QUANTILE HEAD FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models integrate pretrained Vision-Language Models (VLMs) with action heads for robot control. Common action heads have distinct limitations: point regression provides only a point estimate of the action distribution, while standard flow-matching samplers require costly iterative sampling. To address these limitations, we unify regression and flow matching under a shared objective and extend it to derive a quantile objective. This quantile objective guides the design of our Quantile Head, which predicts a median and positive gaps to form ordered marginal action quantiles in one forward pass. These quantiles support multiple sampling strategies without retraining and are jointly supervised to train the default median policy. Our local analysis of this joint supervision shows that, with calibrated nearby quantiles, fixed gaps, and matched correction speed, direct median updates have lower variance than under median-only supervision. Experiments show that this jointly supervised median policy achieves the highest average success rates among the compared methods on LIBERO, LIBERO-Plus, LIBERO-Pro, and two real-robot tasks, together with the shortest mean episode time among matched LIBERO baselines.

1 INTRODUCTION

Vision–Language–Action (VLA) models offer a promising approach to robot control by mapping visual observations and language instructions to actions. To learn this mapping, models such as RT-2 (Brohan et al., 2023), OpenVLA (Kim et al., 2024), and $\pi_{0.5}$ (Black et al., 2025) adapt pretrained Vision-Language Models (VLMs) using robot demonstrations. From these demonstrations, an action head learns to translate multimodal representations into executable controls, making its design central to how actions are represented and predicted.

Two common action head designs, regression and flow matching, offer complementary capabilities. Regression heads predict actions directly in one forward pass, providing a simple interface for deterministic control (Kim et al., 2024). Their point outputs, however, do not explicitly represent conditional action distributions for sampling. Flow-matching heads model these distributions by learning a velocity field that transforms noise into action samples, as in π_0 (Black et al., 2024). Their standard sampling procedures typically require iterative integration and repeated action expert evaluations, adding inference cost (Kim et al., 2025).

These complementary capabilities raise a natural question: *Can we combine the single-pass prediction of regression with the distributional modeling of flow matching?* To this end, we can extend a single action prediction to a set of quantiles, where the median represents the action center and lower and upper quantiles describe its variability. Predicting these quantiles together allows our Quantile Head (Figure 1) to directly characterize the conditional distribution of each action coordinate. To formalize this intuition, we formulate a distributional learning objective and derive its equivalent quantile regression form, leading to Theorem 1.

Building on this quantile objective, we further analyze how auxiliary quantiles can improve the training signal for the median. Specifically, we study the noise in direct median updates under joint quantile supervision, leading to Theorem 2. Under the stated conditions, calibrated quantiles near the median reduce update variance relative to median-only supervision, with gaps held fixed and local correction speed matched. This result motivates the joint supervision of the median and auxiliary quantiles in our Quantile Head.

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

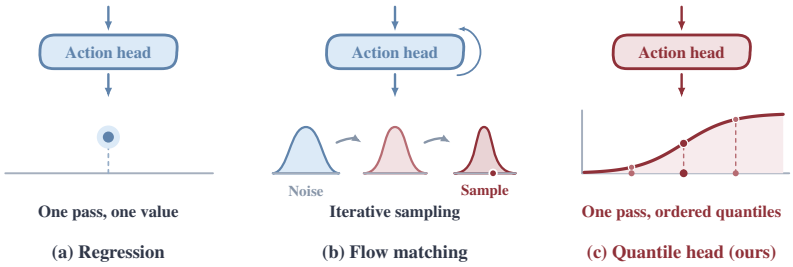


Figure 1: **Action head comparison.** Regression predicts a point, iterative flow matching reuses the same head at updated action states and integration times, and our head predicts ordered marginal quantiles in one pass for direct median control or optional sampling.

We integrate Quantile Head with a pretrained VLM for action learning. To limit the number of trainable parameters, we freeze the VLM and introduce learnable prompts that extract task context from its fixed features. Conditioned on this context, the quantile head jointly predicts a median and gaps between adjacent quantiles on either side for each action coordinate. The gaps are constrained to be positive and cumulatively subtracted from or added to the median, yielding ordered lower and upper quantiles. The quantiles for the entire action chunk are generated together in a single forward pass, capturing the center and variability of each action coordinate. To mitigate overfitting to demonstrations, we use a masked pinball loss that randomly withholds supervision for selected action coordinates, sharing the mask across all quantiles of each coordinate.

Our contributions cover theory, quantile action head design, and experimental evaluation:

1. **Theory.** Starting from a unified supervision framework for regression and flow matching, we derive a quantile objective for action distribution modeling and show that joint quantile supervision reduces noise in direct median updates under stated conditions.
2. **Quantile Head.** We propose a quantile action head that predicts ordered quantiles in a single forward pass. A masked pinball loss jointly supervises these outputs to model each action coordinate's conditional distribution.
3. **Experimental Evaluation.** We achieve state of the art performance on LIBERO and state of the art generalization on LIBERO-Plus and LIBERO-Pro, together with the best performance among the compared methods on two physical robot tasks.

2 Related Work

Regression-style action heads. Regression heads map observation-conditioned features directly to continuous actions using losses such as L_1 or L_2. For an unrestricted predictor, squared loss targets the conditional mean, while absolute loss targets the conditional median (Koenker & Bassett Jr, 1978). OpenVLA-OFT combines continuous L_1 regression with parallel action-chunk decoding, demonstrating the effectiveness of direct action prediction in VLA models (Kim et al., 2025). Such heads produce actions in a single forward pass and provide a simple interface for control. However, a point prediction for each action coordinate does not explicitly represent its conditional distribution, motivating supervision beyond a single central estimate.

Flow-matching-style action heads. Diffusion and flow-matching heads transform noise into samples from conditional action distributions (Zhang et al., 2026). Diffusion Policy learns action sequences through denoising (Chi et al., 2025), while π_0 (Black et al., 2024) and $\pi_{0.5}$ (Black et al., 2025) use flow-matching velocity supervision to connect noise and continuous action chunks. Their standard samplers use successive denoising or numerical integration steps, requiring repeated model evaluations. Recent flow-based VLAs also support one-step generation. SnapFlow uses progressive self-distillation of a pretrained flow policy, adding model evaluations to construct training targets (Luan et al., 2026). *Let It Be Simple* (Chen et al., 2026b) uses high-noise training to enable one-step action generation without distillation. Its LIBERO-Long ablation shows that the success-rate advantage over a uniformly trained ten-step baseline diminishes as the action horizon increases.

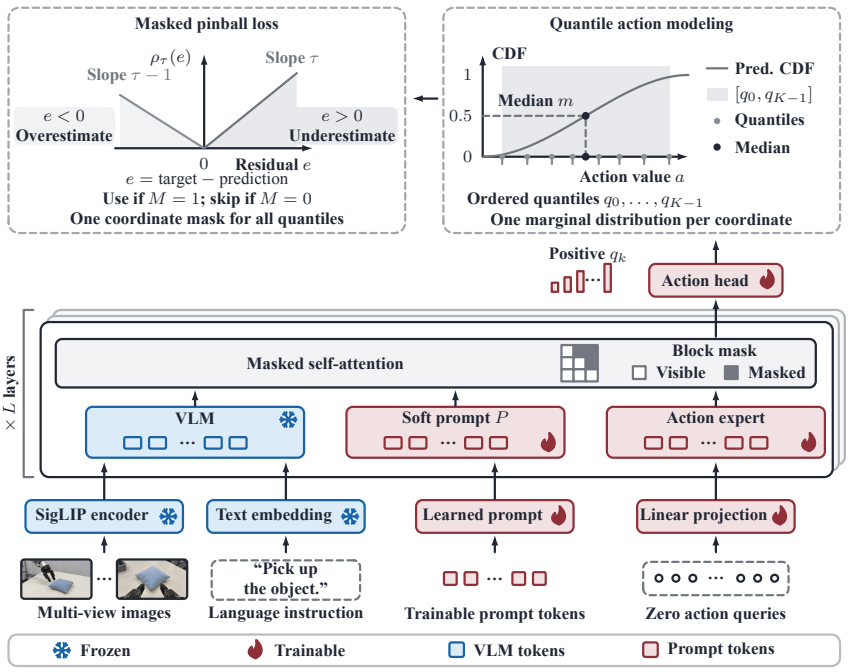


Figure 2: **Architecture of the Quantile Head.** Our model combines a frozen VLM, learnable prompts, and a Quantile Head. The VLM encodes images and instructions. Learnable prompts precede action tokens; masked self-attention lets prompts read VLM features and action tokens read both VLM and prompt features. The action head maps the resulting action features to a median and positive gaps, then accumulates these gaps around the median to form ordered marginal quantiles. We supervise these quantiles with masked pinball loss. Inference uses the median by default or samples from the predicted marginal quantiles.

Quantile-based learning. Richter & Wattenhofer (2019) learn continuous-action policies through advantage-weighted quantile regression. Learning across quantile levels can improve estimation of a target quantile under suitable conditions (Narayan et al., 2024). In probabilistic forecasting, OrderFusion constructs ordered outputs from a median anchor and nonnegative gaps (Yu et al., 2026). Within VLA systems, ReconVLA uses conformally calibrated action-error quantiles to rank candidates from a pretrained policy (Chen et al., 2026a). We jointly supervise ordered action quantiles for VLA control, with median decoding by default and optional quantile sampling.

Novelty. 1. Quantile Head extends point regression to jointly supervised conditional action quantiles, combining single-pass prediction with an explicit representation of action marginals. It predicts action values directly rather than learning a flow-matching velocity field, and the same outputs support median control and optional sampling. 2. Our focus is how auxiliary action quantiles contribute to learning the median used for VLA control. A local analysis identifies conditions under which this joint supervision reduces noise in direct median updates, while matched ablations demonstrate improved closed-loop performance even when inference uses only the median.

3 Method

3.1 Theoretical Analysis

A unified objective for action heads. Let X denote visual observations, language instructions, and the robot's proprioceptive state, and let $A \in \mathbb{R}^{H \times D}$ denote a demonstrated action chunk with H future steps and D action dimensions. For each coordinate $Y = A_{h,d}$, L_2 regression predicts Y from X. Conditional flow matching (CFM) predicts a velocity from X, an interpolated action, and time, then integrates the velocity field to generate actions. We unify their squared losses on action and velocity targets through a reference path. Draw $B \sim \nu(\cdot \mid X)$ with $B \perp Y \mid X$ and an

independent time $\alpha \sim \pi$ on $[0, 1]$, and set $A_\alpha = (1 - \alpha)B + \alpha Y$. The shared objective is

$$\mathcal{L}_{\nu,\pi}(\theta) = \mathbb{E}\left| f_\theta(X, A_\alpha, \alpha) - (Y - B) \right|^2. \tag{1}$$

Setting $B = \alpha = 0$ recovers squared action regression; a Gaussian reference and sampled times give CFM velocity supervision (Lipman et al., 2023). Appendix A.1 derives both cases.

Deriving quantile supervision. Under equation 1, unrestricted regression learns the mean action, whereas CFM learns a mean velocity field whose integration generates actions. Both squared losses have output gradients that grow with residuals (Appendix A.1). To learn action distributions directly, we extend squared supervision to the threshold labels $Z_z = \mathbf{1}\{Y \le z\}$. Their conditional means $F_X(z) = \Pr(Y \le z \mid X)$, over all $z \in \mathbb{R}$, determine the action distribution. The following result connects predicting these probabilities to quantile supervision.

Theorem 1 (proof in Appendix A.1)**.** *Let q_θ be measurable and nondecreasing in its quantile level, and let $G_\theta(\cdot \mid X)$ be the conditional cumulative distribution function (CDF) of $q_\theta(X, U)$, where $U \sim \mathrm{Unif}(0, 1)$ is independent of (X, Y). If both the true action Y and the predicted action $q_\theta(X, U)$ have finite first absolute moments, then*

$$\underbrace{\mathbb{E}_{X,Y} \int_{\mathbb{R}} \big(G_\theta(z \mid X) - \mathbf{1}\{Y \le z\}\big)^2 \, \mathrm{d}z}_{\mathcal{L}_{\mathrm{CDF}}(\theta)} = 2\mathbb{E}_{X,Y} \int_0^1 \rho_\tau\big(Y - q_\theta(X, \tau)\big) \, \mathrm{d}\tau. \tag{2}$$

Here ρ_τ is the pinball loss. If the true conditional distribution is representable, the population-optimal quantile function equals the true conditional quantile function almost everywhere.

Theorem 1 shows that integrated pinball loss is equivalent to CDF supervision up to a factor of two. Conditional action distributions can therefore be learned through quantile supervision, without constructing threshold labels. Our head jointly supervises a finite set of quantiles to represent each action coordinate's conditional distribution.

Local noise reduction in median updates. Theorem 1 provides a distributional basis for quantile supervision. Since our default policy executes the median, we further examine whether auxiliary quantile supervision can reduce noise in direct median updates.

Theorem 2 (Noise reduction in median updates)**.** *Under the assumptions in Appendix A.2, consider calibrated quantiles at distinct levels symmetric about and sufficiently close to $1/2$. With fixed gaps and matched local mean correction rates, the updates at the true median satisfy*

$$\mathrm{Var}(\Delta m_Q) < \mathrm{Var}(\Delta m_M). \tag{3}$$

Here Δm_Q and Δm_M are the joint-quantile and median-only random center updates, respectively.

Theorem 2 shows that auxiliary quantiles can reduce direct median-update variance at matched local correction rates under the stated conditions. This motivates jointly supervising the median and auxiliary quantiles in our Quantile Head. We evaluate the resulting policy empirically with jointly learned gaps and a wider quantile grid.

3.2 Quantile Head

Architecture overview. As shown in Figure 2, our design combines prompt adaptation of a frozen VLM with an ordered quantile head. We freeze the VLM to preserve its pretrained representations and limit the parameters learned from action demonstrations. We insert learnable prompts that read the frozen VLM features and provide task-specific context to the action expert. The expert produces action-token features for the quantile head, which constructs ordered predictions from a median and positive gaps. Joint pinball supervision follows the distributional interpretation in Theorem 1. Theorem 2 isolates a local center-update mechanism under calibrated, fixed gaps. Positive gaps enforce noncrossing marginal outputs for quantile decoding; the effect of joint supervision on median control is assessed in the matched experiments.

Learnable prompt insertion. We prepend N_p learnable prompt tokens $P \in \mathbb{R}^{N_p \times d_e}$ to the action-token sequence at the input of the action expert, where d_e is the expert's hidden width. Prompt embeddings are shared across observations, while their hidden states evolve with action tokens through

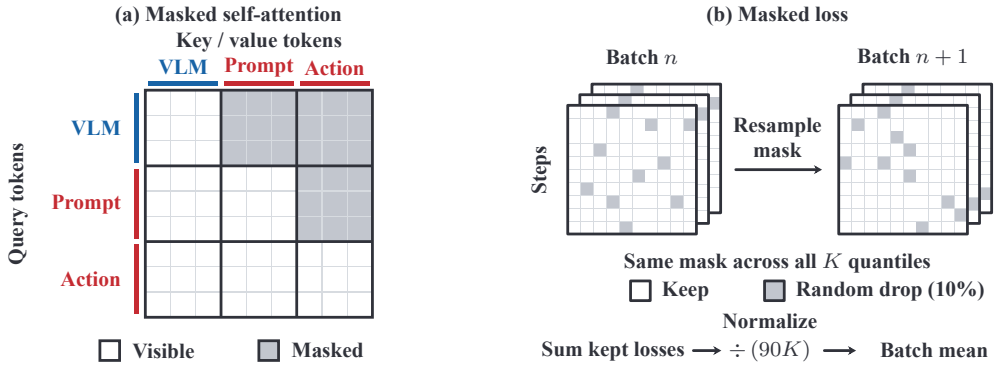


Figure 3: **Attention and loss masks (schematic).** (a) Block self-attention controls information flow among VLM, prompt, and action tokens. (b) Coordinate masks are independently resampled for each example in every training batch and shared across all quantiles.

the expert layers. At each layer, the block attention mask in Figure 3(a) allows prompt queries to attend to VLM and prompt tokens, while action queries attend to VLM, prompt, and action tokens. VLM queries attend only to VLM tokens, and attention is bidirectional within each group. The resulting prompt features provide observation-conditioned context for action prediction. We use $N_p = 16$ and jointly train P, the action expert, and output heads with the VLM frozen.

Ordered quantile head. We feed the final H action-token features from the action expert into two linear projections shared across future steps. For each action coordinate, one projection predicts a median m, and the other predicts raw gaps r_i^-, r_i^+ on its two sides. Let $K = 2c + 1$ and index the quantiles from 0 to $K - 1$, suppressing the step and coordinate indices. We map the raw gaps to positive values and accumulate them around m:

$$g_i^{\pm} = \frac{\delta_0}{\log 2} \operatorname{softplus}(r_i^{\pm}), \qquad q_c = m, \tag{4}$$

$$q_{c-j} = m - \sum_{i=1}^{j} g_i^-, \qquad q_{c+j} = m + \sum_{i=1}^{j} g_i^+, \quad j = 1, \ldots, c. \tag{5}$$

Positive gaps prevent quantile crossing; separately learned left and right gaps allow asymmetric spacing. We use $K = 21$ levels $\tau_k = 0.025 + 0.95k/20$, with $\tau_c = 0.5$. The scale $\delta_0 = 0.02$ sets the gap at zero raw output; the gaps can shrink or expand during training. All K predictions receive pinball supervision with a shared label mask for each action coordinate. For median decoding, we collect $q_c = m$ across steps and coordinates and unnormalize the action chunk.

Action expert inputs. The observation prefix combines SigLIP image embeddings with text embeddings of the task and discretized robot state. The state is normalized before being discretized into 256 bins and included in the text prompt. The action tokens are obtained by linearly projecting H all-zero action queries. Both training and inference fix the time condition to zero; its embedding and MLP still condition the expert's adaptive normalization.

3.3 Masked pinball loss

We apply pinball supervision from Theorem 1 to each action coordinate's K ordered quantiles. For sample b, let a_{bhd} be the normalized action target at future step h and dimension d, and let q_{bhdk} be its predicted quantile at level τ_k. With residual $e_{bhdk} = a_{bhd} - q_{bhdk}$, the pinball loss is

$$\rho_\tau(e) = \max\{\tau e, (\tau - 1)e\}. \tag{6}$$

Underprediction is penalized with weight τ, and overprediction with weight $1 - \tau$.

During training, we randomly withhold supervision for a subset of valid action coordinates. Let $V_{bhd} \in \{0, 1\}$ indicate a non-padded coordinate and $N_b = \sum_{h,d} V_{bhd}$. For each sample, we uniformly select $\min(\lceil rN_b \rceil, \max(N_b - 1, 0))$ valid coordinates without replacement and exclude their

Table 1: Success rates (%) on LIBERO and LIBERO-Pro. Best in **bold**; second-best underlined. Superscripts indicate result sources: [a]Kim et al. (2025); [b]Wu et al. (2026b).

(a) LIBERO

Method	Spatial	Object	Goal	Long	Avg.
Diffusion Policy[a] (Chi et al., 2025)	78.3	92.5	68.3	50.5	72.4
OpenVLA[a] (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5
π_0-FAST[a] (Pertsch et al., 2025)	96.4	96.8	88.6	60.2	85.5
π_0[a] (Black et al., 2024)	96.8	98.8	95.8	85.2	94.2
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
$\pi_{0.5}$ (Black et al., 2025)	98.8	98.2	98.0	92.4	96.85
VLANeXt (Wu et al., 2026a)	99.0	99.2	96.6	94.8	97.4
SimVLA (Luo et al., 2026)	**99.6**	99.8	98.6	96.4	98.6
WLA-0 (Yang et al., 2026)	99.0	**100.0**	97.8	97.6	98.6
InternVLA-A1.5 (Ma et al., 2026)	98.6	99.8	98.6	98.4	98.9
Quantile Head	99.0	**100.0**	**99.6**	**98.6**	**99.3**

(b) LIBERO-Pro

Method	Object	Pos.	Sem.	Task	Avg.
OpenVLA (Kim et al., 2024)	93.0	0.0	97.3	0.0	47.6
π_0 (Black et al., 2024)	90.5	0.0	90.5	0.0	45.3
$\pi_{0.5}$ (Black et al., 2025)	**96.0**	20.8	95.8	0.8	53.3
MolmoAct (Lee et al., 2025)	76.0	1.5	85.8	1.5	41.2
X-VLA (Zheng et al., 2026)	79.0	0.8	96.8	6.8	45.8
OpenVLA-OFT[b] (Kim et al., 2025)	27.6	5.8	59.0	0.6	23.3
X-VLA (CR eval.)[b] (Zheng et al., 2026)	78.4	1.6	82.9	16.4	44.8
VLA-Adapter[b] (Wang et al., 2026)	73.8	0.0	90.8	19.8	46.1
SimVLA (Luo et al., 2026)	81.5	8.3	**98.8**	6.0	48.6
SUREFlow (Islam et al., 2026)	68.5	0.0	87.8	0.0	39.1
Quantile Head	85.8	**31.3**	97.8	**26.0**	**60.2**

labels from the loss, using $r = 0.1$. The mask is redrawn on each training forward pass and retains at least one label whenever $N_b > 0$. Define $M_{bhd} = 1$ for valid, retained coordinates and 0 otherwise. As illustrated in Figure 3(b), each M_{bhd} is shared across all K quantiles, so the entire set of losses for a coordinate is retained or dropped together. This mask acts on the supervision terms.

We average over the retained coordinates and all K quantile levels within each sample, then over the B samples in the batch:

$$\mathcal{L} = \frac{1}{B} \sum_{b=1}^{B} \frac{\sum_{h,d,k} M_{bhd}\, \rho_{\tau_k}(e_{bhdk})}{K \max\Big(1, \sum_{h,d} M_{bhd}\Big)}. \tag{7}$$

Action dimensions introduced for model padding are removed before loss computation. Evaluation losses use all valid coordinates, with random dropping disabled and padded action steps excluded.

4 Experiments

4.1 Experimental Setting

Benchmarks and Tasks. We evaluate our method on three simulation benchmarks and two real-robot tasks. The simulation benchmarks are LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2026), and LIBERO-Pro (Zhou et al., 2025). We use four LIBERO suites: Spatial, Object, Goal, and Long (LIBERO-10), with ten tasks each. LIBERO-Plus perturbs these 40 tasks to create 10,030 instances, covering camera viewpoints, robot initial states, language instructions, lighting, backgrounds, sensor noise, and object layouts. LIBERO-Pro also evaluates robustness to changes in objects, positions, instructions, task goals, and environments. We further evaluate real-world performance on two manipulation tasks, with results in Table 3 and the platform and tasks described in Appendix B.2. Evaluation protocols and implementation details are provided in Appendix B. Baselines cited from other work retain their reported protocols.

4.2 Task Performance with Default Median Decoding

Overall Performance. With median decoding, Quantile Head achieves 99.3% average success across the four LIBERO suites, 0.4 percentage points above the strongest compared baseline (Table 1). On LIBERO-Plus, it reaches 87.1% zero-shot and 89.1% fine-tuned success, exceeding the respective strongest compared baselines by 1.4 and 5.0 percentage points (Table 2). On LIBERO-Pro, it achieves 60.2% without further training, exceeding $\pi_{0.5}$ at 53.3% by 6.9 percentage points. These mean success rates support the generalization of the full median policy across the evaluated benchmarks and perturbations. Matched head ablations examine the benefit of joint quantile supervision in Section 4.3.

Real-robot Performance. With median decoding, Quantile Head achieves 75.0% average success across the two real-robot tasks, compared with 58.5% for $\pi_{0.5}$, a 16.5 percentage point gain

Table 2: Success rates (%) on LIBERO-Plus. Within each setting, best in **bold**; second-best underlined. Superscript sources: [a]Fei et al. (2026); [b]Zhong et al. (2026); [d]Shi et al. (2026a). Unmarked baselines use their original papers.

(a) Zero-shot transfer

Method	Camera	Robot	Lang.	Light	Bkg.	Noise	Layout	Total
OpenVLA[a] (Kim et al., 2024)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
π_0-FAST[a] (Pertsch et al., 2025)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
π_0[a] (Black et al., 2024)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
OpenVLA-OFT[a] (Kim et al., 2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
$\pi_{0.5}$[b] (Black et al., 2025)	75.8	79.4	83.3	95.5	95.0	89.6	87.0	85.7
VLANeXt (Wu et al., 2026a)	90.4	65.7	81.8	95.9	82.5	94.1	80.8	83.9
GAM (Han et al., 2026)	83.1	70.0	84.8	**97.2**	94.3	95.3	79.1	85.5
GE-Act 2.0 (Liu et al., 2026)	**94.1**	50.7	81.4	94.0	60.5	**95.5**	83.1	80.4
ACoT-VLA[b] (Zhong et al., 2026)	68.9	80.3	84.1	95.6	93.1	81.5	**88.3**	83.6
Quantile Head	71.6	**80.7**	**87.6**	96.1	**97.2**	94.8	87.8	**87.1**

(b) Supervised fine-tuning

Method	Camera	Robot	Lang.	Light	Bkg.	Noise	Layout	Total
π_0[b] (Black et al., 2024)	79.6	21.1	72.5	84.7	86.2	68.3	69.4	67.4
$\pi_{0.5}$[b] (Black et al., 2025)	70.3	41.7	81.1	97.3	94.6	71.8	84.9	75.7
OpenVLA-OFT+ (Fei et al., 2026)	92.8	30.3	85.8	94.9	93.9	89.3	77.6	79.6
MemoryVLA[d] (Shi et al., 2026b)	91.4	48.6	79.4	95.2	95.3	94.0	75.7	81.9
MemoryVLA++ (Shi et al., 2026a)	**96.8**	49.7	71.0	96.6	97.0	**96.0**	78.6	82.7
ACoT-VLA[b] (Zhong et al., 2026)	91.2	62.5	80.3	95.1	91.5	88.3	84.9	84.1
Quantile Head (SFT)	88.6	**77.0**	**86.2**	**97.4**	**97.3**	94.1	**87.3**	**89.1**

Table 3: Real-robot success rates (%). The $\pi_{0.5}$ baseline is fully fine-tuned. Δ gives the gain over $\pi_{0.5}$ in percentage points.

Real-robot Task	$\pi_{0.5}$	Quantile Head	Δ
Place apple on yellow plate	31	52	+21
Remove cuboid from blue plate	86	98	+12
Average	58.5	75.0	+16.5

(Table 3). The $\pi_{0.5}$ baseline is fully fine-tuned; both methods use the same demonstration data, optimization settings, and evaluation conditions (Appendix B.2). Success increases from 31% to 52% for apple placement (+21 percentage points) and from 86% to 98% for cuboid removal (+12 percentage points). These results show higher success on both evaluated tasks using deterministic median decoding, although apple placement remains challenging at 52% success.

4.3 Ablation Studies

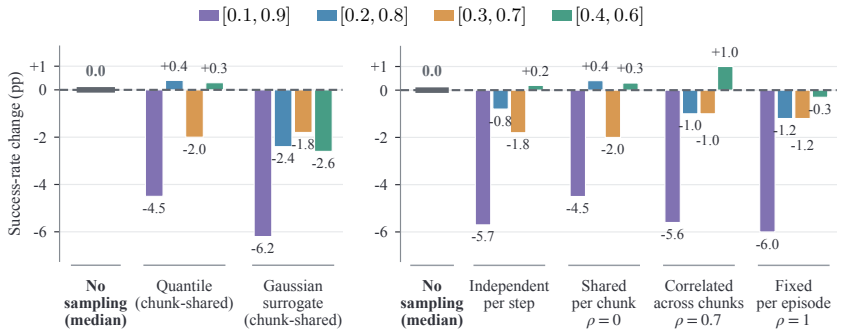
Head and Component Ablations. Under matched training and evaluation conditions, median-decoded Quantile Head achieves 99.3% average success across the four LIBERO suites, compared with 96.0%, 97.3%, and 96.3% for L_2, L_1, and sampled Flow Matching, respectively (Table 4(a)). Mean episode time, including failures, decreases from 8.13 s for L_1 to 5.88 s, a 27.7% reduction.

In Table 4(b), Median-Only reaches 97.5% mean success, compared with 99.3% for the full head. Detached Median Anchor reaches 98.7%: noncentral losses still update shared features and gaps, but their direct gradient to the median projection is blocked. These mean rates are consistent with potential benefits from both shared supervision and the direct gradient path; task success does not measure error to the true conditional median. Symmetric Quantiles reaches 97.9%, below the full head with independently learned left and right offsets. Noise Query and VLM Query reach 93.8% and 92.8%, respectively, showing that all-zero action queries outperform both alternatives.

In (c), joint quantile supervision yields higher mean evaluation success across all four prompt and label-mask configurations: 83.5% to 91.8%, 95.5% to 98.8%, 86.0% to 97.0%, and 97.5% to 99.3%. These matched comparisons suggest potential generalization benefits from joint quantile supervision for the median policy, even when inference uses only the median without quantile sampling.

Table 4: Ablations on the four LIBERO suites: success rates (%) and episode times (s). Episode times in (a) average over all evaluation episodes. See Appendix B for ablation settings.

Setting		Spatial↑	Object↑	Goal↑	Long↑	Avg.↑
(a) Supervised Fine-Tuning	Avg. Episode Time (s)↓					
L_2	8.73	97.0	98.0	96.0	93.0	96.0
L_1	8.13	99.0	99.0	96.0	95.0	97.3
Flow Matching	12.57	97.0	98.0	96.0	94.0	96.3
Quantile Head	5.88	99.0	100.0	99.6	98.6	99.3
(b) Prompt + Label Mask						
Quantile Head		99.0	100.0	99.6	98.6	99.3
Median-Only		98.0	99.0	97.0	96.0	97.5
Detached Median Anchor		99.0	99.0	99.0	97.8	98.7
Symmetric Quantiles		99.0	98.0	98.0	96.6	97.9
Noise Query		94.0	95.0	93.0	93.0	93.8
VLM Query		94.0	96.0	91.0	90.0	92.8
(c) Component Ablations (Frozen VLM)						
Prompts	Pinball Loss / Label Mask					
		83.0	86.0	80.0	85.0	83.5
✓		97.0	97.0	95.0	93.0	95.5
	✓	90.0	94.0	94.0	89.0	91.8
	✓	85.0	88.0	85.0	86.0	86.0
✓	✓	98.0	99.0	97.0	96.0	97.5
✓	✓	99.0	100.0	98.0	98.0	98.8
	✓	97.0	99.0	99.0	93.0	97.0
✓	✓	99.0	100.0	99.6	98.6	99.3



(a) Action decoding. **(b) Temporal sampling structure.**

Figure 4: **Inference-time sampling on LIBERO-Long.** Colors denote quantile windows; bars show success-rate changes from median decoding (98.6%; dashed lines) in percentage points. Quantile (a) and Shared per chunk (b) use the same setting ($\rho = 0$). See Appendix B.

4.4 Exploring Alternative Sampling Strategies

Sampling Analysis. In the LIBERO-Long sampling evaluation, with cross-chunk correlation $\rho = 0.7$, narrowing the sampling window from $[0.1, 0.9]$ to $[0.4, 0.6]$ increases LIBERO-Long success from 93.0% to 99.6%, exceeding median decoding (98.6%) by 1.0 percentage point (Figure 4). These results illustrate flexibility at inference and motivate future work on selecting sampling strategies. Median decoding requires no sampling choices and remains our default. Narrow windows keep sampled quantile levels near 0.5, and temporal correlation couples these levels across chunks; median decoding directly uses the learned center.

Action Distribution Analysis. Figure 5 shows predicted marginal ranges in one LIBERO-Long rollout: narrower at selected transport moments and wider during grasping and placing. This pattern suggests a possible role for local action adjustments around object contact, but does not establish distributional accuracy. Per-dimension ranges appear in Appendix Figure 9.

Sampling in Two Controlled Scenes. We train each head on two demonstrations per scene, with identical initial observations and instructions but opposite detours (Figure 6). Without an obstacle,

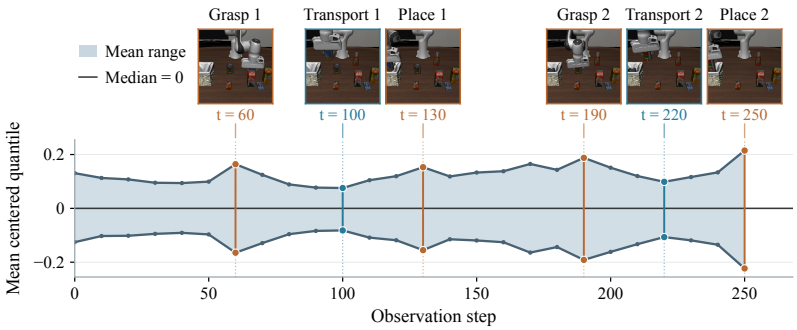


Figure 5: **Mean predicted action spread during manipulation.** Median-centered Q_0–Q_1 ranges averaged over six normalized action dimensions ($h = 0$); endpoints use linear tail extrapolation.

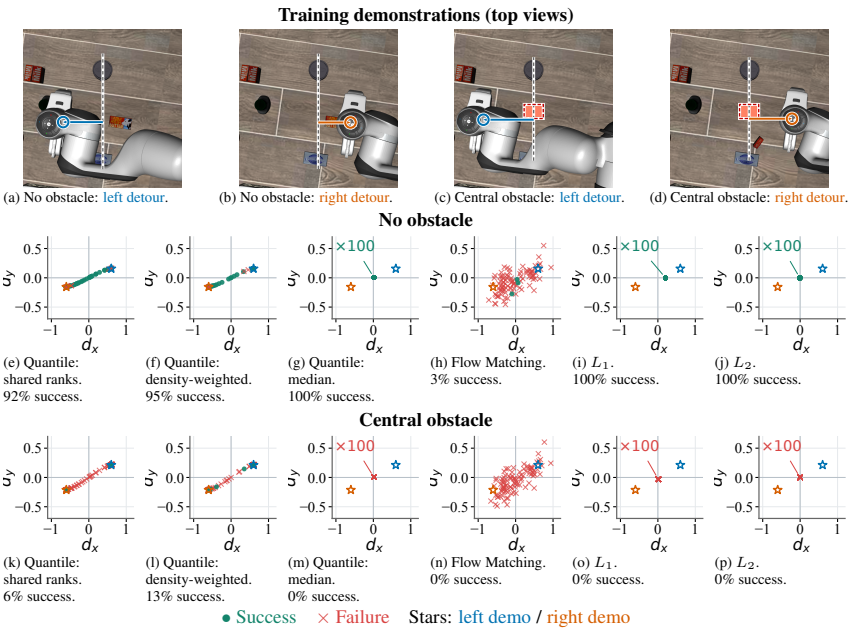


Figure 6: **Training demonstrations and action head outputs.** Top views show demonstration states; points show initial (d_x, d_y) commands from one training run, colored by outcome. Subcaptions report success rates; $\times 100$ marks overlapping points. Details: Appendices B.3–B.4.

median-decoded Quantile Head and both regression heads (L_1, L_2) reach 100% success; uniform and density-weighted quantile sampling reach 92% and 95%. With an obstacle, these sampling rules reach 6% and 13%, while median decoding, both regression heads, and Flow Matching reach 0%. Sampling benefits are thus task-dependent, and obstacle-scene success remains low. Density weighting requires no retraining.

5 Conclusion

We presented Quantile Head, a theory-inspired method that jointly supervises ordered marginal action quantiles for single-pass median control. Our local analysis shows lower median-update variance than median-only supervision with nearby calibrated quantiles, fixed gaps, smooth densities, and matched correction rates. Joint supervision improves mean success across four matched LIBERO configurations. The median policy reaches 99.3% on LIBERO and 75.0% across two real-robot tasks. Sampling yields gains and losses; systematic strategy selection remains future work.

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

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

Kevin Black, Noah Brown, James Darpinian, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025.

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.

Lingling Chen, Zongyao Lyu, and William J. Beksi. ReconVLA: An uncertainty-guided and failure-aware vision-language-action framework for robotic control. *arXiv preprint arXiv:2604.16677*, 2026a.

Yitong Chen, Shiduo Zhang, Jingjing Gong, and Xipeng Qiu. Let it be simple: One-step action generation for vision-language-action models. *arXiv preprint arXiv:2606.05737*, 2026b.

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.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Jisang Han, Seonghu Jeon, Jaewoo Jung, René Zurbrügg, Honggyu An, Tifanny Portela, Marco Hutter, Marc Pollefeys, Seungryong Kim, and Sunghwan Hong. Geometric action model for robot policy learning. *arXiv preprint arXiv:2606.17046*, 2026.

Md Tanvir Islam, Sai Navaneet Peddapalli, Sangmoon Lee, and Sangtae Ahn. SUREFlow: State-space uncertainty-aware REsidual flow matching for robust robot manipulation. *arXiv preprint arXiv:2607.10504*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024.

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

Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

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.

Renhang Liu, Wenzhi Zhao, Zhuo Yang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Guanghui Ren, Youlun Peng, Rongjun Jin, Nan Wang, et al. Ge-act 2.0: Pretraining and scaling a world-action model for robotic manipulation. *arXiv preprint arXiv:2609.05588*, 2026.

Wuyang Luan, Junhui Li, Weiguang Zhao, Wenjian Zhang, Tieru Wu, and Rui Ma. Snapflow: One-step action generation for flow-matching vlas via progressive self-distillation. *arXiv preprint arXiv:2604.05656*, 2026.

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

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, Yuqiang Yang, et al. InternVLA-A1.5: Unifying understanding, latent foresight, and action for compositional generalization. *arXiv preprint arXiv:2607.04988*, 2026.

Taman Narayan, Serena Lutong Wang, Kevin Robert Canini, and Maya Gupta. Expected pinball loss for quantile regression and inverse cdf estimation. *Transactions on Machine Learning Research*, 2024.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Oliver Richter and Roger Wattenhofer. Learning policies through quantile regression. *arXiv preprint arXiv:1906.11941*, 2019.

Hao Shi, Weiye Li, Bin Xie, Yulin Wang, Renping Zhou, Tiancai Wang, Xiangyu Zhang, Ping Luo, and Gao Huang. MemoryVLA++: Temporal modeling via memory and imagination in vision-language-action models. *arXiv preprint arXiv:2606.09827*, 2026a.

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. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026b.

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.

Xiao-Ming Wu, Bin Fan, Kang Liao, Jian-Jian Jiang, Runze Yang, Yihang Luo, Zhonghua Wu, Wei-Shi Zheng, and Chen Change Loy. Vlanext: Recipes for building strong vla models. *arXiv preprint arXiv:2602.18532*, 2026a.

Yueh-Hua Wu, Tatsuya Matsushima, and Kei Ota. Continuous reasoning for vision-language-action. *arXiv preprint arXiv:2606.00229*, 2026b.

Yi Yang, Zhihong Liu, Siqi Kou, Yiyang Chen, Yanzhe Hu, Jianbo Zhou, Boyuan Zhao, Zhijie Wei, Xiao Xia, Xueqi Li, et al. World-language-action model for unified world modeling, language reasoning, and action synthesis. *arXiv preprint arXiv:2606.05979*, 2026.

Runyao Yu, Yuchen Tao, Fabian Leimgruber, Tara Esterl, Jochen Stiasny, Derek W Bunn, Qingsong Wen, Hongye Guo, and Jochen L Cremer. Orderfusion: Encoding orderbook for end-to-end probabilistic intraday electricity price forecasting. *Advanced Engineering Informatics*, 76:105131, 2026.

Yan Zhang, Yinan Wu, Haoran Duan, and Jungong Han. CofactVLA: Deconfounding vision-language-action models via counterfactual intervention. *arXiv preprint arXiv:2608.04396*, 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.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Si Liu, and Guanghui Ren. Acot-vla: Action chain-of-thought for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8152–8162, 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.

Hui Zou and Ming Yuan. Composite quantile regression and the oracle model selection theory. *The Annals of Statistics*, 36(3):1108–1126, 2008.