Zero-Shot Quantization for Vision-Language-Action Models via Trajectory Curvature and Attention Guidance

Sung-hwan Han [1] **Youngmin Yi** [1]

Abstract

Recently, Vision-Language-Action (VLA) models have advanced Embodied AI by integrating LLMs' reasoning into robotic control. While state-of-the-art VLAs combine Large Vision-Language Models (LVLMs) with Diffusion Transformers (DiTs), their substantial memory and computational overhead limit deployment on edge devices. Moreover, existing optimization techniques often require training data, which is frequently inaccessible due to privacy concerns. We introduce—to the best of our knowledge—the first Zero-Shot Quantization (ZSQ) framework for VLA models. By exploiting Flow Matching characteristics, we employ trajectory curvature and an attention-guided masking strategy to generate synthetic calibration data without any access to the original datasets. Our method reduces the memory footprint of the quantized components in $\pi_{0.5}$ and NVIDIA GR00T N1.5 by 70% and 55%, respectively, under the W4A8 setting, while retaining success rates comparable to data-dependent quantization methods.

1. Introduction

Vision-Language-Action (VLA) models have emerged as a cornerstone of Embodied AI, demonstrating enhanced generalization in robot control by leveraging the reasoning capabilities of Large Language Models (LLMs). By processing visual observations and language instructions to generate corresponding actions, models such as the RT series (Brohan et al., 2022; Zitkovich et al., 2023), Octo (Mees et al., 2024), and OpenVLA (Kim et al., 2025b) have shown a capacity for solving complex, open-ended tasks. Recently, a hybrid architecture integrating Large Vision-Language Models (LVLMs) (Liu et al., 2023b) with Diffusion Transformers (DiTs) (Peebles & Xie, 2023) has emerged as a leading paradigm, exemplified by the π series (Black et al., 2024; 2025), GR00T N1.5 (Bjorck et al., 2025), X-VLA (Zheng et al., 2026), and SmolVLA (Shukor et al., 2025).

Despite their high performance, VLAs face significant deployment barriers on robotic edge platforms due to severe constraints in computational power and memory. As VLAs integrate multiple modalities and scale in size, they introduce substantial operational overhead. Crucially, as noted in recent studies such as EfficientVLA (Yang et al., 2026) and QuantVLA (Zhang et al., 2026a), the bulk of this overhead originates not from visual perception (e.g., vision encoders) but from (i) the LVLM which conditions action generation and (ii) the iterative denoising in DiT-based action experts. This bottleneck remains a primary barrier to the deployment of VLAs on real-world robotic systems.

To address these efficiency concerns, previous research has focused on designing lightweight VLA architectures (Wen et al., 2025; Shukor et al., 2025) or reducing redundant computations (Zhang et al., 2026b; Yue et al., 2024; Xu et al., 2026a). While these approaches effectively reduce latency, they remain under-explored for the latest DiT-based architectures. Furthermore, many efficiency-oriented methods require training from scratch or fine-tuning, necessitating access to the original training data.

In the realm of LLMs, Post-Training Quantization (PTQ)—such as SmoothQuant (Xiao et al., 2023) and DuQuant (Lin et al., 2024)—has evolved to reduce memory usage without prohibitive training costs by suppressing activation outliers. Although recent attempts (Zhang et al., 2026a; Xu et al., 2026b) have applied quantization to VLAs, they still rely on access to training datasets. However, such data is frequently inaccessible due to corporate confidentiality, proprietary restrictions, and privacy concerns, making data-dependent quantization impractical for many real-world applications.

While Zero-Shot Quantization (ZSQ) has been extensively researched for CNNs (Cai et al., 2020; Zhong et al., 2022; Li et al., 2023) and ViTs (Li et al., 2022; Choi et al., 2025) to circumvent data dependency, its application to VLA models remains largely unexplored. Moreover, traditional ZSQ

[1]Dept. of AI, Sogang University, Seoul, Korea. Correspondence to: Youngmin Yi <ymyi@sogang.ac.kr>.

AdaptFM: Resource-Adaptive Foundation Model Inference Workshop at 43^{rd} International Conference on Machine Learning, Seoul, South Korea. Copyright 2026 by the author(s).

techniques that rely on class labels or Batch Normalization statistics are inapplicable to VLAs, necessitating a novel approach for high-quality synthetic data generation.

In this paper, we propose—to the best of our knowledge—the first Zero-Shot Quantization framework specifically designed for VLA models. Our approach leverages the observation that DiT-based VLAs commonly adopt Flow Matching (Liu et al., 2023c; Lipman et al., 2023), where the training objective involves a linear trajectory from noise to action. We utilize the curvature of the trained model's output trajectory—its deviation from ideal straight path—as a key metric for generating synthetic data. Furthermore, inspired by observations that VLA attention focuses on task-relevant regions (Xu et al., 2026a), we introduce an attention-guided loss function and a diverse masking strategy. This ensures that synthetic images cover various spatial regions with high attention scores, mimicking real-world task-relevant visual distributions.

Experimental results demonstrate that our proposed ZSQ methodology reduces the memory footprint of $\pi_{0.5}$ and NVIDIA GR00T N1.5 by 70% and 55% on the quantized components, respectively. Despite the absence of original training data, our method matches the performance of data-dependent quantization and approaches the full-precision baseline, under the W4A8 setting, enabling the practical deployment of state-of-the-art VLA models on resource-constrained robotic hardware.

2. Preliminaries

2.1. Problem Formulation and VLM Encoding

A Vision-Language-Action (VLA) model learns a policy $p(A_t|o_t)$ that maps a multimodal observation o_t at time t to an action chunk $A_t \in \mathbb{R}^{H\times D}$, where H denotes the action horizon and D represents the action dimensionality. The observation o_t is a multi-modal input comprising multiple images $I_t^{1,\dots,n}$, a language instruction ℓ_t, and the robot's proprioceptive state q_t:

$$o_t = [I_t^1, \dots, I_t^n, \ell_t, q_t] \tag{1}$$

Modern VLA architectures utilize an LVLM to encode o_t via a single forward pass. The resulting Key-Value (KV) cache is then consumed by an "Action Expert"—typically implemented as a DiT (Peebles & Xie, 2023)—through cross-attention mechanisms. This architecture ensures that the action generation process is grounded in the multimodal context provided by the LVLM.

2.2. Action Expert and Flow Matching

The Action Expert generates the action sequence A_t within the Flow Matching (Liu et al., 2023c; Lipman et al., 2023) framework. Flow Matching establishes an Ordinary Differential Equation (ODE) that connects a Gaussian noise distribution to the data distribution via a straight-line trajectory. For a time variable $\tau \in [0, 1]$, the linear interpolation between noise ϵ and the target action A_t is defined as:

$$A_t^\tau = \tau A_t + (1-\tau)\epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{2}$$

During training, the model is optimized to approximate the velocity vector field that points toward the data distribution. The target vector field $\mathbf{u}$, obtained by differentiating A_t^τ with respect to τ, is given by:

$$\mathbf{u}(A_t^\tau|A_t) = \frac{d}{d\tau}A_t^\tau = A_t - \epsilon \tag{3}$$

In Flow Matching, $\mathbf{u}$ remains constant relative to τ, implying that the model learns a straight trajectory. This linearity significantly enhances sampling efficiency compared to traditional diffusion models. A model $\mathbf{v}_\theta$ with parameters θ is trained by minimizing the Mean Squared Error (MSE) against the target vector field:

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau,\epsilon,(o_t,A_t)}\left[\left\|\mathbf{v}_\theta(\tau, A_t^\tau, o_t) - (A_t - \epsilon)\right\|^2\right] \tag{4}$$

At inference, the model solves the ODE starting from a noise state ($\tau = 0$) toward the data state ($\tau = 1$), conditioned on o_t. Specifically, an Euler solver is employed to iteratively transform the initial noise sample into the final action sequence A_t.

3. Methodology: Synthetic Calibration Data Generation

3.1. Overview

The goal of the proposed method is to generate synthetic calibration data for post-training quantization (PTQ) of pre-trained VLA policies without requiring access to original robot demonstrations or simulators. We specifically exploit the characteristic of Flow Matching (Liu et al., 2023c; Lipman et al., 2023) commonly adopted in DiT-based VLAs. Since Flow Matching learns a straight-line path from noise to action, an effective calibration input should induce a linear velocity field in the Action Expert. To this end, we directly optimize synthetic observations $\hat{o}$ using two guidance signals as illustrated in Figure 1:

- **Trajectory Curvature Guidance**: Minimizes the variance of the sampled velocity vectors to align with the straight-path assumption of Flow Matching.
- **Attention Coverage Guidance**: Encourages the cross-attention of the VLA to focus on diverse spatial regions, mimicking the attention patterns observed during task-relevant interactions.

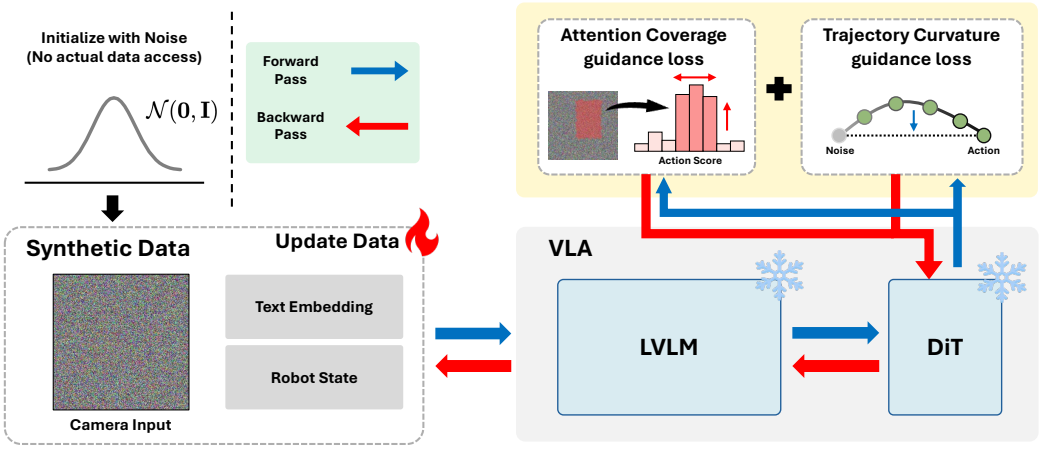

Figure 1. Overall pipeline of our proposed method.

3.2. Parameterization of Synthetic Observations

Following the Zero-Shot Quantization (ZSQ) setting (Cai et al., 2020), we assume no access to original training data, labels, or task-specific statistics. Instead, we synthesize observations $\hat{o}_i$ by optimizing multimodal inputs initialized from random noise. A synthetic sample i is defined as:

$$\hat{o}_i = [\hat{I}_i^1, \ldots, \hat{I}_i^n, \hat{c}_i, \hat{q}_i] \tag{5}$$

where $\hat{I}_i^m$, $\hat{c}_i$, and $\hat{q}_i$ denote the synthetic images, continuous text embedding, and robot state, respectively. To ensure pixel values remain within a valid range, each image is parameterized through a sigmoid mapping from an unconstrained latent variable $\tilde{I}_i^m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\hat{I}_i^m = \sigma(\tilde{I}_i^m), \quad m = 1, \ldots, n \tag{6}$$

The text embedding $\hat{c}_i$ replaces the discrete language instruction in the VLA's embedding space, and the robot state $\hat{q}_i$ is treated as an unconstrained vector; both are optimized directly without reparameterization. The optimization is therefore performed over the set of continuous variables $\Phi_i = \{\hat{I}_i^m, \hat{c}_i, \hat{q}_i\}$.

3.3. Trajectory Curvature Guidance

Flow Matching assumes a linear path $A_t^\tau = \tau A_t + (1-\tau)\epsilon$ between noise ϵ and action A_t. An ideal velocity field should be constant with respect to time τ. However, in a zero-shot setting, the absence of ground-truth A_t makes the use of endpoint-based targets infeasible. We therefore propose a self-consistent curvature objective that enforces consistency among velocities probed at different time steps for the same synthetic observation.

Velocity Variance Loss We measure the deviation from a straight path by calculating the variance of probe velocities along the model's own sampling trajectory. Specifically, starting from $A_i^{\tau_k} = \epsilon_i$ at $k = 0$, we apply K Euler steps with step size $\Delta\tau = 1/K$ to obtain $\{A_i^{\tau_k}\}_{k=1}^K$ at $\tau_k = k/K$, and define the probe velocity at each step as $v_{i,k} = v_\theta(\tau_k, A_i^{\tau_k}, \hat{o}_i)$. The mean probe velocity $\bar{v}_i$ is:

$$\bar{v}_i = \frac{1}{K} \sum_{k=1}^{K} v_{i,k} \tag{7}$$

The curvature guidance loss $\mathcal{L}_{\text{curv}}^{(i)}$ is then defined as:

$$\mathcal{L}_{\text{curv}}^{(i)} = \frac{1}{K} \sum_{k=1}^{K} \|v_{i,k} - \bar{v}_i\|_2^2 \tag{8}$$

This loss injects the inductive bias of a converged Flow Matching model—whose sampling trajectory should be nearly straight—into the synthetic data, where $\mathcal{L}_{\text{curv}}^{(i)} = 0$ implies a perfectly linear trajectory.

3.4. Attention Coverage Guidance

As observed in VLA-cache (Xu et al., 2026a), VLA models tend to allocate high attention weights to task-relevant visual regions during action generation. High-quality zero-shot calibration data should ideally induce similar activation patterns. However, in the absence of real task images, we designate diverse spatial regions within synthetic images as masks and encourage the cross-attention to concentrate on these regions.

Coverage Mask Generation For each synthetic sample i, we sample a pixel-space binary mask $\mathcal{M}_i^{\text{pix}} \in \{0,1\}^{H_{\text{img}} \times W_{\text{img}}}$. To prevent the attention guidance from developing an axis-aligned bias—which may occur if simple rectangles are used—we construct irregular polygons in the style of paint-by-example (Yang et al., 2023). However, since our goal is attention concentration rather than inpainting, perfectly distinct mask boundaries can be suboptimal. Therefore, we introduce random jitter at the edges

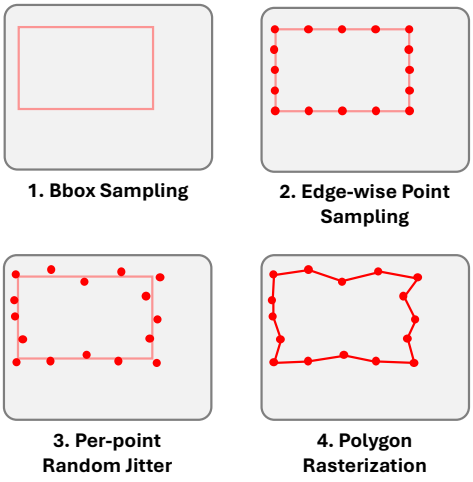

Figure 2. Concept of Attention Mask Generation

to provide structural perturbation. The generation process consists of four stages as illustrated in Figure 2:

1. **Bbox Sampling**: We define an axis-aligned bounding box $B_i = (x_0, y_0, x_1, y_1)$ by sampling the image area fraction $\alpha \sim \mathcal{U}(0.15, 0.25)$ and aspect ratio $r \sim \mathcal{U}(0.5, 2.0)$.

2. **Edge-wise Point Sampling**: Each of the four edges of B_i is divided into J equidistant points (we set $J = 20$), resulting in a set of boundary points $\mathcal{P}_i = \{(u_j, v_j)\}_{j=1}^{4J}$.

3. **Per-point Random Jitter**: Each point is perturbed such that $(u_j, v_j) \leftarrow (u_j + \delta_j \cos\theta_j, v_j + \delta_j \sin\theta_j)$, where $\theta_j \sim \mathcal{U}[0, 2\pi)$ and $\delta_j \sim \mathcal{U}(1, 5)$ pixels. Points falling outside the image boundaries are clipped.

4. **Polygon Rasterization**: The jittered points are connected to form a closed polygon, which is converted into a binary mask $\mathcal{M}_i^{\text{pix}}$ via scanline fill rasterization.

Inter-sample Dispersion via Center Memory Bank. To prevent repetitive sampling in the same image region, which could bias the attention guidance, we maintain a memory bank $\mathcal{B}_{\text{center}} = \{\hat{p}_j\}_{j<i}$ containing the centroids of previously generated masks. A candidate center p_i is accepted only if it satisfies:

$$\forall \hat{p} \in \mathcal{B}_{\text{center}} : \|p_i - \hat{p}\|_2 > \rho \cdot \min(H_{\text{img}}, W_{\text{img}}) \quad (9)$$

where $\rho = 0.15$ is the minimum distance ratio. We perform rejection sampling for up to $T_{\text{retry}} = 20$ attempts; if all fail, the candidate furthest from any point in $\mathcal{B}_{\text{center}}$ is selected as a fallback.

Pixel-to-Token Projection. To align the mask with the vision encoder's patch grid (P_h, P_w), we down-project $\mathcal{M}_i^{\text{pix}}$ into a token-space mask $\mathcal{M}_i^{\text{img}}$ using adaptive average pooling followed by binarization with a threshold $\tau_{\text{mask}} = 0.5$:

$$\mathcal{M}_i^{\text{img}} = \{j : \text{AvgPool}_{(P_h, P_w)}(\mathcal{M}_i^{\text{pix}})_j > \tau_{\text{mask}}\} \quad (10)$$

The final token mask $\mathcal{M}_i$ is obtained by adding the text-token offset T_{text} to the image token indices:

$$\mathcal{M}_i = \{T_{\text{text}} + j : j \in \mathcal{M}_i^{\text{img}}\} \quad (11)$$

The mask is fixed for each sample i throughout all optimization steps R. We experimentally found that resampling masks at each step causes the attention loss gradients to counteract one another across different regions, preventing stable convergence.

3.5. Attention Loss

The proposed attention loss targets the action-to-image attention. Specifically, in the cross-attention mechanism where action tokens serve as queries (Q) and image/text tokens serve as keys/values (KV), we generate synthetic data by inducing attention values to be concentrated on the masked region $\mathcal{M}_i$. This mimics the activation patterns observed during real task execution.

Mass Term: Attention Concentration. The mass term maximizes the total amount of attention each query directs toward the specified mask region. We define the post-softmax attention map for layer ℓ as:

$$\mathbf{A}^{(\ell)} \in \mathbb{R}^{H_{\text{attn}} \times N_Q \times N_{\text{KV}}} \quad (12)$$

where H_{attn} is the number of attention heads, N_Q is the number of query tokens, and N_{KV} is the total number of key/value tokens. For each head h and query q, the attention ratio $r_{h,q}^{(\ell)}$ within the mask is calculated as:

$$r_{h,q}^{(\ell)} = \sum_{j \in \mathcal{M}_i} \mathbf{A}_{h,q,j}^{(\ell)} \quad (13)$$

$$\mathcal{L}_{\text{mass}}^{(\ell,i)} = 1 - \frac{1}{H_{\text{attn}} N_Q} \sum_{h=1}^{H_{\text{attn}}} \sum_{q=1}^{N_Q} r_{h,q}^{(\ell)} \quad (14)$$

Spread Term: Entropy Maximization. The spread term prevents the attention from collapsing into a single token within the mask. By normalizing the attention distribution within the mask to obtain $\tilde{\mathbf{A}}_{h,q,j}^{(\ell)}$ and maximizing its entropy $\mathcal{H}_{h,q}^{(\ell)}$, we encourage uniform activation across the entire masked area:

$$\tilde{\mathbf{A}}_{h,q,j}^{(\ell)} = \mathbf{1}[j \in \mathcal{M}_i] \cdot \frac{\mathbf{A}_{h,q,j}^{(\ell)}}{\sum_{j' \in \mathcal{M}_i} \mathbf{A}_{h,q,j'}^{(\ell)}} \quad (15)$$

Algorithm 1 Synthetic Calibration Sample Generation

input Frozen VLA policy π_θ, Sample count N, Steps R, Probes K
for $i = 1$ **to** N **do**
  $\mathcal{M}_i \leftarrow$ sample irregular mask($\mathcal{B}_{\text{center}}$)
  $\epsilon_i \leftarrow$ sample action noise $\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Initialize $\Phi_i = \{\hat{I}_i^m, \hat{c}_i, \hat{q}_i\} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $r = 0$ **to** $R - 1$ **do**
    $\hat{o}_i \leftarrow$ build observation(Φ_i)
    $V_i \leftarrow \text{VelocityProbe}_\theta(\hat{o}_i, \epsilon_i; K)$
    $\mathcal{L}_{\text{curv}}, \mathcal{L}_{\text{attn}} \leftarrow$ compute losses($V_i, \hat{o}_i, \mathcal{M}_i$)
    if $r == 0$ **then**
      $w_{\text{curv},i}, w_{\text{attn},i} \leftarrow \mathcal{L}_{\text{curv}} + \varepsilon, \mathcal{L}_{\text{attn}} + \varepsilon$
    end if
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{curv}}/w_{\text{curv},i} + \mathcal{L}_{\text{attn}}/w_{\text{attn},i}$
    Update Φ_i via gradient descent
  end for
  $\mathcal{B}_{\text{center}} \leftarrow \mathcal{B}_{\text{center}} \cup \{\text{center}(\mathcal{M}_i)\}$
end for
output $\mathcal{D}_{\text{syn}} = \{\Phi_i\}_{i=1}^N$

$$\mathcal{H}_{h,q}^{(\ell)} = -\sum_{j \in \mathcal{M}_i} \tilde{\mathbf{A}}_{h,q,j}^{(\ell)} \log\left(\tilde{\mathbf{A}}_{h,q,j}^{(\ell)} + \varepsilon\right) \tag{16}$$

$$\mathcal{L}_{\text{spread}}^{(\ell,i)} = 1 - \frac{1}{H_{\text{attn}} N_Q} \sum_{h=1}^{H_{\text{attn}}} \sum_{q=1}^{N_Q} \frac{\mathcal{H}_{h,q}^{(\ell)}}{\log|\mathcal{M}_i|} \tag{17}$$

Total Attention Loss. The final attention loss is computed as the arithmetic mean across all L action-to-image attention layers, applying the guidance uniformly across depth:

$$\mathcal{L}_{\text{attn}}^{(i)} = \frac{1}{L} \sum_{\ell=1}^{L} \left(\mathcal{L}_{\text{mass}}^{(\ell,i)} + \beta \mathcal{L}_{\text{spread}}^{(\ell,i)}\right) \tag{18}$$

where β balances mass concentration against intra-mask uniformity. We set $\beta = 0.5$, and find the result to be insensitive to this choice. Beyond simply reflecting the observation that robots focus on task-relevant areas, this attention loss also acts as a regularizer in preventing the trajectory from becoming excessively straight.

As argued in HAST (Li et al., 2023), the presence of *hard samples* is crucial for effective ZSQ. As shown in Section 4.2, we observed that optimizing solely for curvature leads to the generation of "easy" samples with overly linearized trajectories. The attention loss prevents this collapse by ensuring the optimization does not over-focus on curvature but rather learns to attend to diverse spatial cues in the image input, thereby maintaining the necessary complexity for robust quantization.

3.6. Optimization Objective

To balance the curvature and attention losses, we apply step-0 normalization. At optimization step $r = 0$, we compute fixed scaling constants $w_{\text{curv},i} = \mathcal{L}_{\text{curv},0}^{(i)} + \varepsilon$ and $w_{\text{attn},i} = \mathcal{L}_{\text{attn},0}^{(i)} + \varepsilon$. The final objective is:

$$\mathcal{L}_{\text{total}}^{(i)} = \frac{\mathcal{L}_{\text{curv}}^{(i)}}{w_{\text{curv},i}} + \frac{\mathcal{L}_{\text{attn}}^{(i)}}{w_{\text{attn},i}} \tag{19}$$

Each synthetic sample is optimized independently. We fix the mask and velocity probe noise per sample and update only the parameters Φ_i. The detailed process is summarized in Algorithm 1.

4. Experiments

4.1. Experimental Setup

Models and Framework. To evaluate the effectiveness of our proposed methodology, we conduct experiments on state-of-the-art VLA models that employ DiT-based action heads: $\pi_{0.5}$ (Black et al., 2025) and NVIDIA GR00T N1.5 (Bjorck et al., 2025), following the benchmarks established in QuantVLA (Zhang et al., 2026a). All implementations and experiments are integrated within the *LeRobot* framework (Cadene et al., 2026). We use the pretrained checkpoints provided on the Hugging Face Hub for both models.

Evaluation Benchmark. Performance is assessed using the LIBERO simulation benchmark (Liu et al., 2023a), which consists of four distinct task suites: *Spatial, Object, Goal, and Long*. *Spatial* evaluates reasoning over spatial relations among otherwise identical objects. *Object* evaluates pick-and-place with a diverse set of unique objects, isolating object-level recognition. *Goal* fixes the objects and scene layout but varies the task goal, isolating the transfer of procedural knowledge. *Long* consists of long-horizon manipulation tasks that entangle spatial, object, and goal knowledge. Each task suite comprises 10 individual subtasks. Following the evaluation protocol of OpenVLA-OFT (Kim et al., 2025a), we measure the success rate by conducting 50 evaluation episodes per subtask, resulting in a total of 500 trials per task suite.

Implementation Details We target all linear layers within the LVLM and DiT for quantization. This encompasses the projection layers required for computing Query, Key, Value, and Output in attention mechanisms, as well as the linear layers present in MLPs. However, as noted in prior research (Zhang et al., 2026a), the quantization of attention layers in DiT yields marginal memory savings (approximately 0.1–0.2 GB) while incurring substantial performance degradation due to distribution shifts. Consequently, we exclude the projection layers within the DiT attention mechanisms from

Table 1. W4A8 quantization results on the LIBERO benchmark for $\pi_{0.5}$ and GR00T N1.5. Data Access "O" indicates methods that use original demonstration data for calibration; "X" indicates zero-shot quantization. Memory and Relative Saving are measured on the quantized components only.

Model	Precision	Method	Data Access	LIBERO				Avg.	Memory (GB)	Relative Saving (%)
				Spatial	Object	Goal	Long			
$\pi_{0.5}$	FP16	Baseline	-	95.40	99.20	95.20	94.20	96.00	4.27	–
	W4A8	SmoothQuant	O	79.80	93.80	15.60	74.60	65.95	1.07	75.00
	W4A8	DuQuant	O	92.40	100.00	88.00	94.60	93.75	1.17	72.58
	W4A8	QuantVLA	O	91.80	99.60	88.40	95.60	93.85	1.28	70.09
	W4A8	Ours (Curvature only)	X	93.40	99.80	91.00	97.60	95.45	1.28	70.09
	W4A8	Ours (Curvature + Attn)	X	93.60	99.40	92.40	96.60	95.50	1.28	70.09
GR00T N1.5	FP16	Baseline	-	96.80	99.00	96.40	83.60	93.95	2.00	–
	W4A8	SmoothQuant	O	9.20	24.20	6.40	1.60	10.35	1.16	42.10
	W4A8	DuQuant	O	94.60	98.20	93.40	71.60	89.45	0.73	63.58
	W4A8	QuantVLA	O	95.80	97.80	92.80	75.60	90.50	0.91	54.63
	W4A8	Ours (Curvature only)	X	91.00	94.80	81.40	67.20	83.60	0.91	54.63
	W4A8	Ours (Curvature + Attn)	X	91.00	96.20	90.00	68.60	86.45	0.91	54.63

our quantization targets. Comprehensive details regarding the quantization process and implementation specifics are provided in Appendix A and B, respectively.

4.2. Main Results on LIBERO Benchmark

Comparison Baselines To evaluate the quantization performance of our proposed method on $\pi_{0.5}$ and NVIDIA GR00T N1.5, we compare it against SmoothQuant (Xiao et al., 2023) and DuQuant (Lin et al., 2024), both of which have demonstrated superior performance in LLM quantization. Furthermore, we include QuantVLA (Zhang et al., 2026a), a recent state-of-the-art methodology specifically designed for DiT-based VLA models, as a primary baseline. We exclude QVLA (Xu et al., 2026b), as it is tailored to OpenVLA (Kim et al., 2025b) and does not transfer to modern DiT-based VLA architectures. We note that QuantVLA computes Attention Temperature Matching (ATM) and Output Head Balancing (OHB) statistics on test data. To avoid test-set leakage, we recompute these statistics on the LIBERO training set in our reproduction.

Comparison with Data-Dependent Baselines. The experimental results for W4A8 quantization on the LIBERO benchmark are summarized in Table 1. As noted in prior studies, directly applying LLM-centric quantization methods like SmoothQuant to VLA models leads to a catastrophic performance drop, resulting in average success rates of only 65.95% for $\pi_{0.5}$ and a critical failure of 10.35% for GR00T N1.5. While DuQuant and QuantVLA demonstrate significantly improved robustness, they strictly rely on access to real training data.

In contrast, our proposed zero-shot methodology achieves impressive performance without requiring access to original demonstration data. For $\pi_{0.5}$, our method achieves an average success rate of 95.50%, recovering nearly all of the FP16 performance (96.00%) and even outperforming data-dependent baselines such as DuQuant (93.75%) and QuantVLA (93.85%), while reducing memory consumption by approximately 70%. For NVIDIA GR00T N1.5, our approach reaches an average success rate of 86.45%. While a gap remains relative to the data-dependent baseline (QuantVLA at 90.50%), our method substantially narrows it without using any original demonstration data, while achieving a memory reduction of approximately 55%.

Ablation on Attention Coverage Guidance. The results highlight the role of the attention-guided loss in preventing trajectory collapse during synthesis. Curvature guidance alone can over-optimize the velocity field toward overly linear trajectories, biasing the calibration set toward "easy" samples and degrading performance on harder, longer-horizon tasks. Adding the Attention Coverage Guidance (Section 3.4) regularizes this optimization. The effect is most pronounced on GR00T N1.5: in Goal, the success rate improves from 81.40% to 90.00% ($+8.60$), and the average improves from 83.60% to 86.45%.

5. Conclusion

We introduce a Zero-Shot Quantization (ZSQ) framework for DiT-based Vision-Language-Action (VLA) models. We propose using the curvature of VLA sampling trajectories as an effective metric for optimizing synthetic calibration data, providing an alternative when original demonstration data is unavailable due to privacy, licensing, or proprietary restrictions. We further introduce an Attention Coverage Guidance that exploits the tendency of VLAs to attend to task-relevant image tokens during action generation. Our method achieves memory savings of up to 70% and 55% for $\pi_{0.5}$ and NVIDIA GR00T N1.5, respectively, recovering near full-precision success rates and approaching the accuracy of data-dependent quantization without using any original demonstration data.

References

Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M. R., Finn, C., Fusai, N., Galliker, M. Y., et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, K.-H., Levine, S., Lu, Y., Malla, U., Manjunath, D., Mordatch, I., Nachum, O., Parada, C., Peralta, J., Perez, E., Pertsch, K., Quiambao, J., Rao, K., Ryoo, M., Salazar, G., Sanketi, P., Sayed, K., Singh, J., Sontakke, S., Stone, A., Tan, C., Tran, H., Vanhoucke, V., Vega, S., Vuong, Q., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

Cadene, R., Alibert, S., Capuano, F., Aractingi, M., Zouitine, A., Kooijmans, P., Choghari, J., Russi, M., Pascal, C., Palma, S., Aubakirova, D., Shukor, M., Moss, J., Soare, A., Lhoest, Q., Gallouédec, Q., and Wolf, T. Lerobot: An open-source library for end-to-end robot learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CiZMMAFQR3.

Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13169–13178, 2020.

Choi, K., Lee, H., Kwon, D., Park, S., Kim, K., Park, N., Choi, J., and Lee, J. Mimiq: Low-bit data-free quantization of vision transformers with encouraging inter-head attention similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16037–16045, 2025.

Kim, M. J., Finn, C., and Liang, P. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025a.

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E. P., Sanketi, P. R., Vuong, Q., et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pp. 2679–2713. PMLR, 2025b.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Li, H., Wu, X., Lv, F., Liao, D., Li, T. H., Zhang, Y., Han, B., and Tan, M. Hard sample matters a lot in zero-shot quantization. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 24417–24426, 2023.

Li, Z., Ma, L., Chen, M., Xiao, J., and Gu, Q. Patch similarity aware data-free quantization for vision transformers. In *European conference on computer vision*, pp. 154–170. Springer, 2022.

Li, Z., Chen, G., Liu, S., Wang, S., VS, V., Ji, Y., Lan, S., Zhang, H., Zhao, Y., Radhakrishnan, S., et al. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint arXiv:2501.14818*, 2025.

Lin, H., Xu, H., Wu, Y., Cui, J., Zhang, Y., Mou, L., Song, L., Sun, Z., and Wei, Y. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37: 87766–87800, 2024.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023a.

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

Liu, X., Gong, C., and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023c. URL https://openreview.net/forum?id=XVjTT1nw5z.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Mees, O., Ghosh, D., Pertsch, K., Black, K., Walke, H. R., Dasari, S., Hejna, J., Kreiman, T., Xu, C., Luo, J., et al. Octo: An open-source generalist robot policy. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Wen, J., Zhu, Y., Li, J., Zhu, M., Tang, Z., Wu, K., Xu, Z., Liu, N., Cheng, R., Shen, C., et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

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, S., Wang, Y., Xia, C., Zhu, D., Huang, T., and Xu, C. VLA-cache: Efficient vision-language-action manipulation via adaptive token caching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026a. URL https://openreview.net/forum?id=QZYZ0Xm58q.

Xu, Y., Yang, Y., Fan, Z., Liu, Y., Li, Y., Li, B., and Zhang, Z. QVLA: Not all channels are equal in vision-language-action model's quantization. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=TpL2nXanru.

Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., and Wen, F. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18381–18391, 2023.

Yang, Y., Wang, Y., Wen, Z., Zhongwei, L., Zou, C., Zhang, Z., Wen, C., and Zhang, L. EfficientVLA: Training-free acceleration and compression for vision-language-action models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=SELYlDHZk2.

Yue, Y., Wang, Y., Kang, B., Han, Y., Wang, S., Song, S., Feng, J., and Huang, G. Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. *Advances in Neural Information Processing Systems*, 37:56619–56643, 2024.

Zhang, J., Hsieh, Y., Wan, Z., Lin, H., Wang, X., Wang, Z., Lei, Y., and Zhang, M. Quantvla: Scale-calibrated post-training quantization for vision-language-action models. *arXiv preprint arXiv:2602.20309*, 2026a.

Zhang, R., Dong, M., Zhang, Y., Heng, L., Chi, X., Dai, G., Du, L., Wang, D., Du, Y., and Zhang, S. Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18764–18772, 2026b.

Zheng, J., Li, J., Wang, Z., Liu, D., Kang, X., Feng, Y., Zheng, Y., Zou, J., Chen, Y., Zeng, J., Wang, T., Zhang, Y.-Q., Liu, J., and Zhan, X. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

Zhong, Y., Lin, M., Nan, G., Liu, J., Zhang, B., Tian, Y., and Ji, R. Intraq: Learning synthetic images with intra-class heterogeneity for zero-shot network quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12339–12348, 2022.

Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A. Quantization Detail

For simplicity and hardware-friendly implementation, we adopt symmetric uniform quantization defined by the following equations:

$$\tilde{X} = \text{clip}\left(\left\lfloor \frac{X}{S} \right\rceil, -2^{b-1}, 2^{b-1} - 1\right) \tag{20}$$

$$S = \frac{\max(|X|)}{2^{b-1} - 1} \tag{21}$$

where X denotes the full-precision weight/activation and $\tilde{X}$ denotes the quantized weight/activation in b-bit precision. $\lfloor \cdot \rceil$ denotes the rounding operator, and the clip function ensures the values remain within the representable range of b-bit integers.

Following prior work (Zhang et al., 2026a), we employ Post-Training Quantization (PTQ) to enhance deployment efficiency without heavy training cost. Specifically, we adopt DuQuant (Lin et al., 2024), a state-of-the-art quantization framework for Transformer models, to suppress massive activation outliers. DuQuant performs invertible reparameterization for each linear layer using block-orthogonal rotation matrices $\hat{R}_{(1)}, \hat{R}_{(2)}$ and a zigzag permutation matrix P to redistribute outliers. For the implementation, we build upon the open-source codebase of QuantVLA[1] (Zhang et al., 2026a).

Once the synthetic data is generated as described in Section 3, quantization proceeds following the DuQuant pipeline. For a fair comparison with QuantVLA, we adopt the W4A8 configuration (4-bit weights, 8-bit activations) and use 32 synthetic calibration samples.

B. Implementation Detail

Model Implementation Details. We evaluate two state-of-the-art DiT-based VLA models. For $\pi_{0.5}$, the LVLM is initialized from PaliGemma (Beyer et al., 2024), a Gemma-family vision-language model, and the action expert is implemented as a Gemma-based DiT (Team et al., 2024). For NVIDIA GR00T N1.5, the LVLM uses Eagle (Li et al., 2025), paired with a native DiT-based action transformer. For both models, we use LIBERO-finetuned checkpoints publicly available on the Hugging Face Hub: `lerobot/pi05-libero`[2] for $\pi_{0.5}$ and `ar0s/groot_libero`[3] for GR00T N1.5. All implementations are integrated into the LeRobot framework (Cadene et al., 2026) (version 0.5.0) to ensure reproducibility and consistent inference protocols across both models.

Optimization Setup. The optimization process for synthetic data generation is conducted using the Adam optimizer (Kingma & Ba, 2015) paired with a cosine annealing learning rate scheduler (Loshchilov & Hutter, 2017). The initial learning rates are set to 1×10^{-2} for $\pi_{0.5}$ and 5×10^{-3} for GR00T N1.5. Optimization is performed for 20 and 50 steps, respectively, due to their differing convergence speeds. The entire generation process for 32 samples takes less than one hour on a single NVIDIA A10 GPU, demonstrating the efficiency of our zero-shot approach.

Curvature Probe Configuration. We use $K = 10$ Euler steps when computing the curvature guidance loss $\mathcal{L}_{\text{curv}}$ for both models. Although this differs from the native inference setting of GR00T N1.5, which uses $K = 4$, we empirically found that the choice of K has negligible impact on both the curvature loss and the resulting quantization performance. This is consistent with the linearity assumption of Flow Matching: since a well-trained model produces near-straight trajectories, the velocity-variance-based curvature guidance loss $\mathcal{L}_{\text{curv}}$ is robust to the number of probe points.

[1] https://github.com/AIoT-MLSys-Lab/QuantVLA
[2] https://huggingface.co/lerobot/pi05-libero
[3] https://huggingface.co/ar0s/groot_libero