Rethinking Small VLM Quantization: From Component-Wise Analysis to Hardware-Aware Edge Deployment

Hyeju Shin [*1] **Chorwon Kim** [*1] **Ryangsoo Kim** [1] **Hark Yoo** [1] **Jaein Kim** [1]

Abstract

The emergence of vision language models with fewer than 3 billion parameters has accelerated the implementation of on-device multimodal intelligence. However, a detailed understanding of component-wise quantization remains a bottleneck for optimal deployment. This paper presents a systematic evaluation framework for empirically validating five hypotheses across six quantization configurations on the Jetson Orin NX and AGX. By separating the vision encoder, projector, and large language model backbone yields the following results: (1) Quantization sensitivity is governed by the structural paradigm (MoE vs. dense) rather than scale alone, with MoE backbones mitigating INT4 noise where dense backbones degrade; (2) SigLIP encoders incur disproportionate INT8 latency on Jetson Ampere–a deployment-specific encoder-kernel-hardware interaction, not a SigLIP flaw; (3) Although INT4 quantization of LLMs greatly reduces VRAM consumption, it also causes slower token generation due to dequantization overhead; (4) Composite quantization errors are largely additive, except along the modality-alignment path, which is architecture-dependent; (5) The intelligence-per-joule profile varies significantly across platforms owing to memory bandwidth constraints.

1. Introduction

Recently, large multimodal models (LMMs) have opened a new horizon in artificial intelligence (AI) research owing to their innovative reasoning capabilities, which combine visual understanding and language generation (Liu et al., 2023; Sun et al., 2024; Hurst et al., 2024). Small vision-language models (sVLMs) with 3 billion or less parameters are rapidly emerging as practical candidates for on-device AI. These models can be used in mobile, offline, privacy-protected, and low-power environments (Chu et al., 2023; Zhou et al., 2024; Lin et al., 2026). However, heterogeneous edge system-on-chip (SoC) environments such as the NVIDIA Jetson Orin series are subject to fundamental hardware constraints, including strict memory bandwidth (NX: 102.4 GB/s; AGX: 204.8 GB/s) and limited VRAM capacity (NX: 16 GB; AGX: 64 GB). In such an environment, model quantization is gaining attention as an essential compression technique to ensure practical reasoning capacity (Kim et al., 2022; Cai et al., 2023; Xu et al., 2024).

However, most existing research on the quantization of large language models (LLMs) and vision-language models (VLMs) has generally focused on evaluating the entire model end-to-end in high-performance server-grade GPU environments (Lee et al., 2024; Xiao et al., 2023; Frantar et al., 2022; Dettmers et al., 2022). Existing studies have yet to provide a systematic basis for key deployment-time decisions faced by engineers, including which components should be prioritized for quantization and how bit-widths should be allocated across these components. A VLM typically consists of three major components: a vision encoder that extracts visual features, a projector that maps these features into the embedding space of the language model, and an LLM backbone that performs the final inference (Liu et al., 2023; Alayrac et al., 2022; Chen et al., 2024). Each component has different parameter sizes, model structures, and computational characteristics; thus, the sensitivity of each component to quantization should be evaluated independently. Studies that control and quantify the marginal contribution of each component are rare (Sharshar et al., 2025). Cross-platform comparative studies, especially on heterogeneous edge SoCs, are even rarer.

Motivated by our preliminary observations, we formulate five hypotheses and evaluate each by designing a hypothesis-driven evaluation framework. We evaluate five architecturally diverse sVLMs: Qwen3-VL-2B, DeepSeek-VL2-Tiny, PaliGemma2-3B, LLaVA-OV-0.5B, and Kosmos-2.5. The developed configurations include single-component ablations (`cfg1`, `cfg2`, and `cfg4`), which isolate the effect of quantizing individual components, and composite set-

[1]ETRI, Republic of Korea. Correspondence to: Hark Yoo <harkyoo@etri.re.kr>, Jaein Kim <jaein@etri.re.kr>.

ICML 2026 Workshop on Hypothesis Testing, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

Table 1. Comprehensive evaluation: perception, cognition, and total MME scores with percentage variations relative to `cfg0` (baseline) across all models and configurations.

Model	Config	Jetson NX Orin			Jetson AGX Orin		
		Perc. (Δ%)	Cogn. (Δ%)	Total (Δ%)	Perc. (Δ%)	Cogn. (Δ%)	Total (Δ%)
Qwen3-VL-2B-Instruct	cfg0 (Base)	1508.20	513.57	2021.78	1485.46	497.86	1983.32
	cfg1	1524.25 (+1.06%)	553.57 (+7.79%)	2077.82 (+2.77%)	1535.48 (+3.37%)	533.57 (+7.17%)	2069.05 (+4.32%)
	cfg2	1512.75 (+0.30%)	477.14 (-7.09%)	1989.89 (-1.58%)	1501.77 (+1.10%)	503.57 (+1.15%)	2005.34 (+1.11%)
	cfg3	1514.92 (+0.45%)	531.43 (+3.48%)	2046.35 (+1.22%)	1538.39 (+3.56%)	526.07 (+5.67%)	2064.46 (+4.09%)
	cfg4	1505.65 (-0.17%)	467.86 (-8.90%)	1973.51 (-2.39%)	1504.48 (+1.28%)	523.93 (+5.24%)	2028.41 (+2.27%)
	cfg5	1511.96 (+0.25%)	517.14 (+0.69%)	2029.10 (+0.36%)	1524.09 (+2.60%)	554.64 (+11.40%)	2078.73 (+4.81%)
DeepSeek-VL2-Tiny	cfg0 (Base)	1550.76	360.36	1911.12	1556.35	360.36	1916.71
	cfg1	1584.33 (+2.16%)	368.21 (+2.18%)	1952.54 (+2.17%)	1589.25 (+2.11%)	372.86 (+3.47%)	1962.10 (+2.37%)
	cfg2	1552.34 (+0.10%)	364.64 (+1.19%)	1916.98 (+0.31%)	1548.53 (-0.50%)	376.07 (+4.36%)	1924.60 (+0.41%)
	cfg3	1582.56 (+2.05%)	370.36 (+2.78%)	1952.92 (+2.19%)	1580.41 (+1.55%)	370.36 (+2.78%)	1950.76 (+1.78%)
	cfg4	1545.90 (-0.31%)	351.07 (-2.58%)	1896.98 (-0.74%)	1546.17 (-0.65%)	363.57 (+0.89%)	1909.74 (-0.36%)
	cfg5	1574.16 (+1.51%)	370.36 (+2.78%)	1944.51 (+1.75%)	1584.90 (+1.83%)	368.21 (+2.18%)	1953.11 (+1.90%)
PaliGemma2-3B	cfg0 (Base)	1424.94	236.07	1661.01	1429.53	228.57	1658.10
	cfg1	1387.46 (-2.63%)	211.43 (-10.44%)	1598.89 (-3.74%)	1389.45 (-2.80%)	209.29 (-8.44%)	1598.74 (-3.58%)
	cfg2	1425.10 (+0.01%)	238.57 (+1.06%)	1663.68 (+0.16%)	1424.90 (-0.32%)	228.57 (+0.00%)	1653.47 (-0.28%)
	cfg3	1390.95 (-2.39%)	209.29 (-11.34%)	1600.24 (-3.66%)	1391.08 (-2.69%)	209.29 (-8.44%)	1600.37 (-3.48%)
	cfg4	1429.35 (+0.31%)	238.93 (+1.21%)	1668.28 (+0.44%)	1420.90 (-0.60%)	236.07 (+3.28%)	1656.97 (-0.07%)
	cfg5	1382.68 (-2.97%)	207.14 (-12.25%)	1589.83 (-4.29%)	1379.90 (-3.47%)	209.29 (-8.44%)	1589.19 (-4.16%)
LLaVA-OV-0.5B	cfg0 (Base)	1210.76	151.07	1361.83	1214.02	155.71	1369.73
	cfg1	967.52 (-20.09%)	174.29 (+15.37%)	1141.81 (-16.16%)	1029.01 (-15.24%)	184.64 (+18.58%)	1213.65 (-11.39%)
	cfg2	1209.34 (-0.12%)	151.07 (+0.00%)	1360.41 (-0.1%)	1215.06 (+0.09%)	148.21 (-4.82%)	1363.28 (-0.47%)
	cfg3	965.8 (-20.23%)	171.79 (+13.72%)	1137.59 (-16.47%)	1021.05 (-15.90%)	186.43 (+19.73%)	1207.48 (-11.85%)
	cfg4	1220.27 (+0.79%)	148.93 (-1.42%)	1369.20 (+0.54%)	1212.08 (-0.16%)	153.21 (-1.61%)	1365.30 (-0.32%)
	cfg5	974.79 (-19.49%)	174.29 (+15.37%)	1149.08 (-15.62%)	1026.16 (-15.47%)	189.29 (+21.57%)	1215.45 (-11.26%)
Kosmos-2.5	cfg0 (Base)	464.80	197.50	662.30	464.55	197.50	662.05
	cfg1	454.97 (-2.11%)	196.79 (-0.36%)	651.76 (-1.59%)	453.05 (-2.48%)	196.79 (-0.36%)	649.84 (-1.84%)
	cfg2	466.05 (+0.27%)	198.21 (+0.36%)	664.27 (+0.30%)	465.80 (+0.27%)	197.50 (+0.00%)	663.30 (+0.19%)
	cfg3	457.14 (-1.65%)	196.79 (-0.36%)	653.92 (-1.27%)	459.97 (-0.99%)	196.79 (-0.36%)	656.76 (-0.80%)
	cfg4	465.05 (+0.05%)	201.43 (+1.99%)	666.48 (+0.63%)	464.05 (-0.11%)	198.21 (+0.36%)	663.27 (+0.18%)
	cfg5	457.39 (-1.59%)	199.29 (+0.91%)	656.67 (-0.85%)	453.80 (-2.31%)	196.79 (-0.36%)	650.59 (-1.73%)

Note: Each data point reflects the average of $n \geq 3$ independent runs. Refer to Appendix D for the detailed bar chart representing the mean score across runs.

Table 2. Definition of configurations for component-wise quantization sensitivity analysis.

Config	Vision	Projector	LLM	Description
cfg0 (Base)	FP16	FP16	FP16	Baseline
cfg1	FP16	FP16	**INT4**	LLM Robustness
cfg2	FP16	**INT8**	FP16	Proj. Sensitivity
cfg3	FP16	**INT8**	**INT4**	cfg1 + cfg2
cfg4	**INT8**	FP16	FP16	Enc. Sensitivity
cfg5	**INT8**	FP16	**INT4**	cfg1 + cfg4

tings (`cfg3` and `cfg5`), which capture interactions among jointly quantized components. We conduct all experiments on the Jetson Orin NX and Jetson AGX platforms and analyze each configuration along four axes: MME benchmark accuracy, VRAM footprint, component-wise latency across the vision encoder, projector, and LLM time per output token (TPOT), and intelligence-per-joule (IPJ)(Saad-Falcon et al., 2025).

We formulate the following hypotheses:

- **Scale-Dependent Sensitivity according to Parameter Size (H1):** INT4 quantization of the LLM backbone results in increasingly severe accuracy degradation as the model size decreases. This effect is expected to be especially pronounced in ultra-small sVLMs with fewer than 1B parameters, where limited representational redundancy makes the model less robust to quantization noise.

- **Vision Encoder Architecture-Specific Latency Anomaly (H2):** SigLIP-based vision encoder quantization disproportionately increases the latency relative to the variation in accuracy because of interactions between certain architectures and the quantization software stack that are specific to the hardware.

- **Relationship between Resource Savings and Inference Efficiency (H3):** BitsAndBytes-based LLM INT4 quantization achieves over 50% VRAM savings, whereas TPOT consistently increases due to the dequantization overhead.

- **Non-Additive Interaction of Composite Quantization Errors (H4):** The performance degradation when quantizing heterogeneous modalities is characterized by a non-additive pattern that cannot be explained by simple arithmetic summation of the individual component errors.

- **Stability and Transferability of Performance Rankings across Platforms (H5):** The accuracy ranking of the models is platform-invariant across the quantization configurations and heterogeneous edge platforms (NX vs. AGX), whereas the latency and energy profiles are platform-specific.

The main contributions of this study are as follows: **(1)** We present the first systematic controlled experiment that independently separates the vision encoder, projector, and LLM backbone of VLMs using 3 billion parameters to measure accuracy sensitivity and resource efficiency. **(2)** Through cross-hardware evaluation using two heterogeneous platforms (Jetson Orin NX and AGX), we empirically distinguish platform-consistent accuracy rankings from hardware-specific latency profiles. **(3)** We identify abnormal latency phenomena that occur when applying vision INT8 quantization in SigLIP-based architectures, and analyze these issues from a hardware-software stack perspective. **(4)** By testing the hypotheses, we provide practical deployment guidelines for implementing modality- and hardware-aware precision allocation strategies in edge environments.

2. Related Works

Small Vision-Language Models. A sVLM is a multimodal AI system with 7 billion parameters or less. sVLMs are designed to efficiently process and integrate visual and textual information while supporting implementation in resource-constrained environments, such as edge and mobile devices (Chu et al., 2023). A typical VLM architecture consists of three core components: a vision encoder that extracts visual features from images, a projector that maps visual representations into the embedding space of the language model, and a LLM backbone for text generation and reasoning (Liu et al., 2023). These components serve distinct functions within the model and may, therefore, differ in their sensitivity to quantization (Sharshar et al., 2025). Table 4 summarizes the architectural configurations of the five models selected for this study. Except for Kosmos-2.5, the other four models use SigLIP as the vision encoder. Thus, SigLIP can be treated as a controlled architectural factor in the architecture-effect analysis for RQ2, as demonstrated in Section 4.2.

Quantization of VLMs. Model quantization is a widely used compression technique that reduces the memory footprint and computational overhead by representing model weights and activations in low-precision formats. Existing quantization methods, such as activation-aware weight quantization (AWQ), generative pre-trained transformer quantization (GPTQ), SmoothQuant, and BitsAndBytes-based INT8/INT4 quantization, have demonstrated substantial efficiency gains in large-scale neural models (Lin et al., 2024; Frantar et al., 2022; Xiao et al., 2023; Dettmers et al., 2022).

Table 3. Hardware specifications of Jetson Orin platforms

Specification	Jetson Orin NX	Jetson Orin AGX
GPU	1024-core Ampere, 32 Tensor Cores	2048-core Ampere, 64 Tensor Cores
CPU	8-core ARM CPU	12-core ARM CPU
Memory	16GB LPDDR5	64GB LPDDR5
Memory BW	102.4 GB/s	204.8 GB/s
TDP	25 W (MODE_25W)	50 W (MODE_50W)
AI Perf.	157 TOPS	275 TOPS
Jetpack	6.2.1	6.2.1

These techniques have been extensively explored for LLMs. Nevertheless, their extension to VLMs is less straightforward. VLMs combine multiple heterogeneous components, including a vision encoder, a cross-modal projector, and an LLM backbone. Because these components differ in size, function, and sensitivity to numerical precision, uniform quantization may not provide the best trade-off between efficiency and accuracy cost (Sharshar et al., 2025; Xue et al., 2025; Li et al., 2025). This motivates component-wise mixed-precision quantization, where different bit-widths are assigned to different VLM components based on their quantization sensitivity and deployment requirements.

3. Experimental Setup

3.1. Hardware Setup

The Jetson Orin family is an edge AI platform built on NVIDIA Ampere architecture. It offers multiple memory configurations, such as the NX (16 GB, LPDDR5) and AGX (64 GB, LPDDR5). Even with the same quantization configuration, the latency profile can vary across hardware platforms depending on the level of optimization of the kernels. INT4 quantization on certain platforms, for instance, involves dequantization overhead when using BitsAndBytes. If native INT4 operation support is limited in the Ampere architecture of Jetson Orin, the TPOT may increase (Xu et al., 2024). This explains why VRAM savings do not necessarily correspond to latency improvements, as confirmed by the data presented in Section 4.3.

The Jetson Orin family comprises the Jetson Orin Nano, NX, and AGX. The Jetson Orin Nano was excluded from the experimental set because it consistently encountered out-of-memory (OOM) failures during VLM inference. Prior work has shown that these failures stem from the substantial peak memory overhead introduced during the quantized weight conversion stage in BitsAndBytes (Dettmers et al., 2022). This emphasizes the need to provide additional memory capacity and bandwidth at runtime, beyond that required to simply load the model parameters. Consequently, experiments were carried out only on the Jetson Orin NX and AGX platforms, as summarized in Table 3 (NVIDIA, 2026).

3.2. Models

This study focuses on five 3B-parameter models supported by the VLMEvalKit framework, which is also employed in the Open VLM Leaderboard (OpenCompass, 2026). These models were chosen based on their frequency of academic citation and prevalence in industrial applications. The main selection criteria are architectural diversity (SigLIP vs. non-SigLIP) (Zhai et al., 2023) and variability in the LLM backbones, as shown in Table 4 (Bai et al., 2025; Wu et al., 2024; Steiner et al., 2024; Li et al., 2024; Lv et al., 2024).

Algorithm 1 Component Latency & VRAM Profiling for sVLM

Input: Quantized VLM $\mathcal{M}_Q = \{V, P, L\}$, Evaluation Dataset D, Output token length N
Output: Accuracy Score, Peak VRAM, Latency components (L_V, L_P, L_{TPOT})

 # 1. Accuracy and Peak VRAM Measurement
ResetCUDAMemoryStats()
$Score \leftarrow$ VLMEvalKit_Evaluate$(\mathcal{M}_Q, D)$
$Peak_VRAM \leftarrow$ GetMaxMemoryAllocated()

 # 2. Latency Analysis
$I, X \leftarrow$ Sample(D) ▷ Image I, Text Prompt X

CUDASynchronize()
$t_0 \leftarrow$ GetCurrentTime()
$E_v \leftarrow \mathcal{M}_Q.V(I)$
CUDASynchronize()
$t_1 \leftarrow$ GetCurrentTime()
$L_V \leftarrow t_1 - t_0$
Vision Encoder Latency

$E_p \leftarrow \mathcal{M}_Q.P(E_v)$
CUDASynchronize()
$t_2 \leftarrow$ GetCurrentTime()
$L_P \leftarrow t_2 - t_1$ ▷ Projector Latency

$Inputs \leftarrow$ Concat$(E_p, \text{Embed}(X))$
$\mathcal{M}_Q.L.$Generate$(Inputs, \text{max_new_tokens} = 1)$
CUDASynchronize()
$t_3 \leftarrow$ GetCurrentTime()

$Out_N \leftarrow \mathcal{M}_Q.L.\text{G}(Inputs, \text{max_tokens} = N)$
CUDASynchronize()
$t_4 \leftarrow$ GetCurrentTime()

$Total_Decode_Time \leftarrow t_4 - t_3$
$L_{TPOT} \leftarrow Total_Decode_Time/(N-1)$ ▷ TPOT
return $Score, Peak_VRAM, L_V, L_P, L_{TPOT}$

3.3. Datasets

For evaluation, we use the multimodal large language model evaluation (MME) benchmark, which is the most widely adopted benchmark for VLM models. The MME scores each subtask by combining the ACC (accuracy per question) and ACC+ (granted when both questions for a single image are answered correctly). The overall maximum score is 2,800 points, calculated as the sum of the cognition and perception scores (Fu et al., 2023).

3.4. Quantization Methods

In this study, we applied 4-bit (INT4) quantization exclusively to the LLM to reduce VRAM consumption, given that the LLM typically represents more than 70% of the total parameters in most VLMs (Table 4). Vision-related components (the vision encoder and projector), which are more susceptible to degradation arising from information loss, were kept at INT8 or higher precision. This prevents performance degradation in terms of modality alignment performance. Based on this design, we defined six quantization configurations: three single-component quantization ablations (`cfg1`, `cfg2`, and `cfg4`), in which two components are fixed at FP16 and one component is quantized; and two composite quantizations (`cfg3` and `cfg5`), in which both components are quantized simultaneously. This design incorporates structural pairs that enable independent measurement of the limiting effects of each component and verification of the additivity in composite quantization configurations. The detailed experimental configurations are presented in Table 2.

3.5. Evaluation Metrics

We utilized the open-source VLMEvalKit framework for accuracy evaluation (Duan et al., 2024). The latency was assessed by decomposing it into three parts: vision encoding, projector, and LLM TPOT. The VRAM consumption was reported as the peak usage (GB). The pseudocode used for reproduction is provided in Algorithm 1.

Algorithm 1 is built to precisely pinpoint performance bottlenecks in edge environments. The peak VRAM indicates increases in temporary memory usage introduced by the BitsAndBytes quantization model during dequantization in the inference phase. It also reflects the actual hardware capacity. In addition, the latency was decomposed and measured around each component (vision encoder, projector, and LLM) by recording the execution times before and after each component. Notably, for the LLM, only TPOT was measured independently, while excluding the latency associated with generating the first token (the prefill phase).

Table 4. Detailed architectural specifications and parameter distributions across five multimodal models. All percentages represent the relative share of the total model parameters. For Kosmos-2.5, the output projection is tied to the input embedding, so no separate lm_head parameters are counted.

Model	Vision Encoder		Language Model		Projector	
	Arch	Params Details	Arch	Params Details	Arch	Params Details
Qwen3-VL-2B -Instruct (2.44B)	SigLIP-2	∼306M (12.6%) Linear: 302M Conv3d: 1.6M	Qwen3 (Dense/MoE)	∼1.72B (70.6%) lm_head: 311M (12.8%)	MLP-based Merger	∼100M (4.13%)
DeepSeek-VL2 -Tiny (3.37B)	SigLIP-So400m	∼428M (12.7%) Linear: 426M ViT: 0.8M	DeepSeek -MoE	∼2.93B (87.1%) MoE-Gate: 0.9M	2-layer MLP	∼7.53M (0.22%)
PaliGemma2-3B (3.6B)	SigLIP-So400m	∼412M (11.4%) Linear: 411M Conv2d: 0.7M	Gemma 2	∼2.61B (72.2%) lm_head: 592M (16.4%)	Linear Proj.	∼2.65M (0.07%)
LLaVA-OV-0.5B (1.03B)	SigLIP-So400m	∼397M (38.6%) Linear: 396M Conv2d: 0.7M	Qwen-2	∼494M (48.0%) lm_head: 136M (13.2%)	2-layer MLP	∼1.83M (0.18%)
Kosmos-2.5 (1.37B)	Pix2Struct -based ViT	∼512.8M (37.31%) Linear: 512M	Decoder-only Trans.	∼846.9M (61.61%) lm_head: tied	Resampler	∼14.9M (1.09%)

4. Experimental Results

In this section, we explore the impact of component-wise quantization of the sVLMs on the performance and hardware profiles in edge SoC environments, such as the Jetson Orin NX/AGX, by examining five research questions (RQs).

4.1. RQ1: [Scale and Architecture Sensitivity] For different backbone architectures and model scales, how does INT4 quantization of the LLM backbone affect the change in MME accuracy?

The accuracy variations for each model ($\Delta MME = cfg_N - base$) from the single-component ablations (`cfg1`: LLM INT4; `cfg2`: Projector INT8; and `cfg4`: Vision INT8) are summarized in Table 5. The experiments revealed a complex interplay between parameter scale, LLM architecture, and INT4 quantization sensitivity. While the severe degradation in ultra-small models initially appears to align with a scale-dependent trend, a cross-architectural comparison reveals that algorithmic structure dominates over mere parameter scale. Consequently, the empirical evidence rejects the original hypothesis, H1. Reflecting the interaction between architecture and scale, we revise H1 into ($H1_{rev}$): The quantization sensitivity of sub-3B VLMs is determined by the underlying structural paradigm (MoE vs. Dense), with parameter scale acting as an aggravating factor only within a homogeneous architecture family.

Table 5. ΔMME comparison across models under `cfg1`.

Model	# LLM Params (B)	MoE	ΔMME (NX)	ΔMME (AGX)
Qwen3-VL	1.72	**O**	+56.04	+101.88
DeepSeek-VL2	2.93	**O**	+39.50	+48.18
PaliGemma2	2.61	**X**	-62.12	-59.35
LLaVA-OV	**0.49**	**X**	**-220.02**	**-156.08**

Key Findings:

- **Architecture Dominance Over Scale Sensitivity (Falsification of H1).** Although the substantial performance decline of LLaVA-OV (Qwen2-0.5B, -220.02 points) initially suggests a scale-dependent vulnerability, this hypothesis is obscured by architectural variables. Notably, the larger model, PaliGemma2 (2.61B), shows a significant decline of 62.12 points, while the smaller MoE-based Qwen3 (1.72B) demonstrates an enhancement in accuracy, gaining 56.04 points. This contradiction rejects H1 based exclusively on scale.
- **Impact of Architectural Bifurcation on INT4 Accuracy of LLM.** The LLM architecture fundamentally dictates the impact of `cfg1` on accuracy. Qwen3 (MoE-based, +56.04) and DeepSeek-VL2 (MoE-based, +39.50), achieve consistent improvements across all platforms, whereas the dense PaliGemma2 (Gemma-2B, -62.12) and LLaVA-OV (Qwen2-0.5B, -220.02) suffer pronounced performance declines. This pattern supports a reformulated $H1_{rev}$, aligning with prior studies that identify MoE models as notably resilient to severe quantization noise (Frantar & Alistarh, 2023).

4.2. RQ2: [Architectural Latency Anomaly] Does vision INT8 quantization have a significantly different impact on the latency of vision encoding for SigLIP-based and non-SigLIP architectures? Is this latency change independent of the accuracy changes?

The INT8 quantization (`cfg4`) of the vision encoder resulted in increased latency, which was largely unaffected by changes in the accuracy of the models based on the SigLIP architecture. The latency increase ratio (Table 6) is calculated as $ratio = cfg4/cfg0$. This ratio was rounded to three decimal places.

Table 6. Impact of vision INT8 quantization on vision encoding latency (ms).

Model	Vision Encoder	Jetson NX			Jetson AGX			ΔMME (NX)
		cfg0	cfg4	Ratio ↓	cfg0	cfg4	Ratio ↓	
PG2	**SigLIP-So400m**	311.9	1,203.5	×**3.86**	98.5	459.1	×**4.66**	+7.3
DS-VL2	**SigLIP-So400m**	1,423.3	4,297.3	×3.02	903.5	2,810.7	×3.11	-14.1
LV-OV	**SigLIP-So400m**	719.8	1,879.1	×2.61	381.2	925.1	×2.43	+7.4
QW3-VL	**SigLIP-2**	394.6	684.2	×1.73	160.6	403.3	×2.51	-48.3
Kosmos-2.5	**Pix2 Struct**	11,029.2	12,644.2	×1.15	2,903.6	3,479.4	×1.20	+4.2

Note: Latency is measured in milliseconds (ms).

Table 7. Impact of LLM INT4 quantization (`cfg1`) on VRAM usage and token generation speed (TPOT) relative to `cfg0` (baseline).

	Models	VRAM (GB) ↓			TPOT (ms) ↓		
		cfg0	cfg1	Reduct.	cfg0	cfg1	Incr.
NX	Qwen3-VL	4.06	2.13	**-47.5%**	111.1	173.1	+55.8%
	PaliGemma2	5.85	3.13	-46.5%	166.0	183.3	+10.4%
	LLaVA-OV	1.89	1.43	-24.3%	96.5	143.9	+49.1%
AGX	Qwen3-VL	4.06	2.13	**-47.5%**	136.2	212.4	+55.9%
	PaliGemma2	5.85	3.18	-45.6%	137.9	182.1	+32.0%
	LLaVA-OV	2.25	1.76	-21.8%	90.5	141.4	+56.3%

Key Findings:

- **Sharp Increase in Vision INT8 Latency of SigLIP-So400m-Based Models.** When `cfg4` is applied, the latency of the vision encoders of the three SigLIP-So400m-based models (PaliGemma2, DeepSeek-VL2, and LLaVA-OV) increases sharply (by a factor of 2.43-4.66), compared with that (1.73-2.51-fold) for the SigLIP-2-based Qwen3-VL. The Kosmos-2.5 model with a non-SigLIP architecture exhibits a smaller increase (1.15-1.20-fold). These results support the latency anomaly phenomenon specific to the SigLIP-BitsAndBytes-Ampere interaction established in H2.
- **Independence of Accuracy Changes.** This latency increase occurs regardless of changes in accuracy. For PaliGemma2 and LLaVA-OV, `cfg4` caused Δ MME to increase by 7.3 and 7.4, respectively, indicating a slight improvement in accuracy. However, the vision encoding latency increased by factors of 3.86 and 2.61, respectively. Because there was no loss in accuracy, the intuitive judgment that vision INT8 was efficient overlooks the latency cost. This error was confirmed through actual measurements.
- **Hardware-Software Stack Incompatibility.** This incompatibility arises when the BitsAndBytes INT8 kernel fails to efficiently accelerate the computational patterns of the SigLIP vision transformer on the Jetson Orin Ampere architecture (Cai et al., 2023). Although the Pix2Struct-style Kosmos-2.5 encoder incurs the highest absolute vision latency due to its variable-resolution patching, the relative INT8 overhead remains small precisely because its non-SigLIP computation pattern avoids the kernel fragmentation that BitsAndBytes induces on SigLIP encoders. We verify this with a non-SigLIP control under the same isolated vision-path protocol in Appendix E.

4.3. RQ3: [Resource and Accuracy Trade-offs] When applying LLM INT4 quantization, does the VRAM reduction translate into an actual improvement in the token generation latency (TPOT), or is it offset by the dequantization overhead?

Although INT4 quantization of LLMs significantly lowers the VRAM usage, it also reduces the token generation speed due to the extra computational overhead for dequantization. The formulas used to calculate the percentage change of VRAM and TPOT (Table 7) are as follows:

$$\Delta X = \frac{X_{\text{cfg1}} - X_{\text{cfg0}}}{X_{\text{cfg0}}} \times 100(\%), \quad X \in \{VRAM, TPOT\}$$

Table 8. Energy consumption (joules) and rate of increase owing to `cfg1` quantization across Jetson platforms.

Models	NX			AGX		
	cfg0	cfg1	Incr. ↓	cfg0	cfg1	Incr. ↓
Qwen3-VL	12.31	19.04	**+54.7%**	4.90	7.12	**+45.3%**
PaliGemma2	35.98	45.95	+27.7%	13.31	16.13	+21.2%
LLaVA-OV	21.47	21.87	+1.9%	21.55	21.86	+1.4%

Note: Energy consumption is measured in Joules (J).

Key Findings:

- **Trade-off between VRAM Reduction and TPOT Increase.** LLM INT4 (`cfg1`) lowers the VRAM usage by 21.8–47.5% but raises the TPOT by 10.4–56.3% across all platforms (Table 7). These empirical results demonstrate that within the Jetson Orin Ampere architecture, the dequantization overhead introduced by BitsAndBytes outweighs the native computation speedup provided by INT4 (Xu et al., 2024). This provides strong evidence supporting H3 for balancing resource savings and inference efficiency. Consequently, LLM INT4 implemented via BitsAndBytes should be viewed purely as a VRAM-saving approach, rather than as a technique for reducing latency.
- **Paradoxical Increase in Energy Consumption.** Despite the decrease in VRAM usage, Qwen3-VL increased energy consumption substantially, i.e., by 54.7% for NX and 45.3% for AGX. This increase results from the longer time required for the dequantization operations, which in turn increases power usage. In contrast, LLaVA-OV incurred only a slight increase in energy usage (+1.4% to +1.9%). This indicates that

the computational cost of dequantization is relatively low for small-scale LLMs (0.49B).

4.4. RQ4: [Interaction Dynamics] Can the accuracy degradation caused by composite quantization configurations (`cfg3`: Projector INT8 + LLM INT4 and `cfg5`: Vision INT8 + LLM INT4) be predicted by adding the corresponding single-component ablation results, or is a non-additive pattern observed?

Two configuration pairs are examined to assess additivity:

- cfg3: $\Delta MME(cfg3)$ vs. $\Delta MME(cfg1)$ + $\Delta MME(cfg2)$
- cfg5: $\Delta MME(cfg5)$ vs. $\Delta MME(cfg1)$ + $\Delta MME(cfg4)$

We assess additivity separately for the two composite configurations. For `cfg3`, the observed degradation is nearly additive across all models, closely matching the linear sum of the individual component errors. In contrast, `cfg5` deviates from additivity in a model-specific manner. These results lead us to interpret H4 in a more targeted form: composite quantization is near-additive for `cfg3`, and the hypothesized non-additivity emerges in `cfg5`, in an architecture-dependent manner.

Table 9. Additivity analysis: comparison of expected sum of individual quantization impacts (Δcfg1 + Δcfg2) with the actual combined impact (Δcfg3) in NX.

Model	ΔMME (cfg1)	ΔMME (cfg2)	Expected Sum	Actual Δcfg3	Residual (cfg3-Exp)
Qwen3-VL	+56.04	-31.89	+24.15	+24.57	0.42
DeepSeek-VL2	+39.50	+5.86	+45.36	+41.80	-3.56
PaliGemma2	-62.12	+2.67	-59.45	-60.77	-1.32
LLaVA-OV	-220.02	-1.42	-221.44	-224.24	+2.80
Kosmos-2.5	-10.54	+1.97	-8.57	-8.38	+0.19

Table 10. Additivity analysis: comparison of expected sum of individual quantization impacts (Δcfg1 + Δcfg4) with the actual combined impact (Δcfg5) in NX.

Model	ΔMME (cfg1)	ΔMME (cfg4)	Expected Sum	Actual Δcfg5	Residual (cfg5-Exp)
Qwen3-VL	+56.04	-48.27	+7.77	+7.32	-0.45
DeepSeek-VL2	+39.50	-14.14	+25.36	+33.39	+8.03
PaliGemma2	-62.12	+7.27	-54.85	-71.18	-16.33
LLaVA-OV	-220.02	+7.37	-212.65	-212.75	-0.10
Kosmos-2.5	-10.54	+4.18	-6.36	-5.63	+0.73

Key Findings:

- **Overall Near-Additivity of `cfg3` (Projector + LLM).** For all models, the `cfg3` residuals fall within a ± 4-point range, demonstrating nearly additive behavior, as shown in Table 9. This indicates minimal interference between the Projector INT8 and LLM INT4 components and confirms that their independent effects overlap linearly.
- **Model-Specific Non-Additivity of `cfg5` (Vision + LLM).** The behavior of `cfg5` is non-uniform across models, showing distinct patterns for different architectures. In PaliGemma2, an additional accuracy degradation of 16.33 (NX) is observed in the residuals compared to the expected sum, whereas DeepSeek-VL2 exhibits a super-additive pattern (-8.03), as shown in Table 10. These results suggest that the modality-alignment pathways between the vision encoder and the LLM differ across architectures, underscoring the need for prior architecture-specific evaluation for `cfg5`.

4.5. RQ5: [Hardware Transferability] Is the ordering of the model accuracy across different quantization settings the same on both the Jetson Orin NX and AGX platforms? Are the latency and energy characteristics dependent on the specific platform?

The empirical results demonstrated that the accuracy rankings are strictly preserved across heterogeneous edge platforms regardless of the quantization configuration, thereby obviating the need for hardware-specific accuracy revalidation during deployment (Table 11).

Table 11. Ranking of evaluated models based on MME total scores across Jetson Orin NX and AGX platforms.

Rank	Jetson Orin NX (Score)	Jetson Orin AGX (Score)
1	Qwen3-VL-2B (2021.78)	Qwen3-VL-2B (1983.32)
2	DeepSeek-VL2 (1911.12)	DeepSeek-VL2 (1916.71)
3	PaliGemma2 (1661.01)	PaliGemma2 (1658.10)
4	LLaVA-OV-0.5B (1361.83)	LLaVA-OV-0.5B (1369.73)
5	Kosmos-2.5 (662.30)	Kosmos-2.5 (662.05)

In contrast, energy optimization requires a granular understanding of how architectural bottlenecks interact with platform-specific bandwidth constraints. To quantify this, we adopt the IPJ metric, calculated as the normalized MME accuracy per unit of energy consumed (in Joules) (Saad-Falcon et al., 2025):

$$IPJ(m,h) = \frac{\mathbb{E}_q[acc(m,q)]}{\mathbb{E}_q[P(m,h,q) \cdot \tau(m,h,q)]}$$

where $acc(m,q)$ denotes the normalized MME score ($Score/2800$) for a given model m and processed query q. The denominator represents the energy consumption; specifically, the average power $P(m,h,q)$ is empirically measured using NVIDIA's `tegrastats` utility at 50ms sampling intervals, and the total inference latency $\tau(m,h,q)$ spans the end-to-end model execution, including vision encoding, projector mapping, the prefill phase, and sequential token generation.

Table 12. Comparison of intelligence-per-joule (IPJ) using `cfg0` and `cfg1` on Jetson Orin platforms.

Models	cfg0		cfg1	
	NX	AGX	NX	AGX
Qwen3-VL	5.87	**14.45**	3.90	**10.38**
DeepSeek-VL2	1.37	**3.03**	0.76	**1.72**
PaliGemma2	1.65	**4.45**	1.25	**3.54**
LLaVA-OV	**2.27**	2.25	1.86	**1.98**
Kosmos-2.5	0.12	**0.37**	0.11	**0.36**

Note: IPJ values represent accuracy per Joule (Saad-Falcon et al., 2025).

Key Findings:

- **Platform-Invariant Accuracy Ranking.** The experimental results confirmed that the MME accuracy ranking by model was completely consistent across all quantization configurations for the two platforms, Jetson NX and AGX, in the order: Qwen3-VL > DeepSeek-VL2 > PaliGemma2 > LLaVA-OV > Kosmos-2.5. These results strongly support hypothesis H5, demonstrating that the architecture of the algorithm has a greater effect on the sensitivity of the models to the quantization settings than the available hardware resources.

- **Asymmetric Power Scaling of Power Consumption Profiles.** Even for the same quantization configuration, NX (TDP 25W) and AGX (TDP 50W) exhibit different power consumption patterns. As shown in Table 11, AGX consistently has a lower average power consumption than NX (ratio 0.65–0.70) on Qwen3-VL. However, on PaliGemma2, LLaVA-OV, and Kosmos-2.5, AGX has a higher average power consumption than NX (ratio 1.02–1.24). These results suggest that the higher TDP (50 W) of AGX does not always lead to higher energy consumption. For Qwen3-VL, the memory bandwidth of AGX is twice as wide (204.8 GB/s), which alleviates computational bottlenecks and results in a lower average power consumption.

- **Platform Specificity and Deployment Strategy for Energy Efficiency (IPJ).** In contrast to the accuracy of task execution, the energy efficiency is highly sensitive to specific model-platform combinations. Aligning with earlier work highlighting the importance of hardware advances (Saad-Falcon et al., 2025), the present results indicate that Qwen3-VL delivers up to $2.5\times$ higher IPJ on the AGX platform than on the NX, with some models surpassing a $3\times$ gain. Together, these results reinforce the concept that achieving peak efficiency for edge-AI requires an integrated HW-SW co-design strategy, rather than relying solely on standalone model compression.

5. Conclusion

This study provides important insights into the interaction between the algorithm architecture and hardware deployment by systematically analyzing vision-language models with fewer than 3 billion parameters on heterogeneous edge platforms. The technical implications of these results are as follows: **First,** the architecture has a more profound impact on the quantization stability than the model size. Hypothesis H1 predicted severe performance degradation in smaller models. However, consistent performance improvements were observed with MoE-based architectures such as Qwen3-VL. These improvements include an increase of 56.04 in the MME on the NX platform. This suggests that sparse activation functions are effective in localizing and mitigating the propagation of INT4 quantization noise. This provides a clear advantage over dense backbones, such as LLaVA-OV, which exhibited a significant performance decline of 220.02 points. **Second,** we confirmed significant hardware-software stack friction in SigLIP-based encoders using BitsAndBytes. Although the accuracy remains consistent, the vision encoding latency increases sharply (by a factor of up to 4.66), suggesting that standard quantization kernels (e.g., BitsAndBytes) are not optimized for the specific computational patterns of SigLIP on Ampere-based edge SoCs. This underscores the need to transition from standalone model compression approaches to hardware-software co-design strategies.

Limitations and Future Research. This study presents a modular framework for PTQ sensitivity analysis, but it is limited to the BitsAndBytes framework. Future research should expand the analysis to include new hardware-native quantization formats, such as FP8, and should evaluate the impact of weight-activation quantization (W8A8) to further reduce the dequantization overhead. Furthermore, exploring automatic precision allocation through neural architecture search (NAS) could enhance the modality-specific deployment guidelines presented in this study.

Software

In support of open science and reproducibility, we have released an end-to-end profiling tool. This tool includes a modular quantization framework. The code and related metadata are available in the following repository: `https://anonymous.4open.science/r/EdgeQuant-VLMEvalKit/README.md`

Acknowledgements

This work was supported by ETRI grant funded by Korean government. [26ZK1100, Honam region strategic industries(optical convergence, energy, AI, etc.) technology advancement and commercialization]

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Cai, H., Li, J., Hu, M., Gan, C., and Han, S. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 17302–17313, 2023.

Chen, L., Li, J., Dong, X., Zhang, P., He, C., Wang, J., Zhao, F., and Lin, D. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pp. 370–387. Springer, 2024.

Chu, X., Qiao, L., Lin, X., Xu, S., Yang, Y., Hu, Y., Wei, F., Zhang, X., Zhang, B., Wei, X., et al. Mobilevlm: A fast, strong and open vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35:30318–30332, 2022.

Duan, H., Yang, J., Qiao, Y., Fang, X., Chen, L., Liu, Y., Dong, X., Zang, Y., Zhang, P., Wang, J., et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 11198–11201, 2024.

Frantar, E. and Alistarh, D. Qmoe: Practical sub-1-bit compression of trillion-parameter models. *arXiv preprint arXiv:2310.16795*, 2023.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Kim, S., Gholami, A., Yao, Z., Lee, N., Wang, P., Nrusimha, A., Zhai, B., Gao, T., Mahoney, M. W., and Keutzer, K. Integer-only zero-shot quantization for efficient speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4288–4292. IEEE, 2022.

Lee, J., Park, S., Kwon, J., Oh, J., and Kwon, Y. Exploring the trade-offs: Quantization methods, task difficulty, and model size in large language models from edge to giant. *arXiv preprint arXiv:2409.11055*, 2024.

Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Li, S., Hu, Y., Ning, X., Liu, X., Hong, K., Jia, X., Li, X., Yan, Y., Ran, P., Dai, G., et al. Mbq: Modality-balanced quantization for large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4167–4177, 2025.

Lin, B., Tang, Z., Ye, Y., Huang, J., Zhang, J., Pang, Y., Jin, P., Ning, M., Luo, J., and Yuan, L. Moe-llava: Mixture of experts for large vision-language models. *IEEE Transactions on Multimedia*, 2026.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100, 2024.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Lv, T., Huang, Y., Chen, J., Zhao, Y., Jia, Y., Cui, L., Ma, S., Chang, Y., Huang, S., Wang, W., Dong, L., Luo, W., Wu, S., Wang, G., Zhang, C., and Wei, F. Kosmos-2.5: A multimodal literate model. *arXiv preprint arXiv:2309.11419*, 2024.

NVIDIA. Nvidia jetson orin series. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/, 2026. Accessed: 2026-05-06.

OpenCompass. Open vlm leaderboard. https://huggingface.co/spaces/opencompass/open_vlm_leaderboard, 2026. Accessed: 2026-05-06.

Saad-Falcon, J., Narayan, A., Akengin, H. O., Griffin, J., Shandilya, H., Lafuente, A. G., Goel, M., Joseph, R., Natarajan, S., Guha, E. K., et al. Intelligence per watt: Measuring intelligence efficiency of local ai. *arXiv preprint arXiv:2511.07885*, 2025.

Sharshar, A., Khan, L. U., Ullah, W., and Guizani, M. Vision-language models for edge networks: A comprehensive survey. *IEEE Internet of Things Journal*, 2025.

Steiner, A., Pinto, A. S., Tschannen, M., Keysers, D., Wang, X., Bitton, Y., Gritsenko, A., Minderer, M., Sherbondy, A., Long, S., et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

Sun, Q., Cui, Y., Zhang, X., Zhang, F., Yu, Q., Wang, Y., Rao, Y., Liu, J., Huang, T., and Wang, X. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14398–14409, 2024.

Wu, Z., Chen, X., Pan, Z., Liu, X., Liu, W., Dai, D., Gao, H., Ma, Y., Wu, C., Wang, B., et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pp. 38087–38099. PMLR, 2023.

Xu, M., Yin, W., Cai, D., Yi, R., Xu, D., Wang, Q., Wu, B., Zhao, Y., Yang, C., Wang, S., et al. A survey of resource-efficient llm and multimodal foundation models. *arXiv preprint arXiv:2401.08092*, 2024.

Xue, Y., Huang, Y., Shao, J., and Zhang, J. Vlmq: Efficient post-training quantization for large vision-language models via hessian augmentation. *arXiv preprint arXiv:2508.03351*, 2025.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Zhou, B., Hu, Y., Weng, X., Jia, J., Luo, J., Liu, X., Wu, J., and Huang, L. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024.

A. Experimental Setup

To support reproducibility, we report the generation hyperparameters used for each VLM wrapper. Model weights are not fine-tuned, and all evaluations share identical VLMEvalKit benchmark prompts, prompt templates, and the same exact-matching answer-extraction protocol, applied uniformly across every model and every quantization configuration. All five models are evaluated with greedy decoding (`do_sample=False`); the decoding configuration is held fixed across all quantization settings, so within-model comparisons (ΔMME) are not confounded by decoding differences.

Table 13. Generation configuration of each VLM wrapper. All models use greedy decoding (`do_sample=False`); sampling parameters (temperature, top-p, top-k) are inapplicable and omitted. Settings are held fixed across all quantization configurations. †Kosmos-2.5-chat additionally applies `no_repeat_ngram_size=3` from its released configuration; given MME's short Yes/No answers, this has a negligible effect on accuracy.

Model	Decoding	Max New Tokens
Qwen3-VL-2B-Instruct	Greedy	16384
DeepSeek-VL2-Tiny	Greedy	2048
PaliGemma-3B-mix-448	Greedy	512
LLaVA-OneVision-0.5B	Greedy	2048
Kosmos-2.5-chat†	Greedy	1024

B. Comparison with OpenVLM Leaderboard

To verify our evaluation pipeline, we compare the MME Total Score produced by our measurement tool (in baseline mode) with the publicly reported scores on the Open VLM Leaderboard (OpenCompass, 2026). Table 14 demonstrates that, for the three models (DeepSeek-VL2-Tiny, PaliGemma-3B-mix-448, and LLaVA-OneVision-0.5B), our results closely match the leaderboard scores, supporting the robustness of our evaluation setup. For the remaining two models, the leaderboard and our experiments use different model versions: the leaderboard reports **Qwen2-VL-2B** and **Kosmos-2**, whereas our experiments use **Qwen3-VL-2B-Instruct** and **Kosmos-2.5-chat** respectively. The score differences for these two models therefore reflect the use of different model versions rather than discrepancies in our evaluation pipeline.

Table 14. Comparison of MME Total Score between the Open VLM Leaderboard and our evaluation tool (baseline mode) on two hardware platforms (AGX Orin and Jetson NX). †The leaderboard reports Qwen2-VL-2B, while our experiments use Qwen3-VL-2B-Instruct. ‡The leaderboard reports Kosmos-2, while our experiments use Kosmos-2.5-chat.

Model	Leaderboard MME Score	Ours (AGX) Baseline	Ours (NX) Baseline
Qwen2/3-VL-2B†	1899.1	1983.32	2021.78
DeepSeek-VL2-Tiny	1905.5	1916.71	1911.12
PaliGemma-3B-mix-448	1686.1	1658.10	1661.01
LLaVA-OneVision-0.5B	1392.9	1359.98	1361.83
Kosmos-2.5-chat‡	721.1	662.05	662.30

C. MME Total Score across Quantization Configurations

Figure 1 shows the MME total scores for all five quantization configurations (C0–C5) for each model, as measured on both hardware platforms. Each configuration is evaluated over $n \geq 3$ independent runs, and the outcomes are visualized using bar charts, with each bar representing the mean score across runs. The figure represents the full set of results that are summarized in Table 1.

D. Comprehensive Results across All Quantization Configurations

Table 15 presents the full set of per-model, per-configuration measurements for all five sVLMs, evaluated under seven quantization configurations (`cfg0–cfg6`) on both the Jetson Orin NX and Jetson Orin AGX platforms. For each configuration, we report the MME total score, peak VRAM usage (GB), end-to-end inference latency (ms), average power draw (W),

Table 15. Comprehensive Evaluation of Small VLMs across Quantization Configurations on Jetson Orin NX and AGX.

Model	Config	Jetson Orin NX					Jetson Orin AGX				
		MME Score	VRAM (GB)	Lat. (ms)	Power (W)	Energy (J)	MME Score	VRAM (GB)	Lat. (ms)	Power (W)	Energy (J)
Qwen3-VL-2B-Instruct	cfg0 (Base)	2021.78	4.06	908.11	13.33	12.44	1983.32	4.06	533.85	9.07	4.91
	cfg1	2077.82	2.13	1406.29	13.50	19.04	2069.05	2.13	744.09	9.76	7.08
	cfg2	1989.89	3.95	901.88	13.22	12.32	2005.34	3.95	537.52	8.60	4.74
	cfg3	2046.35	2.04	1445.82	13.41	19.42	2064.46	2.04	801.64	9.19	7.19
	cfg4	1973.51	3.77	1206.78	11.45	14.27	2028.41	3.77	738.98	7.53	5.73
	cfg5	2029.10	1.85	1739.66	12.25	21.43	2078.73	1.85	984.37	8.16	8.06
	cfg6 (All)	2055.09	1.76	1761.46	12.05	21.42	2055.09	1.75	760.82	10.65	8.06
DeepSeek-VL2-Tiny	cfg0 (Base)	1911.12	6.82	4602.58	10.85	50.05	1916.71	6.82	2945.87	7.63	22.57
	cfg1	1952.54	3.51	7883.96	11.55	91.07	1962.10	3.51	6302.31	6.53	41.23
	cfg2	1916.98	6.81	4603.13	10.87	50.14	1924.60	6.81	2958.24	7.67	22.78
	cfg3	1951.55	3.51	8015.77	11.45	91.78	1950.76	3.51	6213.22	6.57	40.91
	cfg4	1896.98	6.44	7482.60	9.95	74.65	1909.74	6.44	4233.30	7.56	32.24
	cfg5	1953.90	3.13	10865.29	10.69	116.20	1953.11	3.13	7627.90	6.63	50.84
	cfg6 (All)	1949.64	3.12	10885.15	10.67	116.26	1949.64	3.12	6098.56	7.58	46.67
PaliGemma-3B-mix-448	cfg0 (Base)	1661.01	5.85	2440.29	14.54	35.85	1658.10	5.85	835.30	15.53	13.28
	cfg1	1598.89	3.18	3127.39	14.55	45.83	1598.74	3.18	1033.13	15.33	16.09
	cfg2	1663.68	5.86	2452.48	14.53	36.01	1653.47	5.86	825.03	15.67	13.25
	cfg3	1600.24	3.18	3132.63	14.55	45.88	1600.37	3.18	1053.67	15.15	16.21
	cfg4	1668.28	5.49	3336.23	12.90	43.40	1656.97	5.49	1203.12	13.16	16.09
	cfg5	1589.83	2.81	4027.42	13.15	53.26	1589.19	2.81	1420.91	13.20	18.98
	cfg6 (All)	1588.56	2.81	4003.57	13.21	53.19	1588.56	2.80	1045.41	22.38	23.78
LLaVA-OneVision-0.5B	cfg0 (Base)	1361.83	1.89	1601.39	13.34	21.56	1369.73	2.25	1522.36	14.25	21.74
	cfg1	1141.81	1.45	1701.17	13.44	23.04	1206.15	1.76	1558.48	14.05	22.05
	cfg2	1360.41	1.88	1606.90	13.33	21.63	1363.28	2.26	1531.80	14.19	21.78
	cfg3	1137.59	1.45	1703.50	13.42	23.05	1207.48	1.76	1561.79	14.14	22.24
	cfg4	1369.20	1.57	2751.94	10.97	30.51	1365.30	1.91	2084.75	12.16	25.60
	cfg5	1149.08	1.13	2859.89	11.09	31.99	1215.45	1.40	2130.10	12.06	26.06
	cfg6 (All)	1215.94	1.12	2883.24	11.04	32.19	1215.94	1.40	1558.00	20.49	32.39
Kosmos-2.5-chat	cfg0 (Base)	662.30	6.45	16225.28	12.11	196.82	662.05	6.45	4181.44	14.79	62.17
	cfg1	651.76	5.59	16495.28	12.25	202.37	649.84	5.59	4300.84	14.74	63.68
	cfg2	664.27	6.45	16227.37	12.12	196.98	663.30	6.45	4196.07	14.90	62.84
	cfg3	653.92	5.58	16488.09	12.28	202.72	656.76	5.58	4334.14	14.93	65.03
	cfg4	666.48	5.98	17773.39	11.95	212.64	663.27	5.98	4748.06	14.25	67.98
	cfg5	656.67	5.13	18046.58	12.09	218.46	650.59	5.13	4864.09	14.10	68.89
	cfg6 (All)	652.51	5.12	18152.66	12.07	219.40	652.51	5.12	3721.97	24.38	91.22

Note: cfg0 (FP16 base), cfg1 (LLM INT4), cfg2 (Proj INT8), cfg3 (cfg1+cfg2), cfg4 (Vis INT8), cfg5 (cfg1+cfg4), and cfg6 (Vis INT8 + Proj INT8 + LLM INT4). Metrics default to: Latency (ms), Avg. Power (W), and Energy (J).

and total energy consumption (J). Configuration definitions follow the notation introduced in Table 2: `cfg0` is the FP16 baseline; `cfg1` applies LLM INT4 only; `cfg2` applies Projector INT8 only; `cfg3` combines `cfg1` and `cfg2`; `cfg4` applies Vision INT8 only; `cfg5` combines `cfg1` and `cfg4`; and `cfg6` applies all three components simultaneously (Vision INT8 + Projector INT8 + LLM INT4). This table constitutes the complete experimental record underlying the summarized results in Table 1. Across models, LLM INT4 (`cfg1`) consistently reduces VRAM by approximately 40–50% but increases latency on both platforms, while Vision INT8 (`cfg4`) incurs a disproportionate latency penalty for SigLIP-based encoders (PaliGemma-3B-mix-448 and DeepSeek-VL2-Tiny). Energy consumption generally scales with latency, and accuracy degradation under quantization varies substantially across model architectures and configuration types.

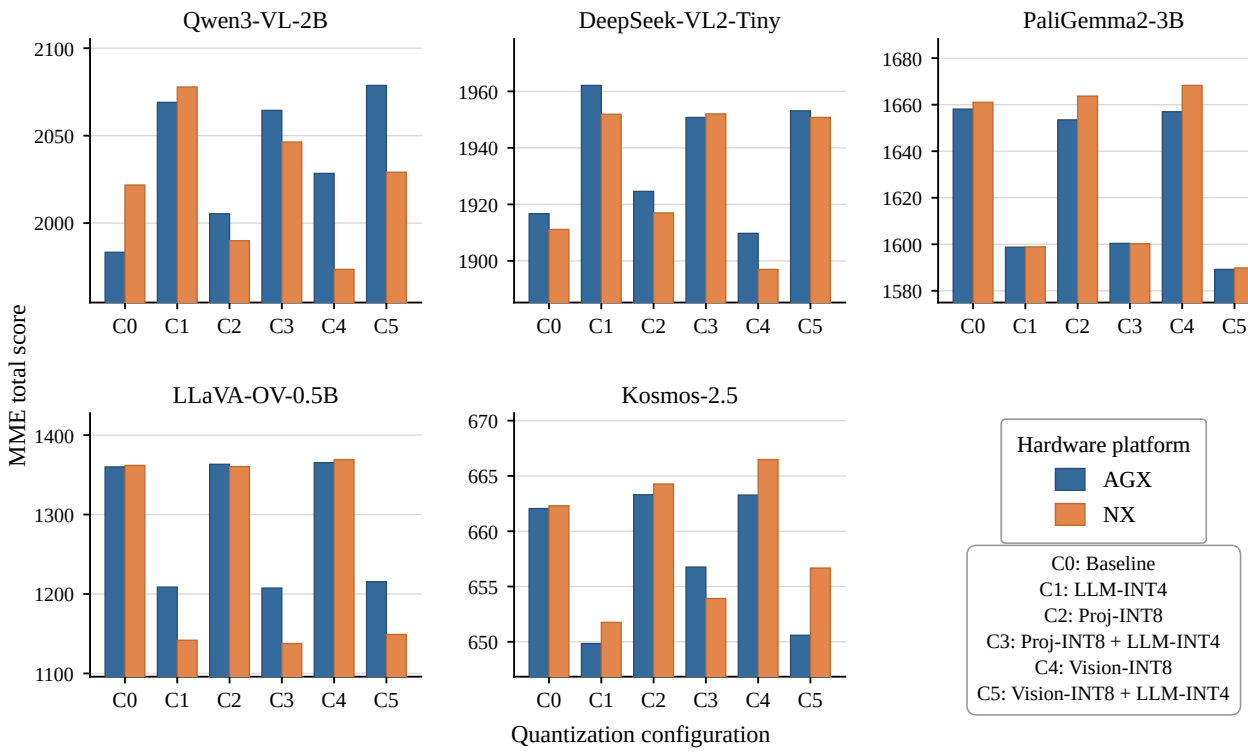

Figure 1. MME total score across quantization configurations (defined in the legend) for five sVLMs on the AGX Orin and Orin NX platforms. Bars show the mean for $n \geq 3$ samples; note that each subplot has its own independently scaled y-axis, which does not necessarily start at zero.

E. Isolated Vision-Path Profiling for the SigLIP INT8 Latency Anomaly

To isolate the vision INT8 anomaly from the host LLM, projector, tokenizer, and decoding loop, we profile the vision path alone using NVTX-annotated ranges with NVIDIA Nsight Systems (50 iterations; Nsight Systems 2026.3.1.157). For PaliGemma2 and Kosmos-2.5 we measure the outer `isolated_vision_encoder` range. For DeepSeek-VL2-Tiny, whose standalone vision call does not reproduce its multimodal path, we instead profile the generation-free `prepare_inputs_embeds` path. We note that this difference in measured scope means the DeepSeek-VL2-Tiny numbers are not strictly comparable to the encoder-only ranges of the other two models; we report it separately and interpret it qualitatively rather than in direct head-to-head comparison. As shown in Table 16, the large slowdown is reproduced for SigLIP-style encoders on both platforms (PaliGemma2: $4.93\times$ on AGX, $4.32\times$ on NX; DeepSeek-VL2-Tiny: $3.10\times$), confirming that the anomaly is not an artifact of the decoding loop or a single AGX-specific measurement. The non-SigLIP Kosmos-2.5 control, by contrast, slows by only $1.19\times$ (AGX) and $1.16\times$ (NX).

Nsight kernel summaries explain this contrast (Table 17). Under BitsAndBytes vision INT8, the PaliGemma2 vision path is restructured into repeated cycles of activation quantization, INT8 CUTLASS GEMM, dequantization, and synchronization. Quantization and dequantization kernels alone account for 22.8–25.7% of listed INT8 GPU kernel time (NX and AGX, respectively), accompanied by a $7.4\times$ rise in kernel launches and a sharp increase in memory copies. This points to a fragmented execution path rather than a faster GEMM substitution. The non-SigLIP Kosmos-2.5 control, already dominated by softmax, copy, and elementwise kernels, dilutes the same overhead to 6.0–6.5%. We restrict the kernel-level breakdown in Table 17 to these two models, as the latency results for DeepSeek-VL2-Tiny are reported only on AGX.

These results lead us to revise H2 toward a narrower, deployment-specific interpretation: the disproportionate slowdown arises from the interaction between SigLIP-style encoders, the BitsAndBytes INT8 execution path, and the Jetson Orin Ampere architecture, rather than from any inherent inefficiency of SigLIP or of INT8 vision quantization in general.

Table 16. Impact of BitsAndBytes vision INT8 quantization on isolated vision-encoding latency (ms), with a non-SigLIP Kosmos-2.5 control. Ratio = Vision INT8 / FP16.

Model	Plat.	Encoder	FP16 (ms)	INT8 (ms)	Ratio ↑
PaliGemma2	AGX	SigLIP-So400m	195.99	966.09	×4.93
PaliGemma2	NX	SigLIP-So400m	313.14	1353.53	×4.32
DeepSeek-VL2-Tiny	AGX	SigLIP-style	1145.81	3555.38	×3.10
Kosmos-2.5	AGX	Pix2Struct-style	7058.22	8373.13	×1.19
Kosmos-2.5	NX	Pix2Struct-style	10701.11	12428.37	×1.16

Table 17. Kernel-level breakdown of BitsAndBytes vision INT8 on the isolated vision path. Quant/dequant share is the fraction of listed INT8 GPU kernel time spent in `kInt8VectorQuant` (activation quantization) and `kdequant_mm_int32_fp16` (dequantization). Kernel-launch ratio is INT8 relative to FP16. DeepSeek-VL2-Tiny is omitted here because its latency is reported on AGX only (Table 14); for reference, its AGX quant/dequant share is 22.5%.

Model	Plat.	Encoder	Q/Dq share	Launches ↑
PaliGemma2	AGX	SigLIP	25.7%	7.4×
PaliGemma2	NX	SigLIP	22.8%	7.4×
Kosmos-2.5	AGX	Pix2Struct-style	6.5%	2.8×
Kosmos-2.5	NX	Pix2Struct-style	6.0%	2.8×

F. Non-BitsAndBytes AWQ Sanity Check

To examine whether the LLM INT4 latency trend reported in RQ3 (Section 4.3) is specific to the BitsAndBytes backend, we additionally evaluate a non-BitsAndBytes AWQ baseline on Qwen3-VL-2B-Instruct (Table 18). This experiment is intended as a sanity check rather than a full replacement for the component-wise BitsAndBytes study. We apply AWQ W4A16 exclusively to the text-decoder Linear layers, excluding the vision encoder, the visual merger, and the lm_head, with calibration on 512 held-out Flickr30k image-text samples. For the AWQ sanity-check run, we report median TPOT rather than relying only on mean end-to-end latency, because a small number of Qwen3-VL samples produced unusually long responses that would otherwise inflate the mean end-to-end latency. The AWQ baseline nearly preserves the MME accuracy of the FP16/BF16 baseline and, unlike BitsAndBytes INT4, does not reproduce the TPOT penalty, indicating that the TPOT increase under BitsAndBytes LLM INT4 is tied to the BitsAndBytes execution path in our runtime, rather than being an inherent property of 4-bit LLM quantization. However, AWQ does not reduce the peak inference VRAM in our current runtime. This suggests that peak allocation during multimodal inference is likely dominated by runtime activations, visual features, the KV cache, and the compressed-tensor execution workspace, rather than by the quantized weight storage alone. We therefore report AWQ as a non-BitsAndBytes sanity-check backend, not as a substitute for the component-wise BitsAndBytes analysis that forms the basis of this study.

Table 18. Non-BitsAndBytes AWQ sanity check on Qwen3-VL-2B-Instruct, compared with the FP16/BF16 baseline and BitsAndBytes (BnB) LLM INT4.

Model	Backend	Quant. comp.	Calib.	MME	Peak VRAM	Median TPOT
Qwen3-VL-2B	FP16/BF16	None	None	2021.78	4.06 GB	111.1 ms
Qwen3-VL-2B	BnB INT4	LLM	None	2077.82	2.13 GB	173.1 ms
Qwen3-VL-2B	AWQ W4A16	LLM	Flickr30k 512	2020.93	4.08 GB	107.5 ms