MODE: Modality-Decomposed Expert-Level Mixed-Precision Quantization for MoE Multimodal LLMs

Yuanteng Chen[1,2,3] *, Nanxin Zeng[2] *, Peisong Wang[1,2] †, Zhilei Liu[1,2], Yuantian Shao[1,5], Shiqiang Lang[3], Tao Liu[3], Chuangyi Li[1,2], Qinghao Hu[1,2], Gang Li[1,2], Jing Liu[1,2,3], Jian Cheng[1,2,3] †,
[1]Institute of Automation, Chinese Academy of Sciences,
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences,
[3]Zhongguancun Academy, [4]School of Computer Science, University of Sydney,
[5]Nanjing University of Science and Technology
Correspondence: {peisong.wang, jcheng}@nlpr.ia.ac.cn

Abstract

Mixture-of-Experts Multimodal Large Language Models (MoE-MLLMs) offer remarkable performance but incur prohibitive GPU memory costs, making compression essential. Among PTQ methods, expert-level mixed-precision quantization has proven effective for MoE-LLMs, yet suffers notable degradation on MoE-MLLMs due to two overlooked biases in expert importance estimation. (1) At the **cross-modal** level, the numerical dominance of vision tokens causes expert selection frequency to be dominated by vision tokens, masking experts that are critical to the text modality; (2) at the **intra-vision** level, the large proportion of redundant vision tokens further skew frequency statistics, obscuring experts critical for informative visual content. To bridge gaps, we propose **MODE**, a **mo**dality-**d**ecomposed **e**xpert-level mixed-precision quantization framework for MoE-MLLMs that decomposes expert selection frequency by modality, filters redundant vision tokens to obtain denoised visual frequency, and further evaluates quantization sensitivity per modality as a complementary signal to frequency-based estimation. These signals are integrated into an Integer Linear Programming formulation to assign per-expert bit-widths under a given budget. Extensive experiments show that MODE is particularly well-suited for MoE-MLLMs, limiting average performance loss to within 2.9% at W3A16, with larger gains at the extreme 2-bit setting. The code is available at Github.

1 Introduction

Multimodal Large Language Models (MLLMs) have achieved remarkable success across a wide range of vision-language tasks (Radford et al., 2021; Team et al., 2026). By aligning visual perception with linguistic reasoning, they empower intelligent systems to perceive, understand, and interact with the visual world in a unified framework.

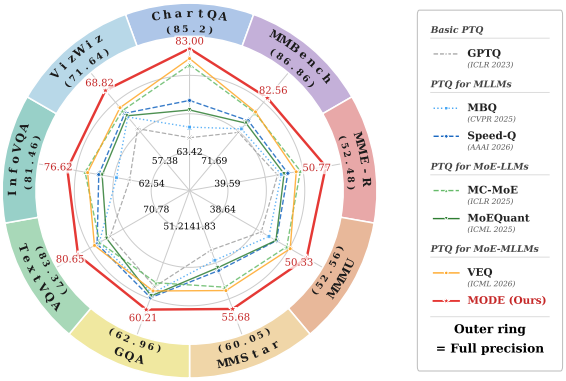

Figure 1: Performance comparison on Qwen3-VL-30B-A3B-Instruct under 3-bit weight quantization (W3A16).

With the growing scale of MLLMs, the Mixture-of-Experts (MoE) architecture (Artetxe et al., 2022) has become a popular solution to manage computational cost, activating only a sparse subset of experts per token to enable efficient parameter scaling while keeping training and inference FLOPs low (Fedus et al., 2022). However, since all expert parameters must reside on GPUs before inference regardless of activation sparsity, deploying large-scale MoE-MLLMs (Lin et al., 2024) like Qwen3-VL-30B-A3B-Instruct (Team, 2025) still demands prohibitive memory, which severely limits their practical use in resource-constrained settings.

Post-Training Quantization (PTQ) offers a practical way to reduce memory footprint without retraining, yet existing PTQ methods target either dense MLLMs or MoE-LLMs, and neither transfers well to MoE-MLLMs. Specifically, PTQ methods designed for dense MLLMs (Li et al., 2024b; Guo et al., 2025) recognize the modality gap between text and vision tokens, but treat the model as a monolithic whole and overlook the sparse activation and unequal expert contribution intrinsic to MoE architectures. Conversely, PTQ methods tai-

* Equal contribution.
† Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 20959–20977
October 24-29, 2026 ©2026 Association for Computational Linguistics

lored for MoE (Zheng et al., 2026; Hu et al., 2025) exploit these structural properties, but ignore the modality differences inherent to MLLMs.

Among PTQ methods for MoE, expert-level mixed-precision quantization (Huang et al., 2025) has emerged as a prevailing paradigm, which measures expert importance through activation frequency and assigns higher bit-widths to more critical experts while aggressively compressing less important ones. However, when applied to MoE-MLLMs, these methods aggregate activation statistics across all tokens regardless of modality, rendering expert importance estimates unreliable and leading to notable performance degradation.

In our study, we identify two distinct levels of bias that arise when expert-level mixed-precision quantization is applied to MoE-MLLMs. At the cross-modal level, vision tokens vastly outnumber text tokens in typical multimodal inputs, so global frequency statistics are dominated by vision-side routing patterns and systematically undervalue experts critical for textual reasoning. At the intra-vision level, pervasive redundancy among vision tokens further distorts frequency-based importance, as many tokens carry near-duplicate information yet each still casts a router vote, inflating the apparent importance of certain experts while obscuring the contributions of others.

Motivated by these observations, we propose **MODE**, a modality-aware quantization framework tailored to MoE-MLLMs that jointly leverages the heterogeneous expert structure of MoE for differentiated precision allocation and accounts for the modality heterogeneity and intra-vision redundancy inherent in multimodal inputs. To resolve the cross-modal bias, MODE separately collects and normalizes expert selection frequencies for text and vision tokens, so that text-critical experts are no longer overshadowed. To address the intra-vision distortion, MODE identifies key vision tokens at each layer and computes vision-side frequencies solely from them, filtering out noise from redundant visual content. Beyond frequency alone, MODE further evaluates each expert's quantization sensitivity separately under text and key vision tokens, yielding a modality-decomposed importance metric. The resulting scores are integrated into an Integer Linear Programming (ILP) formulation that produces an optimal per-expert precision allocation under a given bit budget.

We evaluate MODE on three MoE-MLLM families across ten multimodal benchmarks under various low-bit weight-only quantization settings. As illustrated in Figure 1, MODE consistently surpasses all compared PTQ methods designed for dense MLLMs, MoE-LLMs, and MoE-MLLMs, respectively. In particular, at the W3A16 setting MODE incurs only 2.84% and 2.08% average accuracy degradation on Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct, respectively, making practical deployment of 30B-scale MoE-MLLMs on a single consumer-grade GPU feasible.

2 Related Work

MLLMs Quantization. PTQ methods designed for MLLMs primarily focus on mitigating the distribution heterogeneity between vision and text modalities. SPEED-Q (Guo et al., 2025) adopts a staged quantization strategy that first quantizes the vision encoder and re-calibrates the projection layer for modality alignment. MBQ (Li et al., 2024b) recognizes that vision and language tokens exhibit different quantization sensitivities and introduces gradient-based sensitivity measures into calibration to balance reconstruction quality across modalities.

MoE Quantization. The sparse activation nature of MoE architectures brings new challenges and opportunities to PTQ. MoEQuant (Hu et al., 2025) addresses inter- and intra-expert activation imbalance through an expert-balanced calibration strategy. A more prevalent line of work exploits the heterogeneous expert contributions through mixed-precision quantization. MC-MoE (Huang et al., 2025) leverages expert activation frequency as an importance indicator, assigning higher bit-widths to more critical experts. MoQa (Zheng et al., 2026) builds upon this paradigm by introducing a channel-level dynamic adjustment mechanism. More recently, VEQ (Qin et al., 2026) takes a first step toward MoE-MLLMs by incorporating a modality-affinity-aware Hessian objective based on token–expert affinity to improve accuracy.

3 Motivation

Expert selection frequency, defined as the fraction of tokens routed to a given expert on a calibration set, is a natural and widely adopted importance proxy for mixed-precision quantization, as more frequently activated experts are considered more important in MoE. However, when transferred to MoE-MLLMs, we identify two distinct levels of bias that cause overall selection frequency to significantly deviate from true expert impor-

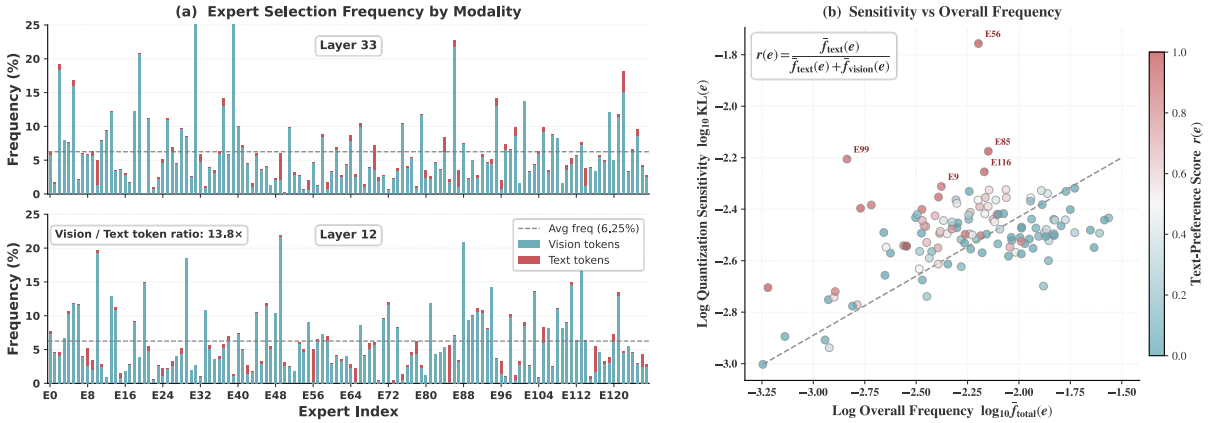

Figure 2: Cross-modal expert frequency bias in Qwen3-VL-30B-A3B-Instruct. (a) Per-expert selection frequency at Layers 12 and 33, decomposed into vision-token (blue) and text-token (red) contributions. (b) Quantization sensitivity (KL divergence between first output-token logits before and after 2-bit RTN quantization, log-scaled) versus overall selection frequency for all Layer-12 experts, colored by the text-preference score $r(e)$.

tance. We elaborate on each below, using Qwen3-VL-30B-A3B-Instruct as the representative model and 512 randomly sampled image-text pairs from ShareGPT4V (Chen et al., 2023) as calibration set.

3.1 Cross-Modal Expert Frequency Bias.

In typical multimodal inputs, a single image is encoded into hundreds or even thousands of visual tokens while the accompanying text prompt remains relatively short, so vision tokens outnumber text tokens by a large margin. This imbalance is an intrinsic property of multimodal inference, and consequently the overall expert selection frequency is dominated by vision-side routing patterns, causing experts that are heavily selected by text tokens but receive little vision traffic to be systematically undervalued under a global frequency ranking. To illustrate this, Figure 2(a) decomposes the per-expert selection frequency at two representative MoE layers into vision-token (blue) and text-token (red) contributions. Clear variation across experts confirms that experts do differ substantially in importance, supporting the general premise of frequency-based estimation; however, the text-token contribution is nearly invisible at this scale, and experts like E56 and E85 in Layer 12, despite being among the most heavily selected by text tokens, fall well below the layer average in total frequency and thus become inconspicuous.

Beyond frequency statistics, we further reveal this bias from the perspective of actual quantization loss. Figure 2(b) plots each Layer-12 expert's quantization sensitivity, measured as the KL divergence between the first output-token logits before and after 2-bit RTN quantization (log-scaled, averaged over all calibration samples), against its log overall frequency, with redder color indicating experts more frequently selected by text tokens but less by vision tokens. At the global level, a broadly positive trend between sensitivity and frequency is visible, supporting frequency as a reasonable importance proxy in general. At the individual expert level, however, several strongly text-preferred experts emerge as pronounced outliers: E56 and E99 exhibit disproportionately high quantization sensitivity despite having only low-to-moderate overall frequencies, indicating that they should be protected as text-critical experts rather than being dismissed by their modest overall activation.

These observations motivate a straightforward design principle: text-token and vision-token frequencies should be normalized independently before being combined into a unified importance score, ensuring that experts critical to each modality receive appropriate protection.

3.2 Intra-Vision Expert Frequency Bias

Recent work on MLLM token compression (Yang et al., 2026) consistently shows that only a small subset of vision tokens carry the core visual semantics, while the vast majority are redundant and can be pruned with negligible performance loss. In the context of MoE-MLLMs, however, a more pertinent question is whether these key vision tokens correspond to a distinct and stable group of experts—those that are truly important for visual modality. Our analysis gives an affirmative answer and further uncovers a second layer of frequency bias within vision modality itself: key and redundant vision tokens systematically activate different

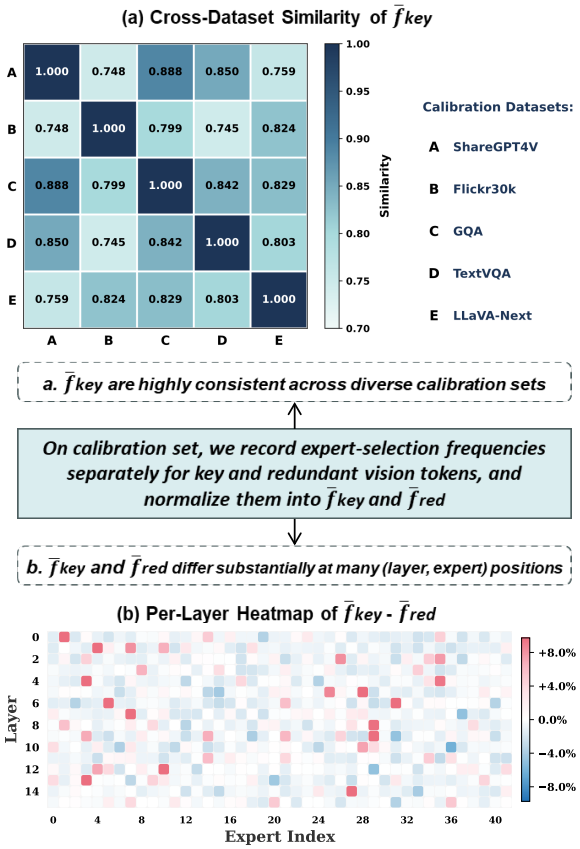

Figure 3: Intra-vision expert frequency bias. **(a)** $\bar{f}_{\text{key}}$ is highly consistent across five calibration datasets. **(b)** Within a single dataset, $\bar{f}_{\text{key}}$ and $\bar{f}_{\text{red}}$ exhibits pronounced deviations across (layer, expert) positions, revealing a systematic key–redundant routing bias.

experts, while the preference of key tokens remains stable across diverse data, as detailed below.

To distinguish key vision tokens from redundant ones, we follow the widely adopted attention-based criterion of SparseVLM (Zhang et al., 2025b): at each layer, we rank vision tokens by the total attention they receive from all text tokens, which reliably indicates their relevance to the current query, and treat the top 20% as key tokens and the remainder as redundant. We apply this selection at every layer, so that the key-token set adapts to how the model attends to visual content at different depths. A detailed description is provided in Appendix C.

With key and redundant vision tokens identified at every layer, we collect their layer-wise normalized expert-selection frequencies, $\bar{f}_{\text{key}}$ and $\bar{f}_{\text{red}}$, on calibration datasets and analyze from the following two complementary angles.

We first examine the cross-dataset behavior of $\bar{f}_{\text{key}}$ via its pairwise cosine similarity across the five datasets (details in Appendix D). As shown in Figure 3(a), the scores are consistently high (all above 0.74), indicating that key vision tokens converge to a stable set of preferred experts regardless of data composition. This stability links key vision tokens to a group of vision-critical experts intrinsic to the model, rather than to dataset-specific artifacts.

We then zoom into a single calibration set (ShareGPT4V) to ask whether redundant tokens share the same preferences. Figure 3(b) visualizes $\bar{f}_{\text{key}} - \bar{f}_{\text{red}}$ over a representative subset of (layer, expert) positions (the full 48×128 grid is omitted for space), where a clear divergence emerges: certain experts are substantially more activated by key tokens, while others are disproportionately used by redundant ones. Since redundant tokens dominate the vision-token population, treating all vision tokens indiscriminately biases the estimated expert importance toward redundant-preferred experts, leaving those most relevant to meaningful visual content under-protected.

Combining the two findings, expert importance for vision modality should be assessed through lens of key vision tokens rather than all vision tokens.

3.3 Concept Verification

To jointly validate the two dimensions of bias, we conduct a controlled mixed-precision experiment on Qwen3-VL-30B-A3B-Instruct. We fix all attention layers at 4-bit precision and quantize every MoE expert to 2 bits via RTN as the starting point. We then incrementally promote experts to 3 bits in steps of 8 per layer (out of 128), where the order of promotion is determined by a *simplified expert importance score* s built from per-layer expert selection frequencies $\bar{f}_x$ (normalized over experts) on different token groups x. We compare four strategies for computing s, differing only in which token-group frequencies are used:

(a) Global: $s = \bar{f}_{\text{total}}$, the selection frequency over all calibration tokens.

(b) Text+Vision: $s = \frac{1}{2}\bar{f}_{\text{text}} + \frac{1}{2}\bar{f}_{\text{vision}}$, balanc-

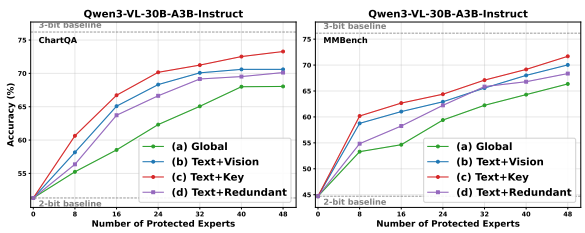

Figure 4: Concept verification. Experts are progressively promoted from 2-bit to 3-bit under four ranking strategies (a)–(d); results on ChartQA and MMBench validate both dimensions of frequency bias.

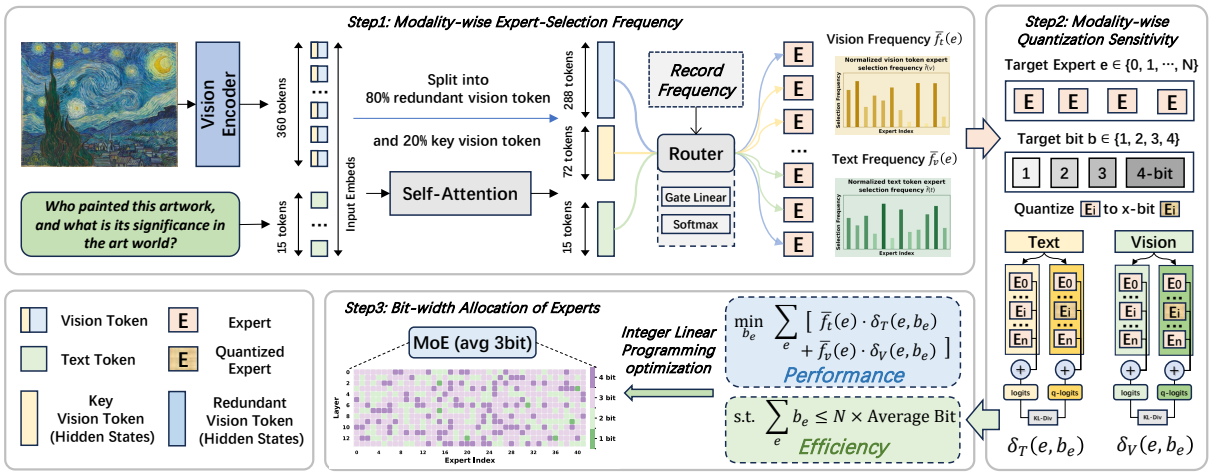

Figure 5: Pipeline of our quantization method for MoE-MLLMs. Modality-wise frequency and sensitivity are profiled per expert, then combined in an ILP that allocates expert-level bit-widths under a target budget.

ing text and all-vision tokens.

(c) Text+Key: $s = \frac{1}{2}\bar{f}_{\text{text}} + \frac{1}{2}\bar{f}_{\text{key}}$, replacing all vision tokens with only key vision tokens.

(d) Text+Redundant: $s = \frac{1}{2}\bar{f}_{\text{text}} + \frac{1}{2}\bar{f}_{\text{red}}$, replacing vision with only redundant vision tokens.

Under each strategy, experts with the highest scores are promoted first. Comparing (a) vs. (b) isolates the effect of modality-balanced ranking (inter-modality bias), while comparing (c) and (d) against (b) isolates the effect of which vision tokens drive the ranking (intra-vision bias).

Figure 4 reports the accuracy curves on ChartQA and MMBench. On both benchmarks, (b) substantially outperforms (a) at every protection level, confirming the importance of separately accounting for text and vision expert preferences. (c) further improves upon (b), while (d) falls slightly below, showing that key vision tokens provide a more faithful signal of vision-side expert importance than redundant ones. Together, these comparisons validate both dimensions of frequency bias.

4 Method

The concept verification in §3.3 shows that modality-aware frequency ranking significantly outperforms the global frequency baseline in expert importance estimation. However, frequency reflects *how often* an expert is selected, which does not fully align with *how much* its quantization degrades model performance. As evidenced by the outliers in Figure 2(b), experts with similar frequencies can exhibit vastly different quantization sensitivities; moreover, even for equally important experts, different bit-widths yield different precision–efficiency trade-offs. To address these, we build upon modality-wise frequency and further introduce modality-wise quantization sensitivity to form a more comprehensive importance metric, then formulate the bit-width allocation as an integer linear programming (ILP) problem that jointly optimizes all expert assignments under a given bit budget. The overall pipeline is illustrated in Figure 5.

Step 1: Modality-wise Expert Selection Frequency. Following the analysis in §3, we collect expert selection frequencies separately for text tokens and key vision tokens (the top 20% of vision tokens with the highest cumulative attention toward text tokens in each layer) on the calibration set. For each MoE layer, we count each expert's selections over all text tokens and all key vision tokens, respectively, and normalize the counts layer-wise so that each modality's distribution sums to one within every layer. Collecting these normalized frequencies across all layers gives $\bar{f}_t(e)$ and $\bar{f}_v(e)$ for every expert e in the model.

Step 2: Modality-wise Quantization Sensitivity. While frequency reflects which experts are important, quantization sensitivity measures how much each expert's quantization hurts model outputs. For each expert e and candidate bit-width b, we quantize *only* expert e to b bits while keeping all other parameters at full precision, and run a forward pass on the calibration set. To disentangle per-modality effects, the quantized weights are applied exclusively to one modality at a time: for text-side sensitivity $\delta_T(e, b)$, only text tokens routed to expert e use the quantized weights; for vision-side sensitivity $\delta_V(e, b)$, only key vision tokens do. Sensitivity is quantified as the KL divergence between the out-

put logits of the original and partially quantized models, averaged over the calibration set:

$$\delta_M(e,b) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} D_{\mathrm{KL}}\Big(p(x) \;\Big\|\; p_M^{(e\to b)}(x)\Big), \tag{1}$$

where $M \in \{T, V\}$ denotes the modality, $p(x)$ the full-precision output logits on sample x, and $p_M^{(e\to b)}(x)$ the logits when expert e is quantized to b bits with the quantized weights applied only to modality-M tokens. This per-expert, per-modality isolation ensures each sensitivity score faithfully reflects the impact of quantizing expert e on the corresponding modality.

Step 3: Bit-width Allocation via ILP. Equipped with modality-wise frequencies and sensitivities, we formulate expert-level bit-width allocation as an ILP problem, a formulation also adopted by prior MoE quantization work but with modality-agnostic importance terms (Huang et al., 2025; Duanmu et al., 2025). The resulting ILP can be solved within seconds for the entire model. We introduce a binary indicator $x_{e,b} \in \{0,1\}$ for each expert e and candidate bit-width $b \in \mathcal{B} = \{1,2,3,4\}$, where $x_{e,b} = 1$ iff expert e is quantized to b bits. The ILP is formulated as:

$$\min \sum_{e} \sum_{b \in \mathcal{B}} \Big[\bar{f}_t(e)\cdot\delta_T(e,b) + \bar{f}_v(e)\cdot\delta_V(e,b)\Big]\, x_{e,b}, \tag{2}$$

$$\text{s.t.} \quad \sum_{e} \sum_{b \in \mathcal{B}} b \cdot x_{e,b} = n \cdot k, \qquad \sum_{b \in \mathcal{B}} x_{e,b} = 1, \quad \forall e, \tag{3}$$

where n is the number of experts and k is the target average bit-width. Each term $\bar{f}_M(e) \cdot \delta_M(e,b)$ couples *how critical* expert e is to modality M with *how much* it degrades that modality at bit-width b, and summing over $M \in \{T, V\}$ yields a modality-balanced cost. Minimizing this cost under the budget constraint naturally drives the solver to assign higher precision to experts that are both frequently routed and highly sensitive, while compressing the rest aggressively.

5 Experiments

In this section, we first describe the experimental setup, then present comprehensive results demonstrating MODE outperforms existing methods across different low-bit quantization settings when applied to MoE-MLLMs. Finally, we show MODE is orthogonal to the recently popular rotation-based quantization techniques and can be combined with them to further reduce quantization loss.

5.1 Experimental Setup

Models and Benchmarks. We conduct experiments on Qwen3-VL-30B-A3B-Instruct, Kimi-VL-A3B-Instruct, and InternVL3.5-30B-A3B (Wang et al., 2025) to cover diverse MoE-MLLMs architectures and validate the generalizability of our approach. Results for InternVL3.5-30B-A3B are provided in Appendix B due to space constraints.

We evaluate on multiple multimodal benchmarks covering diverse capability dimensions, including ChartQA (Masry et al., 2022), InfoVQA (Mathew et al., 2021), and TextVQA (Singh et al., 2019) for OCR-oriented visual question answering; VizWiz-VQA (Gurari et al., 2018), MME-RealWorld (Zhang et al., 2025a), and GQA (Hudson and Manning, 2019) for real-world visual understanding; MMMU (Yue et al., 2024), MM-Star (Chen et al., 2024b), and MMBench (Liu et al., 2024b) for general multimodal reasoning; and POPE (Li et al., 2023) for object hallucination.

Implementation Details. We perform weight-only quantization and report results at average bit-widths of 3-bit and 2-bit. Since the attention and router modules account for less than 3% of the total LLM parameters yet significantly impact overall accuracy, we fix them at 4-bit and adjust the expert bit-width budget accordingly to maintain the target average bit-width; we provide detailed justification and ablation in Appendix G. In the main experiments, we do not quantize the vision encoder, as its parameter count is small relative to the LLM backbone and has negligible impact on the overall memory footprint, which also aligns with most prior MoE-MLLM quantization work. Nevertheless, we additionally report results with the vision encoder quantized in Appendix F for reference.

After the mixed-precision bit-widths are determined, we apply GPTQ for weight calibration. The calibration set consists of 512 text-image pairs from ShareGPT4V, which is the same set used for collecting expert routing frequencies. We adopt symmetric quantization with a group size of 128. All evaluations are conducted using the open-source `lmms-eval` (Li* et al., 2024) framework.

5.2 Main Results

Baselines. To provide a thorough and convincing comparison, besides the basic uniform quantization method GPTQ (Frantar et al., 2023), we cover three categories of PTQ methods designed for MLLMs, MoE-LLMs, and MoE-MLLMs, respectively:

Table 1: Main quantization results on Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct (Team et al., 2025) across 10 multimodal benchmarks. The best results under each setting are shown in **bold**.

Model	Avg-Bit	Method	Benchmarks										Avg.
			MMMU	MMB	MMStar	MME-R	ChartQA	TextV	InfoV	GQA	VizWiz	POPE	
Qwen3-VL -30B-A3B -Instruct	**BF16**	–	52.56	86.86	60.05	52.48	85.20	83.37	81.46	62.96	71.64	89.92	72.65
	W3	GPTQ	44.00	76.63	45.75	45.29	66.16	76.58	66.23	57.73	62.63	88.79	62.98
		MBQ	45.08	77.24	47.48	45.71	68.16	77.98	68.42	58.44	64.52	88.85	64.19
		Speed-Q	46.11	78.65	49.21	46.46	73.20	78.25	71.37	58.92	65.11	88.82	65.61
		MC-MoE	47.56	80.01	52.01	47.88	79.94	77.93	73.65	57.30	65.55	89.21	67.10
		MoEQuant	46.05	78.21	48.74	46.03	71.40	77.05	70.81	58.72	64.71	88.69	65.04
		VEQ-ME	48.04	80.15	52.58	47.41	81.12	78.58	73.23	58.19	66.02	88.96	67.43
		MODE (ours)	**50.33**	**82.56**	**55.68**	**50.77**	**83.00**	**80.65**	**76.62**	**60.21**	**68.82**	**89.44**	**69.81** (+2.38)
	W2	GPTQ	25.56	22.46	30.48	12.94	31.12	40.98	39.07	36.89	37.55	50.84	32.79
		MBQ	35.71	61.23	36.27	31.63	57.44	59.84	53.19	47.16	48.78	72.37	50.36
		Speed-Q	36.92	63.67	37.55	33.98	59.23	62.63	53.42	46.98	50.13	75.41	51.99
		MC-MoE	39.78	66.74	40.48	40.48	63.12	70.42	56.74	53.10	55.32	80.86	56.70
		MoEQuant	36.98	64.49	38.13	38.31	60.18	61.32	56.07	48.01	48.66	73.53	52.57
		VEQ-ME	41.11	71.68	42.08	38.97	67.16	72.38	59.07	54.89	59.95	85.56	59.29
		MODE (ours)	**44.63**	**75.71**	**45.34**	**44.85**	**73.11**	**75.45**	**66.19**	**56.99**	**63.39**	**87.28**	**63.29** (+4.00)
Kimi-VL -A3B -Instruct	**BF16**	–	52.33	82.99	49.41	44.45	89.44	88.69	83.61	62.72	70.59	87.17	71.14
	W3	GPTQ	45.67	75.04	43.14	37.27	82.04	82.05	69.51	56.05	62.07	86.06	63.89
		MBQ	46.02	76.21	44.88	38.52	83.13	81.73	72.69	56.86	63.91	86.73	65.07
		Speed-Q	45.99	76.55	45.57	39.21	82.97	83.01	74.27	57.33	63.89	86.32	65.51
		MC-MoE	46.67	78.13	46.14	40.02	84.26	85.06	75.53	59.05	64.42	86.53	66.58
		MoEQuant	46.31	76.39	45.22	39.45	83.77	82.93	74.33	56.93	63.56	86.84	65.57
		VEQ-ME	47.00	77.84	46.82	39.97	85.64	84.24	77.24	58.17	66.39	86.72	67.00
		MODE (ours)	**48.51**	**80.84**	**48.66**	**41.59**	**87.26**	**86.24**	**79.87**	**61.11**	**69.42**	**87.10**	**69.06** (+2.06)
	W2	GPTQ	27.33	23.44	23.01	24.23	27.62	48.02	22.42	37.09	25.20	70.29	32.87
		MBQ	31.48	38.21	33.54	29.57	36.77	59.98	45.17	41.79	39.33	74.31	43.02
		Speed-Q	35.33	51.44	36.01	31.98	44.62	62.02	50.42	47.09	49.20	77.29	48.54
		MC-MoE	40.92	72.61	41.52	36.01	66.49	75.15	64.23	55.97	58.26	82.58	59.37
		MoEQuant	37.63	63.16	35.78	33.12	53.24	68.84	55.51	50.96	53.69	79.13	53.11
		VEQ-ME	39.78	72.16	42.13	35.52	62.24	74.0 6	64.94	55.45	57.04	83.09	58.64
		MODE (ours)	**45.11**	**75.23**	**44.11**	**38.71**	**75.20**	**79.38**	**67.75**	**58.30**	**64.52**	**85.46**	**63.38** (+4.01)

- **For MLLMs:** MBQ (Li et al., 2024b) and Speed-Q (Guo et al., 2025), both of which account for inter-modality differences during quantization.
- **For MoE-LLMs:** MC-MoE and MoEQuant (Hu et al., 2025), which are tailored for MoE architectures, where MC-MoE similarly employs mixed-precision quantization across experts.
- **For MoE-MLLMs:** VEQ-ME (Qin et al., 2026), the most recent PTQ method explicitly designed for MoE-MLLMs based on GPTQ.

All baseline results are reproduced using their official code repositories. For a fairer comparison with VEQ-ME, whose originally reported results suffer over 10% average accuracy degradation at 3-bit due to unprotected attention weights, we reproduce all its results with the same protection applied.

Results. Table 1 summarizes the quantization results on two representative MoE-MLLMs across 10 multimodal benchmarks under 3-bit and 2-bit settings. MLLM-oriented methods such as MBQ and Speed-Q consider modality-level differences but do not explicitly model the structural characteristics of MoE, which leads to noticeable performance degradation. This becomes more evident under 2-bit quantization, where the average accuracy drops by over 20%. MoE-targeted methods like MC-MoE mitigate severe collapse through mixed-precision bit allocation across experts, yet still incur a non-trivial quantization loss, as they overlook the modality imbalance in expert usage inherent to MoE-MLLMs. VEQ-ME achieves better results than both categories in most settings, though the improvements remain relatively limited.

Our method jointly addresses the heterogeneous expert contributions inherent to MoE architecture and the inter-modality bias together with intra-visual redundancy characteristic of MLLMs, yielding consistent improvements across both models and bit-width settings. At 3-bit, the average accuracy loss is kept within 2.9% and 2.1% on Qwen3-VL and Kimi-VL respectively, substantially improving practical usability. At 2-bit, our method outperforms the best competing approach by over 4% on average, effectively narrowing the performance gap under extreme low-bit quantization.

Table 2: Effect of combining rotation-based quantization (QuaRot) with our method on Qwen3-VL-30B-A3B-Instruct. "+Rot" denotes applying Hadamard rotation on top of our mixed-precision framework.

Method	Bits	MMMU	VizWiz	TextVQA
Baseline (BF16)	16	52.56	71.64	83.37
Ours	3	50.33	68.82	80.65
Ours (+Rot)	3	**50.81**	**69.17**	**81.50**
Ours	2	44.63	63.39	75.45
Ours (+Rot)	2	**46.49**	**66.02**	**76.94**

5.3 Compatibility with Rotation-based Quantization

Rotation-based quantization, exemplified by QuaRot (Ashkboos et al., 2024), has become a mainstream PTQ technique that applies random Hadamard rotations to suppress outliers in weights and activations, thereby reducing quantization error at its source. Since our method focuses on mixed-precision bit allocation along the expert dimension, it is orthogonal to rotation-based approaches. We therefore examine whether combining QuaRot on top of our method can yield further gains.

As shown in Table 2, integrating QuaRot consistently improves performance on top of our method. The benefit is particularly pronounced under the extreme 2-bit setting, where rotation yields an average improvement of around 1.5%, confirming the two techniques address complementary sources of quantization error and can be effectively combined.

5.4 Robustness to Calibration Set

Our method collects expert activation frequencies under different modalities on a calibration set as one indicator of expert importance. It is therefore necessary to verify that this signal is robust to the choice of calibration data. We run our full pipeline with three different calibration sets—ShareGPT4V, Flickr30k (Young et al., 2014), and LLaVA-Next (Liu et al., 2024a), and evaluate the resulting quantized models on the same downstream benchmarks. As shown in Table 3, switching the calibration set introduces negligible variation in average accuracy, confirming that our method is not sensitive to the specific calibration data used. The underlying reason is that MoE-based MLLMs exhibit similar expert selection preferences across different datasets, as demonstrated in Section 3.2 where key vision tokens converge to a stable set of preferred experts regardless of data composition.

Table 3: Performance of our method at 3-bit on Qwen3-VL-30B-A3B-Instruct with different calibration sets.

Calib. Set	MMMU	VizWiz	InfoVQA	MMStar	ChartQA	Avg.
Baseline (BF16)	52.56	71.64	81.46	60.05	85.20	70.18
ShareGPT4V	50.33	68.82	76.62	55.68	83.00	66.89
Flickr30k	49.61	68.27	77.04	54.75	82.54	66.44
LLaVA-Next	50.78	68.91	76.56	56.02	83.36	67.13

6 Deployment Efficiency

For real quantization, we use BitBLAS (Wang et al., 2024) to store quantized weights at varying bit-widths and perform mixed-precision BLAS operations on GPUs. To demonstrate efficiency, we take Qwen3-VL-30B-A3B-Instruct quantized to an average of 3 bits as an example. Figure 6 reports its total weight memory, activation memory, and average accuracy across ten benchmarks, along with two dense MLLMs as references: LLaVA-OneVision-7B (Li et al., 2024a), which has a comparable total weight memory, and Qwen3-VL-2B-Instruct, which has a comparable activation memory.

After 3-bit quantization, the total weight memory drops from 62 GB to around 14 GB, fitting comfortably within a single consumer-grade GPU such as RTX 4090 (24 GB). Compared with LLaVA-OneVision-7B at a similar total weight memory, the quantized model requires far less activation memory thanks to the sparse activation nature of MoE, while achieving higher accuracy. Compared with Qwen3-VL-2B-Instruct at a comparable activation memory, the quantized model surpasses it by a large margin in accuracy. These results confirm that our method enables practical deployment of large-scale MoE-MLLMs on resource-constrained hardware with only a minor performance drop.

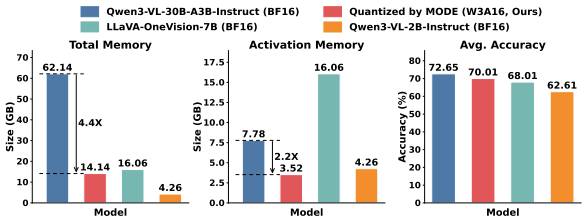

Figure 6: Deployment efficiency of Qwen3-VL-30B-A3B-Instruct quantized by MODE (W3A16) in terms of total / activation memory, and accuracy retention.

Inference speed. Beyond memory footprint, we further examine whether the memory savings translate into practical inference speedups. We randomly sample inputs from ShareGPT4V and measure the end-to-end (E2E) latency together with the prefill and decode throughput on Qwen3-VL-30B-A3B-Instruct, under batch sizes of 1, 2, and

Table 4: End-to-end (E2E) latency and prefill/decode throughput on Qwen3-VL-30B-A3B-Instruct with a maximum output length of 64. BF16 is deployed on two GPUs via Hugging Face Transformers due to its memory requirement, while MODE-W3A16/W2A16 runs on a single GPU via BitBLAS. Gray numbers denote the speedup over the BF16 baseline at the same batch size (BS).

Precision	BS	E2E (s)	Output (tok/s)	Prefill (tok/s)	Decode (tok/s)
BF16	1	10.07	6.36	34.37	16.27
	2	18.65	6.87	37.31	28.73
	4	34.67	7.39	36.67	55.93
W3A16	1	7.82 1.29×	8.19 1.29×	42.96 1.25×	22.45 1.38×
	2	14.55 1.28×	8.80 1.28×	47.38 1.27×	40.51 1.41×
	4	27.28 1.27×	9.39 1.27×	46.94 1.28×	79.42 1.42×
W2A16	1	7.35 1.37×	8.71 1.37×	45.71 1.33×	23.92 1.47×
	2	13.75 1.36×	9.31 1.36×	50.37 1.35×	42.81 1.49×
	4	25.80 1.34×	9.92 1.34×	50.24 1.37×	86.13 1.54×

4 with a maximum output length of 64. Since the BF16 model exceeds the memory capacity of a single GPU, it is deployed with Hugging Face Transformers across two GPUs, whereas MODE-W3A16 and MODE-W2A16 run with BitBLAS on a single GPU. As reported in Table 4, MODE consistently reduces E2E latency and improves both prefill and decode throughput across all batch sizes. At the 3-bit setting, MODE achieves up to a $1.29\times$ E2E speedup and a $1.42\times$ decode throughput gain, and the 2-bit setting further pushes these to $1.37\times$ and $1.54\times$, respectively. Notably, these gains are obtained while using only a single GPU rather than the two GPUs required by BF16, confirming that the memory savings of MODE directly lead to tangible deployment speedups on resource-constrained hardware.

7 Conclusion

We propose MODE, a mixed-precision expert quantization method for MoE-MLLMs. Unlike prior work allocating bit-widths based solely on expert activation statistics, MODE explicitly accounts for both the *inter-modal bias* between textual and visual tokens and the *intra-visual-modal bias* across different visual semantics, enabling a more faithful characterization of expert importance in the multimodal setting. Under a fixed bit budget, MODE efficiently allocates precision across experts, substantially mitigating the performance degradation of weight-only quantization under low-bit regimes. Extensive experiments demonstrate that MODE consistently outperforms existing baselines, narrowing the gap toward practical deployment of MoE-MLLMs on resource-constrained devices.

Limitations

While our work demonstrates the effectiveness of MODE for quantizing MoE-MLLMs, we acknowledge the following limitations and directions for future improvement.

Scaling to larger MoE-MLLMs. Our experiments are conducted on Qwen3-VL-30B-A3B-Instruct, Kimi-VL-A3B-Instruct, and InternVL3.5-30B-A3B, which are deliberately chosen to cover a diverse set of model architectures and thereby validate the generalization of our method. However, due to computational resource constraints, we have not yet verified MODE on larger-scale MoE-MLLMs such as Qwen3-VL-235B-A22B. We leave the evaluation on such larger models to future work when more computational resources become available.

Quantization of the vision module. As discussed in Appendix F, our method is primarily designed for the LLM part of MoE-MLLMs and does not incorporate a dedicated design for the vision module. Although the vision module occupies only a small fraction of the total memory footprint and thus contributes marginally to the deployment cost, we leave a thorough investigation of vision-module quantization in MoE-MLLMs to future work in pursuit of a more comprehensive PTQ design.

Deployment efficiency in modern inference frameworks. As reported in Section 6, for real quantization we leverage BitBLAS (Wang et al., 2024) to store quantized weights at varying bit-widths and perform mixed-precision BLAS operations on GPUs, which achieves significant reduc-

tions in memory consumption and, as validated on the Hugging Face Transformers backbone, translates into consistent end-to-end latency reductions and prefill/decode throughput improvements over the full-precision baseline. Nevertheless, we honestly note that these speedups are obtained relative to a Transformers-based baseline, and our current implementation has not yet been adapted to highly optimized inference frameworks such as vLLM (Kwon et al., 2023). This is mainly because vLLM and similar frameworks incorporate extensive system-level optimizations tailored to MoE architectures, including efficient KV cache management and fused kernels for MoE layers, while currently vLLM does not natively support integrating mixed-precision quantization into its runtime. As a result, the absolute inference speed of the quantized model still lags behind full-precision models served by such frameworks. We hope to integrate our mixed-precision expert quantization scheme into high-performance inference frameworks like vLLM in the future, so as to translate the memory savings into larger practical speedups.

Ethical Considerations

This work focuses on post-training quantization for MoE-based Multimodal Large Language Models, which is a model compression technique and does not involve the collection of new data or human subjects. All models and benchmarks used in our experiments are publicly available and employed in accordance with their respective licenses and intended research use. Since our method reduces the memory and hardware requirements for deploying MoE-MLLMs, it may contribute positively by lowering the barrier to accessing such models and reducing energy consumption during inference.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62572471, 62531026, 62437001), the Zhongguancun Academy (Grant No. C20250505), the Natural Science Foundation of Jiangsu Province under Grant BK20243051.

References

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giridharan Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, and 5 others. 2022. Efficient large scale language modeling with mixtures of experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11699–11732, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. In *Advances in Neural Information Processing Systems*, volume 37, pages 100213–100240. Curran Associates, Inc.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. *Preprint*, arXiv:2403.06764.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024b. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023. Sharegpt4v: Improving large multimodal models with better captions. *arXiv preprint arXiv:2311.12793*.

Haojie Duanmu, Xiuhong Li, Zhihang Yuan, Size Zheng, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. 2025. Mxmoe: Mixed-precision quantization for moe with accuracy and performance co-design. *Preprint*, arXiv:2505.05799.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. *Preprint*, arXiv:2210.17323.

Tianyu Guo, Shanwei Zhao, Shiai Zhu, and Chenguang Ma. 2025. Speed-q: Staged processing with enhanced distillation towards efficient low-bit on-device vlm quantization. *Preprint*, arXiv:2511.08914.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617.

Xing Hu, Zhixuan Chen, Dawei Yang, Zukang Xu, Chen Xu, Zhihang Yuan, Sifan Zhou, and Jiangyong

Yu. 2025. Moequant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance. *Preprint*, arXiv:2505.03804.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. 2025. Mixture compressor for mixture-of-experts llms gains more. *Preprint*, arXiv:2410.06270.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Bo Li*, Peiyuan Zhang*, Kaichen Zhang*, Fanyi Pu*, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. 2024. Lmms-eval: Accelerating the development of large multimoal models.

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

Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. 2024b. Mbq: Modality-balanced quantization for large vision-language models. *Preprint*, arXiv:2412.19509.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. 2024. Moe-llava: Mixture of experts for large vision-language models. *Preprint*, arXiv:2401.15947.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024b. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. 2021. Infographicvqa. *CoRR*, abs/2104.12756.

Guangshuo Qin, Zhiteng Li, Zheng Chen, Weihang Zhang, Linghe Kong, and Yulun Zhang. 2026. Veq: Modality-adaptive quantization for moe vision-language models. *Preprint*, arXiv:2602.01037.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. 2025. Tokencarve: Information-preserving visual token compression in multimodal large language models. *Preprint*, arXiv:2503.10501.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, and 73 others. 2025. Kimi-VL technical report. *Preprint*, arXiv:2504.07491.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, and 74 others. 2026. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *Preprint*, arXiv:2507.01006.

Lei Wang, Lingxiao Ma, Shijie Cao, Quanlu Zhang, Jilong Xue, Yining Shi, Ningxin Zheng, Ziming Miao, Fan Yang, Ting Cao, Yuqing Yang, and Mao Yang. 2024. Ladder: Enabling efficient low-precision deep learning computing through hardware-aware tensor

transformation. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pages 307–323, Santa Clara, CA. USENIX Association.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2026. Visionzip: Longer is better but not necessary in vision language models. *Preprint*, arXiv:2412.04467.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. 2025a. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *Preprint*, arXiv:2408.13257.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2025b. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*.

Zihao Zheng, Xiuping Cui, Size Zheng, Maoliang Li, Jiayu Chen, Yun Liang, and Xiang Chen. 2026. Dynamo: Runtime switchable quantization for moe with cross-dataset adaptation. *Preprint*, arXiv:2503.21135.

A Complete Per-Layer Heatmap of Intra-Vision Expert Frequency Bias

In Section 3.2, due to space constraints, Figure 3(b) only visualizes $\bar{f}_{\text{key}} - \bar{f}_{\text{red}}$ for the first 16 layers and the first 48 experts. However, Qwen3-VL-30B-A3B-Instruct contains 48 layers in total, with 128 experts per layer. For completeness, we provide the full heatmap covering all layers and all experts in Figure 7.

On the magnitude of frequency differences. A natural question is why the absolute values of $\bar{f}_{\text{key}} - \bar{f}_{\text{red}}$ in Figure 7 (and in Figure 3(b)) appear noticeably smaller than the routing frequencies reported in Figure 2 of Section 3.1, where the maximum value can exceed 25%. The discrepancy stems from a difference in normalization rather than a difference in the underlying routing behavior.

In Qwen3-VL-30B-A3B-Instruct, each MoE layer contains 128 experts, and each token activates the top-8 experts. Consequently, even under a perfectly uniform routing distribution, the expected selection frequency of any single expert is

$$\frac{8}{128} = 6.25\%.$$

The frequencies reported in Figure 2 are *unnormalized* selection frequencies, so values around or above this 6.25% baseline are expected, and peaks beyond 25% indicate strong routing preferences.

In contrast, the values in Figure 3(b) and Figure 7 are computed from *normalized* per-token routing distributions, where the activated experts of each token sum to 1. Under this normalization, the expected selection frequency of any single expert under uniform routing is

$$\frac{1}{128} \approx 0.78\%.$$

Relative to this baseline, a difference of $\bar{f}_{\text{key}} - \bar{f}_{\text{red}}$ on the order of 8% already represents an obviously large routing-frequency gap—roughly an order of magnitude above the uniform baseline—indicating a pronounced specialization of certain experts toward dominant (key) versus redundant tokens.

B Additional Results on InternVL3.5-30B-A3B

In the main paper, we report results on Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct. Due to space constraints, we defer the results on InternVL3.5-30B-A3B to this appendix. We follow the exact same evaluation protocol as in the main experiments, evaluating on the same 10 multimodal benchmarks under both average 3-bit and 2-bit settings. As shown in Table 5, our proposed MODE consistently outperforms both the mixed-precision MoE quantization method MC-MoE and the MoE-MLLM-tailored method VEQ-MA at both bit-widths. In particular, under the 3-bit setting,

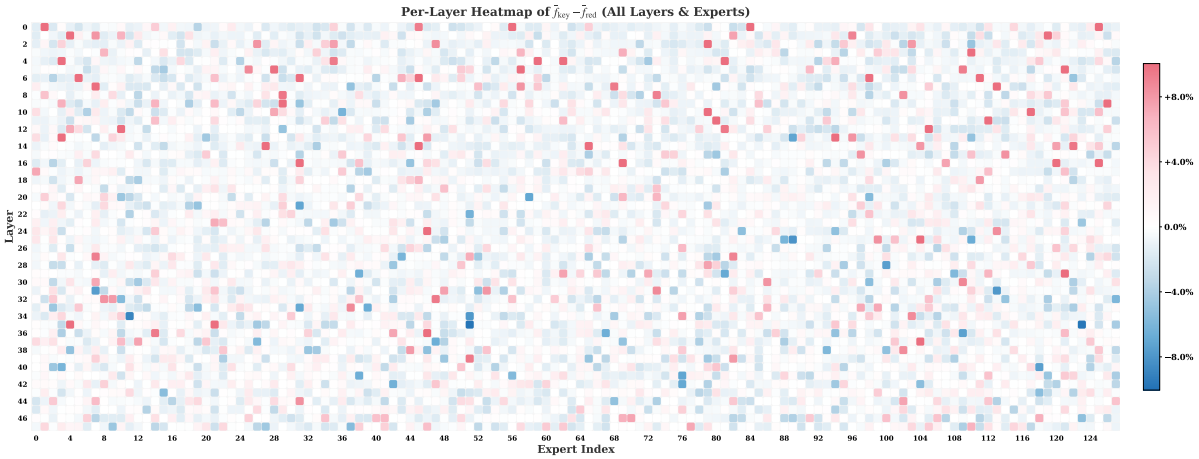

Figure 7: Complete per-layer heatmap of $\bar{f}_{\text{key}} - \bar{f}_{\text{red}}$ on Qwen3-VL-30B-A3B-Instruct, covering all 48 layers and all 128 experts per layer.

MODE limits the average accuracy degradation to within 3% relative to the BF16 baseline (69.25 vs. 72.15), demonstrating the strong generalization of our method across different model families. Under the more challenging 2-bit setting, MODE still surpasses the strongest baseline VEQ-MA by about 2 points on average, further confirming its robustness under extremely low-bit quantization.

C Details of Key Vision Token Selection

In this appendix, we provide a detailed description of the key vision token selection used in our intra-vision expert frequency bias analysis. We first review the attention-based token importance criterion proposed by SparseVLM (Zhang et al., 2025b) and discuss why it is well-motivated for identifying redundant vision tokens (Appendix C.1). We then describe how we adapt this criterion to our calibration setting, with a key difference from prior pruning-oriented usage (Appendix C.2). Finally, we empirically verify that the tokens selected by our procedure indeed behave as "key" and "redundant" vision tokens (Appendix C.3).

C.1 Recap of SparseVLM's Attention-based Criterion

A growing body of work has observed that vision tokens in VLMs/MLLMs are highly redundant: a large fraction of them can be pruned with negligible degradation in downstream performance, while substantially reducing the computation and memory cost of inference (Chen et al., 2024a; Yang et al., 2026; Tan et al., 2025). Among these methods, SparseVLM (Zhang et al., 2025b) stands out as a representative approach. Although many follow-up works have further improved upon it, SparseVLM remains attractive for its simplicity and effectiveness, requiring no additional modules or training. We therefore follow its core criterion in this work and adapt it to our needs.

The key idea of SparseVLM is to reuse the self-attention logits already computed inside the VLM decoder as a measure of how relevant each vision token is to the current text query. Concretely, let $\boldsymbol{A} \in \mathbb{R}^{L\times L}$ denote the self-attention matrix at a given layer, where L is the total sequence length. Let $\mathbb{L}$ and $\mathbb{I}$ denote the index sets of language (text) tokens and image (vision) tokens, respectively, with $|\mathbb{L}| = L_t$ and $|\mathbb{I}| = L_v$. SparseVLM extracts the text-to-vision sub-block

$$\boldsymbol{P} = \boldsymbol{A}_{i,j}, \quad (i,j) \in \mathbb{L} \times \mathbb{I}, \qquad \boldsymbol{P} \in \mathbb{R}^{L_t \times L_v}, \tag{4}$$

and aggregates the attention each vision token receives from all text tokens:

$$\tilde{\boldsymbol{p}} = [\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_{L_v}] = \frac{1}{L_t}\sum_{i=1}^{L_t} \boldsymbol{P}_i. \tag{5}$$

A larger value of $\tilde{p}_j$ indicates that vision token j is more relevant to the current language query and thus more important to retain. Because $\boldsymbol{A}$ is already computed during the forward pass, this estimation is essentially free in terms of additional FLOPs.

C.2 Our Adaptation: Per-Layer Adaptive Selection

The most important difference between our usage and the original pruning-oriented usage in SparseVLM lies in *when* and *how* the selection is applied. In SparseVLM and most subsequent token-pruning

Table 5: Quantization results on InternVL3.5-30B-A3B across 10 multimodal benchmarks. The best results under each setting are shown in **bold**. The number in parentheses denotes the absolute average gain of MODE over the strongest baseline.

Model	Avg-Bit	Method	Benchmarks										Avg.
			MMMU	MMB	MMStar	MME-R	ChartQA	TextV	InfoV	GQA	VizWiz	POPE	
InternVL3.5 -30B-A3B	**BF16**	–	60.11	86.08	71.25	49.71	88.12	78.35	76.27	62.95	59.23	89.38	72.15
	W3	GPTQ	52.56	81.19	63.19	44.17	77.60	70.34	65.44	56.01	45.23	89.13	64.49
		MC-MoE	54.56	81.42	66.57	47.52	82.00	72.67	67.68	57.38	50.28	88.57	66.87
		VEQ-MA	55.49	81.12	67.01	46.05	83.07	73.13	69.75	59.21	53.04	89.08	67.70
		MODE (ours)	**57.33**	**82.56**	**68.26**	**47.97**	**85.64**	**74.45**	**71.53**	**60.15**	**55.55**	**89.03**	**69.25** (+1.55)
	W2	GPTQ	22.78	77.32	31.67	21.68	40.75	36.11	25.62	30.30	30.11	40.18	35.65
		MC-MoE	44.89	76.43	57.40	43.79	70.92	67.94	60.10	55.63	44.07	88.18	60.94
		VEQ-MA	46.11	75.76	59.76	42.33	72.36	67.55	61.68	56.04	48.72	88.46	61.88
		MODE (ours)	**48.67**	**78.69**	**60.97**	**45.31**	**73.68**	**69.83**	**63.65**	**58.40**	**50.87**	**88.66**	**63.87** (+2.00)

methods, the decision must be made at a relatively shallow layer (e.g., the second or third decoder layer): tokens deemed redundant are physically dropped and excluded from all subsequent layers' computation, which is precisely how the speedup is achieved. However, a well-known issue with this design is that the attention distribution over vision tokens varies considerably across depths: a vision token that receives little attention in shallow layers may become highly attended in deeper layers, and vice versa. Committing to a one-shot pruning decision at a shallow layer therefore inevitably introduces selection bias.

In our pipeline, by contrast, the selection is used purely for *calibration*, not for actual pruning during inference: we only need to identify which vision tokens behave as key tokens at each layer, in order to compute the modality-wise expert selection frequency. This frees us from the constraint of making an early, layer-agnostic decision. Accordingly, we apply the SparseVLM criterion *independently at every layer*: at layer ℓ, we compute $\tilde{\boldsymbol{p}}^{(\ell)}$ from that layer's attention, rank vision tokens by $\tilde{\boldsymbol{p}}^{(\ell)}$, and treat the top 20% as key vision tokens and the remaining 80% as redundant vision tokens. The resulting key-token set is thus adaptive to how the model attends to visual content at different depths, avoiding the cross-layer bias that hampers shallow-layer pruning.

C.3 Empirical Verification of the Selected Key and Redundant Tokens

To verify that our per-layer attention-based selection indeed identifies vision tokens that are functionally "key" or "redundant" as intended, we conduct a controlled pruning experiment on Qwen3-VL-30B-A3B-Instruct using ChartQA.

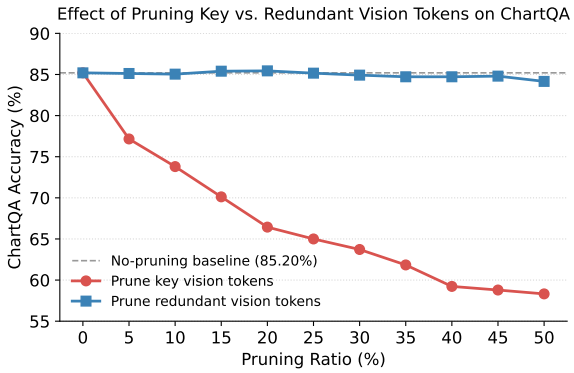

Figure 8: Effect of pruning key vs. redundant vision tokens on ChartQA. Removing tokens identified as *key* (red) causes a rapid accuracy drop, whereas removing tokens identified as *redundant* (blue) leaves accuracy almost unchanged across the entire 5%–50% range.

Protocol. At every decoder layer, we rank vision tokens by their importance score $\tilde{\boldsymbol{p}}^{(\ell)}$ as described in Appendix C.2. We then prune vision tokens at increasing ratios from 5% to 50% in steps of 5%, under two opposite strategies: (i) *key-first pruning*, which removes the highest-ranked tokens first, and (ii) *redundant-first pruning*, which removes the lowest-ranked tokens first. We restrict the pruning ratio to at most 50%, making the two strategies cleanly comparable. For each pruning ratio, we evaluate the resulting model on ChartQA and plot accuracy versus pruning ratio as two curves (key-first vs. redundant-first).

Results. As shown in Figure 8, the two strategies exhibit sharply contrasting trends. Under key-first pruning, accuracy degrades sharply even at very small pruning ratios—dropping from the no-pruning baseline of 85.20% to roughly 66% after pruning only 20% of the vision tokens—which indicates that the tokens we identify as "key" indeed

carry information that the model genuinely relies on. As the ratio further increases, the curve flattens and decreases more gradually, since the additionally removed tokens are progressively less critical. In contrast, under redundant-first pruning, accuracy stays essentially flat throughout the entire range, fluctuating only within roughly 84%–85% and remaining close to the baseline even when half of the vision tokens are discarded. This pronounced asymmetry confirms that our per-layer adaptive selection reliably separates key vision tokens from redundant ones, providing a solid basis for the subsequent analysis that links key vision tokens to critical vision experts.

D Details of Expert Selection Frequency Collection and Cross-Dataset Similarity

In this appendix, we formally describe how expert selection frequencies are collected, normalized, and compared across calibration sets. We first introduce the general collection procedure without distinguishing token types (Appendix D.1), then specialize it to text, key vision, and redundant vision tokens with layer-wise normalization (Appendix D.2), and finally describe the cross-dataset cosine similarity (Appendix D.3).

D.1 General Expert Selection Frequency

Consider an MoE model with L MoE layers and E experts per layer. For each token, the router at every layer activates a top-k subset of experts. Given a calibration dataset $\mathcal{D}$, we run the model on it and, at each layer ℓ, simply count how many tokens activate each expert e:

$$c_e^{(\ell)} \;=\; \#\{\, x \in \mathcal{D} : x \text{ selects expert } e \text{ at layer } \ell \,\}. \tag{6}$$

Stacking over $e = 1, \ldots, E$ gives a per-layer count vector $\boldsymbol{c}^{(\ell)} \in \mathbb{R}^E$.

D.2 Per-Token-Type Frequency and Layer-wise Normalization

We further partition tokens by their role: text tokens (*text*), key vision tokens (*key*, top-20% identified per layer as in Appendix C.2), and redundant vision tokens (*red*, the remaining 80%). For each token type $m \in \{\textit{text}, \textit{key}, \textit{red}\}$, we restrict the counting in Eq. (6) to tokens of type m only, yielding a type-specific count $c_{m,e}^{(\ell)}$.

To make frequencies comparable across layers and across calibration sets of different sizes, we normalize $\boldsymbol{c}_m^{(\ell)}$ at each layer into a probability distribution over experts:

$$\bar{f}_{m,e}^{(\ell)} \;=\; \frac{c_{m,e}^{(\ell)}}{\sum_{e'=1}^{E} c_{m,e'}^{(\ell)}}, \tag{7}$$

so that $\sum_e \bar{f}_{m,e}^{(\ell)} = 1$. Stacking over experts yields the normalized frequency vector $\bar{\boldsymbol{f}}_m^{(\ell)} \in \mathbb{R}^E$.

Notation across sections. The frequency symbols used in the main text are all special cases of $\bar{\boldsymbol{f}}_m^{(\ell)}$, but with slightly different scopes depending on context:

- In Sec. 3.3, four frequencies appear: $\bar{f}_{\textit{total}}$, $\bar{f}_{\textit{text}}$, $\bar{f}_{\textit{key}}$, and $\bar{f}_{\textit{red}}$. Among them, $\bar{f}_{\textit{total}}$ is the overall normalized frequency obtained by counting all tokens (i.e., applying Eq. (7) to the full token set without partitioning), while $\bar{f}_{\textit{text}}$, $\bar{f}_{\textit{key}}$, and $\bar{f}_{\textit{red}}$ are the normalized frequencies computed on the corresponding token subsets.
- In Sec. 4, since our analysis has already focused on text tokens and key vision tokens, we adopt the more compact notation $\bar{f}_t$ and $\bar{f}_v$ to denote the normalized frequencies of the two modalities. Specifically, $\bar{f}_t$ denotes the frequency computed on text tokens (identical to $\bar{f}_{\textit{text}}$ in Sec. 3.3), and $\bar{f}_v$ denotes the frequency computed on key vision tokens only (identical to $\bar{f}_{\textit{key}}$); the redundant vision tokens are no longer involved.

D.3 Cross-Dataset Cosine Similarity

To measure how similar the routing behavior is between two calibration sets $\mathcal{D}_a$ and $\mathcal{D}_b$, we use the normalized frequency vectors above. For a given token type m, we first compute the cosine similarity at every layer:

$$s_m^{(\ell)} \;=\; \cos\!\big(\bar{\boldsymbol{f}}_m^{(\ell)}(\mathcal{D}_a),\; \bar{\boldsymbol{f}}_m^{(\ell)}(\mathcal{D}_b)\big), \tag{8}$$

and then average across all L MoE layers to obtain the final similarity:

$$S_m(\mathcal{D}_a, \mathcal{D}_b) \;=\; \frac{1}{L}\sum_{\ell=1}^{L} s_m^{(\ell)}. \tag{9}$$

Since each $\bar{\boldsymbol{f}}_m^{(\ell)}$ is a probability distribution with non-negative entries, every per-layer similarity satisfies $s_m^{(\ell)} \in [0, 1]$, and consequently $S_m(\mathcal{D}_a, \mathcal{D}_b) \in [0, 1]$. A value closer to 1 indicates that, for token type m, the two datasets induce more consistent expert selection patterns throughout the network, supporting the cross-dataset transferability claim in the main text.

E Performance on Language-Only Tasks

The core of our method is to jointly consider the cross-modal expert importance bias between language and vision modalities and the intra-vision expert importance bias between key and redundant vision tokens. In the main experiments, we primarily focus on evaluating quantized MoE-MLLMs on multimodal benchmarks, where our method significantly outperforms other PTQ methods designed for MLLMs, MoE-LLMs, or MoE-MLLMs. Meanwhile, it is important to ensure that our quantization does not severely degrade the language-only capabilities of MoE-MLLMs.

To this end, we evaluate our method on Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct at both 2-bit and 3-bit average bit-widths across seven widely used language-only benchmarks: ARC-Easy, ARC-Challenge, HellaSwag, LAMBADA, MMLU, PIQA, and WinoGrande. Results are reported in Table 6.

As shown in Table 6, at 3-bit average bit-width, the average performance degradation is limited to approximately 3% on Qwen3-VL-30B-A3B-Instruct and 2% on Kimi-VL-A3B-Instruct. These results confirm that our modality-aware mixed-precision quantization method effectively preserves the language-only capabilities of MoE-MLLMs while substantially improving multimodal performance as demonstrated in the main experiments.

F Quantizing Both Vision Module and LLM

In the main experiments, we focus on quantizing the LLM part of MoE-MLLMs while keeping the vision module (i.e., the vision encoder and the multi-modal projector, plus the merger module for Qwen3-VL-30B-A3B-Instruct) at original precision. We adopt this setting because the vision module accounts for only a negligible fraction of the total weight memory. As reported in Table 7, the vision module occupies merely around 1 GB of memory on both Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct, which is less than 3% of the total model size. Therefore, keeping the vision module at original precision does not noticeably affect the overall memory footprint. Furthermore, our method specifically targets the mixed-precision quantization of MoE experts in MoE-MLLMs and is orthogonal to methods that quantize the vision module, such as Speed-Q.

Nevertheless, to provide more comprehensive results, we further evaluate the setting where both the vision module and the LLM part are quantized. For the LLM part, we apply our mixed-precision expert quantization method as in the main experiments, with average bit-widths of 3-bit and 2-bit. For the vision module, we simply apply GPTQ to quantize it to 4-bit. Evaluation is conducted on the same 10 multimodal benchmarks as in the main experiments, and results are summarized in Table 8. In the table, the column "Vision Q." indicates whether the vision module is additionally quantized to 4-bit (✓) or kept at original precision (×).

As shown in Table 8, on Qwen3-VL-30B-A3B-Instruct, quantizing the vision module to 4-bit yields an additional memory saving of approximately $1.077 \times 3/4 \approx 0.81$ GB, while incurring about 1.7% and 1.5% relative average accuracy drops under the 3-bit and 2-bit LLM settings, respectively. On Kimi-VL-A3B-Instruct, the joint quantization is fairly robust under the 3-bit setting with only a 0.5% relative drop, but the degradation becomes more pronounced under the 2-bit setting, exceeding 2.5%. Although our method is not specifically designed for the vision module, we hope these results can serve as a useful reference for future works on jointly quantizing both modalities of MoE-MLLMs.

G The Importance of Protecting Attention Module at 4-bit

In our experiments, we observe that for MoE-MLLMs, although the attention module accounts for only a small fraction of the total parameters, it is critical to the overall performance, especially under extremely low-bit quantization. For instance, under 2-bit quantization, quantizing the MoE layers to 2-bit causes noticeable accuracy degradation but does not lead to model collapse, whereas directly quantizing the attention module to 2-bit causes the model to collapse entirely. In contrast, keeping the attention module at 4-bit substantially mitigates the quantization loss. We attribute this to the fact that in MoE-MLLMs, experts in the MoE layers are sparsely activated, while the attention module operates on every token, making it far more sensitive to quantization noise. While we do not regard this as a novel contribution, we find it highly valuable for practical implementation and deployment, and therefore discuss it in detail here.

Parameter footprint of the attention module. We first show that the attention module occupies

Table 6: Performance on language-only benchmarks. Our method preserves language capabilities with limited degradation at both bit-widths. Qwen3-VL-30B-A3B and Kimi-VL-A3B are short for Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct, respectively. ARC-E and ARC-C denote ARC-Easy and ARC-Challenge.

Qwen3-VL-30B-A3B	ARC-E	ARC-C	HellaSwag	LAMBADA	MMLU	PIQA	WinoGrande	Avg
Baseline (FP16)	80.56	54.69	62.03	71.49	80.75	80.25	74.59	70.63
W3A16	79.92	52.47	58.67	68.45	75.89	78.84	71.01	67.89
W2A16	77.48	49.23	53.12	62.32	65.38	75.57	66.48	64.23
Kimi-VL-A3B	ARC-E	ARC-C	HellaSwag	LAMBADA	MMLU	PIQA	WinoGrande	Avg
Baseline (FP16)	82.62	53.92	59.57	71.61	69.01	80.20	71.59	69.79
W3A16	80.72	52.13	56.56	70.95	64.71	78.89	70.32	67.75
W2A16	75.67	42.92	48.14	63.09	57.46	76.39	64.64	61.19

Table 7: Weight memory footprint of the vision module versus the LLM part on Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct. The vision module accounts for less than 3% of the total weight memory on both models.

Model	Vision (GB)	Vision (%)	LLM (GB)	LLM (%)	Total (GB)
Qwen3-VL-30B-A3B-Instruct	1.077	1.73%	61.064	98.27%	62.142
Kimi-VL-A3B-Instruct	0.895	2.73%	31.920	97.27%	32.815

only a small fraction of the LLM parameters. As reported in Table 9, the attention module accounts for less than 3% of the LLM weight memory on both Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct, while the MoE layers dominate the parameter budget at over 93%.

Effect of protecting attention at 4-bit. To empirically verify the importance of protecting the attention module, we use GPTQ as a representative quantization method and compare two settings on Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct: (i) **GPTQ-W2**, where all components of the LLM (including both the attention module and the MoE layers) are quantized to 2-bit; and (ii) **GPTQ-W2-Attn4**, where only the MoE layers are quantized to 2-bit while the attention module is kept at 4-bit. Both settings are evaluated on the same 10 multimodal benchmarks as in the main experiments, and the results are summarized in Table 10. GPTQ-W2 denotes uniform 2-bit quantization applied to both attention and MoE layers, whereas GPTQ-W2-Attn4 keeps attention at 4-bit precision while quantizing the MoE layers to 2-bit.

As shown in Table 10, under the pure 2-bit setting, Qwen3-VL-30B-A3B-Instruct exhibits clear performance collapse on multiple benchmarks including ChartQA, MME-R, TextVQA, and InfoVQA, with accuracy on MME-R dropping to as low as 12.94. Once the attention module is protected at 4-bit, the performance is largely recovered, with the average accuracy improving from 34.99 to 50.88 (a relative gain of over 45%). A similar trend is observed on Kimi-VL-A3B-Instruct, where the average accuracy increases from 32.87 to 46.90. These results empirically confirm that protecting the attention module at 4-bit is critical under extremely low-bit quantization.

Moreover, the additional memory cost of protecting the attention module is marginal. Taking Qwen3-VL-30B-A3B-Instruct as an example, the extra memory introduced by promoting attention from 2-bit to 4-bit can be compensated by reducing the bit-width of merely 8 experts (out of 128×48 experts in total) by 1 bit across all layers. Given that the experts are empirically far more robust to quantization than attention, protecting the attention module at 4-bit is a highly cost-effective design choice and is strongly recommended in practice.

H Multi-GPU Parallel GPTQ for MoE-MLLMs

H.1 Motivation and Design

Fine-grained MoE-MLLMs typically contain hundreds of routed experts per layer, each requiring independent GPTQ (Frantar et al., 2023) quantization of its `gate_up_proj` and `down_proj` matrices. Conventional GPTQ pipelines traverse these experts strictly one after another within each MoE layer, and as the number of experts grows, the cumulative cost of this per-expert sequential traversal

Table 8: Performance comparison between LLM-only quantization and joint quantization of both the vision module and the LLM. The "Vision Q." column indicates whether the vision module is quantized to 4-bit using GPTQ (✓) or kept at original precision (×). Qwen3-VL-30B-A3B and Kimi-VL-A3B are short for Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct, respectively.

Qwen3-VL-30B-A3B	**Vision Q.**	ChartQA	MMBench	MME-R	MMMU	MMStar	POPE	GQA	TextVQA	InfoVQA	VizWiz	Avg
Baseline (BF16)	×	85.20	86.86	52.48	52.56	60.05	89.92	62.96	83.37	81.46	71.64	72.65
MODE-W3	×	83.00	82.56	51.77	50.33	55.68	89.44	60.21	80.65	77.62	68.82	70.01
MODE-W3	✓	81.92	80.59	48.40	46.56	52.41	89.04	59.47	80.09	77.13	68.00	68.36
MODE-W2	×	73.11	75.71	44.85	44.63	45.34	87.28	56.99	75.45	66.19	63.39	63.29
MODE-W2	✓	74.59	72.96	41.72	41.52	42.26	86.51	56.33	75.98	66.47	59.34	61.77
Kimi-VL-A3B	**Vision Q.**	ChartQA	MMBench	MME-R	MMMU	MMStar	POPE	GQA	TextVQA	InfoVQA	VizWiz	Avg
Baseline (BF16)	×	89.44	82.99	44.45	52.33	49.41	87.17	62.72	88.69	83.61	70.59	71.14
MODE-W3	×	87.26	80.84	41.59	48.51	48.66	87.10	61.11	86.24	79.87	69.42	69.06
MODE-W3	✓	86.78	80.50	42.16	48.31	46.51	87.37	60.39	85.65	78.80	69.02	68.55
MODE-W2	×	75.20	75.23	38.71	45.11	44.11	85.46	58.30	79.38	67.75	64.52	63.38
MODE-W2	✓	67.04	68.45	38.81	43.77	38.79	85.53	58.45	77.39	65.46	65.09	60.88

Table 9: Weight memory footprint of the attention module versus the MoE layers in the LLM part of Qwen3-VL-30B-A3B-Instruct and Kimi-VL-A3B-Instruct. The attention module accounts for less than 3% of the LLM weight memory on both models, while the MoE layers dominate at over 93%.

Model	**Attention (GB)**	**Attention (%)**	**MoE Layers (GB)**	**MoE Layers (%)**	**LLM Total (GB)**
Qwen3-VL-30B-A3B-Instruct	1.812	2.97%	58.007	94.99%	61.064
Kimi-VL-A3B-Instruct	0.743	2.33%	29.696	93.03%	31.920

makes MoE-layer calibration the dominant bottleneck of the entire quantization procedure on fine-grained MoE models. Since the Hessian construction and weight update for different experts are, however, mutually independent, the expert-quantization stage is in fact an *embarrassingly parallel* workload. We therefore introduce an expert-level multi-GPU scheme that follows a *"serial collection → parallel quantization → serial propagation"* pattern: inter-layer ordering, attention quantization, calibration forward passes, activation/Hessian collection, and the post-quantization re-forward used for layer-wise error propagation all remain strictly serial; only the inner loop over experts within a single quantization stage is dispatched concurrently across GPUs. This scheduling change leaves the GPTQ mathematics, the quantization order, and all inter-layer dependencies untouched, so the output is bit-identical to the serial baseline. Algorithmic equivalence is guaranteed by four invariants: (i) layer order is preserved, so layer $\ell+1$ always sees inputs produced by the fully quantized layer ℓ; (ii) within a layer, `down_proj` quantization strictly follows the re-forward of the quantized `gate_up_proj`, satisfying GPTQ's sequential-quantization requirement; (iii) experts within the same stage are mathematically independent; and (iv) attention submodules and routers are quantized exactly as in the serial baseline.

H.2 Implementation

We take a single projection matrix of a single expert as the minimal task unit, yielding up to $2E$ independent tasks per MoE layer with E experts. Tasks are assigned to G GPUs in a round-robin manner, and each GPU processes its local queue sequentially while the G queues run concurrently; multi-threading is sufficient to drive this parallelism, since CUDA kernels release the Python GIL during execution. Each worker independently transfers the assigned expert weight and its corresponding inputs to its target GPU, performs the GPTQ update locally, and offloads the quantized result back to CPU before releasing its GPU memory. The main thread, residing on a single primary device, retains responsibility for calibration, activation collection, and re-forward propagation between stages, so that quantized expert weights produced in parallel are gathered and reinstalled into the model before the next dependent stage begins. To bound peak memory, expert inputs and Hessians are cloned into per-task structures before dispatch and released immediately afterwards, and explicit cache cleanup and synchronization are performed at the end of each layer. The scheme is implemented as a drop-in replacement for the standard GPTQ entry point and is agnostic to the specific MoE weight organization adopted by different

Table 10: Comparison between fully quantizing the LLM to 2-bit (GPTQ-W2) and protecting the attention module at 4-bit while quantizing the MoE layers to 2-bit (GPTQ-W2-Attn4). Protecting the attention module at 4-bit substantially mitigates the quantization loss on both models.

Qwen3-VL-30B-A3B-Instruct	ChartQA	MMBench	MME-R	MMMU	MMStar	POPE	GQA	TextVQA	InfoVQA	VizWiz	Avg
Baseline (BF16)	85.20	86.86	52.48	52.56	60.05	89.92	62.96	83.37	81.46	71.64	72.65
GPTQ-W2	31.12	22.46	12.94	25.56	30.48	50.84	36.89	40.98	39.07	37.55	34.99
GPTQ-W2-Attn4	58.16	52.68	32.91	36.11	36.08	79.56	45.44	56.38	59.07	52.36	50.88
Kimi-VL-A3B-Instruct	ChartQA	MMBench	MME-R	MMMU	MMStar	POPE	GQA	TextVQA	InfoVQA	VizWiz	Avg
Baseline (BF16)	89.44	82.99	44.45	52.33	49.41	87.17	62.72	88.69	83.61	70.59	71.14
GPTQ-W2	27.62	23.44	24.23	27.33	23.01	70.29	37.09	48.02	22.42	25.20	32.87
GPTQ-W2-Attn4	45.64	50.84	29.71	34.00	32.66	73.10	46.11	64.24	49.27	43.42	46.90

model families.

H.3 Complexity and Speedup

Let E be the number of routed experts per layer, G the number of GPUs, and t the GPTQ time of a single expert projection. The serial baseline costs $\mathcal{O}(E \cdot t)$ per stage, while our scheme reduces this to $\mathcal{O}(\lceil E/G \rceil \cdot t)$, giving an expert-stage speedup of $\min(E, G)$. Denoting the wall-clock time of all serial components by T_s and the serial cost of expert quantization by T_p, the overall speedup can be expected as:

$$\text{Speedup} = \frac{T_s + T_p}{T_s + T_p/G},$$

which approaches G when $T_p \gg T_s$, as is typical for fine-grained MoE-MLLMs with large E. In our experiments, Kimi-VL-A3B-Instruct contains 64 routed experts per layer, while Qwen3-VL-30B-A3B-Instruct and InternVL3.5-30B-A3B each contain 128 experts per layer, placing all three models well within the regime where the parallel speedup approaches G.

I LLM Usage

In the preparation of this manuscript, large language models were used to polish the writing, including grammar refinement, phrasing improvement, and minor stylistic edits.