SPEED-Q: Staged Processing with Enhanced Distillation Towards Efficient Low-Bit On-Device VLM Quantization

Tianyu Guo, Shanwei Zhao, Shiai Zhu*, Chenguang Ma

Ant Group

{guotianyu.gty, shanwei.zsw, shiai.zsa, chenguang.mcg}@antgroup.com

Abstract

Deploying Vision-Language Models (VLMs) on edge devices (e.g., smartphones and robots) is crucial for enabling low-latency and privacy-preserving intelligent applications. Given the resource constraints of these devices, quantization offers a promising solution by improving memory efficiency and reducing bandwidth requirements, thereby facilitating the deployment of VLMs. However, existing research has rarely explored aggressive quantization on VLMs, particularly for the models ranging from 1B to 2B parameters, which are more suitable for resource-constrained edge devices. In this paper, we propose **SPEED-Q**, a novel **S**taged **P**rocessing with **E**nhanc**E**d **D**istillation framework for VLM low-bit weight-only quantization that systematically addresses the following two critical obstacles: (1) significant discrepancies in quantization sensitivity between vision (ViT) and language (LLM) components in VLMs; (2) training instability arising from the reduced numerical precision inherent in low-bit quantization. In SPEED-Q, a staged sensitivity adaptive mechanism is introduced to effectively harmonize performance across different modalities. We further propose a distillation-enhanced quantization strategy to stabilize the training process and reduce data dependence. Together, SPEED-Q enables accurate, stable, and data-efficient quantization of complex VLMs. SPEED-Q is the first framework tailored for quantizing entire small-scale billion-parameter VLMs to low bits. Extensive experiments across multiple benchmarks demonstrate that SPEED-Q achieves up to **6× higher accuracy** than existing quantization methods under 2-bit settings and consistently outperforms prior on-device VLMs under both 2-bit and 4-bit settings.

Code — https://github.com/antgroup/SPEED-Q

1 Introduction

Vision-Language Models (VLMs) have achieved impressive performance across various applications, including visual question answering, robot navigation, and so on (Li et al. 2025b). However, the large size of VLMs presents significant challenges for deployment, particularly on resource-constrained edge devices. To facilitate on-device inference, a series of lightweight VLMs such as SmolVLM (Marafioti et al. 2025), FastVLM (Vasu et al. 2025), and H2OVL (Galib et al. 2024) have been proposed, aiming to reduce storage and memory costs. However, these models typically experience accuracy degradation due to the effects of scaling laws (Kaplan et al. 2020).

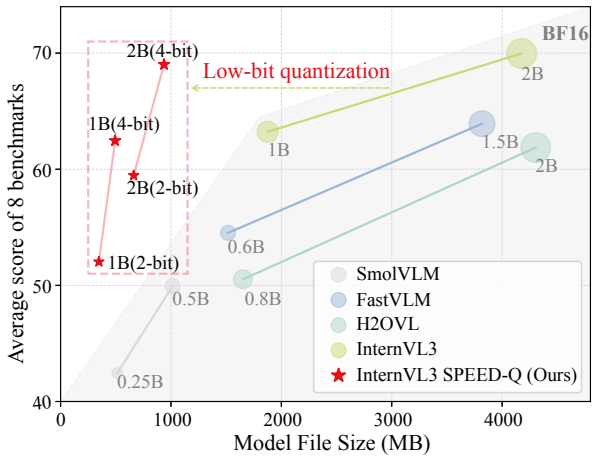

Figure 1: Comparison with SOTA on-device VLMs. The proposed quantized version of InternVL3 outperforms the existing on-device VLMs with a smaller model file size.

Recently, several advanced large model series have released relatively small variants ranging from 1B to 2B parameters, such as InternVL3-2B (Chen et al. 2024c) and Qwen2-VL-2B (Wang et al. 2024b), which achieve performance comparable to earlier larger VLMs. However, their model size still poses significant challenges for on-device applications. For example, InternVL3-2B consumes over 4 GB of memory, pushing memory usage close to system limits and causing application instability.

Starting from the original carefully-designed models, quantization approaches reduce the precision of weights or activations from FP16 to lower bits, and substantially decrease computational resource demand while preserving model capacity. Thus, it has been an effective way to achieve better accuracy–efficiency trade-offs. Most previous works (Yu et al. 2025; Li et al. 2025a; Xie et al. 2024) focus on

*Corresponding author.

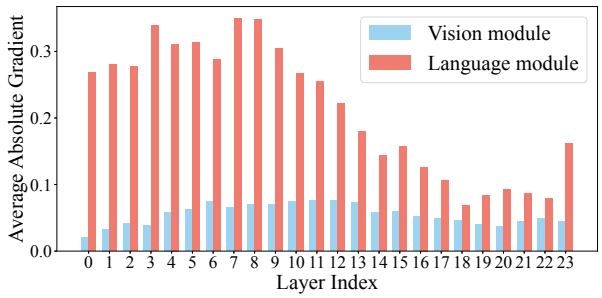

Figure 2: The average absolute gradients of vision (ViT) and language (LLM) module in the InternVL2.5-1B quantization-aware training process.

8/4-bit quantization and have restricted quantization to language modules, leaving the problem of unified quantization for both vision and language components largely unexplored. This limitation poses a significant challenge for deploying VLMs with 1–2B parameters, as the visual module accounts for a substantial portion of the parameters. Taking InternVL3-1B as an example, the ViT encoder contains 304M out of its 938M total parameters. Therefore, efficient deployment requires aggressive compression of both vision and language modules. Motivated by these observations, our work specifically targets low-bit quantization, down to 2 bits, for the entire VLM architecture.

Quantization-aware training (QAT) has been widely demonstrated to preserve accuracy more effectively compared to post-training quantization (PTQ) (Chen et al. 2024b; Du et al. 2024). However, directly applying QAT methods designed for LLMs to VLMs results in significant accuracy degradation, especially in the case of low-bit quantization. **Firstly, there is a large discrepancy in training sensitivity between the language module and the vision part.** As illustrated in Figure 2, the average absolute gradient value of the LLM is significantly higher than that of the ViT, suggesting a substantial sensitivity discrepancy during training. Treating these modules equally could result in considerable accuracy loss. **Secondly, QAT demands large calibration sets, yet multimodal data is far more difficult to collect than text-only data.** Alleviating data dependence appears to be crucial for improving the practicality and generalization of VLM quantization. **Lastly, utilizing QAT on relatively small-scale VLMs in the case of low bit is prone to instability, and often leads to oscillation or poor convergence.** A more robust and targeted optimization strategy is essential to maintain accuracy.

To address the above challenging issues, we built a staged processing with enhanced distillation framework (**SPEED-Q**) for efficient on-device VLMs quantization. Firstly, we propose a simple yet effective staged quantization strategy to gradually quantize different components, alleviating the sensitivity difference between ViT and LLM. Furthermore, we introduce a novel self-distillation strategy enhanced by an asymmetric clipping method, which improves the initial quantization state, along with a multi-loss optimization approach that enables a more constrained training process. To the best of our knowledge, SPEED-Q is the first quantization framework to enable 2/4-bit weight-only quantization of both vision and language modules in small-scale billion-parameter VLMs, achieving superior accuracy–efficiency trade-offs. As shown in Figure 1, our quantized model outperforms existing on-device VLMs by achieving higher accuracy with a smaller model size. For example, 2-bit quantized InternVL3-1B consumes less than 400 MB of running memory while achieving accuracy comparable to the best 0.6B model of FastVLM that requires almost 1.5GB of memory. This advantage unlocks the potential for deploying advanced VLMs on a wider range of edge devices. The main contributions are summarized as follows:

- For the first time, a 2/4-bit QAT framework is proposed to quantize the entire VLMs with 1-2B parameters, which is essential for on-device deployment.
- A staged quantization approach is proposed to reduce the effects of heterogeneous training sensitivities across modules in the VLMs.
- A distillation-enhanced quantization strategy is proposed to reduce data dependence and ultimately mitigate the accuracy collapse commonly observed in low-bit quantization of small VLMs, by employing improved initialization and multiple optimization targets.
- Extensive evaluations demonstrate that the proposed method consistently surpasses SOTA methods on VLMs of different families, sizes, and quantization schemes.

2 Related Work

Mainstream Quantization Paradigms for VLMs. Most quantization approaches for VLMs fall into the category of PTQ, exemplified by Q-VLM (Wang et al. 2024a), MBQ (Li et al. 2025a), P4Q (Sun et al. 2024), GPTQ (Frantar et al. 2023), and AWQ (Lin et al. 2024). The PTQ methods generally require minimal calibration data and are computationally efficient. However, significant accuracy degradation occurs when applied to relatively small VLMs. This collapse usually renders those models nonfunctional. QAT integrates quantization in the training loop and has been widely utilized in LLMs. Its extension to VLMs remains limited. The only notable VLM-QAT method, QSLAW (Xie et al. 2024), employs group-wise scaling and multimodal warm-up, but only quantizes the language component to 4-bit and leaves the vision module at its original precision (i.e., FP16). Thus, a robust approach enabling accurate low-bit quantization of both vision and language modules in compact VLMs remains an open issue.

Quantization Sensitivity across Different Modalities. Recent works have shown that vision and language modules in VLMs are unequally susceptible to quantization, prompting several targeted strategies. MQuant (Yu et al. 2025) employs modality-specific quantization factors, whereas AKVQ-VL (Su et al. 2025) dynamically adjusts bit budgets within attention to accommodate token-level variation and context saliency. Other component-aware efforts (Li et al. 2025a; Hao et al. 2024) focus on balancing vision and text losses or addressing specific bottlenecks such as quantized

prompts or cache outliers. By focusing solely on cross-modality token effects within the LLM, these approaches overlook the sensitivity of the vision component, limiting their effectiveness in full VLM quantization.

3 Preliminaries

Quantization methods for large models often employ group-wise or block-wise quantization, which divides weights into contiguous groups and each group shares a small set of parameters. As observed in *llama.cpp* (Gerganov 2023), further quantizing the group-wise quantization parameters (i.e., scales and zero-points) can yield a strictly lower overall memory footprint with minimal loss of fidelity. We formalize this strategy as *bilevel quantization*.

3.1 Group-wise Quantization

Given a weight tensor $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we split it into N groups of size G, such that $W = [W_1, \ldots, W_N]$, where $W_i \in \mathbb{R}^G$. Each block is quantized independently according to (possibly asymmetric) linear quantization:

$$q_i = \text{clamp}\left(\left\lfloor \frac{w_i}{s} + z \right\rfloor, 0, 2^b - 1\right), \tag{1}$$

where s is the scale, z is the zero-point, and b is the number of bits. For each quantization block, the scale is defined as $s = (x_{\max} - x_{\min})/(2^b - 1)$, mapping the original float range into the range of the targeting bit. The zero point is set as $z = \text{round}(-x_{\min}/s)$, which preserves the alignment between the original values and their quantized counterparts.

3.2 Bilevel Quantization

In practice, storing full-precision quantization parameters for each group requires additional memory, leading to higher resource overhead, especially in low-bit quantization, where a smaller group size is utilized for accuracy maintenance. Bilevel quantization addresses this issue by applying additional group-wise quantization to the first-level quantization parameters. Specifically, for a weight tensor split into N blocks, we denote the first-level scale vector as $\mathbf{S} = [s_1, \ldots, s_N]$ and zero point vector as $\mathbf{Z} = [z_1, \ldots, z_N]$ respectively. We then apply a second-level group quantization (with group size G_q) to $\mathbf{S}$, as follows:

$$\hat{s} = \text{GroupQuantization}(\mathbf{S}, b_s), \tag{2}$$

where b_s is the bit-width for the quantized scales at the second level. Note that zero-points z_i are already rounded to integers during the first quantization stage, and thus are not subjected to further quantization.

4 Method

Quantizing VLMs to low bit-widths exposes substantial sensitivity differences between vision and language modules, making unified strategies unstable. We therefore formalize the VLM quantization as minimizing

$$L = f(Q_{\text{ViT}},\ Q_{\text{LLM}};\ S_{\text{stage}},\ S_{\text{opt}}), \tag{3}$$

where Q denotes the quantization strategy for each module, S_{stage} represents the quantization stage arrangement, and

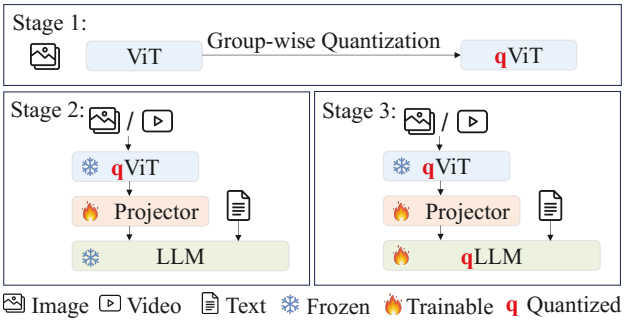

Figure 3: Pipeline of the staged quantization strategy. (a) Stage 1: The less sensitive ViT is quantized using an image-only calibration set. (b) Stage 2: Only the projector is trained to better align the quantized ViT (qViT) and the LLM. (c) Stage 3: qViT is frozen, and both the projector and the more sensitive LLM undergo quantization-aware training.

S_{opt} defines the design of optimization objectives. Effective low-bit VLM quantization therefore hinges on module-wise quantization configuration and staged training, paired with robust and data-efficient objectives, to maintain accuracy and training stability.

4.1 Staged Quantization Strategy

A naive approach for quantizing VLMs is to apply QAT uniformly across all modules. However, our empirical analysis reveals pronounced discrepancies in quantization sensitivity between the vision (ViT) and language (LLM) components: as shown in Figure 2, the language module consistently exhibits higher gradient magnitudes during training, indicating greater vulnerability to low-bit quantization. Motivated by this observation, we design a staged quantization strategy that schedules quantization order adaptively across modules, in order to progressively minimize the end-to-end quantization errors.

Specifically, the staged quantization strategy is illustrated in Figure 3, which decomposes the quantization process into three sequential stages. In Stage 1, group-wise quantization strategy is applied to the ViT blocks, where an adaptive rounding mechanism (Nagel et al. 2020) is utilized to minimize local reconstruction error for each block. In Stage 2, we freeze the quantized ViT and train only the Projector to adapt its output, aligning the quantized ViT features with the original distribution expected by the LLM. In Stage 3, the Projector and quantized LLM are jointly fine-tuned to restore performance and optimize end-to-end alignment under low-bit constraints.

4.2 Distillation-enhanced Quantization

For the quantization of the more sensitive language module, we adopt a distillation-enhanced training procedure that systematically addresses key challenges, including effective initialization, data efficiency, and optimization stability. As shown in Figure 4, favorable initial weight distributions for quantization are obtained through asymmetric clipping. We then introduce a self-distillation regime, where the original

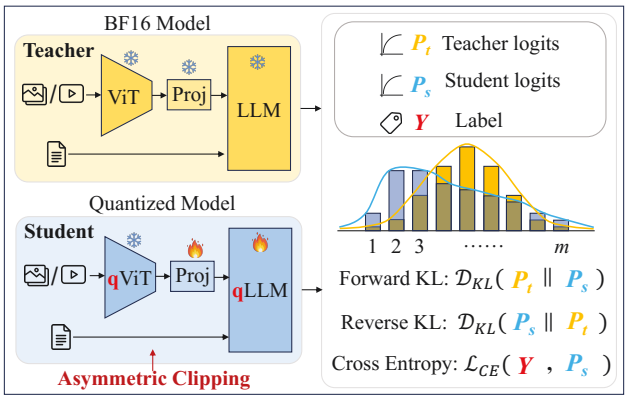

Figure 4: Illustration of the distillation-enhanced quantization. Asymmetric clipping provides a more favorable initialization for quantization-aware training. Our objective combines forward KL, reverse KL, and a task-specific loss. The self-distillation framework, together with this multi-loss optimization, improves training stability and reduces dependence on large calibration datasets.

model guides the training of the quantized model, thus significantly reducing dependence on diverse and large-scale training data. Finally, we combine this with a multi-loss optimization strategy, jointly enforcing consistency with both teacher outputs and ground-truth labels, which further improves convergence and robustness for aggressive low-bit quantization.

Asymmetric Clipping. We perform offline, layer-wise asymmetric clipping as an initialization step before QAT, aiming to stabilize training while incurring minimal computational overhead. Specifically, given input x from a small calibration set, we automatically search for two optimal clipping values α and β for each layer of the model. These values aim to minimize the L_2 distance between the outputs of quantized and original models:

$$\alpha^*, \beta^* = \arg\min_{\alpha,\beta} \left\| \widetilde{\mathrm{W}}_c x - \mathrm{W} x \right\|, \tag{4}$$

where $\mathrm{W}_c = clip(\mathrm{W}, \alpha, \beta)$ clips the weights out of the range $[\alpha, \beta]$, and $\widetilde{\mathrm{W}}_c$ denotes the quantized weights. In this way, the impact of outliers in the initial floating-point weights can be mitigated to some extent.

Self-distillation Strategy. As shown in Figrue 4, we employ self-distillation, where the fixed pre-trained BF16 model serves as the teacher and the quantized model as the student. The distillation loss combines the reverse KL and the forward KL as follows:

$$\mathcal{L}_{distill} = \gamma \mathcal{D}_{KL}(P_s || P_t) + (1-\gamma)\mathcal{D}_{KL}(P_t || P_s), \tag{5}$$

where P_t and P_s represent the output logits of the teacher model and the student model, respectively. The coefficient γ is estimated by the averaged token probability on a randomly selected mini-batch of training samples during the early stage of training, formalized as:

$$\gamma = \mathbb{E}_{(x,y)\sim\mathbb{D}}\left[\frac{1}{|\{y\}|}\sum_{i=1}^{|\{y\}|} P_t(y_i|x, y_{<i})\right]. \tag{6}$$

γ serves as a weighting factor that balances the contributions of forward and reverse KL divergences in the distillation loss. When the teacher is highly confident, reverse KL encourages the student to concentrate on strong predictions; otherwise, forward KL is prioritized to better match the full distribution. This adaptive mechanism leads to more stable and effective knowledge transfer, especially in low-bit quantization scenarios.

Multi-loss Optimization Strategy. We extend our self-distillation framework by incorporating explicit supervision from ground-truth labels to directly regularize the model's output distribution. This dual objective anchors the optimization process to the target task, ensuring that the quantized model remains aligned with the original learning target. We introduce the standard cross-entropy loss for the target task, and the total loss is then defined as

$$\mathcal{L}_{total} = \lambda \mathcal{L}_{distill} + \zeta \mathcal{L}_{CE}(Y, P_s). \tag{7}$$

These two losses are complementary: the former stabilizes feature-space alignment via self-distillation and reduces reliance on large-scale datasets, while the latter ensures fidelity to the target task. Together, they jointly guide the optimization trajectory, significantly alleviating training oscillations and instability. Although alternative weighting schemes of the two losses may yield marginally better performance, we set $\lambda = \zeta = 1$ for simplicity.

5 Experiments

5.1 Experimental Setups

Implementation Details. SPEED-Q is implemented using DeepSpeed (Rasley et al. 2020) and applies quantization to all linear layers in both the ViT and LLM. All training data used in SPEED-Q are drawn from publicly available open-source datasets. During quantization-aware training, we sample 10% of the data from each dataset, resulting in approximately 680,000 training samples in total. The quantization scheme follows the *bilevel quantization* described in Section 3. For 4-bit quantization, the first-level group size is 32 and the second-level group size is 128. For 2-bit quantization, both levels use a group size of 16.

Evaluation Datasets. To evaluate the performance of the quantized model, we conduct experiments on various benchmarks based on the VLMEvalKit (Duan et al. 2024). Specifically, we use MMBench (Liu et al. 2024a) and MMStar (Chen et al. 2024a) for comprehensive multimodal evaluation, ScienceQA (Lu et al. 2022) and MMMU (Yue et al. 2024) to evaluate visual reasoning, AI2D (Kembhavi et al. 2016) and OCRBench (Liu et al. 2024b) for text recognition and comprehension, SEED-Bench (Li et al. 2023) to test visual perception, and HallusionBench (Guan et al. 2024) for hallucination evaluation.

Model	Bitwidth	Method	MMBench	MMStar	MMMU	Hallusion.	AI2D	OCRBench	SEED	ScienceQA	Avg. (↑)
InternVL3-1B	BF16	-	69.08	52.27	41.33	36.25	69.75	79.7	71.16	89.84	63.67
	4-bit	RTN	63.16	48.73	36.67	32.89	65.12	74.1	69.01	83.21	59.11
		GPTQ (ICLR'23)	62.77	**52.2**	35	32.69	63.73	75.6	68.99	83.98	59.37
		AWQ (MLSys'24)	62.62	49.8	36.56	35.95	64.67	74.8	68.59	83.36	59.54
		MBQ (CVPR'25)	63.85	50.93	38.67	31.66	66.87	74.9	69.28	82.88	59.88
		SPEED-Q (Ours)‡	**66.68**	51.07	**40.44**	**37.01**	**68.81**	**79.7**	**71.39**	**84.41**	**62.44**
	2-bit	RTN	0.04	7.6	6.89	0.73	6.67	0.5	6.45	8.01	4.61
		GPTQ (ICLR'23)	0.15	10.2	11.67	0.43	8.19	1.6	11.12	12.83	7.02
		MBQ (CVPR'25)	0.00	7.53	5.33	0.11	5.25	0.1	6.30	5.87	3.81
		SPEED-Q (Ours)‡	**53.13**	**43.4**	**32.89**	**40.21**	**57.58**	**58.5**	**66.93**	**59.18**	**51.48**
InternVL3-2B	BF16	-	78.68	61.07	45.56	41.94	78.76	83.6	74.95	95.23	69.97
	4-bit	RTN	73.57	57	43.33	39.37	75.91	81.2	73.22	92.32	66.99
		GPTQ (ICLR'23)	75.93	57.93	45.89	42.72	75.94	81.2	73.89	92.42	68.24
		AWQ (MLSys'24)	75.12	58.13	**47.22**	41.50	76.88	82.1	73.85	91.23	68.25
		MBQ (CVPR'25)	75.43	**59.47**	46.11	38.25	76.36	82.0	73.37	92.75	67.97
		SPEED-Q (Ours)‡	**77.67**	58.47	44.56	**44.03**	**77.3**	**82.4**	**74.42**	**93.32**	**69.02**
	2-bit	RTN	0.08	2.2	1.89	1.24	1.49	0.3	6.89	1.81	1.99
		GPTQ (ICLR'23)	0.62	10.53	7.11	3.20	11.43	12.7	12.63	12.4	8.83
		MBQ (CVPR'25)	0.12	9.47	8.44	3.48	9.29	5.0	10.18	10.35	7.04
		SPEED-Q (Ours)‡	**68.30**	**49.13**	**40.56**	**38.34**	**69.95**	**64.9**	**72.34**	**72.20**	**59.47**
Qwen2-VL-2B	BF16	-	71.59	46.4	39.44	41.88	72.02	81.0	72.66	73.68	62.33
	4-bit	RTN	**70.98**	45.07	37.22	**42.74**	**71.15**	78.8	72.45	70.67	61.13
		GPTQ (ICLR'23)	70.51	46.33	36.22	39.62	70.53	79.5	72.62	72.10	60.93
		AWQ (MLSys'24)	68.89	44.8	37.33	39.55	70.08	78.9	71.83	**72.15**	60.44
		MBQ (CVPR'25)	70.55	44.53	38.22	39.89	70.21	**80.9**	71.86	71.72	60.98
		SPEED-Q (Ours)‡	69.85	**50.87**	**42.0**	41.71	70.92	76.5	**74.40**	72.01	**62.28**
	2-bit	RTN	1.16	15.6	12.0	16.11	18.91	10.9	17.72	24.65	14.63
		GPTQ (ICLR'23)	1.97	17.33	13.33	3.97	16.09	20.1	23.19	24.51	15.06
		MBQ (CVPR'25)	2.17	17.53	13.56	26.30	18.65	20.4	23.46	26.13	18.52
		SPEED-Q (Ours)‡	**57.31**	**42.93**	**35.22**	**34.11**	**60.33**	**57.0**	**68.51**	**60.13**	**51.94**

Table 1: Main results on InternVL3 and Qwen2-VL families. ‡ indicates that both ViT and LLM are quantized; otherwise, only the LLM is quantized. To ensure a fair comparison, competing methods use 4-bit weight quantization with group size 128 (w4g128), while SPEED-Q employs *bilevel quantization* with w4g32g128, yielding a comparable average bitwidth (4.25 vs. 4.28 bits) calculated on the quantized components in VLMs. Under 2-bit settings, the compared methods exhibit severe accuracy degradation across various configurations. We thus report them under w2g16, while SPEED-Q uses w2g16g16.

Model	Method	BF16	4-bit	2-bit
InternVL3-1B	Quantize LLM only	1789.5	941.7	853.1
	SPEED-Q(Ours)		**485.0**	**315.0**
InternVL3-2B	Quantize LLM only	3984.5	1454.6	1218.7
	SPEED-Q(Ours)		**936.1**	**661.5**

Table 2: Model file sizes (in MB) under different quantization schemes. SPEED-Q quantizes both ViT and LLM, enabling smaller model file sizes.

Model	Bitwidth	Method	DocVQA	RealWorldQA
LLaVA-13B	BF16	-	14.46	48.49
	4-bit	QSLAW	3.46	40.65
		SPEED-Q‡	**13.54**	**55.16**

Table 3: Comparison with QSLAW (Xie et al. 2024). Since QSLAW quantizes LLaVA-13B (Liu et al. 2023) only to 4-bit, we perform comparisons under the same bitwidth setting. ‡ indicates that both ViT and LLM are quantized.

5.2 Comparison with State-of-the-Arts

Table 1 presents a performance comparison between our SPEED-Q and previous state-of-the-art methods across multiple VLM families and benchmarks. Generally speaking, our method quantizes more components while delivering superior performance. Specifically, by utilizing the proposed SPEED-Q, the quantized 4-bit model exhibits a marginal performance decrease of 3% compared to their original BF16 counterpart. This observation is consistent across InternVL3-1B, InternVL3-2B, and Qwen2-VL-2B. Compared to other methods that only quantize the LLM module, SPEED-Q achieves superior performance on most of

Model	Bitwidth	MMBench	MMStar	MMMU	Hallusion.	AI2D	OCRBench	SEED	ScienceQA	Avg. (↑)
BF16:										
SmolVLM-256M (Marafioti et al. 2025)	BF16	25.19	34.6	27.0	26.33	47.09	52.6	54.31	72.39	42.44
SmolVLM-500M (Marafioti et al. 2025)	BF16	41.06	38.33	31.44	29.43	59.52	61.0	62.03	76.73	49.94
FastVLM-600M (Vasu et al. 2025)	BF16	56.39	44.8	33.55	37.11	67.91	58.2	57.69	80.54	**54.52**
H2OVL-800M (Galib et al. 2024)	BF16	48.14	38.93	30.78	28.60	53.47	75.0	60.18	66.48	50.20
Quantized:										
InternVL2.5-1B SPEED-Q	**4-bit**	**66.41**	**50.27**	**36.89**	**41.41**	**68.36**	**75.8**	**71.47**	**89.08**	**62.46**
InternVL2.5-1B SPEED-Q	2-bit	53.68	43.6	33.33	37.04	59.42	61.2	66.78	61.18	52.03
FastVLM-600M SPEED-Q	4-bit	51.70	42.4	32.22	31.71	64.70	53.4	61.43	77.68	51.91
FastVLM-600M SPEED-Q	2-bit	41.99	39.53	29.56	31.93	54.86	50.2	60.72	61.66	46.31

Table 4: Main results on on-device VLMs. 4-bit InternVL2.5-1B (485 MB, model file size) outperforms BF16 FastVLM (1517 MB) by +7.94 points, while the 2-bit InternVL2.5-1B (315 MB) matches its performance at 1/5 size.

Components		MMStar	AI2D	OCRBench
Task Loss	Distillation Loss			
✓	✗	49.33	66.13	75.5
✗	✓	50.40	68.46	77.9
✓	✓	**51.07**	**68.81**	**79.7**
Joint Quantization		49.67	68.46	77.0
Staged Quantization		**51.07**	**68.81**	**79.7**

Table 5: Ablation study on various configurations of training loss and training strategy in our approach.

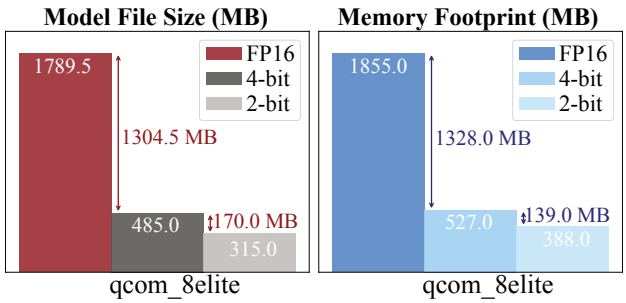

Figure 5: On-device efficiency evaluation of FP16 and low-bit InternVL2.5-1B on Samsung Galaxy S25 Ultra.

the evaluated benchmarks. Furthermore, nearly all existing methods suffer severe accuracy degradation under 2-bit quantization, rendering them practically ineffective. In contrast, SPEED-Q maintains controllable performance degradation under 2-bit quantization, with accuracy dropping by only approximately 15% with respect to the BF16 models. This result further demonstrates the effectiveness and robustness of our method in extreme low-bit. Since the visual features extracted by the ViT encoder are crucial for multimodal understanding, the results indicate that our method effectively mitigates the impact of ViT quantization on the end-to-end performance.

Table 2 lists the model size of different approaches, which is critical for memory usage and inference speed when deployed on edge devices. SPEED-Q successfully compresses VLMs into significantly smaller model sizes by simultaneously quantizing ViT and LLM components.

Table 3 compares the QAT-based method QSLAW with our method on DocVQA (Mathew et al. 2021) and RealWorldQA (Corp 2024). As shown in the table, SPEED-Q demonstrates clear superiority.

5.3 Comparison with Lightweight VLMs

As shown in Table 4, we first compare the quantized models with the existing VLMs designed for edge deployment, such as SmolVLM, FastVLM, and H2OVL, which typically have fewer than 1B parameters. The 4-bit quantized InternVL2.5-1B achieves substantial leadership across multiple benchmarks compared to all the existing BF16 lightweight models. Furthermore, the 2-bit quantized InternVL2.5-1B exhibits comparable accuracy to FastVLM, which exhibits a favorable trade-off between model size and performance. More importantly, the quantized model significantly reduces computational resource requirements. In addition, we apply SPEED-Q on the FastVLM. SPEED-Q successfully limits the performance drop to 4.78% and 15.06% under 4-bit and 2-bit quantization respectively. The results further demonstrate that SPEED-Q incurs manageable accuracy degradation, even on the lightweight VLMs.

5.4 Ablation Study

To evaluate the contribution of each component, we conduct ablation studies on the 4-bit quantization of InternVL3-1B using the MMStar, AI2D, and OCRBench benchmarks.

Effectiveness of Multi-loss Optimization Strategy. As shown in Table 5, using distillation loss results in better accuracy preservation. Jointly optimizing both losses achieves significant performance improvements across all three benchmarks. This result confirms that our multi-loss optimization strategy enhances training stability by identifying more effective quantization targets.

Effectiveness of the Staged Quantization Strategy. As shown in Table 5, simultaneously quantizing both the ViT and LLM (i.e., joint quantization) often leads to unstable training and suboptimal performance. In contrast, staged quantization strategy applies quantization to the ViT and

Model	Bitwidth	Method	MMBench	MMStar	MMMU	Hallusion.	AI2D	OCRBench	SEED	ScienceQA	Avg. (↑)
InternVL3-8B	BF16	-	85.22	68.6	56.89	48.24	85.36	88.2	77.28	97.85	75.95
	4-bit	MBQ (CVPR'25)	**84.75**	**67.4**	**55.67**	48.92	84.10	**87.4**	**77.09**	**97.47**	**75.35**
		SPEED-Q (Ours)‡	84.25	66.47	54.78	**51.99**	84.10	84.6	76.98	97.19	75.04
	2-bit	MBQ (CVPR'25)	66.29	53.13	43.44	38.75	71.57	70.0	70.83	79.30	61.66
		SPEED-Q (Ours)‡	**75.81**	**60.2**	**46.22**	**50.61**	**79.08**	**75.8**	**75.24**	**84.69**	**68.46**

Table 6: Main results of SPEED-Q and MBQ on InternVL3-8B.

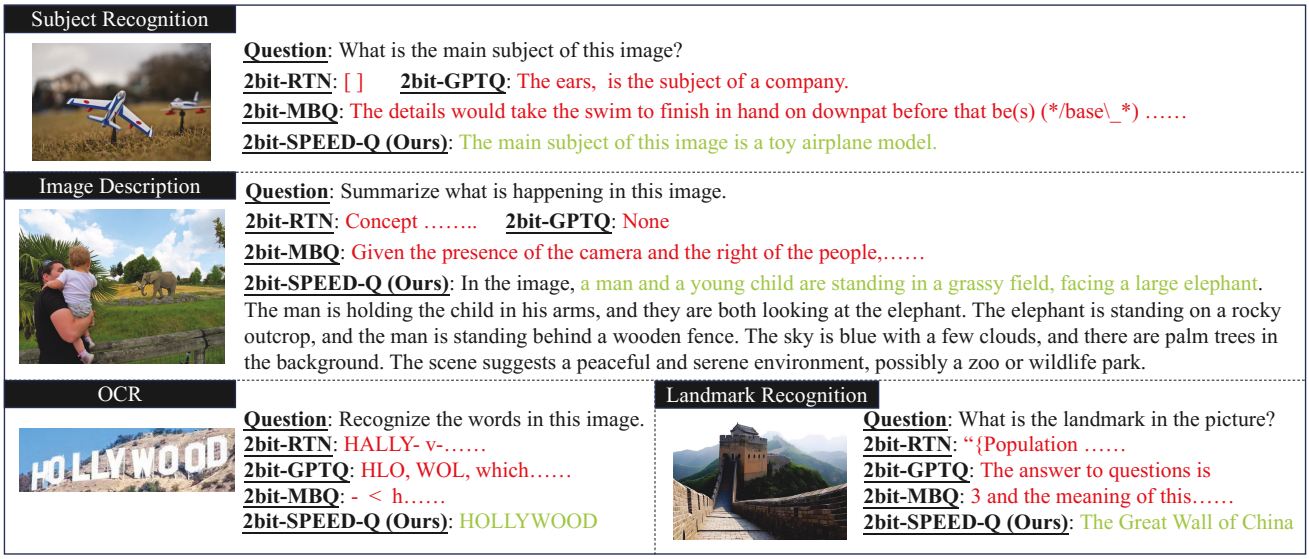

Figure 6: Qualitative comparison on InternVL2.5-1B under 2-bit quantization. SPEED-Q generates more accurate responses.

LLM in a sequential, multi-stage way. The results demonstrate that the staged quantization strategy achieves superior accuracy across multiple benchmarks. It confirms our hypothesis that decoupling the quantization process reduces optimization difficulty, improves training stability, and ultimately yields higher-quality quantized models.

5.5 Generalization on Large VLMs

To further assess the generalization of our method, we evaluate on InternVL3-8B with almost 8 billion parameters. As shown in Table 6, both SPEED-Q and MBQ preserve the accuracy, with SPEED-Q further distinguishing itself by quantizing both the vision and language components. When the model is quantized to 2-bit, SPEED-Q demonstrates superior performance across multiple benchmarks.

5.6 Efficiency Evaluation

Figure 5 illustrates the benefits of low-bit quantization in reducing both model file size and runtime memory footprint. Compared to the FP16 baseline, 4-bit quantization reduces the model size by 1304.5 MB, with an additional 170.0 MB reduction under 2-bit quantization. This advantage makes the deployment of VLMs more practical by requiring less storage and reducing distribution costs. Moreover, lower-bit quantization further reduces runtime memory footprint, which is crucial for enabling efficient and stable inference on resource-constrained edge devices.

5.7 Qualitative Analysis

As shown in Figure 6, SPEED-Q generates meaningful responses, while existing methods fail to produce accurate outputs. The observation is consistent across different tasks, demonstrating superior preservation of semantic and visual understanding under extreme low-bit quantization.

6 Conclusion

In order to further unleash the potential of VLMs deployment on edge devices, we have presented SPEED-Q, a novel quantization framework for 1-2B parameter VLMs. Our staged quantization strategy addresses the divergent training sensitivities between ViT and LLM modules, enabling stable and effective quantization across heterogeneous modalities. Furthermore, we propose distillation-enhanced quantization to stabilize the training process of low-bit VLMs and reduce dependence on large labeled datasets. Extensive experiments demonstrate that SPEED-Q achieves state-of-the-art performance. These results significantly advance the practical deployment of VLMs on edge devices. In future work, we plan to extend SPEED-Q from weight-only to weight-activation quantization, aiming to achieve further improvements in inference efficiency.

References

Chen, L.; Li, J.; Dong, X.; et al. 2024a. Are We on the Right Way for Evaluating Large Vision-Language Models? *arXiv preprint arXiv:2403.20330*.

Chen, M.; Shao, W.; Xu, P.; Wang, J.; Gao, P.; Zhang, K.; and Luo, P. 2024b. EfficientQAT: Efficient Quantization-Aware Training for Large Language Models. *arXiv preprint arXiv:2407.11062*.

Chen, Z.; Wu, J.; Wang, W.; et al. 2024c. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 24185–24198.

Corp, X. 2024. Grok-1.5 Vision Preview: Connecting the Digital and Physical Worlds with Our First Multimodal Model. https://x.ai/blog/grok-1.5v.

Du, D.; Zhang, Y.; Cao, S.; Guo, J.; Cao, T.; Chu, X.; and Xu, N. 2024. BitDistiller: Unleashing the Potential of Sub-4-Bit LLMs via Self-Distillation. *arXiv preprint arXiv:2402.10631*.

Duan, H.; Yang, J.; Qiao, Y.; et al. 2024. VLMEvalKit: An Open-Source Toolkit for Evaluating Large Multi-Modality Models. In *Proceedings of the ACM International Conference on Multimedia*, 11198–11201.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. In *Proceedings of the International Conference on Learning Representations*.

Galib, S.; Wang, S.; Xu, G.; Pfeiffer, P.; Chesler, R.; Landry, M.; and Ambati, S. S. 2024. H2OVL-Mississippi Vision Language Models Technical Report. *arXiv preprint arXiv:2410.13611*.

Gerganov, G. 2023. llama.cpp: Port of Facebook's LLaMA model in C/C++. https://github.com/ggerganov/llama.cpp.

Guan, T.; Liu, F.; Wu, X.; et al. 2024. HallusionBench: An Advanced Diagnostic Suite for Entangled Language Hallucination and Visual Illusion in Large Vision-Language Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 14375–14385.

Hao, T.; Ding, X.; Feng, J.; Yang, Y.; Chen, H.; and Ding, G. 2024. Quantized Prompt for Efficient Generalization of Vision-Language Models. In *Proceedings of the European Conference on Computer Vision*, 54–73.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361*.

Kembhavi, A.; Salvato, M.; Kolve, E.; Seo, M.; Hajishirzi, H.; and Farhadi, A. 2016. A Diagram is Worth a Dozen Images. In *Proceedings of the European Conference on Computer Vision*, 235–251.

Li, B.; Wang, R.; Wang, G.; Ge, Y.; Ge, Y.; and Shan, Y. 2023. SEED-Bench: Benchmarking Multimodal LLMs with Generative Comprehension. *arXiv preprint arXiv:2307.16125*.

Li, S.; Hu, Y.; Ning, X.; et al. 2025a. MBQ: Modality-Balanced Quantization for Large Vision-Language Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 4167–4177.

Li, Z.; Wu, X.; Du, H.; Liu, F.; Nghiem, H.; and Shi, G. 2025b. A Survey of State of the Art Large Vision Language Models: Benchmark Evaluations and Challenges. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 1587–1606.

Lin, J.; Tang, J.; Tang, H.; et al. 2024. AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. In *Proceedings of Machine Learning and Systems*, 87–100.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36: 34892–34916.

Liu, Y.; Duan, H.; Zhang, Y.; et al. 2024a. MMBench: Is Your Multi-modal Model an All-Around Player? In *Proceedings of the European Conference on Computer Vision*, 216–233.

Liu, Y.; Li, Z.; Huang, M.; et al. 2024b. OCRBench: On the Hidden Mystery of OCR in Large Multimodal Models. *Science China Information Sciences*, 67(12): 220102.

Lu, P.; Mishra, S.; Xia, T.; et al. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *Proceedings of the Neural Information Processing Systems*.

Marafioti, A.; Zohar, O.; Farré, M.; Noyan, M.; Bakouch, E.; Cuenca, P.; Zakka, C.; Allal, L. B.; Lozhkov, A.; Tazi, N.; et al. 2025. SmolVLM: Redefining Small and Efficient Multimodal Models. *arXiv preprint arXiv:2504.05299*.

Mathew, M.; et al. 2021. DocVQA: A Dataset for VQA on Document Images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2200–2209.

Nagel, M.; Amjad, R. A.; Van Baalen, M.; Louizos, C.; and Blankevoort, T. 2020. Up or Down? Adaptive Rounding for Post-Training Quantization. In *Proceedings of the International Conference on Machine Learning*, 7197–7206.

Rasley, J.; Rajbhandari, S.; Ruwase, O.; and He, Y. 2020. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3505–3506.

Su, Z.; Shen, W.; Li, L.; Chen, Z.; Wei, H.; Yu, H.; and Yuan, K. 2025. AKVQ-VL: Attention-Aware KV Cache Adaptive 2-Bit Quantization for Vision-Language Models. *arXiv preprint arXiv:2501.15021*.

Sun, H.; Wang, R.; Li, Y.; Cao, X.; Jiang, X.; Hu, Y.; and Zhang, B. 2024. P4Q: Learning to Prompt for Quantization in Visual-language Models. *arXiv preprint arXiv:2409.17634*.

Vasu, P. K. A.; Faghri, F.; Li, C.-L.; et al. 2025. FastVLM: Efficient Vision Encoding for Vision Language Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 19769–19780.

Wang, C.; Wang, Z.; Xu, X.; Tang, Y.; Zhou, J.; and Lu, J. 2024a. Q-VLM: Post-training Quantization for Large Vision-Language Models. *arXiv preprint arXiv:2410.08119*.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024b. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*.

Xie, J.; Zhang, Y.; Lin, M.; Cao, L.; and Ji, R. 2024. Advancing Multimodal Large Language Models with Quantization-Aware Scale Learning for Efficient Adaptation. In *Proceedings of the ACM International Conference on Multimedia*, 10582–10591.

Yu, J.; Zhou, S.; Yang, D.; et al. 2025. MQuant: Unleashing the Inference Potential of Multimodal Large Language Models via Full Static Quantizatio. *arXiv preprint arXiv:2502.00425*.

Yue, X.; Ni, Y.; Zhang, K.; et al. 2024. MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.