# CRAFT: COUPLED ROTATION-AWARE ACTION-FIELD CACHING AND TWO-BIT QUANTIZATION FOR VISION-LANGUAGE-ACTION MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Diffusion and flow-matching vision–language–action (VLA) policies incur substantial weight traffic through repeated action-head evaluations. Prior VLA quantization typically retains head weights at W4 or higher, while scalar W2 can severely degrade task success. Adaptive caching introduces per-step decision overhead that can offset savings on short trajectories ($T \leq 10$), despite often selecting identical evaluation steps. We introduce Coupled Rotation-Aware Action-Field Caching and Two-Bit Quantization (*CRAFT*), a training-free framework that jointly reduces action-head precision and evaluation frequency. Motivated by scalar-W2 error concentration in a few high-activation-energy directions, *CRAFT* quantizes the head to W2A4 using lattice codewords. It selects modules for rotation before deployment using the original weights' tail statistics and reuses the latest predicted field at fixed steps, skipping head evaluation while continuing solver-state updates. With a W4 backbone, W2A4 head, and caching, *CRAFT* achieves LIBERO-Object success rates of 98.4% and 97.8% on GR00T N1.5 and $\pi_{0.5}$ (FP16: 96.4% and 98.0%). Batch-1 end-to-end speedups on RTX 3090 reach $2.06\times$ and $2.46\times$, respectively, versus $1.40\times$ and $1.38\times$ for the fastest prior methods. On GR00T N1.5, nominal weight traffic decreases by 79.6% relative to FP16 and by a factor of 2.1–3.2 relative to prior methods. Head-only W2A4 quantization achieves 80.45% average success across eight models, exceeding the strongest matched-precision baseline by 15.3 percentage points.

## 1 INTRODUCTION

Vision–language–action (VLA) models generate robot actions from visual observations and language instructions (Brohan et al., 2023; Kim et al., 2024), and recent VLAs refine actions over denoising steps with a diffusion or flow-matching action head (Li et al., 2024a; Bjorck et al., 2025; Black et al., 2025). Closed-loop control of a single robot typically uses batch-1 inference because each policy call depends on observations following previous actions. Limited weight reuse at batch 1 lowers arithmetic intensity, making weight transfers a major latency bottleneck on bandwidth-limited GPUs (Lee et al., 2025). We therefore compare compressed configurations by per-call weight traffic and closed-loop success rate (Figure 1). In the default configuration, the backbone runs once per call and the head at every denoising step ($T$ head evaluations, NFE), so different modules dominate storage and traffic or latency (Figure 2): the GR00T N1.5 action head accounts for 39% of the weight storage but 73% of the traffic and 76% of the summed module execution time, measured without concurrent GPU workloads. Reducing both the precision and evaluation count of the head therefore multiplies their effects on its per-call weight traffic.

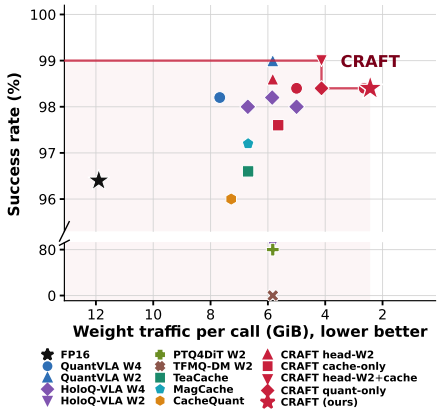


Figure 1: Success rate vs. weight traffic per policy call (GR00T N1.5).



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.

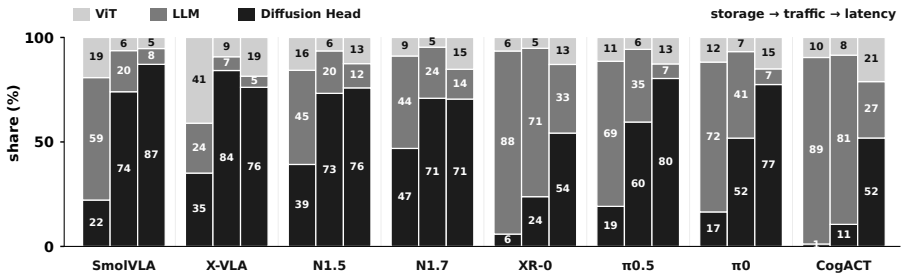


Figure 2: Module shares of storage, weight traffic, and latency across eight diffusion-head VLAs.

Prior VLA quantization quantizes the backbone but keeps the head at W4 or above (Zhang et al., 2026; Wang et al., 2026). A W2 head would cut traffic, but scalar and diffusion post-training quantization (PTQ) at W2A4 drive the closed-loop success rate (SR) of several models close to zero (Figure 1, bottom), unexplained by the weight reconstruction error alone. The output error of scalar W2 concentrates on a few high-energy eigendirections, which existing VLA quantization methods do not address. Moreover, rotating all modules can amplify extreme values in modules whose weights have a lighter tail than a Gaussian, i.e., rarer extremes, so rotation must be decided per module (***Lim. 1***).

Denoising-step caching reduces head evaluations by reusing adjacent-step outputs, and adaptive caching decides reuse per step from the measured change (Liu et al., 2025a; Ma et al., 2026; Ye et al., 2024). With $T \leq 10$ in VLAs, however, few evaluations can be skipped, so the decision cost is large relative to the savings: on GR00T N1.5, the TeaCache decision alone adds $6.5$–$8.3\%$ to per-call latency (Appendix D.4), yet most calls select the same step set. Reusing the solver state halts its update, so what is reused determines the error (***Lim. 2***). Moreover, quantization and caching act on the same predicted field, so their joint degradation must be tested for additivity; prior quantization–caching hybrids cover only image diffusion (Liu et al., 2025c), leaving it unmeasured for low-bit VLA heads (***Lim. 3***).

To address these limitations, we propose Coupled Rotation-Aware Action-Field Caching and Two-Bit Quantization (*CRAFT*), which quantizes the head to W2A4 and reduces its evaluations without further training. Our contributions are: (i) **Extreme-W2 Diagnosis and Analysis-Guided Quantization**: we design the W2 representation and the rotation rule based on output-error concentration and the tail dependence of rotation (solution for ***Lim. 1***); (ii) **Short-Trajectory Whole-Head Evaluation Reduction**: we reuse the predicted field at skipped steps while updating the state (solution for ***Lim. 2***); (iii) **Coupled Precision–Execution Analysis**: we decompose how quantization error at head-evaluation steps (compute nodes) propagates through reuse intervals and measure the interaction over four closed-loop configurations (solution for ***Lim. 3***).

## 2 Background and Related Work

**Diffusion-Head VLA Policies.** A VLA policy encodes the observation and instruction into a condition $z$ with a vision encoder and a language backbone, and an action head generates a multi-step action chunk from $z$ (Brohan et al., 2023; Kim et al., 2024). Diffusion and flow-matching heads refine a noise-initialized solver state over $T$ steps: at step $t$, the head $g$ predicts $f_t = g(x_t, z, t)$ and the solver updates $x_{t+1} = \Phi_t(x_t, f_t)$. The final state $x_T$ is the action chunk to execute; in this formulation, the two families differ in what $g$ predicts (velocity or noise) and in the update $\Phi_t$. The head thus runs $T$ times per policy call whereas the backbone runs once, making its cost central to VLA compression.

**VLA Quantization.** VLA quantization methods, including QuantVLA (Zhang et al., 2026), HoloQ-VLA (Wang et al., 2026), BitVLA (Wang et al., 2025), QVLA (Xu et al., 2026), and Mix-QVLA (Ranjan & Savakis, 2026), correct backbone quantization error but keep the head at W4 or above. ActQuant (Akbari et al., 2026) keeps the action head in full precision, and QVLA and Mix-QVLA allocate bits by layer sensitivity. E8P lattice quantization and block rotation are low-bit LLM techniques from QuIP# (Tseng et al., 2024) and DuQuant (Lin et al., 2024a). Further reducing head precision, however, requires accounting for its timestep-dependent behavior.

**Diffusion Quantization.** Diffusion PTQ must handle timestep-varying activation statistics. PTQ4DiT (Wu et al., 2024) applies salience balancing to the timestep-wise variation of salient DiT channels, and TFMQ-DM (Huang et al., 2024) applies tailored reconstruction to the temporal information block that depends only on the timestep; both use scalar quantization. For VLAs, HoloQ-VLA (Wang et al., 2026) rotates all modules of the DiT action head and quantizes it to W4A4 with a step-wise activation-scale table. These methods target image generation or heads at W4 or above, and address neither the output-error structure of W2A4 heads nor the module-wise effect of rotation (***Lim. 1***).

**Diffusion Caching.** Head cost can also be lowered by reducing repeated evaluations, not only precision. TeaCache (Liu et al., 2025a) and MagCache (Ma et al., 2026) decide reuse from a per-step change signal, and AdaptiveDiffusion (Ye et al., 2024) selects update steps at run time; all three target long FP16 image and video diffusion trajectories. DeepCache (Ma et al., 2024), FORA (Selvaraju et al., 2024), and TaylorSeer (Liu et al., 2025b) instead reuse or Taylor-extrapolate features at fixed intervals without per-step decisions, and EfficientVLA (Yang et al., 2026) applies feature caching to the DiT action head of CogACT. For VLAs with $T \leq 10$, however, the per-step decision cost and the reuse target when skipping the whole head have not been analyzed together (***Lim. 2***).

**Hybrid Compression.** Since quantization and caching reduce the per-evaluation cost and the evaluation count of the head, respectively, they can also be optimized jointly. SQAP-VLA (Fang et al., 2025) combines W4A4 quantization with token pruning in the language backbone of CogACT and analyzes how quantization-distorted attention affects pruning. CacheQuant (Liu et al., 2025c) combines W8A8 quantization with step caching for image diffusion, choosing the schedule by dynamic programming and reducing both errors by affine correction. These methods reduce the number of backbone tokens and image-diffusion steps, respectively, but address neither jointly reducing the precision and evaluation count of the action head nor the closed-loop interaction of quantization and caching on the same predicted field (***Lim. 3***).

# 3 Analysis of Diffusion-Head VLA Compression

## 3.1 Output Error and Rotation Applicability

W2 error is usually measured by the weight reconstruction error $\|\Delta W\|_F^2/\|W\|_F^2$, but its effect on the head depends on the induced output error. An error $\Delta W$ induces $\Delta y = \Delta W x$ for input $x$; with the input second moment $H = \mathbb{E}[xx^\top]$, the expected output error energy and the relative error $\varepsilon$ are

$$\mathbb{E}\big[\|\Delta y\|_2^2\big] = \mathrm{tr}\big(\Delta W H \Delta W^\top\big), \qquad \varepsilon = \frac{\mathrm{tr}(\Delta W H \Delta W^\top)}{\mathrm{tr}(W H W^\top)}. \tag{1}$$

The eigendecomposition $H = U\Lambda U^\top$ splits the output and weight errors by direction:

$$\mathrm{tr}\big(\Delta W H \Delta W^\top\big) = \sum_k \lambda_k \, \|\Delta W u_k\|_2^2, \qquad \|\Delta W\|_F^2 = \sum_k \|\Delta W u_k\|_2^2, \tag{2}$$

where $(\lambda_k, u_k)$ are the eigenpairs of $H$. Since only the output error is weighted by $\lambda_k$, the two errors are proportional for all $\Delta W$ only if $H = cI$. We measure concentration by the output-error share $\kappa = \sum_{k\in\mathcal{K}} \lambda_k \|\Delta W u_k\|_2^2 / \mathrm{tr}(\Delta W H \Delta W^\top)$ of the top-1% directions $\mathcal{K}$ by $\lambda_k$.

Figure 3 compares RTN, GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2024b), HoloQ-VLA (Wang et al., 2026), and our error-redistributed representation under W2 quantization. The top-1% directions account for 45.6% of $\mathrm{tr}(H)$ in the head and 78.8% in the language backbone. Under scalar W2, these directions carry 34–46% of the head's output error, in line with their activation-energy share, whereas error-redistributed W2 (last column) reduces this share to 3.1%. W2 therefore requires reducing the output error along high-energy directions, not only the weight reconstruction error.

Alongside output-error-aware quantization, we examine when rotating the weights pre-quantization helps. The effect of a block rotation $W' = \Theta_{\mathrm{out}} W \Theta$ depends on the tail of the original weights, which we quantify for module $m$'s matrix set $\mathcal{M}_m$ relative to Gaussian matrices of the same size:

$$\rho_\ell = \frac{\max_i |w_{\ell,i} - \bar{w}_\ell|}{\mathrm{RMS}(w_\ell - \bar{w}_\ell)}, \qquad \beta_\ell = \sqrt{2\ln(2N_\ell)}, \qquad D_{\mathrm{rot}}(m) = \frac{\mathrm{median}_{\ell\in\mathcal{M}_m}\, \rho_\ell}{\mathrm{median}_{\ell\in\mathcal{M}_m}\, \beta_\ell}. \tag{3}$$

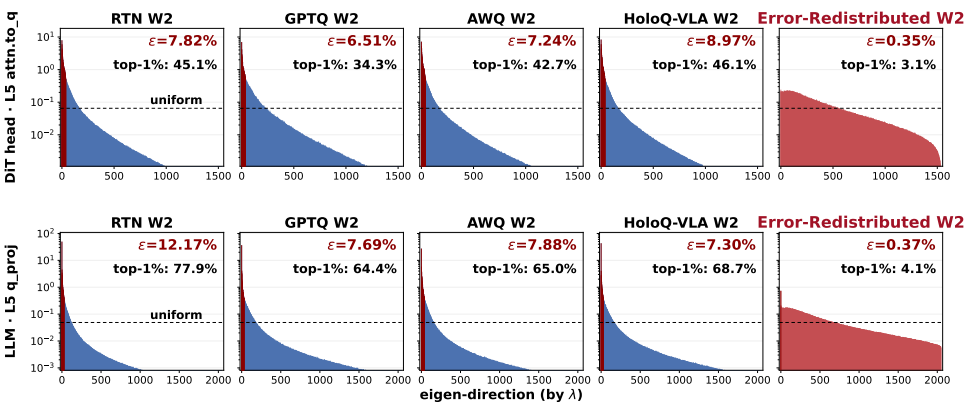


Figure 3: W2 output-error share along the eigen-directions of $H$, sorted by activation energy (log scale; dashed: uniform; dark red: above $8\times$ uniform; top-1%: $\kappa$).

In Eq. (3), $w_{\ell,i}$ is an entry of $W_\ell$, $\bar{w}_\ell$ its mean, $N_\ell$ its number of entries, and $\beta_\ell$ the leading-order max/RMS reference of a same-size Gaussian matrix. Assuming that rotation contracts the module median of max/RMS toward this reference (Assumption 2, Appendix F.2), rotation reduces the heavier-than-reference tail of the GR00T N1.5 action head but increases the lighter tail of the CogACT action head (Figure 4). Across the heads and language backbones of eight models (16 modules), $D_{\mathrm{rot}} > 1$ likewise matches tail reduction (Figure 12, Appendix E.3). $D_{\mathrm{rot}}$ is thus computed from the original weights alone, before deployment, and predicts under Assumption 2 whether rotation reduces the tail.

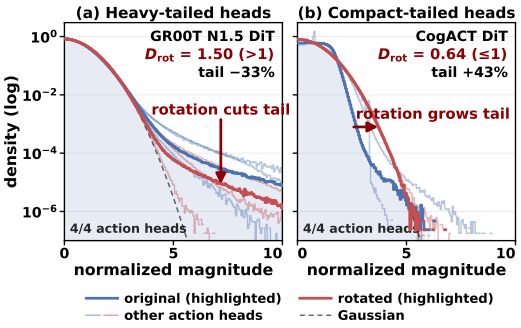


Figure 4: Action-head weight magnitudes before and after block rotation.

### 3.2 Caching under Short Denoising Trajectories

On short trajectories, the benefit of caching also depends on the decision cost. Adaptive caching incurs a decision cost $C_{\mathrm{gate}}$ at every step and a head cost $C_{\mathrm{head}}$ at each evaluated step. With $\hat{T}$ actual head evaluations, the per-call head cost and the condition for a net reduction are

$$C_{\mathrm{base}} = T\,C_{\mathrm{head}}, \quad C_{\mathrm{adapt}} = \hat{T}\,C_{\mathrm{head}} + T\,C_{\mathrm{gate}}, \quad C_{\mathrm{adapt}} < C_{\mathrm{base}} \iff \frac{C_{\mathrm{gate}}}{C_{\mathrm{head}}} < 1 - \frac{\hat{T}}{T}. \quad (4)$$

The right-hand side is the fraction of head evaluations skipped: $0.8$ when 50 evaluations are reduced to 10 but $0.5$ when 4 are reduced to 2, in which case no net reduction is obtained once the decision cost reaches half the head-evaluation cost. Decisions use the head input, so skipped steps also incur decision cost. On GR00T N1.5, the TeaCache decision alone adds 6.5–8.3% to per-call latency, and AdaptiveDiffusion is even slower than no caching ($0.98\times$; Appendix D.4). These results support fixing the evaluation steps in advance on short trajectories. The remaining choice is whether to reuse the solver state or the predicted field.

Figure 5 compares how far the predicted field $f_t$ and the solver state $x_t$ drift from the first step (median over calls, normalized by channel RMS) for SmolVLA (Shukor et al., 2025), X-VLA (Zheng et al., 2026), GR00T N1.5/N1.7 (Bjorck et al., 2025), XR-0 (Cai et al., 2026), $\pi_0$ (Black et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025), and CogACT (Li et al., 2024a). Since $x_t$ is the denoising state itself, reusing it halts the state update. The drift of $f_t$ is smaller than that of $x_t$ in six of the eight models (all but the two GR00T models), and even on GR00T N1.5 its drift after a three-step reuse interval is $0.123$, below $0.128$ for $x$. These measurements motivate *CRAFT* to keep updating the solver state at skipped steps and to reuse only the last predicted field.

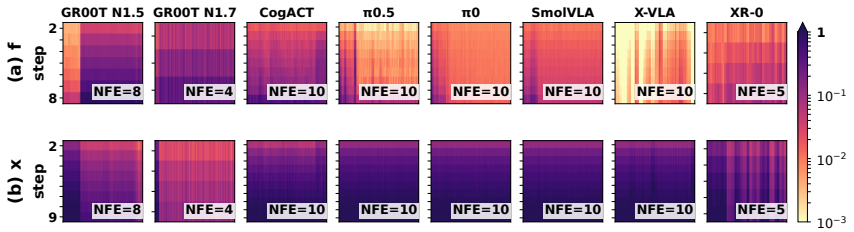


Figure 5: Per-channel drift from the first step, normalized by each channel's RMS, for eight models: (a) predicted field $f$; (b) solver state $x$. Darker means larger drift.

### 3.3 Quantization–Caching Error Decomposition

When a quantized head's predicted field is reused, the reused field carries quantization error, so the reuse error and the quantization error enter the same state update. To separate them, we write the quantized head's output on the same input as $\hat{f}_t = f_t + \delta_t$ and let $k(t)$ be the last step up to $t$ at which the head is evaluated. On the same reference trajectory, the field supplied at step $t$ deviates from the FP16 field as

$$\tilde{f}_t = \hat{f}_{k(t)} = f_{k(t)} + \delta_{k(t)}, \qquad \tilde{f}_t - f_t = \underbrace{\left[f_{k(t)} - f_t\right]}_{\text{reuse error}} + \underbrace{\delta_{k(t)}}_{\text{reused quantization error}}. \tag{5}$$

The first term is the reuse error caused by the choice of evaluation steps alone and is present even with an FP16 head; Figure 5 shows the drift of the predicted field from the first step. The second is the reused quantization error: the error $\delta_{k(t)}$ of an evaluation step carried through its reuse interval. Because both act on the same state evolution, the closed-loop performance of the combination cannot be inferred directly from the individual degradations. Proposition 4 (Appendix F.4) decomposes the squared field error into the two terms and a cross term, bounds it from above and below, and gives the factorial interaction of the field error, and we measure the closed-loop interaction across four configurations (Section 5.2).

## 4 Proposed *CRAFT*

Figure 6 summarizes *CRAFT*. Before deployment, (a) weight-tail statistics determine whether to apply rotation, and (b) the calibration-input second moment $H$ guides W2 lattice quantization. Calibration also provides step-wise A4 activation scales. During inference, (c) the head is evaluated only at fixed compute nodes; skipped steps reuse the last predicted field while the solver state is updated at every step.

### 4.1 Error-Redistributed W2A4 Quantization

**Output-error redistribution.** Scalar W2 concentrates output error on a few eigendirections with high activation energy (Section 3.1). To reduce this concentration, *CRAFT* quantizes weights jointly instead of rounding them individually, redistributing the W2 error. Its error-redistributed W2 representation assigns eight weights to one lattice codeword and uses $H$-weighted sequential block rounding that propagates rounding errors to later blocks, without fine-tuning (Tseng et al., 2024). With $2^{16}$ codewords, a 16-bit index represents eight weights, yielding 2 bits per weight. The $H$-weighted quadratic reconstruction objective $\mathrm{tr}(\Delta W H \Delta W^\top)$ has the form of the activation-aware output error in Eq. (1).

**Tail-Guided Rotation.** Following Section 3.1, *CRAFT* rotates only the action-head modules with $D_{\mathrm{rot}} > 1$:

$$r_m = \mathbf{1}\left[D_{\mathrm{rot}}(m) > 1\right]. \tag{6}$$

$D_{\mathrm{rot}}(m)$ is computed from each module's original weights before deployment, and $r_m$ remains fixed during inference. If $r_m = 1$, block rotation (Lin et al., 2024a) precedes lattice rounding and the inverse transform is applied to activations, preserving the linear map; if $r_m = 0$, no rotation is applied ($\Theta = \Theta_{\mathrm{out}} = I$).

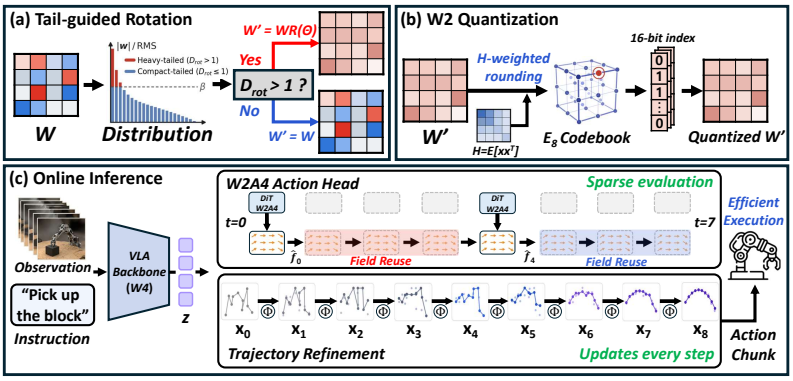


Figure 6: Overview of *CRAFT*. (a) Tail-guided rotation and (b) W2 lattice quantization prepare head weights offline. (c) During inference, the W2A4 head is evaluated at fixed compute nodes; skipped steps reuse the last predicted field while the solver state is updated at every step.

**Quantization pipeline.** Offline, each linear-layer weight matrix of the head is block-rotated according to $r_m$ into $W'$ and rounded to lattice codewords as the W2 weight $\hat{W}'$, and step-wise A4 activation scales $s_t$ computed on calibration inputs are stored in a table (Wang et al., 2026). At run time, each layer rotates its input $a$, quantizes it to A4 with the step-$t$ scale from the table, multiplies it by the W2 weight, and applies the inverse output transform, without computing activation statistics online:

$$y = \Theta_{\text{out}}^{\top}\, \hat{W}'\, Q_{\text{A4}}\big(\Theta^{\top} a;\, s_t\big). \tag{7}$$

**Asymmetric precision.** *CRAFT* uses W2A4 for the action head, evaluated at most $T$ times per call, and W4 weight-only quantization for the vision encoder and language backbone, run once. With $\hat{T}$ head evaluations, the per-call weight traffic is

$$T_{\text{weight}} = S_{\text{bb}}(b_{\text{bb}}) + \hat{T}\, S_{\text{head}}(b_{\text{head}}), \tag{8}$$

where $S_{\text{bb}}$ and $S_{\text{head}}$ are the weight bytes read at each precision; this nominal traffic follows from bit widths and evaluation counts, with $\hat{T} = T$ without caching. Quantization reduces the bytes per evaluation, and caching reduces $\hat{T}$.

### 4.2 Whole-Head Evaluation Reduction for Short Trajectories

**Static compute nodes.** Following Section 3.2, *CRAFT* fixes the evaluated steps before deployment. The compute-node set $\mathcal{C} \subseteq \{0, \ldots, T-1\}$ includes the first step, which has no field to reuse; the W2A4 head is evaluated only at $t \in \mathcal{C}$, other steps reuse $\hat{f}_{k(t)}$ of the latest node $k(t)$, and $\Phi_t$ updates the state at every step:

$$\tilde{f}_t = \begin{cases} \hat{f}_t, & t \in \mathcal{C} \\ \hat{f}_{k(t)}, & t \notin \mathcal{C} \end{cases}, \qquad k(t) = \max\{k \in \mathcal{C} : k \le t\}, \qquad x_{t+1} = \Phi_t(x_t, \tilde{f}_t). \tag{9}$$

With no run-time decision, $C_{\text{gate}} = 0$ in Eq. (4), the head is evaluated $\hat{T} = |\mathcal{C}|$ times per call, and the head traffic is $|\mathcal{C}|/T$ of the baseline. $\mathcal{C}$ is fixed per model (GR00T N1.5 $\{0, 4\}$, GR00T N1.7 $\{0, 2\}$, XR-0 $\{0\}$, $\pi_0$ and $\pi_{0.5}$ $\{0, 5\}$, X-VLA and CogACT $\{0, 4\}$, SmolVLA $\{0, 3, 7\}$), and only the last field is kept between nodes, requiring additional memory equal to the size of one action chunk. This reuse is equivalent to evaluating the head on a coarse time grid with nodes $\mathcal{C}$ and preserves the solver update rule (Appendix C.1); the scale table is indexed by the original step $t$. For GR00T N1.5 in Figure 6(c) ($T = 8$, $\mathcal{C} = \{0, 4\}$), the head is evaluated only at steps 0 and 4; steps 1–3 reuse $\hat{f}_0$ and steps 5–7 reuse $\hat{f}_4$, while the state is updated at every step, so the quantization error $\delta_0$ also enters steps 1–3 via Eq. (5). The deployment configuration $\mathcal{O} = (\mathbf{b}, \mathbf{r}, \mathcal{C})$, comprising module-wise precision $\mathbf{b}$, rotation flags $\mathbf{r}$, and compute nodes $\mathcal{C}$, is thus fixed before deployment; the nominal traffic follows Eq. (8) independently of $\mathbf{r}$, whereas the field errors of quantization and caching combine through Eq. (5).

Table 1: Closed-loop SR (%) on LIBERO-Object (SimplerEnv for CogACT). Best/second among quantized rows: **bold**/underline.

| Method | Precision (Vis. / Lang. / Head / Act.) | Head bpw (eff.) | NFE | Traffic (GiB) | SmolVLA | X-VLA | GR00T N1.5 | GR00T N1.7 | XR-0 | $\pi_{0.5}$ | $\pi_0$ | CogACT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FP16 baseline | 16 / 16 / 16 / A16 | 16 | 8 | 11.90 | 68.2 | 99.2 | 96.4 | 98.4 | 99.6 | 98.0 | 98.4 | 72.1 |
| *Original precision* | | | | | | | | | | | | |
| QuantVLA | 16 / 4 / $4^{m}$ / A8 | 8.22† | 8 | 7.68 | 15.6 | **100.0** | 98.2 | 94.6 | 99.6 | **99.8** | 0.0 | 73.3 |
| HoloQ-VLA | 16 / 4 / 4 / A4 | 5.38 | 8 | 5.85 | 15.2 | 98.8 | 98.2 | 96.6 | 99.8 | 98.8 | 85.2 | 62.1 |
| HoloQ-VLA (head) | 16 / 16 / 4 / A4 | 5.38 | 8 | 6.70 | 59.8 | 98.6 | 98.0 | 97.4 | 99.6 | 98.8 | 95.8 | 70.0 |
| W4 backbone + HoloQ head | 4 / 4 / 4 / A4 | 5.38 | 8 | 5.00 | 56.4 | 99.0 | 98.0 | **99.2** | 99.2 | 98.4 | 93.2 | 69.2 |
| *Matched W2A4 head* | | | | | | | | | | | | |
| QuantVLA-style‡ | 16 / 16 / 2 / A4 | 2.25† | 8 | 5.83 | 0.0 | 98.8 | **99.0** | 95.2 | 99.6 | 96.2 | 12.8 | 19.6 |
| HoloQ-VLA‡ (head) | 16 / 16 / 2 / A4 | 3.25 | 8 | 5.83 | 48.2 | 10.0 | 88.2 | 96.4 | **100.0** | 97.8 | 0.2 | 12.5 |
| PTQ4DiT | 16 / 16 / 2 / A4 | 2.25† | 8 | 5.83 | 0.0 | 0.0 | 80.0 | 56.0 | 99.4 | 23.2 | 0.0 | 42.9 |
| TFMQ-DM | 16 / 16 / 2 / A4 | 2.25† | 8 | 5.83 | 0.0 | 0.0 | 0.0 | 0.0 | 98.2 | 42.8 | 0.0 | 0.0 |
| *Caching and joint quantization–caching* | | | | | | | | | | | | |
| TeaCache | 16 / 16 / 16 / A16 | 16 | 3 | 6.69 | 69.0 | 99.2 | 96.6 | 98.0 | 99.6 | 96.8 | 96.0 | 62.1 |
| MagCache | 16 / 16 / 16 / A16 | 16 | 3 | 6.69 | 69.8 | 99.6 | 97.2 | 98.0 | 99.4 | 98.4 | 97.0 | 70.8 |
| CacheQuant | 16 / 16 / 8 / A8 | 8.00† | 6 | 7.28 | 65.4 | 99.2 | 96.0 | 97.6 | 99.2 | 99.0 | **97.0** | **73.8** |
| *CRAFT (ours)* | | | | | | | | | | | | |
| *CRAFT* w/o quant. | 16 / 16 / 16 / A16 | 16 | 2 | 5.64 | 68.2 | 99.8 | 97.6 | 98.0 | 99.6 | 98.8 | 97.8 | 62.9 |
| *CRAFT* w/o backbone, w/o cache | 16 / 16 / 2 / A4 | 3.14 | 8 | 5.83 | 65.0 | 99.8 | 98.6 | 97.2 | 99.8 | 98.0 | 19.4 | 65.8 |
| *CRAFT* w/o backbone | 16 / 16 / 2 / A4 | 3.14 | 2 | 4.13 | 62.8 | 99.2 | **99.0** | 98.8 | 99.2 | 97.8 | 16.0 | 36.7 |
| *CRAFT* w/o cache | 4 / 4 / 2 / A4 | 3.14 | 8 | 4.13 | 61.0 | 99.6 | 98.4 | 98.0 | **100.0** | 99.2 | 27.8 | 64.2 |
| *CRAFT* | 4 / 4 / 2 / A4 | 3.14 | 2 | 2.43 | 60.4 | 99.8 | 98.4 | 98.4 | 99.8 | 97.8 | 24.0 | 37.5 |
| *CRAFT* w/o cache (W4 head) | 4 / 4 / 4 / A4 | 5.14 | 8 | 5.00 | **67.4** | 99.8 | 98.4 | 98.6 | 99.6 | 99.0 | 83.6 | 65.4 |
| *CRAFT* (W4 head) | 4 / 4 / 4 / A4 | 5.14 | 2 | 2.64 | 64.8 | 99.4 | 98.4 | 97.0 | 99.4 | 99.0 | 69.8 | 62.9 |

‡: matched W2A4 head; $^{m}$: DiT MLP only; †: nominal bpw (others measured on the deployment container).

# 5 Experiments and Analysis

## 5.1 Experimental Setup

We evaluate eight VLAs with diffusion or flow-matching heads: GR00T N1.5/N1.7 (Bjorck et al., 2025), CogACT (Li et al., 2024a), $\pi_0$ (Black et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025), SmolVLA (Shukor et al., 2025), X-VLA (Zheng et al., 2026), and XR-0 (Cai et al., 2026). CogACT uses SimplerEnv (Li et al., 2024b) ($n = 240$) and the others LIBERO-Object (Liu et al., 2023) ($n = 500$); Appendix D reports additional suites and replanning intervals. Prior methods are reported with a W2A4 head (‡), QuantVLA and HoloQ-VLA also at original precision; definitions and controls are in Appendix B. *CRAFT* uses weight-only backbone quantization and applies rotation by Eq. (6). Accuracy is measured on H100 with episodes paired by official evaluation seeds. Latency at batch $= 1$ is the minimum of 36 CUDA-event measurements of a CUDA-graph-captured policy call after 50 warm-ups on RTX 3090/4090 and the minimum wall-clock time on MacBook Pro (M5)/Air (M4). Appendices A and C.1 give implementation and kernel details.

## 5.2 Main Results

Table 1 presents comparisons of the full configuration alongside controls that isolate head quantization, backbone quantization, and caching. The W4-head variants assess precision sensitivity, while matched W2A4 comparisons use an FP16 backbone without caching. Head bpw, NFE, and traffic are reported for GR00T N1.5. By suppressing output error through error-redistributed W2 and tail-guided rotation, the head-only *CRAFT* is at most $1.2\,pp$ below FP16 on five models and outperforms every prior method under the same setting on the other three (***Lim. 1***). TFMQ-DM at the same W2A4 reaches $0.0$ on six models, and QuantVLA and HoloQ-VLA (head) at original precision incur more traffic. Without quantization, *CRAFT* reuses the predicted field while updating the state and, without per-step decisions, reaches NFE 2 and $5.64$ GiB, below TeaCache and MagCache (NFE 3, $6.69$ GiB) while maintaining its SR within $1.6\,pp$ of MagCache on all models but CogACT (***Lim. 2***). By cutting both bytes per evaluation and evaluations, the full configuration lowers the traffic of GR00T N1.5 from $11.90$ to $2.43$ GiB, with SR at or above FP16 on GR00T N1.5/N1.7, XR-0, and X-VLA and $0.2\,pp$ and $7.8\,pp$ below it on $\pi_{0.5}$ and SmolVLA. Decomposing the SR change of the four FP16-backbone configurations into two main effects and the interaction $I_m$ (Appendix F.4) (***Lim. 3***) gives $I_m$ in $-2.8$–$+2.0\,pp$ on all models but CogACT ($-19.9\,pp$).

Table 2: Head components on GR00T N1.5. AWQ, RTN: scalar W2.

| Head config. | Spa. | Obj. | Goal | Long | Avg. |
|---|---|---|---|---|---|
| AWQ W2, A16 | 37.6 | 93.0 | 74.6 | 59.4 | 66.15 |
| RTN W2, A16 | 37.4 | 91.0 | 81.8 | 47.2 | 64.35 |
| ⌞ + rot. | 81.2 | 96.2 | 91.2 | 87.4 | 89.00 |
| Redistributed W2, A16 | 45.4 | 95.4 | 86.8 | 70.4 | 74.50 |
| ⌞ + tail-guided rot. | 88.2 | 98.6 | 95.0 | 87.0 | 92.20 |
| ⌞ + A4 (*CRAFT*) | 95.8 | 98.6 | 97.6 | 91.6 | **95.90** |

Table 3: Precision ladder on GR00T N1.5 (backbone: GPTQ g128).

| Precision | $\varepsilon$ (%) | Spatial | Object | Goal | Long | Avg. ↑ |
|---|---|---|---|---|---|---|
| *Backbone (head W2A4, with caching)* | | | | | | |
| W4 (*CRAFT*) | 0.633 | 98.6 | 98.4 | 97.2 | 90.0 | **96.05** |
| W3 | 1.190 | 94.6 | 96.4 | 95.0 | 90.8 | 94.20 |
| W2 | 3.941 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 |
| *Action head (FP16 backbone, no caching)* | | | | | | |
| W4A4 | 0.024 | 94.0 | 98.0 | 97.8 | 92.0 | 95.45 |
| W3A4 | 0.056 | 95.6 | 96.4 | 97.2 | 92.0 | 95.30 |
| W2A4 (*CRAFT*) | 0.208 | 95.8 | 98.6 | 97.6 | 91.6 | **95.90** |

Table 4: Rounding objectives on GR00T N1.5 (W2A4 head; SR in %, $\varepsilon$ in %).

| Objective | Spa. | Obj. | Goal | Long | Avg. | $\varepsilon$ |
|---|---|---|---|---|---|---|
| Random-basis $H$ (control) | 91.2 | 97.2 | 94.6 | 87.2 | 92.55 | 3.954 |
| $\|\Delta W\|_F^2$ | 95.4 | **98.6** | 96.8 | 91.8 | 95.65 | 1.725 |
| $\mathrm{diag}(H)$ | **96.2** | 98.0 | **97.6** | 92.4 | **96.05** | 1.000 |
| Block $H$ (32) | 94.0 | 98.0 | 96.4 | **93.0** | 95.35 | 0.719 |
| Block $H$ (128) | 95.8 | 98.0 | 96.2 | 92.4 | 95.60 | 0.550 |
| $H$ (*CRAFT*) | 95.8 | **98.6** | **97.6** | 91.6 | 95.90 | **0.208** |

Table 5: Rotation policies for the W2A4 head.

| Model | $D_{\mathrm{rot}}$ | $r_m$ | None | Always | Rule |
|---|---|---|---|---|---|
| X-VLA | 1.67 | 1 | 94.6 | 99.8 | 99.8 |
| GR00T N1.5 | 1.50 | 1 | 99.0 | 98.6 | 98.6 |
| GR00T N1.7 | 1.37 | 1 | 97.6 | 97.2 | 97.2 |
| $\pi_{0.5}$ | 1.21 | 1 | 98.8 | 98.0 | 98.0 |
| XR-0 | 0.92 | 0 | 99.8 | 99.6 | 99.8 |
| $\pi_0$ | 0.84 | 0 | 19.4 | 22.4 | 19.4 |
| CogACT | 0.64 | 0 | 65.8 | 55.4 | 65.8 |
| SmolVLA | 0.62 | 0 | 65.0 | 59.2 | 65.0 |
| Avg. | — | — | 80.00 | 78.78 | **80.45** |

### 5.3 Ablation Studies

**Complementary Quantization Components.** In Table 2, error-redistributed W2 without rotation outperforms AWQ on all four suites, and scalar RTN has lower SR than redistributed W2 on three suites even with the same rotation. Rotation raises the mean of RTN from $64.35$ to $89.00$ and of redistributed W2 from $74.50$ to $92.20$, and A4 further raises the latter to $95.90$. The rotation gain is largest on Spatial (RTN $37.4 \rightarrow 81.2$, redistributed W2 $45.4 \rightarrow 88.2$).

**Precision Allocation.** With an FP16 backbone, lowering the head from W4 through W3 to W2 raises $\varepsilon$ from $0.024$ to $0.208$, with mean SRs of $95.45$, $95.30$, and $95.90$ (Table 3); $\varepsilon$ compares errors across configurations rather than predicting SR, which saturates at the FP16 level here. Since the backbone is read once per call and the head NFE times, the same bit reduction per weight saves more traffic in the head (Eq. 8). In the deployment configuration of Table 1, a W4 head raises the mean SR by $+9.3\,pp$, concentrated on $\pi_0$ ($+45.8\,pp$) and CogACT ($+25.4\,pp$) while the other six average $+0.6\,pp$; *CRAFT* uses W2 as its default head precision. W3 and W2 backbones give mean SRs of $94.20$ and $0.00$, so the backbone is fixed at W4.

**Output versus Weight Error.** Across the 40 configurations of the four W2A4 rows in Table 1 and *CRAFT* head-only with rotation on every head (Always, Table 5), the relative output error $\varepsilon$ of the head has a Spearman correlation of $-0.52$ with $\Delta\mathrm{SR} = \mathrm{SR} - \mathrm{SR}_{\mathrm{FP16}}$, whereas the weight reconstruction error has the opposite sign ($+0.37$); within-model ranks over 68 configurations (28 more W2 variants) show the same trend (Appendix E.1). When only the rounding objective varies (Table 4), the random-basis control has the largest $\varepsilon$ ($3.954$) and the lowest mean SR ($92.55$), below even FP16 ($94.00$). The other five objectives, spanning $\varepsilon = 0.208$–$1.725$, all reach or exceed FP16, so SR saturates and their differences are marginal ($0.70\,pp$). In this regime $\varepsilon$ distinguishes the objectives; we deploy full $H$, which requires no additional hyperparameters and yields the lowest $\varepsilon$.

**Tail-Guided Rotation.** The shaded column in Table 5 corresponds to Rule; the eight-head mean SRs are $80.45$ (Rule), $80.00$ (None), and $78.78$ (Always). On heads with $D_{\mathrm{rot}} \leq 1$, omitting rotation improves SR by $10.4\,pp$ on CogACT and $5.8\,pp$ on SmolVLA, whereas on X-VLA ($D_{\mathrm{rot}} > 1$) it reduces SR by $5.2\,pp$. GR00T N1.5, N1.7, and $\pi_{0.5}$ already reach 97–99%, so rotation changes their SR by only $0.4$–$0.8\,pp$. Thus the rule of Eq. (6), based on the tail analysis, identifies heads with $D_{\mathrm{rot}} \leq 1$ before evaluation and omits their rotation, attaining a higher mean SR than Always.

Table 6: End-to-end speedup over the FP16 baseline on the same device (MacBook: BF16).

| Method | MacBook Pro (M5) | | | MacBook Air (M4) | | | RTX 3090 | | | RTX 4090 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GR00T N1.5 | CogACT | $\pi_{0.5}$ | GR00T N1.5 | CogACT | $\pi_{0.5}$ | GR00T N1.5 | CogACT | $\pi_{0.5}$ | GR00T N1.5 | CogACT | $\pi_{0.5}$ |
| FP16 baseline | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| *Quantization* | | | | | | | | | | | | |
| QuantVLA | 0.81 | 0.90 | 0.85 | 0.83 | 0.67 | 0.78 | 0.99 | 1.04 | 0.94 | 0.99 | 0.85 | 0.93 |
| HoloQ-VLA | 0.65 | 0.67 | 0.61 | 0.61 | 0.57 | 0.85 | 1.21 | 2.45 | 1.38 | 0.94 | 1.67 | 1.00 |
| PTQ4DiT/TFMQ-DM | 1.00 | 1.01 | 1.01 | 0.93 | **1.05** | 0.85 | 1.13 | 1.01 | 1.01 | 1.03 | 1.02 | 0.98 |
| *Caching and joint quantization–caching* | | | | | | | | | | | | |
| TeaCache | 1.22 | 1.06 | 1.18 | 1.19 | 1.00 | 1.07 | 1.19 | 1.04 | 1.23 | 1.16 | 1.04 | 1.31 |
| MagCache | 1.32 | 1.06 | 1.19 | 1.30 | 1.00 | 1.11 | 1.38 | 1.07 | 1.25 | 1.40 | 1.09 | 1.32 |
| CacheQuant | 1.00 | 0.99 | 1.05 | 0.95 | 0.77 | 0.90 | 1.40 | 1.05 | 1.26 | 1.30 | 1.09 | 1.24 |
| *CRAFT (ours)* | | | | | | | | | | | | |
| *CRAFT* w/o backbone, w/o cache | 0.89 | 1.00 | 0.98 | 0.90 | 1.00 | 0.87 | 1.08 | 1.02 | 0.98 | 0.89 | 1.05 | 0.87 |
| *CRAFT* w/o head, w/o cache | 1.00 | 1.00 | 0.99 | 0.93 | 0.73 | 1.00 | 1.20 | 2.65 | 1.50 | 1.13 | 1.90 | 1.36 |
| *CRAFT* w/o cache | 0.89 | 1.00 | 0.98 | 0.93 | 1.00 | 0.87 | 1.31 | 2.86 | 1.44 | 1.00 | 2.11 | 1.13 |
| *CRAFT* w/o quant. | **1.48** | 1.05 | 1.23 | 1.35 | 0.80 | **1.13** | 1.50 | 1.10 | 1.37 | 1.50 | 1.12 | 1.48 |
| *CRAFT* | 1.41 | **1.08** | **1.25** | **1.40** | 1.03 | 1.02 | **2.06** | **3.20** | **2.46** | **1.75** | **2.48** | **2.27** |
| FP16 (Hz) | 2.5 | 1.6 | 1.1 | 0.6 | 0.4 | 0.4 | 11.1 | 6.4 | 6.3 | 20.6 | 16.4 | 11.3 |
| *CRAFT* (Hz) | **3.6** | **1.7** | **1.4** | **0.9** | **0.4** | **0.4** | **22.8** | **20.5** | **15.5** | **36.0** | **40.7** | **25.7** |

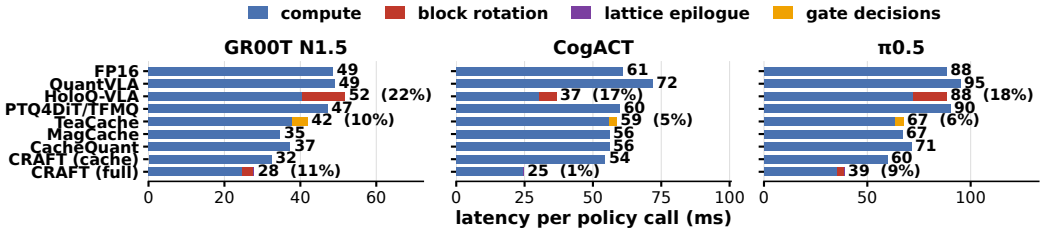


Figure 7: Policy-call time breakdown on RTX 4090. QuantVLA, PTQ4DiT/TFMQ-DM, and CacheQuant report no component split.

### 5.4 Efficiency Results and Analysis

**End-to-End Speedup.** The *CRAFT* row in Table 6 corresponds to the full configuration in Table 1 (backbone: Algorithm 4). Speedups are measured relative to the baseline of the same model on the same device (RTX: FP16; MacBook: BF16), without quantization or caching. *CRAFT* cuts GR00T N1.5 traffic by 79.6% and the measured head storage from 16 to 3.14 bpw, and is faster than all prior methods for GR00T N1.5 on all four measured devices: 2.06× (RTX 3090), 1.75× (RTX 4090), 1.41× (MacBook Pro), and 1.40× (MacBook Air) (Table 6). On both GPUs and all three models, it achieves greater speedup than caching alone (1.10–1.50×) and quantization alone (1.00–2.86×), demonstrating the benefit of combining quantization and caching.

**Overhead Analysis.** In the policy-call time breakdown of Figure 7, TeaCache decisions take 2.77–4.07 ms and the rotation of HoloQ-VLA 6.3–16.3 ms, whereas the block rotation of *CRAFT*, run only at compute nodes, takes 3.2–3.3 ms, its lattice epilogue at most 0.18 ms, and its static schedule needs no runtime decisions. In 14 of 16 cases, the adaptive gates also select a single step set, with NFE no lower than the static schedule (Appendix D.4). Rotated head-only quantization pays this cost at every evaluation (0.87–1.08×), but caching reduces it with the evaluation count.

## 6 Conclusion

W2 action-head SR is more consistently rank-correlated with output error along a few high-activation-energy directions than with weight reconstruction error. In the evaluated modules, block rotation reduced extremes in heavy-tailed heads and increased them in light-tailed ones, and on short trajectories decision cost can offset the caching gain. Combining W2A4 lattice quantization, tail-guided rotation, and predicted-field reuse at fixed compute nodes with a W4 backbone, *CRAFT* maintains FP16-level SR on GR00T N1.5 and $\pi_{0.5}$, reduces GR00T N1.5 traffic by 79.6%, and accelerates these two models by 2.06× and 2.46× on RTX 3090.

### AI USE STATEMENT

We used AI tools to improve the wording of the manuscript and to assist with implementing the authors' research ideas in code and running experiments. The authors reviewed all technical content, experimental results, and the final manuscript.

### REPRODUCIBILITY STATEMENT

The evaluated models, benchmarks, $n$ per configuration, replanning intervals, and the hardware and procedures for measuring accuracy and latency are described in Section 5.1; the offline preparation (block rotation, lattice codeword rounding, and step-wise A4 activation scale tables) and the execution order in Section 4.1 and Algorithms 1 and 2 of Appendix C.1; and the compute-node sets and predicted-field reuse rule in Section 4.2. The evaluated configurations, $n$, and seed convention were fixed before execution (Appendix A), and the full evaluation grid with the reasons for infeasible combinations is given in Appendix D.1. Seeds follow the convention of each model's official evaluation code, and their values are recorded in the pre-execution records.

### ETHICS STATEMENT

We evaluate eight VLA checkpoints on the LIBERO and SimplerEnv simulation benchmarks and use no data involving human subjects or personal information. Deploying compressed policies on physical systems requires separate safety validation; all results in this paper were measured in simulation (Appendix G).

## REFERENCES

Arash Akbari et al. Actquant: Sub-4-bit action-guided quantization for vision-language-action models. *arXiv preprint arXiv:2605.24011*, 2026.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, June 2025. doi: 10.15607/RSS.2025.XXI.010.

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

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, et al. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026.

Hengyu Fang, Yijiang Liu, Yuan Du, Li Du, and Huanrui Yang. Sqap-vla: A synergistic quantization-aware pruning framework for high-performance vision-language-action models. *arXiv preprint arXiv:2509.09090*, 2025.

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

Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7362–7371, June 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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

Jaeyong Lee, Hyeunjoo Kim, Sanghun Oh, Myoungjun Chun, Myungsuk Kim, and Jihong Kim. Aif: Accelerating on-device llm inference using in-flash processing. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, pp. 529–543, 2025.

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

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024b.

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. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024a.

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. *Proceedings of machine learning and systems*, 6: 87–100, 2024b.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363. IEEE, 2025a.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15853–15863, October 2025b.

Xuewen Liu, Zhikai Li, and Qingyi Gu. Cachequant: Comprehensively accelerated diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23269–23280. IEEE, 2025c.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15762–15772, June 2024.

Zehong Ma, Longhui Wei, Feng Wang, Shiliang Zhang, and Qi Tian. Magcache: Fast video generation with magnitude-aware cache. *Advances in Neural Information Processing Systems*, 38: 34348–34380, 2026.

Navin Ranjan and Andreas Savakis. Mix-qvla: Task-evidence-aware mixed-precision quantization of vision-language-action models. *arXiv preprint arXiv:2606.19565*, 2026.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP$\#$: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=9BrydUVcoe.

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

Xinyu Wang, Mingze Li, Sicheng Lyu, Dongxiu Liu, Kaicheng Yang, Ziyu Zhao, Yufei Cui, Xiao-Wen Chang, and Peng Lu. Holoq-vla: Uniform w4a4 quantization of vision-language-action models. *arXiv preprint arXiv:2605.28803*, 2026.

Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. *Advances in neural information processing systems*, 37: 62732–62755, 2024.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=TpL2nXanru.

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

Hancheng Ye, Jiakang Yuan, Renqiu Xia, Xiangchao Yan, Tao Chen, Junchi Yan, Botian Shi, and Bo Zhang. Training-free adaptive diffusion with bounded difference approximation strategy. *Advances in Neural Information Processing Systems*, 37:306–332, 2024.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39539–39549, June 2026.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

# A Implementation Details

**Models.** We select eight VLAs with repeated action-head execution across denoising steps, spanning head families (seven flow-matching models and one diffusion-based DiT model), sizes and evaluation frequencies (48–192 quantized linear layers and 4–10 head evaluations per call), and deployment frequencies (replanning intervals of 1–50). Parentheses give (head linear-layer count, $R$, head evaluations per call).

- **GR00T N1.5** (96, $R$=8, 8) and **GR00T N1.7** (192, $R$=8, 4): Both combine a vision–language backbone with a flow-matching DiT action head and share backbone conventions, but differ twofold in head depth and head evaluations per call; we use N1.5 for method development. For GR00T N1.5, we use 8 denoising steps following the official evaluation scripts of QuantVLA and HoloQ-VLA.
- **CogACT** (48, $R$=1, 10): The only evaluated policy with a diffusion-based rather than flow-matching DiT head ($\eta = 0$ DDIM); its head accounts for only $1\%$ of storage and $11\%$ of weight traffic, limiting the attainable system-level gain from head-only compression. We use SimplerEnv since no LIBERO checkpoint is available; $R$=1 is fixed since only the first action of each 16-step chunk is executed.
- $\pi_0$ and $\pi_{0.5}$ (126, $R$=5, 10): Both place a flow-matching action expert alongside the backbone ($\pi_{0.5}$ is the successor); frequent replanning makes per-call cost directly visible in the control cycle.
- **SmolVLA** (112, $R$=50, 10): For rotation experiments, we use block width 16 because its action expert input dimension is not divisible by 64. Its head accounts for $87\%$ of latency, the largest share. Its FP16 success rate of $68.2$ is the lowest among the eight models.
- **X-VLA** (98 = block 96 + conditioning projection 2, $R$=30, 10): Its flow-matching head uses a noise remixing update and accounts for $84\%$ of weight traffic, the largest share among the eight models.
- **XR-0** (80, $R$=10, 5): With only 5 head evaluations per policy call, XR-0 allows us to evaluate caching when few head evaluations are available to skip.

**Benchmarks.** We measure accuracy on four closed-loop benchmarks selected for their coverage across models, manipulation types, and perturbation conditions.

- **LIBERO-Object** (10 tasks $\times$ 50 initial states, $n = 500$): The common benchmark for seven models and reference for main-text Table 1. Only the target object varies, isolating object identification and pickup.
- **SimplerEnv** (`gr_va_min` 4 tasks $\times$ 60 episodes, $n = 240$): We evaluate CogACT, which lacks a LIBERO checkpoint, in this simulator approximating a real robot's observation distribution. Since the environment differs, the CogACT column supports only within-model configuration comparisons.
- **LIBERO-Spatial, -Goal, -Long** ($n = 500$ each): These evaluate spatial-arrangement, goal-specification, and long-sequence difficulty. LIBERO-Long contains sequential subgoals and evaluates field reuse over longer task horizons (results appear in Appendix D.1).
- **LIBERO-Plus** ($n = 400$ each): Evaluates compressed-configuration robustness under camera position, lighting, language instruction, table, and background perturbations.

**Evaluation protocol.** Except for the original-precision QuantVLA row (DiT MLP only, $^{\mathrm{m}}$ in Table 1), all configurations that quantize the action head cover attention and feed-forward linear layers in action head transformer blocks (including X-VLA conditioning projections), keeping adaLN, the timestep embedder, the action encoder/decoder, and the output projection in FP16. We report success rates only when the quantized-layer replacement count matches this count. Methods for each model share the episode-pairing seed convention; configurations are compared using the $95\%$ Newcombe interval for a difference of proportions ($\pm 0.9$–$6.0\,pp$ at $n = 500$). Replanning intervals and compute-node sets are fixed per model before measurement; methods share the checkpoint, evaluation episode range, and calibration conditions (Appendix D.1). For LIBERO, $n = 500$ uses all 50 initial states per task. Each lattice codebook in Table 3 uses its original repository's scale.

**Latency measurement.** On RTX 3090/4090, we capture each entire policy call in a CUDA graph and report minimum CUDA event time (also median) over 36 measurements per combination across 6 rounds with shuffled configuration order, after 50 warm-up calls. Speedup is relative to FP16 under identical capture conditions. In Table 6's RTX columns, PTQ4DiT/TFMQ-DM run their W4A8 head with INT8 GEMM. The RTX 3090 clock is fixed at 1695 MHz; we verify no other compute processes before and after measurement on either device. H100 energy in Appendix D.5 uses NVML cumulative measurement over 24 rounds $\times$ 40 calls; batch experiments report p50 over 30 rounds without CUDA graphs. We exclude runs with repeated measurements differing by more than $1\,\%$ in latency or $2\,\%$ in energy. The MacBook Pro (Mac17,2; Apple M5) and MacBook Air (Mac16,12; Apple M4) have a 10-core CPU/GPU and 24 GB unified memory. On PyTorch MPS, we report minimum wall-clock time over 40 repetitions after 30 warm-up calls (also p50 and p90), relative to BF16 in the same session. MacBook and RTX columns should not be compared because some low-bit head operations use reference paths. Software versions are PyTorch 2.14.0, 2.13.0, and 2.11.0 (GR00T N1.5, CogACT, and $\pi_{0.5}$) and MLX 0.32.2.

## B Prior Methods

**Comparison protocol.** We control four factors to match evaluation conditions across methods.

(i) We measure all methods using the same checkpoint, benchmark, evaluation episode range, and replanning interval.
(ii) QuantVLA and HoloQ-VLA are reported at original precision and with a W2A4 head (‡), and PTQ4DiT and TFMQ-DM with a W2A4 head, giving matched-precision and original-setting comparisons.
(iii) Prior methods and our method allow model-specific hyperparameters. TeaCache uses threshold $0.05$ for SmolVLA and X-VLA and $0.25$ otherwise; MagCache sets $\delta$ and $K$ from a table captured from an FP run; ***CRAFT*** sets $r_m$ using only pretrained weights.
(iv) Weight traffic in Table 1 is computed from nominal bit widths and evaluation counts and is therefore independent of how rotation is stored.

Each method retains its original paper's weight and activation quantization schemes. With public code and weights, we keep algorithms unchanged, adapting only the VLA-head interface (decision inputs, cache targets) as below.

**QuantVLA.** This method quantizes all LLM linear layers and DiT MLPs to W4 and activations to A8, retains FP DiT attention Q/K/V/O, and uses DuQuant rotation and ATM and OHB scales. The ‡ variant uses a W2A4 head.

**HoloQ-VLA.** This method quantizes the LLM and DiT head to W4A4 with SVD–Hadamard rotation and a step-wise DiT activation table, retaining FP ViT and VL self-attention. The head row quantizes only the head (FP16 backbone); W4 backbone + HoloQ head adds ***CRAFT***'s W4 backbone.

**PTQ4DiT.** This method reports DiT W8A8 and W4A8 with Channel-wise Salience Balancing (CSB), Spearman's $\rho$-guided Salience Calibration, and AdaRound/BRECQ reconstruction. For X-VLA, we observe that CSB folding increases the row-wise dynamic range of the wide `fc1` layers ($1024 \rightarrow 4096$) from $5.2$ to $17.7$. Under the original PTQ4DiT configuration, W4A8 and W2A4 yield $0.0$, whereas a diagnostic configuration without CSB achieves $99.4$ at W4A8. Table 1 reports the W2A4-head configuration.

**TFMQ-DM.** This method isolates the timestep-only path in a Temporal Information Block for independent reconstruction (TIAR), using finite-set calibration to match step-wise activation distributions. It targets temporal-feature perturbation from W4 weight quantization in image diffusion; Table 1 reports the W2A4-head configuration.

**TeaCache and MagCache.** Both methods reuse the entire head output (predicted field) at skipped steps. TeaCache rescales and thresholds the relative L1 change of the first block's modulated input (head input $x_t$ without AdaLN) using a model-specific fourth-degree polynomial. MagCache uses a table of residual magnitude ratios captured from an FP run. Table 1 uses NFE 3 for GR00T N1.5; Appendix D.4 gives each model's actual NFE.

**CacheQuant.** This method selects compute steps with an offline DPS schedule and uses DEC to correct skipped-step outputs through a channel-wise affine transformation. The cache stores $\Delta$ from the middle DiT block at W8A8 precision. The only prior method in the main-text comparison combining quantization and caching, it appears in Table 1 with a W8A8 head and NFE 6 (GR00T N1.5).

**AdaptiveDiffusion.** This method selects update steps at run time using third-order finite differences. We use it only to measure decision overhead (Section 3.2), not in success-rate tables.

**Bit accounting and traffic.** Table 1's bpw, NFE, and weight traffic are for GR00T N1.5. Measured bpw divides serialized head-file bits by the head block linear-layer parameter count ($465{,}567{,}744$), including rotation, permutations, and step-wise activation tables alongside codes and scales. † marks nominal calculations for rows without serialized files, excluding rotation and activation tables. Weight traffic estimates weight bytes read per policy call from nominal bit widths: backbone linear layers $\sum_m P_m b_m/8$ and their FP16 remainder are counted once; head block linear layers $P_{\text{head}} b_{\text{head}}/8$ and their FP16 remainder, NFE times. Activation bit widths and quantizer types do not affect this estimate; for CacheQuant, skipped steps still read the eight recomputed blocks with their adaLN and the head's input/output layers (7.28 GiB).

# C Algorithm and Kernel Implementation

## C.1 Deployment Procedure

We obtain $H$ by normalizing the calibration-input second moment by its diagonal mean and adding a small damping term; rotation block width is 64 (16 only for the SmolVLA action expert). For Euler updates, the update between consecutive compute nodes $k$ and $k'$ is $x_{k'} = x_k + (\sum_{t=k}^{k'-1} \Delta_t)\hat{f}_k$, so Eq. (9) is equivalent to evaluation on a coarse time grid. We verify numerically that the final open-loop actions agree within floating-point precision for $\eta = 0$ DDIM (Song et al., 2021) and X-VLA's noise remixing. Algorithm 1 generates a deployment pack offline once per model; Algorithm 2 executes per call. Both fix the rotation flags and compute-node sets before the reported evaluations.

**Algorithm 1** *CRAFT* offline pack construction (once per model)

**Require:** linears $\{W_\ell\}$ of head $m$, backbone $W_{\text{bb}}$, calibration set $\mathcal{D}$, trajectory length $T$, block width $w$
**Ensure:** pack $\big(\{\hat{W}'_\ell, \Theta_\ell, \Theta_{\text{out},\ell}\}, \{s_t\}, \mathcal{C}, \hat{W}_{\text{bb}}\big)$

1: $D_{\text{rot}}(m) \leftarrow \text{median}_{\ell\in\mathcal{M}_m} \rho_\ell / \text{median}_{\ell\in\mathcal{M}_m} \beta_\ell$; $r_m \leftarrow \mathbf{1}[D_{\text{rot}}(m) > 1]$ ▷ Eq. (3), (6); one decision per head, weights only
2: **for** each head linear $\ell$ **do**
3: $(\Theta_\ell, \Theta_{\text{out},\ell}) \leftarrow$ width-$w$ block-diagonal orthogonal transforms if $r_m = 1$, else $(I, I)$
4: $W'_\ell \leftarrow \Theta_{\text{out},\ell} W_\ell \Theta_\ell$; $H_\ell \leftarrow$ damped input second moment over $\mathcal{D}$
5: $\hat{W}'_\ell \leftarrow$ block-sequential lattice rounding of $W'_\ell$ minimizing $\text{tr}(\Delta W H_\ell \Delta W^\top)$, carrying each block's residual forward ▷ 2 bit/weight, no fine-tuning
6: **end for**
7: $s_t \leftarrow$ A4 activation scales at step $t$ over $\mathcal{D}$, for $t = 0, \ldots, T-1$ ▷ read by table index at run time
8: $\hat{W}_{\text{bb}} \leftarrow \text{W4}(W_{\text{bb}})$, weight-only; $\mathcal{C} \subseteq \{0, \ldots, T-1\}$ with $0 \in \mathcal{C}$ ▷ fixed before evaluation

**Algorithm 2** *CRAFT* execution of one policy call

**Require:** observation $o$, pack, trajectory length $T$, solver updates $\{\Phi_t\}$
**Ensure:** action chunk
$z \leftarrow \textsc{Backbone}_{\mathrm{W4}}(o)$; initialize $x_0$; $k \leftarrow 0$ ▷ backbone runs once per call
**for** $t = 0$ **to** $T-1$ **do**
  **if** $t \in \mathcal{C}$ **then**
    $\tilde{f}_t \leftarrow \textsc{HeadForward}_{\mathrm{W2A4}}(x_t, z, t)$; $k \leftarrow t$ ▷ each linear as in Eq. (7)
  **else**
    $\tilde{f}_t \leftarrow \tilde{f}_k$ ▷ head skipped; no run-time decision, $C_{\mathrm{gate}} = 0$
  **end if**
  $x_{t+1} \leftarrow \Phi_t(x_t, \tilde{f}_t)$ ▷ Eq. (9); state updated every step
**end for**
**return** action read out from $x_T$ ▷ head evaluations per call $= |\mathcal{C}|$

### C.2 Low-Bit Kernels

RTX latency uses integer operands without reconstructing low-bit weights in FP16. The kernels are a CUTLASS 3.5.1-based PyTorch C++/CUDA extension (INT4 GEMM and auxiliary kernels) and INT8 GEMM for A8 prior methods; E8P12 and block rotation come from QuIP# and DuQuant. Head codewords expand losslessly into INT4 codes at load time, so the kernel reads a 4 bit operand per element and the main text's 2-bit weight traffic is nominal. The FP32 epilogue avoids FP16 integer precision loss at $|A| > 2048$, and GEMM is captured on the current stream and graph pool.

**Launch structure.** Within a CUDA graph, each quantized linear layer launches three kernels (input-side quantization, integer GEMM, and epilogue; Algorithms 3, 4), whereas FP16 and Marlin layers launch one. The first kernel applies input permutation and width-64 block rotation only for $r_m = 1$, then generates A4 codes; it receives the step-wise activation scales as a channel vector $u$ and a step scalar $c_t$, reading one $c_t$ at run time. Codewords expand once at load time into integer codes $\mu$ with $4v = 2\mu - 1$, so no run-time codebook lookup is needed, and an all-ones weight row computes $\sum_k \hat{q}_{ik}$ within GEMM for the epilogue's affine correction. Backbone layers (Algorithm 4) share this structure without rotation or codeword expansion.

**Algorithm 3** W2A4 lattice head linear at a compute step (RTX kernel path, three launches)

**Require:** Input rows $X \in \mathbb{R}^{M\times I}$; INT4 codes $\mu \in [-5,6]^{O\times I}$ with an appended all-ones row; output scales $\sigma \in \mathbb{R}^{O}$; input vector $u \in \mathbb{R}^{I}$; step scalar $c_t$; bias $b$; rotation flag $r_m$ with permutation $\pi$ and width-$w$ blocks $\{\Theta_j\}, \{\Theta_{\mathrm{out},j}\}$
**Ensure:** Output rows $Y \in \mathbb{R}^{M\times O}$
*Offline (at load):* $\mu \leftarrow (4\,\textsc{DecodeE8P12}(\mathrm{indices}) + 1)/2$; pack $\mu$ as nibbles ▷ no run-time codebook lookup
**if** $r_m = 1$ **then** ▷ launch 1: `rot_quant_s4_v6`
  $Z_{i,j} \leftarrow X_{i,\pi(j)}\,\Theta_j$ for every row $i$ and block $j$ ▷ FP32 accumulation
**else** ▷ launch 1: vector quantize
  $Z \leftarrow X$
**end if**
$\hat{q}_{ik} \leftarrow \mathrm{clamp}\big(\mathrm{round}(Z_{ik}\,u_k/c_t), -8, 7\big)$; pack two nibbles per byte
$A \leftarrow \textsc{CutlassInt4Gemm}(\hat{q}, \mu) \in \mathbb{Z}^{M\times(O+8)}$ ▷ launch 2; $A_{i,O} = \sum_k \hat{q}_{ik}$ from the ones row
$\tilde{y}_{in} \leftarrow c_t\,\sigma_n\,(2A_{in} - A_{i,O})/4$ ▷ launch 3 (FP32); uses $4v = 2\mu - 1$
**if** $r_m = 1$ **then**
  $Y_{i,j} \leftarrow \tilde{y}_{i,j}\,\Theta_{\mathrm{out},j} + b_j$ ▷ same launch: `epi_rot_out_v6`
**else**
  $Y_i \leftarrow \tilde{y}_i + b$
**end if**
**return** $Y$ in the input dtype

**Algorithm 4** W4A4 backbone linear (RTX kernel path; three launches, once per policy call)

**Require:** Input rows $X \in \mathbb{R}^{M\times I}$; INT4 weight nibbles $W_q$ with per-output-channel scales $\sigma \in \mathbb{R}^{O}$; static activation scale $s_a$; bias $b$
**Ensure:** Output rows $Y \in \mathbb{R}^{M\times O}$
1: $\hat{q}_{ik} \leftarrow \mathrm{clamp}\big(\mathrm{round}(X_{ik}/s_a), -8, 7\big)$; pack two nibbles per byte ▷ launch 1
2: $A \leftarrow \textsc{CutlassInt4Gemm}(\hat{q}, W_q) \in \mathbb{Z}^{M\times O}$ ▷ launch 2
3: $Y_{in} \leftarrow s_a\, \sigma_n A_{in} + b_n$ ▷ launch 3 (FP32)
4: **return** $Y$ in the input dtype

**Verification and cost.** The quantization kernel with fused rotation has no INT4 code mismatch against the FP64 reference (width 64; $0/3{,}094{,}240$ at width 16 for SmolVLA on both GPUs); epilogue relative error is $2 \times 10^{-7}$. Lattice dequantization has a latency comparable to scalar INT4 dequantization (731.0 versus 722.9 µs across 96 head layers); without rotation, the lattice head achieves $1.162\times$ (GR00T N1.5), $1.050\times$ (CogACT), and $1.030\times$ ($\pi_{0.5}$) speedups over FP16 on RTX 4090. The unfused deployment configuration executes the lattice epilogue and rotation outside GEMM.

**Build and scope.** Using CUTLASS 3.5.1 (`f7b19de`) headers, we build for RTX 3090 (CUDA 12.6, `sm_86`, PyTorch 2.5.1/2.10.0) and RTX 4090 (nvcc 12.1, `sm_89`, PyTorch 2.5.1/2.2.0/2.7.1); an independent clean build on RTX 3090 reproduces kernel outputs and latency ratios within $1\%$. For latency, the backbone runs RTN W4 weights with output-channel scales and A4 activations (Algorithm 4) instead of the W4 weight-only backbone used for accuracy; replacing group scales with output-channel scales leaves the GEMM workload unchanged, and changing only weight values alters latency by at most $0.063\%$. CUTLASS tiles are tuned for RTX 3090 and suboptimal for some RTX 4090 head shapes. Rotation cost per linear-layer invocation is the policy-call latency difference between rotation ON and OFF in the same measurement, divided by the number of head linear-layer invocations (head linear-layer count $\times$ head evaluations per call); it follows block width and layer shape rather than the model (RTX 4090: 3.9 µs for SmolVLA at block 16; 11.9–14.2 µs for CogACT and $\pi_{0.5}$ and 14.7–16.5 µs for GR00T N1.5 at block 64). In the full configuration with two compute nodes, a policy call makes $96 \times 2 = 192$ head linear-layer invocations for GR00T N1.5 and $126 \times 2 = 252$ for $\pi_{0.5}$, and rotation takes 3.16 and 3.27 ms per policy call, respectively (Figure 7). This cost enters Table 6's head-only entries only for the rotated GR00T N1.5 and $\pi_{0.5}$ heads. Table 7 times each kernel launch separately, outside the policy call, splitting rotation into an input-side launch (rotation and A4 quantization) and an output-side launch (epilogue and output rotation). On RTX 4090 at block width 64, the two have similar latencies (5.58 and 5.11 µs), and their sum, which includes quantization and the epilogue, is the non-GEMM cost per linear-layer invocation. In Table 7, each block-64 launch takes $1.4$–$1.9\times$ as long as its block-16 counterpart on both devices. On RTX 4090 at block 64, the per-invocation rotation cost measured within the policy call exceeds this sum. Without rotation, the input-side launch takes 1.05 µs on RTX 4090 and 1.21 µs on RTX 3090, and no output-side rotation runs. Since the epilogue without rotation was not timed, the difference from this column is an upper bound on the isolated-kernel cost of $r_m = 1$ (Eq. (6)), which modules with $D_{\mathrm{rot}} \le 1$ avoid.

Table 7: Isolated time per kernel launch of one linear layer (µs) on RTX 4090 / RTX 3090 ($M$=50; $(I, O) = (720, 960)$ at block 16, $(1024, 1024)$ at block 64 and without rotation; —: not measured).

| Kernel | Block 16 | Block 64 | No rotation |
|---|---|---|---|
| Rotate + quantize | 2.93 / 3.55 | 5.58 / 5.51 | 1.05 / 1.21 |
| Epilogue + output rotation | 2.84 / 3.60 | 5.11 / 5.19 | — |

# D Extended Experimental Results

## D.1 Full Evaluation Grid

Tables 8 and 9 report Table 1's method rows on the LIBERO suites and LIBERO-Plus, including only runs satisfying Appendix A's evaluation conditions. Values marked § use official evaluation paths that do not record per-site call counts; activation statistics confirm quantization at all 180 sites. The *CRAFT* rows follow the main text's model-specific rotation rule, so SmolVLA and XR-0 include only deployment configurations without rotation. "—" denotes structurally infeasible configurations or configurations whose FP16 baseline has low success. Calibration conditions follow checkpoint structure and are identical across method rows within each model. GR00T N1.5 uses suite-specific checkpoints and calibration rollouts (Spatial calibration for LIBERO-Plus, and GPTQ obtained on Object for the W4 backbone). Single-checkpoint models SmolVLA, X-VLA, XR-0, and $\pi_{0.5}$ reuse deployment packs and activation scale tables calibrated on Table 1's benchmarks across the additional benchmarks without recalibration. Since $n$ and perturbation conditions differ across benchmarks, results should be compared only within the same benchmark column.

In Table 8, all four *CRAFT* configurations for GR00T N1.5 match or exceed FP16 on LIBERO-Spatial, -Goal, and -Long. On SmolVLA, head-only *CRAFT* achieves 22.8–31.4, versus only 0.0–5.8 for PTQ4DiT and TFMQ-DM under the same W2A4 head setting. On Long, whose sequential subgoals involve repeated field reuse over longer horizons, head-only *CRAFT* with caching also matches or exceeds FP16 for X-VLA, GR00T N1.5, and $\pi_{0.5}$. On LIBERO-Plus (Table 9), *CRAFT* matches or exceeds FP16 under four of five perturbations for GR00T N1.5 and $\pi_{0.5}$; all four *CRAFT* configurations for $\pi_{0.5}$ remain within 2.75 *pp* of FP16 under all five.

Table 8: Performance comparison on LIBERO-Spatial (S), -Goal (G), and -Long (L). Results are measured with $n = 500$ per suite.

| Method | SmolVLA | | | X-VLA | | | GR00T N1.5 | | | XR-0 | | | $\pi_{0.5}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S | G | L | S | G | L | S | G | L | S | G | L | S | G | L |
| FP16 baseline | 64.2 | 80.4 | 49.8 | 98.0 | 98.2 | 96.0 | 94.4 | 95.8 | 89.4 | 98.4 | 98.8 | 95.6 | 97.8 | 98.0 | 92.0 |
| QuantVLA | 50.4 | 45.8 | 46.4 | 97.6 | 94.8 | 89.4 | 95.8 | 95.6 | 93.2 | 97.0 | 95.4 | 95.2 | 98.0§ | 75.4§ | 88.0§ |
| HoloQ-VLA (head) | 61.4 | 71.4 | 50.6 | 97.4 | 98.6 | 95.0 | 97.4 | 97.0 | 93.0 | 98.2 | 98.0 | 96.6 | 98.0 | 98.4 | 93.2 |
| PTQ4DiT | 0.0 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 75.8 | 63.2 | 33.2 | 97.0 | 98.4 | 96.0 | 53.6 | 52.2 | 9.0 |
| TFMQ-DM | 1.6 | 5.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 95.8 | 98.0 | 89.2 | 87.8 | 76.2 | 20.0 |
| TeaCache | 64.2 | 78.4 | 50.0 | 98.0 | 97.8 | 97.6 | 96.2 | 96.0 | 93.2 | 98.8 | 98.8 | 96.6 | 98.4 | 98.2 | 89.6 |
| MagCache | 63.6 | 78.4 | 48.2 | 97.8 | 98.2 | 96.4 | 95.0 | 96.0 | 89.8 | 99.4 | 98.8 | 95.6 | 98.4 | 97.6 | 93.4 |
| CacheQuant | 59.8 | 75.4 | 44.8 | 96.4 | 97.2 | 97.0 | 95.6 | 96.4 | 90.6 | 98.6 | 98.8 | 96.6 | 98.6 | 97.2 | 92.4 |
| *CRAFT* w/o backbone, w/o cache | 22.8 | 31.4 | 27.2 | 97.4 | 96.2 | 95.4 | 95.8 | 97.6 | 91.6 | 98.4 | 98.6 | 95.0 | 99.0 | 96.2 | 91.0 |
| *CRAFT* w/o backbone | 25.0 | 28.4 | 27.8 | 97.8 | 97.2 | 96.2 | 95.6 | 97.6 | 91.6 | 97.2 | 98.6 | 94.0 | 97.6 | 97.0 | 93.2 |
| *CRAFT* w/o cache | 14.4 | 10.8 | 11.8 | 97.6 | 96.0 | 94.4 | 97.8 | 96.8 | 91.2 | 97.8 | 98.8 | 96.4 | 99.2 | 98.2 | 91.8 |
| *CRAFT* | 13.8 | 12.2 | 12.0 | 97.4 | 96.6 | 94.8 | 98.6 | 97.2 | 90.0 | 97.6 | 98.8 | 96.2 | 98.6 | 96.8 | 92.6 |

Table 9: Performance comparison on LIBERO-Plus under camera (Cam), lighting (Lgt), language (Lng), table (Tab), and table+background (T+B) perturbations. Results are measured with $n = 400$ per perturbation.

| Method | GR00T N1.5 | | | | | X-VLA | | | | | XR-0 | | | | | $\pi_{0.5}$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cam | Lgt | Lng | Tab | T+B | Cam | Lgt | Lng | Tab | T+B | Cam | Lgt | Lng | Tab | T+B | Cam | Lgt | Lng | Tab | T+B |
| FP16 baseline | 76.50 | 97.25 | 89.50 | 96.50 | 97.00 | — | 97.00 | 83.00 | 97.25 | 96.00 | 51.00 | 90.75 | 95.25 | 96.75 | 94.50 | 81.50 | 98.75 | 85.25 | 99.25 | 99.50 |
| QuantVLA | 68.75 | 96.50 | 87.50 | 97.00 | 96.25 | — | 97.75 | 76.00 | 97.25 | 98.50 | 45.50 | 90.00 | 87.50 | 90.50 | 93.50 | 60.75§ | 97.75§ | 79.50§ | 98.75§ | 98.75§ |
| HoloQ-VLA (head) | 74.75 | 97.50 | 89.75 | 96.25 | 98.00 | — | 96.50 | 79.25 | 96.00 | 96.25 | 47.75 | 91.75 | 97.00 | 94.25 | 96.00 | 77.25 | 99.00 | 81.25 | 98.00 | 98.50 |
| PTQ4DiT | 45.50 | 73.75 | 60.50 | 78.50 | 59.75 | — | 0.25 | 0.00 | 0.00 | 1.00 | 48.25 | 91.00 | 93.50 | 92.75 | 92.75 | 19.75 | 42.75 | 23.25 | 28.75 | 36.00 |
| TFMQ-DM | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | — | 0.00 | 0.00 | 0.00 | 0.00 | 43.75 | 82.50 | 92.00 | 85.75 | 79.00 | 54.75 | 88.00 | 70.25 | 70.75 | 69.00 |
| TeaCache | 75.50 | 98.00 | 90.50 | 95.25 | 98.00 | — | 97.50 | 81.50 | 96.50 | 95.50 | 51.00 | 92.00 | 96.50 | 94.25 | 94.50 | 80.25 | 99.75 | 84.00 | 98.75 | 98.25 |
| MagCache | 75.50 | 97.25 | 89.75 | 96.00 | 98.00 | — | 97.00 | 82.50 | 96.75 | 95.75 | 48.50 | 92.75 | 95.75 | 94.50 | 96.00 | 83.50 | 99.00 | 84.50 | 98.75 | 99.00 |
| CacheQuant | 76.50 | 96.75 | 90.75 | 96.75 | 98.25 | — | 97.50 | 82.50 | 97.50 | 96.50 | 47.50 | 93.00 | 96.75 | 94.50 | 95.50 | 78.00 | 99.75 | 83.75 | 98.75 | 100.00 |
| *CRAFT* w/o backbone, w/o cache | 74.75 | 97.25 | 89.75 | 97.50 | 96.75 | — | 94.50 | 76.50 | 96.75 | 96.25 | 49.75 | 91.25 | 94.25 | 93.75 | 92.75 | 79.50 | 98.50 | 85.00 | 99.25 | 99.25 |
| *CRAFT* w/o backbone | 74.25 | 97.25 | 90.75 | 97.75 | 98.00 | — | 96.50 | 74.50 | 96.50 | 97.00 | 46.50 | 89.50 | 95.75 | 93.00 | 91.75 | 82.00 | 99.00 | 85.75 | 99.50 | 99.00 |
| *CRAFT* w/o cache | 70.00 | 97.50 | 90.25 | 97.75 | 97.50 | — | 95.75 | 72.75 | 97.75 | 96.50 | 46.50 | 90.50 | 93.75 | 94.00 | 92.25 | 81.00 | 99.00 | 82.50 | 98.50 | 98.75 |
| *CRAFT* | 68.75 | 97.50 | 89.75 | 97.75 | 98.50 | — | 95.50 | 74.00 | 96.75 | 96.25 | 48.75 | 88.25 | 95.00 | 93.25 | 90.25 | 82.50 | 99.00 | 85.25 | 98.50 | 100.00 |

### D.2 Replanning Interval

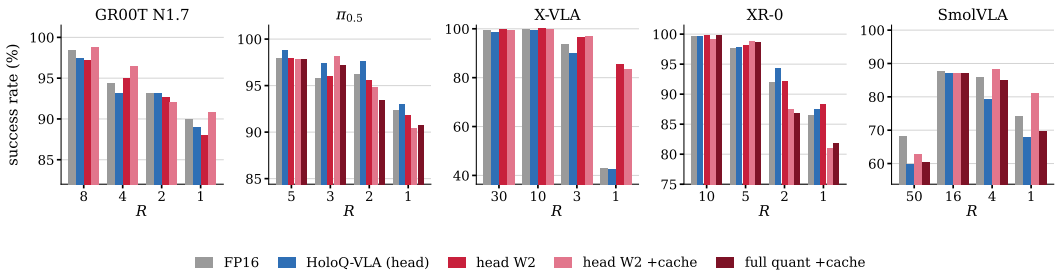


Figure 8: Success rate versus replanning interval $R$ on LIBERO-Object, from each model's default $R$ to $R = 1$. For each model, only configurations evaluated at every $R$ are shown.

**Replanning interval.** As $R$ decreases from the deployment default to 1, *CRAFT* configurations remain within $2.8\,pp$ of FP16 for GR00T N1.7 and $\pi_{0.5}$. For X-VLA at $R$=1, FP16 and HoloQ-VLA (head) fall to 42.8 and 42.4, respectively, while *CRAFT* head-only maintains 83.4–85.6 with and without caching (Figure 8). For XR-0, head-only without caching differs from FP16 by $+0.2$ and $+1.8\,pp$ at $R$=2 and $R$=1, respectively; *CRAFT*, evaluating the head once in five steps, achieves 86.8 and 81.8, respectively. CogACT executes one step of its 16-step chunk, fixing $R$ at 1. Caching bars use model-specific compute-node sets, so evaluation NFE varies across models. The full *CRAFT* configuration on GR00T N1.7 and X-VLA and the head-only configuration on SmolVLA were evaluated only at the default $R$ and are not shown.

### D.3 Activation-Precision Ablation

Table 10 varies only activation bit width, fixing the GR00T N1.5 action head's W2 weights. Sharing one activation-scale table across all steps in the same head yields 95.2, 98.6, 97.0, and 91.8 on Spatial, Object, Goal, and Long, respectively, within $0.6\,pp$ of the step-wise tables in Table 2 (95.8, 98.6, 97.6, 91.6) across all suites. With W2 weights fixed, PTQ4DiT and TFMQ-DM average 92.70 and 91.90 at A8, falling to 63.05 and 0.00 at A4. By contrast, *CRAFT* retains an average SR of 95.90 at A4, compared with 94.00 for FP16, and remains above FP16 on each of the four suites. This is consistent with error-redistributed W2 reducing the output-error share in high-activation-energy eigendirections (Figure 15) and retaining success rate despite additional A4 activation error.

Table 10: Activation precision at fixed W2 head weights on GR00T N1.5 (LIBERO, $n = 500$).

| Method | Act. | Spatial | Object | Goal | Long | Avg. |
|---|---|---|---|---|---|---|
| FP16 | | 94.4 | 96.4 | 95.8 | 89.4 | 94.00 |
| PTQ4DiT | A16 | 34.0 | 84.8 | 54.0 | 50.6 | 55.85 |
| | A8 | 92.4 | 96.2 | 90.0 | **92.2** | 92.70 |
| | A4 | 75.8 | 80.0 | 63.2 | 33.2 | 63.05 |
| TFMQ-DM | A16 | 21.6 | 54.2 | 48.4 | 31.8 | 39.00 |
| | A8 | 90.0 | 97.4 | 93.2 | 87.0 | 91.90 |
| | A4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 |
| *CRAFT* | A4 | **95.8** | **98.6** | **97.6** | 91.6 | **95.90** |

### D.4 COMPUTE-NODE SCHEDULES

Table 11 reports, for one evaluation interval per model (LIBERO-Object for seven models and SimplerEnv `gr_va_min` $n = 240$ for CogACT, as in Table 1), TeaCache's and MagCache's distinct step-set counts across policy calls, the most frequent set and its frequency, and measured mean head evaluations, alongside the proposed static schedule's NFE. Gates select one set across all calls in fourteen of sixteen combinations; exceptions are TeaCache on CogACT (six sets; most frequent: $86.0\%$) and GR00T N1.5 (two sets, $\{0, 6, 7\}$ and $\{0, 5, 7\}$; most frequent: $75.7\%$). Table 12 reports per-call decision overhead in RTX latency measurements relative to a uniform schedule with the same realized NFE. Static and offline schedules range from $-0.34$ to $+0.43$ ms (null-pass lower bound: CogACT $0.24$, $\pi_{0.5}$ $0.003$ ms); TeaCache incurs $+2.77$ to $+7.05$ ms on all three models on both devices, and AdaptiveDiffusion $+3.51$ and $+4.53$ ms on GR00T N1.5. The concentration of both adaptive gates on a single dominant step set motivates model-specific static schedules in the evaluated setting; across all 16 model–gate combinations, ***CRAFT*** (the Ours column) requires no more head evaluations than the corresponding gate.

Table 11: Step sets chosen by the dynamic caching gates in closed loop (window: suite $\cdot\, n \cdot R \cdot T$; $^{\ddagger}$ official default $R$).

| Model | Gate | Window | #Sets | Top set | Share (%) | Calls | NFE | Ours | SR |
|---|---|---|---|---|---|---|---|---|---|
| SmolVLA | TeaCache | Object·500·50·10 | 1 | $\{0, 4, 8, 9\}$ | 100 | 1,983 | 4 | 3 | 69.0 |
| SmolVLA | MagCache | Object·500·50·10 | 1 | $\{0, 3, 6, 9\}$ | 100 | 1,964 | 4 | 3 | 69.8 |
| X-VLA | TeaCache | Object·500·30·10 | 1 | $\{0, 9\}$ | 100 | 2,522 | 2 | 2 | 99.2 |
| X-VLA | MagCache | Object·500·30·10 | 1 | $\{0, 3, 6, 9\}$ | 100 | 2,466 | 4 | 2 | 99.6 |
| GR00T N1.5 | TeaCache | Object·500·8·8 | 2 | $\{0, 6, 7\}$ | 75.7 | 9,095 | 3 | 2 | 96.6 |
| GR00T N1.5 | MagCache | Object·500·8·8 | 1 | $\{0, 3, 6\}$ | 100 | 9,006 | 3 | 2 | 97.2 |
| GR00T N1.7 | TeaCache | Object·500·8·4 | 1 | $\{0, 3\}$ | 100 | 9,084 | 2 | 2 | 98.0 |
| GR00T N1.7 | MagCache | Object·500·8·4 | 1 | $\{0, 3\}$ | 100 | 9,084 | 2 | 2 | 98.0 |
| XR-0 | TeaCache | Object·500·10·5 | 1 | $\{0, 4\}$ | 100 | 6,809 | 2 | 1 | 99.6 |
| XR-0 | MagCache | Object·500·10·5 | 1 | $\{0, 3\}$ | 100 | 6,799 | 2 | 1 | 99.4 |
| $\pi_{0.5}$ | TeaCache | Object·500·5$^{\ddagger}$·10 | 1 | $\{0, 9\}$ | 100 | 14,737 | 2 | 2 | 96.8 |
| $\pi_{0.5}$ | MagCache | Object·500·5$^{\ddagger}$·10 | 1 | $\{0, 3, 6, 9\}$ | 100 | 14,472 | 4 | 2 | 98.4 |
| $\pi_0$ | TeaCache | Object·500·5$^{\ddagger}$·10 | 1 | $\{0, 9\}$ | 100 | 14,476 | 2 | 2 | 96.0 |
| $\pi_0$ | MagCache | Object·500·5$^{\ddagger}$·10 | 1 | $\{0, 3, 6, 9\}$ | 100 | 14,578 | 4 | 2 | 97.0 |
| CogACT | TeaCache | SimplerEnv·240·1·10 | 6 | $\{0, 8, 9\}$ | 86.0 | 19,200 | 2.98 | 2 | 62.1 |
| CogACT | MagCache | SimplerEnv·240·1·10 | 1 | $\{0, 3, 6, 9\}$ | 100 | 19,200 | 4 | 2 | 70.8 |

Table 12: Per-call gate overhead (ms) over a uniform schedule with the same realized NFE (FP16 head, batch 1). G/C/$\pi$ = GR00T N1.5 / CogACT / $\pi_{0.5}$.

| Method | Decision | NFE (G/C/$\pi$) | Overhead RTX 4090 (G/C/$\pi$) | Overhead RTX 3090 (G/C/$\pi$) |
|---|---|---|---|---|
| *CRAFT* (static) | none | 2/2/2 | $+0.43$ / $+0.24$ / $-0.00$ | — |
| LeMiCa | offline schedule | 4/5/5 | $-0.14$ / $-0.15$ / $+0.09$ | — |
| BAC-DP | offline schedule | 4/5/5 | $+0.03$ / $-0.31$ / $+0.08$ | — |
| CQ-DPS | offline schedule | —/5/5 | — / $-0.34$ / $+0.08$ | — |
| TaylorSeer + ours | extrapolation | 2/2/2 | $-0.05$ / $+0.09$ / $+0.14$ | $+0.25$ / $+0.03$ / $+0.01$ |
| MagCache | online magnitude ratio | 3/4/4 | $-0.18$ / $-0.14$ / $+0.03$ | $-0.07$ / $-0.01$ / $-0.00$ |
| TeaCache | online signal | 4/4/3 | $+4.07$ / $+2.77$ / $+3.98$ | $+5.83$ / $+4.09$ / $+7.05$ |
| AdaptiveDiffusion | online accumulated error | 7/9/9 | $+3.51$ / $-0.13$ / $+0.37$ | $+4.53$ / $+1.25$ / $+0.65$ |

**Schedule enumeration.** For GR00T N1.5 with $T = 8$, we enumerate all 64 evaluation-step sets $C$ containing step 0 with $|C| \leq 4$ $(1+7+21+35)$ and measure FP16 and W2A4 offline final-action error over 512 windows (Table 13), relative to all-step evaluation, in units of the noise floor (mean error between two full-step runs differing only in noise). From 1 to 4 evaluations, the minimum error falls from $1.43$ to $0.15$ floors (FP16) and from $1.35$ to $0.34$ (W2A4), whereas for each evaluation count from 2 to 4 the lowest- and highest-error candidates differ by $0.72$–$0.91$ (FP16) and $0.44$–$0.66$ (W2A4), both below one floor. The offline rankings of the two heads nearly coincide (Spearman $0.96$–$1.00$). The seven closed-loop two-node placements yield SRs of $98.2$–$99.0$ for W2A4 and $96.4$–$97.6$ for FP16, with a spread of at most $1.2\,pp$ within each precision setting, and closed-loop SR does not vary monotonically with offline error at a fixed count.

Table 13: Compute-node enumeration for GR00T N1.5 ($T = 8$, $|C| \leq 4$): offline terminal action error in units of the noise floor (best / median / worst).

| | | FP16 head (error / floor) | | | W2A4 head (error / floor) | | | Closed-loop SR range | |
|---|---|---|---|---|---|---|---|---|---|
| $\lvert C\rvert$ | # cand. | best | median | worst | best | median | worst | FP16 | W2A4 |
| 1 | 1 | | 1.43 | | | 1.35 | | — | — |
| 2 | 7 | 0.56 | 0.88 | 1.28 | 0.77 | 0.90 | 1.21 | 96.4–97.6 | 98.2–99.0 |
| 3 | 21 | 0.25 | 0.54 | 1.16 | 0.46 | 0.69 | 1.12 | — | — |
| 4 | 35 | 0.15 | 0.43 | 1.03 | 0.34 | 0.55 | 0.99 | — | — |
| 8 | 1 | | 0 (reference) | | | 0 (reference) | | 96.4 | 98.6 |

### D.5 Hardware, Energy, and Batch Scaling

**Caching speedup ceiling.** Caching accelerates only the action head, so for head share $h$ of CUDA-graph end-to-end latency (unlike summed module times in Figure 2) and skipped fraction $\rho_{\text{skip}} = 1-\hat{T}/T$, end-to-end speedup is bounded by $1/(1 - h\rho_{\text{skip}})$. Under Table 14's deployment execution conditions, $h$ ranges from $45.4\%$ for GR00T N1.5 to $14.3\%$ for CogACT, yielding bounds of $1.52\times$ and $1.13\times$. Measured caching speedup reaches $98.7$–$99.9\%$ of the bound across three models, consistent with the absence of run-time scheduling decisions. Skipped steps retain only the solver state update, costing less than a head evaluation; the remaining $0.1$–$1.3\%$ gap is consistent with this computation. With comparable skip fractions ($75$–$80\%$), the bound is determined mainly by the head latency share $h$. Caching gains thus depend on architecture. This model dependence motivates combining head-evaluation reduction with backbone W4 quantization rather than relying on caching alone. Backbone W4 reduces weight traffic even for small $h$: on RTX 4090, CogACT's caching bound is $1.13\times$, but *CRAFT* reaches $2.48\times$, above TeaCache ($1.04\times$) and MagCache ($1.09\times$) (Table 6).

Table 14: Action-head latency share and the resulting speedup ceiling for caching (RTX 4090, CUDA graph, batch $= 1$).

| Model | Head share (%) | Ceiling | Attained |
|---|---|---|---|
| GR00T N1.5 | 45.4 | 1.52× | 98.7 % |
| $\pi_{0.5}$ | 40.5 | 1.48× | 99.9 % |
| CogACT | 14.3 | 1.13× | 99.5 % |

**Hardware dependence and energy.** On H100, we use dense weights for fake-quant activations and rotation rather than a packed kernel, so latency and energy include low-bit emulation costs. Under this emulated setting, full quantization with caching remains $1.18\times$ faster than FP16, while FP16 with static caching reduces latency by $48.2\%$ and energy per call by $33.2\%$ (Table 15). Static caching stores only the last predicted field. FP16 with caching has the same measured peak memory as FP16 alone (5.31 GB), whereas full quantization with caching uses 11.34 GB in this emulated setting because the harness keeps FP16 weights, dense copies of the quantized weights, and dense rotation matrices resident. Per-call traffic falls from $11.899$ GiB for FP16 to $5.642$ GiB for FP16 +caching and $2.425$ GiB for full quant +caching. On bandwidth-limited GPUs, reduced traffic translates to lower latency, consistent with full quantization with caching being faster than FP16 with caching for GR00T N1.5 on RTX 3090 and RTX 4090 (Table 6).

**Batch scaling.** On H100, head-only +caching maintains a $1.20$–$1.21\times$ speedup over FP16 from $B{=}1$ to 8 (Table 16), versus $0.415$–$0.580\times$ for prior methods at $B{=}1$. At $B{=}16$, it remains on par with FP16 ($0.998\times$), whereas *CRAFT* reaches $2.06\times$ at batch 1 on RTX 3090 with packed kernels (Table 6). Latency gains appear in the small-batch, bandwidth-limited regime assumed in §4.2.

Table 15: Latency, energy, and peak memory per policy call on an H100 (GR00T N1.5).

| Configuration | $J_{\text{marginal}}$/call | vs. FP16 | Wall | vs. FP16 | Peak HBM |
|---|---|---|---|---|---|
| FP16 (NFE 8) | 6.565 J | — | 121.88 ms | — | 5.31 GB |
| ***CRAFT* full quant +caching** (NFE 2) | 8.625 J | +31.4% | 103.14 ms | −15.4% | 11.34 GB |
| FP16 +caching (NFE 2) | 4.384 J | −33.2% | 63.18 ms | −48.2% | 5.31 GB |

Table 16: End-to-end latency (p50, ms) and speedup over FP16 across batch sizes on an H100.

| Configuration | $B$=1 | $B$=4 | $B$=8 | $B$=16 |
|---|---|---|---|---|
| FP16 baseline | 113.21 | 116.09 | 123.21 | 162.18 |
| HoloQ-VLA W4A4 | 272.98 (0.415×) | 284.56 (0.408×) | 296.53 (0.416×) | 368.11 (0.441×) |
| PTQ4DiT W2A4 | 205.60 (0.551×) | 210.48 (0.552×) | 218.34 (0.564×) | 264.27 (0.614×) |
| TFMQ-DM W2A4 | 196.35 (0.577×) | 202.09 (0.574×) | 210.46 (0.585×) | 253.37 (0.640×) |
| CacheQuant W8A8 | 195.36 (0.580×) | 200.82 (0.578×) | 208.54 (0.591×) | 252.58 (0.642×) |
| *CRAFT* head-only | 264.84 (0.427×) | 274.27 (0.423×) | 277.39 (0.444×) | 332.17 (0.488×) |
| *CRAFT* full quant | 264.99 (0.427×) | 273.54 (0.424×) | 278.54 (0.442×) | 332.18 (0.488×) |
| ***CRAFT* head-only +caching** | **93.79 (1.207×)** | **96.09 (1.208×)** | **102.87 (1.198×)** | 162.53 (0.998×) |

### D.6 Head Storage

Storage depends on the rotation representation. Between *CRAFT*'s W2 and W4 heads, only weights change (2.000 versus 4.000); rotation 0.9968, activation tables 0.1261, $SU/SV$ 0.0155, and container 0.0026 remain unchanged, totaling 3.1410 and 5.1410 bpw. HoloQ-VLA's W4 head, with its rotations stored block-diagonally, takes 5.38 bpw (Figure 9).

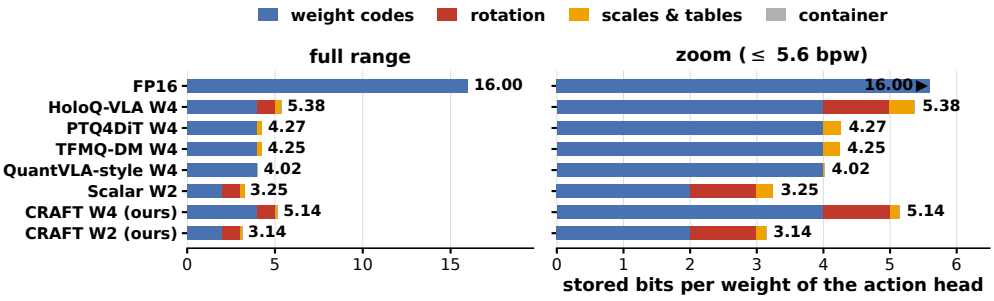


Figure 9: Stored bits per weight of the GR00T N1.5 action head.

# E Analysis and Visualization

## E.1 Error Metrics and Closed-Loop Performance

Table 17 reports rank correlations of two head error metrics with $\Delta\mathrm{SR} = \mathrm{SR} - \mathrm{SR}_{\mathrm{FP16}}$, both computed at each head's first attention projection. The first column covers 40 configurations across eight models: the four W2A4 head rows in Table 1 and *CRAFT* head-only with rotation on every head (Always in Table 5). The second adds 28 other W2 head configurations of the same models, combining all 68 after centering ranks within each model. Here, $\varepsilon$ is the activation-weighted relative output error in Eq. (1)(at the module level, the sum of the layer-wise numerators divided by the sum of the denominators), whereas weight reconstruction error $\|\Delta W\|_F^2/\|W\|_F^2$ excludes activation statistics. We use rank correlation to assess monotonic associations without assuming a linear relationship between error and success rate. Within-model rankings reduce confounding from differences in task difficulty and FP16 success across models. In both columns, larger $\varepsilon$ is associated with greater success-rate degradation ($-0.52$, $-0.63$), whereas weight reconstruction error has the opposite sign ($+0.37$, $+0.45$). Thus, lower weight reconstruction error does not necessarily correspond to higher closed-loop success. In these experiments, output error is negatively correlated with success-rate change, supporting the use of an activation-aware error metric.

Table 17: Spearman correlations between $\Delta\mathrm{SR}$ and two head error metrics.

| Metric | Main-table rows ($n$=40) | Within-model ($n$=68) |
|---|---|---|
| $\varepsilon$ (Eq. 1) | $-0.52$ ($p$=0.0006) | $-0.63$ ($p<10^{-8}$) |
| $\|\Delta W\|_F^2/\|W\|_F^2$ | $+0.37$ ($p$=0.02) | $+0.45$ ($p<10^{-4}$) |

**Error-metric correlation subset.** The correlation-sign difference in Table 17 persists after excluding the two diffusion PTQ rows (PTQ4DiT and TFMQ-DM), which exhibit near-zero success rates across multiple models. Of the 40 configurations, retaining only the three rows (QuantVLA-style[‡], HoloQ-VLA[‡], and *CRAFT* head-only, Always) yields 24 configurations with rank correlations with $\Delta\mathrm{SR}$ of $-0.60$ ($p = 0.002$) for $\varepsilon$ and $+0.13$ ($p = 0.54$) for weight reconstruction error. The weight reconstruction error correlation decreases from $+0.37$ to $+0.13$, whereas the $\varepsilon$ correlation strengthens from $-0.52$ to $-0.60$. Thus, the sign difference in Table 17 does not depend solely on the two diffusion PTQ rows.

## E.2 Activation and Output-Error Structure

**Activation surface.** Figure 10 shows input activation magnitudes and layer-wise absmax at five linear layers during GR00T N1.5 FP16 rollouts: two action head, one vision encoder, and two language backbone layers. Pronounced channel-wise activation outliers appear only in the two language backbone layers (absmax 14.7, 74), not the two action head layers. Thus, the channel-wise activation outliers commonly observed in the language backbone cannot alone explain action head W2 degradation.

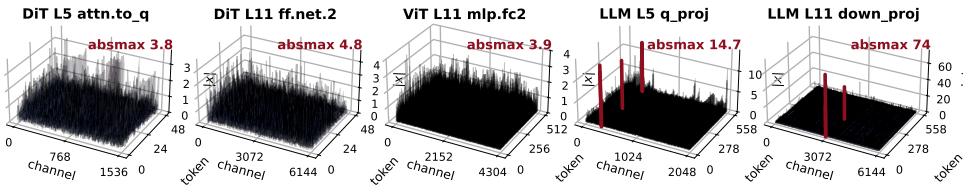


Figure 10: Input activation magnitude $|x|$ at five linear sites of GR00T N1.5 (FP16 rollout).

**Output-error eigen-directions across models.** Figures 15 and 16 extend Figure 3's eigendirection decomposition (Eq. (2)) to the first action head attention projection and first language backbone `q_proj` of all eight models. At both locations in every model, the five scalar W2 columns' output error concentrates in high-activation-energy eigendirections: the top $1\%$ account for $26.1$–$97.4\%$ of head error and $42.0$–$96.8\%$ of backbone error. Error-redistributed W2 reduces these shares to $3.3$–$41.4\%$ and $3.9$–$16.5\%$, respectively, yielding the smallest $\varepsilon$ in every row. Since $H$'s spectral concentration varies across matrices (e.g., all scalar columns for the XR-0 head are at least $96.6\%$), we compare columns within rows rather than absolute values across rows. Consistent with Table 17's rank correlations, error-redistributed W2 redistributes quantization error toward lower-energy eigendirections and reduces output error.

Figure 11 compares the same methods in weight space. Each panel plots $|\Delta W_{ij}|^2 H_{jj}$ elementwise (top: action head; bottom: language backbone projection), showing where $H_{jj}$-weighted error concentrates across input channels. FP panels show max/median input-energy ratios of $36\times$ (head) and $662\times$ (backbone), indicating backbone input activation energy concentrates on fewer channels. Under the three scalar quantizers, error is concentrated in these channels, whereas error-redistributed W2 substantially reduces this concentration.

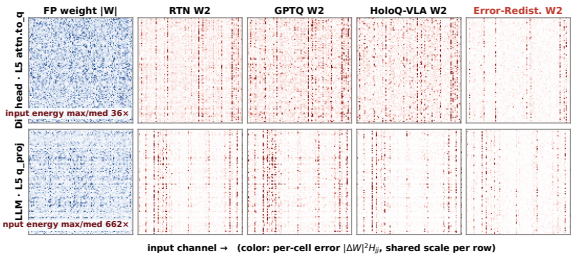


Figure 11: FP weight magnitude (blue) and per-cell W2 error (red) under each quantizer on GR00T N1.5.

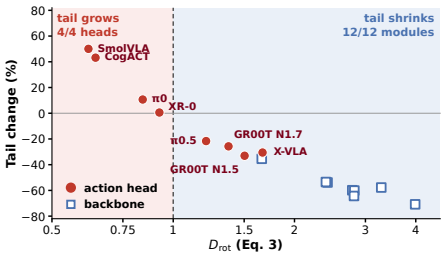


Figure 12: Tail change after block rotation versus $D_{\mathrm{rot}}$ for eight VLAs.

## E.3 Rotation Response

**Rotation response of all modules.** Figure 12 shows $D_{\mathrm{rot}}$ and post-rotation tail change for 16 action head and language backbone modules across eight models; Figure 13 shows each module's distribution. In Eq. (3), $\mathcal{M}_m$ contains analyzed-layer projection matrices with dimensions divisible by the rotation block size; tail change compares the median max/RMS of $W'_\ell$ with the median of $\rho_\ell$. For SmolVLA, we use block-16 rotation in this diagnostic analysis because its action expert input dimension is not divisible by 64.

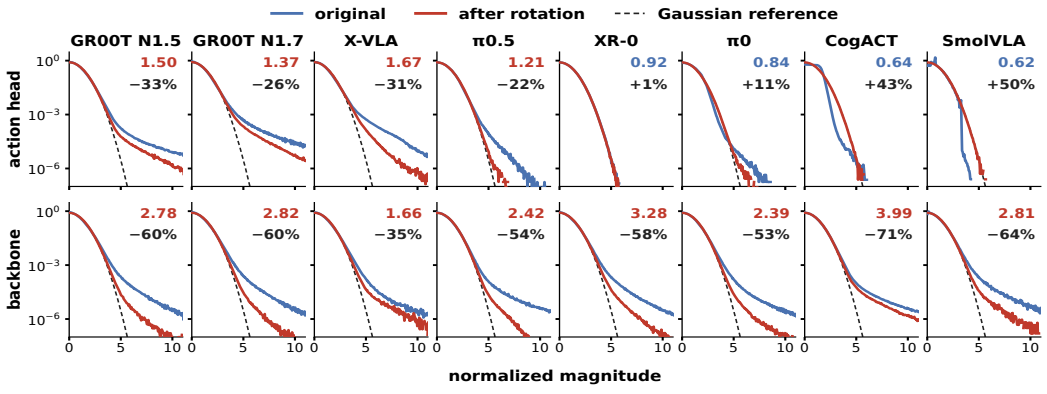


Figure 13: Weight magnitudes before and after block rotation (action head and backbone, eight VLAs); numbers: $D_{\mathrm{rot}}$ (red: $> 1$) and tail change.

Figure 14 shows four modules from Figure 13 with their $D_{\text{rot}}$ values. Rotation reduces tails by 33% and 60% for the GR00T N1.5 action head ($D_{\text{rot}} = 1.50$) and backbone (2.78), respectively, whose tails exceed the Gaussian reference. Tails increase by 43% for the light-tailed CogACT action head (0.64) and change little for the near-boundary XR-0 action head (0.92; +1%). These four representative modules span both sides of the $D_{\text{rot}} = 1$ boundary and the near-boundary regime, consistent with the directional criterion underlying Eq. (6): modules with $D_{\text{rot}} > 1$ show tail reduction, whereas those below the boundary show tail growth or little change. CogACT's closed-loop results exhibit the corresponding directional trend: rotating its head, whose tail increases, lowers success from 65.8 to 55.4, whereas settings differ by 0.2 *pp* for the near-boundary XR-0 head (Table 5). Since $D_{\text{rot}}$ uses only weight statistics (Eq. (3)), *CRAFT* selects rotation before deployment pack generation, without closed-loop evaluation.

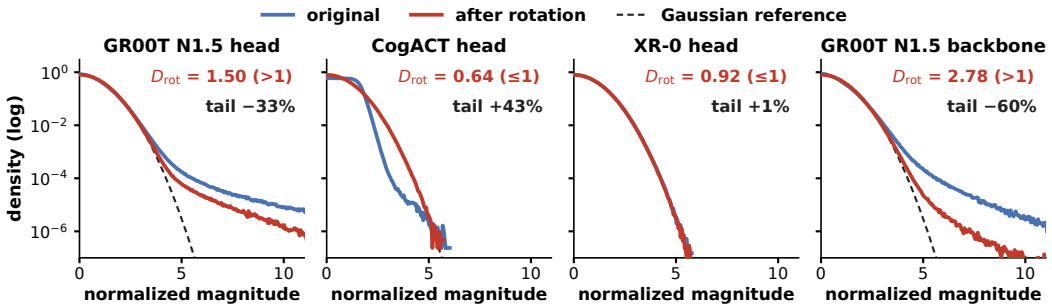


Figure 14: Weight magnitudes before and after block rotation for four representative modules. Rotation is applied here for diagnosis only; CogACT and XR-0 heads deploy with $r_m = 0$.

**Rotation ON/OFF across benchmarks.** Table 18 compares deployment packs with rotation (ON) and without rotation (OFF) for XR-0 and $\pi_0$ heads with $r_m = 0$, under the same *CRAFT* configurations; each pack's stored rotation fields identify rotation status. For each model, we compare rotation ON and OFF in 32 matched configuration–benchmark pairs. For near-boundary XR-0 ($D_{\text{rot}} = 0.92$), ON-minus-OFF differences range from $-2.25$ to $+3.50$ *pp* across the thirty-two entries. For $\pi_0$ ($D_{\text{rot}} = 0.84$), ON exceeds OFF by 0.25–7.00 *pp* in 26 of 32 comparisons, is equal in five, and is 0.25 *pp* lower in one; however, all configurations remain below 8% SR. This matches Always exceeding None by 3.0 *pp* on LIBERO-Object (Table 5). For $\pi_0$, increasing head precision to W4 yields a gain (+45.8 *pp*, Table 1) exceeding rotation-setting differences (at most 7.00 *pp*). The table excludes benchmark-specific ON/OFF comparisons for models with $D_{\text{rot}} > 1$.

Table 18: Rotation ON versus OFF on the $r_m = 0$ heads of XR-0 and $\pi_0$. Results are measured with $n = 500$ per suite and $n = 400$ per perturbation.

| Method | Rot. | S | G | L | Cam | Lgt | Lng | Tab | T+B |
|---|---|---|---|---|---|---|---|---|---|
| XR-0 ($D_{\text{rot}} = 0.92$) | | | | | | | | | |
| *CRAFT* w/o backbone, w/o cache | ON | 98.4 | 98.2 | 96.8 | 47.50 | 91.75 | 95.75 | 93.50 | 94.50 |
| *CRAFT* w/o backbone, w/o cache | OFF | 98.4 | 98.6 | 95.0 | 49.75 | 91.25 | 94.25 | 93.75 | 92.75 |
| *CRAFT* w/o backbone | ON | 98.6 | 98.6 | 96.8 | 46.25 | 93.00 | 95.50 | 95.50 | 95.25 |
| *CRAFT* w/o backbone | OFF | 97.2 | 98.6 | 94.0 | 46.50 | 89.50 | 95.75 | 93.00 | 91.75 |
| *CRAFT* w/o cache | ON | 98.8 | 99.2 | 97.2 | 46.50 | 91.75 | 96.25 | 95.25 | 93.50 |
| *CRAFT* w/o cache | OFF | 97.8 | 98.8 | 96.4 | 46.50 | 90.50 | 93.75 | 94.00 | 92.25 |
| *CRAFT* | ON | 99.4 | 99.2 | 97.6 | 47.25 | 91.75 | 95.50 | 92.75 | 91.25 |
| *CRAFT* | OFF | 97.6 | 98.8 | 96.2 | 48.75 | 88.25 | 95.00 | 93.25 | 90.25 |
| $\pi_0$ ($D_{\text{rot}} = 0.84$) | | | | | | | | | |
| *CRAFT* w/o backbone, w/o cache | ON | 4.0 | 2.6 | 0.0 | 0.75 | 2.25 | 0.50 | 1.25 | 5.25 |
| *CRAFT* w/o backbone, w/o cache | OFF | 0.2 | 0.0 | 0.0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.25 |
| *CRAFT* w/o backbone | ON | 1.8 | 2.2 | 0.0 | 0.25 | 3.75 | 0.00 | 1.50 | 7.00 |
| *CRAFT* w/o backbone | OFF | 0.0 | 0.0 | 0.0 | 0.50 | 0.00 | 0.00 | 0.00 | 0.00 |
| *CRAFT* w/o cache | ON | 2.4 | 1.8 | 0.0 | 0.25 | 3.75 | 0.25 | 2.75 | 5.75 |
| *CRAFT* w/o cache | OFF | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.25 |
| *CRAFT* | ON | 2.2 | 1.6 | 0.0 | 0.25 | 2.75 | 0.25 | 0.75 | 5.00 |
| *CRAFT* | OFF | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

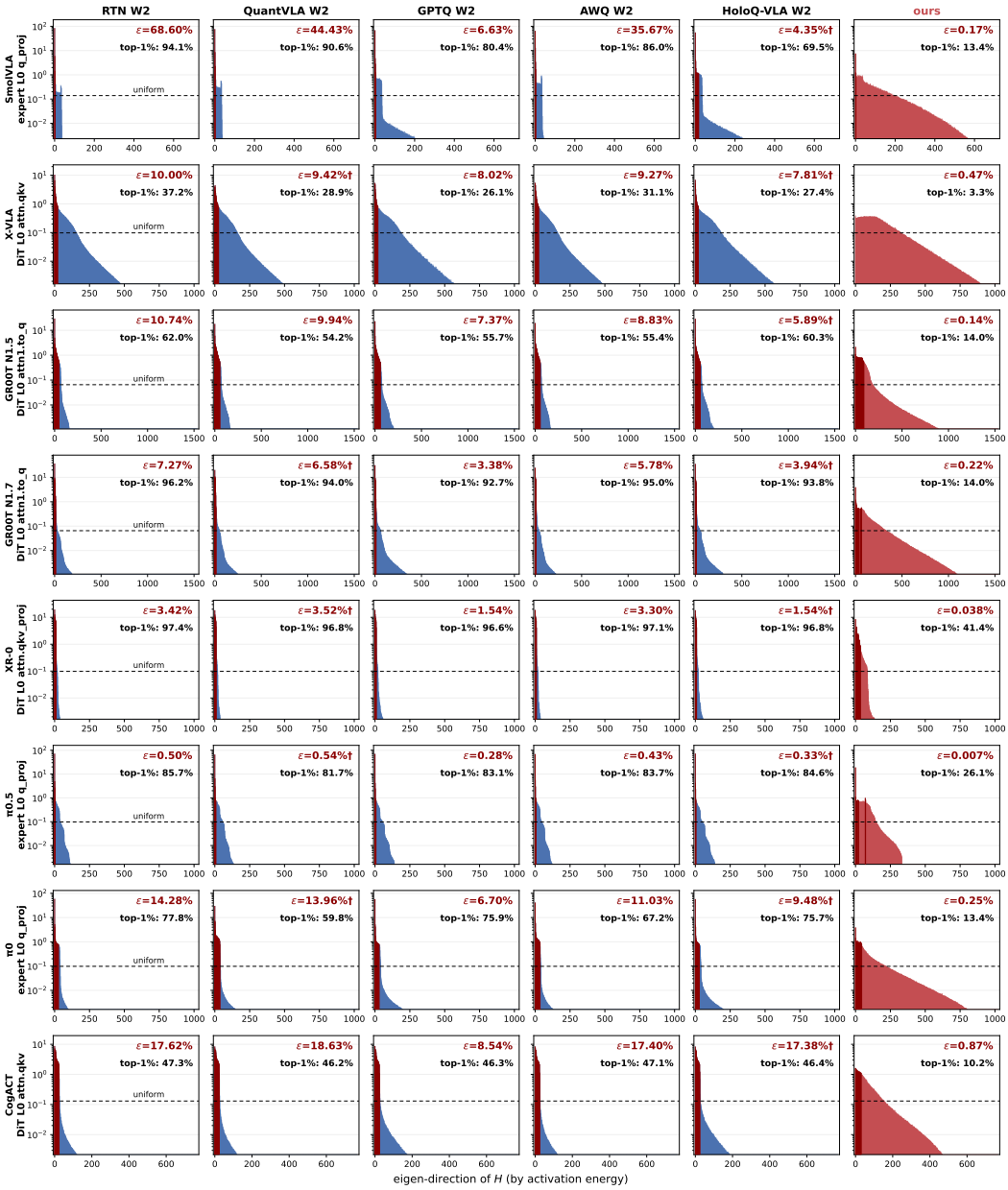


Figure 15: W2 output-error share along the eigen-directions of $H$ at the first action-head attention projection of eight VLAs.

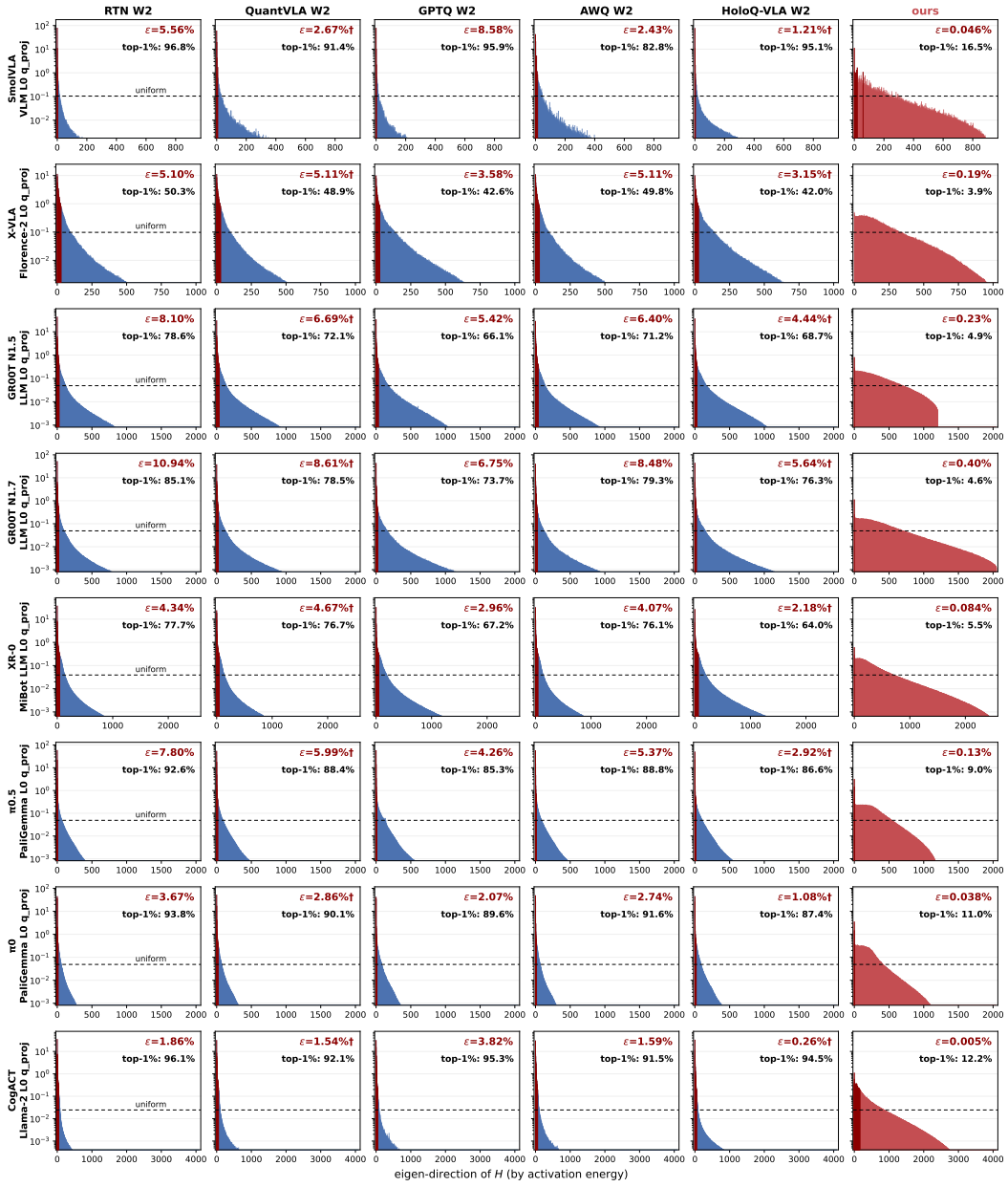


Figure 16: W2 output-error share along the eigen-directions of $H$ at the first language-backbone `q_proj` of eight VLAs.

# F THEORETICAL ANALYSIS

Propositions state mathematical consequences under the specified assumptions, whereas empirical claims summarize experimental observations. Notation follows the main text ($f_t$: predicted field of the FP16 head, $\hat{f}_t$: predicted field of the quantized head).

## F.1 ACTIVATION-EIGENDIRECTION DECOMPOSITION

**Assumption 1.** $x \in \mathbb{R}^d$ has a finite second moment, with $H = \mathbb{E}[xx^\top] \succeq 0$ and $\mathrm{tr}(WHW^\top) > 0$. For $H = U\Lambda U^\top$, let $a_k = \|\Delta W u_k\|_2^2$.

**Proposition 1 (Output-error alignment).** With $\Lambda_H = \sum_k \lambda_k$ and $A = \sum_k a_k$,

$$\mathrm{tr}(\Delta W H \Delta W^\top) = \sum_k \lambda_k a_k = \frac{\Lambda_H A}{d} + d\,\mathrm{Cov}_u(\lambda_k, a_k), \qquad \|\Delta W\|_F^2 = \sum_k a_k. \tag{10}$$

The quantities are proportional with the same constant for all $\Delta W$ if and only if $H = cI$. Suppose $A > 0$, $\sum_k \lambda_k a_k > 0$, and sums of $a_k$ over $\mathcal{K}$ and its complement are both positive. For a proper subset $\mathcal{K}$ of high-activation-energy directions, define the Frobenius error fraction $\alpha = \sum_{k\in\mathcal{K}} a_k / A$ and output error fraction $\kappa = \alpha\bar{\lambda}_{\mathcal{K}}^{(a)} / \{\alpha\bar{\lambda}_{\mathcal{K}}^{(a)} + (1-\alpha)\bar{\lambda}_{\overline{\mathcal{K}}}^{(a)}\}$, where $\bar{\lambda}_{\mathcal{A}}^{(a)} = \sum_{k\in\mathcal{A}} \lambda_k a_k / \sum_{k\in\mathcal{A}} a_k$. Then, for $0 < \tau < 1$,

$$\kappa \geq \tau \iff (1-\tau)\sum_{k\in\mathcal{K}} \lambda_k a_k \;\geq\; \tau \sum_{k\notin\mathcal{K}} \lambda_k a_k. \tag{11}$$

Thus, the difference between the two fractions depends on the relative magnitudes of the eigenvalues weighted by $a_k$, and $\kappa \geq \alpha$ when $\mathcal{K}$ is a set of the largest eigenvalues.

*Proof.* Substituting $H = U\Lambda U^\top$ and using trace cyclicity gives the first identity; expanding the covariance gives the second term. For the converse yielding $H = cI$, choose $\Delta W$ with one nonzero row $v^\top$ and use $v^\top H v = c\|v\|_2^2$ for all $v$. Multiplying both sides by the positive denominator and rearranging gives the inequality for $\kappa$.

**Empirical claim 1.** For the GR00T N1.5 head (Figure 3), the four scalar W2 methods spread weight error nearly uniformly or toward low-energy eigendirections (0.4–1.0% in the top 1%), so their output-error share $\kappa$ (34.3–46.1%) follows the activation-energy share (45.6%); error-redistributed W2 lowers these to 0.005% and 3.1%. This offline W2A16 observation (Figures 3, 15, 16) is empirical rather than implied by the identity; its relationship to closed-loop performance is evaluated in Appendix E.1.

## F.2 ROTATION AND GAUSSIAN EXTREME VALUES

**Assumption 2.** Let $g_i \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ and $M_N = \max_i |g_i|$, and assume the centered RMS denominator in Eq. (3) is positive for all matrices considered. Assume the module medians before and after rotation, $\widetilde{\rho}_m, \widetilde{\rho}'_m$, and the Gaussian reference median $\widetilde{\beta}_m$ satisfy

$$\widetilde{\rho}'_m - \widetilde{\beta}_m = \gamma_m(\widetilde{\rho}_m - \widetilde{\beta}_m) + e_m, \quad 0 \leq \gamma_m < 1, \quad |e_m| < (1-\gamma_m)|\widetilde{\rho}_m - \widetilde{\beta}_m| \tag{12}$$

for modules with $D_{\mathrm{rot}}(m) \neq 1$ (no conclusion is drawn when $D_{\mathrm{rot}}(m) = 1$). Empirical claim 2 tests the direction predicted by this conditional contraction.

**Proposition 2 (Gaussian reference and conditional direction).** $\mathbb{E}[M_N] \leq \sqrt{2\ln(2N)}$, and the centered, sample-RMS-normalized statistic $R_N = \max_i |g_i - \bar{g}| / \{N^{-1}\sum_i (g_i - \bar{g})^2\}^{1/2}$ also satisfies $R_N = \sqrt{2\ln(2N)} + o_p(\sqrt{\ln N})$. Thus, $\beta_\ell = \sqrt{2\ln(2N_\ell)}$ is a leading-order reference, not the exact expectation for finite $N_\ell$. Under Assumption 2,

$$D_{\mathrm{rot}}(m) > 1 \Rightarrow \widetilde{\rho}'_m < \widetilde{\rho}_m, \qquad D_{\mathrm{rot}}(m) < 1 \Rightarrow \widetilde{\rho}'_m > \widetilde{\rho}_m. \tag{13}$$

*Proof.* Applying Jensen's inequality to $\exp(\lambda M_N) \le \sum_i \{\exp(\lambda g_i) + \exp(-\lambda g_i)\}$ gives $\mathbb{E}[M_N] \le \ln(2N)/\lambda + \lambda/2$; setting $\lambda = \sqrt{2\ln(2N)}$ yields the upper bound. The standard extreme-value result $M_N = \sqrt{2\ln(2N)} + o_p(\sqrt{\ln N})$ for the maximum of absolute Gaussian variables is not derived here (the lower bound follows from the Gaussian tail). Since $|\max_i |g_i - \bar{g}| - M_N| \le |\bar{g}| = O_p(N^{-1/2})$ and the sample second moment converges to 1, $R_N$ has the same leading term. The final implication follows as the margin condition gives $\widetilde{\rho}'_m - \widetilde{\rho}_m = -(1-\gamma_m)(\widetilde{\rho}_m - \widetilde{\beta}_m) + e_m$ the same sign as $-(\widetilde{\rho}_m - \widetilde{\beta}_m)$, and $\widetilde{\beta}_m > 0$ makes $D_{\mathrm{rot}}(m) \gtrless 1$ equivalent to $\widetilde{\rho}_m \gtrless \widetilde{\beta}_m$.

**Empirical claim 2.** The predicted and observed directions of tail change agree for all 16 modules (eight models $\times$ 2 modules) and 1,217/1,326 matrices, providing empirical support for the rule in Eq. (6). In closed loop, disabling rotation raises SR by $10.4\,pp$ (CogACT) and $5.8\,pp$ (SmolVLA) with $D_{\mathrm{rot}} \le 1$ and lowers it by $5.2\,pp$ for X-VLA with $D_{\mathrm{rot}} > 1$ (Table 5).

### F.3 Gate Cost and Amdahl Bound

**Assumption 3.** $T \in \mathbb{N}_{>0}$, $1 \le \hat{T} \le T$, $h, \rho_{\mathrm{skip}} \in [0,1]$, and $h\rho_{\mathrm{skip}} < 1$. Head evaluation cost $C_{\mathrm{head}} > 0$ is constant, the adaptive gate costs $C_{\mathrm{gate}} \ge 0$ per step, and shared backbone and solver costs are identical across both configurations and additive.

**Proposition 3 (Caching cost condition).** When the number of head evaluations decreases from $T \to \hat{T}$,

$$C_{\mathrm{adapt}} - C_{\mathrm{base}} = -(T - \hat{T})C_{\mathrm{head}} + TC_{\mathrm{gate}}, \tag{14}$$

so Eq. (4) is necessary and sufficient. An additive cost model with head fraction $h$, skip fraction $\rho_{\mathrm{skip}} = 1 - \hat{T}/T$, and additional cost $G \ge 0$ gives $L_1 = L_0(1 - h\rho_{\mathrm{skip}}) + G$ and $L_0/L_1 \le (1 - h\rho_{\mathrm{skip}})^{-1}$.

*Proof.* Subtract $C_{\mathrm{base}} = TC_{\mathrm{head}}$ from $C_{\mathrm{adapt}} = \hat{T}C_{\mathrm{head}} + TC_{\mathrm{gate}}$ and divide by $TC_{\mathrm{head}} > 0$. Removing exactly $h\rho_{\mathrm{skip}}L_0$ gives $L_1$; taking reciprocals with $G \ge 0$ gives the upper bound (equality when $G = 0$).

### F.4 Quantization–Caching Error Decomposition

**Assumption 4.** $f_t$ and $\hat{f}_t = f_t + \delta_t$ are evaluated at the same state, and $k(t)$ is fixed on the same reference trajectory. Differing states add a state-feedback term.

**Proposition 4 (Field-error interaction).** If $a_t = f_{k(t)} - f_t$ and $b_t = \delta_{k(t)}$, then

$$\|\tilde{f}_t - f_t\|_2^2 = \|a_t\|_2^2 + \|b_t\|_2^2 + 2\langle a_t, b_t\rangle, \qquad (\|a_t\|_2 - \|b_t\|_2)^2 \le \|\tilde{f}_t - f_t\|_2^2 \le (\|a_t\|_2 + \|b_t\|_2)^2, \tag{15}$$

and the squared-field-error factorial interaction is $I_t^{\mathrm{field}} = 2\langle a_t, \delta_{k(t)}\rangle + \|\delta_{k(t)}\|_2^2 - \|\delta_t\|_2^2$. It reduces to the cross term only when the last two norms are equal.

*Proof.* Expanding $\|a_t + b_t\|_2^2$ gives the first expression; Cauchy–Schwarz gives the inequality. The factorial expression is the difference-in-differences $E_{11} - E_{10} - E_{01} + E_{00}$ of squared errors across four configurations: $E_{00} = 0$, $E_{10} = \|\delta_t\|_2^2$, $E_{01} = \|a_t\|_2^2$, and $E_{11} = \|a_t + \delta_{k(t)}\|_2^2$.

**Empirical claim 3.** The accuracy interaction $I_m = S_m(1,1) - S_m(1,0) - S_m(0,1) + S_m(0,0)$ of the success probability $S_m(q,c)$ includes the nonlinear solver and success criterion, so it differs from $I_t^{\mathrm{field}}$ and requires evaluating all four configurations; the local identity in Eq. (5) alone cannot predict closed-loop degradation.

### F.5 Latency Product Law

**Assumption 5.** Measurements for $N_0$ without caching and $N_1 < N_0$ with caching use the same device, capture mode, and configuration.

**Proposition 5 (Exact product and sign criterion).**

$$G_{\mathrm{QC}} = \frac{L_{\mathrm{fp}}(N_0)}{L_q(N_1)} = \underbrace{\frac{L_{\mathrm{fp}}(N_0)}{L_{\mathrm{fp}}(N_1)}}_{G_{\mathrm{C}}} \underbrace{\frac{L_{\mathrm{fp}}(N_1)}{L_q(N_1)}}_{G_{\mathrm{Q}}(N_1)}, \qquad E_{\mathrm{naive}} = \frac{G_{\mathrm{Q}}(N_0)}{G_{\mathrm{Q}}(N_1)} - 1. \tag{16}$$

If $L_{\mathrm{fp}}(N) = a + bN$ and $L_q(N) = a' + b'N$, then $E_{\mathrm{naive}} = (N_0 - N_1)(ba' - b'a)/\{(a' + b'N_0)(a + bN_1)\}$, so when all coefficients are positive, $E_{\mathrm{naive}} > 0 \iff b'/b < a'/a$.

*Proof.* Canceling $L_{\mathrm{fp}}(N_1)$ gives the first identity. Substituting the linear expressions and canceling common terms gives the numerator $(N_0 - N_1)(ba' - b'a)$.

### F.6 Rank-One Folding of Activation Scales

**Assumption 6.** For layer $\ell$ with $\|S_\ell\|_F > 0$, write the scale table $S_\ell = c_\ell g_\ell^\top + E_\ell$ (first singular component and residual), let $\eta_\ell = \|E_\ell\|_F / \|S_\ell\|_F$, and fix the integer activation code $q_s$.

**Proposition 6 (Folded form and residual).**

$$W \operatorname{diag}(s_{\ell s,:}) q_s = c_{\ell s} W_{\mathrm{fold}} q_s + e_{\ell s}, \quad W_{\mathrm{fold}} = W \operatorname{diag}(g_\ell), \quad \|e_{\ell s}\|_2 \le \|W\|_2 \|E_{\ell s,:}\|_2 \|q_s\|_\infty. \tag{17}$$

Thus, $g_\ell$ can be folded into the weights offline, leaving only $c_{\ell s}$ as one scalar per layer and denoising step; the expression bounds the omitted term.

*Proof.* Substituting $s_{\ell s,:} = c_{\ell s} g_\ell^\top + E_{\ell s,:}$ gives the decomposition, and applying the operator norm inequality to the residual gives the upper bound.

**Scope.** Over the 96 layers of GR00T N1.5, mean $\eta_\ell$ is $0.0411$ and the leading singular component accounts for 99.77% of the energy; a rank-1 activation table changes closed-loop SR by $-0.8$–$+0.4$ *pp* versus the channel-wise table.

### F.7 High-Resolution Lattice-Gain Reference

In the high-resolution limit, per-coordinate distortion is $D \simeq G(\Lambda) V^{2/d}$. Comparing the $\mathbb{Z}^8$ product lattice and $E_8$ at equal cell volume, using the established normalized second moments $G(\mathbb{Z}^8) = 1/12$ and $G(E_8) = 929/12960$, therefore gives

$$\frac{D_{\mathbb{Z}}}{D_{E_8}} \to \frac{1080}{929} = 1.1625 \quad (0.654\ \mathrm{dB}) \tag{18}$$

(MSE $-13.98\%$, RMSE $-7.25\%$ versus the scalar reference). This established high-resolution ratio serves only as a reference for using a lattice in $H$-weighted error redistribution.

### F.8 Binomial Uncertainty and Ceiling Headroom

**Assumption 7.** Episode success outcomes follow $Y_i \overset{\mathrm{iid}}{\sim} \mathrm{Bernoulli}(p)$ within each condition and are independent across conditions; paired seeds or shared episodes require an added covariance term.

**Proposition 7 (Variance and ceiling bound).** $\mathrm{Var}(\hat{p}) = p(1-p)/n$, and the accuracy interaction across four configurations, $\hat{I}_m = \hat{p}_{11} - \hat{p}_{10} - \hat{p}_{01} + \hat{p}_{00}$, has variance $\mathrm{Var}(\hat{I}_m) = \sum_{q,c} p_{qc}(1 - p_{qc})/n_{qc}$. If all four probabilities lie in $[1 - \eta, 1]$, then $|I_m| \le 2\eta$. For $\sigma_I^2 = \mathrm{Var}(\hat{I}_m) > 0$ and $z = z_{1-\alpha/2}$, under alternatives satisfying $2\eta \le z\sigma_I$, two-sided normal-approximation test power using the true standard error $\sigma_I$ is at most $\frac{1}{2} + \Phi(-2z)$ (not finite-sample exact binomial power).

*Proof.* The variance expressions follow from Bernoulli variance and the variance of a linear combination of independent variables. Each term in $I_m = (p_{11} - p_{10}) + (p_{00} - p_{01})$ has absolute value at most $\eta$, giving $|I_m| \le 2\eta$. The upper bound follows from $\delta = |I_m|/\sigma_I \le z$ and monotonicity of the power $1 - \Phi(z - \delta) + \Phi(-z - \delta)$.

**Scope.** We use the Newcombe interval for the difference between two proportions, and the main text reports $\hat{I}_m$ as the effect size.

## G LIMITATIONS

Our results hold under the following conditions: (i) latency gains are measured at batch 1 on bandwidth-limited devices (RTX 3090, RTX 4090, MacBook; Section 5.4); (ii) success rates are evaluated using fake quantization, the reported $79.6\%$ reduction refers to nominal weight traffic, and the packed kernel expands head weights to INT4 at load time (Tables 15, 7); (iii) compute-node sets are compared in closed loop on GR00T N1.5 (Appendix D.4); (iv) head precision is evaluated down to W2; (v) all evaluations use LIBERO, LIBERO-Plus, and SimplerEnv simulations; (vi) per-benchmark rotation ON/OFF comparisons use XR-0 and $\pi_0$ ($D_{\mathrm{rot}} \leq 1$; Appendix E.3); and (vii) each model's compute-node set is a static schedule that does not adapt to task difficulty.