QVLA: NOT ALL CHANNELS ARE EQUAL IN VISION-LANGUAGE-ACTION MODEL'S QUANTIZATION

Yuhao Xu[1,3,4*], Yantai Yang[1,2*], Zhenyang Fan[1], Yufan Liu[3,4†], Yuming Li[5], Bing Li[3†], Zhipeng Zhang[1†]
[1]Shanghai Jiao Tong University; [2]Anyverse Dynamics; [3]CASIA; [4]University of Chinese Academy of Sciences; [5]Ant Group

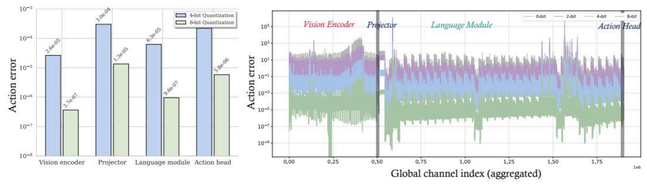
INTRODUCTION: THE VLA QUANTIZATION GAP

The Deployment Bottleneck

- Vision-Language-Action (VLA) models, such as OpenVLA, represent a significant leap in embodied intelligence but are hindered by immense computational demands, requiring over 14 GB of memory for a 7B parameter model.
- On standard robotic platforms like the NVIDIA Jetson AGX Orin, inference latency can reach hundreds of milliseconds, obstructing the real-time control loops necessary for safe physical interaction.
- While aggressive compression techniques like pruning and distillation exist, low-bit quantization specifically tailored for VLA architectures lacks systematic analysis.

Why LLM Methods Fail

- Naive application of quantization strategies from Large Language Models (LLMs), such as SmoothQuant, fails because VLAs output continuous action values for the physical world rather than passive text labels.
- In closed-loop robotic settings, minor quantization noise amplifies autoregressively over long horizons, leading to catastrophic failures like unstable grasps and trajectory deviations.
- Our analysis reveals significant intra-layer channel heterogeneity: while vision encoders are robust due to redundant data, the cross-modal projector and action head are acutely sensitive, serving as the critical nexus for action translation.



(a) Quantization Results Across Modules (b) Quantization Results Across Channels

METHOD: ACTION-SENSITIVITY GUIDED QUANTIZATION

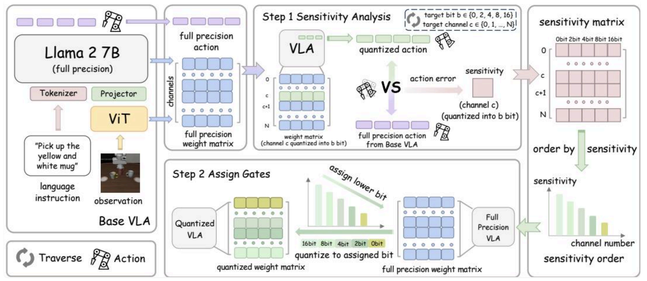
Action-Centric Formulation

- QVLA anchors the quantization objective in the action space. We minimize the KL divergence between the original policy Π_θ and the quantized policy $\Pi_{Q(\theta)}$:

$$\mathrm{Q}^* = \arg\min_{Q} \quad \mathbb{E}\left[D_{KL}\left(\Pi_\theta(a_t|\mathcal{V}_t, p, \mathcal{H}_t) \parallel \Pi_{Q(\theta)}(\mathcal{A}_t|\mathcal{V}_t, p, \mathcal{H}_t)\right)\right]$$

- To account for error accumulation in autoregressive tasks, we define a cumulative sensitivity metric $S_{l,c}^{(b)}$ over the episode horizon T:

$$S_{l,c}^{(b)} = \mathbb{E}\left[\sum_{t=1}^{T} \tilde{\mathcal{A}}_{l,c}^{(b)}(\mathcal{V}_t, l) - \mathcal{A}^*(\mathcal{V}_t, l)_2\right]$$



Efficient Estimation & Optimization

- We approximate action deviation using a fast Taylor-series proxy based on the Jacobian norm, avoiding exhaustive computation:

$$\|\Delta \boldsymbol{A}\| \approx \|J_{\boldsymbol{A},\mathbf{X}_{l,c}}\| \cdot \|\Delta \mathbf{X}_{l,c}\|$$

- The bit allocation is solved as a constrained optimization problem to minimize total sensitivity under a global average budget $\bar{B}$:

$$\min_{\{b_{l,c}\}} \sum_{l,c} s_{l,c}^{(b_{l,c})} \quad \text{s. t.} \quad \frac{1}{N}\sum_{l,c} b_{l,c} \le \bar{B}$$

- A global greedy algorithm sequentially demotes channels from 16-bit down to $\{8, 4, 2, 0\}$ based on the sensitivity-to-bit ratio $\rho_{l,c}$, effectively unifying quantization and pruning:

$$\rho_{l,c} = \frac{s_{l,c}^{(b_{\text{lo}})} - s_{l,c}^{(b_{\text{hi}})}}{b_{\text{hi}} - b_{\text{lo}}}$$

EXPERIMENTAL RESULTS: LIBERO BENCHMARK

Superior Performance & Efficiency

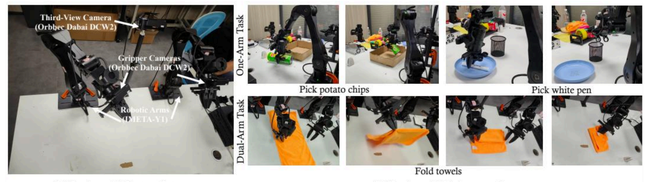
- Evaluated on the LIBERO benchmark, QVLA reduces VRAM usage to of the original model (compressing to ~7.0 GB) while retaining of the baseline performance.
- In the challenging W4A16 setting, OpenVLA with QVLA maintains a success rate (+0.0% vs FP16), significantly outperforming the weight-only method AWQ, which drops to 70.8% (-4.7%).
- For aggressive weight-activation quantization (W4A4), QVLA retains 99.3% relative performance (0.5% drop), whereas SmoothQuant suffers a massive 13.3% degradation and OmniQuant drops by 3.2%.

Model	Setting	Method	Spatial	Object	Goal	Long	Avg ↑	Δ	Mem. (GB) ↓	Speedup ↑
OpenVLA	FP Model	-	84.7%	88.4%	79.2%	53.7%	76.5%	-	15.2	1×
	W8A8	SmoothQuant	84.2%	87.8%	77.8%	53.2%	75.8%	-0.7%	7.4	1.40×
		OmniQuant	82.6%	86.2%	74.8%	51.7%	73.8%	-2.7%	7.8	1.26×
		QVLA	85.2%	88.0%	77.6%	54.2%	76.3%	-0.2%	7.1	1.42×
	W4A4	SmoothQuant	69.2%	73.2%	69.6%	40.9%	63.2%	-13.3%	4.7	1.52×
		OmniQuant	82.2%	85.4%	75.4%	50.3%	73.3%	-3.2%	5.4	1.43×
		QVLA	84.4%	87.6%	78.8%	53.0%	76.0%	-0.5%	4.3	1.47×
OpenVLA -OFT	FP Model	-	97.6%	98.4%	97.9%	94.5%	97.1%	-	15.4	1×
	W8A8	SmoothQuant	96.4%	97.8%	95.4%	94.3%	96.0%	-1.1%	7.7	1.41×
		OmniQuant	95.4%	96.2%	93.0%	92.6%	94.3%	-2.8%	8.0	1.30×
		QVLA	97.2%	98.2%	95.8%	94.3%	96.4%	-0.7%	7.2	1.36×
	W4A4	SmoothQuant	77.2%	70.0%	77.8%	68.6%	73.4%	-23.7%	4.9	1.53×
		OmniQuant	95.0%	94.4%	94.0%	92.0%	93.9%	-3.2%	5.7	1.37×
		QVLA	96.2%	97.6%	96.4%	93.8%	96.0%	-1.1%	4.5	1.49×

REAL-WORLD ROBOTIC DEPLOYMENT

System Setup

- We deployed the quantized model on a bimanual system featuring two IMETA-Y1 robotic arms and three Orbbec DaBai DCW2 cameras (two wrist-mounted, one global).
- The system executes complex manipulation tasks, including single-arm object picking (pen, chips) and dual-arm towel folding, driven by the quantized OpenVLA model.



(a) Real-world System Setup (b) Real-world Task examples

Method	Setting	One-Arm Task		Dual-Arm Task	Average	SpeedUp ↑
		Pick white pen	Pick potato chips	Fold towels		
π_0	-	8/10	7/10	4/10	63.3%	1.00×
QVLA	W8A16	8/10	6/10	5/10	63.3%	1.28×

Deployment Results

- In real-world trials, QVLA under W8A16 settings achieves a success rate, exactly matching the performance of the full-precision π_0 baseline.
- The optimized model delivers a inference speedup on a single consumer-grade NVIDIA RTX 4070 GPU, validating QVLA as a practical solution for deploying generalist VLA models on resource-constrained hardware.