SP-QAT: SPECTRAL-PRESERVING QUANTIZATION-AWARE TRAINING FOR EXTREME LOW-BIT VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

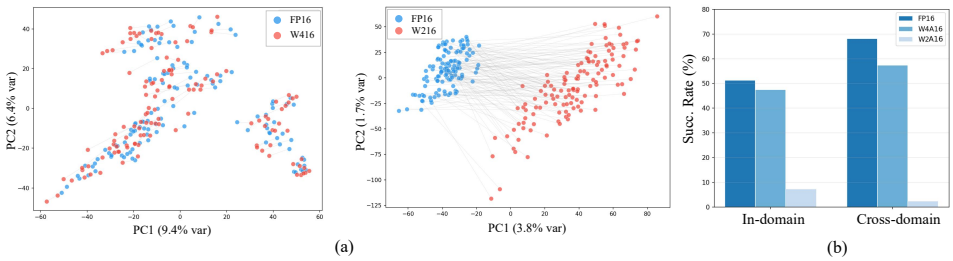
Vision-language-action (VLA) models are rapidly advancing and becoming central to embodied AI, yet their deployment on edge devices remains challenging. Since in VLA model, LLM backbone dominates memory consumption and inference latency, aggressive quantization of this LLM module presents a straightforward solution. Unfortunately, existing VLA quantization techniques mostly operate at 4-bit or higher bit widths and avoid fixed sub-4-bit quantization because of its catastrophic performance collapse, leaving it largely underexplored. To bridge this gap, we propose SP-QAT, an efficient quantization-aware training framework (QAT) that enables fixed 2-bit quantization of the Large Language Model (LLM) backbone in VLA, to our knowledge, for the first time. Through theoretical and experimental analysis, we identify that preserving the cognitive feature spectral subspace is the key to extreme low-bit quantization for the LLM backbone of VLAs. Therefore, we first constrain the weight-update subspace to preserve the pretrained model priors. Second, we propose Soft-Mask Gradients with Severity-Aware Scale Regulation to mitigate outlier-driven step size expansion from causing quantized weights to collapse into noise. Finally, we propose Difficulty-Aware Curriculum Learning with SNR-Guided Static Reweighting to emphasize hard samples and low-noise timesteps, thereby preserving the action-sensitive energy in the cognition feature spectrum. Extensive experiments on SIMPLER with CogACT and $\pi 0$ show SP-QAT only results in a small drop of $\leq 2.6\%$ compared to FP16 baselines, with 75%–84% memory savings and $1.6\times$ inference speedup. Code is available as supplementary material.

1 INTRODUCTION

Vision-Language-Action (VLA) models Ma et al. (2026) form a transformative paradigm for generalist robotic manipulation, unifying visual perception, multimodal reasoning and continuous action generation in an end-to-end policy. Diffusion- and flow-matching-based architectures Song et al. (2026); Hou et al. (2025) dominate this landscape, with representative models such as CogACT Li et al. (2024a) and π series Black et al. (2024); Intelligence et al. (2025). They typically consist of a vision encoder to process visual inputs, a massive LLM backbone Touvron et al. (2023) pretrained on web-scale data to provides powerful multimodal understanding, and a lightweight action head that generate continuous actions via iterative denoising or flow matching. Despite their impressive capabilities, these designs struggle with edge deployment on resource-constrained hardware. In both CogACT and π, the LLM takes over 85% of total memory (FP16) and accounts for nearly 90% of the FLOPs Yang et al. (2026), making aggressively quantizing the LLM backbone a natural remedy.

However, existing VLA quantization methods Zhang et al. (2026); Park et al. (2025) predominantly target 4-bit or higher bitwidth, and encounter severe performance collapse under fixed sub-4-bit quantization. While ActQuant Akbari et al. (2026) explores this regime, it incurs substantial success rate degradation at 2.5 BPW and fails to support unified bit-width quantization. As shown in Fig.1(a), compared with 4-bit quantization, 2-bit quantization drastically reduces representation bitwidth, and induces a severe drift in LLM representation space. Relying on limited calibration data and simplified loss surrogates, post-training quantization (PTQ) determines quantization parameters

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) (b)

Figure 1: (a) Visualization of cognition feature distributions under W4A16 and W2A16 quantization of ω-QVLA on CogACT. (b) Success Rate drop after quantization with ω-QVLA.

in one shot without task-level dynamic adjustment, making them less promising than QAT under ultra-low-bit quantization.

However, directly applying standard LLM QAT pipelines Esser et al. (2019); Bhalgat et al. (2020) to VLA backbones yields no benefits and instead leads to catastrophic failure in sub-4-bit quantization. To account for this phenomenon, we carry out a systematic diagnostic analysis. It shows that VLA action generation critically depends on the energy distribution across the singular spectrum of cognition features encoded by its LLM backbone, especially in action-sensitive subspaces.

Therefor we propose **SP-QAT**, an efficient QAT framework that preserves the action-sensitive energy concentration in the singular spectrum of cognition features under ultra-low-bit quantization. We first constrain weight updates to a low-rank subspace and jointly optimize them with quantization parameters, limiting QAT-induced training drift to preserve pretrained priors. Second, we propose *Soft-Mask Gradients with Severity-Aware Scale Regulation*, which redefines the gradient update scheme for weights and quantization parameters, mitigating outlier-driven step size expansion. By constraining weight drift and quantization noise, these two mechanisms prevent spectral flattening of the singular spectrum of cognition features, thereby preserving the internal concentration of action-sensitive energy. Finally, we propose *Difficulty-Aware Curriculum Learning with SNR-Guided Static Reweighting* emphasizing hard samples and low-noise timesteps to preserve the action-sensitive energy within the cognition-feature spectrum. Our contributions are as follows:

1) We systematically investigate why standard LSQ fails in fixed 2-bit quantization of VLA LLM backbones, offering theoretical and empirical guidance for QAT framework design.

2) We propose **SP-QAT**, a novel efficient QAT framework that enables fixed 2-bit LLM quantization for VLA for the first time, through preserving the action-sensitive energy concentration in the singular spectrum of cognition features under extreme low-bit quantization.

3) Extensive experiments on SIMPLER show that SP-QAT achieves state-of-the-art performance over CogACT and $\pi 0$. Under approximately **84%** memory savings, **in-domain** average closed-loop success **exceeds** FP16 baselines by **1%**, and **cross-domain** average success rate drop **within 5.7%**. Experiments on the Franka Panda arm further validate the real-world transferability of our method.

2 Related Work

2.1 Vision-Language-Action models

VLA models bridge high-level perception and low-level control by processing visual observations and language instructions through a visual encoder, an LLM, and an action head. They fall into two mainstream paradigms by action generation mechanism. Represented by RT-2 Brohan et al. (2023) and OpenVLA Kim et al. (2024), the first paradigm adopts autoregressive discrete-token prediction. While reusing the LLM token prediction pipeline, these early VLAs suffer error accumulation and discretization artifacts in high-frequency continuous control. To address these limitations, generative VLA paradigms model full continuous action distributions rather than discrete tokens. This branch includes two major subfamilies: diffusion-based and flow-matching policies. Diffusion-based VLA methods like Octo Team et al. (2024) and CogACT generate smooth action trajectories via iterative conditional denoising on action chunks. In contrast, flow-matching methods, represented by the OpenPi family and GR00T series Bjorck et al. (2025), learn continuous vector fields to map random noise to target trajectories, offering faster inference and better training stability. As VLAs grow larger, their LLM backbone dominates parameter count, memory footprint and inference cost. Extreme low-bit quantization for this LLM backbone is thus critical for onboard edge deployment.

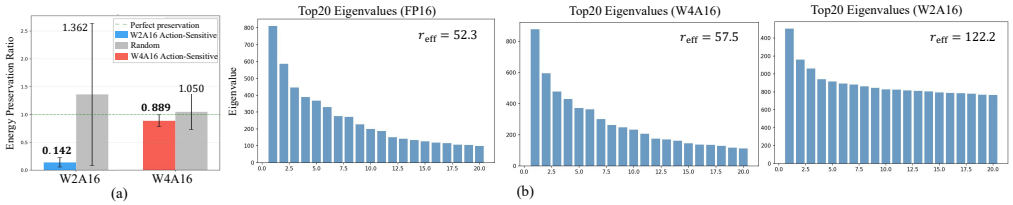


Figure 2: (a) Jacobian projection energy comparison between action-sensitive and random projections for W2A16-LSQ and W4A16-LSQ quantized models. (b) Singular value spectrum of cognition features under FP16, W4A16-LSQ and W2A16-LSQ.

2.2 Model Quantization

Model quantization compresses full-precision models into low-bit representations to reduce memory and speed up inference. Classified by whether training is required, PTQ incurs severe degradation at ultra-low fixed bitwidths without retraining, whereas QAT preserves accuracy via in-optimization quantization simulation. Although these techniques are mature for LLMs, extending them to VLA models face new challenges. In VLA, most existing works Xu et al. (2026b) target 4-bit or higher bitwidths. Represented by QuantVLA Zhang et al. (2026) and SQIL Park et al. (2025), the former is a PTQ method that recovers scales after quantization via attention-temperature matching and output-head balancing, while the latter realizes imitation-learning QAT via quantization-robust action distillation. Although ActQuant Akbari et al. (2026) explores sub-4-bit settings, it still relies on mixed precision and exhibits substantial success-rate degradation, dropping by 7% to 50% at 2.5 BPW and 2 BPW, even under in-domain evaluations. BitVLA Wang et al. (2025) builds a native 1.58-bit VLA from scratch with a ternary-weight LLM backbone and distillation-aware quantized vision encoder, which consists of re-pre-training and multi-stage post-training, incurring prohibitive compute cost. Moreover, it only supports 1.58-bit quantization and cannot generalize to other bitwidths.

3 Methodology

3.1 Preliminaries

Vision-Language-Action models. A VLA model always comprises 4 components: a vision encoder $\mathcal{E}$ that extracts visual features from raw observations $o_t \in \mathbb{R}^{H\times W\times 3}$, an MLP projector $\mathcal{P}$ that maps visual features to the embedding space of an LLM backbone $\mathcal{G}$, which processes the fused visual-language context, and an action head $\mathcal{H}$ that decodes the final control command $a_t \in \mathbb{R}^{d_a}$ from the LLM's output representations. The overall policy can be written as

$$\pi(a_t \mid o_t, l, s_t) = \mathcal{H}\big(\mathcal{G}\big(\mathcal{P}(\epsilon(o_t)),\, \psi(l),\, s_t\big)\big), \tag{1}$$

where l is the language instruction, s_t denotes optional proprioceptive states (included in $\pi 0$ but omitted in CogACT), and $\psi(\cdot)$ tokenizes the text. The action head $\mathcal{H}$ is often implemented as a diffusion-based conditional generative model (e.g., DiT Peebles & Xie (2023) or Flow Matching Lipman et al. (2022)). To present a unified view, let $\mathbf{x}_0 = a_t$ denote the clean action and define a forward diffusion process $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_t$ being a noise schedule. Both DiT and Flow Matching train a network $\mathcal{H}_\phi(\mathbf{x}_t, t, c)$, conditioned on the LLM context c and timestep t, to predict either the injected noise $\boldsymbol{\epsilon}$ (DiT) or the velocity field $\partial \mathbf{x}_t/\partial t$ (Flow Matching). Their training objectives can be unified as:

$$\mathcal{L}_{\text{gen}} = \mathbb{E}_{t\sim\mathcal{U}(0,1),\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\mathcal{H}_\phi(\mathbf{x}_t, t, c) - \mathcal{T}(\mathbf{x}_0, \boldsymbol{\epsilon})\right\|_2^2\right], \tag{2}$$

where $\mathcal{T}$ is the prediction target (noise or velocity).

Model Quantization. For a linear layer with weight $\mathbf{W} \in \mathbb{R}^{m\times n}$, we use a per-channel weight-only quantization scheme, assigning learnable step size $\boldsymbol{\delta} \in \mathbb{R}^m$ and bias $\boldsymbol{\beta} \in \mathbb{R}^m$ per output channel. Under asymmetric uniform quantization with bit-width b, quantized weight $\widehat{\mathbf{W}}$ is

$$\widehat{\mathbf{W}}_{i,:} = \delta_i \cdot \text{clip}\left(\left\lfloor \frac{\mathbf{W}_{i,:} - \beta_i}{\delta_i} \right\rceil, Q_n, Q_p\right) + \beta_i, \quad i = 1, \ldots, m, \tag{3}$$

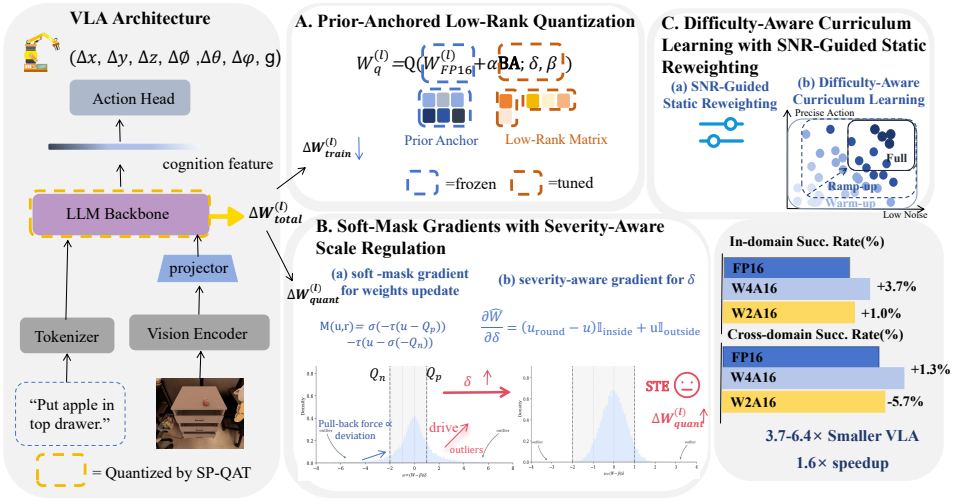


Figure 3: **Overview of SP-QAT**.This framework consists of three components. *Prior-Anchored Lor-Rank Quantization*: constrain weight updates to a low-rank subspace to limit QAT-induced training drift; *Soft-Mask Gradients with Severity-Aware Scale Regulation*: prevent outliers from forcing blind expansion of δ to reduce quantization-induced noise; *Difficulty-Aware Curriculum Learning with SNR-Guided Static Reweighting* to use SNR reweighting and difficulty curriculum to steer gradient quality toward low-noise and quantization-sensitive hard samples, preserving action-sensitive singular energy under ultra-low bit quantization.

where $Q_n = -2^{b-1}$, $Q_p = 2^{b-1} - 1$. To make the rounding function $\lfloor \cdot \rceil$ differentiable for QAT, straight-through estimator (STE) is commonly employed, defining the gradient as:

$$\left.\frac{\partial \widehat{W}_{i,j}}{\partial W_{i,j}}\right|_{\text{STE}} = \mathbf{1}\left\{Q_n \le u_{i,j} \le Q_p\right\}, \quad u_{i,j} = \frac{W_{i,j} - \beta_i}{\delta_i}. \tag{4}$$

Under the behavioral cloning paradigm, imitation learning QAT optimizes the quantized policy to match its full-precision counterpart by minimizing expected action discrepancy:

$$\widehat{\theta}^* = \arg\min_{\widehat{\theta}} \mathbb{E}_{(o,l,s)\sim\mathcal{D}}\Big[\mathcal{L}_{\text{act}}\big(\pi_\theta(a \mid o,l,s),\, \pi_{\widehat{\theta}}(a \mid o,l,s)\big)\Big], \tag{5}$$

where θ and $\widehat{\theta}$ denote full and quantized LLM parameters, respectively, and $\mathcal{L}_{\text{act}}$ is a distance metric on actions (e.g., mean squared error for continuous actions). Since $\mathcal{G}$ dominates the parameter count and computational cost, we quantize all linear layers of $\mathcal{G}$ (except LayerNorm and embeddings) and leave $\mathcal{E}$, $\mathcal{P}$, and $\mathcal{H}$ in full precision.

3.2 Revisiting the Failure of LSQ in Sub-4-bit VLA Quantization

Although QAT holds greater potential for extreme low-bit quantization, directly applying LSQ to VLA backbones exhibits catastrophic performance collapse under fixed 2-bit quantization. To understand this failure, we conduct a systematic diagnostic analysis, tracing backward from observable representational symptoms toward possible causes.

First, we inspect the singular spectra of cognition features produced by $\mathcal{G}$ under FP16, W4A16, and W2A16 settings. Formally, let the representation matrix be $\mathbf{H} \in \mathbb{R}^{J\times d}$ with SVD decomposition $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_d)$. We formalize the spectral energy concentration ratio for the top-k singular components as $E_k = \frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{i=1}^{d}\sigma_i^2}$ and effective rank by the entropy effective rank $r_{\text{eff}} = \exp\left(-\sum_{i=1}^{d} p_i \log p_i\right), p_i = \frac{\sigma_i^2}{\sum_{j=1}^{d}\sigma_j^2}$. Fig.3(b) shows that unlike FP16 and W4A16, W2A16 triggers a sharp expansion of r_{eff}, and σ_{20} is no longer clearly separated from mid-spectrum singular values. This substantially inflates the spectral tail and flattens the overall spectrum, where the exponential decay of σ_i is lost and energy leaks from a few dominant directions into many tail directions. To determine the functional consequence of such spectral pathology, we fix the

action-sensitive subspace $\mathbf{U}_s$ from the FP16 model and measure the energy preservation ratio $R = E^{\text{W2A16}}/E^{\text{FP16}}$ on both the action-sensitive subspace and random subspaces of the same dimension. Fig.3(a) shows that random subspaces retain $R_{\text{rand}} \approx 1.36$ of their FP16 energy, consistent with the spectral inflation induced by quantization. In contrast, the action-sensitive subspace retains only $R_s \approx 0.14$. The fact that random directions gain energy while action-critical directions lose it demonstrates that the spectral distortion is not a uniform global effect, but selectively erodes the action-sensitive subspace.

To trace the origin of such spectral distortion, we measure layer-wise weight SNR against the pre-QAT FP16 weights:

$$\text{SNR}^{(l)}w = 10\log 10\left(\frac{|\mathbf{W}_{\text{pre}}{}^{(l)}|F^2}{|\hat{\mathbf{W}}^{(l)} - \mathbf{W}_{\text{pre}}{}^{(l)}|F^2}\right) = 10\log 10\left(\frac{\left|\mathbf{W}_{\text{pre}}{}^{(l)}\right| F^2}{\left|\Delta\mathbf{W}_{\text{total}}{}^{(l)}\right|_F^2}\right) \text{ (in dB).} \quad (6)$$

Under W2A16-LSQ, layer-wise weight SNR stays persistently low across layers, with mean 4.61 dB and minimum 2.92 dB. Visualization in Appendix. J. It means that the perturbation is of the same order as the weights themselves. These layer-wise perturbations accumulate and mix into an aggregated feature perturbation $\mathbf{E}_q$. When $\mathbf{E}_q$ spreads energy across many feature directions, the covariance expansion $\mathbf{H}_q^\top\mathbf{H}_q = \mathbf{H}^\top\mathbf{H} + \mathbf{H}^\top\mathbf{E}_q + \mathbf{E}_q^\top\mathbf{H} + \mathbf{E}_q^\top\mathbf{E}_q$ contains a positive semidefinite term $\mathbf{E}_q^\top\mathbf{E}_q$, which raises the tail singular values, slows spectral decay, and increases r_{eff}. This spectral dispersion is functionally harmful because the action-sensitive directions no longer dominate the energy, and the action head receives corrupted cognition features.

These observations give rise to a possible failure hypotheses about why directly applying LSQ to VLA LLM weight quantization fails. We suggest that this failure is driven by three possible causes:
Cause 1: Unconstrained full-rank space of weight updates. The QAT-induced weight deviation measured above can be decomposed as:

$$\Delta\mathbf{W}_{\text{total}}^{(l)} = \hat{\mathbf{W}}^{(l)} - \mathbf{W}_{\text{pre}}^{(l)} = \underbrace{\left(\mathbf{W}_{\text{qat}}^{(l)} - \mathbf{W}_{\text{pre}}^{(l)}\right)}_{\Delta\mathbf{W}_{\text{train}}^{(l)}\text{: QAT-induced training drift}} + \underbrace{\left(\hat{\mathbf{W}}^{(l)} - \mathbf{W}_{\text{qat}}^{(l)}\right)}_{\Delta\mathbf{W}_{\text{quant}}^{(l)}\text{: quantization error}}. \quad (7)$$

Since LSQ imposes no rank-related restrictions on weight updates, $\Delta\mathbf{W}_{\text{train}}^{(l)}$ is permitted to propagate across all directions of weight space. However, unlike pre-training, VLA QAT is trained on comparatively limited data that spans only a narrow set of real-world scenarios. This risks catastrophic forgetting of the LLM backbone's pretrained vision-language priors under unconstrained full-weight updates.
Cause 2: STE-induced quantization-resolution collapse and quantization error amplification. Eq. 4 shows that any weight with $u \notin [Q_n, Q_p]$ receives zero gradient, so clipped outliers cannot be pulled back toward the center. Moreover, in standard LSQ, the gradient of δ is

$$\frac{\partial\widehat{\mathbf{W}}}{\partial\delta} = (u_{\text{round}} - u)\mathbb{I}_{\text{in}} + Q_n\mathbb{I}_{\text{low}} + Q_p\mathbb{I}_{\text{high}}. \quad (8)$$

The inside term is bounded by 0.5, whereas the outside term is a constant Q_n or Q_p of magnitude $\mathcal{O}(2^{b-1})$. Once a value is clipped, the local sensitivity of δ is dominated by this constant boundary term, creating a strong outlier-dominated pressure to expand δ. This over-expansion coarsens the quantization grid and reduces the effective resolution. In 2-bit quantization, where only $2^b = 4$ levels exist, a few extreme outliers can force most weights to share few effective levels, making $\widehat{\mathbf{W}}$ uninformative about $\mathbf{W}$ and substantially amplifying $\Delta\mathbf{W}_{\text{quant}}^{(l)}$. Detailed derivation in Appendix A.
Cause 3: Uniform single-step action MSE. Standard imitation-learning QAT minimizes the unified generative objective $\mathcal{L}_{\text{gen}}$ in Eq.5, which treats all timesteps and samples equally. However, as clarified in Appendix B, the sensitivity to weight quantization of the LLM backbone varies across diffusion timesteps and samples. This uniform objective minimizes the average generative error, yet fails to prioritize timesteps or samples that are more sensitive to quantization error, yielding suboptimal optimization.

3.3 Prior-Anchored Low-Rank Quantization

To target *cause 1*, we treat the pre-trained full-precision weights as an prior anchor to limit QAT-induced training drift. Specifically, we freeze the pretrained weight $\mathbf{W}_{\text{base}} \in \mathbb{R}^{m\times n}$, and introduce

a trainable low-rank branch $\Delta\mathbf{W} = \alpha\mathbf{B}\mathbf{A}$, with a scaling factor α, and $\mathbf{B} \in \mathbb{R}^{m\times r}$, $\mathbf{A} \in \mathbb{R}^{r\times n}$, $r \ll \min(m, n)$. The quantizer operates on their sum:

$$\mathbf{W}_q = \mathcal{Q}\big(\mathbf{W}_{\text{base}} + \alpha\mathbf{B}\mathbf{A}; \boldsymbol{\delta}, \boldsymbol{\beta}\big) = \boldsymbol{\delta} \cdot \text{clip}\left(\left\lfloor \frac{\mathbf{W}_{\text{base}} + \alpha\mathbf{B}\mathbf{A} - \boldsymbol{\beta}}{\boldsymbol{\delta}} \right\rceil, Q_n, Q_p\right) + \boldsymbol{\beta} \quad (9)$$

where the quantization step size $\boldsymbol{\delta} \in \mathbb{R}^m$ and bias $\boldsymbol{\beta} \in \mathbb{R}^m$ are also learnable parameters. We adopt per-channel scaling in our implementation to balance representation resolution and computational efficiency. As for initialization, we adopt a min-max asymmetric initialization for the per-channel quantizer parameters $\boldsymbol{\delta}$ and $\boldsymbol{\beta}$. The low-rank matrices $\mathbf{A}$ and $\mathbf{B}$ are initialized using the standard LoRA scheme: $\mathbf{A}$ is sampled from Gaussian distribution and $\mathbf{B}$ is initialized to zero.

3.4 Soft-Mask Gradients with Severity-Aware Scale Regulation

To target *cause 2*, we propose *Soft-Mask Gradients with Severity-Aware Scale Regulation* (SMG-SASR), a lightweight gradient-reparameterization scheme that simultaneously exerts a corrective pull on outlier weights and prevents the step size from over-expanding for extreme values.

Soft-Mask Gradient for Weights Update. As discussed in *cause 2*, the hard indicator in Eq.4 assigns zero weight gradient to any $u \notin [Q_n, Q_p]$, which is especially harmful in ultra-low-bit quantization. We replace it with a temperature-controlled soft mask: $\mathcal{M}(u;\tau) = \sigma\big(-\tau\cdot(u-Q_p)\big)\cdot\sigma\big(\tau\cdot(u-Q_n)\big)$, where $\sigma(\cdot)$ is the sigmoid function, and $\tau > 0$ controls the sharpness of the transition. $\mathcal{M}$ is close to one inside the range and decays smoothly outside, so clipped weights receive non-zero, deviation-dependent pull-back gradients. Larger τ recovers the hard indicator, while smaller τ yields a smoother and more stable pull-back. To smoothly transition from soft to strict quantization, we linearly anneal τ from τ_{start} to τ_{end} over training: $\tau(t) = \tau_{\text{start}} + (\tau_{\text{end}} - \tau_{\text{start}}) \cdot \min\left(\frac{t}{T}, 1\right)$, where t is the current training step and T is the total annealing steps. This progressively sharpens the mask, giving stable early pull-back and near-hard quantization at convergence.

Severity-Aware Scale Regulation. Rescuing weight gradients alone is insufficient, so we also reformulate the step-size gradient. To remove this fixed-constant pressure that force δ to expand sharply, we define:

$$\frac{\partial\widehat{W}}{\partial\delta} = \Big(\big(u_{\text{round}} - u\big)\cdot\mathbb{I}_{\text{inside}} + u\cdot\mathbb{I}_{\text{outside}}\Big), \quad (10)$$

where $\mathbb{I}_{\text{inside}}$ and $\mathbb{I}_{\text{outside}}$ indicate whether u falls within or outside $[Q_n, Q_p]$. Unlike LSQ, which uses constants $\pm Q_n$ or Q_p for outliers, we use the normalized value u itself. This gives a scale gradient proportional to the outlier's actual deviation: mild outliers yield modest updates, while extreme outliers yield stronger ones. The optimizer therefore receives graded, severity-aware signals for scale adjustment, avoiding fixed-constant over-expansion and preserving effective quantization resolution, and thus reduces the layer-wise quantization error $\Delta\mathbf{W}^{(l)}_{\text{quant}}$.

3.5 Difficulty-Aware Curriculum Learning with SNR-Guided Static Reweighting

To target *cause 3*, we propose *Difficulty-Aware Curriculum Learning with SNR-Guided Static Reweighting* progressively treat quantization-sensitive samples as primary learning targets.

SNR-Guided Static Reweighting. To incorporate timestep-dependent difficulty priors, we reweight each per-sample diffusion MSE by the inverse posterior variance, up-weighting low-noise timesteps. For a variance-preserving schedule with cumulative product $\bar{\alpha}_t$, the per-sample squared error reads

$$\ell_i = \left\| \mathcal{H}_\phi(\mathbf{x}_t^{(i)}, t, \mathbf{c}^{(i)}) - \mathcal{T}^{(i)} \right\|_2^2, \quad (11)$$

with timestep-dependent weight defined as:

$$w_i^{\text{snr}} = \min\left(\frac{1}{1-\bar{\alpha}_t + \varepsilon}, \gamma\right), \quad \varepsilon = 10^{-4}, \gamma = 10.0. \quad (12)$$

The final weighted diffusion loss is $\mathcal{L}_{\text{snr}} = \frac{1}{B}\sum_{i=1}^{B} w_i^{\text{snr}}\ell_i$, where $\mathcal{T}^{(i)}$ denotes the prediction target, corresponding to injected noise for DiT and velocity field for Flow Matching.

Difficulty-Aware Curriculum Learning strategy. While the SNR-compensated loss adjusts difficulty at the timestep level, samples at the same timestep show varying quantization sensitivity. To strengthen learning for quantization-sensitive samples, we introduce an online difficulty-aware curriculum that progressively prioritizes these samples as primary learning targets in the mid-to-late training stage while preserving training stability.

To achieve this goal, we decompose it into two issues: **how to define sample difficulty** and **how to avoid destabilizing QAT by overemphasizing hard samples in early training**.

To address the first issue, we use the per-sample diffusion MSE $d_i = \ell_i$ as a label-free online difficulty score. To make the score comparable across batches, we normalize it by an EMA estimate of its batch mean, $\tilde{d}_i = d_i/(\mu_{\text{ema}} + \delta)$.

Second, to avoid destabilizing QAT by overemphasizing hard samples in early training, we make the difficulty emphasis progressive. This is necessary because the quantizer and LoRA adapters are still forming coarse representations at the beginning of quantized training, and an immediate focus on hard samples would let their gradients dominate and amplify quantization error before stable features emerge. We therefore schedule the curriculum in **three phases**:**warmup**, **ramp-up** and **full**. Specifically, let λ_i^{curr} denotes per-sample curriculum coefficient. 1) During **warmup**, difficulty emphasis is disabled by setting $\lambda_i^{\text{curr}} = 1$ for all samples, so the model first learns robust coarse representations under uniform signals. 2) During **ramp-up**, to turn the normalized difficulty into a bounded emphasis, we first define the per-sample curriculum coefficient λ_i^{curr} as $\lambda_i^{\text{curr}} = m_i \min\left(\exp\left(\frac{\tilde{d}_i}{\tau}\right), \lambda_{\max}\right)$, where $m_i \in \{0, 1\}$ is the Top-k mask. We set $m_i = 1$ if sample i ranks among the hardest $k \cdot B$ samples in the batch, and set $m_i = 0$ otherwise, where B is the batch size. Here k controls how many hard samples receive emphasis, τ controls how sharply the emphasis concentrates on the hardest samples, and $\lambda_{\max}$ caps the influence of any single selected sample. To gradually strengthen the emphasis on difficult samples, we linearly anneal k, τ and $\lambda_{\max}$. 3) During the **full phase**, k, τ, and $\lambda_{\max}$ are fixed their final values τ_{end}, $k_{\text{end}} = 1.0$, and $w_{\max}^{\text{end}}$, maintaining sustained emphasis on discriminative samples.

Overall, this progressive schedule keeps early quantized training stable and shifts the training emphasis to quantization-sensitive samples in the mid-to-late stage, thereby preserving the action-sensitive energy within the cognition-feature spectrum.

4 Experiments

4.1 Experimental Setup

We evaluate SP-QAT across two mainstream VLA policies with different action prediction architectures, CogACT and $\pi 0$. Since no authoritative Bridge-finetuned version of $\pi 0.5$ is available, we choose the more mature $\pi 0$ to ensure a more reliable evaluation, which shares a familiar architecture. We conduct simulation evaluations on SimplerEnv Li et al. (2024b), a standardized benchmark for robotic manipulation that includes embodiment shifts and diverse task perturbations, providing a more challenging and rigorous test scenario for quantized VLA models. Similar to DA-PTQ Xu et al. (2026a), we assess in-domain performance over four standard manipulation tasks using the WidowX robot, and further investigate zero-shot cross-embodiment generalization on the Google Robot via Visual Matching and the more challenging Variant Aggregation settings. We train all models on 2 NVIDIA H100 GPUs with a global batch size of 96, which takes about 4 hours for CogACT and an hour for $\pi 0$. More training details are in Appendix E.

4.2 Comparison with Baselines

We compare our SP-QAT with two PTQ methods, GPTQ, Ω-QVLA and two QAT methods: LSQ and SQIL. All methods are configured with per-channel weight quantization. Details about baselines can be found in Appendix F.

In-Domain Comparison. Tab.1 shows that our SP-QAT significantly outperforms the second-best PTQ and QAT methods under both W4A16 and W2A16 quantization settings on CogACT and $\pi 0$, achieving 7.1% to 7.7% improvements in average success rate. Under the W2A16 setting, where

Table 1: **In-Domain Comparison on Simpler's WidowX Robot Tasks**. Success rates are averaged across 5 seeds for CogACT and 10 seeds for $\pi0$, following common practice.

Model	Setting	Method	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg. ↑	Δ	Mem. (GB) ↓
CogACT	FP	–	71.7%	50.8%	15.0%	67.5%	51.3%	–	14.0
	4/16	GPTQ	68.3%	49.2%	10.8%	59.0%	46.8%	-4.5%	3.8
		LSQ	57.1%	41.7%	16.3%	75.0%	47.5%	-3.8%	3.8
		Ω-QVLA	**72.0%**	49.2%	12.0%	56.6%	47.5%	-3.8%	4.1
		SQIL	67.5%	**64.2%**	18.8%	38.8%	47.3%	-4.0%	3.8
		Ours	58.3%	55.0%	**22.5%**	**83.3%**	**55.0%**	+3.7%	3.8
	2/16	GPTQ	15.8%	1.7%	0%	2.5%	5.0%	-46.3%	2.2
		LSQ	0%	0%	0%	0%	0%	-51.3%	2.2
		Ω-QVLA	16.7%	4.2%	0%	8.3%	7.3%	-44.0%	2.5
		SQIL	**62.5%**	45.4%	13.3%	34.2%	38.9%	-12.4%	2.2
		Ours	**62.5%**	**51.7%**	**16.7%**	**78.3%**	**52.3%**	+1.0%	2.2
$\pi0$	FP	–	61.7%	58.8%	23.8%	81.3%	56.4%	–	6.7
	4/16	GPTQ	**62.5%**	50.8%	16.7%	72.5%	50.6%	-5.8%	2.4
		LSQ	54.2%	49.2%	10.8%	77.5%	47.9%	-8.5%	2.4
		Ω-QVLA	58.3%	52.1%	16.7%	75%	50.5%	-5.9%	5.8
		SQIL	49.2%	**62.5%**	18.8%	59.6%	47.5%	-8.9%	2.4
		Ours	56.3%	**62.5%**	**24.6%**	**87.5%**	**57.7%**	+1.3%	2.4
	2/16	GPTQ	0%	0%	0%	0%	0%	-56.4%	1.7
		LSQ	0%	0%	0%	0%	0%	-56.4%	1.7
		Ω-QVLA	0%	0%	0%	0%	0%	-56.4%	1.8
		SQIL	34.2%	22.1%	2.5%	57.9%	29.2%	-27.2%	1.7
		Ours	**53.3%**	**61.7%**	**16.7%**	**83.3%**	**53.8%**	-2.6%	1.7

most existing methods fail, our SP-QAT surpasses the FP16 baseline by 1% on CogACT and incurs only a 2.6% success rate drop on $\pi0$.

Table 2: **Cross-Domain Comparison on Simpler's Google Robot Tasks**.

Google Robot	Setting	Method	Pick Coke Can	Move Near	Open/Close Drawer	Open Top Drawer And Place Apple	Avg. ↑	Δ
Visual Matching	FP	–	91.3%	85.0%	71.8%	50.9%	74.8%	–
	4/16	GPTQ	**94.3%**	79.6%	74.1%	38.0%	**71.5%**	-3.3%
		Ω-QVLA	88.4%	63.3%	64.8%	39.8%	64.1%	-10.7%
		LSQ	89.0%	62.5%	68.5%	**55.6%**	68.9%	-5.9%
		SQIL	86.3%	55.8%	65.9%	41.7%	62.4%	-12.4%
		Ours	91.2%	**84.6%**	**75.9%**	44.4%	**74.0%**	-0.8%
	2/16	GPTQ	0%	4.2%	3.7%	0%	2.0%	-72.8%
		Ω-QVLA	0%	3.3%	1.8%	0%	1.3%	-73.5%
		SQIL	39.2%	9.2%	52.7%	2.1%	25.8%	-49.0%
		Ours	**86.3%**	**63.3%**	**77.8%**	**47.2%**	68.7%	-6.1%
Variant Aggregation	FP	–	89.6%	80.8%	28.3%	46.6%	61.3%	–
	4/16	GPTQ	69.3%	**75.8%**	32.3%	47.6%	56.3%	-5.0%
		Ω-QVLA	64.2%	74.8%	32.5%	30.7%	50.6%	-10.7%
		LSQ	**82.6%**	53.3%	24.1%	36.5%	49.1%	-12.2%
		SQIL	60.6%	45.2%	20.3%	16.7%	35.7%	-25.6%
		Ours	69.7%	70.8%	**65.9%**	**52.9%**	**64.8%**	+3.5%
	2/16	GPTQ	2.8%	9.8%	0.8%	0%	3.3%	-58.0%
		Ω-QVLA	3.3%	10.2%	0%	0%	3.4%	-57.9%
		SQIL	26.7%	9.2%	45.4%	0%	20.3%	-41.0%
		Ours	69.3%	47.5%	61.1%	46.0%	56.0%	-5.3%

Cross-Domain Comparison. Since $\pi0$'s fixed-dimension proprioceptive encoder is robot-specific unlike proprio-free designs, it cannot support cross-proprioception experiments. Thus we only evaluate CogAct for cross-domain comparison. Tab.2 shows that under the W4A16 quantization setting, our SP-QAT achieves average success rates within 0.8% of the FP16 baseline for both visual

Table 3: **Ablation Study of Individual Components**. "PA-LoQ" refer to *Prior-Anchored Low-Rank Quantization*. "DA-CL" denotes the difficulty-aware curriculum learning strategy. Experiments are performed under the W2A16 quantization setting with CogACT as the backbone.

Method	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg. ↑	Δ
LSQ (Baseline)	0%	0%	0%	0%	0%	-
+PA-LoQ	24.2%	16.7%	3.3%	25.8%	17.5%	+17.5%
+ SMG-SASR	**62.5%**	**56.7%**	15.0%	41.7%	44.0%	+44.0
+ DACL-SNR (w/o DA-CL)	61.7%	53.3%	11.7%	59.2%	46.5%	+46.5
+ DACL-SNR (**SP-QAT**)	**62.5%**	51.7%	**16.7%**	**83.3%**	**52.3%**	+52.3

matching and variant aggregation protocols, outperforming the second-best PTQ method by 4.1% to 8.8%. Under the W2A16 setting, where most existing methods largely collapse, our SP-QAT still yields average success rates of 68.7% (-6.1%) and 56.0% (-5.3%) for visual matching and variant aggregation, respectively. This demonstrates that SP-QAT preserves most of the cross-domain generalization capability even under ultra-low-bit W2A16 quantization, a critical property absent from existing PTQ and QAT VLA quantization approaches.

4.3 Component Study

In Tab.3, we evaluate the contribution of each component on CogACT under the W2A16 quantization setting. The results show that restricting weight updates to a low-rank subspace rescues the model from total failure and achieves an average success rate of 17.5%. Incorporating *ATA-SASR* yields over 38% gains on the *put spoon on towel* and *put carrot on plate* tasks, with a 26.5% rise in average success rate. This demonstrates that reducing quantization errors caused by unconstrained δ inflation further improves overall model performance. Moreover, applying SNR-Guided Static Reweighting alone degrades performance on the carrot task, yet substantially lifts the underperforming *eggplant* task by 17.5%. Building upon SNR-Guided Static Reweighting, our *Difficulty-Aware Curriculum Learning strategy* delivers consistent gains across most tasks and brings a 5.8% improvement in average success rate. **More ablation study are in Appendix I**.

4.4 PERFORMANCE ON REAL-WORLD TASKS

To validate our quantized model for real-world manipulation, we deploy SP-QAT on a Franka Panda arm across four robotic tasks (Tab. H). Our W2A16 CogACT model achieves a 60% average success rate (-5% vs. FP16), with $1.6\times$ speedup and 84.3% memory savings. Task and implementation details are in the Appendix. H. **Real-world rollout videos are included in the Supplementary Material.**

Table 4: **Real-world manipulation results on Franka Panda**: FP16 CogACT vs W2A16 CogACT.

Method	Pick Green Box	Pick Handle	Pick Apple	Pick Red Cup	Avg.↑	Latency (ms)	SpeedUp ↑
CogACT	8/10	5/10	6/10	7/10	65.0 %	241	1.0×
SP-QAT	8/10	4/10	6/10	6/10	60.0 %	151	1.6×

5 Conclusion

We present **SP-QAT**, an efficient QAT framework that enables fixed 2-bit LLM quantization for VLA for the first time. Prior-anchored low-rank quantization limits QAT-induced training drift to a low-rank subspace. Soft-mask gradients with severity-aware scale regulation curb outlier-driven step-size expansion and resolution degradation, reducing quantization error before it can flatten the spectrum. Difficulty-aware curriculum learning with SNR-guided static reweighting strengthens learning for hard samples and low-noise timesteps, preserving action-sensitive energy within the cognition feature spectrum. Experiments on SimplerEnv and the Franka Panda arm verify that SP-QAT maintains task performance and generalization comparable to the FP16 baseline, with a 3.9–$6.4\times$ smaller model.

AI USE STATEMENT

In this work, we used generative AI tools for none of the required-disclosure research tasks. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, interpreting results, and the rest of the required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for editing the manuscript to improve readability, summarizing and analyzing existing literature and sourcing or searching for information during the survey phase, drafting parts of a research paper, and suggesting a structure for the paper. We have reviewed all AI-assisted work. LLM-suggested wording changes were accepted only after comparing against the authors' original intent and verifying technical accuracy; LLM-produced literature summaries and retrieved references were checked against the primary sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Arash Akbari, Arman Akbari, Masih Eskandar, Qitao Tan, Yixiao Chen, Jingwu Luo, Bertha Pangaribuan, Liyun Zhang, Jennifer Dy, Geng Yuan, et al. Actquant: Sub-4-bit action-guided quantization for vision-language-action models. *arXiv preprint arXiv:2605.24011*, 2026.

Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2978–2985. IEEE, 2020.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, et al. Dita: Scaling diffusion transformer for generalist vision-language-action policy. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7686–7697. IEEE, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024b.

Yixin Lin and Austin Wang. Polymetis: A robot control framework for pytorch. `https://facebookresearch.github.io/fairo/polymetis/`, 2021. Facebook AI Research.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision–language–action models for embodied ai. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

Seongmin Park, Hyungmin Kim, Sangwoo Kim, Wonseok Jeon, Juyoung Yang, Byeongwook Jeon, Yoonseon Oh, and Jungwook Choi. Saliency-aware quantized imitation learning for efficient robotic control. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13140–13150. IEEE, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18549–18557, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

Hongyu Wang, Chuyan Xiong, Ruiping Wang, and Xilin Chen. Bitvla: 1-bit vision-language-action models for robotics manipulation. *arXiv preprint arXiv:2506.07530*, 2025.

Xinyu Wang, Mingze Li, Sicheng Lyu, Dongxiu Liu, Kaicheng Yang, Ziyu Zhao, Yufei Cui, Xiao-Wen Chang, and Peng Lu. Ω-qvla: Robust quantization for vision-language-action models via composite rotation and per-step scaling. *arXiv preprint arXiv:2605.28803*, 2026.

Siyuan Xu, Tianshi Wang, Fengling Li, Lei Zhu, and Heng Tao Shen. Da-ptq: Drift-aware post-training quantization for efficient vision-language-action models. *arXiv preprint arXiv:2604.11572*, 2026a.

Yuhao Xu, Yantai Yang, Zhenyang Fan, Yufan Liu, Yuming Li, Bing Li, and Zhipeng Zhang. Qvla: Not all channels are equal in vision-language-action model's quantization. *arXiv preprint arXiv:2602.03782*, 2026b.

Yantai Yang, Yuhao Wang, Zichen Wen, Luo Zhongwei, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. Efficientvla: Training-free acceleration and compression for vision-language-action models. *Advances in Neural Information Processing Systems*, 38:40891–40914, 2026.

Jingxuan Zhang, Yunta Hsieh, Zhongwei Wan, Haokun Lin, Xin Wang, Ziqi Wang, Yingtie Lei, and Mi Zhang. Quantvla: Scale-calibrated post-training quantization for vision-language-action models. *arXiv preprint arXiv:2602.20309*, 2026.

A Detailed Derivation for Cause 2

For any $u \notin [Q_n, Q_p]$, the STE gradient with respect to the weight is zero, i.e., $\partial\widehat{\mathbf{W}}/\partial\mathbf{W} = 0$, so clipped outliers cannot be pulled back by gradient descent. For the step size δ, standard LSQ uses

$$\frac{\partial\widehat{\mathbf{W}}}{\partial\delta} = \big(u_{\text{round}} - u\big)\mathbb{I}_{\text{in}} + Q_n\mathbb{I}_{\text{low}} + Q_p\mathbb{I}_{\text{high}}. \tag{13}$$

The inside term is the rounding residual, bounded by 0.5 in magnitude. The outside term is the constant boundary index Q_n or Q_p, whose magnitude scales as $\mathcal{O}(2^{b-1})$. Therefore, once any value is clipped, the local sensitivity of δ is dominated by the large constant boundary term rather than by the bounded rounding residual, creating a strong pressure to expand δ. Over-expanding δ directly coarsens the quantization grid: with step size δ, the resolution is $\epsilon = \delta$, and the number of grid points covering the bulk of the weight distribution decreases. Let the weight distribution have entropy $H(\mathbf{W})$. Under coarse quantization, the quantization noise entropy $H(\mathbf{W} \mid \widehat{\mathbf{W}})$ grows with ϵ, so the mutual information $I(\mathbf{W}; \widehat{\mathbf{W}})$ can be computed as:

$$I(\mathbf{W}; \widehat{\mathbf{W}}) = H(\mathbf{W}) - H(\mathbf{W} \mid \widehat{\mathbf{W}}) \tag{14}$$

decreases monotonically as ϵ grows. When $\epsilon \gg \sigma_w$, almost all near-center weights collapse onto a single quantization level, so $I(\mathbf{W}; \widehat{\mathbf{W}}) \to 0$. In 2-bit quantization, where $2^b = 4$, a few extreme outliers can force δ to expand until most weights share very few effective levels, leaving $\widehat{\mathbf{W}}$ uninformative about $\mathbf{W}$ and substantially amplifying $\Delta\mathbf{W}^{(l)}_{\text{quant}}$.

B Visualization of Sensitivity to Quantization Error over Diffusion Timesteps

Fig. 4 (a) visualizes the impact of quantization error across DDIM steps, while Fig. 4 visualizes sensitivity to weight quantization of the LLM backbone across different samples. Specifically, we randomly select 200 samples from the BridgeData v2 validation set. Using LSQ with W4A16 quantization on CogACT, we evaluate the MAE of output predictions between the quantized model and its FP16 counterpart at every DDIM sampling step. Fig. 4 (a) illustrates that compared with high-noise timesteps, low-noise denoising timesteps are more vulnerable to weight quantization of the LLM backbone. Fig. 5 demonstrates that the action output error induced by quantization varies across input samples by up to a factor of 5.7. This shows that the impact of LLM backbone weight quantization on final action outputs is sample-dependent.

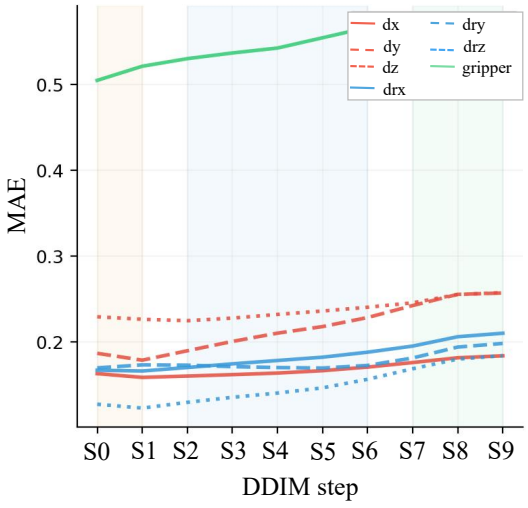


Figure 4: **Sensitivity to Quantization Error over Diffusion Timesteps.**

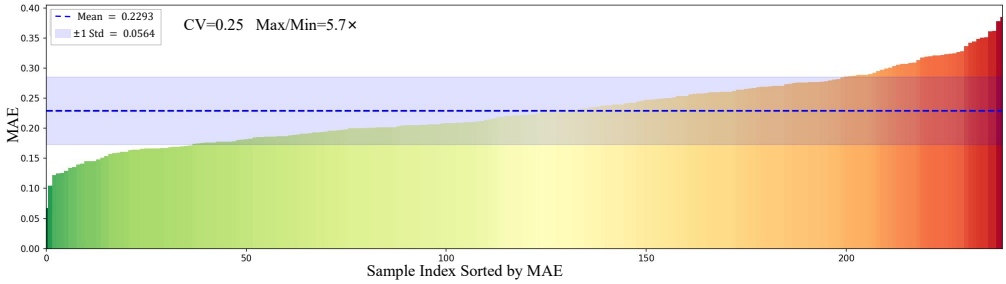


Figure 5: **Sensitivity to Quantization Error over Different Samples.**

C Complete Quantization Parameter Update Rules

The full backward gradient formulation for our *Soft-Mask Gradients with Severity-Aware Scale Regulation* is explicitly defined as:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mathbf{W}} = \dfrac{\partial \mathcal{L}}{\partial \widehat{\mathbf{W}}} \cdot \sigma\big(-\tau(t)\cdot(u-Q_p)\big)\cdot\sigma\big(\tau(t)\cdot(u-Q_n)\big), \quad \tau(t) = \tau_{\text{start}} + (\tau_{\text{end}} - \tau_{\text{start}}) \cdot \min\left(\dfrac{t}{T}, 1\right), \\ \dfrac{\partial \mathcal{L}}{\partial \delta} = \dfrac{\partial \mathcal{L}}{\partial \widehat{\mathbf{W}}} \cdot \Big((u_{\text{round}} - u)\cdot \mathbb{I}_{\text{inside}} + u \cdot \mathbb{I}_{\text{outside}}\Big), \quad \dfrac{\partial \mathcal{L}}{\partial \delta} = \dfrac{\partial \mathcal{L}}{\partial \delta} \cdot \text{sign}(\delta), \\ \dfrac{\partial \mathcal{L}}{\partial \beta} = \dfrac{\partial \mathcal{L}}{\partial \widehat{\mathbf{W}}} \cdot \mathbb{I}_{\text{outside}}. \end{cases} \tag{15}$$

D Algorithm

Algorithm 1 SP-QAT: Spectral-Preserving Quantization-Aware Training

Input: Pretrained weights $\mathbf{W}_{\text{pre}}$; dataset $\mathcal{D}$; bit-width b; LoRA rank r; total steps T; temperature schedule $(\tau_{\text{start}}, \tau_{\text{end}})$; curriculum schedule $\mathcal{C}$.
Output: Quantized VLA model with LoRA and learned scale parameters $\boldsymbol{\delta}, \boldsymbol{\beta}$.
Init: Freeze $\mathbf{W}_{\text{pre}}$; init LoRA $\mathbf{A} \sim \text{Kaiming}$, $\mathbf{B} \leftarrow \mathbf{0}$; init $\boldsymbol{\delta}, \boldsymbol{\beta}$ from $\mathbf{W}_{\text{pre}}$ statistics.
for $t = 0$ **to** $T-1$ **do**
  // Innovation 1: Prior-Anchored Low-Rank Quantization
  $\mathbf{W} \leftarrow \mathbf{W}_{\text{pre}} + \frac{\alpha}{r}\mathbf{BA}$ ▷ limit QAT drift to low-rank subspace
  $\mathbf{W}_q \leftarrow \text{Quantize}(\mathbf{W}, \boldsymbol{\delta}, \boldsymbol{\beta}, \tau)$ ▷ jointly optimize quantization parameters
  // Innovation 3: Difficulty-Aware Curriculum Learning with SNR-Guided Reweighting
  Sample batch; $\mathbf{z} \leftarrow \text{VLM}(\mathbf{o})$; $\ell_i \leftarrow \|\hat{\boldsymbol{\epsilon}}_i - \boldsymbol{\epsilon}_i\|^2$
  $w_i \leftarrow \text{SNRWeight}(t_d) \cdot \text{DifficultyWeight}(\ell_i, \mathcal{C}(t))$ ▷ emphasize hard samples and low-noise timesteps
  $\mathcal{L} \leftarrow \frac{1}{B}\sum_i w_i \ell_i$
  // Innovation 2: Soft-Mask Gradients with Severity-Aware Scale Regulation
  $\mathcal{M}(\mathbf{u}; \tau) \leftarrow \sigma(-\tau(\mathbf{u} - Q_p))\sigma(\tau(\mathbf{u} - Q_n))$ ▷ soft mask for weights
  $\frac{\partial \widehat{\mathbf{W}}}{\partial \boldsymbol{\delta}} \leftarrow (\mathbf{u}_{\text{round}} - \mathbf{u})\mathbb{I}_{\text{in}} + \mathbf{u}\mathbb{I}_{\text{out}}$ ▷ severity-aware scale gradient
  Update $\mathbf{A}, \mathbf{B}, \boldsymbol{\delta}, \boldsymbol{\beta}$ via $\nabla\mathcal{L}$ using $\mathcal{M}$ and severity-aware scale gradient
  Update action head
end for

E Training Details

We use BridgeData V2 dataset Walke et al. (2023) as our training dataset and follow its official train–validation split. We train our models on 2 NVIDIA H100 with a per-device batch size of 16 and a global batch size of 96, accumulated via gradient accumulation to maintain effective batch size. For both CogACT and $\pi 0$, we set curriculum with 200 warmup steps and 400 ramp-up steps, where temperature linearly anneals from 5.0 to 1.0, top-k from 0.1 to 1.0, and max weight from 2.0 to 5.0. The LoRA rank and α are set to 16 and 32 for both policies. We use AdamW, with bfloat16

mixed precision and gradient checkpointing enabled for both models. For CogACT, we set a learning rate of 2×10^{-5}, weight decay of 0.0, and gradient clipping at a maximum norm of 1.0. For $\pi 0$, the learning rate are set as 2×10^{-4}, weight decay of 0.0, and gradient clipping at a maximum norm of 1.0. We use a cosine scheduler with first-cycle steps set to 10^7 (effectively no decay), a minimum learning rate of 2×10^{-6}, and 50 warmup steps for both branches. Flow sampling adopts uniform distribution. The vision backbone is frozen, while the LLM backbone, action projection layers, and diffusion action module are fine-tuned. For the quantized model, we quantize all linear projections (q, k, v, o, up, down, gate) of the LLM.

F Baseline Details

All methods apply weight-only per-channel quantization exclusively to all linear projections (q, k, v, o, up, down, gate) of the LLM backbone in the VLA model.

GPTQ Frantar et al. (2022) GPTQ is a Hessian-based PTQ method that quantizes weights column by column while compensating for the induced error using second-order information from the layer-wise Hessian. It extends Optimal Brain Quantization with a fixed quantization order, batched weight updates, and Cholesky-based Hessian inverse computation to scale to large models. In our experiments, GPTQ uses the same BridgeData V2 dataset as ours, with 128 randomly selected samples as the calibration set.

Ω-QVLA Wang et al. (2026) Ω-QVLA is also a PTQ framework specialize for VLA models. It combines a composite SVD-Hadamard rotation for equalizing per-channel outliers with per-step dynamic scaling for the diffusion head. In our comparison, we follow the same setting as ours and apply Ω-QVLA only to the LLM backbone of the VLA model. It uses the BridgeData V2 dataset, with 128 randomly selected samples as the calibration set.

LSQ Esser et al. (2019) LSQ is a standard QAT method that treats the quantization step size as a learnable parameter optimized via backpropagation, enabling the network to adapt its quantization grid to the data distribution during training. In our experiments, LSQ is trained under the same setting as our method, including identical training data, optimizer configuration, and learning rate schedule.

SQIL Park et al. (2025) SQIL identifies task-critical states via saliency scores and applies weighted distillation of the full-precision policy's action distribution at those states during quantization-aware training. In our comparison, SQIL uses the BridgeData V2 dataset for training, consistent with our method. For CogACT, we use a global batch size of 96 and train for 1,500–2,000 steps with a constant learning rate of 2×10^{-4}. For π_0, the learning rate is set to 5×10^{-4} with a cosine scheduler, where the first-cycle steps are set to 10^7 (effectively no decay), a minimum learning rate of 2×10^{-6}, and 50 warmup steps for both branches.

G Implementation details of the difficulty-aware curriculum.

We maintain an EMA of the batch-mean difficulty,

$$\mu_{\text{ema}} \leftarrow \rho \mu_{\text{ema}} + (1-\rho)\bar{\ell}, \tag{16}$$

where $\rho = 0.99$ and $\bar{\ell}$ is the batch mean of ℓ_i. We use $\tilde{d}_i = d_i/(\mu_{\text{ema}} + \delta)$ with $\delta = 10^{-8}$. In each batch, let $\mathcal{I}_{\text{top}}$ be the indices of the top $k \cdot B$ samples ranked by d_i, and let

$$m_i = \begin{cases} 1, & i \in \mathcal{I}_{\text{top}}, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The curriculum coefficient is clamped as $\lambda_i^{\text{curr}} \leftarrow \min(\lambda_i^{\text{curr}}, \lambda_{\max})$. Training is partitioned into three phases. During warmup, difficulty emphasis is disabled, i.e., $\lambda_i^{\text{curr}} = 1$ for all samples. Following the L4Q protocol, warmup is split into two halves: in the first half, only LoRA matrices are trained with quantization disabled; in the second half, quantization is activated and quantizer parameters are updated. During ramp-up, τ linearly decreases from τ_{start} to τ_{end}, k linearly increases from k_{start} to 1.0, and $\lambda_{\max}$ linearly increases from a conservative cap to $\lambda_{\max}^{\text{end}}$. During the full phase, $(\tau, k, \lambda_{\max})$ stay at $(\tau_{\text{end}}, 1.0, \lambda_{\max}^{\text{end}})$.

H Details about Real-world experiment

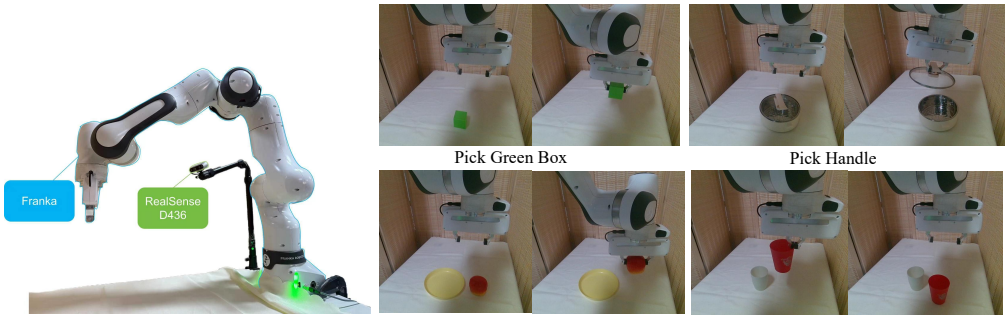
H.1 Real-world Setup

As shwon in Fig. 6 (a), we conduct all real-world experiments using a Franka Panda robotic arm equipped with a parallel gripper. We use a fixed Intel RealSense D436 RGB camera positioned beside the robot and facing the tabletop workspace as the sole visual observation for the policy, without using depth information or wrist-mounted cameras. The captured RGB images are cropped to the task-relevant workspace and resized to 224×224 before being fed into the policy.

To accommodate the computational demand of the policy, we adopt a client-server deployment architecture. The robot-side workstation is responsible for RGB acquisition, communication, and local robot execution, while model inference is performed on a remote server equipped with an NVIDIA A100 GPU. At each policy step, the latest RGB observation is transmitted to the inference server, where the policy predicts a 7-dimensional action

$$\mathbf{a}_t = [\Delta x, \Delta y, \Delta z, \Delta\phi, \Delta\theta, \Delta\psi, g], \tag{18}$$

consisting of relative end-effector translation, relative roll-pitch-yaw rotation, and a binary gripper command. The 6-DoF relative end-effector motion is defined in the Franka base frame. The predicted action is then transmitted back to the robot-side workstation, where the relative motion command is applied to the current end-effector pose to obtain the target Cartesian pose. The target pose is executed on the Franka Panda through the Polymetis control framework Lin & Wang (2021), while g determines the opening or closing of the gripper. This procedure forms a closed perception–inference–execution loop throughout each rollout.



(a) Physical Robot Setup (b) Real-world Tasks

Figure 6: **Physical robot setup and real-world task demonstrations.** The system consists of a Franka Panda arm and third-person view camera.

H.2 Tasks and Datasets

Tasks We evaluate our method on four real-world tabletop manipulation tasks involving objects with different appearances, geometries, and grasping requirements: picking up a green box, a handle, a red apple, and a red cup, as shwon in Fig. 6 (b). For each task, the target object is initialized within a predefined workspace with variations in its position and orientation. The robot is required to follow the corresponding language instruction and successfully grasp and lift the target object.

Datasets For each task, we collect 200 real-world demonstrations through teleoperation, resulting in a total of 800 trajectories.

H.3 Implementation Details

For the real-world experiments, we use the full-precision CogACT model as the baseline and compare it with its W2A16 SP-QAT counterpart.

For a fair comparison, the full-precision and quantized models use the same visual preprocessing pipeline, action representation, robot controller, and task-specific demonstration data. Both models predict the same 7-dimensional relative end-effector action and are executed through the same Polymetis-based control pipeline described in Sec. H.1.

All real-world inference is performed on the same NVIDIA A100 GPU. Our W2A16 quantized model leverages BitBLAS for low-bit computation acceleration. Latency is averaged over 200 input samples. The full-precision CogACT model requires an average inference latency of 241 ms, while SP-QAT reduces the latency to 151 ms, corresponding to a $1.6\times$ inference speedup.

H.4 Evaluation Protocol

For each task, we conduct 10 independent real-world trials. A trial is considered successful if the robot successfully grasps the target object with the gripper and lifts it from the tabletop, completing the pick-up task. We report the success rate over the 10 trials for each task. As shown in Fig. 7, the quantized SP-QAT model remains effective on all four real-world tasks, producing successful grasp-and-lift behaviors across diverse objects and task settings. **Real-robot rollout videos are included in the Supplementary Material.**



Figure 7: **Qualitative real-world results of the quantized SP-QAT model on four Franka Panda manipulation tasks.** For each task, we show representative frames from a successful rollout, illustrating the grasping and lifting process.

I Additional Ablation Studies

I.1 Ablation on Full-Rank vs. Low-Rank Weight Updates.

Tab. 5 compares full-rank updates with our *Prior-Anchored Low-Rank Quantization* under W2A16. In full-rank updates, the base weight, step, and bias are all trainable, while in our *Prior-Anchored Low-Rank Quantization* base weight is frozen, and only LoRA metrics, step, and bias are trainable. As shown in Tab. 5, compared with our low-rank updates, full-rank weight updates cause severe degradation on most tasks and yield an average success drop of 34.4%.

Table 5: **Ablation on Full-Rank vs. Low-Rank Weight Updates** under W2A16 Quantization for CogACT.

Method	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg.
Full-Rank	50.0%	11.7%	3.3%	6.7%	17.9%
Low-rank (Ours)	**62.5%**	**51.7%**	**16.7%**	**83.3%**	**52.3%**

I.2 Ablation on LoRA Capacity.

To evaluate the impact of low-rank capacity, we vary LoRA rank and α under W2A16 on the CogACT policy. Tab.6 sweeps the LoRA rank $r \in \{8, 16, 32\}$ with α fixed to $2r$. The success rate peaks at $r = 16$, while smaller ranks and larger ranks yield lower results. Tab.7 fixes $r = 16$ and varies $\alpha \in \{0.5, 1, 2\}$ to examine the effect of LoRA scaling. Although a LoRA scaling factor of 0.5 or 1 yields better performance on several individual tasks, the setting with $\alpha = 2$ achieves superior overall performance.

Table 6: **Ablation on LoRA Rank** under W2A16 Quantization for CogACT.

Rank	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg.
8	57.5%	50.0%	13.3%	51.2%	43.0%
16	**62.5%**	51.7%	**16.7%**	**78.3%**	**52.3%**
32	58.3%	**54.2%**	**16.7%**	60.4%	47.4%

Table 7: **Ablation on LoRA scaling** under W2A16 Quantization for CogACT.

Alpha	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg.
0.5	49.2%	**54.2%**	16.7%	75.8%	49.0%
1	55.4%	50.8%	**21.7%**	78.3%	51.6%
2	**62.5%**	51.7%	16.7%	**83.3%**	**52.3%**

I.3 Ablation on Temperature Annealing Schedule.

In Tab.8, We investigate the effects of different temperature annealing schedules on model performance. Specifically, we compare fixed, linear, and cosine temperature schedules under W2A16 quantization on CogACT. It shows that neither persistently high nor low temperature during training yields optimal performance for the quantized model. The linear temperature annealing achieves a 0.8% higher average success rate compared with cosine temperature annealing. This suggests that the linearly rising temperature schedule enables the model to learn better quantization than with the cosine schedule.

Table 8: **Ablation on Temperature Annealing Schedule** under W2A16 Quantization for CogACT. "$T = 2$" and "$T = 2$" denote constant temperature schedules with $T = 2$ and $T = 5$, respectively. "Linear" denotes linear annealing from $T = 2$ to $T = 5$. "Cosine" denotes cosine annealing from $T = 2$ to $T = 5$.

Method	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg.
$T = 2$	55.0%	50.8%	11.2%	63.3%	45.2%
$T = 5$	45.8%	**53.3%**	9.2%	65.0%	43.3%
Linear	**62.5%**	51.7%	16.7%	**83.3%**	**52.3%**
Cosine	55.8%	50.0%	**20.8%**	79.2%	51.5%

I.4 Ablation on Loss.

Tab. 9 compares plain MSE, SNR-only, difficulty-only, and SNR×difficulty. SNR-only balances timesteps, while difficulty-only focuses on hard samples. The combination achieves the best average, suggesting the two weightings are complementary.

I.5 Ablation on Difficulty Curriculum Schedule.

Tab. 10 compares three difficulty curriculum schedules. "Fixed" means directly applies the final difficulty setting (top-k$=0.1$, $T = 1$, cap$=5$) from the beginning. "Linear" denotes linearly anneals top-k from 1.0 to 0.1, T from 5 to 1, and cap from 2 to 5. While "Cosine" follows the same ranges with a cosine curve. The results indicate that progressive learning from easy to hard samples

Table 9: **Ablation on Loss Weighting Composition** under W2A16 Quantization for CogACT.

Loss	Spoon	Carrot	Stack	Eggplant	Avg.
MSE	**62.5%**	**65.0%**	15.0%	34.2%	44.2%
SNR only	61.7%	53.3%	11.7%	59.2%	46.5%
Difficulty only	64.2%	50.0%	13.3%	79.3%	51.7%
SNR×Diff (Ours)	**62.5%**	51.7%	**16.7%**	**83.3%**	**52.3%**

outperforms the fixed strategy that constantly focuses on hard samples. Unlike the cosine schedule with rapidly rising difficulty, a smooth transition from easy to hard samples balances early stability and late hard-sample mining and attains the best average success rate.

Table 10: **Ablation on Difficulty Curriculum Schedule** under W2A16 Quantization for CogACT. "Fixed": final difficulty setting from step 0. "Linear": linear top-k decay from 1.0 to 0.1, T decreases from 5 to 1, cap increases from 2 to 5."Cosine": same ranges with cosine annealing.

Schedule	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg.
Fixed	52.5%	50.8%	9.2%	73.3%	46.5%
Linear	**62.5%**	**51.7%**	**16.7%**	**78.3%**	**52.3%**
Cosine	56.7%	**51.7%**	12.5%	75.0%	49.0%

I.6 Ablation on Curriculum Scheduling.

Tab. 11 compares different curriculum scheduling setups and their effects on model quantization performance. We alter the stage fractions of warmup, rampup, and full phases while keeping all other hyperparameters fixed to isolate the impact of curriculum timing. The results show that carefully balanced stage fractions yield higher average success rates, and both insufficient and excessive ramp-up periods degrade overall task performance.

Table 11: **Ablation on Curriculum Scheduling** under W2A16 Quantization for CogACT.

Schedule	Warmup	Rampup	Full	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg.
No Warmup	0	0.5	0.5	55.0%	50.8%	9.2%	73.3%	47.1%
Short warmup	0.1	0.45	0.45	58.3%	47.5%	13.3%	50.8%	42.5%
Long warmup	0.3	0.35	0.35	52.5%	47.5%	12.5%	51.0%	40.9%
Short rampup	0.2	0.3	0.5	**63.3%**	**55.8%**	13.3%	54.2%	46.7%
Long rampup	0.2	0.5	0.3	54.2%	**55.8%**	15.0%	**83.3%**	52.1%
Ours	0.2	0.4	0.4	62.5%	51.7%	**16.7%**	78.3%	**52.3%**

J Per-layer Weight SNR

Fig. 8, Fig. 9, and Fig. 10 visualize the per-layer weight SNR of CogACT under the W4A16 and W2A16 quantization settings for LSQ, as well as the per-layer weight SNR of CogACT for our method under the W2A16 setting. We observe that the average per-layer weight SNR of the LLM backbone degrades substantially for W2A16-LSQ compared with W4A16-LSQ, dropping to merely around 0.5 dB. In contrast, our method significantly recovers the per-layer weight SNR across model layers.

K Visualization of Spectral and Action-Sensitive Subspace Preservation

To verify that our method mitigates the spectral distortion observed under W2A16-LSQ, we visualize the cognition-feature singular spectra and the action-sensitive subspace preservation under our

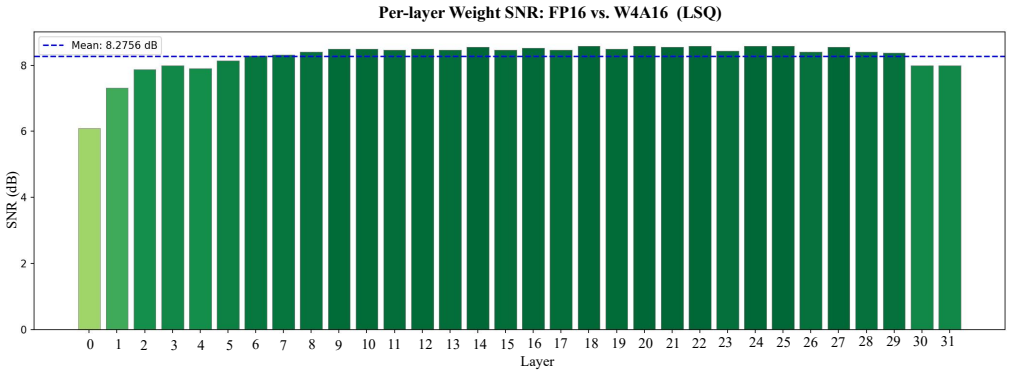


Figure 8: Per-layer weight SNR under W4A16-LSQ.

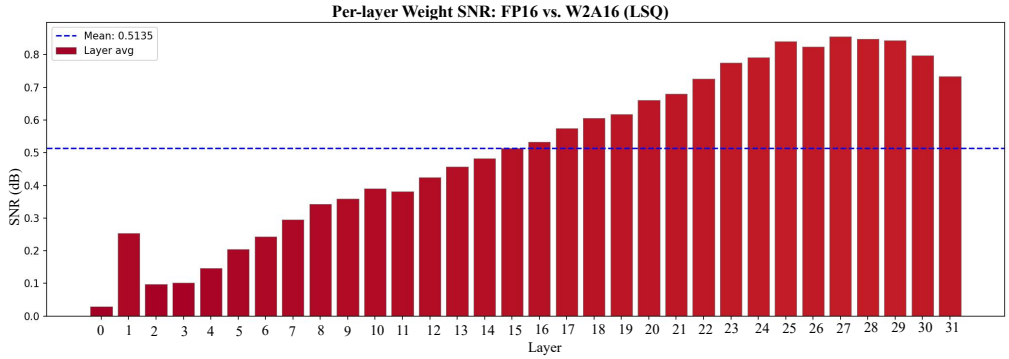


Figure 9: Per-layer weight SNR under W2A16-LSQ.

W2A16 quantized model. Fig. 11 shows that, unlike LSQ, our method maintains a sharply decaying spectrum close to FP16 and the tail singular values are no longer inflated. Consequently, the spectral energy remains concentrated in a few dominant directions instead of leaking into the tail.

We further fix the action-sensitive subspace $\mathbf{U}_s$ from the FP16 model and measure its energy preservation ratio R_s under our W2A16 model. As reported in Fig. 12, our method yields $R_s \approx 0.87$, compared with $R_{\text{rand}} \approx 0.2$ under W2A16-LSQ. This indicates that our method well preserves the action-critical directions. Together, these visualizations confirm that our method effectively mitigates ultra-low-bit-quantization-induced singular value spectrum flattening for cognition features and preserves the action-sensitive subspace within cognition features that condition the action head.

L Impact Statement

This paper introduces **SP-QAT**, an efficient QAT framework that enables fixed 2-bit LLM quantization for VLA models for the first time. By combining prior-anchored low-rank quantization, soft-mask gradients with severity-aware scale regulation, and difficulty-aware curriculum learning with SNR-guided static reweighting, SP-QAT limits QAT-induced training drift, curbs outlier-driven step-size expansion, which further leads to a reduction of effective resolution, and emphasize the learning of quantization-sensitive samples.

M Limitations

While SP-QAT makes substantial progress in ultra-low-bit quantization for VLA models, certain limitations merit discussion. Our SP-QAT still requires more training samples than PTQ calibration, introducing certain data collection overhead. Our training procedure relies on several hyperparameters that demand tuning, and future work will explore adaptive hyperparameter adjustment to reduce

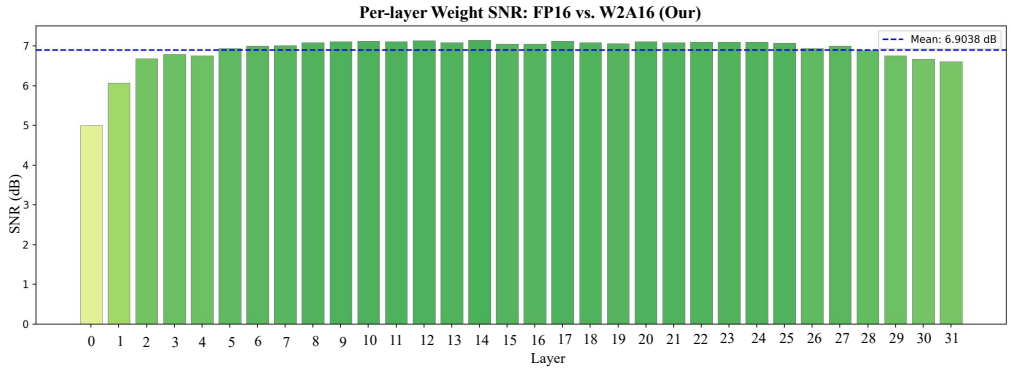


Figure 10: Per-layer weight SNR under W2A16-Our.

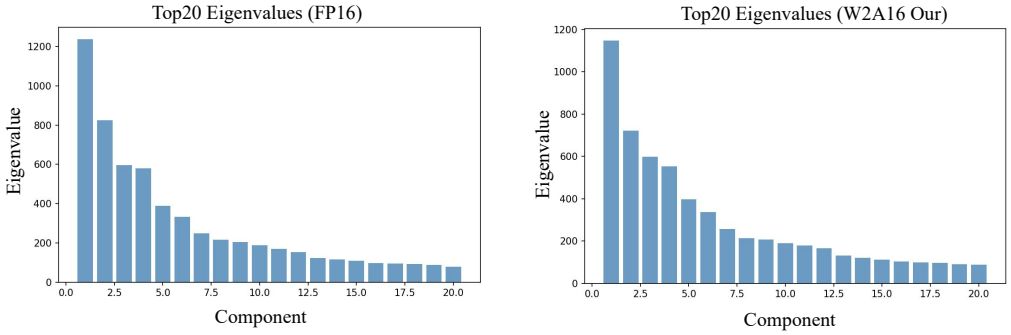


Figure 11: Singular value spectrum of cognition features: FP16 CogACT and W2A16 SP-QAT quantized CogACT models.

manual configuration efforts. Furthermore, though our method has certain embodiment generalization, data collection and fine-tuning are still needed for robotic arms with large discrepancies.

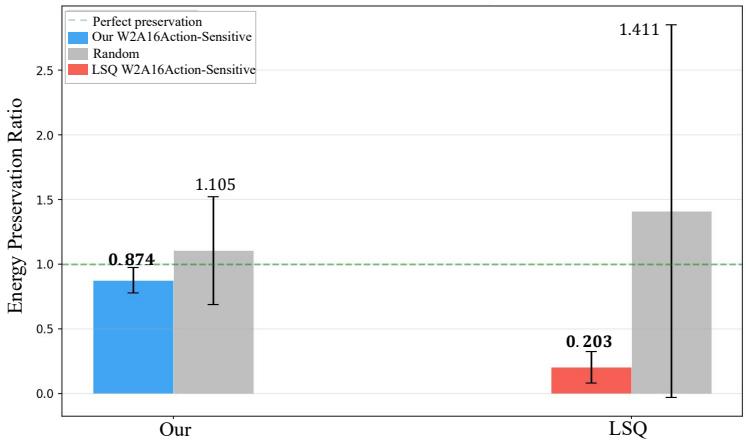


Figure 12: Jacobian Projection Energy: Action-Sensitive vs. Random Projections for W2A16 SP-QAT Quantized CogACT Model and W2A16 LSQ Quantized CogACT Model.