LOOPQUANT: LOOP-AWARE POST-TRAINING QUANTIZATION FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models achieve strong manipulation performance, yet their substantial memory footprint and latency hinder real-time execution on robot accelerators. In practice, a VLA operates across two nested loops: an inner loop over denoising steps and an outer control loop that initiates each re-planning request from a new observation. Under W4A8 post-training quantization (PTQ), static four-bit weights cause reconstruction errors to recur at every denoising step and compound across consecutive requests through executed actions. Meanwhile, dynamic activations shift across denoising steps, and we find that their channel peaks drift across observations, requiring offline calibration thresholds to cover both loops. We propose **LoopQuant**, a training-free W4A8 PTQ framework tailored to these nested loops. **Request-aware multi-observation activation extremal calibration** (RMAEC) records per-channel token percentiles at every denoising step, aggregates their maxima within each request and across diverse observations, and freezes the resulting scales. An extreme-value model predicts that these scales grow sub-linearly with the number of observations. To reduce weight error, **exact non-uniform codebook construction** aggregates repeated weight values to make dynamic programming tractable at layer scale, yielding globally optimal levels for fixed row scales. At runtime, **fused packed execution** decodes four-bit weights in a Triton kernel for half-precision matrix multiplication and executes the expert core with a single launch per denoising step. Averaged across four LIBERO suites, LoopQuant achieves 97.3% closed-loop success on $\pi0.5$ and 84.3% on GR00T-N1.5, outperforming the strongest quantized baseline by 2.5 and 4.0 percentage points, respectively. The packed $\pi0.5$ runtime serves requests $1.88\times$ faster than the unquantized BF16 model, requiring only $0.59\times$ of its resident GPU memory.

1 INTRODUCTION

Vision-language-action (VLA) models generate robot motor actions directly from camera observations and natural-language instructions (Zhong et al., 2025; Kawaharazuka et al., 2025). Typically, a modern VLA couples a vision-language encoder (Li et al., 2022; Zhai et al., 2023) and a pretrained language backbone (Touvron et al., 2023; Yang et al., 2025a) with a diffusion or flow-matching action expert (Peebles & Xie, 2023; Lipman et al., 2023). While models such as $\pi0.5$ (Black et al., 2024; Physical Intelligence et al., 2025) and GR00T-N1.5 (Bjorck et al., 2025) achieve strong manipulation performance, memory and latency severely constrain their deployment. Several gigabytes of weights must remain resident on robot accelerators, and each replanning request must finish before the subsequent action is issued. Short action horizons tighten this time budget further. These constraints motivate low-precision representations paired with efficient execution.

Prior work explores several routes to reduce these deployment costs. Training compact architectures from scratch (Wen et al., 2024; Shukor et al., 2025) and pruning computation (Yang et al., 2025c; Xu et al., 2025; Zhang et al., 2026b) reduce parameter counts and compute budgets, but typically recover task accuracy only after extensive fine-tuning on large demonstration datasets. Post-training quantization (PTQ) (Liu et al., 2021; Nagel et al., 2020) reduces weight storage with a small calibration set and no retraining, which is crucial when original training data are proprietary. We target the W4A8 regime evaluated by recent VLA quantizers (Zhang et al., 2026a): four-bit weights are packed into integer storage, and activations are rounded onto an eight-bit grid with static per-channel thresh-

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.

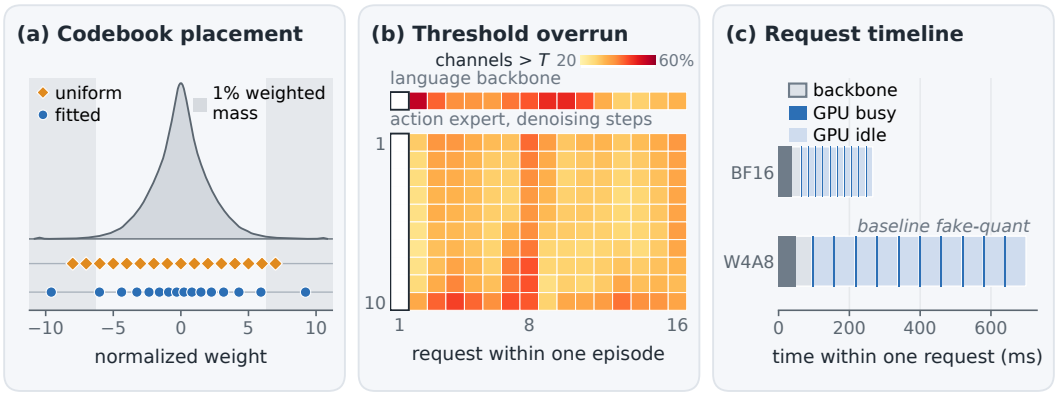

Figure 1: **(a)** Fitted and uniform four-bit levels on one $\pi 0.5$ projection; the density is activation-weighted, and its outer 1% is shaded. **(b)** Share of channels whose activation peak exceeds T during one BF16 closed-loop episode on LIBERO SPATIAL, where T is the maximum token percentile over all ten denoising steps of request 1, so its outlined column is zero. Rows show one language-backbone projection, then one action-expert projection per step. **(c)** Mean request timeline under eager execution on one RTX 5090 for BF16 and the released QuantVLA code's W4A8 fake-quantization. Blue segments are the ten denoising steps; darker shades mark GPU kernel time.

olds. Our runtime dequantizes rounded activations for half-precision matrix multiplication; packed storage reduces resident weight memory, while a single launch of the expert transformer core per denoising step curtails dispatch overhead.

VLA inference repeatedly invokes the same quantized projections across two nested loops. The *inner loop* evaluates the action expert across the J denoising steps of a request. The *outer loop* executes predicted actions, receives a new observation, and initiates the next request. The two loops affect quantization differently: since weights are quantized once offline, the same reconstruction error recurs at every denoising step and can compound across requests via executed actions (Ross & Bagnell, 2010). In contrast, activations fluctuate per forward pass while thresholds are frozen offline; calibration must cover intra-request denoising steps and inter-request observation shifts.

Figure 1 illustrates three challenges shaping our design. First, uniform four-bit grids space levels evenly across the weight range (Nagel et al., 2020; Frantar et al., 2023; Lin et al., 2024b), whereas levels fitted to empirical distributions provide finer resolution near dense centers (Figure 1a). Second, per-channel activation peaks shift across observations; peaks in later requests frequently exceed thresholds calibrated over all ten denoising steps of the initial request (Figure 1b). The first challenge calls for fitted weight codebooks, and the second for cross-observation calibration. Recent methods such as QuantVLA (Zhang et al., 2026a), ActQuant (Akbari et al., 2026), and QVLA (Xu et al., 2026) use selective or mixed precision, an orthogonal route to protect sensitive projections.

Third, non-uniform codebooks and iterative denoising complicate hardware execution. In baseline VLA quantizers, W4A8 inference is evaluated via eager fake-quantization, where per-operator quantize–dequantize operations introduce repeated kernel launches. Because an action expert iteratively evaluates short action sequences across J denoising steps per request, CPU dispatch overhead dominates, leaving the GPU idle for most of the request (Figure 1c).

We propose **LoopQuant**, a W4A8 PTQ framework tailored to the dual-loop structure of VLA inference (Figure 2). **Request-aware multi-observation activation extremal calibration (RMAEC)** records activation peaks across all denoising calls over diverse offline observations and freezes channel thresholds (Section 3.1). **Exact non-uniform codebook construction** finds globally optimal four-bit levels for fixed row scales via dynamic programming (Section 3.2). **Fused packed execution** pairs compact weight storage with a single expert-core launch per denoising step (Section 3.3). Quantizing all transformer attention and MLP projections in the language backbone and action expert, while retaining the vision encoders, embeddings, input and state projections, normalizations, and action heads in half precision, LoopQuant reaches average success rates of 97.3% on $\pi 0.5$ and 84.3% on GR00T-N1.5 across the four LIBERO suites (Liu et al., 2023). Our contributions are:

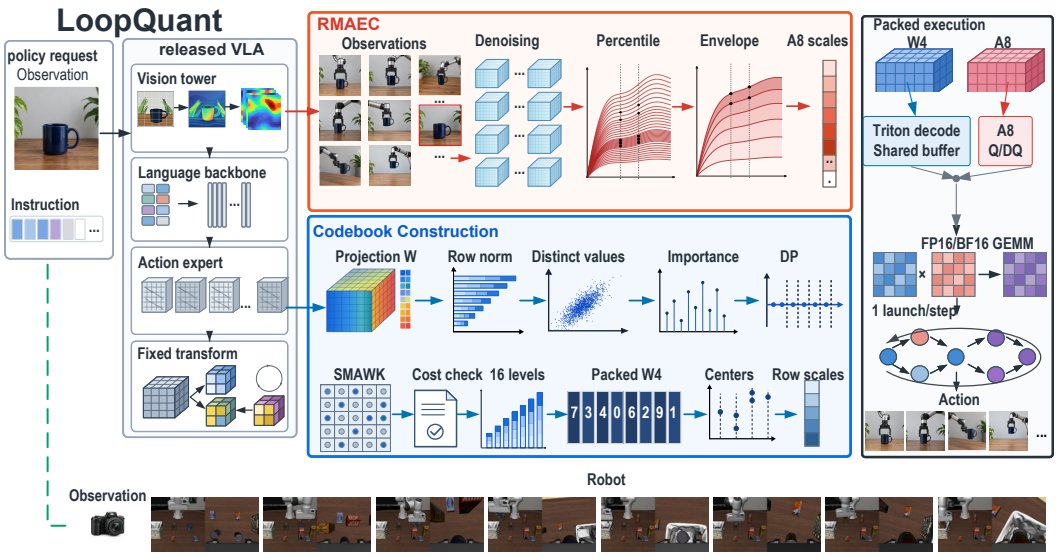

Figure 2: Overview. RMAEC sets channel thresholds from activation extremes over frames and steps (Section 3.1). Exact non-uniform codebooks fit four-bit levels per projection by dynamic programming (Section 3.2). Fused packed execution uses one expert-core launch per denoising step (Section 3.3).

- **Analysis of dual-loop quantization failure modes.** We trace how errors enter both loops: the same weight error recurs at every denoising step and can compound across requests, and we find that activation peaks shift across observations, so single-request calibration suffers clipping. An extreme-value model describes the sub-linear growth of thresholds with the frame budget.
- **The LoopQuant framework with three complementary modules. LoopQuant** is a training-free W4A8 PTQ framework tailored to dual-loop VLA execution. It unites: (i) **RMAEC** to bound activations across steps and scenes; (ii) exact non-uniform codebooks via Monge-accelerated dynamic programming; and (iii) fused packed execution with a single expert-core launch per step.
- **Closed-loop validation and hardware efficiency.** Evaluating on π0.5 and GR00T-N1.5 across four LIBERO suites against ten baselines, LoopQuant matches unquantized success. Our packed runtime reproduces the simulation build bit for bit while achieving substantial latency and resident memory gains on an RTX 5090.

2 Related Work

2.1 Post-Training Quantization for Transformer Stacks

Post-training quantization (PTQ) for VLA models stems largely from language-model compression. Weight-only methods target memory-bound decoding: OPTQ (Frantar et al., 2023) compensates for rounding errors via Hessian reconstruction, while AWQ (Lin et al., 2024b) protects salient channels through activation scaling. Joint weight-activation quantization is harder because outliers inflate dynamic ranges and coarsen quantization steps. One family rescales or groups channels: RPTQ (Yuan et al., 2023) clusters channels into tighter ranges, while SmoothQuant (Xiao et al., 2023) and OmniQuant (Shao et al., 2024) migrate outlier scales to weights. A complementary family rotates the representation basis to suppress activation peaks, as in QuaRot (Ashkboos et al., 2024), DuQuant (Lin et al., 2024a), FlatQuant (Sun et al., 2025), and OSTQuant (Hu et al., 2025). These principles extend to diffusion models (Wu et al., 2024; Li et al., 2024; Zhao et al., 2025; 2024) and vision-language backbones (Wang et al., 2024; Li et al., 2025). Across these lines of work, weights round onto a uniform *grid*, and ranges calibrate on external inputs.

Non-uniform codebooks replace uniform grids with discrete levels adapted to empirical weight distributions. SqueezeLLM (Kim et al., 2024) weights centroids by sensitivity proxies, while NF4 (Dettmers et al., 2023) fixes quantile levels under assumed distributions. VPTQ (Liu et al.,

2024) and any4 (Elhoushi & Johnson, 2025) learn codebooks via data-driven objectives, and related low-bit designs also depart from uniform grids (Dong et al., 2025; Yang et al., 2025b; 2024). These methods minimize reconstruction error via heuristic clustering or analytic priors, lacking exact layer-scale optimality guarantees.

2.2 Quantization of Vision-Language-Action Models

Quantizing VLA models, such as π0.5 (Physical Intelligence et al., 2025) and GR00T-N1.5 (Bjorck et al., 2025), is an active frontier. QuantVLA (Zhang et al., 2026a) sets W4A8 baselines across backbone and expert projections, folding calibration scalars into dequantization scales. QVLA (Xu et al., 2026) and ActQuant (Akbari et al., 2026) allocate mixed precision for sub-four-bit closed loop. To remove reliance on demonstrations, ZSQ (Han & Yi, 2026) synthesizes calibration samples from flow curvature and attention. In parallel, QAIL (Park et al., 2024) and SQIL (Park et al., 2025) fine-tune policies for four-bit weights, weighting loss toward salient states.

Because VLA action heads rely on iterative diffusion, our work connects to diffusion quantization (Shen et al., 2025): Q-Diffusion (Li et al., 2023) uses timestep calibration, PTQ4DiT (Wu et al., 2024) tracks temporal outliers, and Q-DiT (Chen et al., 2025) quantizes activations dynamically per step. While calibration data critically shapes quantization fidelity in language models (Ji et al., 2025; Williams & Aletras, 2024), current VLA quantizers rely on a small offline buffer (Zhang et al., 2026a) without accounting for inter-request activation drift across scenes.

3 Method

A VLA maps observations and instructions to actions through a vision-language backbone f_θ and an iterative action expert Ψ_θ (Black et al., 2024; Bjorck et al., 2025). The outer loop receives observations; the inner loop executes J denoising steps per request. DuQuant's blockwise rotations (Lin et al., 2024a) act as a fixed transform on weights and activations before calibration and quantization. In LoopQuant (Figure 2), RMAEC (Section 3.1) bounds activation shifts across denoising steps and scenes; exact codebook construction (Section 3.2) minimizes weight error weighted by activations from both loops; and fused packed execution (Section 3.3) eliminates dispatch overhead.

3.1 Request-Aware Multi-Observation Activation Extremal Calibration

Dual-loop activation shifts. In W4A8 quantization, weights are fixed offline, while diffusion VLA activations shift dynamically during execution along two coupled axes: (i) *intra-request shifts* across the J denoising steps as noise is iteratively removed, and (ii) *inter-request shifts* across observations, where channel activation peaks migrate from one request to the next.

The clipping trade-off and a first-forward baseline. Per-channel signed eight-bit quantization maps continuous activations x_c of channel c onto integers bounded by $q_{\max} = 127$. A clipping threshold $P_c > 0$ sets the channel range and uniform step size $s_c = P_c/q_{\max}$:

$$s_c = \frac{P_c}{q_{\max}}, \qquad \hat{x}_c = s_c \cdot \mathrm{clamp}\left(\left\lfloor \frac{x_c}{s_c} \right\rceil,\ -q_{\max} - 1,\ q_{\max} \right), \tag{1}$$

where $\lfloor \cdot \rceil$ rounds to the nearest integer. The threshold P_c trades off two errors. Setting P_c too low clips outlier activations and irreversibly discards high-amplitude action features, whereas setting it too high inflates s_c and lowers resolution for typical tokens. We keep P_c static, avoiding the per-sample range estimation overhead of dynamic schemes (Chen et al., 2025). A simple static baseline, which we call *first-forward calibration*, freezes each threshold at the projection's first input. For the action expert, however, that input covers only one denoising step of one request; peaks at subsequent steps or on new observations frequently exceed the frozen threshold, causing severe clipping.

Two-stage extremal envelope construction. To cover both sources of activation variation, we propose **request-aware multi-observation activation extremal calibration (RMAEC)**, which extends min-max range setting (Nagel et al., 2021) to both loops with a hierarchical envelope over tokens, denoising steps, and requests, and with a frame budget sized by a tolerance limit. RMAEC builds these static thresholds offline in two stages. First, for the request on frame $m \in \{1, \ldots, M\}$,

we filter isolated outliers with the $p = 99.9$th percentile $\mathcal{Q}_p$ over the L tokens of each denoising step. We then take the maximum over all J steps to cover the full inner loop:

$$P_{m,c} = \max_{j=1,\dots,J} \mathcal{Q}_p\big(\{|x_{m,j,\tau,c}|\}_{\tau=1}^{L}\big). \tag{2}$$

Second, to cover the outer loop, we aggregate these peaks over M frames from diverse scenes:

$$s_{\text{held},c} = \frac{1}{q_{\max}} \max_{m=1,\dots,M} P_{m,c}, \qquad s_{\text{first},c} = \frac{P_c^{\text{first}}}{q_{\max}}, \tag{3}$$

where $s_{\text{held},c}$ is the deployed RMAEC scale and $s_{\text{first},c}$ is the first-forward baseline scale from percentile P_c^{first}. Treating each request as a block, Stage 2 takes a maximum of block maxima (Coles, 2001) to track worst-case extrema across complete requests. The peak percentile of every calibration request thus enters the threshold undiluted. RMAEC determines which calls each threshold covers: all J denoising steps of each request, and M requests on diverse observations. This calibration pass concurrently accumulates the channel second moment $h_c = \mathbb{E}[x_c^2]$ used in Section 3.2. Table 3 compares RMAEC with the first-forward baseline, a single-frame envelope, and a percentile pooled over the same calls.

Coverage and resolution of the frame budget. The frame budget M governs two coupled quantities: coverage of unseen deployment requests and quantization resolution. First, for *coverage*, if a channel's per-request peaks are independent, identically distributed draws from the deployment distribution, their maximum forms a distribution-free upper tolerance limit (Wilks, 1941). With confidence $1-\delta$, the peak of a new request exceeds this maximum with probability at most β whenever $M \geq \lceil \log\delta / \log(1-\beta) \rceil$, giving $M = 32$ at $\beta = 0.05$ and $\delta = 0.20$. This non-parametric calculation guides our task-stratified frame budget (evaluated in Section 4.3). Second, for *resolution*, an expanded maximum coarsens step size s_c. We describe this cost with an extreme-value model that treats per-frame peaks as independent log-normal variables $\log P_{m,c} \sim \mathcal{N}(\mu_c, \sigma^2_{\log P,c})$. Letting $a_M = \mathbb{E}[\max_{m \leq M} Z_m]$ denote the expected maximum of M standard normal variates ($a_1 = 0$), the geometric-mean threshold ratio between budgets M_1 and M_2 depends solely on the channel dispersion $\sigma_{\log P,c}$:

$$\kappa_c(M_1 \to M_2) = \exp\big(\sigma_{\log P,c}\,(a_{M_2} - a_{M_1})\big). \tag{4}$$

Setting $M_1 = 1$ and $M_2 = M$ yields $\kappa_c(1 \to M)$, quantifying how far single-frame calibration underestimates the multi-frame envelope. Because normal maxima scale as $a_M \sim \sqrt{2\ln M}$, this resolution cost grows sub-linearly in M, while the probability $(1-\beta)^M$ of missing the coverage target falls geometrically. For positively correlated frames whose log-peaks are jointly normal with equal variance, Slepian's inequality establishes $\kappa_c(1 \to M)$ as an upper bound on geometric-mean threshold growth. Appendix B fits Eq. 4 to measured growth up to $M = 128$.

3.2 Exact Non-Uniform Codebook Construction

Activation-weighted reconstruction objective. With activation ranges established, we turn to four-bit weights. A uniform grid spends scarce levels on the sparse tails of bell-shaped weight distributions (Figure 1a). Weight errors also contribute unequally across channels: in matrix multiplication, the error along channel c is scaled by that channel's activation energy. We therefore weight each coordinate's error by the channel second moment $h_c = \mathbb{E}[x_c^2]$, which RMAEC collects over both loops, optimizing the codebook without gradients, labels, or a second calibration pass:

$$\mathcal{L}(\widehat{\mathbf{W}}) = \sum_{r=1}^{R}\sum_{c=1}^{C} h_c \left(W_{rc} - \widehat{W}_{rc}\right)^2. \tag{5}$$

Row-normalized shared codebook. We share one compact $k = 16$ codebook $\boldsymbol{\gamma} = (\gamma_1 < \cdots < \gamma_k)$ across the whole layer and absorb amplitude differences between output rows with an independent scale $\alpha_r > 0$ for each row r. A grid search around $\max_c |W_{rc}|/w_{\max}$ ($w_{\max} = 7$) picks the scale minimizing the row's unweighted mean squared reconstruction error on a uniform four-bit grid (Appendix A details candidate grid sizes and states the full solver as Algorithm 1). We normalize each row as $\widetilde{W}_{rc} = W_{rc}/\alpha_r$. The shared centers decode each weight as

$$\widehat{W}_{rc} = \alpha_r\, \gamma_{q(r,c)}, \tag{6}$$

with four-bit codes $q(r,c) \in \{1,\dots,k\}$. Substituting Eq. 6 into Eq. 5 gives $\mathcal{L}(\boldsymbol{\gamma}, q) = \sum_{r,c} \omega_{rc}(\widetilde{W}_{rc} - \gamma_{q(r,c)})^2$, where $\omega_{rc} = h_c \alpha_r^2$.

Fitting on the finite weight support. A 16-bit weight format admits at most $2^{16} = 65{,}536$ distinct bit patterns, enabling substantial problem reduction by aggregating repeated normalized weight values. We sort the distinct values of $\widetilde{\mathbf{W}}$ into $\mathcal{V} = \{v_1 < \cdots < v_N\}$ and assign each value the aggregate coordinate weight $\omega_n = \sum_{(r,c):\, \widetilde{W}_{rc} = v_n} \omega_{rc}$, leaving the reconstruction objective unchanged. Because weights cluster around zero, N stays in the low thousands per projection, two orders of magnitude below the weight count, rendering exact dynamic programming (Wang & Song, 2011; Grønlund et al., 2017) feasible at layer scale. Appendix A.1 bounds N and plots its layer-wise distribution on $\pi 0.5$ (Figure 4).

Monge-accelerated dynamic programming. On the sorted set $\mathcal{V}$, the optimal 1D scalar quantizer partitions values into $k = 16$ contiguous intervals centered at weighted centroids $\gamma(i,j) = S_1/S_0$, where $S_0 = \sum_{n=i}^{j} \omega_n$ and $S_1 = \sum_{n=i}^{j} \omega_n v_n$. Substituting $\gamma(i,j)$ yields the interval cost in closed form:

$$\mathrm{cost}(i,j) = S_2 - \frac{S_1^2}{S_0}, \qquad \gamma(i,j) = \frac{S_1}{S_0} \quad (S_0 > 0), \tag{7}$$

where $S_2 = \sum_{n=i}^{j} \omega_n v_n^2$. With prefix sums, $\mathrm{cost}(i,j)$ evaluates in $O(1)$ time. The minimum cost $D_\ell(j)$ of splitting $v_1,\dots,v_j$ into ℓ intervals satisfies the recurrence

$$D_\ell(j) = \min_{\ell \le i \le j} \big[\, D_{\ell-1}(i-1) + \mathrm{cost}(i,j) \,\big]. \tag{8}$$

Because the cost satisfies the quadrangle inequality (Monge property) (Wu, 1991; Grønlund et al., 2017), SMAWK (Aggarwal et al., 1987) finds all optimal split points for level ℓ in $O(N)$ time, lowering complexity from $O(N^2 k)$ to $O(Nk)$ (Grønlund et al., 2017). A single deterministic pass over $\mathcal{V}$ thus recovers the globally optimal codebook for the chosen row scales without initialization or restarts. Fitting takes seconds per layer. As a cost check, the optimum is provably no worse than a uniform grid on this objective, since uniform cells form a feasible contiguous partition.

3.3 Efficient Inference with Fused Packed Execution

Real-time deployment bottlenecks. With codebooks and activation thresholds determined offline, we deploy the quantized model on a single GPU. Packed weight storage reduces resident memory, while fused decoding and single-launch core dispatch eliminate CPU scheduling overhead at each denoising step.

Fused decoding into a shared buffer. Weights are stored as packed four-bit integers (two indices per byte), alongside the 16 centers $\boldsymbol{\gamma} \in \mathbb{R}^{16}$ and row scales $\boldsymbol{\alpha} \in \mathbb{R}^{R}$. A custom Triton kernel fuses index unpacking, codebook lookup, and row scaling ($\widehat{W}_{rc} = \alpha_r \gamma_{q(r,c)}$ in equation 6) into one pass, writing decoded weights just in time into a device buffer shared across all layers. Activations are rounded onto the eight-bit grid of equation 1 with static scales s_c, followed by a half-precision GEMM with the decoded weights.

One CPU dispatch per denoising step. To cut repeated CPU dispatch overhead in the inner loop, we exploit an invariant property of diffusion action generation: across all J denoising steps of a request, the action expert executes identical computations over tensors of fixed shape. We therefore package the entire expert transformer core into a single launch unit dispatched once per denoising step. Input preparation, dynamic tensor copies, and action-state updates remain outside this unit, while the vision-language backbone prefix runs strictly once per request. Combined with packed four-bit storage and fused decoding, this runtime lowers memory footprint and request latency without adding arithmetic approximation beyond the quantized parameters. Section 4.4 measures both effects.

Table 1: Closed-loop success rate (%) and resident weight memory (GiB) on LIBERO across precision regimes. All evaluations use a single seed; each configuration maintains its native activation quantizer and calibration settings. Bold marks the highest quantized success in each suite column for each model.

Model	Method	Venue	Precision	Mem. (GiB)	SPAT.	OBJ.	GOAL	LONG	Avg.↑
π0.5	BF16	–	BF16	4.17	98.0	99.0	96.5	95.0	97.1
	OPTQ	ICLR'23	W4A16	1.23	86.5	87.0	74.0	75.0	80.6
	AWQ	MLSys'24	W4A16	1.23	86.5	87.5	71.5	83.5	82.3
	OmniQuant	ICLR'24	W4A16	1.23	86.5	88.0	73.0	79.0	81.6
	SqueezeLLM	ICML'24	W4A16	1.27	93.5	94.5	82.0	80.0	87.5
	ActQuant	arXiv'26	W2.5A16	1.74	95.0	98.5	76.0	75.0	86.1
	QVLA	ICLR'26	W2.5A16	1.74	94.0	98.5	72.0	56.0	80.1
	SmoothQuant	ICML'23	W4A8	1.07	89.0	93.0	96.0	90.0	92.0
	DuQuant	NeurIPS'24	W4A8	1.17	97.0	98.5	94.0	89.0	94.6
	QuantVLA	CVPR'26	W4A8	1.17	84.0	98.5	80.0	67.0	82.4
	ZSQ	ICML'26W	W4A8	1.28	94.0	98.5	95.0	91.5	94.8
	LoopQuant	ours	W4A8	1.17	**99.0**	**99.5**	**97.5**	**93.0**	**97.3**
GR00T-N1.5	FP16	–	FP16	2.02	88.0	98.0	82.0	80.0	87.0
	OPTQ	ICLR'23	W4A16	0.74	79.5	82.0	78.0	70.0	77.4
	AWQ	MLSys'24	W4A16	0.74	80.0	88.5	50.5	74.5	73.4
	OmniQuant	ICLR'24	W4A16	0.74	80.0	68.0	85.5	72.0	76.4
	SqueezeLLM	ICML'24	W4A16	0.76	77.5	85.0	84.0	74.0	80.1
	ActQuant	arXiv'26	W2.5A16	0.76	79.5	88.0	64.0	57.0	72.1
	QVLA	ICLR'26	W2.5A16	0.76	76.0	84.5	61.5	43.0	66.3
	SmoothQuant	ICML'23	W4A8	0.68	41.5	76.0	50.0	67.0	58.6
	DuQuant	NeurIPS'24	W4A8	0.73	76.0	84.5	84.0	76.5	80.3
	QuantVLA	CVPR'26	W4A8	0.74	62.5	78.0	74.0	76.5	72.8
	ZSQ	ICML'26W	W4A8	0.91	81.0	86.0	77.0	67.0	77.8
	LoopQuant	ours	W4A8	0.61	**82.0**	**90.0**	**88.0**	**77.0**	**84.3**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate LoopQuant on two representative open-weight VLA models: π0.5 (Physical Intelligence et al., 2025) (252 quantized transformer projections) and GR00T-N1.5 (Bjorck et al., 2025) (180 quantized transformer projections). Both are evaluated in closed loop across all four LIBERO suites (Liu et al., 2023), evaluating 200 episodes per suite (800 per configuration) on an NVIDIA RTX 5090 GPU.

We quantize all transformer attention and MLP projections in the language backbone and action expert to W4A8; vision encoders, embeddings, input and state projections, normalizations, and action heads remain in 16-bit half precision. LoopQuant sets its static activation clipping thresholds via RMAEC (Eq. 3) using $M = 32$ frames drawn from held-out demonstrations of tasks 0–7 in each suite (sampling protocol in Appendix C.2). We compare against weight-only baselines (OPTQ, AWQ, OmniQuant, SqueezeLLM), prior W4A8 methods (SmoothQuant, DuQuant, QuantVLA, ZSQ), and selective-quantization schemes (ActQuant (Akbari et al., 2026), QVLA (Xu et al., 2026)). For a fair comparison, we align the quantized layers of some baselines with ours, such as the action-expert attention projections in QuantVLA, and keep all other settings from the original papers. Closed-loop rollouts execute in PyTorch via quantize–dequantize operators.

4.2 MAIN RESULTS

In Table 1, weight memory covers the targeted linear projections and their quantization metadata. Section 4.4 reports the full runtime footprint. Across both architectures, LoopQuant attains the highest average success among all quantized methods. On π0.5, it reaches 97.3%, matching the unquantized BF16 baseline (97.1%) while reducing weight footprint from 4.17 to 1.17 GiB. On GR00T-N1.5, it attains 84.3%, within three points of FP16 (87.0%), and reduces memory from 2.02

Table 2: Component ablation at W4A8, averaged across the four LIBERO suites. Built upon QuantVLA as the base framework, we evaluate the exact codebook and RMAEC individually and jointly, holding all other hyper-parameters fixed. Codebook: ✓ exact codebook (Section 3.2), ✗ uniform grid. RMAEC: ✓ thresholds from M held-out observations, ✗ the first-forward baseline. Δ: gain over the row with neither component, computed from the displayed success rates.

Model	Codebook	RMAEC	Success (%)	Δ (pp)
π0.5	✗	✗	82.4	–
	✓	✗	90.8	+8.4
	✗	✓	97.1	+14.7
	✓	✓	**97.3**	**+14.9**
GR00T-N1.5	✗	✗	72.8	–
	✓	✗	80.5	+7.7
	✗	✓	81.0	+8.2
	✓	✓	**84.3**	**+11.5**

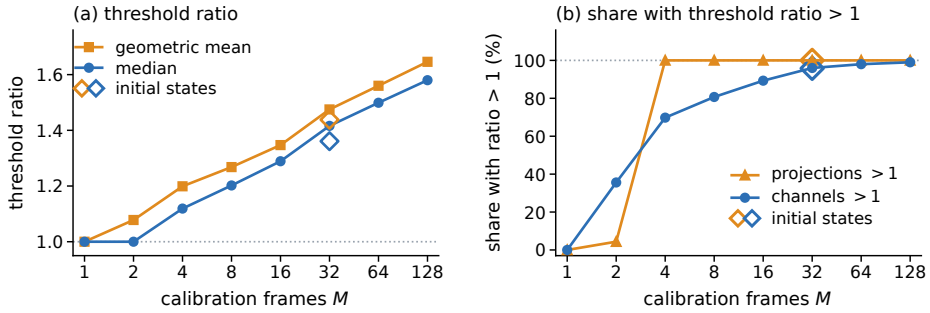

Figure 3: Clipping-threshold ratio relative to $M = 1$ across all 718,848 channels in π0.5. **(a)** Median and geometric-mean threshold ratios across calibration frame budgets. **(b)** Proportion of channels with ratio > 1 and of projections with median channel ratio > 1, reaching 96.0% and 100% at $M = 32$. Diamonds denote the initial-state control.

to 0.61 GiB. LoopQuant outperforms the strongest competing quantized baseline by 2.5 pp on π0.5 and 4.0 pp on GR00T-N1.5. Tables 2 and 3 next isolate the contributions of codebook construction and RMAEC calibration.

4.3 Ablations and Analysis

Contribution of each component. In Table 2, exact codebooks and RMAEC each raise success on both models over uniform grids and the first-forward baseline, respectively, and perform best together. RMAEC drives most of the gain on π0.5, while the exact codebook adds 3.3 pp over RMAEC on GR00T-N1.5. On π0.5, both codebook rows fit centroids with activation second moments from held-out frames. Against uniform grids, the exact codebook cuts weighted reconstruction error by a median of 39.3% per projection on π0.5 and 38.5% on GR00T-N1.5.

Frame coverage and aggregation. Table 3 decouples frame coverage from aggregation. Covering ten steps on a single frame yields negligible gain (82.5% vs. 82.4%), whereas RMAEC outperforms a 99.9th percentile pooled over the same 32 frames by 12.1 pp. Closed-loop success thus requires diverse observations and an extremal operator preserving per-request envelopes.

Threshold growth with the calibration budget. Figure 3 and Table 5 report threshold ratios across all quantized channels in π0.5. Because all budgets maximize over the same J denoising steps, these ratios isolate observation diversity. Quadrupling the budget from $M = 32$ to $M = 128$ raises the median ratio only from 1.416 to 1.580, showing that quantization steps coarsen slowly with added frames. Frame timing matters far less than frame count: at $M = 32$, calibrating on initial frames gives a median ratio of 1.361 against 1.416 for mid-episode frames, and the channel-wise median of $s_{\text{mid}}/s_{\text{init}}$ is 1.015.

Table 3: Calibration ablation on $\pi 0.5$ with uniform W4 weights (800 LIBERO episodes per configuration; both 32-frame rows use the same observations).

Calibration	Frames	Steps/frame	Success (%)	Successes
First-forward baseline	–	First call	82.4	659/800
Single-frame envelope	1	10	82.5	660/800
Pooled percentile	32	10	85.0	680/800
RMAEC	32	10	**97.1**	**777/800**

Table 4: Inference latency and resident GPU memory on $\pi 0.5$ (RTX 5090, 10 denoising steps, 30 timed requests, half-precision GEMM). Resident memory includes model weights and retained runtime tensors.

Configuration	Latency (s)	s.d.	Speedup	Mem. vs. BF16
BF16, unquantized	0.290	0.007	$1.00\times$	$1.00\times$
LoopQuant, W4A8, packed + single-launch core	**0.154**	0.002	$\mathbf{1.88}\times$	$\mathbf{0.59}\times$

Activation clipping in closed loop. Across 32 evaluation episodes (960 requests), a single-frame envelope reduces clipped mass in the action expert from 29.78% (first-forward baseline, Table 10) to 6.98%, yet barely improves success (82.5% vs. 82.4%). In contrast, RMAEC suppresses clipping to 2.14% (Table 10) while elevating success to 97.1%. Aggregate clipped mass thus cannot explain closed-loop performance: preventing outlier distortion across unseen trajectories is paramount.

4.4 Latency and Memory Analysis

Table 4 shows that LoopQuant at W4A8 processes an action request in $0.154\,\text{s}$, $1.88\times$ faster than the $0.290\,\text{s}$ of unquantized BF16, using $4.100\,\text{GiB}$ of resident GPU memory ($0.59\times$ the BF16 footprint). Table 8 separates the retained unquantized modules, packed linear layers, and other runtime allocations. Appendix C.4 shows that the latency gain comes from the denoising steps of the action expert.

While Table 1 evaluates in-place decoding in simulation, our packed runtime matches this reference bit for bit across all 252 projections and emitted actions on the timed benchmark requests.

Evaluation scope. All closed-loop rollouts use LIBERO simulation with one seed per configuration; runtime benchmarks are collected on a desktop RTX 5090 GPU. Paired ablations and statistical resolution are detailed in Appendix C.5.

5 Conclusion

In this work, we traced W4A8 quantization errors through the two nested loops of VLA inference. We showed that channel activation peaks drift across successive requests, and that an extreme-value model captures the sub-linear threshold growth with calibration frames. Grounded in this analysis, we introduced LoopQuant, a W4A8 PTQ framework tailored to these loops. For activations, RMAEC aggregates diverse observations while preserving per-request extrema, with ablations confirming that both broad coverage and extremal aggregation are critical to closed-loop success. For weights, an exact non-uniform codebook determines globally optimal four-bit levels under fixed row scales. Evaluations on $\pi 0.5$ and GR00T-N1.5 show that LoopQuant nearly matches the closed-loop success of unquantized models. At runtime, packed storage reduces resident weight memory, while launching the expert transformer core once per denoising step minimizes dispatch overhead, enabling LoopQuant to serve requests faster than the unquantized BF16 model.

AI USE STATEMENT

We used generative AI tools in three ways: to revise and polish the manuscript text, to retrieve and summarize relevant prior work, and to support the research, where they provided feedback on the feasibility of the method, assisted in designing experimental protocols, and helped optimize the experimental and kernel code. We reviewed all AI-assisted output and take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

Section 3.1 defines the two-stage activation calibration and its frame budget M. Section 3.2 gives the weight quantization objective and the codebook construction, and Section 4.3 describes the ablation protocol. Appendix A states the exact solver and checks it against a reference implementation. It also compares the reconstruction error of the exact codebook with that of a uniform grid. Appendix B measures how the thresholds grow with the calibration budget. Appendix C.2 specifies the evaluation protocol, the quantized layers, and the percentile level, and names the identifiers that select RMAEC and the first-forward baseline in our implementation. Appendix C.3 gives the timing protocol, Appendix C.1 the bit-exactness check of the fused decode, and Appendix C.5 the scope of the closed-loop results. The supplementary material contains anonymized modular source code.

REFERENCES

Alok Aggarwal, Maria M. Klawe, Shlomo Moran, Peter W. Shor, and Robert E. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(2):195–208, 1987. doi: 10.1007/BF01840359. URL https://doi.org/10.1007/BF01840359.

Arash Akbari, Arman Akbari, Masih Eskandar, Qitao Tan, Yixiao Chen, Jingwu Luo, Bertha Pangaribuan, Liyun Zhang, Jennifer G. Dy, Geng Yuan, Xue Lin, Gaowen Liu, Stratis Ioannidis, and Yanzhi Wang. Actquant: Sub-4-bit action-guided quantization for vision-language-action models. *CoRR*, abs/2605.24011, 2026. doi: 10.48550/ARXIV.2605.24011. URL https://doi.org/10.48550/arXiv.2605.24011.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/b5b939436789f76f08b9d0da5e81af7c-Abstract-Conference.html.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith LLontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: an open foundation model for generalist humanoid robots. *CoRR*, abs/2503.14734, 2025. doi: 10.48550/ARXIV.2503.14734. URL https://doi.org/10.48550/arXiv.2503.14734.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024. doi: 10.48550/ARXIV.2410.24164. URL https://doi.org/10.48550/arXiv.2410.24164.

Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers. In

IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025, pp. 28306–28315. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02636. URL https://openaccess.thecvf.com/content/CVPR2025/html/Chen_Q-DiT_Accurate_Post-Training_Quantization_for_Diffusion_Transformers_CVPR_2025_paper.html.

Stuart Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer Series in Statistics. Springer, London, 2001. doi: 10.1007/978-1-4471-3675-0.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract-Conference.html.

Peijie Dong, Lujun Li, Yuedong Zhong, Dayou Du, Ruibo Fan, Yuhan Chen, Zhenheng Tang, Qiang Wang, Wei Xue, Yike Guo, and Xiaowen Chu. STBLLM: breaking the 1-bit barrier with structured binary llms. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=6XUSDvBFkV.

Mostafa Elhoushi and Jeff Johnson. any4: Learned 4-bit numeric representation for llms. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/elhoushi25a.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Allan Grønlund, Kasper Green Larsen, Alexander Mathiasen, and Jesper Sindahl Nielsen. Fast exact k-means, k-medians and bregman divergence clustering in 1d. *CoRR*, abs/1701.07204, 2017. URL http://arxiv.org/abs/1701.07204.

Sung-hwan Han and Youngmin Yi. Zero-shot quantization for vision-language-action models via trajectory curvature and attention guidance. In *AdaptFM: Resource-Adaptive Foundation Model Inference Workshop at ICML 2026*, 2026. URL https://openreview.net/forum?id=9Ljgbk6H3O.

Xing Hu, Yuan Cheng, Dawei Yang, Zhixuan Chen, Zukang Xu, Jiangyong Yu, Chen Xu, Zhihang Yuan, Zhe Jiang, and Sifan Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=rAcgDBdKnP.

Yixin Ji, Yang Xiang, Juntao Li, Qingrong Xia, Ping Li, Xinyu Duan, Zhefeng Wang, and Min Zhang. Beware of calibration data for pruning large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=x83w6yGIWb.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 13:162467–162504, 2025. doi: 10.1109/ACCESS.2025.3609980. URL https://doi.org/10.1109/ACCESS.2025.3609980.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23901–23923. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/kim24f.html.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900. PMLR, 2022. URL https://proceedings.mlr.press/v162/li22n.html.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *CoRR*, abs/2411.05007, 2024. doi: 10.48550/ARXIV.2411.05007. URL https://doi.org/10.48550/arXiv.2411.05007.

Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. MBQ: modality-balanced quantization for large vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 4167–4177. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00394. URL https://openaccess.thecvf.com/content/CVPR2025/html/Li_MBQ_Modality-Balanced_Quantization_for_Large_Vision-Language_Models_CVPR_2025_paper.html.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 17489–17499. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01608. URL https://doi.org/10.1109/ICCV51070.2023.01608.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/9febda1c8344cc5f2d51713964864e93-Abstract-Conference.html.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: activation-aware weight quantization for on-device LLM compression and acceleration. In Phillip B. Gibbons, Gennady Pekhimenko, and Christopher De Sa (eds.), *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org, 2024b. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10*

- *16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. VPTQ: extreme low-bit vector post-training quantization for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 8181–8196. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.467. URL https://doi.org/10.18653/v1/2024.emnlp-main.467.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 28092–28103, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ec8956637a99787bd197eacd77acce5e-Abstract.html.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7197–7206. PMLR, 2020. URL http://proceedings.mlr.press/v119/nagel20a.html.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *CoRR*, abs/2106.08295, 2021. URL https://arxiv.org/abs/2106.08295.

Seongmin Park, Hyungmin Kim, Wonseok Jeon, Juyoung Yang, Byeongwook Jeon, Yoonseon Oh, and Jungwook Choi. Quantization-aware imitation-learning for resource-efficient robotic control. *CoRR*, abs/2412.01034, 2024. doi: 10.48550/ARXIV.2412.01034. URL https://doi.org/10.48550/arXiv.2412.01034.

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 *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 13140–13150. IEEE, 2025. doi: 10.1109/ICCV51701.2025.01221. URL https://doi.org/10.1109/ICCV51701.2025.01221.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.org/10.1109/ICCV51070.2023.00387.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054, 2025. doi: 10.48550/ARXIV.2504.16054. URL https://doi.org/10.48550/arXiv.2504.16054.

Stephane Ross and Drew Bagnell. Efficient reductions for imitation learning. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 661–668, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/ross10a.html.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

Hui Shen, Jingxuan Zhang, Boning Xiong, Rui Hu, Shoufa Chen, Zhongwei Wan, Xin Wang, Yu Zhang, Zixuan Gong, Guangyin Bao, Chaofan Tao, Yongfeng Huang, Ye Yuan, and Mi Zhang. Efficient diffusion models: A survey. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=wHECkBOwyt.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andrés Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Rémi Cadène. Smolvla: A vision-language-action model for affordable and efficient robotics. *CoRR*, abs/2506.01844, 2025. doi: 10.48550/ARXIV.2506.01844. URL https://doi.org/10.48550/arXiv.2506.01844.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. Flatquant: Flatness matters for LLM quantization. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/sun25l.html.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-VLM: post-training quantization for large vision-language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/cffbaf4f47546ece96bb42c0edda40ee-Abstract-Conference.html.

Haizhou Wang and Mingzhou Song. Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *R J.*, 3(2):29, 2011. doi: 10.32614/RJ-2011-015. URL https://doi.org/10.32614/rj-2011-015.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *CoRR*, abs/2409.12514, 2024. doi: 10.48550/ARXIV.2409.12514. URL https://doi.org/10.48550/arXiv.2409.12514.

Samuel S. Wilks. Determination of sample sizes for setting tolerance limits. *The Annals of Mathematical Statistics*, 12(1):91–96, 1941. doi: 10.1214/aoms/1177731788.

Miles Williams and Nikolaos Aletras. On the impact of calibration data in post-training quantization and pruning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pp. 10100–10118. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.544. URL https://doi.org/10.18653/v1/2024.acl-long.544.

Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/72d32f4fe0b7af03732bd227bf1c4a5f-Abstract-Conference.html.

Xiaolin Wu. Optimal quantization by matrix searching. *J. Algorithms*, 12(4):663–673, 1991. doi: 10.1016/0196-6774(91)90039-2. URL https://doi.org/10.1016/0196-6774(91)90039-2.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. Vla-cache: Towards efficient vision-language-action model via adaptive token caching in robotic manipulation. *CoRR*, abs/2502.02175, 2025. doi: 10.48550/ARXIV.2502.02175. URL https://doi.org/10.48550/arXiv.2502.02175.

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. In *International Conference on Learning Representations, ICLR 2026*, 2026. doi: 10.48550/ARXIV.2602.03782. URL https://doi.org/10.48550/arXiv.2602.03782.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025a. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Lianwei Yang, Haisong Gong, and Qingyi Gu. Dopq-vit: Towards distribution-friendly and outlier-aware post-training quantization for vision transformers. *CoRR*, abs/2408.03291, 2024. doi: 10.48550/ARXIV.2408.03291. URL https://doi.org/10.48550/arXiv.2408.03291.

Lianwei Yang, Haokun Lin, Tianchen Zhao, Yichen Wu, Hongyu Zhu, Ruiqi Xie, Zhenan Sun, Yu Wang, and Qingyi Gu. Lrq-dit: Log-rotation post-training quantization of diffusion transformers for text-to-image generation. *CoRR*, abs/2508.03485, 2025b. doi: 10.48550/ARXIV.2508.03485. URL https://doi.org/10.48550/arXiv.2508.03485.

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. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025c. URL http://papers.nips.cc/paper_files/paper/2025/hash/3a2ef31a1e45908901adc0ca853a8faf-Abstract-Conference.html.

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. RPTQ: reorder-based post-training quantization for large language models. *CoRR*, abs/2304.01089, 2023. doi: 10.48550/ARXIV.2304.01089. URL https://doi.org/10.48550/arXiv.2304.01089.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11941–11952. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01100. URL https://doi.org/10.1109/ICCV51070.2023.01100.

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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2026*, 2026a. doi: 10.48550/ARXIV.2602.20309. URL https://doi.org/10.48550/arXiv.2602.20309.

Rongyu Zhang, Menghang Dong, Yuan Zhang, Liang Heng, Xiaowei Chi, Gaole Dai, Li Du, Dan Wang, Yuan Du, and Shanghang Zhang. Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 18764–18772. AAAI Press, 2026b. doi: 10.1609/AAAI.V40I22.38945. URL https://doi.org/10.1609/aaai.v40i22.38945.

Tianchen Zhao, Xuefei Ning, Tongcheng Fang, Enshu Liu, Guyue Huang, Zinan Lin, Shengen Yan, Guohao Dai, and Yu Wang. Mixdq: Memory-efficient few-step text-to-image diffusion models with metric-decoupled mixed precision quantization. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XIV*, volume 15072 of *Lecture Notes in Computer Science*, pp. 285–302. Springer, 2024. doi: 10.1007/978-3-031-72630-9_17. URL https://doi.org/10.1007/978-3-031-72630-9_17.

Tianchen Zhao, Tongcheng Fang, Haofeng Huang, Rui Wan, Widyadewi Soedarmadji, Enshu Liu, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=E1N1oxd63b.

Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, Zhiquan Qi, Yitao Liang, Yuanpei Chen, and Yaodong Yang. A survey on vision-language-action models: An action tokenization perspective. *CoRR*, abs/2507.01925, 2025. doi: 10.48550/ARXIV.2507.01925. URL https://doi.org/10.48550/arXiv.2507.01925.

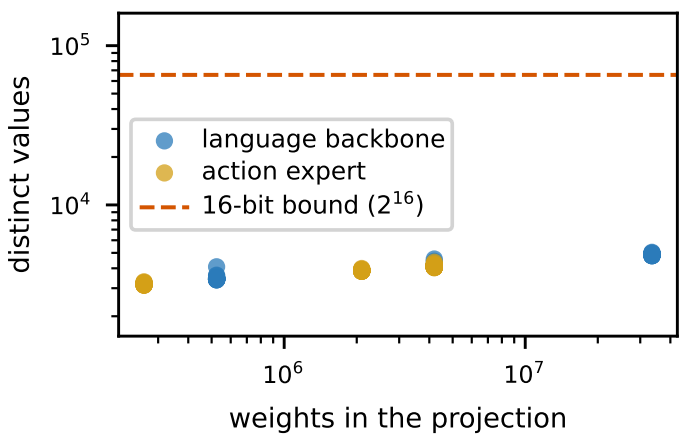

Figure 4: Distinct normalized weight values across projections in $\pi 0.5$. Each projection exhibits only a few thousand unique values, well below the theoretical limit of 2^{16} representable bit patterns.

A Verifying the Exact Solver

Algorithm 1 details the complete codebook construction of Section 3.2. The grid search for each row scale α_r evaluates five candidate values.

Algorithm 1 Exact Non-Uniform Codebook Construction

Require: Transformed weight $\mathbf{W}$, second moments h, row scales α, codebook size k
Ensure: Centers $\gamma_1 < \cdots < \gamma_k$, codes q

$\widetilde{\mathbf{W}} \leftarrow \mathbf{W} \oslash \alpha$
Group equal values of $\widetilde{\mathbf{W}}$; sum $h_c \alpha_r^2$ within each group
$(\mathcal{V}, \omega) \leftarrow$ sorted distinct values and positive group weights; $N \leftarrow |\mathcal{V}|$
Build prefix sums of ω, ωv, and ωv^2
$D_1(j) \leftarrow \mathrm{cost}(1, j)$ for $j = 1, \ldots, N$
for $\ell = 2$ to k **do**
    $D_\ell \leftarrow \textsc{SmawkRowMinima}(D_{\ell-1}, \mathrm{cost})$
end for
Backtrack $k-1$ splits; $\gamma \leftarrow$ interval weighted means (Eq. 7)
$q(r, c) \leftarrow$ index of interval containing $\widetilde{W}_{rc}$
return γ, q

We verify the accelerated solver against a reference $O(N^2 k)$ implementation of Eq. 8. Across random sets of distinct values, both implementations yield identical splits, centers, and objective values.

A.1 Distinct Values in Half-Precision Weights

Throughout this work, half precision refers to the 16-bit floating-point checkpoint format. Weights and runtime measurements for $\pi 0.5$ use BF16 ($\mu = 7$ stored mantissa bits), whereas FP16 has $\mu = 10$. A binade is the interval between consecutive powers of two. Within each binade, a 16-bit format with μ stored mantissa bits provides 2^μ representable magnitudes. Consequently, the count N of distinct values in a tensor spanning B binades satisfies

$$N \;\le\; 2 \cdot 2^\mu \cdot B + 1, \tag{9}$$

accounting for signs and zero. In $\pi 0.5$, the 252 quantized projections hold 0.26–33.55 million weights each but only 3,131–5,040 distinct normalized values (median 4,077). The maximum count is merely 7.7% of the 2^{16} patterns representable by the format.

A.2 Reconstruction Error of the Exact Codebook

The comparison against uniform grids in Section 4.3 evaluates the weighted reconstruction error of Eq. 5. Row scales and channel second moments remain fixed, isolating the effect of changing the sixteen code values. Reported reductions relative to the uniform grid are medians across the 252 projections of $\pi 0.5$ and the 180 projections of GR00T-N1.5, measured on weights alone.

B Calibration Budget Analysis

The expected normal maximum a_M of Section 3.1 has the asymptotic expansion

$$a_M \;\sim\; \sqrt{2\ln M} - \frac{\ln(4\pi \ln M) - 2\gamma_{\text{Euler}}}{2\sqrt{2\ln M}}, \tag{10}$$

where $\gamma_{\text{Euler}} \approx 0.5772$ is Euler's constant. For our budget, numerical integration yields $a_{32} \approx 2.070$.

Raising the confidence of the tolerance calculation in Section 3.1 from 80% to 90% or 95% at the same $\beta = 0.05$ requires $M \geq 45$ or $M \geq 59$, respectively. Each guarantee applies pointwise to a single channel and does not hold simultaneously across all channels.

Because the task-stratified frames of Appendix C.2 can contain dependent observations, we treat the sample size from the tolerance calculation as a principled reference budget, while Section 4.3 directly evaluates its empirical coverage.

Table 6 evaluates Eq. 4 with dispersions estimated across all channels over 960 requests. Each estimate is the standard deviation of a log-percentile across the invocations of its projection: 960 calls for language projections and 9,600 for action-expert projections. For the action expert, the estimates reflect variance across both denoising steps and requests, whereas the theoretical model requires the dispersion of the per-request maximum across steps. Table entries present predictions conditioned on these dispersion estimates. At the median dispersion $\sigma_{\log P} = 0.249$, the model predicts $\kappa(1{\to}32) \approx 1.67$, against an empirically measured median of 1.416.

While Slepian's inequality indicates that positive correlations dampen maximum growth relative to independent variables, the empirical difference alone cannot fully confirm the underlying dependence conditions. The columns for $M^* = 320$ and $M^* = 1{,}000$ extrapolate beyond our empirical measurement range, which terminates at $M = 128$.

Descriptive growth and frame-position checks. Under the independent model, the across-channel average of $\log \kappa_c(1{\to}M)$ is $\bar{\sigma} a_M$, so the logarithm of the measured geometric-mean ratio G_M would grow in proportion to a_M. Table 7 fits this form. The ratio $\log G_M / a_M$ rises from 0.133 at $M = 2$ to 0.192 at $M = 128$, so the form holds only approximately. A least-squares fit of $\log G_M = \bar{\sigma} a_M$ over the seven budgets gives $\hat{\sigma} = 0.184$ and reproduces every G_M within 3%, above the measured value for $M \leq 16$ and below it for $M \geq 32$. We therefore use Eq. 4 as a descriptive model of threshold growth. The fitted $\hat{\sigma}$ lies below the median dispersion estimate of 0.249.

At $M = 32$, mid-episode thresholds exceed initial-state thresholds on 56.2% of channels. We incorporate frame position as an empirical control, because Eq. 4 contains no term for where a frame sits within its episode.

C Experimental Details

C.1 Bit-Exact Decode

The fused decoding kernel reproduces reference decoding bit for bit. PyTorch computes half-precision elementwise products by promoting operands to FP32, multiplying, and rounding once back to half precision. The fused Triton kernel executes this identical arithmetic sequence. Both paths return $+0$ when the row scale is zero, even if the paired codebook center is non-finite, and the fused kernel enforces this rule in FP32 before rounding back.

Table 5: Measured threshold growth relative to single-frame calibration at $M{=}1$ across all $718{,}848$ channels of $\pi0.5$, as plotted in Figure 3. Channels >1 is the share of channels with a ratio above one. Projections >1 counts the quantized projections, out of 252, whose median channel ratio exceeds one; Figure 3(b) plots this count as a percentage. At $M{=}32$, the median channel-wise ratio of mid-episode to initial-state thresholds is 1.015. The initial-state geometric mean is derived from measured geometric-mean ratios.

Calibration protocol	Median	Geo. mean	Channels >1	Projections >1
$M = 2$	1.000	1.078	35.6%	11
$M = 4$	1.119	1.199	69.8%	252
$M = 8$	1.202	1.268	80.7%	252
$M = 16$	1.289	1.347	89.3%	252
$M = 32$	1.416	1.475	96.0%	252
$M = 64$	1.499	1.560	98.0%	252
$M = 128$	1.580	1.646	99.0%	252
$M = 32$ (initial states)	1.361	1.438	95.8%	252

Table 6: Threshold growth $\kappa_c(M{\to}M^*)$ from the $M{=}32$ budget to larger budgets M^*, predicted by Eq. 4 across five quantiles of dispersions fitted over all $718{,}848$ channels of $\pi0.5$.

σ quantile	$\sigma_{\log P}$	$\to M^*{=}320$	$\to M^*{=}1{,}000$
p1	0.036	1.030	1.043
p25	0.154	1.136	1.198
p50	0.249	1.229	1.338
p75	0.374	1.364	1.551
p99	0.642	**1.703**	**2.122**

C.2 Evaluation Protocol

The LIBERO benchmark (Liu et al., 2023) contains four suites with ten manipulation tasks each: Spatial, Object, Goal, and Long. An episode terminates upon task success or when reaching the step limit of its suite. The quantization scope follows Section 4.1: all transformer attention and MLP projections in the language backbone and action expert use W4A8, whereas vision encoders, embeddings, input and state projections, normalizations, and action heads remain in half precision. Because the action expert processes only a compact sequence of action tokens at each denoising step, the 99.9th token percentile in Stage 1 closely tracks the per-step maximum on its projections.

Our implementation builds on the released QuantVLA codebase (Zhang et al., 2026a), where RMAEC and the first-forward baseline of Eq. 3 are designated `external_32_batch` and `legacy_first_forward`, respectively. We sample calibration frames deterministically, drawing one mid-episode frame per task for tasks 0–7 across all four suites. Demonstrations are held out, though their tasks appear during evaluation. Tasks 8 and 9 contribute no calibration observations. RMAEC runs these frames offline without stepping the environment simulator, freezing scales after all 32 frames. On $\pi0.5$, each frame executes $J = 10$ denoising steps. The first-forward baseline performs no separate calibration pass, freezing thresholds during the initial forward pass of closed-loop evaluation.

Results for the first-forward baseline and RMAEC in Table 3 are shared with Table 2. The single-frame control sets $M = 1$ in Eq. 3. The pooled-percentile control aggregates absolute activations across tokens, steps, and frames before taking the 99.9th percentile of each channel. All four configurations employ static thresholds. Success rates are computed from episode counts and rounded to one decimal place. Percentage-point gains are differences between the displayed success rates.

The threshold ratios in Figure 3 are calculated from stored channel thresholds. The frame-position control isolates frame timing by comparing deployed thresholds against a control set built using the identical procedure and frame budget, with every frame index shifted to zero. Recomputing thresholds at the original mid-episode positions reproduces deployed thresholds bit for bit across all 252 projections, verifying that the comparison strictly isolates frame position.

Table 7: Fit of the growth form in Eq. 4 to the geometric-mean ratios G_M of Table 5. Values of a_M are obtained via numerical integration; the last column is $\exp(\hat{\sigma} a_M)$ with the least-squares $\hat{\sigma} = 0.184$.

M	a_M	G_M	$\log G_M / a_M$	Fit
2	0.564	1.078	0.133	1.109
4	1.029	1.199	0.176	1.208
8	1.424	1.268	0.167	1.300
16	1.766	1.347	0.169	1.384
32	2.070	1.475	0.188	1.464
64	2.344	1.560	0.190	1.539
128	2.595	1.646	0.192	1.612

Table 8: Resident GPU memory breakdown of the packed π0.5 runtime. Ratios are relative to the total resident GPU memory of the unquantized BF16 baseline.

Component	GiB	Relative to BF16
Unquantized modules (vision encoder, embeddings, `lm_head`, input projections, norms, action heads)	2.788	0.400×
Packed linear layers and metadata	1.174	0.168×
Other runtime allocations	0.138	0.020×
Total	**4.100**	**0.588×**

Because calibration retains only the Stage 1 envelope maximum over denoising steps and discards per-step percentiles, all subsequent threshold and clipping evaluations use these resulting thresholds. Activation clipping is quantified at the exact upper saturation point.

C.3 Timing Protocol and the Two Builds

The NVIDIA RTX 5090 evaluated in Table 4 features compute capability 12.0 and runs CUDA 13.0 with the Triton backend. In timing measurements, we synchronize the device immediately before starting and stopping the timer, discard the first five requests as warmup runs, and fix the action horizon to ten.

During initialization, the quantizer registers discrete codes, centers, thresholds, and scales, then releases the source half-precision weights. The simulator build applies an eight-bit quantize–dequantize operation to activations for each half-precision matrix product, requiring 0.589 s per request within the PyTorch simulation loop. The packed build deploys the fused kernel runtime of Section 3.3. Both builds execute half-precision GEMM operations over an identical eight-bit activation grid.

In Table 4, the packed runtime and the unquantized half-precision baseline are benchmarked under matched conditions, sharing the physical device, request tensor dimensions, action horizon, number of denoising steps, and warmup and timing procedures.

The resident GPU memory in Table 4 includes the full model and retained runtime tensors. Table 8 details the 4.100 GiB footprint of the sealed packed-INT4 runtime with single-launch core dispatch. The 252 quantized projections account for 1.174 GiB, including packed weights, rotation matrices, scales, and permutations; Table 1 reports this component as 1.17 GiB.

C.4 Latency by Inference Phase

Table 9 times the two configurations of Table 4 phase by phase at batch size 1, under the protocol of Appendix C.3.

The speedup of the packed build comes from the denoising steps. One step takes 5.74 ms in the packed build and 20.3 ms in the BF16 model, so the ten steps of a request finish 145.5 ms sooner. The prefix lies outside the single launch unit of Section 3.3. It runs eagerly and decodes the quantized backbone weights before their matrix products, taking 31.9 ms longer than in the BF16 model.

Table 9: Mean latency in milliseconds of each inference phase on $\pi 0.5$ at batch size 1, under the setting of Table 4. The prefix is the vision-language backbone pass that opens a request, and a step is one action-expert pass.

Configuration	Prefix	Per step	10 steps	Request
BF16, unquantized	**63.3**	20.3	202.9	290
LoopQuant, W4A8, packed + single-launch core	95.2	**5.74**	**57.4**	**154**

Table 10: Activation clipping on $\pi 0.5$ over 32 unseen episodes (960 requests). Columns report the proportion of clipped activation elements and clipped absolute activation mass, pooled across all projections or restricted to the action-expert stack.

Calibration	Elements clipped	Mass clipped (pooled)	Mass clipped (action expert)
First-forward baseline	2.81%	7.43%	29.78%
RMAEC (held-out)	0.14%	0.63%	2.14%

Because the prefix runs once per request and the action expert runs at every denoising step, the saving over the ten steps outweighs this cost.

C.5 Scope of the Closed-Loop Results

Our experimental scope has several natural boundaries: evaluations are conducted in simulation rather than on physical robot hardware or secondary benchmarks, and execution on embedded robotic edge accelerators remains untested. In Section 4.3, activation clipping is measured on the arm with uniform weight grids, and frame position is studied directly on threshold magnitudes.

The GR00T-N1.5 configurations of Table 2 execute within a unified job on a single request stream. Every episode index therefore selects the identical task, initial state, and protocol seed across all configurations, ensuring matched-pair comparisons. On $\pi 0.5$, arms evaluating RMAEC and the first-forward baseline are paired analogously. Unquantized reference runs are conducted independently over the identical episode suite.

Statistical resolution of reported success rates depends on the underlying episode count. On $\pi 0.5$, the 95% Wilson score interval for a success rate near 97% across 800 episodes has a half-width of approximately 1.2 pp. This margin provides scale for assessing the single-episode gap to the BF16 reference, though it does not capture variability across independent evaluation seeds.

On tasks 8 and 9, which receive no calibration frames, LoopQuant succeeds 94.4% of the time, compared to 98.0% on calibration-represented tasks 0–7.

Figure 5 breaks down the pooled clipped mass of Table 10 across projection types. Panel (a) shows clipped activation mass for each of the seven projection types in a transformer block under RMAEC and the first-forward baseline, reported separately for the language backbone and action expert. Under the first-forward baseline, every projection type in the action expert loses approximately a quarter or more of its activation mass, whereas corresponding language-backbone projections lose a much smaller share. In contrast, held-out RMAEC calibration keeps all fourteen groups below 4%. A single-frame envelope clips 4.77% of pooled mass, confirming that the reduction to 0.63% under RMAEC directly stems from scaling the frame budget.

Panel (b) plots clipped mass under the first-forward baseline against $\sigma_{\log P}$, the empirical dispersion of a projection's clipping percentile across invocations. Because action-expert projections are invoked once per denoising step, their dispersion reflects variation across both steps and requests. Action-expert groups exhibit both the largest dispersion and the highest clipped mass. The fourteen points cluster into two groups corresponding to the two stacks, so overall correlation primarily reflects this stack separation. Within the language backbone, where one invocation corresponds to one request, the seven groups correlate strongly with dispersion (Pearson $r = 0.81$), whereas intra-stack ordering in the action expert is weaker. Across all 252 projections, a linear regression of clipped mass on $\sigma_{\log P}$ with a stack indicator yields a statistically significant positive coefficient for disper-

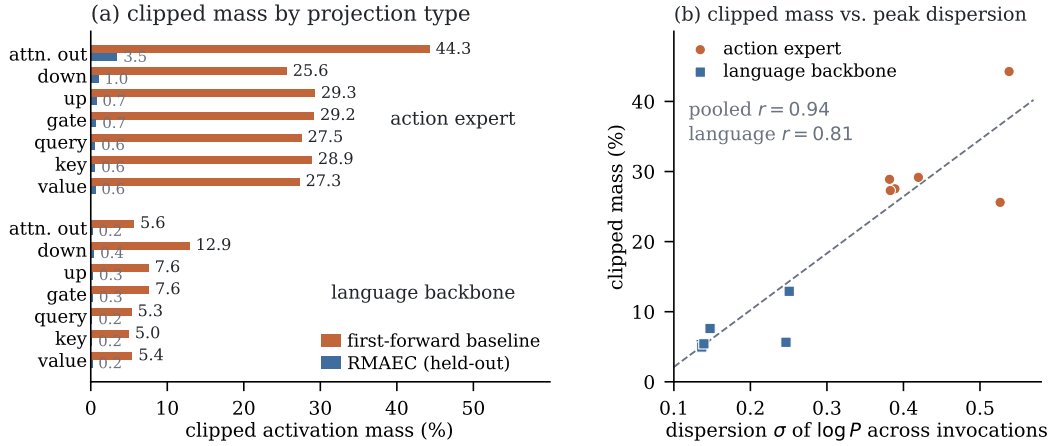

Figure 5: Activation clipping analysis under static calibration schemes. **(a)** Clipped activation mass across projection types and stacks for the first-forward baseline and RMAEC. **(b)** Clipped mass under the first-forward baseline versus invocation dispersion $\sigma_{\log P}$ across the fourteen projection–stack groups. The pooled Pearson correlation r is computed over all fourteen groups, while the language correlation r is evaluated over the seven groups of the language backbone.

sion ($t = 6.6$). This association demonstrates where static thresholds frozen from a single forward pass are most detrimental.

D Qualitative Rollout Comparison

Figure 6 shows closed-loop rollouts of π0.5 and GR00T-N1.5 with unquantized weights and under W4A8 LoopQuant. For each model, we select one task from each of the four LIBERO suites (Spatial, Object, Goal, Long). Each panel displays six chronological frames in a 2×3 grid: unquantized baseline on top, LoopQuant below, with early, middle, and late manipulation stages from left to right.

In each pair, the unquantized and quantized models receive identical instructions and scene initializations. Both reliably identify the target object, navigate the end-effector toward it, execute required interactions, and attain the target configuration, such as placing a bowl on a plate, putting ketchup in a basket, moving a bowl to cabinet storage, sliding a plate in front of the stove, or placing two objects in a basket. While paired rollouts naturally exhibit minor variations in joint poses, end-effector positions, approach angles, and intermediate trajectories, LoopQuant faithfully preserves manipulation capabilities without instability or failure. These eight pairs illustrate execution dynamics on selected tasks, while success rates in Table 1 evaluate comprehensive task completion across the benchmark.

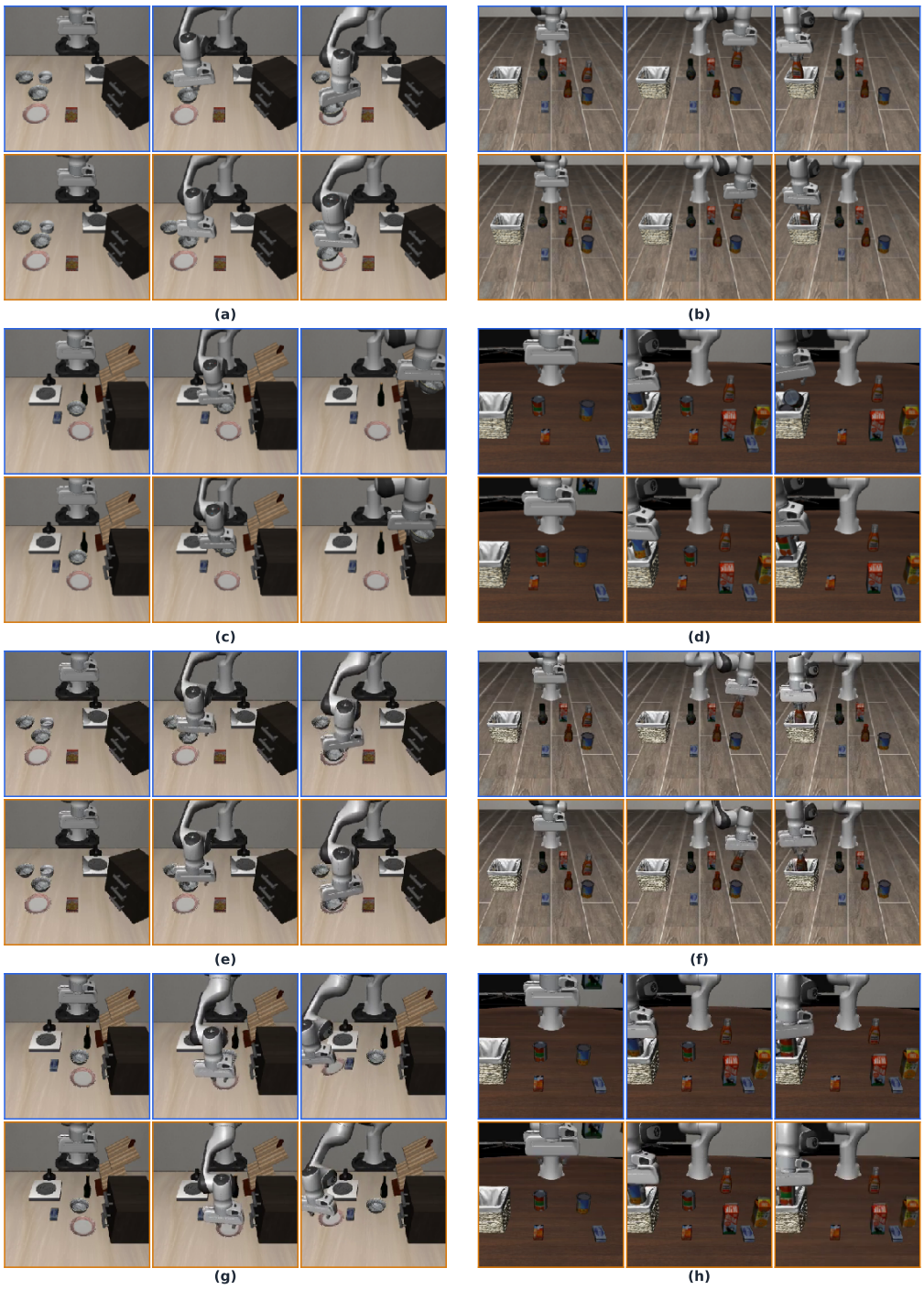

Figure 6: Closed-loop LIBERO rollouts of π0.5 (a–d) and GR00T-N1.5 (e–h), three frames per row from early (left) to late (right): unquantized reference on top (blue), W4A8 LoopQuant below (orange). Tasks: SPATIAL (a, e), OBJECT (b, f), GOAL (c, g), LONG (d, h). All tasks shown succeed under quantization.