WHEN TEXT MATTERS: DESIGN PRINCIPLES FOR VISUAL TOKEN PRUNING IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token pruning has been widely studied as a practical approach to reducing the computational cost of large vision-language models. However, it struggles to preserve essential visual information, which can lead to substantial performance degradation. In particular, image-based token selection can overlook task-relevant details, while text-guided token selection may fail to capture the text–visual relationships needed for complex reasoning. We find that applying textual guidance too early can limit its ability to identify answer-relevant visual regions, whereas text-to-visual attention becomes more informative at intermediate decoder depths. This finding motivates our training-free method, which separates early vision-guided pruning from deferred text-guided reselection. We first prune visual tokens using vision-encoder attention, retain additional candidates until the decoder midpoint, and then use text-to-visual attention to determine the final visual-token set. Across eight benchmarks and three models, our method outperforms the best-performing baselines by an average of 11.10 and 16.84 percentage points in performance recovery at 80% and 90% pruning, respectively, with comparable or lower LLM-prefill latency than most baselines.

1 INTRODUCTION

Large vision-language models (VLMs) support diverse tasks, including visual question answering, image captioning, and multimodal reasoning (Liu et al., 2023; Dai et al., 2023). Their capabilities, however, come with substantial computational costs: representing images with numerous visual tokens increases the sequence length processed by the language model, particularly for high-resolution inputs (Chen et al., 2024a; Wang et al., 2024a). Visual token pruning addresses this overhead by shortening the visual sequence. Training-free approaches are especially practical, as they reduce inference costs without updating model parameters or training an additional selector (Chen et al., 2024a; Zhang et al., 2024).

The central challenge is deciding which visual information to preserve. Image-based token selection uses visual importance, redundancy, or sensitivity to identify tokens without conditioning on the prompt (Yang et al., 2025a; Kim et al., 2026). Text-guided token selection instead uses interactions between textual and visual tokens to identify information relevant to the requested task (Xing et al., 2024; Zhang et al., 2024). Despite these differences, both approaches can discard essential visual information, leading to substantial performance degradation.

Figure 1 illustrates representative failure cases of image-based and text-guided visual token pruning. For image-based pruning, ZOO-Prune (Kim et al., 2026) retains prominent visual features of the calculator but discards the region containing the brand name requested by the question. For text-guided pruning, SparseVLM (Zhang et al., 2024) retains support for the phrase “I was the victim” but loses the corresponding value, illustrating a difficulty in capturing the relationships between textual and visual information needed to answer the question. Full-benchmark evaluations further reveal substantial performance losses. At 80% pruning on Qwen3-VL-8B, ZOO-Prune loses 28.8% and 34.7% relative to Dense on TextVQA and TextCaps, respectively, which rely on scene-text understanding (Singh et al., 2019; Sidorov et al., 2020). SparseVLM loses more than 50% on ChartQA and In-

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.

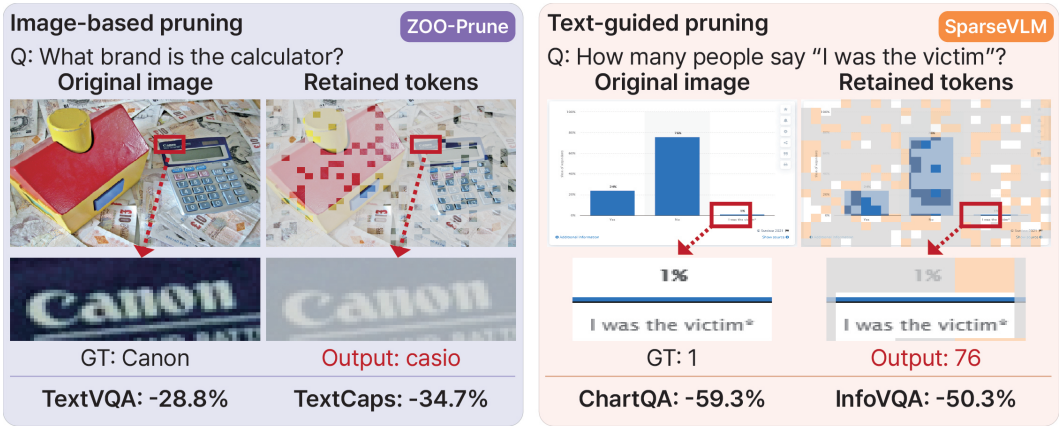

Figure 1: Failure cases on Qwen3-VL-8B at 80% pruning. Image-based visual token pruning discards the visual region containing the requested text, while text-guided visual token pruning preserves related support but loses the corresponding value. Gray: discarded regions; orange: merged support. Percentages denote full-benchmark performance losses relative to Dense.

foVQA, which require complex reasoning over relationships among textual and visual elements (Masry et al., 2022; Mathew et al., 2022).

These failures highlight the difficulty of identifying the visual information a VLM needs to reason toward a correct answer. This raises a key question: how can textual guidance be used more effectively to identify visual evidence that is relevant to the prompt? To investigate this, we analyze the ability of text-to-visual attention to distinguish answer-relevant regions across decoder depths. We find that this ability becomes substantially stronger at intermediate layers than near the input (Section 3.1), indicating that the timing of textual guidance is critical to effective token selection.

Building on this finding, we introduce a simple training-free method that combines early image-based pruning with deferred text-guided selection. Specifically, we first prune visual tokens before the LLM using vision-encoder attention, while retaining additional candidates beyond the final token budget. At the decoder midpoint, text-to-visual attention reselects from these candidates to form the final token set. This design deliberately defers textual guidance to the midpoint until it becomes more informative, while relying solely on image information for early pruning. Across eight benchmarks and three VLMs, our method consistently achieves the highest average Dense-relative performance retention across all evaluated pruning ratios, with the margin over the strongest baseline increasing to 11.10 and 16.84 percentage points at 80% and 90% pruning, respectively.

2 Related Work

2.1 Large Vision-Language Models

VLMs extend large language models (Brown et al., 2020; Grattafiori et al., 2024) with visual encoders and multimodal alignment (Li et al., 2023). CogVLM introduces visual expert modules for deeper vision–language fusion, while Idefics2 systematically studies architecture and training choices for effective multimodal learning (Wang et al., 2024b; Laurençon et al., 2024). Models such as LLaVA-1.5, LLaVA-OneVision, and InternVL advance visual understanding through improved architectures, training, and visual representations (Liu et al., 2024; Li et al., 2024; Chen et al., 2024c), while high-resolution processing increases visual-token costs (Guo et al., 2024). We evaluate our token pruning method on Qwen3-VL and LLaVA-OneVision-1.5 (Bai et al., 2025; An et al., 2025).

2.2 Visual Token Pruning

Reducing the computational cost of VLMs has been explored through knowledge distillation (Hinton et al., 2015; Cao et al., 2025), quantization (Li et al., 2025; Xiang et al., 2026), and visual token

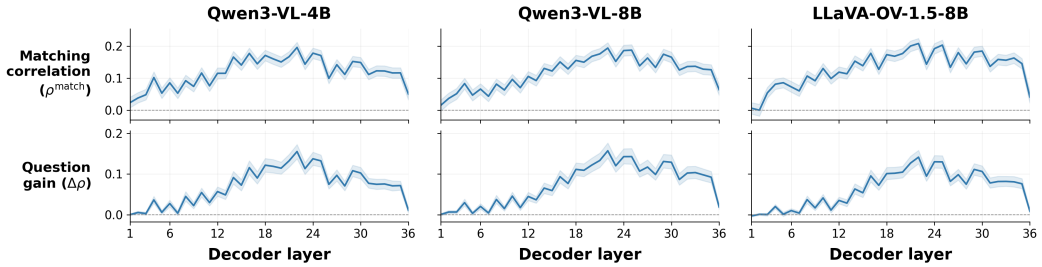

Figure 2: Text-guided evidence identification across decoder depth. Top: correlation with masking-based region importance. Bottom: matching-versus-swapped question gain. Results average four QA benchmarks equally, each with 300 images and two questions per image.

Table 1: Selection-depth ablation at 80% pruning under matched decoder visual-token processing budgets. Observed peak denotes block 22, where the mean question gain in Figure 2 is highest. Values are eight-task Dense-relative recovery (%), with the best results in bold.

Boundary	Qwen3-VL-4B	Qwen3-VL-8B	LLaVA-OV-1.5-8B	Mean
$D/4$ (9 blocks)	82.119	86.551	85.898	84.856
$2D/4$ (18 blocks)	**90.860**	**94.787**	**92.769**	**92.806**
Observed peak (22 blocks)	89.185	94.014	92.299	91.833
$3D/4$ (27 blocks)	87.508	93.431	91.162	90.701

reduction. Token reduction encompasses merging in vision transformers (Bolya et al., 2022; Norouzi et al., 2024) and training-based pruning and merging for VLMs (Cao et al., 2023; 2024). Training-free approaches use pretrained signals to select tokens, drawing on visual importance, diversity, and sensitivity (Zhang et al., 2025), text–visual attention and instruction-conditioned relevance (Zhang et al., 2026; Liang et al., 2026), or combinations of visual saliency, textual relevance, and diversity (Liu et al., 2026). Complementary strategies preserve information through merging or recycling (Yang et al., 2025b) and correct token-reduction distortions (Cho et al., 2026). Building on these studies, we examine when textual guidance becomes informative for visual token selection, motivating a separation between early visual pruning and later text-guided reselection.

3 Method

3.1 Motivation and Design Rationale

To determine when text-to-visual attention provides a reliable signal for token selection, we evaluate its ability to identify answer-relevant visual regions across decoder depth on four multimodal QA benchmarks (Singh et al., 2019; Masry et al., 2022; Mathew et al., 2022; Kembhavi et al., 2016). For each benchmark, we sample 300 images with two distinct questions per image and divide each image's native visual-token grid into eight spatial regions. Using the unpruned model (Dense), we mask one region at a time throughout all decoder blocks and measure its effect on the correct answer. Let $\mathcal{L}_{\text{NLL}}(\mathcal{M})$ denote the token-averaged negative log-likelihood of the correct answer when regions $\mathcal{M}$ are masked. We define the importance of region $\mathcal{R}_r$ as

$$h_r = \mathcal{L}_{\text{NLL}}(\mathcal{R}_r) - \mathcal{L}_{\text{NLL}}(\varnothing), \tag{1}$$

where $\varnothing$ denotes the unmasked condition. A larger h_r means that masking region $\mathcal{R}_r$ has a larger negative effect on answering the question. We next ask whether text-to-visual attention identifies these important regions and whether this ability is specifically conditioned on the question. For this analysis, we use two different questions q and q' about each image. For both questions, we define

$$h^u = [h^u_1, \ldots, h^u_8], \qquad s^u_\ell = [s^u_{\ell,1}, \ldots, s^u_{\ell,8}], \qquad u \in \{q, q'\}, \tag{2}$$

where h^u contains the masking-based importance of the eight regions with respect to the correct answer for question u, and s^u_ℓ contains their region-level text-to-visual attention at decoder block ℓ.

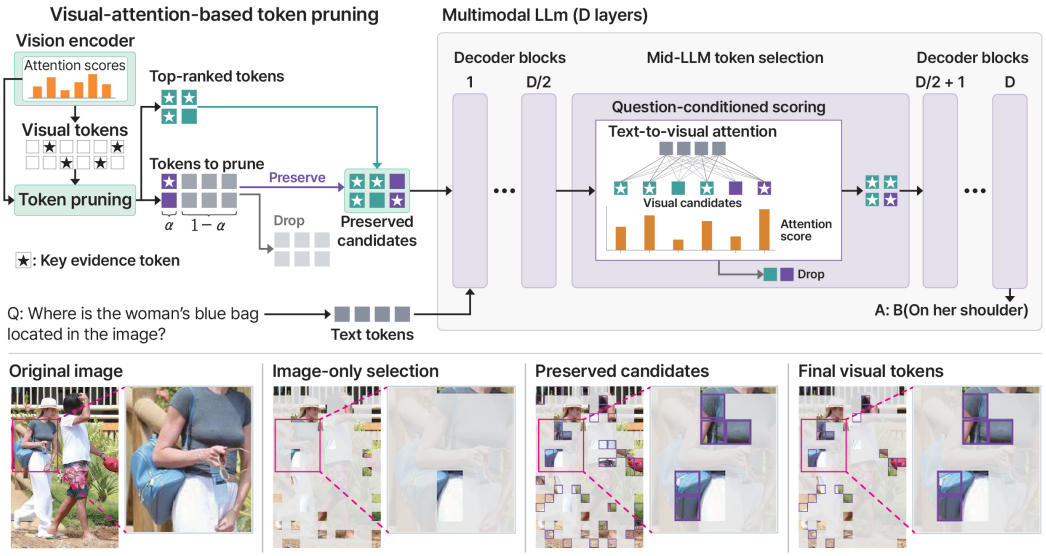

Figure 3: Overview of our two-stage pruning. Before the LLM, vision-encoder attention guides visual-token pruning while preserving an expanded candidate set. At the decoder midpoint, text-to-visual attention selects the final visual-token set. The reserve fraction $\alpha \in [0.1, 0.3]$ controls the number of additional candidates; we use $\alpha = 0.2$ by default.

We first test whether each question's attention ranks the important regions for answering that question. Let $\rho(\cdot, \cdot)$ denote Spearman rank correlation. Averaging over the two questions, we define

$$\rho_\ell^{\text{match}} = \frac{1}{2}\left[\rho(s_\ell^q, h^q) + \rho(s_\ell^{q'}, h^{q'})\right]. \tag{3}$$

A larger ρ_ℓ^{match} indicates better agreement between text-to-visual attention and the corresponding region importance.

We next test whether this alignment is specific to the question by comparing each question's region importance with attention from the other question while keeping the image fixed:

$$\rho_\ell^{\text{swap}} = \frac{1}{2}\left[\rho(s_\ell^{q'}, h^q) + \rho(s_\ell^q, h^{q'})\right], \qquad \Delta\rho_\ell = \rho_\ell^{\text{match}} - \rho_\ell^{\text{swap}}. \tag{4}$$

Here, ρ_ℓ^{swap} measures how well attention from the other question ranks the important regions for the current question. A positive $\Delta\rho_\ell$ therefore means that the matching question identifies its own important regions better than the other question. Figure 2 reports ρ_ℓ^{match} in the top row and $\Delta\rho_\ell$ in the bottom row. Both are weak in the early decoder layers but become substantially stronger at intermediate depths. This shows that text-to-visual attention provides limited guidance near the decoder input, while becoming much more effective at identifying question-relevant visual evidence after sufficient decoder processing.

These results motivate deferring text-guided selection beyond the early decoder layers. The reselection depth must also account for efficiency: a deeper boundary processes retained visual tokens through more decoder blocks, so fewer can be retained under the same visual-token processing budget. We therefore ablate the reselection depth in our method under matched budgets. Table 1 shows that the decoder midpoint achieves the highest recovery across all three models. Together, the depth-wise analysis and budget-controlled ablation motivate the decoder midpoint as the reselection boundary, where textual guidance is informative while sufficient candidates can be retained at practical cost. We therefore use $L = D/2$ for text-guided reselection in our method.

3.2 Proposed Method

Early visual pruning with reserve. As illustrated in Figure 3, our method first performs vision-guided pruning before the LLM while preserving additional visual candidates for later reselection.

Prior training-free pruning methods have shown that attention within the vision encoder provides an effective signal for identifying visually informative tokens (Yang et al., 2025a; Zhang et al., 2025; Kim et al., 2026). We therefore use this signal for pre-LLM pruning, avoiding decoder computation. Unlike methods that incorporate text–visual interactions into early token selection (Xing et al., 2024; Zhang et al., 2024), we keep this stage purely vision-based, following our observation in Section 3.1 that textual guidance is less informative in early decoder layers.

Let N be the number of prunable visual tokens, D the decoder depth, and $K \approx (1-p)N$ the final token budget at pruning ratio p. We first obtain the head-averaged vision-encoder attention

$$\mathbf{A}^{\text{img}} = \frac{1}{H}\sum_{h=1}^{H} \text{softmax}\left(\frac{\mathbf{Q}_h\mathbf{K}_h^\top}{\sqrt{d_h}}\right), \qquad s_i^{\text{img}} = \begin{cases} A_{\text{cls},i}^{\text{img}}, & \text{with a class token,} \\ \sum_{j\in\mathcal{V}} A_{j,i}^{\text{img}}, & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathcal{V}$ denotes the visual-token positions. Thus, class-token-based encoders use CLS-to-visual attention, whereas encoders without a class token use the self-attention received by each visual token from other visual queries.

Rather than immediately pruning to K tokens, we preserve additional image-ranked candidates:

$$M = \min\{N, K + \lceil \alpha(N-K) \rceil\}, \qquad \mathcal{C} = \text{TopK}_M(s^{\text{img}}), \tag{6}$$

where $\alpha \in [0.1, 0.3]$ is a scalar hyperparameter that controls the fraction of otherwise pruned tokens preserved as additional candidates; we use $\alpha = 0.2$ by default. The set $\mathcal{C}$ contains the image-only top-K together with the next $M-K$ candidates. We process these candidates until the decoder midpoint $L = D/2$, preserving alternatives that may become important once textual guidance becomes sufficiently informative.

Deferred text-guided reselection. At the decoder midpoint, we revise the image-only selection with text-to-visual attention. Let $\mathcal{T}$ and $\mathcal{C}$ denote the input text tokens and visual candidates, respectively. Using the decoder native queries and keys, we compute the head-averaged attention as

$$\mathbf{A}^{\text{text}} = \frac{1}{H}\sum_{h=1}^{H} \text{softmax}\left(\frac{\mathbf{Q}_h^{\mathcal{T}}\mathbf{K}_h^\top}{\sqrt{d_h}}\right), \qquad s_i^{\text{text}} = \sum_{t\in\mathcal{T}} A_{t,i}^{\text{text}}, \tag{7}$$

where the softmax follows the decoder's native attention mask and s_i^{text} is the attention received by candidate i from the input text tokens. We select the final visual-token set as

$$\mathcal{S} = \text{TopK}_K\left(s^{\text{text}}\big|_{\mathcal{C}}\right). \tag{8}$$

Overall, image scores are used only to construct the candidate set $\mathcal{C}$, while the final selection at the decoder midpoint depends solely on text-to-visual attention. This allows candidates outside the initial vision-guided top-K to enter the final set when they become more relevant to the input. By separating early vision-guided pruning from deferred text-guided reselection, the method reduces visual-token processing early without committing to the final token set before textual guidance becomes sufficiently informative.

4 Experiments

4.1 Experimental Setup

Models and baselines. We evaluate Qwen3-VL-4B, Qwen3-VL-8B (Bai et al., 2025), and LLaVA-OneVision-1.5-8B (An et al., 2025). We compare with five training-free visual-token reduction methods: FastV (Chen et al., 2024a), SparseVLM (Zhang et al., 2024), VisPruner (Zhang et al., 2025), ZOO-Prune (Kim et al., 2026), and RESTORE (Cho et al., 2026). Since SparseVLM progressively changes the visual-token count across decoder depth, we match its average decoder token usage to ours for a comparable processing budget.

Benchmarks. We evaluate eight full benchmarks: TextVQA (Singh et al., 2019) for scene-text understanding; ChartQA (Masry et al., 2022) and InfoVQA (Mathew et al., 2022) for chart and infographic reasoning; AI2D (Kembhavi et al., 2016), MMMU (Yue et al., 2024), and MMStar (Chen et al., 2024b) for diagram and general multimodal reasoning; and NoCaps (Agrawal et al., 2019) and TextCaps (Sidorov et al., 2020) for open-ended captioning. Together, they test preservation of both global context and localized key evidence.

Table 2: Qwen3-VL-4B results across eight benchmarks. Avg. Rel.: mean performance relative to Dense (%); higher is better. Bold/underline: best/second-best pruned results at each ratio.

Method	TextVQA	MMMU	AI2D	MMStar	ChartQA	NoCaps	TextCaps	InfoVQA	Avg. Rel.
Dense	81.30	46.22	81.87	56.40	82.92	66.51	81.86	77.43	100.00
Prune 70% of vision tokens									
FastV (ECCV2024)	74.89	43.33	72.22	46.53	45.64	63.16	**84.84**	44.51	83.46
SparseVLM (ICML2025)	73.21	**45.44**	73.87	48.80	50.72	65.68	74.94	45.78	84.46
VisPruner (ICCV2025)	64.76	44.33	72.31	47.07	65.00	66.45	68.25	46.48	83.63
ZOO-Prune (CVPR2026)	65.64	45.11	76.75	51.20	72.84	66.12	58.97	53.92	86.48
RESTORE (ICML2026)	66.17	43.89	72.83	50.67	55.44	**67.03**	73.34	37.77	82.64
Ours	**77.57**	44.78	**78.79**	**52.67**	**75.92**	66.59	81.56	**61.32**	**94.05**
Prune 80% of vision tokens									
FastV (ECCV2024)	68.57	42.44	66.84	42.00	28.72	58.98	**83.07**	33.93	75.11
SparseVLM (ICML2025)	66.59	44.11	71.60	46.93	34.44	63.75	67.98	38.35	77.24
VisPruner (ICCV2025)	48.16	42.56	67.58	44.93	49.08	65.61	50.76	36.53	72.57
ZOO-Prune (CVPR2026)	56.14	44.00	72.44	47.73	63.16	65.91	52.04	43.66	79.07
RESTORE (ICML2026)	55.64	43.67	69.11	48.07	42.12	**66.58**	63.86	30.62	75.13
Ours	**74.91**	**45.11**	**76.78**	**50.53**	**71.20**	66.57	80.57	**53.73**	**90.86**
Prune 90% of vision tokens									
FastV (ECCV2024)	46.46	39.22	63.08	33.00	15.60	47.82	70.71	25.46	60.94
SparseVLM (ICML2025)	48.95	43.78	67.52	42.07	19.36	57.96	53.31	28.15	65.49
VisPruner (ICCV2025)	30.69	41.78	66.48	38.27	32.52	62.27	35.74	27.14	61.09
ZOO-Prune (CVPR2026)	40.33	42.67	67.36	43.67	44.32	63.84	41.14	30.78	67.63
RESTORE (ICML2026)	37.81	43.78	65.77	43.80	28.12	**65.08**	45.93	25.71	65.04
Ours	**69.27**	**44.11**	**72.96**	**47.67**	**55.68**	63.99	**73.21**	**43.54**	**82.91**

Table 3: Qwen3-VL-8B results across eight benchmarks. Avg. Rel.: mean performance relative to Dense (%); higher is better. Bold/underline: best/second-best pruned results at each ratio.

Method	TextVQA	MMMU	AI2D	MMStar	ChartQA	NoCaps	TextCaps	InfoVQA	Avg. Rel.
Dense	82.95	51.67	83.68	62.73	83.48	63.54	81.62	81.19	100.00
Prune 70% of vision tokens									
FastV (ECCV2024)	74.54	49.44	72.73	50.27	43.60	63.11	**84.87**	42.53	82.56
SparseVLM (ICML2025)	76.12	51.33	76.68	54.93	47.52	63.54	78.13	48.95	85.41
VisPruner (ICCV2025)	75.85	**51.56**	78.47	52.87	66.68	63.13	79.61	52.60	88.85
ZOO-Prune (CVPR2026)	69.31	51.00	78.14	56.27	70.80	63.39	63.42	54.70	86.87
RESTORE (ICML2026)	73.33	51.00	78.01	56.00	66.24	**64.27**	78.52	51.00	88.64
Ours	**81.53**	51.44	**82.74**	**59.20**	**78.24**	63.77	84.15	**70.16**	**96.84**
Prune 80% of vision tokens									
FastV (ECCV2024)	67.10	47.11	69.43	47.27	29.12	61.51	83.83	33.96	75.83
SparseVLM (ICML2025)	70.36	50.22	74.51	52.40	33.96	62.99	72.45	40.36	79.11
VisPruner (ICCV2025)	67.88	50.22	74.48	49.73	54.40	62.03	73.52	42.00	81.49
ZOO-Prune (CVPR2026)	59.08	**51.67**	74.58	53.80	59.92	62.70	53.26	43.53	79.43
RESTORE (ICML2026)	64.80	50.22	74.42	51.87	53.28	**64.18**	71.00	40.35	81.05
Ours	**80.48**	51.33	**81.96**	**57.47**	**75.20**	63.49	**83.95**	**64.57**	**94.79**
Prune 90% of vision tokens									
FastV (ECCV2024)	49.64	45.89	66.52	38.40	19.40	55.95	76.82	27.99	66.16
SparseVLM (ICML2025)	54.24	50.00	70.43	46.40	20.40	58.93	58.80	29.73	68.27
VisPruner (ICCV2025)	45.48	48.33	68.10	42.80	30.88	59.08	53.94	30.43	66.44
ZOO-Prune (CVPR2026)	42.04	49.67	68.01	46.33	41.00	60.71	38.17	31.75	66.56
RESTORE (ICML2026)	46.70	49.56	68.39	46.00	30.76	61.57	52.27	30.26	67.79
Ours	**76.93**	**50.56**	**79.86**	**55.53**	**69.12**	**62.88**	**80.71**	**56.83**	**90.65**

Implementation and metrics. We evaluate 70%, 80%, and 90% visual-token pruning with midpoint reselection. We report each benchmark's native metric, while *Avg. Rel.* is the mean task score normalized by the corresponding Dense score; additional details are in Appendix A.

4.2 Main Results

Tables 2–4 show that our method consistently achieves the highest average relative performance across all nine model–pruning settings. The advantage grows under more aggressive pruning: averaged over the three models, our method exceeds the strongest baseline by 11.10 points at 80%

Table 4: LLaVA-OneVision-1.5-8B results across eight benchmarks. Avg. Rel.: mean performance relative to Dense (%); higher is better. Bold/underline: best/second-best pruned results at each ratio.

Method	TextVQA	MMMU	AI2D	MMStar	ChartQA	NoCaps	TextCaps	InfoVQA	Avg. Rel.
Dense	79.71	55.78	84.62	67.33	86.64	110.40	123.13	77.41	100.00
Prune 70% of vision tokens									
FastV (ECCV2024)	66.67	54.33	75.84	51.80	48.20	103.94	114.19	43.81	80.84
SparseVLM (ICML2025)	74.68	55.00	78.21	57.33	62.76	108.44	113.21	48.83	86.94
VisPruner (ICCV2025)	50.50	52.89	74.29	54.20	53.40	109.30	75.89	37.70	74.68
ZOO-Prune (CVPR2026)	71.09	55.22	**81.83**	**62.87**	77.40	**110.07**	107.16	54.16	90.54
RESTORE (ICML2026)	63.90	52.56	75.58	54.13	44.60	103.38	104.05	34.78	77.33
Ours	**77.56**	**56.22**	81.38	62.67	**83.40**	107.32	**120.73**	**64.02**	**95.20**
Prune 80% of vision tokens									
FastV (ECCV2024)	57.01	51.44	73.22	47.47	33.04	98.38	102.52	36.49	72.30
SparseVLM (ICML2025)	67.95	**54.56**	75.97	53.40	46.72	104.13	100.64	39.87	79.21
VisPruner (ICCV2025)	41.84	52.44	71.60	50.40	42.16	105.42	67.30	31.96	68.26
ZOO-Prune (CVPR2026)	65.04	53.33	79.95	60.20	69.68	**108.73**	97.84	44.10	84.56
RESTORE (ICML2026)	58.11	52.78	72.22	51.33	35.32	101.90	94.82	31.02	72.41
Ours	**75.97**	54.44	**80.38**	**61.13**	**81.12**	106.74	**116.32**	**60.89**	**92.77**
Prune 90% of vision tokens									
FastV (ECCV2024)	40.80	51.78	69.04	41.60	20.56	83.65	79.75	29.52	61.22
SparseVLM (ICML2025)	56.89	52.56	72.73	49.40	27.88	94.84	86.05	30.58	69.05
VisPruner (ICCV2025)	30.18	49.56	69.82	44.00	31.52	95.50	53.59	28.38	59.70
ZOO-Prune (CVPR2026)	52.86	50.44	75.23	52.93	52.48	**104.93**	80.48	33.71	73.60
RESTORE (ICML2026)	46.90	50.89	70.05	47.20	26.20	98.31	76.96	28.29	65.16
Ours	**71.66**	**52.89**	**78.21**	**57.80**	**72.72**	103.93	**101.99**	**52.52**	**86.47**

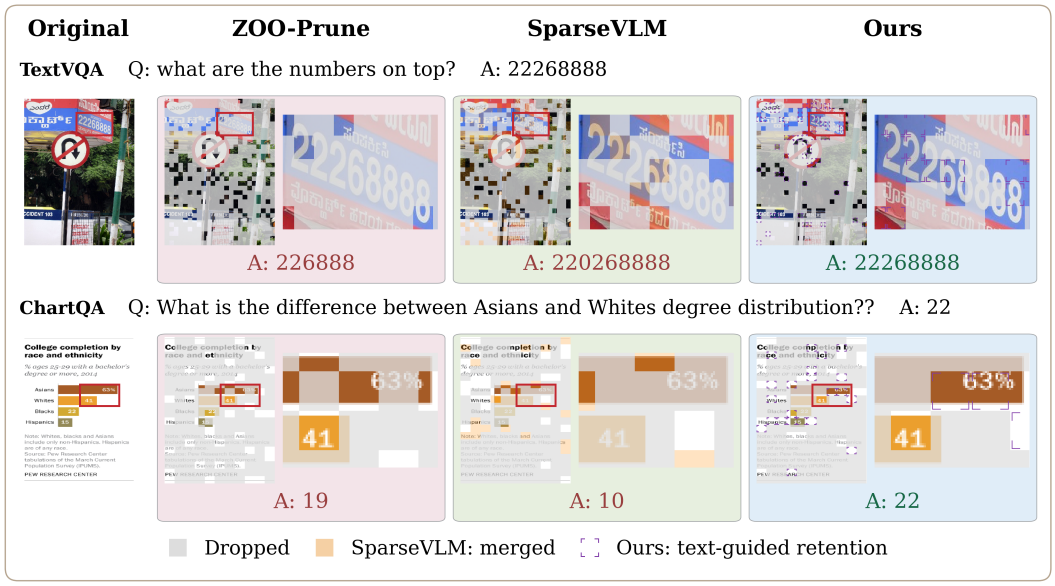

Figure 4: TextVQA and ChartQA examples on Qwen3-VL-8B at 80% pruning. Purple: final tokens outside the image-only top-K; gray: discarded regions; orange: merged support.

pruning and 16.84 points at 90%. Notably, at 90% pruning on Qwen3-VL-8B, our method retains 90.65% of Dense performance, while all competing methods remain below 70%.

The gains are especially pronounced on TextVQA, ChartQA, and InfoVQA, where performance depends on preserving localized, prompt-relevant evidence. Averaged across the three backbones, the margins over the strongest competing results increase from 3.66, 5.51, and 10.91 points at 70% pruning to 8.16, 11.59, and 15.97 points at 80%, and further to 19.26, 19.91, and 18.88 points at 90%, respectively. As pruning becomes more aggressive, early removal is more likely to discard the few tokens containing key evidence; retaining additional candidates until textual guidance becomes informative therefore provides greater benefit in these challenging settings.

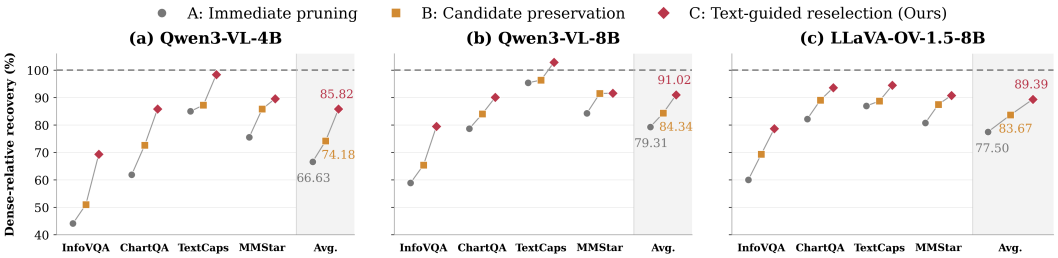

Figure 5: Candidate preservation and text-guided reselection at 80% pruning. A: immediate image-only top-K selection; B: preserve additional candidates until the midpoint, then retain the same initial top-K as A; C: use the same candidates and token-count trajectory as B, but reselect the final K tokens using text-to-visual attention (Ours). Values are Dense-relative recovery (%); the dashed line denotes Dense (100%).

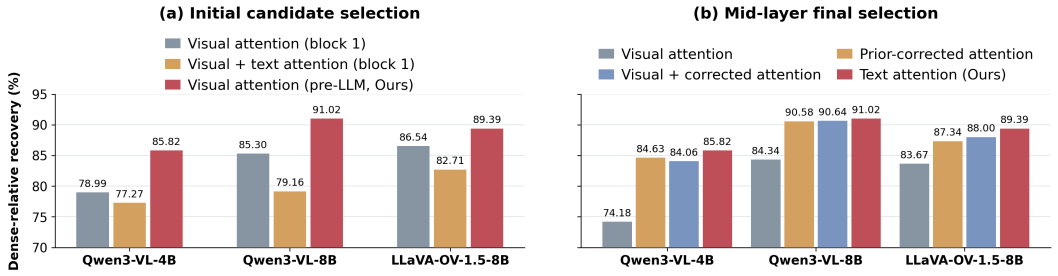

Figure 6: Pruning-score ablations at 80% pruning. (a) Encoder-based candidate selection before the LLM versus after block 1, with or without text attention. (b) Midpoint scoring with fixed candidates and token budgets. Results average Dense-relative recovery across InfoVQA, ChartQA, TextCaps, and MMStar.

Figure 4 further illustrates how deferred selection preserves key evidence. In both examples, answer-relevant regions fall outside the initial image-only top-K but remain in the candidate reserve. Midpoint text-to-visual attention later promotes them into the final set, preserving localized evidence that immediate pruning would discard. This shows how the reserve allows textual guidance to revise the initial visual ranking when relevant evidence is not visually dominant.

4.3 Ablation Studies

Figure 5 separates the contributions of candidate preservation and text-guided reselection. Immediate image-only pruning retains the image-only top-K from the beginning. Candidate preservation instead keeps the larger candidate pool until the midpoint but ultimately retains the same image-only top-K, isolating the benefit of delaying the final pruning decision. Text-guided reselection follows the same token-count trajectory as candidate preservation but reselects the final K tokens using midpoint text-to-visual attention. The substantial gain from candidate preservation to text-guided reselection shows that the improvement is not simply due to retaining more tokens early, but also to using midpoint text-to-visual attention to revise the final selection.

Figure 6 examines which information should guide token selection at each stage. In Figure 6(a), using text-to-visual attention in the first decoder block reduces recovery compared with vision-only pre-LLM pruning across all three models, showing that textual guidance can be detrimental when applied too early. In contrast, Figure 6(b) shows that, with the candidate pool, reselection depth, and final budget fixed, midpoint text-to-visual attention alone achieves the highest average recovery on all three models. These results support a clear separation of roles: vision-encoder attention for early pruning and text-to-visual attention for midpoint reselection.

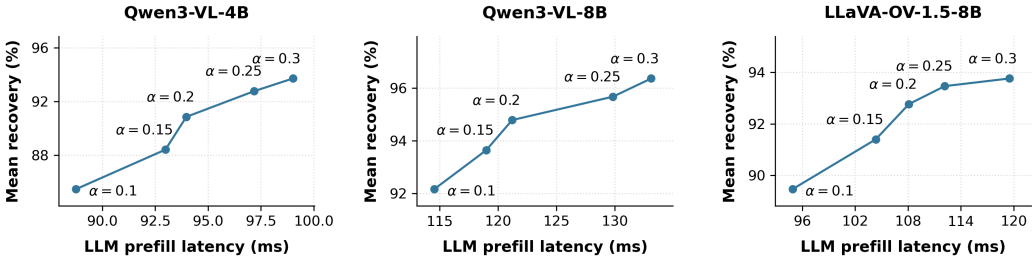

Figure 7: Performance–latency trade-off as the reserve fraction α varies at 80% pruning. Recovery is averaged equally across eight benchmarks relative to Dense. Latency includes token-selection overhead and is averaged over three timed repetitions on one RTX A6000 GPU, using 100 fixed InfoVQA inputs stratified by image size and aspect ratio.

Table 5: Qwen3-VL-8B inference latency (ms). Measurements average three timed repetitions on one RTX A6000 GPU using 100 fixed InfoVQA inputs stratified by image size and aspect ratio. Prefill includes token-selection overhead. End-to-end spans GPU-ready inputs through the first output token, excluding CPU preprocessing and input transfer.

	80% pruning		90% pruning	
Method	LLM prefill	End-to-end	LLM prefill	End-to-end
Dense	226.66	364.13	226.66	364.13
FastV	116.46	256.15	106.97	247.22
SparseVLM	134.79	276.36	122.16	263.63
VisPruner	131.25	292.39	123.80	284.79
ZOO-Prune	137.28	401.81	111.69	375.04
RESTORE	128.87	288.37	116.46	276.09
Ours	121.20	282.35	108.37	268.50

4.4 Analysis

The reserve fraction α controls the trade-off between candidates preserved for midpoint reselection and inference cost. Figure 7 shows that increasing α improves average recovery across all three models at the cost of higher LLM-prefill latency. Across $\alpha \in [0.1, 0.3]$, this trade-off is approximately linear, allowing α to be adjusted to the desired efficiency–accuracy balance. Even $\alpha = 0.1$ exceeds the strongest baseline average on every backbone, and we use $\alpha = 0.2$ by default.

Table 5 shows that our method maintains practical inference latency despite preserving additional visual candidates in the early decoder layers. Unlike many existing approaches that introduce additional token-merging or learned selection modules, our method simply uses vision-encoder attention for early pruning and text-to-visual attention for midpoint reselection. This targeted use of existing model signals yields substantial performance gains with only modest additional inference cost. Additional analyses in the Appendix quantify the use and performance contribution of reselected candidates and extend the comparison to additional progressive pruning methods.

5 Conclusion

We show that textual guidance is most effective for visual token pruning when applied at an appropriate decoder depth. Based on this observation, we introduce a simple training-free method that uses vision-encoder attention for early pruning and text-to-visual attention for midpoint reselection, achieving strong performance preservation across three VLMs and eight benchmarks, especially under aggressive pruning. Despite its simplicity, the method provides substantial gains with practical inference efficiency. We hope this approach can serve as a strong baseline for future work toward more effective and general visual token pruning in large vision-language models.

AI USE STATEMENT

Generative AI was used to assist with manuscript restructuring and revision, English phrasing, literature identification and reference formatting, and to provide feedback on the presentation and interpretation of experimental results. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final text, claims, results, and artifacts.

REPRODUCIBILITY STATEMENT

We provide the model backbones, benchmark datasets, pruning settings, evaluation metrics, and baseline adaptation details used in our experiments. Our method is training-free and requires no additional model optimization. Implementation details for candidate construction, midpoint text-to-visual reselection, evaluation protocols, and latency measurements are provided in the main text and Appendix. The supplementary material further includes the implementation and evaluation code needed to reproduce the reported results.

REFERENCES

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8947–8956. IEEE, 2019.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

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

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jiajun Cao, Yuan Zhang, Tao Huang, Ming Lu, Qizhe Zhang, Ruichuan An, Ningning Ma, and Shanghang Zhang. Move-kd: Knowledge distillation for vlms with mixture of visual encoders. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19846–19856. IEEE, 2025.

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15710–15719. IEEE, 2024.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. Pumer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12890–12903, 2023.

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.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024b.

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

Hyeonwoo Cho, Donghyeon Baek, Yewon Kim, and Bumsub Ham. Improving visual token reduction via rectifying distortions for efficient multimodal llm inference. *arXiv preprint arXiv:2606.01711*, 2026.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, pp. 390–406. Springer, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. Zooprune: training-free token pruning via zeroth-order gradient estimation in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39572–39582, 2026.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37:87874–87907, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

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

Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, et al. Mbq: Modality-balanced quantization for large vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4167–4177. IEEE, 2025.

Yuxuan Liang, Xu Li, Xiaolei Chen, Haotian Chen, Yi Zhen, 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*, 2026.

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

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

Ziniu Liu, Shuheng Zhou, Mingqing Liu, Hao Deng, and Huijia Zhu. Crisprune: Combining contextual relevance and intrinsic saliency for efficient visual token pruning in mllms. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13546–13564, 2026.

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

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591. IEEE, 2022.

Narges Norouzi, Svetlana Orlova, Daan De Geus, and Gijs Dubbelman. Algm: Adaptive local-then-global token merging for efficient semantic segmentation with plain vision transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15773–15782. IEEE, 2024.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *European conference on computer vision*, pp. 742–758. Springer, 2020.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8309–8318. IEEE, 2019.

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

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

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499, 2024b.

Ziwei Xiang, Fanhu Zeng, Hongjian Fang, Rui-Qi Wang, Renxing Chen, Yanan Zhu, Yi Chen, Peipei Yang, and Xu-Yao Zhang. Fine-grained post-training quantization for large vision language models with quantization-aware integrated gradients. *arXiv preprint arXiv:2603.17809*, 2026.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

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

Sihan Yang, Runsen Xu, Chenhang Cui, Tai Wang, Dahua Lin, and Jiangmiao Pang. Vflowopt: A token pruning framework for lmms with visual information flow-guided optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23924–23934. IEEE, 2025b.

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

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

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *Advances in Neural Information Processing Systems*, 38:25438–25468, 2026.

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

A Additional Protocol Details

Benchmarks and metrics. The full evaluation sets contain 5,000 TextVQA (Singh et al., 2019), 900 MMMU (Yue et al., 2024), 3,088 AI2D (Kembhavi et al., 2016), 1,500 MMStar (Chen et al., 2024b), 2,500 ChartQA (Masry et al., 2022), 4,500 NoCaps (Agrawal et al., 2019), 3,166 TextCaps (Sidorov et al., 2020), and 2,801 InfoVQA (Mathew et al., 2022) examples. TextVQA uses VQA scoring without additional OCR text in the prompt; MMMU, AI2D, and MMStar use accuracy; ChartQA uses relaxed correctness; NoCaps and TextCaps use CIDEr; and InfoVQA uses ANLS. CIDEr and ANLS are displayed after multiplication by 100. *Avg. Rel.* is the equal-weighted mean of each task score normalized by the corresponding Dense score and can exceed 100%. Four-task pruning ablations average InfoVQA, ChartQA, TextCaps, and MMStar equally, whereas the depth diagnostic in Figure 2 uses TextVQA, ChartQA, InfoVQA, and AI2D.

Evaluation and baseline adaptation. For task evaluation, Qwen models use FlashAttention-2 except for RESTORE, whose implementation requires SDPA to apply additive attention biases. All methods on LLaVA-OneVision-1.5 use SDPA. KV caching is disabled for all methods to maintain a consistent. evaluation configuration. Within each backbone, all methods share inputs, preprocessing, generation settings, and scorers. Baselines are adapted from their official implementations. FastV retains two initial decoder blocks before pruning. Since SparseVLM progressively changes the number of visual tokens across decoder depth, we match its average decoder visual-token usage to ours rather than its final token count. VisPruner, ZOO-Prune, and RESTORE perform their initial visual-token reduction before the decoder; RESTORE additionally applies position and attention correction inside the language model. Appendix C provides additional comparisons with progressive pruning methods.

A.1 Selection Implementation

Encoder scores. Qwen aggregates vision-encoder attention received by each visual key across visual queries and heads. LLaVA-OneVision uses head-averaged CLS-to-patch attention from the penultimate vision-encoder block. Scores are aligned with each backbone's visual-token geometry before top-M selection.

Midpoint scores. Midpoint scores are computed from the decoder queries and keys using the input prompt tokens as queries. Correct answers and generated answer tokens are not used for token selection. When attention probabilities are not directly exposed by the attention kernel, the required scores are recomputed from the corresponding queries and keys.

Computational budget. For reselection after block L, we measure visual-token processing using the token–block count

$$B_{\text{vis}} = LM + (D - L)K \approx DK + \alpha L(N - K). \tag{9}$$

This measure is used to match decoder visual-token workloads across selection depths and progressive pruning baselines. For the analytical FLOPs results in Figure 8, decoder cost is computed from the sequence length processed before and after reselection and averaged over individual inputs.

A.2 Reserve-Size Trade-off

Figure 8 examines the effect of the reserve fraction α using two measures of computational cost: analytical decoder-prefill FLOPs and GPU-input-to-first-token latency. Each point corresponds to $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30\}$ at 80% pruning, with recovery averaged equally over all eight benchmarks.

Across all three backbones, larger reserves improve recovery while increasing computation, with diminishing gains at larger reserve fractions. The same trend under analytical FLOPs confirms

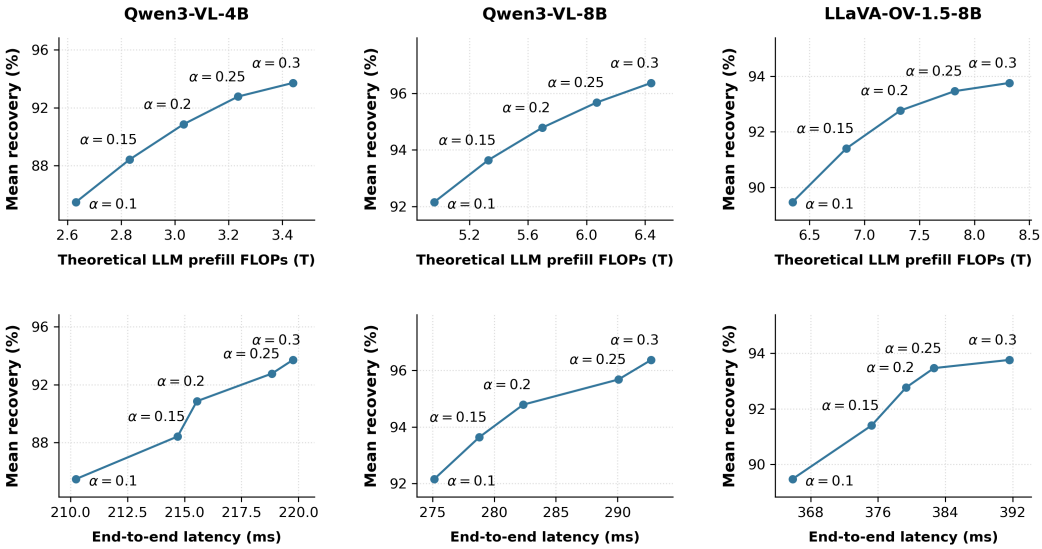

Figure 8: Reserve-size trade-offs at 80% pruning across the three backbones. Recovery is averaged equally across eight benchmarks relative to Dense. Top: analytical decoder-prefill FLOPs, excluding the vision frontend and token-selection overhead. Bottom: measured GPU-input-to-first-token latency, including the vision frontend and token-selection overhead but excluding CPU preprocessing and input transfer. Labels indicate $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30\}$, the fraction of otherwise pruned tokens preserved as additional candidates until midpoint reselection.

Table 6: Effect of candidate preservation and midpoint reselection at 80% pruning. Ret. is the input-averaged percentage of final tokens selected from outside the initial vision-guided top-K. Δ is the Dense-relative recovery gain (percentage points) from candidate preservation to text-guided reselection under the same token-count trajectory.

	Qwen3-VL-4B		Qwen3-VL-8B		LLaVA-OV-1.5-8B	
Task	Ret. (%)	Δ (pp)	Ret. (%)	Δ (pp)	Ret. (%)	Δ (pp)
InfoVQA	42.09	+18.37	42.96	+14.09	44.28	+9.28
ChartQA	39.03	+13.27	42.00	+6.04	41.18	+4.57
TextCaps	40.65	+11.15	41.47	+6.47	44.74	+5.75
MMStar	43.84	+3.78	41.55	+0.11	44.38	+3.27

that this trade-off is not specific to measured latency. The relative increase in first-token latency is smaller because it includes vision processing and other reserve-independent costs. Overall, α directly controls the trade-off between candidate preservation and inference cost.

B Additional Ablation Analysis

Effect of candidate preservation and reselection. Table 6 reports the fraction of final tokens selected from outside the initial vision-guided top-K, together with the recovery gain from candidate preservation to text-guided reselection. Across tasks and backbones, approximately 39–45% of the final visual tokens come from the reserved candidates, showing that midpoint text-to-visual attention substantially revises the initial visual ranking. The largest recovery gains appear on InfoVQA and ChartQA.

Table 7: Comparison with two additional progressive pruning methods on Qwen3-VL-8B under matched per-input visual token–block budgets. The 80% and 90% settings denote Ours' final pruning ratios; baseline final token counts may differ under the matched workloads. Avg. Rel. is the eight-task mean performance relative to Dense. Bold/underline indicate the best/second-best pruned results.

Method	TextVQA	MMMU	AI2D	MMStar	ChartQA	NoCaps	TextCaps	InfoVQA	Avg. Rel.
Dense	82.95	51.67	83.68	62.73	83.48	63.54	81.62	81.19	100.00
Matched to Ours at 80% pruning									
PyramidDrop	61.77	48.22	70.76	47.33	45.80	62.41	66.92	33.58	75.53
FlowCut	<u>77.23</u>	<u>50.44</u>	<u>75.71</u>	<u>51.07</u>	<u>50.40</u>	<u>62.47</u>	<u>83.83</u>	<u>47.29</u>	<u>85.28</u>
Ours	**80.48**	**51.33**	**81.96**	**57.47**	**75.20**	**63.49**	**83.95**	**64.57**	**94.79**
Matched to Ours at 90% pruning									
PyramidDrop	43.18	47.78	67.78	41.33	25.28	57.12	48.34	25.87	62.83
FlowCut	<u>72.50</u>	<u>49.89</u>	<u>71.99</u>	<u>49.87</u>	<u>44.32</u>	<u>61.64</u>	<u>79.10</u>	<u>42.97</u>	<u>81.18</u>
Ours	**76.93**	**50.56**	**79.86**	**55.53**	**69.12**	**62.88**	**80.71**	**56.83**	**90.65**

C Comparison with Progressive Pruning

Table 7 extends our evaluation to two additional progressive pruning methods, PyramidDrop (Xing et al., 2024) and FlowCut (Tong et al., 2026). To account for their different pruning schedules, we match each baseline to Ours' per-input visual token–block budget at $\alpha = 0.2$, up to integer rounding. The 80% and 90% settings denote Ours' final pruning ratios; progressive baselines may therefore use different final token counts under the matched workloads. Ours achieves the highest average relative performance at both budgets. At 80% pruning, Ours exceeds FlowCut and PyramidDrop by 9.51 and 19.26 percentage points, respectively; at 90%, the margins increase to 9.47 and 27.82 points.

D Qualitative Results

Figures 9–16 provide qualitative comparisons across all eight benchmarks at 80% pruning on three VLMs. The examples highlight a recurring limitation of existing pruning strategies: visually small but task-relevant evidence can be removed even when surrounding or semantically related regions are preserved. This is particularly visible on TextVQA, ChartQA, and InfoVQA, where answering correctly often depends on retaining localized text, numbers, or chart elements. ZOO-Prune can preserve visually salient regions while missing such details, whereas SparseVLM can retain related visual support yet lose the precise evidence or relationships required for the answer.

In contrast, our deferred reselection can preserve these initially lower-ranked candidates until text-to-visual attention becomes more informative, allowing the final token set to better retain question-relevant evidence. The AI2D, MMMU, and MMStar examples further show that this behavior extends beyond scene text to diagrams and general multimodal reasoning, where relevant evidence may be spatially sparse or distributed across multiple regions. Finally, the TextCaps and NoCaps examples indicate that the benefit is not limited to question-conditioned tasks: our method also preserves visual information needed for more faithful open-ended descriptions. Together, these examples qualitatively support the main finding that delaying text-guided selection helps preserve task-relevant evidence that early pruning may otherwise discard.



Figure 9: TextVQA examples across three backbones at 80% pruning. Columns compare the original image, ZOO-Prune, SparseVLM, and Ours.

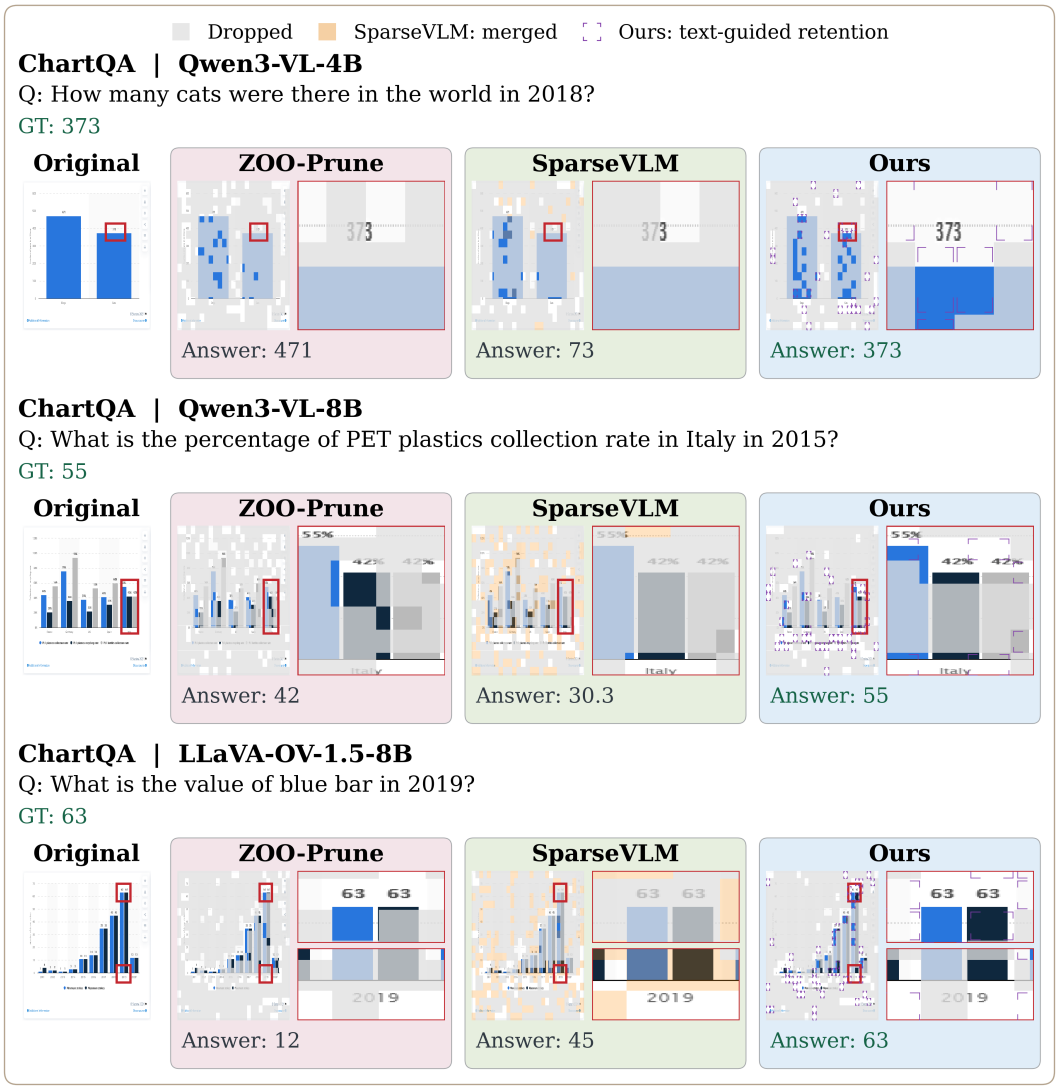

Figure 10: ChartQA examples across three backbones at 80% pruning. Columns compare the original image, ZOO-Prune, SparseVLM, and Ours.

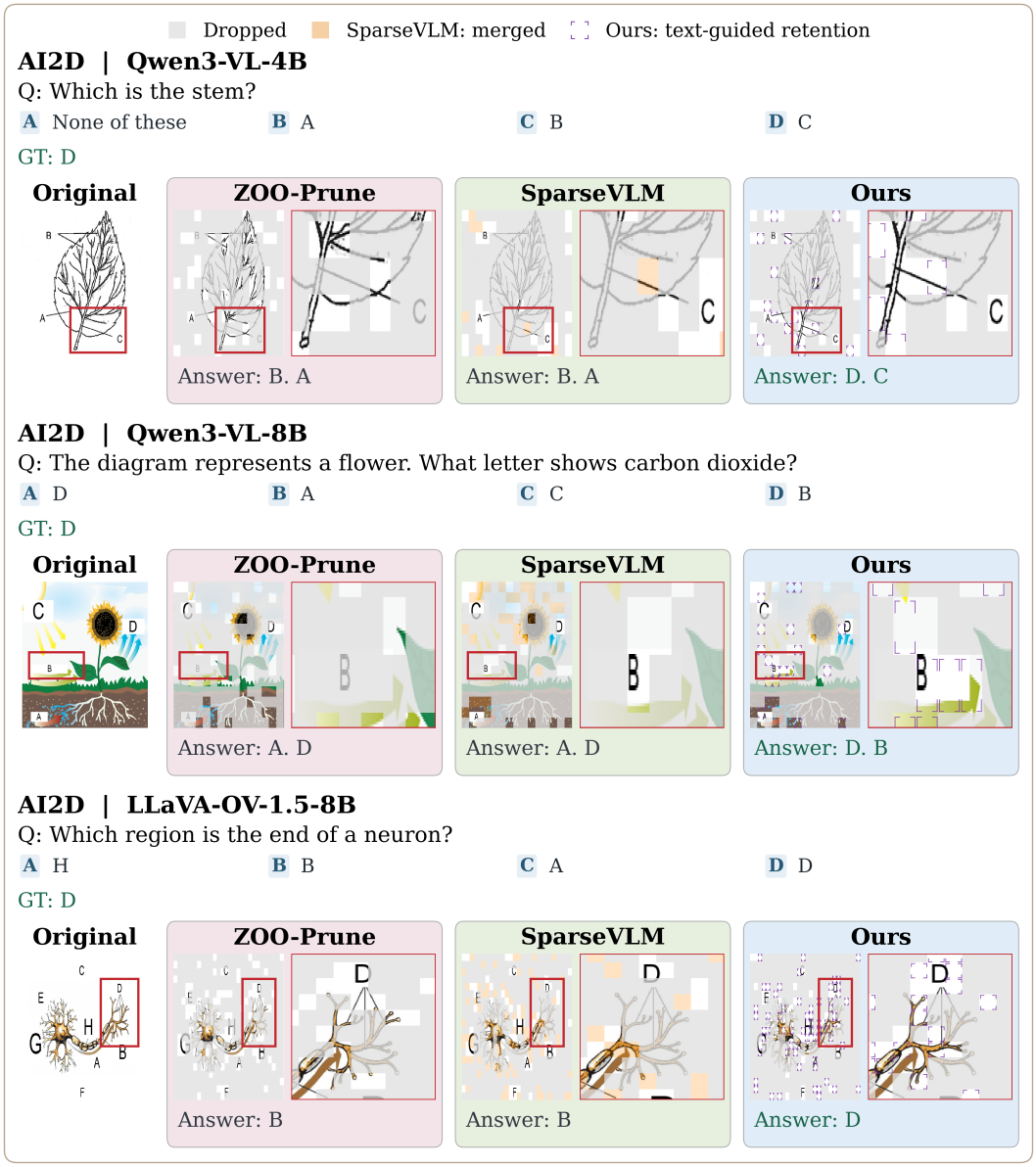

Figure 11: AI2D examples across three backbones at 80% pruning. Columns compare the original image, ZOO-Prune, SparseVLM, and Ours.

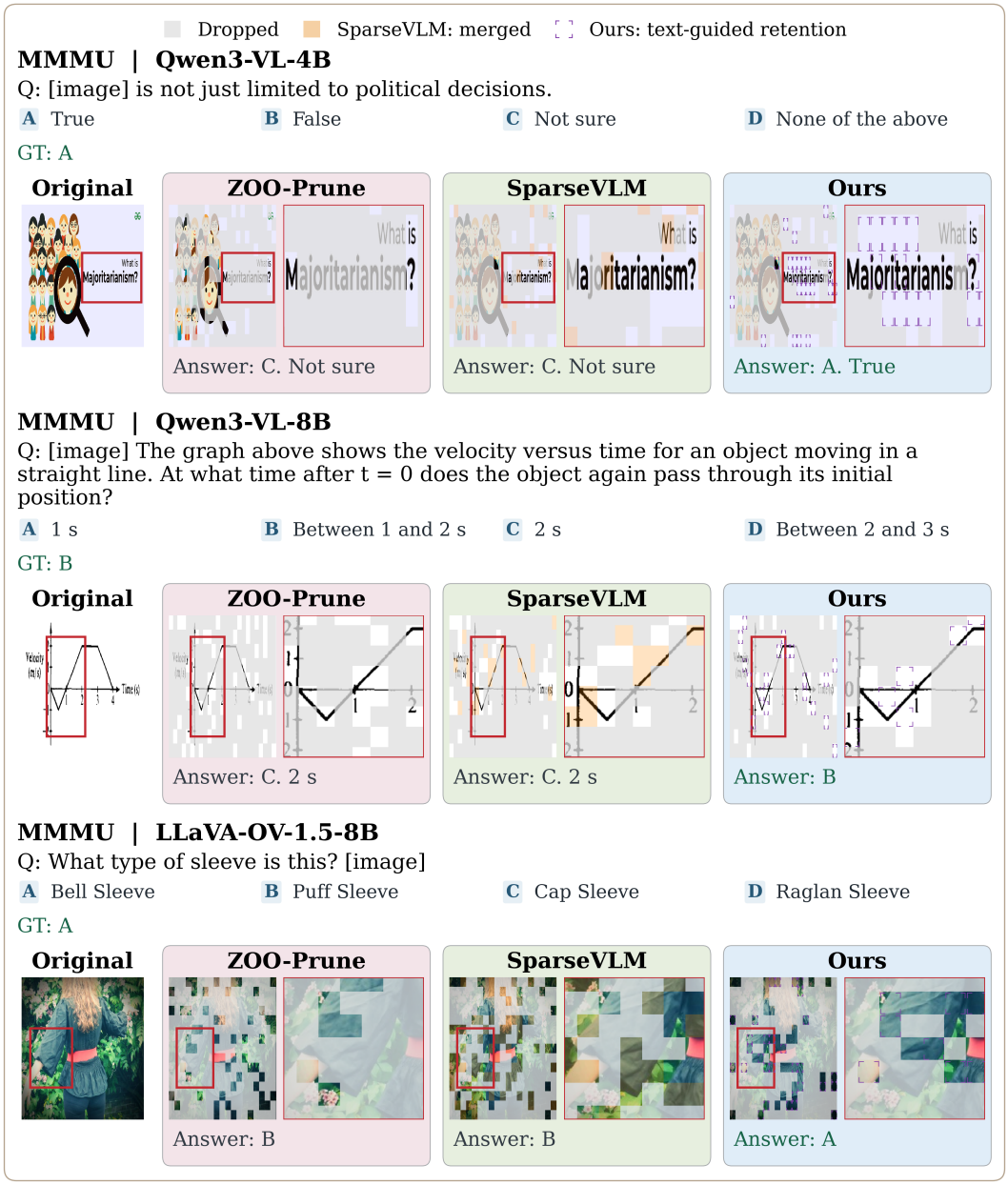

Figure 12: MMMU examples across three backbones at 80% pruning. Columns compare the original image, ZOO-Prune, SparseVLM, and Ours.

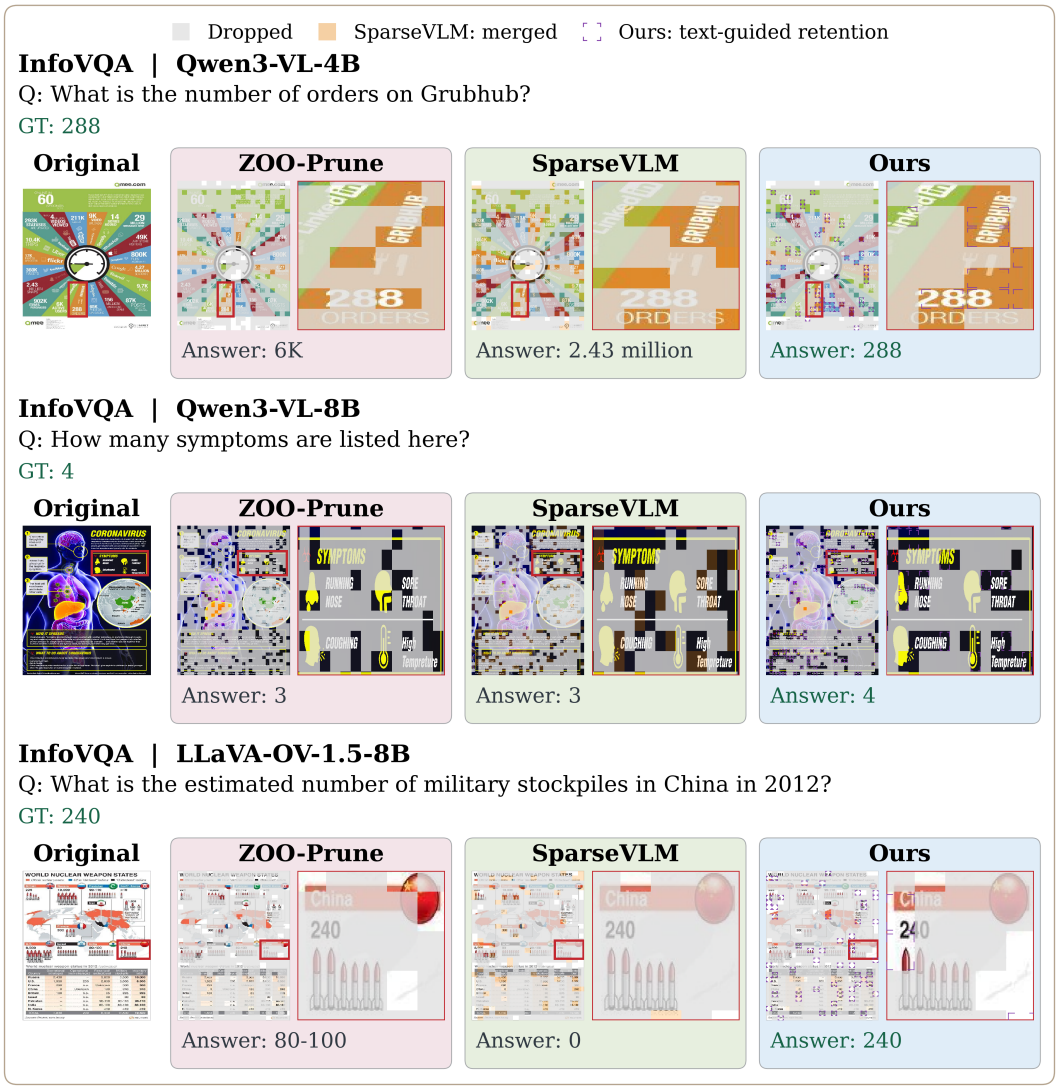

Figure 13: InfoVQA examples across three backbones at 80% pruning. Columns compare the original image, ZOO-Prune, SparseVLM, and Ours.

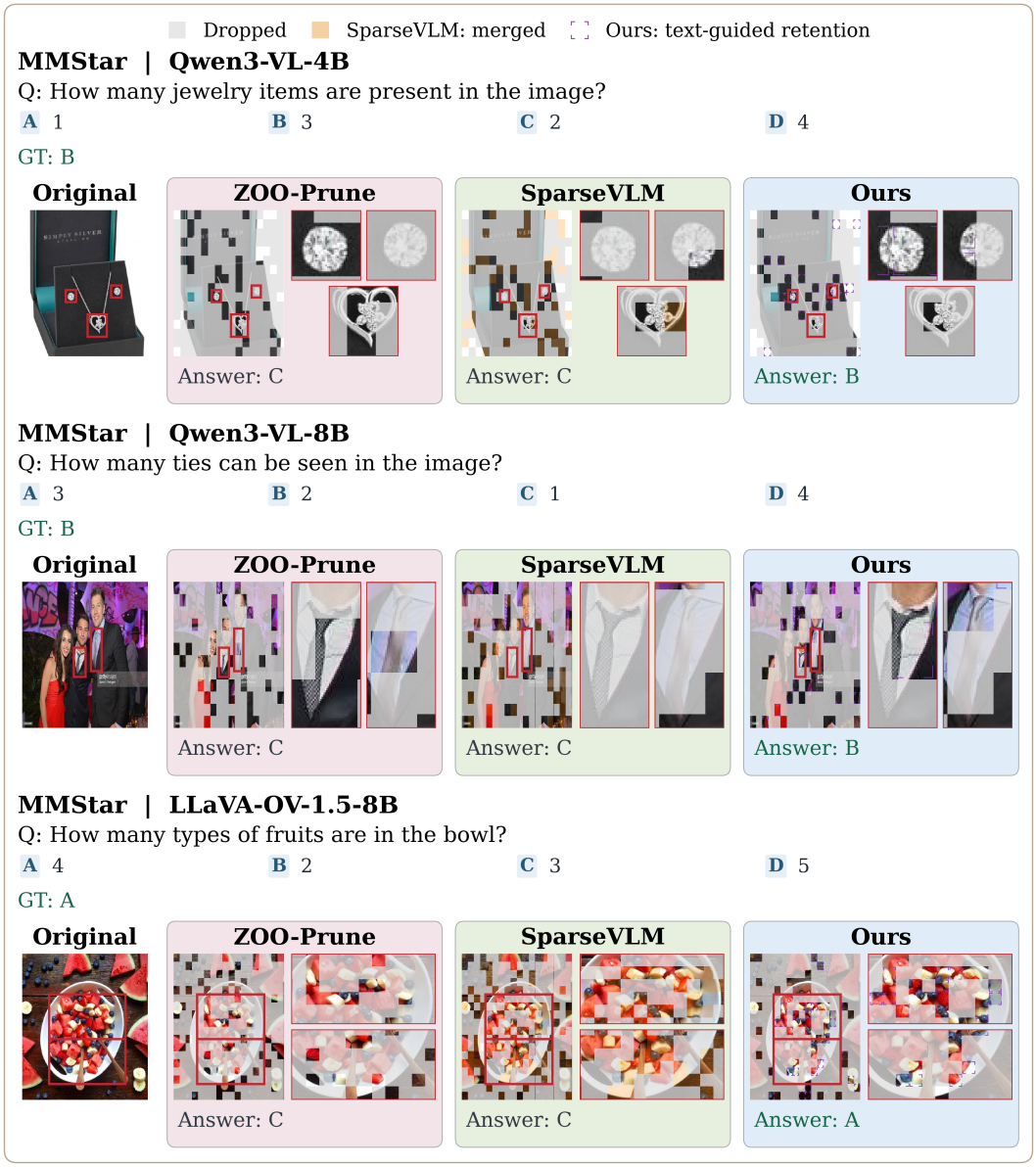

Figure 14: MMStar examples across three backbones at 80% pruning. Columns compare the original image, ZOO-Prune, SparseVLM, and Ours.

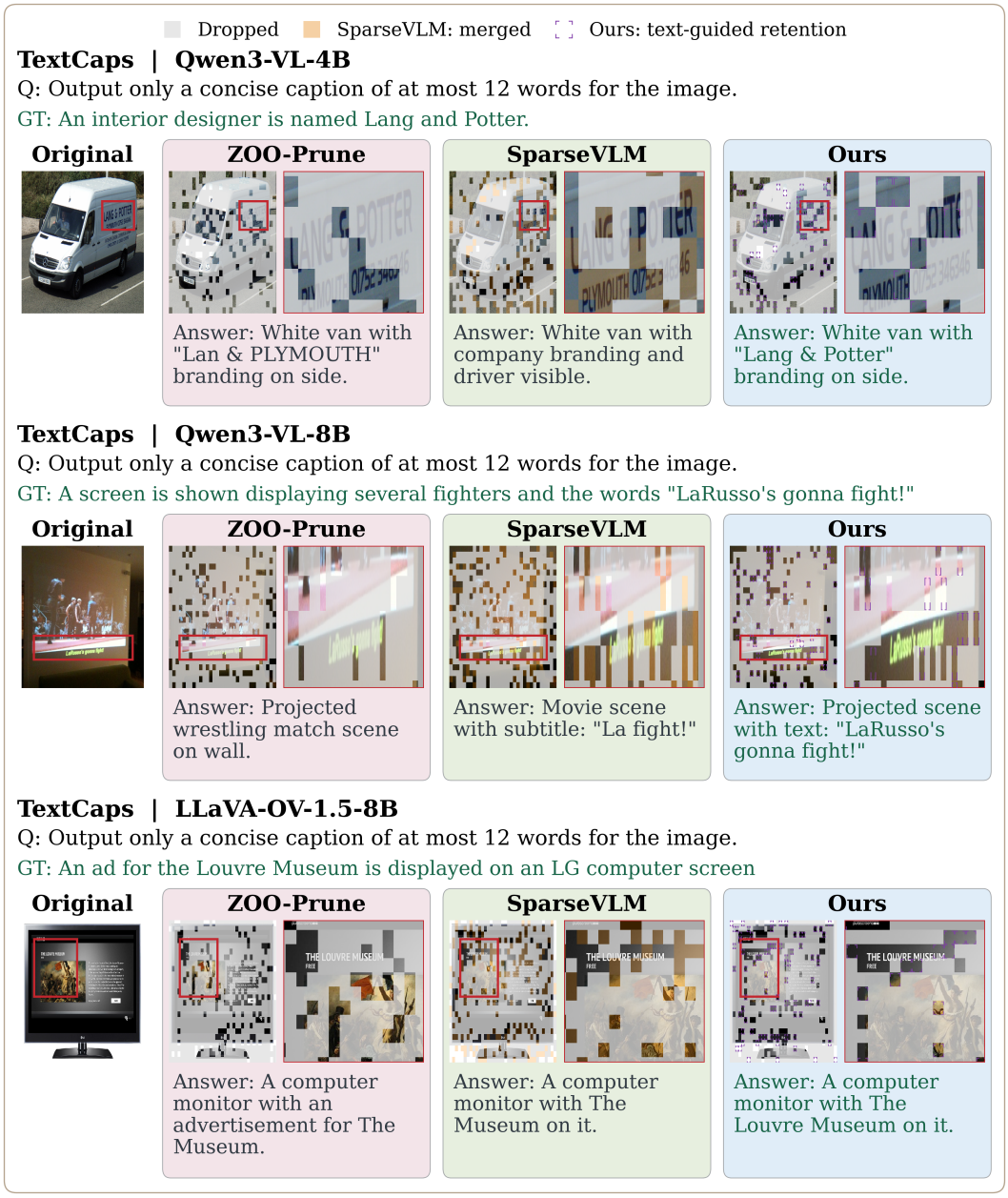

Figure 15: TextCaps examples across three backbones at 80% pruning. Columns compare the original image, ZOO-Prune, SparseVLM, and Ours.

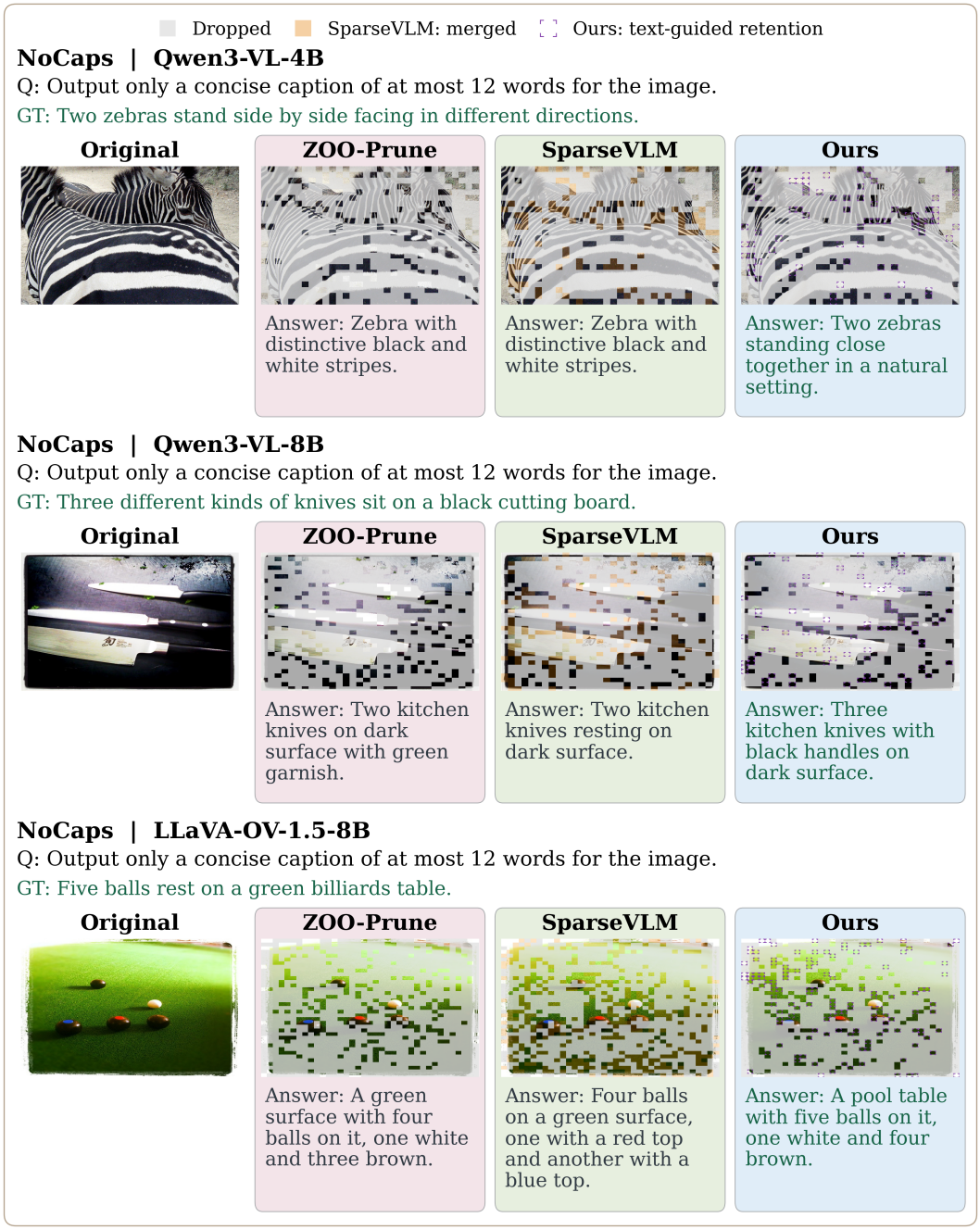

Figure 16: NoCaps examples across three backbones at 80% pruning. Columns compare the original image, ZOO-Prune, SparseVLM, and Ours.