QUOVLA: QUOTIENT SPACE FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models adapt pretrained visual-language features for robot control through action supervision. Action supervision can guide task-specific feature learning and the suppression of action-irrelevant variation. To limit this variation, we introduce QuoVLA, which combines joint feature learning with a quantized prefix adapter. The adapter processes visual features and language embeddings before a shared language backbone and action expert, using flow matching and relative temporal-complexity regularization. This regularization uses a stop-gradient raw-prefix reference to discourage excess temporal variation in the flow predictions of the learned policy. We interpret this policy through an action quotient that groups representations with the same chosen optimal action law. For this fixed action law, a conditional characterization defines an ideal compression target without assuming sufficient pretrained features or guaranteeing that training attains the target. The target motivates paired representation diagnostics alongside task success evaluation. Task success improves over the primary baseline on four simulation benchmarks and four real-robot tasks, while the diagnostics support selective compression under the evaluated perturbations.

1 INTRODUCTION

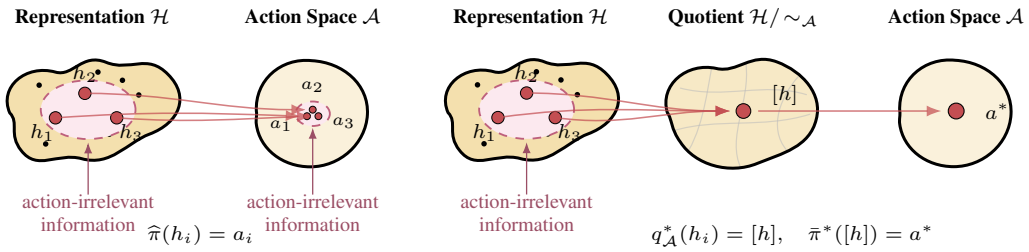
Pretrained Vision-Language Models (VLMs) provide visual and linguistic features for robot control (Radford et al., 2021; Liu et al., 2023b). Vision-Language-Action (VLA) models adapt these features to predict actions from observations and instructions (Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025). Effective control requires preserving distinctions that change the required action while limiting sensitivity to changes that leave it unchanged.

Existing approaches fine-tune pretrained backbones with action supervision (Zitkovich et al., 2023; Kim et al., 2024; Zhang et al., 2026), or limit action gradients to reduce interference with visual-language learning (Driess et al., 2025; Intelligence et al., 2026). A complementary question is which distinctions an action representation should preserve: changes in appearance or wording may leave the required action unchanged, whereas changes in object position or task goals may require different actions. This motivates selective compression guided by action supervision.

We formalize selective compression through an *ideal action quotient*, which groups representations with the same chosen Bayes-optimal action law. Under sufficiency for this law, the quotient is the coarsest representation that preserves it (Figure 1). Coarseness here concerns which distinctions are retained: variation within an action-equivalence class can be removed, while classes with different optimal action laws remain distinguishable. Demonstrations supervise actions without directly labeling these ideal classes. Since the classes depend on the unknown optimal action law, the quotient defines an idealized compression target.

QuoVLA uses quantization as a *practical approximation strategy toward this ideal target*. We jointly train the VLM, a quantized prefix adapter, and the action expert, allowing action supervision to guide which information is retained or discarded. The adapter acts before the shared language backbone, and its gated surrogate gradient lets action supervision update the upstream features. No nuisance coordinates or desired equivalence classes are specified in advance. Quantization limits feature precision but does not identify the ideal action-equivalence classes. Thus, we aim to approximate the quotient's selective compression, without claiming that the learned quantized representation realizes the ideal quotient.

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



(a) A possible learned action mapping.

(b) The ideal action quotient.

Figure 1: **The action-quotient principle.** For illustration, the optimal action law is deterministic and the marked representations share the same optimal action a^*. (a) A learned decoder may produce different predictions from these representations; this is a possible behavior, not a consequence of representation injectivity alone. (b) The ideal quotient merges representations with the same optimal action law. The diagram describes the conditional theoretical target; it does not establish that the learned quantizer realizes these action-equivalence classes from finite training demonstrations.

To support this approximation, a relative temporal-complexity (RTC) penalty compares the adapted route with a raw-prefix reference whose gradients are stopped. The two routes share the language backbone and action expert and use the same noisy action input and flow time. Consequently, the raw route tracks the current model as the shared parameters are updated. RTC discourages excess variation in the flow-velocity sequence across the predicted action chunk, while flow matching supplies action supervision. This regularizer encourages temporal consistency without guaranteeing action sufficiency.

We evaluate QuoVLA on LIBERO, LIBERO-PRO, LIBERO-Plus, RoboTwin 2.0, and four real-robot tasks. Task success, paired representation diagnostics, and alternative controls assess the utility of the method and whether the learned approximation exhibits selective compression under the tested perturbations, without treating these diagnostics as proof of quotient recovery.

Our contributions cover the ideal target, its practical approximation, and empirical evaluation:

- We characterize an ideal action quotient and give sufficient conditions under which a learned representation lowers expected squared test error.
- We introduce QuoVLA as a practical approximation strategy, combining a jointly trained quantized prefix adapter, gated surrogate gradients, and relative temporal-complexity regularization.
- We evaluate task success in simulation and on real robots, and use paired diagnostics and alternative controls in simulation to assess selective compression.

2 Related Work

Adapting pretrained VLMs for action generation. VLA methods connect pretrained visual-language features to control through several prediction interfaces. RT-2 and OpenVLA discretize actions and fine-tune autoregressive models with action supervision (Zitkovich et al., 2023; Kim et al., 2024). UniVLA studies a shared tokenized interface (Bu et al., 2025). Other methods use policy heads or continuous action experts, including RoboFlamingo, π_0, and $\pi_{0.5}$ (Wang, 2025; Black et al., 2024; Intelligence et al., 2025). These choices determine the form of the action output and the interface through which pretrained features support control. The information carried by those features remains a separate design question. Action supervision adapts pretrained features and action predictors to the distinctions needed for control.

Controlling interference from action learning. Compact policy decoders and adapters provide alternative interfaces between pretrained features and robot actions (Wen et al., 2025; Wang et al., 2026). Knowledge-insulated designs separate learning signals to preserve visual-language capabilities while training continuous action experts (Driess et al., 2025; Intelligence et al., 2026). These methods address gradient interference and knowledge retention. Preserving visual-language knowledge concerns the effect of action learning on the backbone, whereas selective compression concerns the distinctions

supplied to the action predictor. QuoVLA jointly fine-tunes the VLM, prefix adapter, and action expert while limiting prefix precision and regularizing the temporal variation of action predictions.

Representation regularization and quantization. RS-CL guides representation learning using relative proprioceptive-state distances in a contrastive objective (Kim et al., 2025b). VQ-VLA learns a vector-quantized tokenizer for action sequences (Wang et al., 2025). These two directions structure either the relationships between representations or the representation of actions themselves. QuoVLA instead applies scalar quantization to conditioning-prefix features and trains a continuous action expert, with a raw-prefix reference for temporal regularization. Its learned compression aims to approximate the ideal action quotient, without claiming that quantization recovers the ideal action-equivalence classes. These methods share the aim of structuring action learning but differ in the object they quantize or regularize; we do not claim a head-to-head advantage over them.

Novelty. QuoVLA jointly learns action features and their compression through a quantized prefix adapter and relative temporal-complexity regularization, allowing action supervision to guide which information is discarded. We characterize the ideal action quotient and derive sufficient conditions for a learned quotient to achieve lower expected squared error on unseen samples.

3 Action-Quotient Perspective

3.1 Preliminary

VLA formulation. Let $P_{X,A}$ be a distribution of visual-language inputs $X = (o_{1:K}, \ell) \in \mathcal{X}_{\mathrm{VL}}$ and expert action trajectories $A_{1:T} \in \mathcal{A}_{\mathrm{seq}} := \mathcal{A}^T$. Here ℓ denotes the complete conditioning prompt, including the instruction and any state tokens used by the policy. A representation map h_θ and a downstream action predictor define the policy

$$H = h_\theta(X), \qquad \pi_{\theta,\phi}(\cdot \mid x) = G_\phi^{\mathrm{act}}(\cdot \mid h_\theta(x)). \tag{1}$$

This abstract factorization applies to embedding-stage or contextual features. Sufficiency must be evaluated on the complete conditioned input, including any additional state, and need not hold for the implemented adapter without a separate sufficiency assumption.

Role of the representation. The representation h_θ may be learned with action supervision. The minimal-quotient analysis below considers the conditional case in which this representation is sufficient for a selected optimal action law. Neither pretrained initialization nor subsequent training is assumed to guarantee this condition. Finite-context injectivity results (Nikolaou et al., 2026) do not verify it for our trained VLA model (Appendix A.2).

3.2 Minimal VLM-Action Quotient

For a loss ℓ_{law} on predicted action laws, assume a measurable Bayes-optimal selection exists and fix

$$p_X^\star(\cdot \mid x) \in \arg\min_{\mu \in \Delta(\mathcal{A}_{\mathrm{seq}})} \mathbb{E}[\ell_{\mathrm{law}}(\mu, A_{1:T}) \mid X = x]. \tag{2}$$

If the minimizer is not unique, all subsequent equivalence and minimality claims refer to this fixed selection. For the ideal characterization below, consider a representation h_θ, possibly learned with action supervision, that is sufficient for this selection: there exists a measurable map $p_H^\star$ such that

$$p_X^\star(\cdot \mid x) = p_H^\star(\cdot \mid h_\theta(x)) \quad \text{for } P_X\text{-almost every } x. \tag{3}$$

This condition is relative to the supplied input and distribution; it is not a claim about pretrained features or a guarantee of training. Compression cannot restore distinctions already lost by a fixed representation, supply state information absent from X, or ensure that a particular action head can recover the selected law from the compressed representation alone.

Choose a full-P_X-measure set $\mathcal{X}_0$ on which Eq. (3) holds, and write $\mathcal{H}_0 = h_\theta(\mathcal{X}_0)$. Define

$$h \sim_{\mathcal{A}} h' \quad \Longleftrightarrow \quad p_H^\star(\cdot \mid h) = p_H^\star(\cdot \mid h'), \qquad h, h' \in \mathcal{H}_0. \tag{4}$$

The equivalence classes collect latents with the same selected optimal action law.

Theorem 1 (Minimal VLM-Action quotient). *Under Eq.* (3)*, let* $q^\star_{\mathcal{A}}(h) = [h]_{\sim_{\mathcal{A}}}$ *and* $\mathcal{H}^\star_{\mathcal{A}} = \mathcal{H}_0/\sim_{\mathcal{A}}$*. There exists a decoder* $\bar{p}^\star_{\mathcal{A}}$ *on the quotient satisfying*

$$p^\star_H(\cdot \mid h) = \bar{p}^\star_{\mathcal{A}}(q^\star_{\mathcal{A}}(h)), \qquad h \in \mathcal{H}_0. \tag{5}$$

Moreover, if a representation s *admits a decoder* $\bar{p}_s$ *with* $p^\star_H(\cdot \mid h) = \bar{p}_s(s(h))$ *on* $\mathcal{H}_0$*, then* $q^\star_{\mathcal{A}} = \rho \circ s$ *for some map* ρ*. Thus* $q^\star_{\mathcal{A}}$ *is the coarsest partition that preserves the selected law, uniquely up to a relabeling of these equivalence classes on the representation domain* $\mathcal{H}_0$*.*

The proof and a constructed example with nontrivial equivalence classes appear in Appendix A.3. The characterization defines an ideal target, without asserting identifiability from finite demonstrations or realizability by the learned bottleneck.

3.3 Generalization Comparison

Let S_n collect n training demonstrations and all training randomness, and write the complete QuoVLA policy as $f_Q = F_{S_n}(Z_{S_n}(X), \xi)$, where Z_{S_n} contains the actual dequantized prefix and all input-dependent side conditions. Let f_H denote a trained baseline and $\mathcal{R}_{j,n} = \mathbb{E}\|f_j - A\|^2_{\mathrm{A}}$, $j \in \{H, Q\}$, their independent-test action risks. Here $\|a\|^2_{\mathrm{A}} = \|a\|^2_F/(Td_a)$ for T actions with d_a evaluated coordinates. Expectations include training and inference randomness. The learned interface defines equivalence classes by equal Z_{S_n}; these classes need not preserve all action information.

Theorem 2 (Risk comparison for a learned quotient). *Assume measurable learned maps on standard Borel spaces, square-integrable actions and predictions, a common test pair* (X, A) *independent of* S_n*, and inference noise* ξ *independent of* (S_n, X, A)*. Define* V_n, I_n, D_n, B_n *as in Eq.* (25) *of Appendix A.4. The expected independent-test action risks then satisfy*

$$\mathcal{R}_{H,n} - \mathcal{R}_{Q,n} \geq V_n - 2\sqrt{V_n I_n} - D_n - 2\sqrt{B_n D_n}. \tag{6}$$

Thus $\mathcal{R}_{Q,n} < \mathcal{R}_{H,n}$ *whenever the right-hand side is positive. If* $I_n = D_n = 0$*, then* $\mathcal{R}_{Q,n} = \mathcal{R}_{H,n} - V_n$*.*

Here V_n is baseline variation within a learned class; I_n is discarded conditional-mean action information; D_n is QuoVLA's deviation from the baseline's class average; and B_n is that average's error relative to the class-averaged conditional action mean. Class averages condition on S_n and ξ. The bound balances removed variation against information loss and prediction deviation. Appendix A.4 gives the exact definitions, proof, and discussion of the assumptions; quantization and RTC do not by themselves guarantee a positive bound.

4 Methodology

4.1 Transformer-based Prefix Quantization

QuoVLA combines action-supervised representation learning with a quantized prefix bottleneck. In the supplied implementation, this bottleneck operates before the PaliGemma language backbone. For input $x = (o, \ell)$, the image encoder and multimodal projector produce visual embeddings, while the language embedding layer embeds the complete prompt. The prompt includes the instruction and discretized robot state; state conditioning therefore enters through the prefix. Their concatenation is

$$E = e_\eta(o, \ell) \in \mathbb{R}^{M \times d}, \tag{7}$$

where M is the prefix length. These are embedding-stage tokens; the contextual language backbone processes them after the adapter. The encoder and adapter jointly learn which input distinctions the quantized prefix retains; no discarded nuisance variables or equivalence classes are prescribed. This placement is motivated by the ideal action quotient.

The adapter contains L_q pre-normalized Transformer layers. Starting from $Z_0 = E$, layer j computes

$$U_j = Z_{j-1} + \mathrm{MHA}_j\big(\mathrm{LN}^{\mathrm{att}}_j(Z_{j-1})\big), \tag{8}$$

$$Z_j = U_j + W_{2,j}\, \mathcal{Q}_b\Big(\mathrm{SiLU}\Big(W_{1,j}\mathrm{LN}^{\mathrm{ff}}_j(U_j) + c_{1,j}\Big)\Big) + c_{2,j}. \tag{9}$$

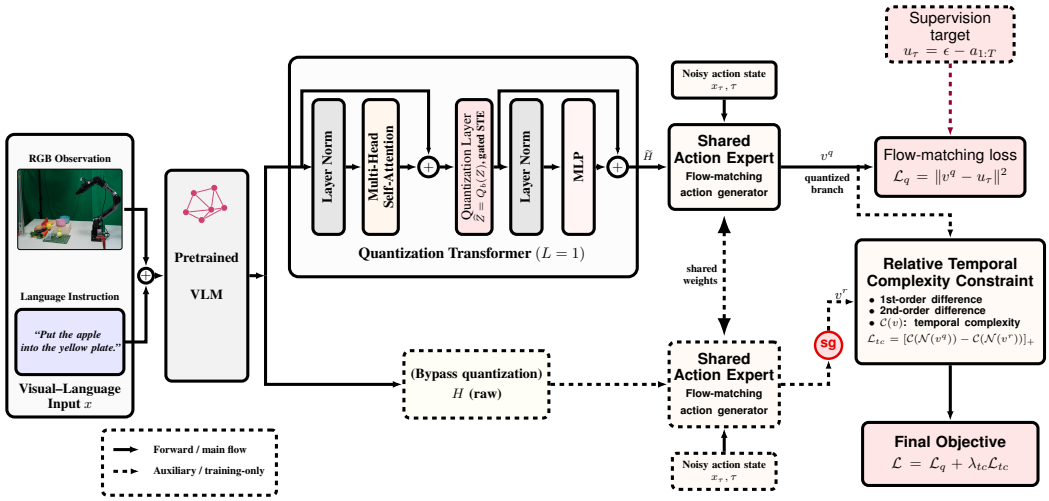


Figure 2: **Overview of the QuoVLA architecture.** This conceptual overview shows the quantized and raw-prefix routes. The circled sg (stop-gradient) blocks RTC gradients through the raw route; shared parameters are updated through the main route. See Section 4.1 for the two quantization steps within the feed-forward sublayer and at the adapter output.

Thus, quantization is applied inside the feed-forward sublayer, after its activation and before its second linear map. The adapter adds an outer residual connection and applies a second quantization step:

$$E^q = \mathcal{Q}_b\big(\mathrm{LN}_{\mathrm{out}}(E + Z_{L_q})\big)\,. \tag{10}$$

Prefix padding is masked in attention. The zero-layer variant uses $E^q = \mathcal{Q}_b(\mathrm{LN}_{\mathrm{out}}(E))$.

For each token vector z and $2 \le b < 16$, symmetric quantization and dequantization give

$$\begin{aligned} q_{\max} &= 2^{b-1} - 1, \qquad s(z) = \frac{\max(\|z\|_\infty, \epsilon_q)}{q_{\max}}, \\ Q_b(z) &= s(z)\,\mathrm{clip}\left(\mathrm{round}\left(\frac{z}{s(z)}\right), -q_{\max}, q_{\max}\right). \end{aligned} \tag{11}$$

The subsequent layers receive dequantized floating-point activations. Because the scale depends on the token, equal integer codes need not yield equal dequantized representations. The quantizer therefore permits information loss without guaranteeing a finite set of output representations or identifying action-equivalence classes. Historical unquantized controls and reference-code bypass settings are distinguished in Appendix A.10.

We use a gated straight-through estimator for optimization (Bengio et al., 2013):

$$g = g_{\min} + (1 - g_{\min})\sigma(\alpha), \tag{12}$$

$$\mathcal{Q}_b(z) = \mathrm{sg}(Q_b(z) - gz) + gz, \tag{13}$$

where sg denotes stop-gradient and α is trainable. The forward value is $Q_b(z)$, but the surrogate Jacobian is gI, stopping derivatives through rounding and the scale. The gate affects upstream operations, including trainable image and token embeddings; downstream backbone and expert parameters receive their usual gradients. Setting $g = 1$ gives standard STE. The reported learned gate reaches approximately 0.35; this endpoint does not establish an advantage over fixed gates or quantify capacity.

4.2 Dual-Branch Action Generation

Let $\mathcal{G}_\omega$ denote the shared PaliGemma language backbone and action expert, including their joint attention computation. The main route supplies the adapted prefix E^q; the reference route supplies E and bypasses the entire adapter. The two routes therefore differ in prefix processing, including both the added Transformer and quantization.

For a demonstrated action chunk $a_{1:T}$, we sample noise ϵ and flow time τ, then set

$$x_\tau = \tau\epsilon + (1 - \tau)a_{1:T}, \qquad u_\tau = \epsilon - a_{1:T}. \tag{14}$$

Table 1: Results on LIBERO and LIBERO-PRO. Left: LIBERO success rates (%), with best/second-best marked by **bold**/underline; right: LIBERO-PRO robustness success rates (%), with best in **bold**.

(a) LIBERO benchmark

Model	Spatial	Object	Goal	Long	Avg
Diffusion Policy (Chi et al., 2025)	78.5	87.5	73.5	64.8	76.1
OpenVLA (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5
SpatialVLA (Qu et al., 2025)	88.2	89.9	78.6	55.5	78.1
CoT-VLA (Zhao et al., 2025)	87.5	91.6	87.6	69.0	83.9
UniVLA (Bu et al., 2025)	96.5	96.8	95.6	92.0	95.2
OpenVLA-OFT (Kim et al., 2025a)	97.6	98.4	97.9	94.5	97.1
MemoryVLA (Shi et al., 2025)	98.4	98.4	96.4	93.4	96.7
π_0 (Black et al., 2024)	96.8	98.8	95.8	85.2	94.2
$\pi_{0.5}$ (Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9
VLA-Adapter (Wang et al., 2026)	97.8	99.2	97.2	95.0	97.3
ABot-M0 (Yang et al., 2026)	98.8	99.8	99.0	96.6	98.6
Being-H0.5 (Luo et al., 2026a)	97.0	98.2	99.0	96.2	97.6
FLOWER (Reuss et al., 2025)	97.1	96.7	95.6	93.5	95.7
X-VLA (Zheng et al., 2025)	98.2	98.6	97.8	97.6	98.1
VOTE (Lin et al., 2025)	98.8	99.8	97.6	95.6	98.0
SimVLA (Luo et al., 2026b)	99.6	99.8	98.6	96.4	98.6
QuoVLA	**99.8**	**99.9**	**100.0**	**98.7**	**99.6**

(b) LIBERO-PRO benchmark

Suite	Model	Ori	Obj	Pos	Sem	Task
Spatial	OpenVLA (Kim et al., 2024)	98.0	97.0	0.0	97.0	0.0
	$\pi_{0.5}$ (Intelligence et al., 2025)	98.0	97.0	20.0	97.0	1.0
	SimVLA (Luo et al., 2026b)	99.0	98.0	29.0	98.0	0.0
	QuoVLA	**100.0**	**99.0**	**40.0**	**99.0**	**10.0**
Object	OpenVLA (Kim et al., 2024)	99.0	**98.0**	0.0	98.0	0.0
	$\pi_{0.5}$ (Intelligence et al., 2025)	98.0	**98.0**	17.0	96.0	1.0
	SimVLA (Luo et al., 2026b)	**100.0**	85.0	1.0	**100.0**	4.0
	QuoVLA	**100.0**	87.0	**26.0**	**100.0**	**26.0**
Goal	OpenVLA (Kim et al., 2024)	98.0	96.0	0.0	98.0	0.0
	$\pi_{0.5}$ (Intelligence et al., 2025)	97.0	**97.0**	38.0	97.0	0.0
	SimVLA	99.0	82.0	0.0	99.0	10.0
	QuoVLA	**100.0**	85.0	**57.0**	**100.0**	**42.0**
Long	OpenVLA (Kim et al., 2024)	93.0	81.0	0.0	96.0	0.0
	$\pi_{0.5}$ (Intelligence et al., 2025)	93.0	**92.0**	8.0	93.0	1.0
	SimVLA (Luo et al., 2026b)	96.0	61.0	3.0	98.0	10.0
	QuoVLA	**99.0**	82.0	**21.0**	**99.0**	**24.0**

Using the same x_τ, τ, and noise in both routes, we compute

$$v^q = \mathcal{G}_\omega(x_\tau, \tau \mid E^q), \tag{15}$$

$$v^r = \mathrm{sg}[\mathcal{G}_\omega(x_\tau, \tau \mid E)], \tag{16}$$

$$\mathcal{L}_q = \frac{1}{TD}\|(v^q - u_\tau)_{:,1:D}\|_F^2, \tag{17}$$

where D is the original action dimension: the policy removes padded dimensions before reducing the training loss. The reference is evaluated without gradients and has no separate supervised loss. Its parameters are shared with the main route, so it follows the current model rather than a fixed teacher.

For the D valid action dimensions, we regularize temporal variation in the predicted flow velocity via

$$\mathcal{C}(v) = \frac{\lambda_1}{(T-1)D}\sum_{t=1}^{T-1}\|v_{t+1} - v_t\|_2^2 + \frac{\lambda_2}{(T-2)D}\sum_{t=1}^{T-2}\|v_{t+2} - 2v_{t+1} + v_t\|_2^2. \tag{18}$$

Unavailable difference terms are zero for chunks shorter than the required order. Let $\mathcal{N}$ denote optional per-dimension scaling and masking before these differences. The reference code uses $\mathcal{N}(v) = v$ unless external action standard deviations or a dimension mask are supplied. The relative temporal-complexity loss compares the two routes as follows:

$$\mathcal{L}_{\mathrm{tc}} = \left[\mathcal{C}(\mathcal{N}(v^q)) - \mathcal{C}(\mathcal{N}(v^r)) - \rho\big(\mathcal{C}(\mathcal{N}(v^r)) + \epsilon_c\big)\right]_+, \tag{19}$$

With optional slack ratio ρ (zero by default), the sample-averaged training objective is

$$\mathcal{L} = \lambda_q\mathcal{L}_q + \lambda_{\mathrm{tc}}\mathcal{L}_{\mathrm{tc}}. \tag{20}$$

This is an empirical regularizer on a flow-matching vector field, not a constraint on physical robot velocity. Lower temporal complexity alone does not imply correct actions or action sufficiency; action learning is supervised by $\mathcal{L}_q$. The shared raw route provides a within-sample complexity reference, without isolating changes caused by quantization alone. At inference, only the adapted route is used: the language backbone first computes prefix keys and values, which condition the action expert throughout all steps of flow integration.

5 Experiments

5.1 Experimental Setting

Simulation Benchmarks. We evaluate QuoVLA on four simulation benchmarks: *LIBERO* (Liu et al., 2023a), *LIBERO-PRO* (Zhou et al., 2025), *LIBERO-Plus* (Fei et al., 2025), and *RoboTwin 2.0* (Chen et al., 2025). We use the official success predicate of each environment

Table 2: Comparison on LIBERO-Plus. *Zero-Shot Transfer* directly evaluates LIBERO-trained models on LIBERO-Plus, while *Supervised Fine-Tuning* fine-tunes them on the LIBERO-Plus training split. Best results within each evaluation regime are in **bold**.

Models	Camera	Robot	Language	Light	Background	Noise	Layout	Macro[a]
Zero-Shot Transfer								
OpenVLA (Kim et al., 2024)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	16.3
UniVLA (Bu et al., 2025)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	45.8
OpenVLA-OFT (Kim et al., 2025a)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	71.4
π_0 (Black et al., 2024)	61.0	40.8	63.5	89.3	84.1	80.1	76.4	70.7
$\pi_{0.5}$ (Intelligence et al., 2025)	75.8	79.4	83.3	95.5	95.0	89.6	87.0	86.5
ABot-M0 (Yang et al., 2026)	60.4	67.9	86.4	96.2	91.6	86.4	82.6	81.6
VLA-JEPA (Sun et al., 2026)	63.3	67.1	85.4	95.6	93.6	66.3	85.1	79.5
ACoT-VLA (Zhong et al., 2026)	72.6	82.6	87.5	97.7	**96.5**	87.8	88.1	87.5
Ours	**82.3**	**87.6**	**88.2**	**98.2**	95.9	**90.8**	**89.3**	**90.3**
Supervised Fine-Tuning								
π_0 (Black et al., 2024)	79.6	21.1	72.5	84.7	86.2	68.3	69.4	68.8
$\pi_{0.5}$ (Intelligence et al., 2025)	70.3	41.7	81.1	**97.3**	94.6	71.8	84.9	77.4
ACoT-VLA (Zhong et al., 2026)	96.6	70.4	79.7	95.1	97.1	**95.9**	85.0	88.5
Ours	**98.4**	**73.5**	**83.7**	96.3	**99.0**	94.8	**90.4**	**90.9**

[a]Macro is the unweighted arithmetic mean of the seven displayed category success rates, computed from their displayed values and rounded to one decimal place. It is not a pooled success rate over evaluation episodes; categories receive equal weight regardless of their episode counts. A common aggregation rule does not imply identical training or evaluation protocols across methods.

and report the task success rate averaged over the evaluated tasks within each benchmark suite. These benchmarks provide complementary coverage of standard language-conditioned manipulation, generalization under distribution shifts, fine-grained robustness to nuisance perturbations, and diverse bimanual manipulation. Detailed benchmark suites and perturbations are provided in Appendix A.1.

Real-robot Setup. We further evaluate QuoVLA on four real-robot manipulation tasks: moving a tennis ball from a yellow plate to a blue plate, placing a red cube into a yellow plate, putting an apple on a yellow plate, and removing a cuboid from a blue plate. For each task, we use 100 training demonstrations and train the policy on a single NVIDIA A800 GPU for 5,000 steps.

Implementation Details. QuoVLA follows the $\pi_{0.5}$ setup unless otherwise specified, using the PaliGemma-2.6B VLM and a 300M action expert. Images are resized to 224×224; the policy takes one visual-state-language step and predicts a 50-step action chunk, with states and actions padded to 32 dimensions and quantile-normalized. The default reported adapter uses 8-bit quantization with a one-layer, eight-head Transformer. We fully fine-tune the VLM together with the adapter and action expert. The adapted route receives the supervised loss; the raw route is evaluated without gradients as a complexity reference. Models are optimized with AdamW using a 2.5×10^{-5} learning rate, 0.01 weight decay, 1.0 gradient clipping, and cosine decay with warmup. Our simulation runs use 8 NVIDIA A800-80GB GPUs. We train for approximately 10 epochs on the LIBERO-family datasets and for only 5 epochs on RoboTwin 2.0. Baseline descriptions, evaluation protocols, reference-code defaults, and remaining run-level details are provided in Appendix A.10.

5.2 Main Results

Overall performance. Tables 1, 2, and 3 report strong performance across the evaluated simulation benchmarks. QuoVLA reaches the highest reported average success rate on LIBERO and the best or tied-best result in 17 of the 20 LIBERO-PRO settings. It also performs well on LIBERO-Plus under both zero-shot transfer and supervised fine-tuning. Across all 50 RoboTwin 2.0 tasks, the Hard-setting average increases from 43.84% for $\pi_{0.5}$ to 58.6% for QuoVLA, while the Easy-setting average increases from 42.98% to 45.1%. These results support the empirical utility of the proposed adaptation method. Success rates alone do not identify which representation changes cause the gains or establish that the learned representation realizes the ideal action quotient.

Table 2 shows macro averages of 90.3% and 90.9% for QuoVLA in zero-shot transfer and fine-tuning, versus 86.5% and 77.4% for $\pi_{0.5}$. Fine-tuned Camera and Robot gain 28.1 and 31.8 percentage points, while Light loses 1.0 point. RoboTwin results also vary (Table 3): QuoVLA's Easy average remains below DP3's 55.2%. Among the displayed Hard tasks, Blocks Ranking RGB improves

Table 3: Selected RoboTwin 2.0 tasks. The average is computed over all 50 tasks in Table 9, including those omitted here. Highlighting marks the best result for each task and setting.

Simulation Task	ACT		π_0		$\pi_{0.5}$		DP3		QuoVLA	
	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Adjust Bottle	97%	23%	90%	56%	79%	83%	99%	3%	18%	68%
Beat Block Hammer	56%	3%	43%	21%	63%	50%	72%	8%	20%	42%
Blocks Ranking RGB	1%	0%	19%	5%	43%	35%	3%	0%	50%	62%
Blocks Ranking Size	0%	0%	7%	1%	8%	14%	2%	0%	43%	45%
Click Alarmclock	32%	4%	63%	11%	97%	93%	77%	14%	75%	72%
Click Bell	58%	3%	44%	3%	75%	76%	90%	0%	27%	48%
Dump Bin Bigbin	68%	1%	83%	24%	30%	42%	85%	53%	35%	60%
Grab Roller	94%	25%	96%	80%	90%	89%	98%	2%	14%	86%
Handover Block	42%	0%	45%	8%	18%	19%	70%	0%	45%	50%
				...						
Move Pillbottle Pad	0%	0%	21%	1%	33%	29%	41%	0%	36%	45%
Average (in %)	29.7	1.7	46.4	16.3	42.98	43.84	55.2	5.0	45.1	58.6

Table 4: Real-robot evaluation. Task rows show successful attempts out of 100; the final row gives the mean success rate. Δ is the improvement in percentage points.

Real-robot Task	$\pi_{0.5}$	QuoVLA	Δ (pp)
Move tennis ball from yellow plate to blue plate	93/100	97/100	+4
Pick red cube into yellow plate	48/100	74/100	+26
Put apple on yellow plate	31/100	83/100	+52
Remove cuboid from blue plate	86/100	98/100	+12
Average	64.5%	88%	+23.5

Table 5: Ablations on LIBERO. Reported success-rate point estimates (%). Definitions of the control variants and details of the reported settings appear in Appendix A.6.

Knob	**Value**	**Spatial**	**Object**	**Goal**	**Long**	**Avg**
QuoVLA	Reported default	**99.8**	**99.9**	**100.0**	**98.7**	**99.6**
Quantization parameters	(default: $L_q = 1$, $b_q = 8$, Adaptive STE)					
w/o Quantization	–	98.8	98.2	98.0	92.4	96.85
Quantization depth L_q	$L_q = 1$	99.8	99.9	100.0	98.7	99.6
	$L_q = 2$	96.4	96.1	95.6	94.8	95.7
	$L_q = 6$	93.7	92.9	91.8	90.5	92.2
Quantization bit-width b_q	$b_q = 4$	84.4	87.3	85.0	73.7	82.6
	$b_q = 8$	99.8	99.9	100.0	98.7	99.6
Unquantized + gate	–	98.3	98.8	97.2	94.1	97.1
w/o Adaptive STE	–	98.4	98.5	99.1	97.8	98.45

from 35% to 62% relative to $\pi_{0.5}$, whereas Click Alarmclock falls from 93% to 72%. Aggregate gains therefore coexist with individual category and task regressions.

Real-robot performance. Table 4 reports four real-robot tasks, each using 100 training demonstrations and 5,000 training steps on one NVIDIA A800. QuoVLA achieves 352 successes in 400 test attempts versus 258 for $\pi_{0.5}$: an equal-task mean gain of 23.5 percentage points. The largest increases occur in the apple and red cube tasks, from 31% to 83% and from 48% to 74%, respectively. Tennis-ball transfer and cuboid removal gain 4 and 12 points. All four observed changes contribute positively to the mean, with different magnitudes. Independent reset tests used matched task-wise initial-condition distributions and checkpoints and hyperparameters fixed before evaluation. Corollary 1 gives a one-sided 95% lower confidence bound of 9.9 points on the population mean advantage for these tasks. Qualitative sequences appear in Appendix A.11.

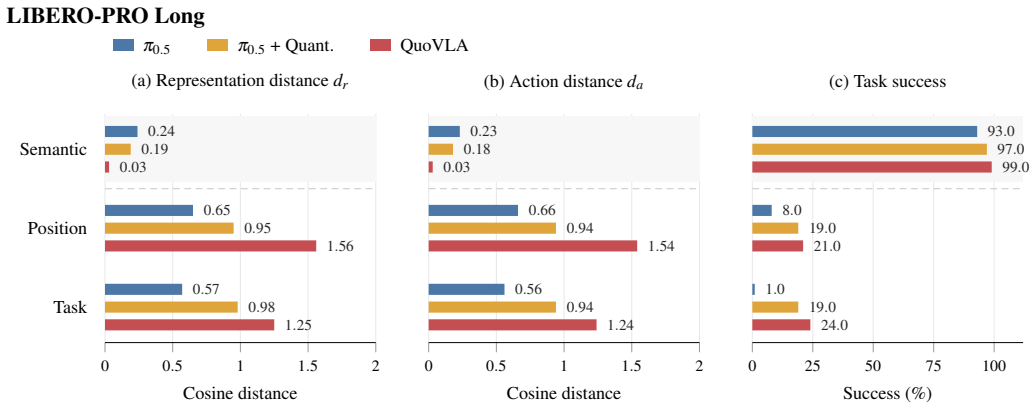


Figure 3: Paired diagnostics on LIBERO-PRO Long. Each group compares $\pi_{0.5}$, a parameter-free 8-bit quantization-only interface, and QuoVLA. The shaded Semantic group is intended to preserve the required behavior; Position and Task require changed actions. QuoVLA contracts representation and action distances for Semantic pairs while separating the action-distinct pairs. Success is shown alongside both distances because distance separation alone does not establish action correctness. All bars are reported point estimates; distances use $1 - \cos$.

5.3 Ablation Study

Component ablations. Table 5 gives the full four-suite results. The default average is 99.6%; increasing adapter depth to two or six layers gives 95.7% or 92.2%, and removing Adaptive STE gives 98.45%. Reducing precision from 8 to 4 bits lowers the average to 82.6%, with Long dropping from 98.7% to 73.7%. The decline also appears in Spatial, Object, and Goal, whose rates fall to 84.4%, 87.3%, and 85.0%, respectively. The observed degradation at 4 bits is therefore spread across all four suites. The unquantized control retains the gate and reaches 97.1%. These comparisons favor the reported 8-bit setting over coarser quantization and the unquantized control; adding layers gives no gain in this configuration. Additional comparisons of control mechanisms and finer-grained sensitivity analyses are provided in Appendices A.5 and A.8.

5.4 Representation Diagnostics

Figure 3 compares $\pi_{0.5}$, parameter-free quantization alone, and QuoVLA using representation and action cosine distances (d_r, d_a) and task success. Semantic is intended to preserve behavior; Position and Task require changed actions. Relative to $\pi_{0.5}$, QuoVLA reduces Semantic distances from $(0.24, 0.23)$ to $(0.03, 0.03)$, while success rises from 93% to 99%. For Position and Task, action distances increase from 0.66 to 1.54 and 0.56 to 1.24, alongside success gains from 8% to 21% and 1% to 24%, respectively. These point estimates support selective compression. Perturbation labels do not verify optimal-action equivalence, cosine distances omit scale, and separation alone establishes neither action correctness nor recovery of the ideal quotient. LIBERO-Plus diagnostics appear in Appendix A.5.

6 Conclusion

QuoVLA jointly learns action features with a quantized prefix adapter, gated surrogate gradients, and RTC. Our analysis characterizes an ideal quotient and shows when the learned representation lowers test prediction error by reducing within-fiber variation while limiting information loss and learning error. Paired diagnostics and controls support selective compression, and a separate certificate establishes a mean success advantage for the evaluated policies on four real-robot task distributions. Verifying the learned-representation risk condition and advantages on untested tasks or across retraining remains open. Detailed limitations are discussed in Appendix A.12.

AI USE STATEMENT

We used generative AI tools to assist with language polishing, including refining sentence structure and improving the clarity and readability of the manuscript.

REFERENCES

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

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.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

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.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025a.

Taeyoung Kim, Jimin Lee, Myungkyu Koo, Dongyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. Contrastive representation regularization for vision-language-action models. *arXiv preprint arXiv:2510.01711*, 2025b.

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Tian Nian, Liuao Pei, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

Juyi Lin, Amir Taherin, Arash Akbari, Arman Akbari, Lei Lu, Guangyu Chen, Taskin Padir, Xiaomeng Yang, Weiwei Chen, Yiqian Li, et al. Vote: vision-language-action optimization with trajectory ensemble voting. *arXiv preprint arXiv:2507.05116*, 2025.

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, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023b.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-h0. 5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026a.

Yuankai Luo, Woping Chen, Tong Liang, Baiqiao Wang, and Zhenguo Li. Simvla: A simple vla baseline for robotic manipulation. *arXiv preprint arXiv:2602.18224*, 2026b.

Giorgos Nikolaou, Tommaso Mencattini, Donato Crisostomi, Andrea Santilli, Yannis Panagakis, and Emanuele Rodolà. Language models are injective and hence invertible. In *The Fourteenth International Conference on Learning Representations*, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. *arXiv preprint arXiv:2509.04996*, 2025.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

Sheng Wang. Roboflamingo-plus: Fusion of depth and rgb perception with vision-language models for enhanced robotic manipulation. *arXiv preprint arXiv:2503.19510*, 2025.

Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. VQ-VLA: Improving vision-language-action models via scaling vector-quantized action tokenizers. *arXiv preprint arXiv:2507.01016*, 2025. URL https://arxiv.org/abs/2507.01016.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pp. 18638–18646, 2026.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. Vlm4vla: Revisiting vision-language-models in vision-language-action models. *arXiv preprint arXiv:2601.03309*, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1702–1713, 2025.

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. *arXiv preprint arXiv:2510.10274*, 2025.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Maoqing Yao, Si Liu, and Guanghui Ren. Acot-vla: Action chain-of-thought for vision-language-action models. *arXiv preprint arXiv:2601.11404*, 2026.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.