical prompt templates and coordinate postprocessing as the main grounding benchmark. Visual budgets are set as fractions of the image's visual tokens: base budget $K_{\text{base}} = 15\%$ and final budget $K_{\text{final}} = 25\%$. On Qwen2.5-VL with dynamic resolution, this corresponds to an average of 55 tokens at 15% and 92 tokens at 25%; on LLaVA-1.5 with 576 tokens, these correspond to 86 and 144 tokens, respectively.

Oracle Target ranks tokens by squared Euclidean distance to the ground-truth box: tokens inside the box are prioritized by distance to the box center, followed by outside tokens by distance to the box boundary. *+Target* adds tokens via the same ranking to those outside the native 15% selection. *+Context* arranges uniformly spaced 2D grid anchors across the canvas and greedily assigns each anchor its nearest unselected visual token.

C.5 Efficiency Measurement

Analytical prefill FLOPs estimation. To quantify the computational reductions reported in Table 5, we analytically estimate decoder-prefill FLOPs on the measured benchmark workload itself: each of the 32 RefCOCO validation images is tokenized, and the estimate averages the per-sample cost over the workload (LLaVA-1.5: 105 text + 576 visual tokens, retaining 64; Qwen2.5-VL: 37 text + 350 visual tokens at dynamic resolution, retaining $V/9$). For a decoder block with hidden dimension d, intermediate MLP dimension m, and total key/value projection dimension d_{kv}, the computational cost for a sequence of length n is:

$$f(n) = 4nd(d + d_{\text{kv}}) + 4n^2d + 6ndm, \tag{13}$$

accounting for Q/K/V/O projections, attention matrix multiplications, and the SwiGLU feed-forward network (counting two FLOPs per multiply-add). Dense inference across L decoder layers costs

$L \cdot f(n)$, while a method pruning after layer p costs:

$$\text{FLOPs} = (p+1) \cdot f(n) + (L-p-1) \cdot f(n'), \tag{14}$$

where n' is the shortened sequence length, plus each method's selection/scorer overhead (our probe $\approx$1.3 GFLOPs; IVC-Prune's value-similarity selection $\approx$2.7 GFLOPs; GlimpsePrune's ViP computed with its official estimator, $\approx$9.1/5.5 GFLOPs on LLaVA/Qwen). The vision tower, LM head, and autoregressive decoding are excluded.

For our method, Qwen2.5-VL-7B executes 17 dense-sequence blocks followed by 11 shortened blocks, while LLaVA-1.5-7B executes 13 dense-sequence blocks followed by 19 shortened blocks. The resulting estimates and baseline comparisons are reported in Table 5.

Empirical benchmarking protocol. All efficiency measurements are conducted on a single NVIDIA GeForce RTX 4090 GPU (48 GB variant) under PyTorch 2.8.0, CUDA 12.8, and Transformers 4.51.3. Both Qwen2.5-VL-7B and LLaVA-1.5-7B process the same 32 distinct RefCOCO validation images at batch size 32, with eight warmup batches and four full-workload repeats. Generation is fixed to 32 output tokens for all methods.

The evaluation framework is the vendored `lmms-eval` 0.7.2 under Python 3.12. Qwen uses BF16 with FlashAttention 2; LLaVA uses FP16 with PyTorch SDPA.

D CASE STUDIES

Figures 3–5 present three visualization examples on RefCOCO validation using Qwen2.5-VL-7B, comparing GlimpsePrune, IVC-Prune, Nüwa, and our method under a shared per-case visual-token budget of approximately 11.1%.

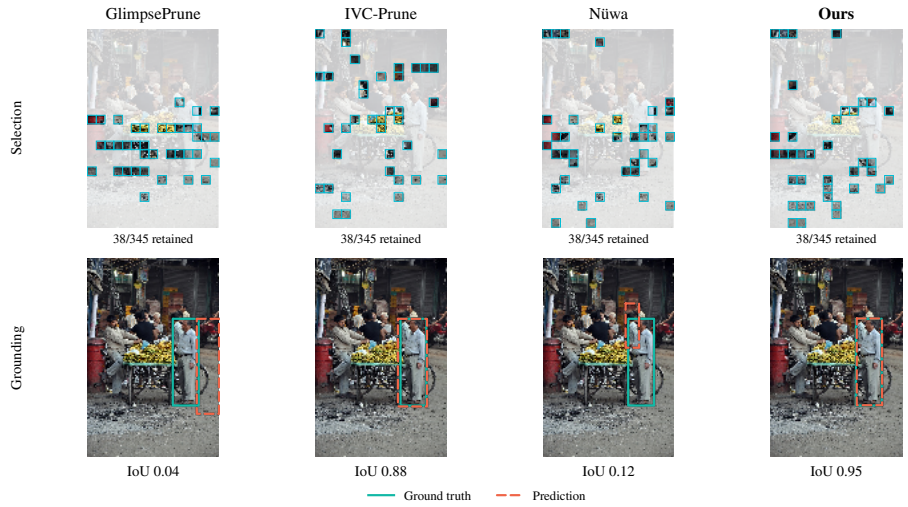

Figure 3: **Case A: Appearance and relational reference.** Expression: *"man white shirt next to bananas."* Our method accurately localizes the target (IoU = 0.95) by combining appearance features with surrounding spatial context. Each method retains 38 out of 345 visual tokens ($\approx$ 11.0%). Solid teal boxes indicate ground truth; dashed coral boxes indicate predictions.

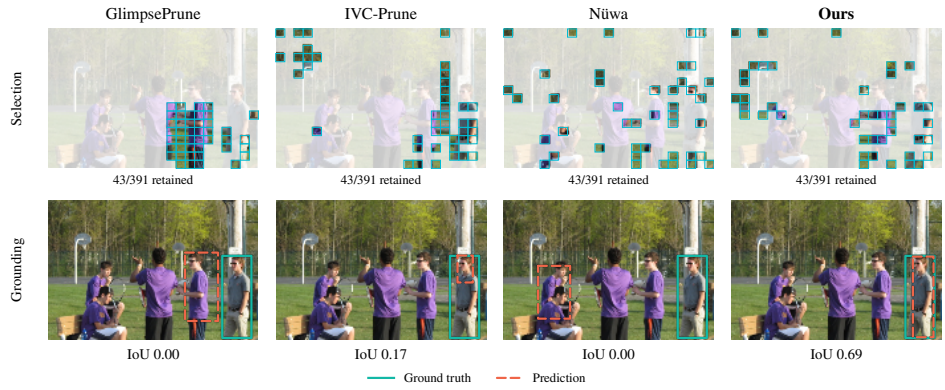

Figure 4: **Case B: Person disambiguation.** Expression: *"dude who's way too cool to wear a purple t-shirt."* Our method achieves IoU $= 0.69$ by attending to the distinguishing attribute and surrounding context to resolve among multiple people. Each method retains 43 out of 391 visual tokens ($\approx 11.0\%$). Solid teal boxes indicate ground truth; dashed coral boxes indicate predictions.



Figure 5: **Case C: Failure — incomplete localization.** Expression: *"A LADY WITH A BLACK SWIM SUIT HOLDING A UNBRELLA ON THE BEACH."* Our method partially localizes the target (IoU $= 0.37$), covering only part of the subject. Nüwa localizes the full target more accurately in this case. Each method retains 43 out of 391 visual tokens ($\approx 11.0\%$). Solid teal boxes indicate ground truth; dashed coral boxes indicate predictions.