P4Q: CO-DESIGNING TOKEN PRUNING AND QUANTIZATION FOR VISION-LANGUAGE MODEL ACCELERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision language models have achieved strong performance across a wide range of multimodal applications, yet their substantial computational and memory costs hinder efficient deployment. Visual token pruning and post-training quantization reduce inference overhead along two complementary dimensions, namely sequence length and numerical precision. Existing workflows typically optimize these techniques independently or apply them sequentially. Their distinct optimization objectives leave critical interactions unaddressed and constrain the achievable compression performance. We revisit these designs and present P4Q, a practical co-design framework that jointly optimizes visual token pruning and low-bit quantization for efficient VLM inference. First, P4Q introduces a quantization-aware visual token selection strategy before the LLM. It applies fake quantization to copies of the features produced by the projector and selects visual tokens using statistics computed from these fake-quantized features, thereby conditioning the selector's feature-based decisions on simulated low-bit perturbations. Second, P4Q introduces a pruning-aware quantization calibration strategy. It uses the same selection strategy as pruning to calibrate the quantized model on the retained-token distribution, thereby aligning the calibration process with the pruned execution path used during deployment. By coupling these two components, P4Q achieves substantial inference speedups while maintaining comparable task performance, resulting in a better efficiency-accuracy trade-off than independently optimized pipelines. For instance, on LLaVA-NeXT, P4Q achieves an average end-to-end inference speedup of 2.8$\times$ across eight distinct test sets, while retaining higher accuracy than prior compression and quantization methods.

1 INTRODUCTION

Modern VLMs commonly encode each image into hundreds or thousands of visual tokens, although these representations exhibit substantial redundancy (Chen et al., 2024; Alvar et al., 2025; Shang et al., 2025). This inefficiency has motivated two complementary compression paradigms: post-training quantization (PTQ) and visual token pruning. PTQ reduces the numerical precision of model weights and activations, lowering memory and computation costs through calibrated low-bit representations (Sun et al., 2025; Li et al., 2025; Wang et al., 2024). Visual token pruning instead shortens the multimodal sequence by removing redundant or less informative image tokens before or within the language model. Applying both should reduce the cost of each operation and the number of tokens on which it operates. The difficulty is that the two tools also change the conditions under which the other was designed.

Current visual token pruning methods typically identify and remove redundant or weakly informative visual tokens before or within the language model to reduce subsequent computation (Chen et al., 2024; Huang et al., 2024; Alvar et al., 2025). However, recent studies show that attention-based importance estimates can be unstable, poorly aligned with actual token utility, and spatially biased (Wen et al., 2025; Zhang et al., 2025; Endo et al., 2025). This weakness becomes particularly consequential near the top-K boundary: when two tokens have a small score margin, a perturbation exceeding that margin can reverse their order and discontinuously change the retained set. We further observe that low-bit quantization introduces an additional source of such perturbations, as errors

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.

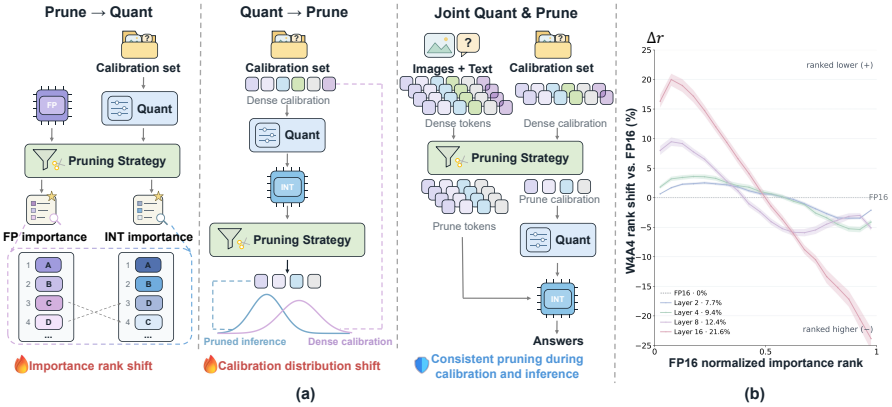


Figure 1: **Comparison of sequential and joint pipelines for low-bit calibration and visual-token pruning, together with a visualization of the visual-token importance ranking shifts induced by low-bit quantization across LLM layers.** **(a)** Prune-then-quantize suffers from quantization-induced importance-rank shifts, whereas quantize-then-prune creates a calibration-inference distribution shift. P4Q jointly prunes inference and calibration sequences using the same selector and token budget, matching their retained-token support and sequence length. **(b)** W4A4 induces layer-dependent visual token rank shifts relative to FP16 across representative in-LLM pruning layers in LLaVA-NeXT-7B.

accumulated in hidden states and linear transformations propagate to the attention scores used for selection. Consequently, under low-bit execution, attention-based importance estimation inside the language model becomes more unstable, making the retained visual-token set increasingly sensitive to quantization-induced perturbations.

A parallel mismatch arises during quantization calibration. Calibration-based PTQ fits low-bit parameters to the activation distributions induced by the calibration data. Specifically, calibration samples generate layer-wise activations from which quantization scales, clipping ranges, and transformation parameters are estimated. These parameters are therefore specialized to the activation regime observed during calibration. Prior studies show that calibration-data selection can substantially affect the downstream performance of compressed LLMs (Williams & Aletras, 2024). A mismatch between calibration and inference sequence lengths can further degrade quantized-model accuracy (Lee et al., 2023). VLM-specific studies further demonstrate the importance of calibration design at low bit widths (Wang et al., 2024; Li et al., 2025). Q-VLM optimizes layer-wise rounding while accounting for cross-layer dependencies, whereas MBQ employs modality-balanced reconstruction to address unequal vision language sensitivity. This dependence creates a systematic mismatch when a densely calibrated VLM is subsequently combined with visual-token pruning. During deployment, each sample presents the low-bit language model with a shorter, sample-dependent sequence than that used during calibration. Pruning alters the retained visual-token support, attention context, sequence length, and downstream activation statistics. Dense calibration therefore optimizes quantization parameters for a computational graph and activation distribution that differ from those encountered during pruned inference.

Taken together, independently optimizing or sequentially applying visual token pruning and quantization creates two coupled mismatches. Quantization perturbations can distort visual-token selection, while dense-sequence calibration fits the quantizer to an execution path different from that encountered during pruned inference.

To study these interactions, we introduce P4Q, a practical co-design framework that jointly optimizes visual token pruning and low-bit quantization for efficient VLM inference. First, P4Q introduces a quantization-aware visual token selection strategy before the LLM. The selector determines token indices from fake-quantized copies of projector features without using quantized decoder states for scoring. Second, P4Q introduces a pruning-aware quantization calibration strategy. For each calibration sample, it uses the inference-time selector and token budget to construct the retained-token sequence on which the quantizer is fitted. By coupling these two components,

P4Q achieves substantial inference speedups while preserving strong task performance, providing a better efficiency-accuracy trade-off than independently optimized pipelines. Our main contributions are summarized as follows.

- We rethink visual token pruning and quantization as a coupled compression problem and introduce P4Q, a practical co-design framework for efficient VLM inference. P4Q employs a quantization-aware pre-LLM selector that uses fake-quantized copies of projector features in its feature-based scoring stages, allowing token selection to account for simulated low-bit perturbations. It further performs pruning-aware PTQ calibration on sequences produced by the same selector and token budget used at inference, exposing the quantizer to the resulting retained-token support and sequence length.
- We validate P4Q across eight multimodal benchmarks under W4A4 quantization and an 88.9% visual token pruning ratio. On LLaVA-NeXT-7B, P4Q achieves an average $2.8\times$ end-to-end inference speedup while reducing peak GPU memory usage to 51.3% of the vanilla baseline. Under the same setting, P4Q achieves an average $2.49\times$ end-to-end speedup on LLaVA-1.5-7B and reduces peak GPU memory usage by 58% relative to the vanilla baseline.

2 Related Work

Visual Token Pruning for VLMs. VLMs typically encode an image as a long sequence of patch tokens, many of which are redundant for a given input and task. A prominent family of pruning methods determines which tokens to retain inside the language model, using cross-modal attention or intermediate representations as task-aware signals. FastV, for example, ranks visual tokens using text-to-vision attention from a shallow decoder layer and removes weakly attended tokens from subsequent computation (Chen et al., 2024). Although such signals reflect explicit vision language interaction, they are also conditioned on the decoder execution path and the layer at which they are measured. This conditionality may make the retained set sensitive to the decoder-side scoring process. Wen et al. (2025) systematically show that attention-based importance does not always identify useful visual tokens reliably and that carefully designed pruning methods can even underperform random selection. FEATHER further demonstrates that early decoder attention can inherit RoPE-induced positional bias, producing spatially skewed retained sets and pronounced degradation on localization-intensive tasks (Endo et al., 2025). Together, these findings suggest that decoder-side importance is not an intrinsic property of a visual token, but a conditional estimate shaped by attention geometry, pruning depth, input characteristics, and model state.

Against this backdrop, a growing line of work performs visual-token selection outside the language model, deriving retention signals from vision-side representations or lightweight pre-LLM interactions. LLaVA-PruMerge uses sparse class-to-patch attention from the vision encoder to identify visual anchors, then clusters and merges the remaining tokens to supplement the selected content before it reaches the LLM (Shang et al., 2025). DivPrune removes reliance on decoder attention by formulating pre-LLM selection as a max–min diversity problem, thereby preserving broad coverage of the visual feature space (Alvar et al., 2025). CRISP restores instruction dependence before decoder execution by grounding question nouns in projected visual features and completing the retained set with diversity-based contextual tokens (Li et al., 2026b). Collectively, these methods construct retained sets from vision-side saliency, feature coverage, or lightweight task cues rather than relying on a single decoder-side importance signal.

Post-Training Quantization for VLMs. PTQ enables low-bit deployment by reducing the precision of model weights and/or activations without full-model retraining, thereby lowering storage and memory-traffic costs. Unlike text-only LLMs, however, VLMs couple a vision encoder and projector with a language backbone, while visual and textual tokens exhibit different activation statistics and sensitivities to quantization error. Q-VLM addresses cross-layer error dependencies by using activation entropy to partition correlated blocks and search their rounding strategies jointly (Wang et al., 2024). MBQ instead accounts for modality-specific sensitivity by reweighting the reconstruction loss for visual and textual tokens (Li et al., 2025). These methods establish that VLM quantization requires calibration objectives adapted to multimodal inference rather than homogeneous, layer-isolated reconstruction.

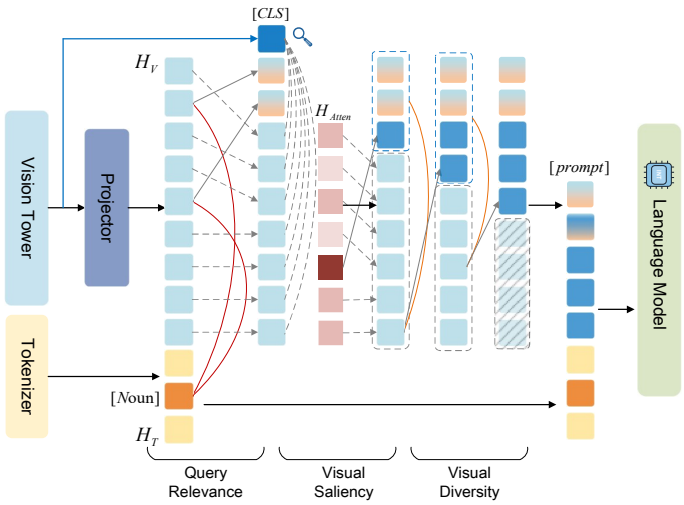


Figure 2: Overview of quantization-aware pruning in P4Q. It first retains query-relevant tokens to capture task semantics, then selects visually salient tokens based on vision-tower CLS-to-patch attention, and finally applies max-min diversity sampling to reduce redundancy and broaden visual coverage. The query relevance and visual diversity stages use fake-quantized feature copies, while visual saliency uses vision tower attention.

Recent work further refines the granularity and execution fidelity of calibration. QIG uses quantization-aware integrated gradients to estimate token-level sensitivity and capture inter- and intra-modality interactions after visual and textual representations have been mixed (Xiang et al., 2026). VLM-PTQ corrects residual-induced target shifts and combines modality-specific Hessian estimates to distinguish channel importance across visual and textual tokens (Deng & Huang, 2026). TLQ constructs a token-importance-aware calibration set and exposes each layer-wise calibration stage to the preceding quantized path, bringing calibration closer to the actual low-bit forward pass (Shang et al., 2026). Collectively, these methods show that calibration data, error weighting, optimization targets, and forward-path simulation are integral components of VLM quantization design.

Joint Quantization and Visual Token Pruning. Quantization and visual token pruning reduce complementary inference costs by lowering numerical precision and shortening the sequence processed by the language backbone, respectively. Joint methods therefore coordinate token selection with the low-bit regime used at deployment. In the vision language action setting, SQAP-VLA applies quantization-aware pruning criteria to an aggressively quantized model and adapts the quantizer to improve pruning effectiveness (Fang et al., 2025). For multimodal language models, QAPruner scores tokens using semantic relevance, simulated group-wise quantization error, and activation-outlier intensity, thereby accounting for both task relevance and low-bit numerical stability (Wang et al., 2026). QUOTA instead aligns calibration with pruned execution by converting low-bit calibration signals into an offline layer-wise token-allocation schedule and evaluating token importance under deployed W4A4 operators and a quantized KV cache (Li et al., 2026a). Together, these methods couple compression through quantizer adaptation, quantization-aware token scoring, or calibration-derived pruning schedules, marking a shift from independently applied compressors toward deployment-aware joint design.

3 METHOD

In this section, we first motivate our method by examining how the serial composition of visual-token pruning and PTQ creates token-ranking and calibration-distribution mismatches. We then introduce P4Q, detailing how to select complementary visual evidence before the low-bit language backbone and calibrate the quantized model on sequences pruned using the inference-time selector and token budget.

3.1 MOTIVATION

Low-bit visual token ranking shift. Existing in-LLM pruning methods commonly estimate token importance from language-model attention or hidden activations, rank the visual tokens accordingly, and retain the Top-K tokens. Let $\mathbf{s}_\ell^{\mathrm{FP}}$ denote the importance scores obtained from the FP16 states at layer ℓ. When the same estimator is applied under low-bit execution, it instead receives quantized hidden states and attention statistics, yielding

$$\mathbf{s}_\ell^{\mathrm{INT}} = \Psi\big(H_\ell^{\mathrm{INT}}, A_\ell^{\mathrm{INT}}\big) = \Psi\big(H_\ell^{\mathrm{FP}}, A_\ell^{\mathrm{FP}}\big) + \boldsymbol{\delta}_\ell, \qquad \mathcal{S}_{\ell,K}^{\mathrm{INT}} = \mathrm{TopK}\big(\mathbf{s}_\ell^{\mathrm{FP}} + \boldsymbol{\delta}_\ell, K\big), \tag{1}$$

where Ψ is the importance estimator, and H_ℓ and A_ℓ denote the hidden states and attention statistics available at layer ℓ, respectively. The perturbation $\boldsymbol{\delta}_\ell$ captures the token-dependent changes in importance scores induced by low-bit computation. The corresponding FP16 selection is $\mathcal{S}_{\ell,K}^{\mathrm{FP}} = \mathrm{TopK}(\mathbf{s}_\ell^{\mathrm{FP}}, K)$. Because Top-$K$ selection depends on relative ordering rather than absolute score magnitude, even moderate perturbations can reverse the order of tokens with similar FP16 scores, particularly near the selection boundary. Consequently, $\mathcal{S}_{\ell,K}^{\mathrm{INT}}$ may differ from $\mathcal{S}_{\ell,K}^{\mathrm{FP}}$, producing a quantization-induced shift in the retained visual-token support.

To visualize quantization-induced ranking drift, we align each visual token between the FP16 and W4A4 runs using its original packed index and define its signed normalized rank displacement as:

$$\Delta r_\ell^{\mathrm{INT}\leftarrow\mathrm{FP16}} = r_\ell^{\mathrm{INT}} - r_\ell^{\mathrm{FP16}} = \frac{\mathrm{rank}_\downarrow\big(s_\ell^{\mathrm{INT}}\big) - \mathrm{rank}_\downarrow\big(s_\ell^{\mathrm{FP16}}\big)}{N-1}, \tag{2}$$

where s_ℓ denotes the importance score of a visual token at layer ℓ, and $\mathrm{rank}_\downarrow$ assigns rank one to the largest score. Accordingly, $\Delta r > 0$ indicates that a token moves toward a less important position under W4A4, whereas $\Delta r < 0$ indicates a move toward a more important position. We evaluate this displacement on 128 fixed samples using identical dense inputs, attention masks, positional indices, and packed token indices. In Fig. 1 (b), tokens are grouped by their same-layer FP16 normalized importance rank, from the most important (0) to the least important (1), and each curve reports the sample-balanced mean displacement at a representative in-LLM pruning layer of LLaVA-NeXT-7B. W4A4 quantization induces substantial rank rearrangement at representative in-LLM pruning layers, indicating that token importance rankings derived under FP16 precision are not necessarily preserved within a low-bit language backbone. More details are provided in Appendix A.

Quantization-aware selection requires low-bit statistics. Recent studies have begun to show that pruning strategies intended for low-bit deployment should account for the numerical regime in which their selection statistics are produced. In particular, quantization can distort the attention or feature statistics used for token ranking, making selection criteria derived exclusively from full-precision states unreliable when directly applied to quantized models (Fang et al., 2025). Quantization-aware pruning methods consequently evaluate token importance using quantized-model statistics or explicitly incorporate simulated quantization errors into the selection criterion (Wang et al., 2026). These observations motivate us to test whether pruning under low-bit inference benefits from token scoring that accounts for simulated perturbations of the target quantization configuration.

Pruning-induced calibration mismatch. Calibration-based post-training quantization is inherently dependent on the activation distribution induced by the calibration data. Calibration samples are forwarded through the model to generate layer-wise activations, from which the quantization scales, clipping ranges, and transformation parameters are estimated. Consequently, these parameters are optimized for the activation regime observed during calibration rather than being independent of the calibration inputs. Prior studies have shown that the choice of calibration data can cause substantial variation in the downstream performance of compressed LLMs (Williams & Aletras, 2024). In particular, a mismatch between the sequence lengths used for calibration and inference can degrade the accuracy of the quantized model, highlighting the importance of aligning the calibration sequence length with the deployment-time inference sequence length (Lee et al., 2023).

Specifically, this dependence can be formalized through the calibration objective. Let $H_{\ell,n}^{\mathrm{dense}}$ denote the input activation of language-model block ℓ for calibration sample n when all visual tokens are retained. Under conventional dense calibration, the quantization parameters are obtained by minimizing the reconstruction error between the full-precision block and its quantized counterpart:

$$\theta_{\ell,\mathrm{dense}}^* = \arg\min_{\theta_\ell} \frac{1}{B}\sum_{n=1}^{B} \mathrm{MSE}\left(f_\ell^{\mathrm{INT}}\left(H_{\ell,n}^{\mathrm{dense}}\right), f_\ell^{\mathrm{FP}}\left(H_{\ell,n}^{\mathrm{dense}}\right)\right), \tag{3}$$

where B is the number of calibration samples, f_ℓ^{FP} and f_ℓ^{INT} denote the full-precision block and its low-bit counterpart, respectively, and θ_ℓ contains the quantization parameters optimized during calibration. Therefore, $\theta^*_{\ell,\mathrm{dense}}$ is fitted to the activation distribution induced by dense visual sequences. After visual token pruning, however, the same block receives $H_{\ell,n}^{\mathrm{pruned}}$, whose sequence length and visual-token support are determined by the retained tokens, thereby altering the attention support, the activation distributions observed by subsequent blocks, and consequently the quantization parameters θ_ℓ obtained during calibration. These findings from prior studies, together with the preceding theoretical analysis, indicate that effective PTQ calibration should align the calibration-data distribution, sequence structure, and resulting activation ranges with those encountered during quantized inference.

3.2 Quantization-Aware Pre-LLM Selection

Motivated by the low-bit visual-token ranking shift and the need for quantization-consistent selection statistics identified in Section 3.1, P4Q performs quantization-aware visual token selection before the language backbone. This placement avoids relying on progressively perturbed attention values or hidden states produced inside the quantized decoder to determine the retained token set. P4Q applies selector-only fake quantization to the projector features used by its feature-based scoring stages. This exposes those selection decisions to a simulated low-bit perturbation. The fake-quantized projector features are used exclusively to determine the retained token indices, whereas the corresponding original FP16 representations are forwarded to the language backbone for subsequent inference. As illustrated in Fig. 2, P4Q further integrates three complementary priors to enrich the visual evidence preserved by the selected tokens: query-relevant information, intrinsically salient visual content, and complementary visual information that improves the diversity of the retained set.

Specifically, P4Q partitions the total retention budget among three sequential signals using the budget ratios α, β, and γ. The overall selection is expressed as:

$$\mathcal{S}_{\mathrm{keep}} = \underbrace{\alpha\,\mathrm{TopK}_{i\in\mathcal{S}_{\mathrm{v}}}\left(\mathrm{CosSim}\left(\widetilde{\mathbf{q}}_j,\widetilde{\mathbf{v}}_i\right)\right)}_{\text{Query Relevance}} + \underbrace{\beta\,\mathrm{TopK}_{i\in\mathcal{S}_{\mathrm{v}'}}\left(A_i^{\mathrm{cls}}\right)}_{\text{Visual Saliency}} + \underbrace{\gamma\mathrm{Div}\left(\mathrm{CosSim}\left(\widetilde{\mathbf{v}}_i,\widetilde{\mathbf{v}}_d\right)\right)}_{\text{Visual Diversity}}. \quad (4)$$

Here, $\mathcal{S}_{\mathrm{v}}$ denotes the complete set of visual token indices, $\mathbf{q}_j$ is the mean subword embedding of the j-th noun or proper noun extracted from the question, and $\widetilde{\mathbf{v}}_i$ and $\widetilde{\mathbf{q}}_j$ are fake-quantized copies of the projector feature. The fake-quantized features are used only by the selector, whereas the corresponding original FP16 features are forwarded to the language backbone after selection.

The three signals are applied sequentially and serve complementary purposes. First, the **Query Relevance** stage independently ranks the visual tokens for each query concept according to the cosine similarity between $\widetilde{\mathbf{q}}_j$ and $\widetilde{\mathbf{v}}_i$. For every concept embedding $\mathbf{q}_j$, P4Q constructs a concept-specific ranking of the candidate visual tokens and retains the highest-scoring tokens within the allocated quota. This concept-wise selection preserves localized visual evidence associated with multiple entities or attributes and prevents a single dominant concept from occupying the entire semantic budget. However, the resulting signal captures only visual evidence directly aligned with the explicit noun concepts in the query and may overlook salient objects or scene-level context that is not explicitly mentioned.

P4Q therefore performs **Visual Saliency** selection over the remaining visual-token sequence $\mathcal{S}_{\mathrm{v}'}$, which contains the tokens not retained by the preceding query-guided stage. Specifically, P4Q extracts the CLS-to-patch attention weights from the final multi-head self-attention layer of the vision tower. In this attention operation, the final-layer [CLS] token serves as the query, while the patch tokens indexed by $\mathcal{S}_{\mathrm{v}'}$ serve as the candidate visual keys. The attention weights are then averaged across heads to obtain the saliency score A_i^{cls} for each remaining visual token i. The tokens with the highest saliency scores are retained within the budget, supplementing the query-relevant evidence with intrinsically salient objects and global scene information.

Although the first two stages preserve query relevant and visually salient evidence, the selected tokens may remain concentrated in a small number of semantically or spatially correlated regions. The remaining budget is therefore allocated to **Visual Diversity** completion. Specifically, for each

token in the remaining candidate pool, the maximum cosine similarity to any tokens in the current retained set is computed. The candidate with the lowest maximum similarity is retained, as it is the least redundant with the current selection. After each selection, the current retained set is updated and the similarity scores of the remaining candidates are recomputed. This max-min procedure is repeated until the allocated budget is exhausted, progressively extending the retained visual coverage to complementary objects, background regions, and spatial context.

We set the budget ratios to $\alpha = 0.3$, $\beta = 0.2$, and $\gamma = 0.5$, allocating 30%, 20%, and 50% of the retained-token budget to Query Relevance, Visual Saliency, and Visual Diversity, respectively. Performance remains stable under reasonable variations of these ratios, indicating that P4Q is not sensitive to their exact values. Detailed ablation results are provided in Appendix B.2.

3.3 Pruning-Aware Quantization Calibration

Post-training quantization estimates its transformation and clipping parameters from the activation distributions observed during calibration. Dense calibration therefore produces parameters adapted to sequences containing all visual tokens, whereas the deployed model operates on shorter and sample-dependent sequences after pruning. Visual-token removal changes the token support and sequence length processed by the language model, which in turn modifies attention interactions and the activation distributions of subsequent blocks. To reduce this mismatch, we apply the deployment-time visual token selector before collecting calibration activations.

Specifically, we construct a calibration set of 128 samples using a fixed random seed. For each sample, the fixed P4Q selector receives only the image and question and generates the pruned multimodal sequence using exactly the same token budget and selection configuration as during inference. We then perform block-wise quantization calibration on the resulting pruned activations. For language-model block ℓ, the calibration objective is:

$$\theta^*_{\ell,\mathrm{keep}} = \arg\min_{\theta_\ell} \frac{1}{B} \sum_{n=1}^{B} \frac{1}{m_n^{\mathrm{keep}} d} \left\| f^{\mathrm{W4A4}}_{\ell,\theta_\ell}\left(H^{\mathrm{FP,keep}}_{\ell,n}\right) - f^{\mathrm{FP}}_{\ell}\left(H^{\mathrm{FP,keep}}_{\ell,n}\right) \right\|_F^2 . \tag{5}$$

Here, $B = 128$ is the number of calibration samples, and n and ℓ index the sample and language-model block, respectively. $H^{\mathrm{FP,keep}}_{\ell,n}$ denotes the input to block ℓ obtained by propagating the P4Q-retained sequence through the preceding full-precision blocks, where keep distinguishes this sequence from its dense counterpart. m_n^{keep} is the corresponding multimodal sequence length, and d is the hidden dimension. The functions f^{FP}_{ℓ} and $f^{\mathrm{W4A4}}_{\ell,\theta_\ell}$ represent the full-precision block and its W4A4 fake-quantized counterpart. Their outputs are reconstructed from the same pruned input by optimizing the diagonal channel scales and the weight- and activation-clipping parameters in θ_ℓ. Calibration proceeds block by block, with the full-precision output of block ℓ serving as the input to block $\ell + 1$, thereby avoiding the accumulation of upstream quantization errors. $\theta^*_{\ell,\mathrm{keep}}$ denotes the resulting optimal parameters for the P4Q-pruned calibration path. Further details are provided in Appendix B.3.

4 Experiments

4.1 Experimental Setup

Benchmarks and Baselines. We evaluate P4Q on eight established image-understanding benchmarks: MMMU (Yue et al., 2024), VizWiz (Gurari et al., 2018), ScienceQA-TEST (SQA) (Lu et al., 2022), MME (Fu et al., 2025), GQA (Hudson & Manning, 2019), MMBench-CN ($\mathrm{MMB}^{\mathrm{CN}}$)(Liu et al., 2024), POPE (Li et al., 2023b), and SEEDBench-IMG (SEED) (Li et al., 2023a). We compare against five representative baselines: three visual token pruning methods, VisPruner (Zhang et al., 2025), VisionZip (Yang et al., 2025), and SpecFlow (Li et al., 2026c), and two PTQ methods, QuaRot (Ashkboos et al., 2024) and FlatQuant (Sun et al., 2025). All experiments use NVIDIA RTX 4090 GPUs with 24 GB of memory. End-to-end inference speed and peak GPU memory consumption are measured on a single GPU with a batch size of 1.

4.2 Main Results

Table 1: Main comparison of pruning, quantization and our joint method. E2E denotes the end-to-end speedup, and Mem. denotes the peak memory footprint. The best performance is marked in red.

Methods	MMMU	VizWiz	SQA	MME	GQA	MMB^{CN}	POPE	SEED	Avg.↑	E2E↑	Mem.↓
LLaVA-NeXT-7B	*Upper Bound, 2880 Tokens* **(100%)**										
Vanilla	36.7	58.7	69.4	1725.1	63.9	58.8	86.5	69.1	100%	1×	100%
Pruning	*Retain 320 Tokens* (↓ 88.9%)										
VisPruner(ICCV25)	35.3	56.2	66.8	1585.8	57.7	52.5	81.3	60.9	92.7%	1.63×	96.9%
VisionZip(CVPR25)	35.9	57.5	66.4	1616.9	58.1	54.3	79.3	62.9	93.9%	1.95×	98.6%
SpecFlow(ICML26)	36.6	57.8	67.7	1651.6	58.6	54.9	79.9	63.6	95.1%	1.62×	98.8%
Quant	*W4A4*										
QuaRot(NeurIPS24)	29.8	53.2	60.6	1596.9	61.1	37.3	83.7	65.7	87.8%	1.45×	70.1%
FlatQuant(ICML25)	33.0	54.5	64.2	1713.2	62.1	54.8	84.9	68.4	95.3%	1.83×	70.8%
Joint	*Retain 320 Tokens* (↓ 88.9%) *& W4A4*										
P4Q(Ours)	36.7	56.5	67.3	1706.2	59.2	51.7	87.8	62.3	95.5%	2.80×	51.3%
LLaVA-1.5-7B	*Upper Bound, 576 Tokens* **(100%)**										
Vanilla	34.8	54.3	66.3	1650.7	60.6	54.2	81.8	63.8	100%	1×	100%
Pruning	*Retain 64 Tokens* (↓ 88.9%)										
VisPruner(ICCV25)	33.3	54.2	67.0	1514.6	53.0	51.7	70.5	55.6	93.1%	1.23×	97.9%
VisionZip(CVPR25)	35.6	54.5	67.1	1535.2	53.4	51.9	72.2	57.1	94.8%	1.46×	97.8%
SpecFlow(ICML26)	36.7	54.2	66.9	1475.0	56.8	53.4	70.0	56.2	95.2%	1.35×	98.6%
Quant	*W4A4*										
QuaRot(NeurIPS24)	31.5	54.2	59.2	1274.1	56.2	33.5	69.9	59.2	86.2%	1.69×	49.8%
FlatQuant(ICML25)	35.1	53.7	65.0	1644.0	54.4	47.5	82.6	57.9	95.8%	1.75×	50.9%
Joint	*Retain 64 Tokens* (↓ 88.9%) *& W4A4*										
P4Q(Ours)	34.2	54.0	66.4	1585.3	55.2	48.3	83.7	58.8	96.1%	2.49×	42.0%

Table 2: P4Q Generalization on Qwen2.5-VL-7B.

Methods	MMMU	VizWiz	SQA	MMB^{CN}	Avg.↑
Qwen2.5-VL-7B	*Upper Bound, 576 Tokens* **(100%)**				
Vanilla	51.4	71.2	87.7	81.6	100%
	Retain 192 Tokens (↓ 66.7%)				
SpecFlow(ICML26)	49.0	69.1	82.7	77.8	95.5%
P4Q(Ours)	50.7	69.4	83.5	78.4	96.8%
	Retain 64 Tokens (↓ 88.9%)				
SpecFlow(ICML26)	47.3	63.6	80.9	70.3	89.9%
P4Q(Ours)	46.9	63.9	81.3	70.7	90.1%

Table 1 compares P4Q with five representative baselines, across eight image-understanding benchmarks. The pruning-based configurations retain 11.1% of the visual tokens, while the quantization-based configurations use W4A4 settings. It is worth emphasizing that the reported performance retention, end-to-end speedup, and peak GPU memory usage are all averaged across eight benchmarks. Overall, P4Q maintains comparable task performance on LLaVA-NeXT-7B while achieving an average 2.8× end-to-end inference speedup and using only 51.3% of the peak GPU memory required by the vanilla model. On LLaVA-1.5-7B, P4Q likewise maintains comparable performance while delivering an average 2.49× end-to-end inference speedup and reducing peak GPU memory usage by 58% relative to the vanilla model. Together, these results show that P4Q combines quantization-aware token selection with pruning-aware calibration to shorten visual-token sequences while using W4A4 weights and activations. On the two evaluated LLaVA models, this co-design maintains comparable task performance while substantially reducing inference latency and peak GPU memory usage relative to the uncompressed baselines. Further details of the efficiency evaluation are provided in Appendix B.4.

To further evaluate the generalizability of P4Q across different VLM architectures, we extend our experiments to Qwen2.5-VL-7B, whose vision backbone and multimodal projector differ substantially from those of the LLaVA family. As shown in Table 2, P4Q achieves average performance comparable to SpecFlow under both visual-token pruning budgets. Notably, P4Q follows the same joint W4A4 quantization and pruning configuration as in Table 1, whereas SpecFlow is evaluated with visual token pruning alone. These results demonstrate that P4Q remains effective on VLM architectures beyond the LLaVA family.

4.3 Apple-to-Apple Comparison with Peer Work

To the best of our knowledge, no formally published work has jointly combined quantization and pruning to accelerate VLMs. Nevertheless, several concurrent works have appeared on arXiv, which we also include in our comparisons. Results are listed in Table 3. The comparison shows that with merely 11% token retention, our accuracy exceeds QUOTA's result at 20% token retention by 1.1 percentage points. When compared under the more similar setting (QUOTA with 10% retained tokens), our method gains a 3 percentage point advantage.

Table 3: Comparison on LLaVA-1.5-7B. QUOTA (Li et al., 2026a) reports performance retention (%) from its original paper.

Methods	MME	GQA	POPE	SEED	Avg.↑
	Upper Bound, 576 Tokens **(100%)**				
Vanilla	1650.7	60.6	81.8	63.8	100%
	Retain 20% Tokens (↓ 80.0%) *& W4A4*				
QUOTA(arXiv26)	90.82%	93.53%	97.56%	95.09%	94.3%
	Retain 10% Tokens (↓ 90.0%) *& W4A4*				
QUOTA(arXiv26)	90.32%	90.36%	96.01%	93.02%	92.4%
	Retain 64 Tokens (↓ 88.9%) *& W4A4*				
P4Q(Ours)	1585.3	55.2	83.7	58.8	95.4%

4.4 Ablation Study

We evaluate the components of Quantization-Aware Pre-LLM Selection and Pruning-Aware Quantization Calibration on LLaVA-NeXT-7B. Table 4 reports average accuracy retention across five settings. Row 1 is the vanilla FP16 model. W4A4 indicates post-training quantization, while Pruning indicates an 88.9% visual-token pruning rate. When Pruned Calibration is disabled, quantization is calibrated on dense sequences. Fake Quant for Pruning applies the selector-only fake quantization described in Section 3.2 to the projector features used for token scoring. When it is disabled, selection uses the native FP16 features. Row 5 is the full P4Q setting used in Table 1.

Table 4: Ablation study of quantization-aware token selection and pruning-aware calibration. PC is pruned calibration. FQP is fake quant for pruning.

Row	Quantization		Pruning		Avg. Accuracy Retention (%)
	W4A4	PC	Pruning	FQP	
1 (Vanilla)	–	–	–	–	100.0%
2	✓	–	✓	–	92.1%
3	✓	–	✓	✓	92.5%
4	✓	✓	✓	–	93.2%
5 (Ours)	✓	✓	✓	✓	95.5%

Specifically, Row 2 corresponds to the setting where the quantized model is first loaded, followed by token pruning. It achieves a performance 3.4 percentage points lower than the final joint optimization method. This verifies that the superiority of our approach does not merely stem from the simple combination of quantization and pruning, and demonstrates the necessity of jointly designing quantization and pruning strategies. Comparing the experimental results of Rows 3, 4 and 5 verifies the effectiveness of the quantization and pruning techniques proposed in this paper.

5 Conclusion

We rethink visual token pruning and quantization as a coupled compression problem and introduce P4Q, a practical co-design framework for efficient VLM inference. P4Q combines quantization-aware pre-LLM token selection with pruning-aware calibration to shorten visual-token sequences while using W4A4 weights and activations. Experiments across eight multimodal benchmarks on the two evaluated LLaVA models show comparable task performance under W4A4 quantization and an 88.9% visual-token pruning rate, while controlled ablations on LLaVA-NeXT-7B support the benefit of combining the two design choices. Relative to the uncompressed baselines, P4Q achieves average end-to-end speedups of $2.8\times$ and $2.49\times$ on LLaVA-NeXT-7B and LLaVA-1.5-7B, respectively, while substantially reducing peak GPU memory usage.

AI USE STATEMENT

In this work, generative AI tools were used to assist with language editing and manuscript checks, including grammar, clarity, concision, presentation, and consistency. They were not used to generate the core research ideas or design the methodology or experiments. All AI-assisted edits and suggestions were carefully reviewed and approved by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

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

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 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. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Juncan Deng and Kejie Huang. Vlm-ptq: Efficient post-training quantization for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24696–24705, 2026.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 11198–11201, 2024.

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22826–22835. IEEE, 2025.

Hengyu Fang, Yijiang Liu, Yuan Du, Li Du, and Huanrui Yang. Sqap-vla: A synergistic quantization-aware pruning framework for high-performance vision-language-action models. *arXiv preprint arXiv:2509.09090*, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4899. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d79a27cf2772fe00be7f341efc0eb517-Paper-Datasets_and_Benchmarks_Track.pdf.

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

Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *European conference on computer vision*, pp. 214–230. Springer, 2024.

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

Janghwan Lee, Minsoo Kim, Seungcheol Baek, Seok Joong Hwang, Wonyong Sung, and Jungwook Choi. Enhancing computation efficiency in large language models through weight and activation quantization. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14726–14739, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.910. URL https://aclanthology.org/2023.emnlp-main.910/.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

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.

Xinqing Li, Xin He, Xindong Zhang, Ming-Ming Cheng, Lei Zhang, and Yun Liu. Towards joint quantization and token pruning of vision-language models. *arXiv preprint arXiv:2604.17320*, 2026a.

Xu Li, Yi Zheng, Mengyang Zhao, Yuxuan Liang, Zhe Liu, Rui Zhu, Xiaolei Chen, Wei Zhou, Baoquan Zhao, and Juncen Guo. Crisp: Pre-llm yet text-driven visual token pruning for efficient lvlm inference. *arXiv preprint arXiv:2607.16326*, 2026b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023b.

Zhaoyang Li, Yanjun Li, Wangkai Li, Yujia Chen, and Tianzhu Zhang. Spectral heat flow for conservative token condensation in vision-language models. In *Forty-third International Conference on Machine Learning*, 2026c. URL https://openreview.net/forum?id=QNdqUl17Un.

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

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

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

Zhenhao Shang, Haizhao Jing, Guoting Wei, Haokui Zhang, Rong Xiao, Jianqing Gao, and Peng Wang. Rethinking practical and efficient quantization calibration for vision-language models. *arXiv preprint arXiv:2602.07899*, 2026.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness matters for LLM quantization. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57587–57613. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/sun25l.html.

Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-vlm: Post-training quantization for large vision-language models. *Advances in Neural Information Processing Systems*, 37:114553–114573, 2024.

Xinhao Wang, Zhonyu Xia, Zhiwei Lin, Zhe Li, and Yongtao Wang. Qapruner: Quantization-aware vision token pruning for multimodal large language models. *arXiv preprint arXiv:2604.02816*, 2026.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15537–15549, 2025.

Miles Williams and Nikolaos Aletras. On the impact of calibration data in post-training quantization and pruning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10100–10118, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.544. URL https://aclanthology.org/2024.acl-long.544/.

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.

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, 2025.

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.

A Additional Details of Low-Bit Rank Shift Analysis

Experimental details. We collected visual token ranking statistics on LLaVA-NeXT-7B using 128 fixed samples from the ScienceQA training split with random seed 42. FP16 and W4A4 inference used identical images, questions, packed input sequences, attention masks, and positional indices to ensure token-wise correspondence between the two precision settings. All samples were processed with the complete visual-token sequence, and no visual-token pruning was applied during statistics collection.

Only the language-model decoder was quantized, while the vision tower and multimodal projector remained unchanged. The W4A4 setting applied symmetric round-to-nearest quantization with per-output-channel 4-bit weights and per-token 4-bit activations. Both weight and activation group sizes were set to -1, the activation clipping ratio was set to 1.0, and no INT8 exception was applied to the down-projection layers. For each sample, the final non-special token of the user question was used as the text query, and only image patch tokens were included as candidate visual keys, excluding the row-newline tokens introduced by LLaVA-NeXT AnyRes packing. For each target decoder layer, visual-token importance was measured using the attention scores from the selected question token to the candidate visual patch tokens.

Sample-balanced W4A4 rank shift relative to FP16. For each sample n and decoder layer ℓ, the visual tokens are ranked in descending order of their importance scores. The resulting rank is normalized by the number of valid visual patch tokens N_n:

$$r^{\tau}_{n,\ell,i} = \frac{\mathrm{rank}^{\mathrm{avg}}_{\downarrow}\left(s^{\tau}_{n,\ell,i}\right) - 1}{N_n - 1}, \qquad \Delta r_{n,\ell,i} = r^{\mathrm{W4A4}}_{n,\ell,i} - r^{\mathrm{FP16}}_{n,\ell,i}, \tag{6}$$

where $\tau \in \{\mathrm{FP16}, \mathrm{W4A4}\}$, and average ranks are assigned to tokens with tied importance scores. Accordingly, $r = 0$ denotes the most important visual token and $r = 1$ denotes the least important one. A positive Δr indicates that W4A4 moves a token toward a less important position, whereas a negative Δr indicates that the token is promoted toward a more important position. FP16 and W4A4 observations are aligned using each visual token's original packed index, ensuring that the displacement is computed between the same token under the two precision settings.

To construct Fig. 1(b), we divide the same-layer FP16 normalized rank into $B = 20$ equal-width intervals, denoted by $\mathcal{B}_b$. For each sample, we first average the signed rank shifts of all visual tokens whose FP16 ranks fall within the same interval. The plotted value is then obtained by averaging these sample-level means over the 128 samples:

$$\overline{\Delta r}_{\ell,b} = \frac{1}{128}\sum_{n=1}^{128} \mathrm{Mean}_{\,i:\, r^{\mathrm{FP16}}_{n,\ell,i} \in \mathcal{B}_b} \left(\Delta r_{n,\ell,i}\right). \tag{7}$$

The horizontal coordinate is the center of the corresponding FP16-rank interval, while the vertical coordinate is $100\overline{\Delta r}_{\ell,b}$, expressed in normalized-rank percentage points. This sample-balanced aggregation gives every sample equal weight and prevents images containing more AnyRes visual tokens from dominating the curve. The shaded regions indicate 95% confidence intervals obtained from 2,000 non-parametric bootstrap resamples over the 128 sample-level means.

Because positive and negative displacements can cancel, the signed profile alone does not measure the overall magnitude of ranking rearrangement. We therefore additionally report the sample-balanced mean absolute rank shift, denoted as PRAS:

$$\mathrm{PRAS}_{\ell} = \frac{1}{128}\sum_{n=1}^{128}\left(\frac{1}{N_n}\sum_{i=1}^{N_n}\left|\Delta r^{\mathrm{W4A4}\leftarrow\mathrm{FP16}}_{n,\ell,i}\right|\right). \tag{8}$$

Here, ℓ denotes the decoder layer, n indexes the 128 evaluation samples, N_n is the number of valid visual patch tokens in sample n, and i indexes an individual visual token. The quantity $\Delta r^{\mathrm{W4A4}\leftarrow\mathrm{FP16}}_{n,\ell,i}$ is the signed normalized-rank displacement of token i under W4A4 relative to FP16, while the absolute-value operator removes its movement direction. The inner average measures the mean absolute rank shift within one sample, and the outer average assigns equal weight to all 128 samples. PRAS therefore measures the average magnitude of W4A4-induced rank rearrangement

without cancellation between positive and negative displacements. For example, $\mathrm{PRAS}_\ell = 0.216$ corresponds to an average absolute displacement of 21.6 rank percentage points. The PRAS values shown in the legend are 7.7%, 9.4%, 12.4%, and 21.6% for Layers 2, 4, 8, and 16, respectively.

Qualitative visualization of Top-K support changes induced by rank shift. To complement the sample-balanced quantitative analysis, we directly visualize the selected visual-token masks for a specific sample from the ScienceQA training split. For each layer and precision, the retained set is constructed independently from the corresponding attention scores:

$$\mathcal{T}^{\tau}_{n,\ell}(K) = \mathrm{TopK}_i\left(s^{\tau}_{n,\ell,i}, K\right), \qquad \rho^{\mathrm{repl}}_{n,\ell} = 1 - \frac{\left|\mathcal{T}^{\mathrm{FP16}}_{n,\ell}(K) \cap \mathcal{T}^{\mathrm{W4A4}}_{n,\ell}(K)\right|}{K}. \tag{9}$$

Here, $\mathcal{T}^{\tau}_{n,\ell}(K)$ denotes the indices of the K highest-scoring visual tokens under precision τ, and $\rho^{\mathrm{repl}}_{n,\ell}$ measures the fraction of the FP16 Top-K set that is replaced under W4A4.

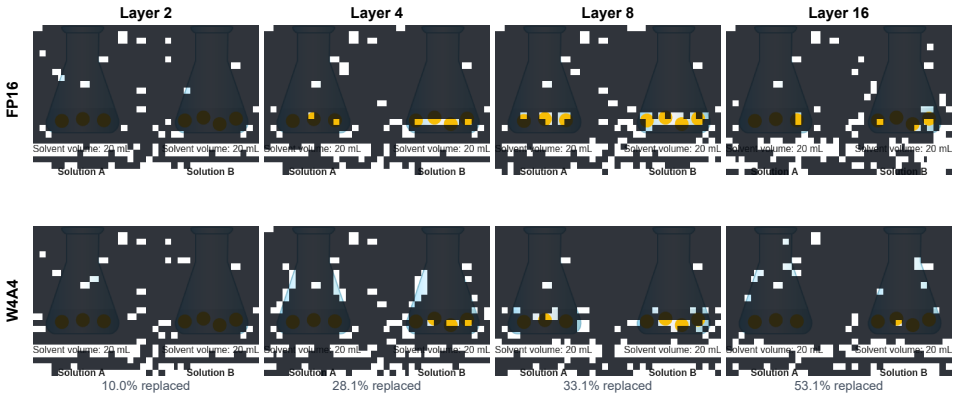


Figure 3: Qualitative visualization of the W4A4-induced visual-token selection changes for a specific sample. Each column compares the independently selected FP16 and W4A4 Top-K sets at the same decoder layer. All settings retain 11.1% visual patch tokens. Bright regions correspond to retained visual tokens, whereas dark regions correspond to masked tokens. The values below the W4A4 panels report the fraction of the FP16 Top-K set replaced under W4A4: 10.0%, 28.1%, 33.1%, and 53.1% at Layers 2, 4, 8, and 16, respectively.

This visualization shows the spatial locations of the visual tokens retained under FP16 at representative decoder layers and compares them with the tokens retained by the same selection strategy at the corresponding layers under W4A4. This single-sample comparison provides an intuitive qualitative illustration of how W4A4-induced rank shift changes the retained token set, whereas Fig. 1(b) presents the corresponding quantitative analysis over all 128 samples.

B Additional Experimental Details

This section provides additional implementation details for the experiments reported in the main manuscript, including the experimental configurations, calibration and evaluation settings, hyperparameter choices, and statistical procedures. These details complement the descriptions in the main text and facilitate the interpretation and reproduction of our experiments. By documenting the settings underlying the reported results, we aim to provide a more transparent account of the empirical evidence supporting our main findings and conclusions.

B.1 Experimental Settings

All experiments were conducted on a server equipped with NVIDIA GeForce RTX 4090 GPUs, each with 24 GB of memory. The implementation used Python 3.10.20, PyTorch 2.6.0 and CUDA 11.8. We evaluate P4Q on a diverse suite of eight established image-understanding benchmarks, including MMMU (Yue et al., 2024), VizWiz (Gurari et al., 2018), ScienceQA_TEST (SQA) (Lu et al., 2022), MME (Fu et al., 2025), GQA (Hudson & Manning, 2019), MMBench_CN (MMB^{CN}) (Liu et al., 2024), POPE (Li et al., 2023b), and SEEDBench_IMG (SEED) (Li et al., 2023a). All benchmarks are evaluated using VLMEvalKit (Duan et al., 2024), following their official evaluation protocols and default settings.

B.2 Sensitivity to Token-budget Allocation

Table 5: Sensitivity analysis of the token-budget allocation hyperparameters on LLaVA-1.5-7B at an 88.9% visual-token pruning ratio. α, β, and γ denote the proportions allocated to text-guided anchors, visual-saliency anchors, and diversity completion.

Row	Token-Budget			Benchmark			Avg. Accuracy Retention (%)
	α	β	γ	MMMU	SQA	MME	
Vanilla	–	–	–	34.8	66.3	1650.7	100.0%
1	0.42	0.28	0.30	34.3	65.5	1600.8	98.1%
2	0.20	0.30	0.50	34.9	66.0	1564.5	98.2%
3	0.40	0.10	0.50	34.7	65.7	1574.7	98.1%
4	0.12	0.18	0.70	33.3	66.3	1586.1	97.3%
5	0.18	0.12	0.70	34.0	65.9	1572.9	97.5%
6	0.24	0.06	0.70	34.0	66.1	1545.2	97.0%
7 (Ours)	0.30	0.20	0.50	34.2	66.4	1585.3	98.2%

Table 5 evaluates the sensitivity of P4Q to the allocation ratios α, β, and γ while keeping the total visual-token budget and the 88.9% pruning ratio fixed. Across all tested allocations, the average performance retention remains between 97.0% and 98.2%, corresponding to a maximum variation of only 1.2 percentage points. The variations on the individual benchmarks are also limited: MMMU ranges from 33.3 to 34.9, ScienceQA from 65.5 to 66.4, and MME from 1545.2 to 1600.8. In particular, when the diversity budget is fixed at $\gamma = 0.5$, substantially redistributing the remaining budget between Query Relevance and Visual Saliency changes the average performance retention by at most 0.1 percentage points. Allocating a larger proportion to Visual Diversity ($\gamma = 0.7$) results in a small reduction to 97.0–97.5%, suggesting that excessive diversity completion may reduce the budget available for query-relevant and visually salient evidence. We therefore adopt $\alpha = 0.3$, $\beta = 0.2$, and $\gamma = 0.5$ as a balanced default. This setting achieves 98.2% average performance retention, tying the best result among the evaluated allocations. Overall, the results show that the three budget ratios can cause modest benchmark-level fluctuations but that P4Q remains robust to reasonable variations in their values.

B.3 PTQ Calibration Implementation Details

For the W4A4 setting, we quantized the language-model decoder to 4-bit weights and 4-bit activations while retaining the key and value caches in FP16; the vision tower and multimodal projector remained in full precision. Weight quantization was symmetric and per output channel, activation

quantization was symmetric and per token, and no additional group-wise subdivision was used. We sampled 128 fixed image-question pairs from the ScienceQA training split with random seed 42 as the calibration set. To align calibration with the deployment-time inference path, the same P4Q pruning strategy and model-specific visual-token retention ratio used during inference were applied before capturing the decoder calibration inputs. Calibration was then performed sequentially over the decoder blocks. For each block, its full-precision outputs on the pruned calibration sequences were used as reconstruction targets, and the corresponding quantized block was optimized by minimizing the mean-squared reconstruction error. The learnable calibration parameters included factorized channel transformations, per-channel diagonal scales, and learnable weight and activation clipping factors; the original model parameters were frozen. The diagonal scales were initialized from the observed activation ranges and weight ranges using an interpolation exponent of 0.3. Each decoder block was calibrated for 15 epochs with an effective batch size of 1 using AdamW. The learning rate was set to 5×10^{-3} for the transformation matrices and diagonal scales and 5×10^{-2} for the clipping parameters, followed by cosine annealing to 5×10^{-6}. After calibration, the learned transformations and diagonal scales were folded into the decoder weights, and the resulting weights were quantized using round-to-nearest quantization for W4A4 inference.

B.4 Efficiency Evaluation Details

This section provides additional details on the computational efficiency improvements reported in Table 1. Specifically, E2E denotes the average per-sample end-to-end latency measured across the eight evaluation datasets with a batch size of 1. The latency measurement covers both data loading and the complete VLM inference pipeline. All experiments were conducted on a single NVIDIA GeForce RTX 4090 GPU with 24 GB of memory. Table 6 presents the dataset-level measurements underlying the average efficiency improvements reported in Table 1. Notably, for LLaVA-NeXT-7B, the end-to-end inference speedups on MMMU, MME, and POPE all exceeded **3**×, with the largest speedup of **3.26**× observed on MME.

Table 6: End-to-end inference latency and peak GPU memory of P4Q on LLaVA-NeXT-7B and LLaVA-1.5-7B. Latency is reported as the mean processing time per sample in milliseconds, and peak memory is reported in GB. For latency, the Avg. column reports the average dataset-wise speedup over Vanilla. For memory, it reports the average memory consumption relative to Vanilla. Lower latency and memory consumption are better.

Metric	Method	MMMU	VizWiz	SQA	MME	GQA	MMB^{CN}	POPE	SEED	Avg.
LLaVA-NeXT-7B										
E2E Latency	Vanilla	2144.33	2031.77	635.79	886.13	378.12	1044.38	552.55	367.08	1.00×
(ms/sample)	P4Q	707.36	806.95	246.78	271.57	165.86	367.40	179.10	130.63	2.80×
Peak Memory	Vanilla	20.31	14.88	14.95	14.90	14.90	15.14	14.89	14.95	100.0%
(GB)	P4Q	14.64	7.22	7.08	7.02	6.62	8.01	7.29	7.32	51.3%
LLaVA-1.5-7B										
E2E Latency	Vanilla	600.48	1001.80	184.44	411.05	361.51	367.36	354.53	297.55	1.00×
(ms/sample)	P4Q	230.92	366.19	111.12	178.87	125.90	165.15	114.86	121.77	2.49×
Peak Memory	Vanilla	15.00	13.56	13.82	13.53	13.53	14.10	13.52	13.58	100.0%
(GB)	P4Q	7.31	5.34	5.67	5.48	4.66	6.61	5.73	5.77	42.0%

We further investigate and report the stage-wise acceleration underlying the end-to-end inference gains. Specifically, on LLaVA-NeXT-7B, we measure the average per-sample prefill latency across eight benchmarks. The reported prefill latency covers the language-model prefill computation from cache readiness to prefill completion. As shown in Fig. 4, P4Q consistently achieves substantial prefill acceleration across all eight benchmarks, with an average speedup of **3.33**× over FP16. In particular, the largest improvement is observed on VizWiz, where P4Q achieves a prefill speedup of **4.58**×.

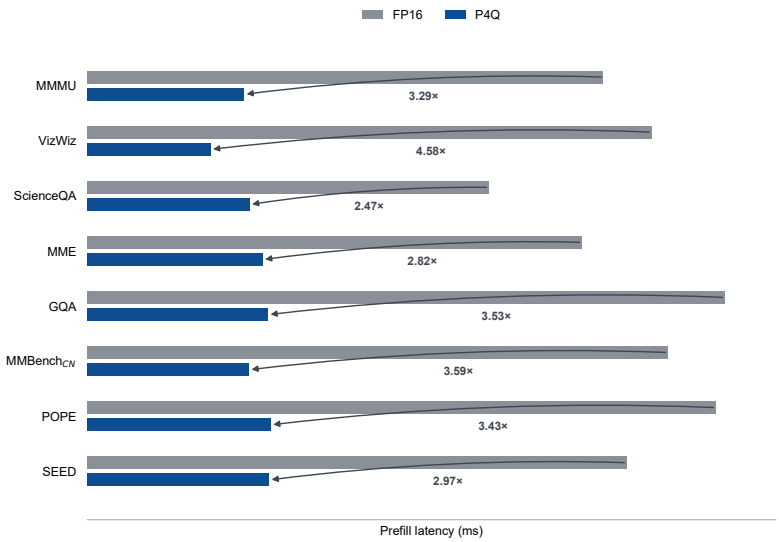


Figure 4: Per-sample prefill latency and speedup of P4Q over FP16 on eight benchmarks using LLaVA-NeXT-7B. Each pair of horizontal bars compares the average prefill latency of FP16 and P4Q, while the annotated values indicate the corresponding speedup. P4Q achieves an average prefill speedup of $3.33\times$.