SLIMVLM: SENSITIVITY-AWARE DYNAMIC STRUCTURED PRUNING WITH ADAPTIVE VISUAL TOKEN SELECTION FOR EFFICIENT VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured pruning reduces the parameter footprint of vision-language models (VLMs), but its effectiveness depends on how multimodal inputs are used to estimate component importance. Visual tokens often outnumber text tokens, and including redundant visual content in calibration can impair pruning decisions. We propose **SlimVLM**, a structured pruning framework that combines adaptive visual token selection with sensitivity-aware allocation of pruning ratios. At each decoder layer selected for pruning, SlimVLM uses average text-to-visual attention to select visual tokens for importance estimation while retaining all text tokens. It then adjusts the pruning ratio of each attention or MLP module according to the correlation between its pruned and original outputs, removes attention heads and MLP channels, and compensates for output deviations through least-squares fitting. Token selection operates during calibration; the deployed model uses the reduced parameter structure. Experiments on LLaVA-1.5, LLaVA-NeXT, and Qwen2.5-VL demonstrate the effectiveness of this approach across eight multimodal benchmarks. At a 20% pruning ratio on LLaVA-1.5-7B, SlimVLM retains 93.80% of the unpruned model's average relative performance and leads the evaluated structured-pruning baselines on eight of nine reported task settings. Ablations further show that selecting visual tokens for calibration improves post-pruning performance in both SlimVLM and FLAP.

1 INTRODUCTION

Reducing the parameter footprint of vision-language models (VLMs) supports deployment on resource-constrained devices. By combining visual representations with large language models (LLMs), VLMs (**?**Bai et al., 2025; Wang et al., 2025) support visual question answering (Guo et al., 2022; Yu et al., 2025; Agrawal et al., 2015), multimodal reasoning (Yue et al., 2023), embodied task planning (Mendez-Mendez et al., 2023), and dialogue applications (Li et al., 2017). Their language backbones impose substantial memory and computation costs. Structured pruning reduces these costs by removing attention heads and MLP channels.

VLMs introduce a calibration challenge for parameter pruning: visual and text tokens differ in both number and task relevance. A visual projector maps image-encoder features into the language model's input space, producing hundreds or thousands of visual tokens alongside a shorter text prompt. LLaVA-1.5 (Liu et al., 2024a), for example, uses 576 visual tokens, while high-resolution inputs increase the visual sequence length in models such as Qwen2.5-VL (Bai et al., 2025). Many of these tokens represent redundant or weakly attended image regions (Yang et al., 2024). Methods that compress visual sequences can reduce inference computation (Chen et al., 2024; Liu et al., 2025), but the model's parameter footprint remains unchanged. For structured pruning, visual redundancy affects the estimation of component importance. Component importance thus depends on both the scoring criterion and the calibration tokens.

Our central observation is that calibration tokens influence structured pruning quality. Estimating importance over all visual tokens mixes task-relevant information with redundant visual content. In our ablations, selecting visual tokens using text-to-visual attention improves the performance of the pruned model, both within SlimVLM and when applied to FLAP.

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.

We introduce **SlimVLM**, a structured pruning framework that couples adaptive visual token selection with module-specific pruning ratios. At each decoder layer selected for pruning, SlimVLM averages text-to-visual attention over text queries and selects visual tokens whose scores meet an adaptive threshold. All text tokens are retained. Token selection operates during calibration, and the resulting parameter-pruned model processes its full input sequence at inference. The adaptive threshold allows the retained visual-token count to vary across prompts and layers.

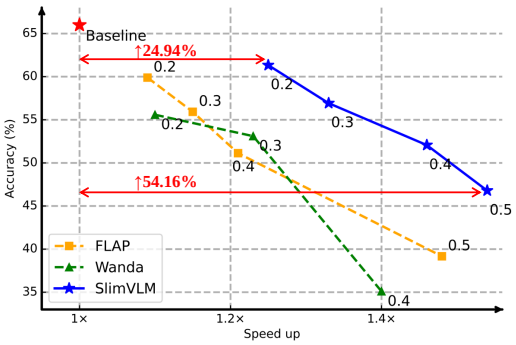

Figure 1: Accuracy and inference speed on ScienceQA under structured pruning.

The framework also accounts for differences in pruning sensitivity across modules. SlimVLM measures the Pearson correlation between a module's pruned and original outputs and uses this signal to adjust its pruning ratio. Attention-head selection searches for subsets that preserve output correlation, while MLP channels are scored using their activations and their contributions in the principal-component basis of the module output. Least-squares fitting compensates for the remaining output deviation. The attention-head procedure applies to multi-head attention and extends to query-head pruning in grouped-query attention.

We evaluate SlimVLM on LLaVA-1.5, LLaVA-NeXT, and Qwen2.5-VL across eight benchmarks, reporting nine task settings including both English and Chinese MMBench. At 20% pruning on LLaVA-1.5-7B, SlimVLM leads the evaluated structured-pruning baselines on eight settings, with scores of 56.44 on MMBench and 54.55 on VizWiz, compared with 52.58 and 48.72 for FLAP. Figure 1 illustrates the accuracy–speed trade-off on ScienceQA. The experiments connect the gains to visual token selection and sensitivity-aware pruning, and examine higher pruning ratios, larger models, and optional recovery through fine-tuning.

Our main contributions are as follows:

- **Adaptive visual token selection.** We propose a text-guided strategy that selects visual calibration tokens using an adaptive threshold on text-to-visual attention. This strategy focuses component-importance estimation on visual content attended to by the prompt while retaining all text tokens.
- **Sensitivity-aware structured pruning.** We introduce SlimVLM, which integrates adaptive token selection with module-specific pruning ratios. The framework uses the correlation between pruned and original outputs to adjust pruning ratios across attention and MLP modules, removes complete heads and channels, and compensates for output deviations.
- **Evaluation across models and pruning settings.** We evaluate SlimVLM on LLaVA-1.5, LLaVA-NeXT, and Qwen2.5-VL across eight multimodal benchmarks, multiple model sizes, and pruning ratios.

2 Related Work

Vision-language models. Open language models such as LLaMA (Touvron et al., 2023) and Qwen (Bai et al., 2023) provide backbones for integrating visual representations with text generation. VLMs align these modalities and process the resulting inputs autoregressively (Yin et al., 2024). Recent models increase language-model capacity and support higher-resolution images to improve multimodal understanding (Bai et al., 2025; Wang et al., 2025). These capabilities increase deployment costs through both model parameters and visual sequence length. Studies of visual-token redundancy show that much of the sequence can carry overlapping information (He et al., 2024b; Alvar et al., 2025). SlimVLM uses this observation to select the calibration information used for parameter pruning.

Parameter pruning. Pruning removes redundant weights or structural components from a neural network (Bercovich et al., 2024; He et al., 2024a; Sengupta et al., 2025). Calibration-based methods

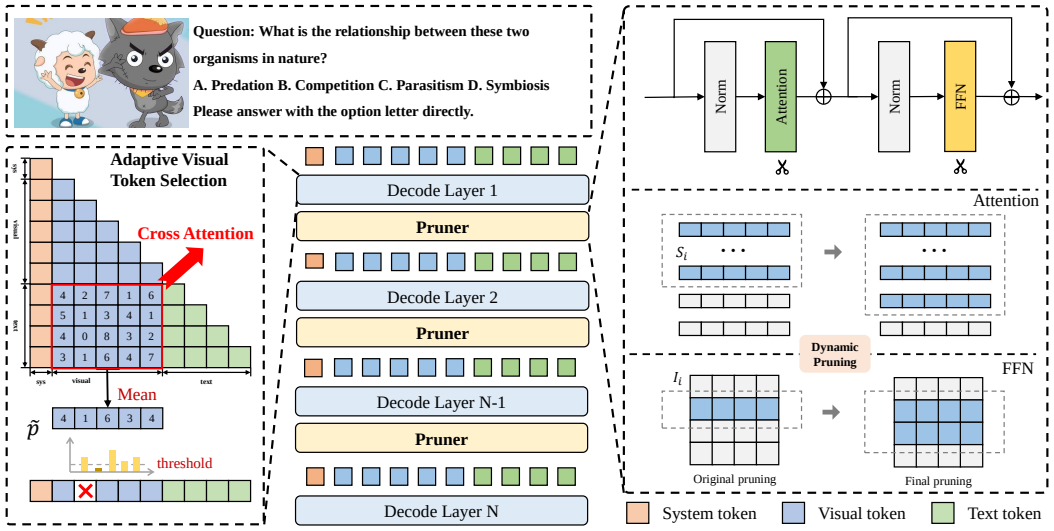

Figure 2: **Overview of SlimVLM.** Adaptive visual token selection uses decoder attention from text queries to visual keys to construct a layer-specific mask for module-importance estimation. Structured pruning removes attention heads and MLP channels using scores S_i and I_i, with module-specific pruning ratios determined by output correlation. The visual-token mask is used during calibration; inference uses the resulting pruned decoder without this mask.

estimate component importance from a small sample of inputs. DLP (Chen et al., 2025) prunes individual weight connections, producing unstructured sparsity whose acceleration depends on hardware and kernel support. Structured methods remove larger units: ShortGPT (Men et al., 2024) removes decoder layers, while SlimLLM (Guo et al., 2025) prunes attention heads and channels using correlation-based head selection, principal-component-based channel scoring, and output regression. SlimLLM allocates layer pruning ratios using input–output similarity. SlimVLM combines text-conditioned selection of visual calibration tokens with module-specific pruning ratios determined by the correlation between pruned and original outputs. This combination targets the multimodal calibration process and the sensitivity of each module to parameter removal.

Visual token compression. VLM token compression reduces inference computation by shortening visual sequences (Yang et al., 2025; Shang et al., 2024; Zhang et al., 2025; Liu et al., 2024c). Attention scores are commonly used to identify tokens for retention or aggregation, with methods operating on encoder representations (Shang et al., 2024; Yang et al., 2024) or within the decoder (Ye et al., 2024; Lin et al., 2024). SlimVLM uses attention-based selection to construct calibration statistics for removing model parameters. This role makes it complementary to inference-time token compression: the structurally pruned model can subsequently use a token compression method to reduce sequence-dependent computation.

3 METHOD

SlimVLM compresses the language decoder of a VLM by pruning attention heads and MLP channels. The framework combines two decisions: which tokens to use when estimating structural importance, and how much to prune from each module. Adaptive visual token selection focuses importance estimation on text tokens and visual tokens attended to by the prompt. Sensitivity-aware pruning then adjusts each module's pruning ratio according to the correlation between its pruned and unpruned outputs. Figure 2 summarizes the framework.

3.1 ADAPTIVE VISUAL TOKEN SELECTION

Visual tokens account for a large share of VLM inputs, so redundant image regions can disproportionately influence structural importance estimates. We select visual tokens using text-to-visual cross attention, defined as the attention from text queries to visual keys within the decoder's causal

self-attention matrix. For each decoder layer selected for pruning, we construct a separate mask from this attention block to select the tokens contributing to module-importance estimation. The mask is confined to calibration and is not applied during inference.

The decoder uses causal self-attention (Vaswani et al., 2017). For a single attention head, let $A \in \mathbb{R}^{L\times L}$ denote the attention matrix over all L visual and text tokens. With the causal mask implicit, the attention weights are written as

$$A = \text{Attention}(Q, K) = \text{Softmax}(\frac{QK^T}{\sqrt{D}}), \tag{1}$$

where $Q \in \mathbb{R}^{L\times D}$ and $K \in \mathbb{R}^{L\times D}$ are the query and key matrices, and D is the dimension of each query and key vector.

We extract the text-to-visual cross-attention block $P \in \mathbb{R}^{L_t\times L_v}$ from A, where L_t and L_v are the numbers of text and visual tokens, respectively:

$$P = [A_{i,j}]_{i\in\mathbb{T},\, j\in\mathbb{I}}\,, \tag{2}$$

where $\mathbb{T}$ and $\mathbb{I}$ index the text prompt and image tokens, respectively. Thus, the rows of P correspond to text queries and its columns to visual keys within the decoder's self-attention matrix.

To aggregate the attention assigned to each visual token across the prompt, we average over text-query positions:

$$p = [p_1, p_2, \cdots, p_{L_v}] = \frac{1}{L_t}\sum_{i=1}^{L_t} P_i, \tag{3}$$

where P_i is the row of visual-token attention weights for the i-th text query. The resulting score vector p provides a prompt-conditioned ranking of visual tokens at the current layer.

Prior work reports that visual-token information becomes sparser in deeper VLM layers (Xing et al., 2024), motivating layer-specific selection. We retain all text tokens and select visual tokens with scores at or above $\alpha = \text{mean}(p)\times\beta$, where β controls the threshold relative to the mean visual-token score. We use $\beta = 0.2$. The selection mask is $M = \{M^T, M^V\}$: every entry of the text mask M^T is True, and the visual mask is

$$M_i^V = \begin{cases}\text{False}, & \text{if } p_i < \alpha,\\ \text{True}, & \text{otherwise.}\end{cases} \tag{4}$$

where True includes the token in importance estimation and False excludes it. This threshold adapts the selected visual-token set to the attention distribution of each layer and prompt.

3.2 Sensitivity-Aware Pruning Ratios

Attention and MLP modules differ in their sensitivity to structural removal. We use the Pearson correlation between a module's pruned and unpruned outputs to guide its pruning ratio:

$$L = \text{Pearson}(O_{pruned}, O_{raw}), \tag{5}$$

where O_{pruned} and O_{raw} denote the outputs of the pruned and unpruned MHA or MLP module. This score measures linear dependence and serves as a sensitivity signal: higher correlation favors a larger pruning ratio. We map the score to a module-specific ratio through

$$r = r_0 e^{\gamma(L/T-1)}, \tag{6}$$

where r_0 is the initial pruning ratio, γ controls the adjustment strength, and T is the target correlation threshold. The adjusted ratio equals r_0 at $L = T$, increases relative to r_0 when $L > T$, and decreases when $L < T$.

After pruning, we fit an affine output correction (Guo et al., 2025) to reduce the discrepancy between the pruned and unpruned outputs:

$$J = \min_{\eta,\zeta} ||(\eta O_{pruned} + \zeta) - O_{raw}||_2, \tag{7}$$

where η and ζ are the fitted scale and offset. These parameters are fused into the output linear layer of the corresponding MHA or MLP module.

3.3 Structured Decoder Pruning

We evaluate correlation-based head scores and output-space channel scores (Guo et al., 2025) on the selected calibration tokens to remove complete attention heads and MLP intermediate channels.

Head pruning. We score each attention head by the Pearson correlation between the original MHA output and the output with that head removed:

$$S_i = \text{Pearson}(O^{MHA}, O^{MHA} - O_i^{MHA}), \tag{8}$$

where O^{MHA} is the full MHA output and O_i^{MHA} is the output contribution of head i. A high S_i indicates that removing the head preserves greater correlation with the original output. For the target pruning ratio, we initialize the retained set with the lowest-scoring heads, whose removal has the largest effect on output correlation. We then iteratively exchange retained and pruned heads to improve the correlation between the retained heads' combined output and the original MHA output.

For decoders with grouped-query attention, we apply head pruning to query heads. The retained query-head count can differ across key–value groups, allowing the pruning pattern to follow head importance within each group.

Channel pruning. We score MLP intermediate channels using their activations and their contributions to the output feature space. Let $W^{down} \in \mathbb{R}^{D\times M}$ be the down-projection matrix, where M is the intermediate width and D is the output width. Principal component analysis of the MLP output $O^{MLP} \in \mathbb{R}^{N\times D}$ yields eigenvalues $E \in \mathbb{R}^D$ and eigenvectors $W' \in \mathbb{R}^{D\times D}$, with N denoting the number of token observations. We weight the principal directions using

$$C = \text{sigmoid}(E/\bar{E}), \tag{9}$$

and compute the importance of channel i as

$$I_i = \left\| \left[C_d (W'_d)^\intercal W_i^{down} \right]_{d=1}^{D} \right\|_2 \cdot \|X_i\|_2, \tag{10}$$

where $\bar{E}$ is the mean eigenvalue, W'_d is the eigenvector for principal component d, W_i^{down} is column i of the down-projection matrix, and X_i contains the corresponding channel activations. The score combines the activation magnitude with the weighted projections of the channel's output direction onto the principal components. Channels with the lowest scores are pruned.

4 Experiments

4.1 Setup

Models. We evaluate SlimVLM on LLaVA-1.5-7B (Liu et al., 2024a), LLaVA-NeXT-7B (Liu et al., 2024b), and Qwen2.5-VL-7B (Bai et al., 2025) to examine different image resolutions and attention architectures. LLaVA-1.5-7B uses a Vicuna-7B language decoder and 576 visual tokens. LLaVA-NeXT supports higher-resolution inputs with up to 2880 visual tokens. Qwen2.5-VL uses grouped-query attention (GQA), allowing us to evaluate pruning with shared key–value heads. We also evaluate LLaVA-1.5-13B to examine scaling with model size.

Benchmarks and metrics. We evaluate eight benchmarks: GQA (Hudson & Manning, 2019), MMBench (Liu et al., 2023), MME (Fu et al., 2023), POPE (Li et al., 2023), ScienceQA (Lu et al., 2022), VQA-v2 (Goyal et al., 2016), TextVQA (Singh et al., 2019), and VizWiz (Gurari et al., 2018). Reporting MMBench in both English and Chinese gives nine task settings. For each setting, we normalize the pruned model's score by the corresponding unpruned model's score. The mean of these nine ratios, expressed as a percentage, is denoted by Average.

Baselines. Our main comparisons use FLAP (An et al., 2023) and a structured variant of Wanda (Sun et al., 2023). We construct the latter by scoring each weight group with its L2 norm and refer to it as Wanda-sp. The main LLaVA comparisons apply the same adaptive visual token selection to all three methods, isolating differences in structured pruning under a common token selection policy. The appendix includes comparisons with LLM-Pruner, LoRAP, Puzzle, and SlimLLM, as well as visual token compression methods.

Implementation details. We estimate module importance using calibration examples sampled from the English and Chinese versions of MMBench, with a default sample size of 24. We prune layers

Table 1: Structured pruning results on LLaVA-1.5-7B and LLaVA-NeXT-7B. Average denotes the mean relative score over the nine task settings, normalized by the corresponding unpruned model. All methods use the same adaptive visual token selection. Bold and underlined values indicate the best and second-best scores at each pruning ratio.

	Ratio	Method	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
LLaVA-1.5-7B	0%	Vanilla	61.94	62.02	50.26	1780.64	85.89	65.99	69.56	48.07	54.48	100
	20%	Wanda-sp	57.19	51.72	40.64	1406.53	84.12	55.58	**66.62**	41.20	37.34	85.31
		FLAP	55.45	52.58	41.24	1645.92	84.59	59.89	64.32	40.59	48.72	89.37
		SlimVLM	**58.00**	**56.44**	**45.88**	**1682.96**	**85.49**	**61.33**	65.90	**41.56**	**54.55**	**93.80**
	40%	Wanda-sp	42.77	23.63	14.66	1215.90	82.39	35.1	53.24	**30.73**	26.17	60.25
		FLAP	42.11	35.74	20.70	796.17	64.31	51.12	45.32	21.87	32.35	59.32
		SlimVLM	**46.44**	**43.57**	**33.81**	**1331.00**	**83.71**	**52.06**	**54.10**	25.31	**44.89**	**75.16**
LLaVA-NeXT-7B	0%	Vanilla	63.67	66.15	55.33	1783.04	87.04	69.36	75.36	64.09	58.78	100
	20%	Wanda-sp	56.97	55.15	27.32	1544.80	85.74	57.56	66.92	57.46	53.66	84.46
		FLAP	58.28	54.64	39.86	1630.89	85.36	59.44	70.46	57.20	**61.92**	89.95
		SlimVLM	**61.22**	**59.52**	**47.28**	**1725.55**	**85.87**	**63.47**	**71.82**	**60.12**	55.18	**93.50**
	40%	Wanda-sp	32.36	24.71	16.75	1092.31	68.09	23.60	45.46	32.21	10.22	46.66
		FLAP	**45.95**	40.98	19.33	798.45	52.29	49.28	54.50	35.34	**47.72**	61.51
		SlimVLM	45.76	**47.14**	**36.78**	**1245.70**	**85.25**	**53.77**	**60.54**	**42.88**	40.66	**74.59**

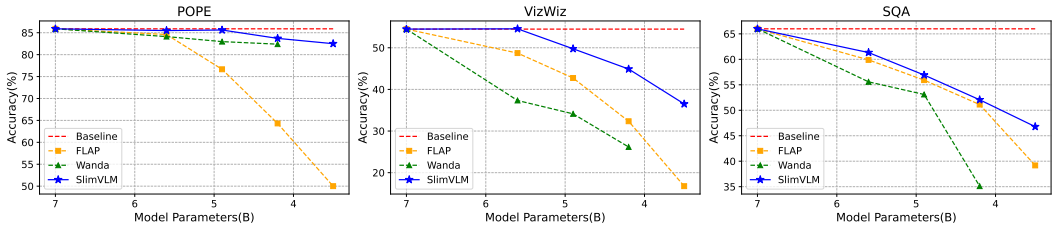

Figure 3: Performance of pruned LLaVA-1.5 across POPE, VizWiz, and ScienceQA at different pruning ratios.

3–30 in the LLaVA series and layers 2–27 in Qwen2.5-VL, retaining the other layers. Pruning runs on a single GPU with 32 GB of memory. At a 20% pruning ratio, Equation 6 uses $\gamma = 5$ and $T = 1$; at 40%, we set $\gamma = 1$. We evaluate models with lmms-eval (Li et al., 2024) on eight GPUs with 32 GB of memory each. For VQA-v2, we use lmms-eval-lite to reduce evaluation memory usage. For Qwen2.5-VL, we set the maximum pixel count to $1280 \times 28 \times 28$.

4.2 Main Results

Performance across tasks. Table 1 compares SlimVLM with Wanda-sp and FLAP on LLaVA-1.5-7B and LLaVA-NeXT-7B. At 20% pruning on LLaVA-1.5-7B, SlimVLM achieves the highest score on eight of the nine task settings. On VizWiz, it scores 54.55, compared with 48.72 for FLAP and 37.34 for Wanda-sp; on English MMBench, the corresponding scores are 56.44, 52.58, and 51.72. Its POPE score remains close to the unpruned model's score, and its VizWiz score is slightly higher. Wanda-sp scores higher on VQA-v2, at 66.62 versus 65.90.

Across the nine task settings, SlimVLM retains 93.80% of the unpruned model's average relative performance at 20% pruning, compared with 89.37% for FLAP and 85.31% for Wanda-sp. These correspond to gains of 4.43 and 8.49 percentage points, respectively.

At 40% pruning, SlimVLM retains 75.16% of the unpruned model's average relative performance, compared with 60.25% for Wanda-sp and 59.32% for FLAP. The gain over the stronger baseline increases to 14.91 percentage points. Performance retention varies by task: SlimVLM scores 83.71 on POPE but falls to 25.31 on TextVQA, corresponding to 52.65% of the unpruned TextVQA score. This task-level variation remains visible despite the higher aggregate score.

Table 2: **Structured pruning on LLaVA-1.5-13B.** Average is the mean benchmark score relative to the corresponding unpruned model, expressed as a percentage. Bold and underlined values indicate the best and second-best pruned results at each pruning ratio, respectively.

	Ratio	Method	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
LLaVA-1.5-13B	0%	Vanilla	61.85	66.07	54.81	1664.98	85.53	71.44	74.36	52.92	59.20	100
	20%	Wanda-sp	58.06	61.17	<u>50.85</u>	1469.49	78.97	<u>66.93</u>	<u>69.02</u>	<u>48.32</u>	57.77	92.80
		FLAP	<u>59.04</u>	**63.14**	50.43	<u>1539.38</u>	<u>82.97</u>	66.14	68.64	47.08	<u>58.41</u>	<u>93.89</u>
		SlimVLM	**60.23**	**63.14**	**51.91**	**1625.57**	**84.32**	**67.72**	**70.62**	**49.24**	**59.86**	**96.42**
	40%	Wanda-sp	42.92	41.32	24.07	1200.56	<u>78.07</u>	46.60	51.32	33.23	42.67	67.60
		FLAP	<u>50.02</u>	<u>50.46</u>	<u>33.42</u>	<u>1375.61</u>	76.51	<u>60.19</u>	<u>60.43</u>	<u>38.55</u>	<u>55.30</u>	<u>80.23</u>
		SlimVLM	**54.65**	**53.87**	**34.89**	**1442.90**	**83.72**	**63.21**	**62.86**	**38.93**	**57.61**	**84.67**

Table 3: SlimVLM pruning results on Qwen2.5-VL-7B, with the maximum pixel count set to $1280 \times 28 \times 28$. Average is normalized by the unpruned model.

Ratio	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
0%	58.28	80.67	62.16	2218.06	87.82	80.61	76.42	76.80	66.86	100
20%	54.47	75.23	55.32	2033.76	86.78	73.34	72.48	69.55	64.27	93.19
40%	43.86	61.78	43.96	1726.55	84.37	64.32	63.79	57.18	42.68	77.56

Performance at different pruning ratios. Figure 3 extends the pruning-ratio comparison on POPE, VizWiz, and ScienceQA. FLAP and SlimVLM have similar ScienceQA scores through 40% pruning, with a larger gap at 50%. At 50% pruning, Wanda-sp fails to answer questions in this evaluation. The curves show how each method's performance changes as more structure is removed.

Higher-resolution inputs. On LLaVA-NeXT-7B, SlimVLM retains 93.50% of the unpruned model's average relative performance at 20% pruning, exceeding FLAP and Wanda-sp by 3.55 and 9.04 percentage points, respectively. At 40% pruning, SlimVLM retains 74.59%, compared with 61.51% for FLAP. It leads on seven of the nine task settings; FLAP scores higher on GQA and VizWiz. These results extend the observed gains to a model that processes up to 2880 visual tokens.

Scaling to larger models. We evaluate SlimVLM on LLaVA-1.5-13B using the experimental setup described above. As shown in Table 2, SlimVLM achieves the best or joint-best performance across all nine metrics among the evaluated pruning methods at 20% pruning. It matches FLAP on MMBench and outperforms both baselines on the remaining eight metrics. On VizWiz, SlimVLM achieves 59.86, compared with 58.41 for FLAP and 57.77 for Wanda-sp; on MME, the corresponding scores are 1625.57, 1539.38, and 1469.49.

At 20% pruning, SlimVLM retains 96.42% average relative performance, corresponding to a reduction of 3.58 percentage points from the unpruned model. It exceeds FLAP and Wanda-sp by 2.53 and 3.62 percentage points, respectively. At 40% pruning, SlimVLM retains 84.67%, compared with 80.23% for FLAP and 67.60% for Wanda-sp. The widening performance gaps show that SlimVLM's advantage persists and increases under more aggressive pruning.

Grouped-query attention. We further evaluate SlimVLM on Qwen2.5-VL-7B to examine its applicability to grouped-query attention. SlimVLM prunes query heads while allowing different numbers of heads to be retained across key–value groups. As reported in Table 3, it retains 93.19% and 77.56% of the unpruned model's average relative performance at 20% and 40% pruning, respectively. These results extend the evaluation from the MHA-based LLaVA models to the GQA-based Qwen2.5-VL architecture.

4.3 Ablation Studies

We ablate visual token selection and dynamic pruning ratios on LLaVA-1.5-7B. Unless otherwise specified, these experiments use a 20% pruning ratio.

Adaptive visual token selection. Table 4 compares calibration with all visual tokens (None), a random subset matched to the text-token count (Random), and adaptive selection (Ours). For

Table 4: Effect of visual token selection during importance estimation. None retains all visual tokens; Random samples as many visual tokens as text tokens; Fixed retains the top-k visual tokens, with k equal to the text-token count; Ours uses the adaptive attention threshold.

Method	Selection Strategy	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
FLAP	None	46.64	46.99	28.44	1174.77	76.68	48.69	53.90	36.11	52.88	76.26
	Random	52.94	50.10	37.68	1286.12	82.46	57.30	52.75	38.48	46.77	82.00
	Ours	55.45	52.58	41.24	1645.92	84.59	59.89	64.32	40.59	48.72	89.37
SlimVLM	None	47.09	35.57	22.08	1065.51	83.97	52.70	57.06	37.01	46.80	73.30
	Random	58.17	55.50	42.61	1488.59	84.67	59.25	65.76	41.15	46.45	89.51
	Fixed	57.52	56.01	43.99	1683.80	85.28	60.93	65.70	41.01	47.92	91.63
	Ours	58.00	56.44	45.88	1682.96	85.49	61.33	65.90	41.56	54.55	93.80

Table 5: Structured pruning with different visual token selections for module importance estimation. Each strategy retains 20% of visual tokens from the bottom, middle, or top of the attention-score ranking.

Model	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
Vanilla	61.94	62.02	50.26	1780.64	85.89	65.99	69.56	48.07	54.48	100.00
Bottom 20%	54.79	49.91	40.77	1530.37	84.05	55.38	61.25	38.17	42.58	84.82
Middle 20%	56.58	53.69	42.16	1596.78	84.37	58.12	64.06	39.49	50.16	89.34
Top 20%	57.37	55.31	44.08	1658.73	85.53	60.71	65.18	40.90	53.82	92.42

Table 6: Effect of module-specific pruning ratios on LLaVA-1.5-7B at 20% pruning.

Method	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
w/o dynamic pruning ratios	56.77	54.73	42.78	1677.93	85.66	59.99	66.60	40.10	44.89	90.16
w/ dynamic pruning ratios	58.00	56.44	45.88	1682.96	85.49	61.33	65.90	41.56	54.55	93.80

SlimVLM, adaptive selection increases average relative performance from 73.30% with all tokens to 93.80%, and from 89.51% with random selection to 93.80%. For FLAP, adaptive selection improves scores on eight of the nine task settings over using all tokens. Thus, selecting calibration tokens improves the evaluated pruning methods across different structural scoring criteria. SlimVLM also outperforms Fixed selection, which retains the top-k visual tokens with k equal to the text-token count, by 2.17 percentage points. This comparison supports adapting the retained token count to attention scores rather than fixing it to the prompt length.

Effect of attention ranking. We compare visual tokens from the bottom, middle, and top of the attention-score ranking, retaining 20% of visual tokens in each case. As shown in Table 5, selecting the top 20% yields the highest scores across all evaluated benchmarks among the three token-selection strategies, achieving 92.42% average relative performance compared with 89.34% for the middle 20% and 84.82% for the bottom 20%. The 7.60-percentage-point gain over bottom-ranked selection supports using high-attention visual tokens for module importance estimation.

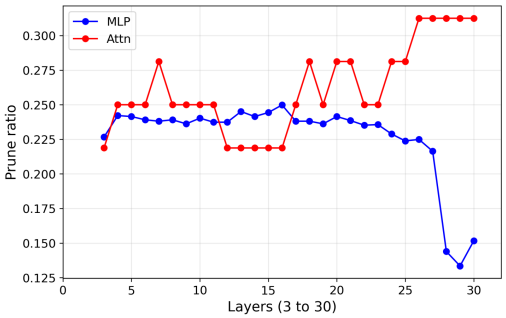

Figure 4: Module-specific pruning ratios for attention heads and MLP channels in LLaVA-1.5 at a 20% pruning setting.

Module-specific pruning ratios. Table 6 isolates the effect of the correlation-based ratio adjustment in Equation 6. Dynamic allocation increases average relative performance from 90.16% to 93.80%, a gain of 3.64 percentage points. The effect varies across tasks: VizWiz increases from 44.89 to 54.55, while POPE and VQA-v2 decrease slightly. Figure 4 shows the resulting allocation across attention and MLP modules. Deeper MLP modules retain more channels, middle-layer attention modules

Table 7: Recovery of pruned LLaVA-1.5-7B with LoRA fine-tuning at rank 32. Average is normalized by the unpruned model.

Ratio	Tune	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
0%	-	61.94	62.02	50.26	1780.64	85.89	65.99	69.56	48.07	54.48	100
20%	×	58.00	56.44	45.88	1682.96	85.49	61.33	65.90	41.56	54.55	93.80
	✓	59.67	58.99	48.26	1712.84	85.45	65.18	67.18	44.38	54.66	96.80
40%	×	46.44	43.57	33.81	1331.00	83.71	52.06	54.10	25.31	44.89	75.16
	✓	51.28	46.63	42.78	1511.74	84.33	58.64	57.19	33.17	48.95	84.01

Table 8: Task scores and pruning memory usage at different calibration sample sizes.

Samples	GQA	MMB	MME	SQA^{I}	Memory (GB)
12	57.88	55.84	1695.25	61.68	17.74
36	56.63	57.13	1578.76	59.59	26.11
24	58.00	56.44	1682.96	61.33	21.14

Table 9: Inference throughput before and after 20% pruning. Speedup is relative to the unpruned model.

Ratio	Method	Tokens/s	Speedup
0%	Vanilla	85.39	1×
20%	Wanda-sp	94.76 (↑ 10.97%)	1.11×
	FLAP	93.33 (↑ 9.30%)	1.09×
	SlimVLM	106.69 (↑ 24.94%)	1.25×

retain more heads, and deeper attention modules receive higher pruning ratios. These allocations reflect the module-specific sensitivity measured by the pruning criterion.

4.4 Recovery with LoRA

We fine-tune pruned LLaVA-1.5-7B using LoRA with rank 32 on the MobileVLM (Chu et al., 2024) dataset. As shown in Table 7, recovery increases average relative performance from 93.80% to 96.80% at 20% pruning and from 75.16% to 84.01% at 40% pruning. The respective gains are 3.00 and 8.85 percentage points, with the larger gain occurring at the higher pruning ratio.

Calibration sample size. Table 8 compares sample sizes of 12, 24, and 36 on GQA, MMBench, MME, and ScienceQA. Increasing the sample count does not consistently improve task scores. For example, 12 samples yield higher MME and ScienceQA scores than 24 samples, while 24 samples perform better on GQA and MMBench. Memory usage rises from 17.74 GB at 12 samples to 21.14 GB at 24 and 26.11 GB at 36. We use the 24-sample setting in the main experiments.

4.5 Inference Speed

Table 9 reports inference throughput before and after 20% pruning. SlimVLM increases throughput from 85.39 to 106.69 tokens/s, a 24.94% gain or approximately $1.25\times$ speedup in the measured setting. Wanda-sp and FLAP reach 94.76 and 93.33 tokens/s, respectively. These measurements show that the SlimVLM-pruned model delivers the highest throughput among the compared configurations.

5 Conclusion

SlimVLM combines text-conditioned visual token selection with sensitivity-aware structured pruning for vision-language models. Selecting calibration tokens focuses importance estimation on visual content attended to by the prompt, while module-specific pruning ratios account for differences in output sensitivity. Experiments show higher performance retention than the evaluated structured pruning baselines on LLaVA-1.5 and LLaVA-NeXT, and establish applicability to Qwen2.5-VL with grouped-query attention. The improvements obtained by applying visual token selection to FLAP further support its usefulness across different structural pruning criteria. On LLaVA-1.5-7B, SlimVLM retains 93.80% of average relative performance at 20% pruning and achieves a measured throughput of 106.69 tokens/s. Together, the results identify calibration-token selection as an effective component of VLM parameter compression.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and language polishing, focusing on improving clarity and readability. We take full responsibility for the final content of this work, including all AI-assisted text. All AI-assisted text was reviewed and revised by the authors.

REFERENCES

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. VQA: Visual question answering, 2015.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models, 2025.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models, 2023.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, et al. Qwen technical report, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, et al. Qwen2.5-VL technical report, 2025.

Akhiad Bercovich, Tomer Ronen, Talor Abramovich, Nir Ailon, Nave Assaf, Mohammad Dabbah, Ido Galil, et al. Puzzle: Distillation-based NAS for inference-optimized LLMs, 2024.

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, 2024.

Yuli Chen, Bo Cheng, Jiale Han, Yingying Zhang, Yingting Li, and Shuhao Zhang. DLP: Dynamic layerwise pruning in large language models, 2025.

Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, et al. MobileVLM V2: Faster and stronger baseline for vision language model, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, et al. MME: A comprehensive evaluation benchmark for multimodal large language models, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering, 2016.

Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. SlimLLM: Accurate structured pruning for large language models, 2025.

Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven C. H. Hoi. From images to textual prompts: Zero-shot VQA with frozen large language models, 2022.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, et al. VizWiz grand challenge: Answering visual questions from blind people, 2018.

Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed, 2024a.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. ZipVL: Efficient large vision-language models with dynamic token sparsification, 2024b.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering, 2019.

Bo Li, Peiyuan Zhang, Kaichen Zhang, Fanyi Pu, Xinrun Du, Yuhao Dong, Haotian Liu, et al. LMMs-Eval: Accelerating the development of large multimodal models, March 2024. Software, version v0.1.0.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-completion neural dialogue systems, 2017.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, June 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, January 2024b.

Jizhihui Liu, Feiyi Du, Guangdao Zhu, Niu Lian, Jun Li, and Bin Chen. HiPrune: Training-free visual token pruning via hierarchical attention in vision-language models, 2025.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model, 2024c.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, et al. MMBench: Is your multi-modal model an all-around player?, 2023.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, et al. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, et al. ShortGPT: Layers in large language models are more redundant than you expect, 2024.

Jorge Mendez-Mendez, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Embodied lifelong learning for task and motion planning, 2023.

Ayan Sengupta, Siddhant Chaudhary, and Tanmoy Chakraborty. You only prune once: Designing calibration-free model compression with policy learning, 2025.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models, 2024.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, et al. Towards VQA models that can read, 2019.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, et al. LLaMA: Open and efficient foundation language models, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, et al. Attention is all you need, 2017.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, et al. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction, 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, 2024.

Senqiao Yang, Junyi Li, Xin Lai, Bei Yu, Hengshuang Zhao, and Jiaya Jia. VisionThink: Smart and efficient vision language model via reinforcement learning, 2025.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models, 2024.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, November 2024. ISSN 2053-714X. doi: 10.1093/nsr/nwae403.

Zhou Yu, Xuecheng Ouyang, Zhenwei Shao, Meng Wang, and Jun Yu. Prophet: Prompting large language models with complementary answer heuristics for knowledge-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6797–6808, August 2025. ISSN 1939-3539. doi: 10.1109/TPAMI.2025.3562422.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI, 2023.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Yaqi Xie, et al. VScan: Rethinking visual token reduction for efficient large vision-language models, 2025.

6 APPENDIX

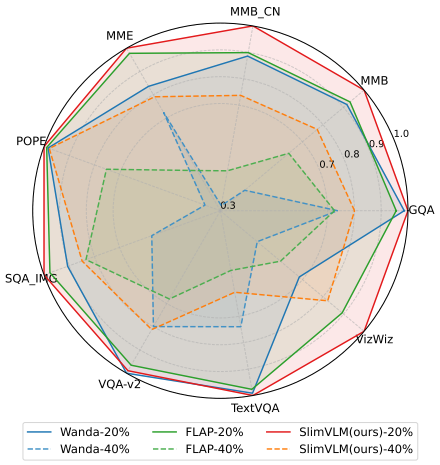

Figure 5: Benchmark performance of SlimVLM, Wanda-sp, and FLAP at different pruning ratios.

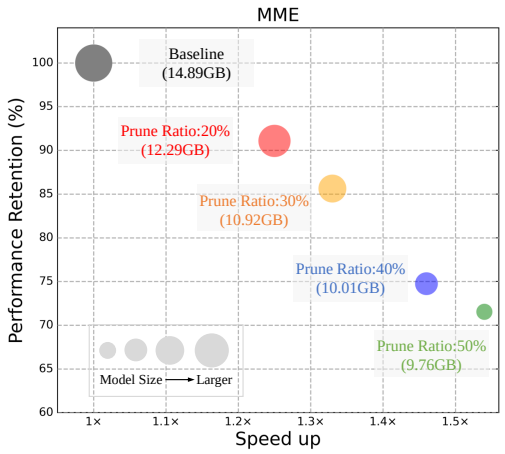

Figure 6: MME score, inference speed, and GPU memory usage of SlimVLM at different pruning ratios.

A BENCHMARK DATASETS

We evaluate visual understanding, reasoning, text recognition, and object hallucination using the benchmarks below. Table 10 lists the evaluation metrics and system prompts.

GQA (Hudson & Manning, 2019). GQA evaluates visual scene understanding and reasoning using images, scene graphs, and questions. Its image annotations include spatial and object features.

MMBench (Liu et al., 2023). MMBench evaluates perception and reasoning through a three-level hierarchy: two core abilities at the first level, six sub-abilities at the second, and twenty ability dimensions at the third. We report results for both the English and Chinese versions.

MME (Fu et al., 2023). MME evaluates perceptual and cognitive abilities across 14 subtasks using manually curated instruction–answer pairs.

POPE (Li et al., 2023). POPE evaluates object hallucination through binary questions about whether objects are present in an image. It measures accuracy, recall, precision, and F1 under three sampling strategies.

ScienceQA (Lu et al., 2022). ScienceQA covers natural science, language science, and social science. Its questions span 26 topics, 127 categories, and 379 skills, supporting evaluation of multimodal understanding and multi-step reasoning.

VQA-v2 (Goyal et al., 2016). VQA-v2 evaluates visual perception through open-ended questions about 265,016 real-world images. Each question has ten human-annotated reference answers.

TextVQA (Singh et al., 2019). TextVQA evaluates question answering that requires reading text in images and reasoning about that text in its visual context.

VizWiz (Gurari et al., 2018). VizWiz contains over 31,000 visual questions collected from blind users, who photographed scenes with mobile phones and recorded spoken questions. Each question has ten crowdsourced answers.

Table 10: Benchmarks, evaluation metrics, and system prompts.

Benchmark	Metric	System Prompt
GQA (Hudson & Manning, 2019)	exact match	Answer the question using a single word or phrase.
MMBench (Liu et al., 2023)	GPT evaluation score	Answer with the option's letter from the given choices directly.
MME (Fu et al., 2023)	perception score, cognition score	Answer the question using a single word or phrase.
POPE (Li et al., 2023)	accuracy	Answer the question using a single word or phrase.
ScienceQA (Lu et al., 2022)	exact match	Answer with the option's letter from the given choices directly.
VQA-v2 (Goyal et al., 2016)	exact match	Answer the question using a single word or phrase.
TextVQA (Singh et al., 2019)	exact match	Answer the question using a single word or phrase.
VizWiz (Gurari et al., 2018)	exact match	Answer the question using a single word or phrase.

Table 11: **Structured pruning on LLaVA-1.5-7B at a 20% pruning ratio.** Average is the mean benchmark score relative to the corresponding unpruned model, expressed as a percentage. Bold and underlined values indicate the best and second-best pruned results, respectively.

Method	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
Vanilla	61.94	62.02	50.26	1780.64	85.89	65.99	69.56	48.07	54.48	100.00
Wanda-sp	57.19	51.72	40.64	1406.53	84.12	55.58	**66.62**	<u>41.20</u>	37.34	85.31
FLAP	55.45	52.58	41.24	<u>1645.92</u>	84.59	<u>59.89</u>	64.32	40.59	48.72	89.37
LLM-Pruner	56.14	50.37	41.11	1562.36	84.33	57.67	63.47	41.08	42.57	86.87
LoRAP	56.71	52.34	41.89	1588.37	84.70	58.15	63.18	40.17	44.36	87.89
Puzzle	<u>57.34</u>	<u>55.03</u>	<u>43.17</u>	1569.41	84.44	59.30	63.54	39.47	46.61	89.17
SlimLLM	57.21	54.78	42.01	1611.38	<u>84.72</u>	59.17	64.57	40.88	<u>51.07</u>	<u>90.52</u>
SlimVLM	**58.00**	**56.44**	**45.88**	**1682.96**	**85.49**	**61.33**	<u>65.90</u>	**41.56**	**54.55**	**93.80**

B Extended Experimental Results

B.1 Performance Comparison

Figure 5 compares SlimVLM, Wanda-sp, and FLAP across benchmarks and pruning ratios.

Figure 7 shows performance as a function of the remaining parameter count on six LLaVA-1.5 tasks: GQA, MMBench, MMBench (CN), MME, VQA-v2, and TextVQA. The curves summarize how performance changes with compression for the three methods.

B.2 Comparison with Additional Pruning Baselines

Table 11 compares SlimVLM with LLM-Pruner, LoRAP, Puzzle, and SlimLLM, alongside Wanda-sp and FLAP. At a 20% pruning ratio on LLaVA-1.5-7B, SlimVLM achieves the highest average relative performance among the compared pruning methods. Its average relative score of 93.80% exceeds that of SlimLLM by 3.28 percentage points. The comparison establishes the performance of the combined framework against additional structured pruning methods.

B.3 Sensitivity Analysis

Sensitivity to γ and T. Table 12 varies the sensitivity coefficient γ and correlation reference T in Equation 6. Using $\gamma = 5$ and $T = 1$ as the reference, the reported average relative scores range from 97.34% to 100% across the tested configurations. The default setting yields the highest aggregate score within this sweep.

B.4 Threshold for Token Selection

Adaptive visual token selection retains visual token i when its average attention score p_i satisfies $p_i \geq \alpha = \text{mean}(p) \times \beta$. Table 13 evaluates $\beta \in \{0.05, 0.1, 0.2, 0.3, 0.5\}$. A larger β filters more visual tokens before module importance estimation. Among the tested values, $\beta = 0.2$ yields the highest average relative performance; both lower and higher thresholds reduce this average.

Table 12: Sensitivity to γ and T in Equation 6. Average is the mean ratio of each task score to that of the default pruned configuration ($\gamma = 5$, $T = 1$), expressed as a percentage. The individual task columns report raw scores.

γ	T	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
1	1	56.32	55.21	44.24	1619.79	85.46	57.24	64.46	41.29	53.45	97.34
3	1	56.40	55.84	44.59	1623.69	85.43	58.29	65.15	41.48	53.57	97.97
5	1	58.00	56.44	45.88	1682.96	85.49	61.33	65.90	41.56	54.55	100
5	0.95	57.38	56.07	45.18	1646.96	85.43	59.79	64.62	41.23	54.07	98.71
5	0.9	57.19	55.87	44.73	1595.14	85.44	59.38	64.98	40.99	53.99	98.09

Table 13: Effect of the visual token selection threshold coefficient β on post-pruning performance.

β	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
0.05	46.00	52.34	38.62	1060.14	82.41	55.38	56.02	36.90	50.97	80.64
0.1	53.41	51.72	36.68	1474.75	85.84	60.78	61.72	38.21	46.93	85.76
0.2	58.00	56.44	45.88	1682.96	85.49	61.33	65.90	41.56	54.55	93.80
0.3	57.93	54.98	42.44	1526.02	85.23	59.94	66.12	39.98	48.39	89.94
0.5	57.94	54.81	41.24	1483.12	84.69	59.84	64.26	38.10	41.14	87.08

B.5 Comparison with VLM Token Compression Methods

Visual token compression reduces the number of tokens processed during inference, while SlimVLM reduces model parameters through structured pruning. Table 14 compares their benchmark performance. SlimVLM retains 93.80% average relative performance, compared with 93.46% for FastV and 95.22% for PyramidDrop (PDrop) (Xing et al., 2024). LoRA-based recovery raises SlimVLM to 96.80%, the highest average among the compressed configurations in the table.

Table 14: Benchmark performance of SlimVLM and visual token compression methods on LLaVA-1.5-7B. † indicates LoRA-based recovery.

Method	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Average(%)
Vanilla	61.94	62.02	50.26	1780.64	85.89	65.99	69.56	48.07	54.48	100
FastV	52.70	58.18	49.05	1560.68	80.70	63.88	65.78	43.12	55.54	93.46
PDrop	57.38	57.14	47.17	1691.47	82.30	64.84	65.77	45.66	54.33	95.22
SlimVLM	58.00	56.44	45.88	1682.96	85.49	61.33	65.90	41.56	54.55	93.80
SlimVLM†	59.67	58.99	48.26	1712.84	85.45	65.18	67.18	44.38	54.66	96.80

B.6 Combining SlimVLM with Token Pruning Methods

Table 15 evaluates FastV on the SlimVLM-pruned model before and after recovery. This combination applies token compression to a model whose parameter count has already been reduced. The per-benchmark results show the performance changes introduced by enabling FastV in each configuration.

Table 15: Benchmark performance of SlimVLM-pruned LLaVA-1.5-7B with FastV enabled or disabled, before and after LoRA recovery. Drop denotes the mean relative decrease across the nine task settings caused by enabling FastV, using the corresponding model without FastV as the reference within each recovery setting.

	FastV	GQA	MMB	MMB^{CN}	MME	POPE	SQA^{I}	VQA^{2}	VQA^{T}	VizWiz	Drop
w/o LoRA recovery	×	58.00	56.44	45.88	1682.96	85.49	61.33	65.90	41.56	54.55	-
	✓	57.62	55.67	44.84	1706.18	85.18	61.08	65.68	42.11	51.11	↓ 0.99%
w/ LoRA recovery	×	59.67	58.99	48.26	1712.84	85.45	65.18	67.18	44.38	54.66	-
	✓	58.62	57.81	47.11	1708.18	85.18	64.04	66.68	43.17	53.73	↓ 1.52%

B.7 Inference Efficiency

Figure 6 summarizes the relationship between MME score, inference speed, and GPU memory consumption for LLaVA-1.5-7B at different SlimVLM pruning ratios.

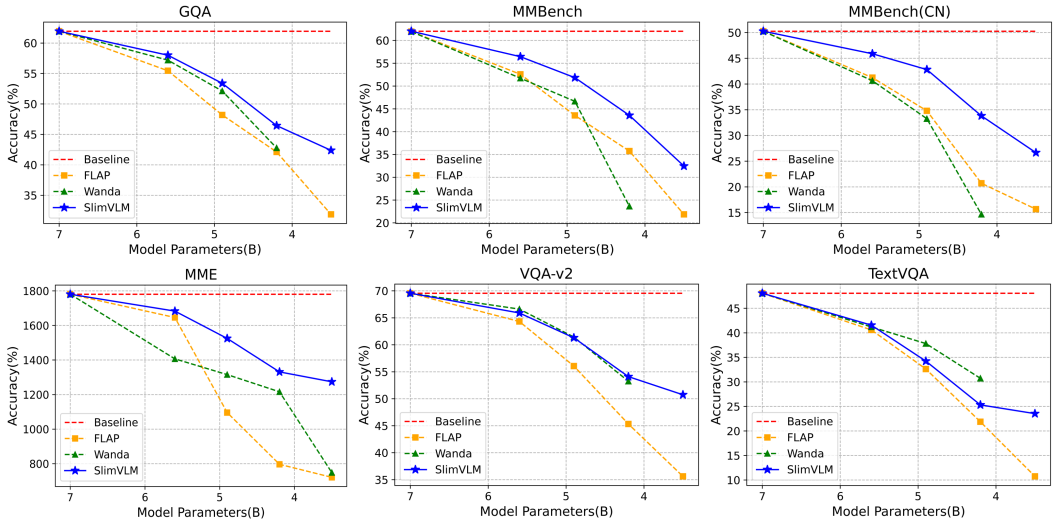

Figure 7: Performance of SlimVLM, Wanda-sp, and FLAP on six LLaVA-1.5 task settings as the remaining parameter count changes. The horizontal axis shows model parameters; the vertical axis shows the benchmark score.

C Visualization of Adaptive Visual Token Selection

Figure 8 illustrates the visual tokens retained at three depths for an MMBench example. Many discarded tokens correspond to background regions, and the retained set becomes more concentrated on target-relevant visual content in deeper layers.

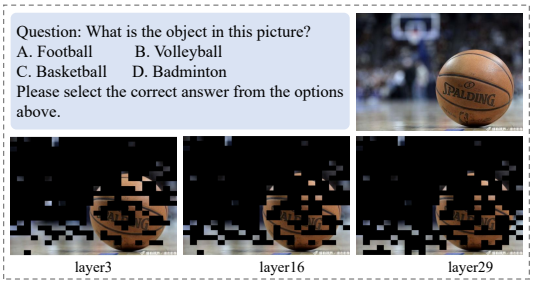

Figure 8: Visual tokens retained at layers 3, 16, and 29 for an MMBench example.

D Limitations

SlimVLM estimates module importance and allocates pruning ratios using a small calibration set. The reliability of these estimates depends on how well the calibration examples represent the target image and language distributions, particularly for specialized multimodal tasks. Our largest evaluated model is LLaVA-1.5-13B. Evaluation on larger models, video-language models, and additional deployment settings remains open.

E QUALITATIVE ANSWER EXAMPLES

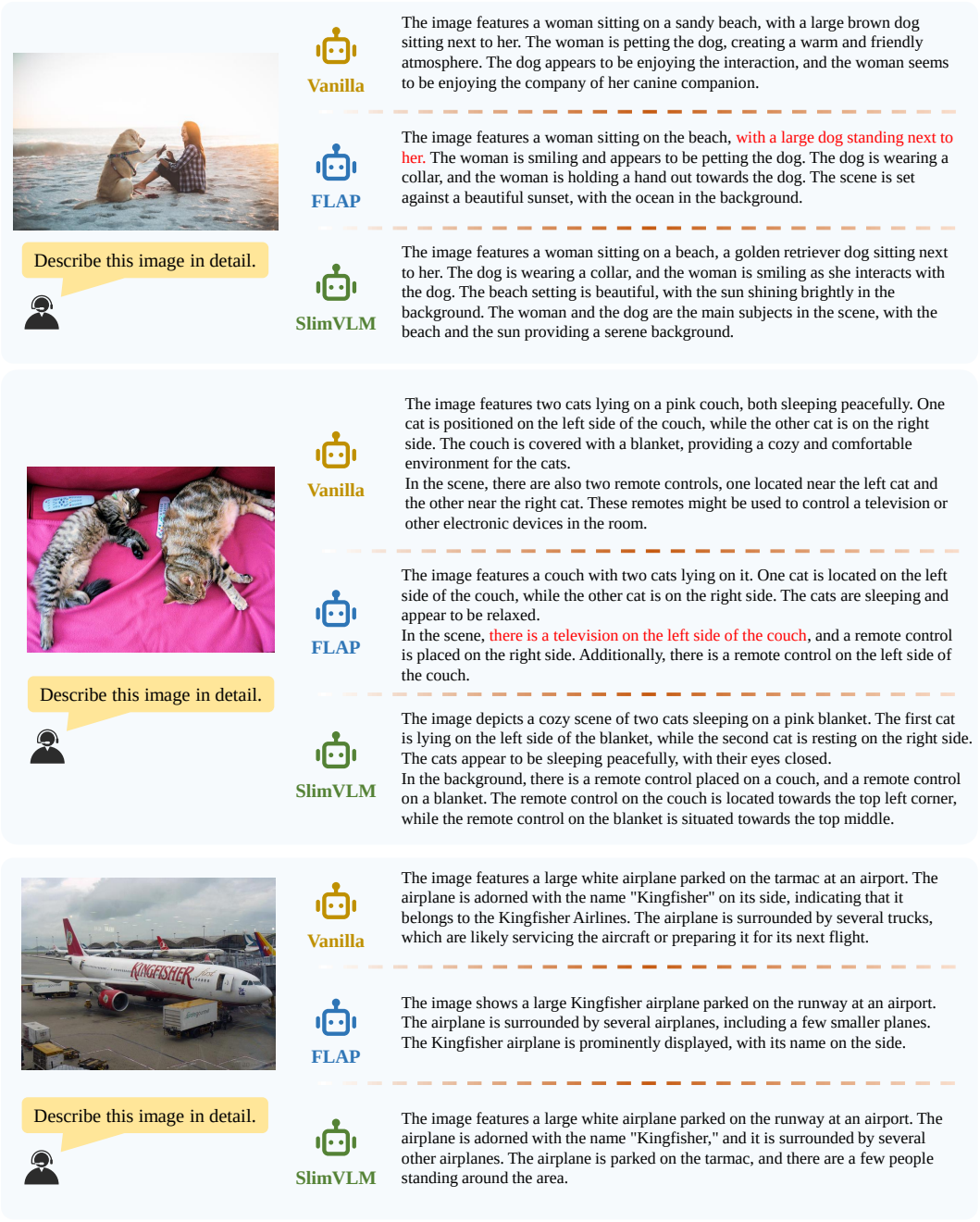

Figure 9: Visual question-answering examples from LLaVA-1.5-7B at a 20% pruning ratio.