ACTIONPIECE: RETHINKING ACTION TOKENIZATION FOR AUTOREGRESSIVE VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action tokenizers play a central role in autoregressive vision-language-action (VLA) models, determining both the targets for policy training and the executable commands recovered from predicted tokens. Tokenizer fidelity is commonly evaluated using pointwise reconstruction metrics such as mean squared error (MSE), yet small individual errors do not fully characterize how faithfully action adjustments across demonstrations are preserved. After compression, similar actions may still cluster around a representative motion, while the adjustments needed for different contexts are diminished, distorted, or even reversed. To assess whether similar actions retain their relative physical proximity after reconstruction, we introduce physical rank consistency (PRC), which measures local physical distance-order preservation beyond pointwise reconstruction accuracy. We further present ActionPiece, which preserves physical action relationships through joint supervision of representation learning and quantization. Physical Rank Preservation (PRP) supervises near-far ordering in encoder and quantized feature distances, while Quantization Regularization (QR) applies the same ordering to codeword assignment distributions. Together, these objectives guide tokenizer learning to preserve physical distance ordering alongside reconstruction accuracy. Under the Qwen3-VL-4B policy training setup, ActionPiece achieves 94.8% success on LIBERO and 68.8% on unseen LIBERO-Plus, with additional evaluations reaching 71.9% on SimplerEnv and 82.2%, 42.7%, 29.5% on VLA-Arena L0, L1, and L2, respectively. Code is provided in the supplementary material and will be publicly released upon acceptance.

1 INTRODUCTION

Autoregressive vision-language-action (VLA) models generate discrete tokens that are decoded into executable robot commands (Brohan et al., 2023; Kim et al., 2024; Pertsch et al., 2025). The action tokenizer connects continuous demonstrations to this discrete prediction process: it determines both the targets used for policy training and the commands recovered from predicted tokens. Existing approaches construct this interface through coordinate discretization, frequency-domain compression, or learned vector quantization (Kim et al., 2024; Pertsch et al., 2025; Dong et al., 2026; Liu et al., 2026c;a;b). These designs enable compact action generation, but compression also changes the actions that the robot ultimately executes. An important question is therefore what an action tokenizer should preserve to support precise control.

Across multiple demonstrations of the same task, similar motions recur with adjustments to object positions, robot states, and execution stages. These adjustments allow similar motions to accommodate different contexts. In operations requiring precise alignment, grasping, or contact, even small action differences can affect execution outcomes. A tokenizer compresses these demonstrations into a discrete representation with finite capacity and reconstructs executable commands from it. Although reconstruction objectives such as mean squared error (MSE) keep individual actions close to their originals, they do not explicitly coordinate reconstruction errors across demonstrations. Small individual reconstruction errors therefore do not fully characterize how faithfully the adjustments between demonstrations are preserved. After compression, similar actions may still cluster around a representative motion, while the adjustments needed to accommodate different contexts are distorted,

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.

diminished, or even reversed. These changes persist even when a policy correctly predicts the corresponding token sequence: the decoder still executes the reconstructed action.

This motivates studying **relational fidelity alongside pointwise reconstruction accuracy**. We characterize action differences using a physical distance that combines translation, rotation, and gripper state, and examine whether their relative magnitudes survive tokenization. To measure this property, we introduce **physical rank consistency** (PRC). For each action chunk, PRC compares its distance rankings to the same original neighbors before and after reconstruction. Computing these rankings in decoded action space provides a common physical reference across token vocabularies, sequence lengths, and decoder architectures. Together, reconstruction accuracy and PRC describe how faithfully a tokenizer reproduces individual commands and preserves the local relationships among them.

To preserve these relationships during tokenization, we develop ActionPiece. Physical distances between original action chunks provide near–far supervision for both representation learning and codeword assignment. **Physical rank preservation** (PRP) loss encourages the corresponding ordering in a combination of encoder and quantized feature distances. **Quantization regularization** (QR) loss applies this ordering to the divergence between codeword assignment distributions, extending supervision to how actions are assigned to the discrete vocabulary. Together, these objectives address two stages of compression: learning action features and mapping them to codewords. They are optimized jointly with reconstruction in a Transformer tokenizer with residual vector quantization. After tokenizer training, the frozen encoder and quantizer supply discrete policy targets, while the frozen decoder converts autoregressively predicted tokens into executable action chunks.

Under a shared Qwen3-VL-4B policy training setup, ActionPiece outperforms all compared action tokenizers, achieving 94.8% success on LIBERO and 68.8% on unseen LIBERO-Plus, exceeding the strongest baseline on each benchmark by 1.1 and 4.5 percentage points, respectively (Table 1). Evaluations on SimplerEnv and VLA-Arena extend the comparison to real-to-sim transfer and challenging control conditions, where ActionPiece achieves the highest aggregate success rates of 71.9% and 51.5%, respectively, among the evaluated methods. These results are obtained through standard next-token prediction using ActionPiece as the action tokenizer. Controlled ablations show that PRP and QR jointly improve both decoded physical rank consistency and downstream policy success. Comparisons across tokenizers further examine the relationship between reconstruction fidelity, PRC, and execution performance (Figure 2).

Our contributions are threefold:

- We introduce PRC, a measure of local physical distance-order preservation that complements reconstruction accuracy and supports comparison across action tokenizers.
- We develop ActionPiece, jointly supervising physical relationships in learned features and codeword assignment distributions to construct discrete action representations for autoregressive VLA policies.
- We evaluate ActionPiece across four benchmarks and real-world manipulation tasks, using controlled comparisons and ablations to examine how physical relationship supervision affects representation quality and robot execution.

2 Related Work

2.1 Action-Space Design in Robotic Foundation Models

Robot foundation models connect pretrained vision–language representations to motor commands through discrete, continuous, or hybrid action interfaces. Discrete autoregressive approaches cast control as next-token prediction: RT-2 and OpenVLA discretize action coordinates into bins (Brohan et al., 2023; Kim et al., 2024), while FAST compresses action chunks into shorter token sequences (Pertsch et al., 2025). VLM2VLA instead expresses commands using the VLM's existing natural-language vocabulary (Hancock et al., 2025). These interfaces retain the token-prediction objective of pretrained VLMs, providing a direct route to transfer language grounding and semantic knowledge to robot control.

Continuous approaches predict real-valued action chunks without categorical decoding. OpenVLA-OFT uses parallel action prediction with an ℓ_1 regression objective (Kim et al., 2025); diffusion and flow-matching policies, including RDT, π_0, and the GR00T family, model continuous action distributions (Liu et al., 2025b; Black et al., 2025b; NVIDIA et al., 2025; GEAR Team et al., 2025). Continuous heads support fine-grained control, and generative heads can represent multiple valid motions while generating action chunks without autoregressively decoding every action token.

Hybrid designs combine discrete supervision with continuous execution. $\pi_{0.5}$ uses action tokens during pretraining and introduces a flow-matching action expert during post-training (Black et al., 2025a). Knowledge Insulation trains the backbone with discrete action targets while blocking gradients from the continuous expert, retaining pretrained knowledge alongside efficient execution (Driess et al., 2025). HybridVLA jointly trains autoregressive and diffusion predictions and adaptively combines them (Liu et al., 2025a); Fast-in-Slow couples autoregressive supervision with a fast diffusion-based execution module (Chen et al., 2025). Such designs seek to combine the semantic benefits of token prediction with responsive continuous control.

2.2 Action Discretization and Tokenization

Some action tokenizers discretize coordinates directly or apply predefined transforms to action chunks. Coordinate-wise binning provides a simple action vocabulary, as in RT-2 and OpenVLA (Brohan et al., 2023; Kim et al., 2024), but represents each coordinate and time step separately. FAST applies a discrete cosine transform to action chunks, quantizes the coefficients, and learns byte-pair encoding (BPE) to compress the resulting sequences (Pertsch et al., 2025). Its frequency basis is fixed, while the BPE vocabulary is learned from action data.

Learned latent tokenizers instead optimize an encoder and decoder around a quantized representation. Vector-quantized approaches learn codebooks from action data; FASTer combines residual vector quantization (RVQ), structured action patching, temporal and spectral reconstruction, and blockwise autoregressive generation (Liu et al., 2026c). OAT uses finite scalar quantization with nested dropout and causal attention to obtain compact, fully decodable tokens with coarse-to-fine ordering (Liu et al., 2026a;b). ActionCodec studies overlap, token budget, vision–language alignment, and residual grammar from the perspective of VLA optimization (Dong et al., 2026). X-Tokenizer adds semantic supervision from a frozen visual–language teacher (Kang et al., 2026). These methods develop action interfaces through compression, token ordering, and multimodal alignment. ActionPiece introduces explicit physical-order supervision during encoding and quantization, and uses PRC to measure neighborhood-order preservation in decoded action space.

3 Physical Structure in Action Tokenization

3.1 Preserving Action Variations

Demonstrations of the same task contain recurring motions with adjustments to object positions, robot states, and execution stages. These adjustments allow similar motions to accommodate different contexts. In operations requiring precise alignment, grasping, or contact, even small action differences can affect execution outcomes.

An action tokenizer compresses continuous action chunks into discrete token sequences with finite representational capacity, then decodes them into executable commands. This lossy compression introduces reconstruction errors. Although tokenizers are commonly trained with an MSE objective to keep reconstructed actions close to their originals, this objective penalizes individual reconstruction errors without explicitly coordinating their directions across actions. Consider the normalized Euclidean components x_i, x_j of two demonstration chunks, with reconstruction errors $e_i = \widehat{x}_i - x_i$. Their reconstructed difference satisfies

$$\widehat{x}_j - \widehat{x}_i = (x_j - x_i) + (e_j - e_i). \tag{1}$$

The first term is the action variation present in the demonstrations; the second is the change introduced by tokenization. This additional term can attenuate the original difference, amplify it, or even reverse its direction. Small individual reconstruction errors therefore do not fully characterize how faithfully the adjustments between demonstrations are preserved. After compression, similar actions may still cluster around a representative motion, while their original near-to-far ordering is altered.

These distortions carry through to policy execution. Even when a VLA correctly predicts the token sequence encoding a demonstrated action, the decoder still produces its reconstructed version. If compression weakens or distorts the adjustments required by different contexts, correct token prediction cannot recover the original adjustments. Action tokenization should therefore preserve physical distinctions among demonstrations alongside accurate reconstruction of individual actions.

3.2 Physical Rank Consistency

We assess relationship preservation through the relative ordering of physical distances between action chunks. Our distance d_{phys} combines translation, shortest geodesic rotation on $\mathrm{SO}(3)$, and gripper differences across a chunk, as defined in Section 4.2. For each anchor A_i, let $\mathcal{N}_i^k$ contain its k nearest neighbors in the original action space, and let $\widehat{A}_i = \mathcal{D}(\mathcal{T}(A_i))$ denote its reconstructed action chunk, where $\mathcal{T}$ maps action chunks to discrete tokens and $\mathcal{D}$ is the corresponding decoder. Physical Rank Consistency (PRC) compares distances to these same neighbors before and after reconstruction:

$$\mathrm{PRC@}\,k = \frac{1}{N}\sum_{i=1}^{N} \rho_{\mathrm{S}}\Big([d_{\text{phys}}(A_i, A_j)]_{j\in\mathcal{N}_i^k}, [d_{\text{phys}}(\widehat{A}_i, \widehat{A}_j)]_{j\in\mathcal{N}_i^k}\Big), \tag{2}$$

where ρ_{S} denotes Spearman correlation. We use $k = 32$.

Higher PRC indicates better preservation of distance ordering within the original neighborhoods. Comparing ranks captures the relative size of action differences without requiring every distance to remain numerically identical. Evaluation in decoded action space provides a common physical reference across token vocabularies, sequence lengths, and decoder architectures. MSE measures the fidelity of individual reconstructions, while PRC measures the preservation of physical distance ordering. Figure 2 examines both properties in relation to downstream policy success. Guided by this perspective, ActionPiece introduces physical-order supervision into representation learning and codeword assignments, preserving action relationships alongside pointwise reconstruction.

4 ActionPiece

ActionPiece preserves physical action relationships alongside reconstruction accuracy through joint supervision of encoding and quantization, following Section 3. A shallow Transformer encoder and decoder with residual vector quantization (RVQ) provide the codec. Physical rank preservation structures the learned action representations; quantization regularization carries the same ordering into codeword assignments. Both augment reconstruction during tokenizer training. The resulting vocabulary supplies the targets for standard autoregressive policy learning. Figure 1 summarizes the tokenizer training objectives and the resulting policy learning and execution pipeline.

4.1 Compact, Fully Decodable Tokens

For an action chunk $A = (a_1, \ldots, a_H) \in \mathbb{R}^{H\times D}$, an encoder produces latent features, which are discretized by residual vector quantization (RVQ) into n discrete tokens. A decoder reconstructs the complete action chunk in one pass from their codebook embeddings:

$$Z = \mathcal{T}(A) = (z_1, \ldots, z_n), \qquad \widehat{A} = \mathcal{D}(Z). \tag{3}$$

Architecture and quantization details are provided in Appendix D.

4.2 Physical Action Distance

For two end-effector commands $a = (p, R, g)$ and $a' = (p', R', g')$, we measure rotational distance by the shortest geodesic angle on $\mathrm{SO}(3)$:

$$d_{\mathrm{SO}(3)}(R, R') = \left\|\mathrm{Log}(R^\top R')^\vee\right\|_2, \tag{4}$$

where $\mathrm{Log}(\cdot)^\vee$ is the rotation-vector form of the principal matrix logarithm. The per-step physical distance combines translation, rotation, and gripper state:

$$\delta_{\text{phys}}(a, a') = \alpha_p \left\|\frac{p - p'}{s_p}\right\|_2^2 + \alpha_R \left(\frac{d_{\mathrm{SO}(3)}(R, R')}{s_R}\right)^2 + \alpha_g |g - g'|^2. \tag{5}$$

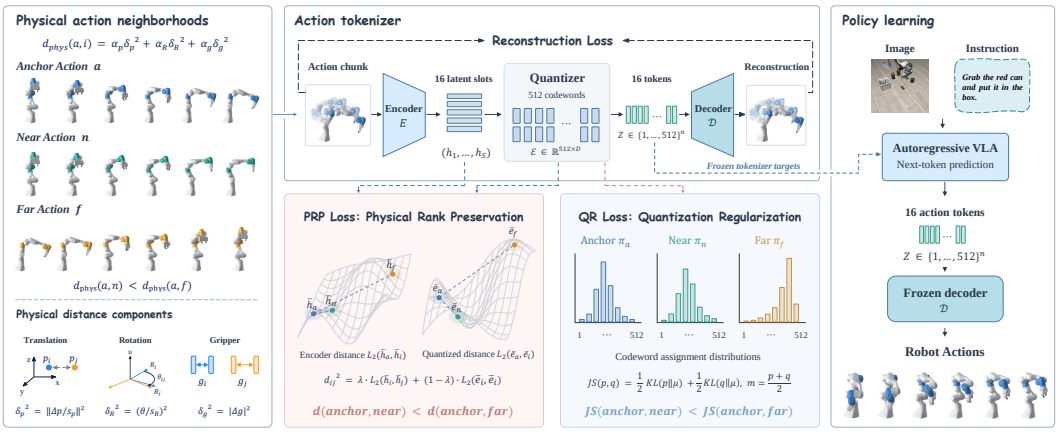

Figure 1: **Overview of ActionPiece.** Physical action distance (defined in Section 4.2) combines normalized translation, shortest-path rotation on $\mathrm{SO}(3)$, and gripper differences to select anchor, near, and far action chunks within a training batch. The tokenizer compresses each eight-step chunk into 16 discrete tokens and reconstructs executable actions. Alongside reconstruction, physical rank preservation (PRP, defined in Section 4.3) encourages the near–far ordering in encoder and quantized feature distances, while quantization regularization (QR, defined in Section 4.4) encourages the same ordering in the Jensen–Shannon divergence between codeword assignment distributions. Here L_2 denotes the slot-averaged squared distance between normalized features.

The scales are fitted on training actions, with group weights fixed within each embodiment (Appendix D.3). For action chunks, d_{phys} averages per-step distances over corresponding time steps. Rotations are mapped to valid elements of $\mathrm{SO}(3)$ before distance computation. This distance defines the near-to-far ordering for both training objectives and PRC in decoded action space.

4.3 Physical Rank Preservation

Physical rank preservation aligns the relative distances of learned action representations with those of physical motions. We measure representation distance both before and after quantization. Let S denote the number of latent feature slots. Let $\bar{h}_{i,s}$ be the encoder representation of action chunk A_i at slot s, and let $\bar{e}_{i,s}$ be its selected codeword after the quantizer output projection. Both are normalized by LayerNorm followed by unit ℓ_2 normalization. Their distances are

$$d^2_{\text{enc}}(i,j) = \frac{1}{S}\sum_{s=1}^{S} \|\bar{h}_{i,s} - \bar{h}_{j,s}\|_2^2,$$
$$d^2_{\text{quant}}(i,j) = \frac{1}{S}\sum_{s=1}^{S} \|\bar{e}_{i,s} - \bar{e}_{j,s}\|_2^2. \tag{6}$$

We use the weighted representation distance

$$d^2_{\text{rep}}(i,j) = \lambda\, d^2_{\text{enc}}(i,j) + (1-\lambda)\, d^2_{\text{quant}}(i,j), \qquad D_{ij} = \sqrt{d^2_{\text{rep}}(i,j)}. \tag{7}$$

We emphasize quantized features because they directly feed the decoder, while retaining supervision on encoder features. Implementation details are provided in Appendix D. A differentiable codeword estimator transmits gradients through code selection; Appendix D.1 specifies the estimator, and Appendix D.2 covers multiple quantization levels.

For each anchor action chunk A_i, we exclude the anchor itself and rank the remaining action chunks in the training batch by physical distance. We select the positive (near) action $A_{j_i^+}$ at the 5th percentile ($q_{0.05}$) and the negative (far) action $A_{j_i^-}$ at the 95th percentile ($q_{0.95}$). The rank objective is

$$\mathcal{L}_{\text{rank}} = \frac{1}{B}\sum_i \text{softplus}\Big(D_{ij_i^+} - D_{ij_i^-} + m_r\Big), \qquad m_r = 0.1. \tag{8}$$

The loss encourages representations to retain the near-to-far order of physical motions. Both physical-order objectives use these pairs; Appendix D.4 specifies the selection rule.

4.4 Quantization Regularization

Quantization regularization applies physical-order supervision directly to codeword probabilities. Let $\pi_{i,s}$ denote the soft probability distribution over codewords for slot s of action chunk A_i, obtained from encoder-to-codeword distances as specified in Eq. (15). We compare two action chunks using the average Jensen–Shannon divergence between corresponding slots:

$$d_{\mathrm{JS}}(i,j) = \frac{1}{S}\sum_{s=1}^{S} \mathrm{JS}(\pi_{i,s}, \pi_{j,s}), \tag{9}$$

where $\mathrm{JS}(p,q) = \frac{1}{2}\,\mathrm{KL}(p\|\mu) + \frac{1}{2}\,\mathrm{KL}(q\|\mu)$ and $\mu = (p+q)/2$. Using the same neighbors j_i^+ and j_i^- as the rank objective, we define

$$\mathcal{L}_{\mathrm{quant}} = \frac{1}{B}\sum_{i} \mathrm{softplus}\big(d_{\mathrm{JS}}(i,j_i^+) - d_{\mathrm{JS}}(i,j_i^-) + m_q\big), \qquad m_q = 0.1. \tag{10}$$

Physical rank preservation supervises distances between action representations, while quantization regularization supervises the probabilities that determine codeword selection. Together, they impose physical order on these two aspects of the tokenizer.

4.5 Training and Policy Integration

The reconstruction objective measures mean squared error in normalized action coordinates:

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{BHD}\sum_{i,t,d}\left(\frac{A_{i,t,d} - \widehat{A}_{i,t,d}}{s_d}\right)^2. \tag{11}$$

The complete tokenizer objective combines reconstruction, the standard commitment loss (van den Oord et al., 2017), and the two physical-order objectives:

$$\mathcal{L}_{\mathrm{tok}} = \mathcal{L}_{\mathrm{MSE}} + \lambda_{\mathrm{vq}}\mathcal{L}_{\mathrm{commit}} + \lambda_r\mathcal{L}_{\mathrm{rank}} + \lambda_q\mathcal{L}_{\mathrm{quant}}. \tag{12}$$

We set $\lambda_{\mathrm{vq}} = 0.25$ and $\lambda_r = \lambda_q = 6.25 \times 10^{-4}$. The component ablation in Table 4 removes either physical-order objective or both from this formulation. After training, the tokenizer is frozen and its indices are added to the VLM vocabulary. The VLA is optimized with standard next-token prediction,

$$\mathcal{L}_{\mathrm{VLA}} = -\sum_{t=1}^{n} \log p_\theta(z_t \mid I, \ell, z_{<t}), \tag{13}$$

and the generated sequence is decoded into an executable action chunk.

5 Experiments

We relate tokenizer fidelity and physical structure to policy success, then test ActionPiece through controlled comparisons, transfer evaluations, and ablations. We evaluate on LIBERO (Liu et al., 2023), unseen LIBERO-Plus (Fei et al., 2025), SimplerEnv (Li et al., 2024c), VLA-Arena (Zhang et al., 2025), and real-world manipulation tasks (Appendix B), covering in-distribution performance, perturbation robustness, real-to-sim transfer, and challenging control conditions. LIBERO-Plus is evaluated without its demonstrations in policy training, while VLA-Arena policies train on L0 and are evaluated across L0–L2. Controlled LIBERO/LIBERO-Plus comparisons and ablations use Qwen3-VL-4B for 30K policy training steps with matched data and execution settings. Full data protocols, tokenizer and policy training hyperparameters, hardware, and evaluation budgets are provided in Appendix A.

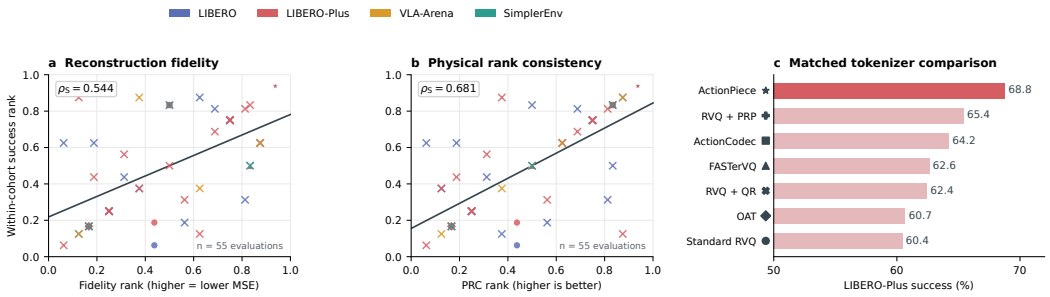

Figure 2: **Reconstruction fidelity, physical structure, and policy success.** Panels (a–b) show normalized within-group ranks across 55 tokenizer–benchmark evaluations. Groups follow the original experimental batches, with benchmarks evaluated separately. Colors indicate benchmarks; symbols identify the methods shown in (c), and crosses denote additional variants. Panel (c) reports LIBERO-Plus success for selected tokenizers and ablations using Qwen3-VL-4B at 30K policy training steps. PRP and QR denote physical rank preservation and quantization regularization.

Table 1: **Matched action-tokenizer comparison on LIBERO and LIBERO-Plus.** All policies use the same Qwen3-VL-4B backbone, training data, prompt, a global batch size of 128, an 8-step prediction and execution horizon, and 30K policy training steps. LIBERO-Plus is evaluated out of distribution: no LIBERO-Plus demonstrations are used for policy training. FASTerVQ* denotes our implementation following the published method. Best results are bold.

	LIBERO					LIBERO-Plus							
Tokenizer	Spatial	Object	Goal	Long	Overall	Camera	Robot	Language	Light	Bkg.	Noise	Layout	Overall
ActionCodec (Dong et al., 2026)	93.4	**99.0**	92.4	89.8	93.7	33.3	38.8	80.8	88.4	90.2	**58.0**	75.9	64.2
OAT (Liu et al., 2026a;b)	85.6	96.2	87.2	75.2	86.1	32.2	47.8	79.5	82.8	82.7	46.4	67.4	60.7
FASTerVQ* (Liu et al., 2026c)	91.8	95.6	90.5	87.4	91.3	39.0	42.3	74.5	85.8	84.9	56.6	69.0	62.6
FAST (Pertsch et al., 2025)	94.7	98.0	93.1	82.6	92.1	33.8	47.1	80.0	89.6	88.4	51.3	75.9	64.3
Standard RVQ	91.8	98.2	91.6	82.0	90.9	31.5	43.9	75.3	86.0	88.0	42.5	72.9	60.4
ActionPiece (Ours)	**96.2**	**99.0**	**93.4**	**90.6**	**94.8**	**45.1**	**49.6**	**82.2**	**91.9**	**91.4**	57.6	**78.0**	**68.8**

5.1 Reconstruction and Physical Structure

Figure 2 examines how tokenizer reconstruction fidelity and neighborhood-order preservation relate to downstream policy success, building on the action-space property introduced in Section 3. We compute normalized ranks within the original experimental groups and summarize their association across 55 tokenizer–benchmark evaluations. Both metrics exhibit positive associations with success, with PRC yielding a higher Spearman correlation (0.681) than reconstruction fidelity (0.544). This observation motivates examining physical structure alongside pointwise reconstruction. Controlled comparisons and ablations assess the benefits of physical supervision, with additional RVQ objectives evaluated in Appendix C.

5.2 Matched Action-Tokenizer Comparison

We compare ActionPiece with ActionCodec (Dong et al., 2026), OAT (Liu et al., 2026a;b), FASTer (Liu et al., 2026c), FAST (Pertsch et al., 2025), and standard RVQ through one Qwen3-VL-4B policy implementation. The policy data, prompt, optimizer, and evaluation are shared, and all policies are trained for 30K steps. Each tokenizer retains its native output length and vocabulary. The comparison evaluates the policy trained with each action representation.

Table 1 reports 94.8% on in-distribution LIBERO for ActionPiece, compared with 93.7% for ActionCodec, the strongest baseline on this benchmark. The separation is larger on unseen LIBERO-Plus: ActionPiece reaches 68.8%, compared with 64.3% for the strongest baseline, and leads on six of seven shift categories. All policies are trained on LIBERO demonstrations; LIBERO-Plus measures transfer under camera, robot, language, illumination, background, noise, and layout changes. The advantage over the strongest baseline grows from 1.1 points on LIBERO to 4.5 on LIBERO-Plus.

Table 2: **SimplerEnv WidowX results.** We follow the official SimplerEnv evaluation protocol (Li et al., 2024c). All values are success rates (%). Best and second-best averages are bold and underlined, respectively.

Method	Put Spoon on Towel	Put Carrot on Plate	Stack Block on Block	Put Eggplant in Basket	Avg.
OpenVLA-OFT (Kim et al., 2025)	34.2	30.0	30.0	72.5	41.8
RoboVLM (Li et al., 2024b)	50.0	37.5	0.0	83.3	42.7
Magma (Yang et al., 2025)	37.5	29.2	20.8	91.7	44.8
CogACT (Li et al., 2024a)	71.7	50.8	15.0	67.5	51.3
SpatialVLA (Qu et al., 2025)	20.8	20.8	25.0	70.8	34.4
TraceVLA (Zheng et al., 2025)	12.5	16.6	16.6	65.0	27.7
VideoVLA (Shen et al., 2025)	75.0	20.8	45.8	70.8	53.1
π_0 (Black et al., 2025b)	29.2	62.5	29.2	91.6	53.1
$\pi_{0.5}$ (Black et al., 2025a)	49.3	64.7	44.7	69.7	57.1
Isaac-GR00T-N1.6-Bridge (GEAR Team et al., 2025)	64.5	65.5	5.5	93.0	57.1
LangForce (Lian et al., 2026)	89.6	63.8	33.3	79.2	<u>66.5</u>
ActionCodec-BAR (Dong et al., 2026)	71.7	64.2	55.0	70.0	65.2
ActionPiece (Ours)	83.3	54.2	58.3	91.7	**71.9**

Gains across six shift categories support preserving action adjustments as a useful principle for execution beyond the training distribution.

5.3 Evaluation on SimplerEnv and VLA-Arena

In both benchmarks, we integrate ActionPiece by replacing the action tokenizer while retaining the policy architecture and training protocol. Policies are trained with standard next-token prediction, without benchmark-specific policy modifications.

For each SimplerEnv task, we average success over five independent 24-episode repeats, then macro-average the four task scores. ActionPiece achieves the highest average success of 71.9% in this real-to-sim evaluation, with task-specific strengths varying across methods. ActionPiece leads block stacking at 58.3%, a task requiring precise placement, alongside strong spoon (83.3%) and eggplant (91.7%) results. These outcomes extend the evidence for physical relationship supervision to tokenizers trained on real-robot actions.

Across all 3,400 VLA-Arena rollouts, ActionPiece reaches 82.2%/42.7%/29.5% on L0/L1/L2 and an equal-weight 33-cell mean of 51.5%, the highest overall average among the compared models. As on SimplerEnv, this aggregate advantage accompanies variation in task-level performance. Full results, including cumulative safety costs, are reported in Appendix E. Despite training only on L0, ActionPiece leads the L1 and L2 averages by 7.0 and 5.0 points. Leading L2 results on static obstacles, dynamic obstacles, and unseen objects extend this advantage to obstacle-constrained execution and object generalization.

5.4 Component Ablation

We evaluate the two physical-order objectives defined in Sections 4.3 and 4.4. Table 4 adds each objective individually and both together to standard RVQ. Standard RVQ reaches 90.9% on LIBERO and 60.4% on LIBERO-Plus. Adding physical rank preservation raises these scores to 93.8% and 65.4%; adding only quantization regularization yields 92.9% and 62.4%. Combining both components achieves the highest overall success on both benchmarks: 94.8% and 68.8%. Notably, the gain over standard RVQ is larger on unseen LIBERO-Plus (+8.4 percentage points) than on LIBERO (+3.9 points).

The same ablations raise PRC from 0.902 for standard RVQ to 0.947 with physical rank preservation and 0.916 with quantization regularization; the combined model reaches 0.953. The policy improvements thus accompany better physical neighborhood preservation. Physical rank preservation contributes the larger individual gain; adding quantization regularization further raises LIBERO-Plus success by 3.4 percentage points, supporting their joint use. Combining the objectives exceeds either

Table 3: **VLA-Arena results** (Zhang et al., 2025). SR is success rate (%). Avg equally weights the 11 suites. For each task and difficulty level, the highest SR across the six models is bold, including ties; the same rule applies to Avg. Cumulative costs (CC) are omitted for space; the full SR and CC results are provided in Appendix E, Table 8. Baselines include $\pi_{0.5}$ (Black et al., 2025a), GR00T-N1.6 (GEAR Team et al., 2025), Evo-Depth (Lin et al., 2026), and Motus (Bi et al., 2025). Qwen-GR00T denotes Qwen3-VL (Bai et al., 2025) with the GR00T N1 action head (NVIDIA et al., 2025).

Task	**Metric**	$\pi_{0.5}$			GR00T-N1.6			Qwen-GR00T			Evo-Depth			Motus			**ActionPiece**		
		L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2
Safety																			
Static obstacles	SR ↑	90.0	62.0	40.0	72.0	30.0	14.0	**91.0**	18.0	5.0	88.0	66.0	48.0	64.0	67.0	41.0	**91.0**	**83.0**	**57.0**
Cautious grasp	SR ↑	50.0	14.0	0.0	16.0	2.0	0.0	77.0	8.0	2.0	**78.0**	24.0	0.0	62.0	**25.0**	**11.0**	77.0	18.0	1.0
Hazard avoidance	SR ↑	58.0	30.0	36.0	64.0	4.0	10.0	67.0	15.0	19.0	40.0	0.0	14.0	65.0	**34.0**	21.0	**68.0**	22.0	**46.0**
State preservation	SR ↑	58.0	56.0	54.0	66.0	50.0	38.0	86.0	56.0	47.0	88.0	66.0	56.0	63.0	63.0	**72.0**	**94.0**	**80.0**	64.0
Dynamic obstacles	SR ↑	50.0	44.0	22.0	74.0	50.0	2.0	81.0	56.0	3.0	**82.0**	60.0	6.0	58.0	33.0	11.0	**82.0**	**76.0**	**27.0**
Distractor																			
Static distractors	SR ↑	88.0	16.0	16.0	46.0	**32.0**	6.0	91.0	6.0	2.0	**94.0**	20.0	**24.0**	69.0	19.0	13.0	92.0	21.0	22.0
Dynamic distractors	SR ↑	80.0	66.0	**54.0**	70.0	**72.0**	18.0	**92.0**	49.0	25.0	86.0	60.0	32.0	77.0	51.0	26.0	90.0	63.0	43.0
Extrapolation																			
Prep. combinations	SR ↑	62.0	**24.0**	**6.0**	48.0	0.0	0.0	51.0	1.0	0.0	66.0	0.0	0.0	41.0	7.0	1.0	**69.0**	15.0	0.0
Task workflows	SR ↑	38.0	20.0	**22.0**	42.0	0.0	0.0	51.0	3.0	9.0	32.0	0.0	0.0	38.0	25.0	14.0	**64.0**	**33.0**	21.0
Unseen objects	SR ↑	48.0	60.0	20.0	26.0	18.0	16.0	63.0	46.0	26.0	**78.0**	52.0	4.0	63.0	**65.0**	19.0	**78.0**	59.0	**43.0**
Long Horizon																			
Long horizon	SR ↑	85.0	0.0	0.0	29.0	2.0	0.0	96.0	0.0	0.0	93.0	0.0	0.0	61.0	**4.0**	**1.0**	**99.0**	0.0	0.0
Avg	SR ↑	64.3	35.6	24.5	50.3	23.6	9.5	76.9	23.5	12.5	75.0	31.6	16.7	60.1	35.7	20.9	**82.2**	**42.7**	**29.5**

Table 4: **Component ablation on LIBERO and LIBERO-Plus.** Starting from standard RVQ, we add physical rank preservation (PRP), quantization regularization (QR), or both. All policies use Qwen3-VL-4B with 30K policy training steps. Best results are bold.

	LIBERO					**LIBERO-Plus**							
Variant	Spatial	Object	Goal	Long	Overall	Camera	Robot	Language	Light	Bkg.	Noise	Layout	Overall
Standard RVQ	91.8	98.2	91.6	82.0	90.9	31.5	43.9	75.3	86.0	88.0	42.5	72.9	60.4
+ PRP	**96.2**	97.4	93.0	88.4	93.8	35.5	45.6	**83.0**	85.9	90.3	55.1	77.1	65.4
+ QR	95.4	97.2	91.2	87.8	92.9	33.0	38.8	77.7	90.8	89.7	50.2	74.3	62.4
ActionPiece (both)	**96.2**	**99.0**	**93.4**	**90.6**	**94.8**	**45.1**	**49.6**	82.2	**91.9**	**91.4**	**57.6**	**78.0**	**68.8**

alone in six LIBERO-Plus categories, including a camera-shift gain from 35.5% with PRP to 45.1%. This complementarity supports guiding both feature geometry and discrete code assignment with physical relationships.

6 Conclusion

We presented ActionPiece, an action tokenizer that combines reconstruction with physical-order supervision in representation learning and quantization. We also introduced physical rank consistency (PRC) to evaluate local physical distance-order preservation in decoded action space, complementing pointwise reconstruction metrics. Controlled comparisons and ablations show improvements in PRC and downstream policy success, with evaluations spanning four simulation benchmarks and real-world manipulation tasks. These findings highlight the value of assessing differences between related actions alongside individual reconstruction accuracy. An important direction for future work is to examine when distortions of these differences affect closed-loop execution, particularly in tasks requiring precise, context-dependent action adjustments.

AI USE STATEMENT

We used generative AI tools to assist with translation, manuscript polish, and general coding tasks. The research motivation, conceptual framework, hypotheses, experimental design, model training, and interpretation of results were developed or conducted by the authors without generative AI assistance. We did not use generative AI for synthetic data generation, mathematical claims or proofs, data cleaning, or qualitative analysis. All AI-assisted outputs were manually reviewed and double-checked by the authors, who take responsibility for the final content of the paper, code, and figures.

ETHICS STATEMENT

The robot demonstration data used in this work, including data collected in real-world environments, do not involve the collection or disclosure of personally identifiable information. When deploying policies using our action tokenizer on physical robots, appropriate safety measures should be in place, including a physical emergency-stop button, workspace separation, and suitable motion and force limits. These precautions help reduce the risk of injury or property damage during real-world execution.

REPRODUCIBILITY STATEMENT

Appendix A documents the experimental protocols, while Appendix D provides implementation details, including physical-distance normalization and near–far sampling. We will provide the source code for training in the supplementary materials to facilitate reproduction of our results.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics, 2025. URL https://arxiv.org/abs/2511.08544.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model, 2025. URL https://arxiv.org/abs/2512.13030.

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

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.

π_0: : A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, June 2025b. 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, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, and Pheng-Ann Heng. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning, 2025. URL https://arxiv.org/abs/2506.01953.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, and Jianye Hao. Actioncodec: What makes for good action tokenizers. *arXiv preprint arXiv:2602.15397*, 2026.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better, 2025. URL https://arxiv.org/abs/2505.23705.

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

GEAR Team, Allison Azzolini, Johan Bjorck, Valts Blukis, Fernando Castañeda, Rahul Chand, Yan Chang, Danyi Chen, Nikita Cherniadev, Xingye Da, Runyu Ding, Shunjia Ding, Hassan Eslami, Linxi "Jim" Fan, Yu Fang, Max Fu, Shenyuan Gao, Yunhao Ge, Fengyuan Hu, Spencer Huang, Joel Jang, Xiaowei Jiang, Yunfan Jiang, Ryan Julian, Kaushil Kundalia, Jan Kautz, Zhiqi Li, Kevin Lin, Wei Liu, Runyu Lu, Zhengyi Luo, Loic Magne, Yunze Man, Ajay Mandlekar, Abhishek Mishra, Avnish Narayan, Connor Pederson, Nadun Ranawaka, Scott Reed, Sunil Srinivasa, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Shihao Wang, Jimmy Wu, Yubo Wu, Yuqi Xie, Tianyi Xiong, Mengda Xu, Yinzhen Xu, Fu-En Yang, Seonghyeon Ye, Zhiding Yu, K. R. Zentner, Zhe Zhang, Kaiyuan Zheng, Ruijie Zheng, and Yuke Zhu. Gr00t n1.6: An improved open foundation model for generalist humanoid robots. https://research.nvidia.com/labs/gear/gr00t-n1_6/, December 2025.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting, 2025.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Miracle Kang, Lights Shi, Lucy Liang, Roy Gan, Dongxiu Liu, Pushi Zhang, Sylas Chen, Shawn Qin, Yinan Zheng, Jinliang Zheng, Hao Wang, Xianyuan Zhan, and Hang Su. X-tokenizer: A multimodal action tokenizer for vision-language-action pretraining, 2026. Preprint.

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 *Annual Conference on Robot Learning (CoRL)*, 2024.

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

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation, 2024a.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models, 2024b.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, 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. In *Annual Conference on Robot Learning (CoRL)*, 2024c.

Shijie Lian, Bin Yu, Xiaopeng LIN, Laurence Tianruo Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Cong Huang, and Kai Chen. Langforce: Bayesian decomposition of vision language action models via latent action queries. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=DUJL3a5Y57.

Tao Lin, Yuxin Du, Jiting Liu, Nuobei Zhu, Yunhe Li, Yuqian Fu, Yinxinyu Chen, Hongyi Cai, Zewei Ye, Bing Cheng, Kai Ye, Yiran Mao, Yilei Zhong, MingKang Dong, Junchi Yan, Gen Li, and Bo Zhao. Evo-depth: A lightweight depth-enhanced vision-language-action model. *arXiv preprint arXiv:2605.14950*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

Chaoqi Liu, Xiaoshen Han, Jiawei Gao, Yue Zhao, Haonan Chen, and Yilun Du. OAT: Ordered Action Tokenization. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026a. doi: 10.15607/RSS.2026.XXII.075. URL https://www.roboticsproceedings.org/rss22/p075.html.

Chaoqi Liu, Yue Zhao, Haonan Chen, Xiaoshen Han, Jiawei Gao, Ehsan Adeli, and Yilun Du. Ordered action tokens for visuomotor policy learning, 2026b. URL https://arxiv.org/abs/2607.21670.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, Chengkai Hou, Mengdi Zhao, KC alex Zhou, Pheng-Ann Heng, and Shanghang Zhang. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model, 2025a. URL https://arxiv.org/abs/2503.10631.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, pp. 29982–30009, 2025b.

Yicheng Liu, Shiduo Zhang, Zibin Dong, Baijun Ye, Tianyuan Yuan, Xiaopeng Yu, Linqi Yin, Chenhao Lu, Junhao Shi, Luca Jiang-Tao Yu, Liangtao Zheng, Jingjing Gong, Tao Jiang, Xipeng Qiu, and Hang Zhao. FASTer: Toward powerful and efficient autoregressive vision–language–action models with learnable action tokenizer and block-wise decoding. In *The Fourteenth International Conference on Learning Representations*, 2026c. URL https://openreview.net/forum?id=k6nTUFoqeT.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang

Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025.

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, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020. ISBN 9781728199986.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators, 2025.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1723–1736. PMLR, 06–09 Nov 2023.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, and Jianfeng Gao. Magma: A foundation model for multimodal ai agents. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14203–14214, 2025. doi: 10.1109/CVPR52734.2025.01325.

Borong Zhang, Jiahao Li, Jiachen Shen, Yishuai Cai, Yuhao Zhang, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. Vla-arena: An open-source framework for benchmarking vision-language-action models, 2025.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies, 2025.

A Experimental Protocol

Benchmarks and data. We select four benchmarks spanning three action-data sources and complementary generalization settings. LIBERO (Liu et al., 2023) provides the in-distribution reference, using human teleoperation demonstrations collected in simulation. LIBERO-Plus (Fei et al., 2025) evaluates these LIBERO-trained policies under seven types of unseen perturbations, without training on LIBERO-Plus data. SimplerEnv (Li et al., 2024c) evaluates a WidowX policy trained on real-robot BridgeData V2 demonstrations (Walke et al., 2023), testing real-to-sim transfer. For VLA-Arena (Zhang et al., 2025), we use keyboard-teleoperated simulation demonstrations at L0 and evaluate across L0–L2, with L1/L2 testing generalization to more demanding task configurations. Together, these settings test tokenizers across different demonstration sources and assess policy performance both within and beyond the training distribution.

Tokenizer training. Action data are sampled at 20 Hz for LIBERO, 10 Hz for VLA-Arena, and 5 Hz for BridgeData V2. Tokenizers are trained on action chunks using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 1×10^{-4} and a batch size of 128 for 100K steps. For fair comparison, each baseline tokenizer is trained using the remaining optimizer hyperparameters specified in its original work. Within each controlled comparison, the action representation, normalization, and execution settings are fixed.

VLA training. All experiments use eight NVIDIA RTX PRO 6000 GPUs. For policy training, we follow the default StarVLA protocol (Community, 2026) and use AdamW (Loshchilov & Hutter, 2019) with an initial learning rate of 1×10^{-5} and a cosine annealing schedule. We use DeepSpeed ZeRO-2 (Rajbhandari et al., 2020), gradient clipping with a maximum norm of 1.0, and no gradient accumulation.

Matched policy evaluation. Within each controlled comparison, the VLM backbone, prompt, robot demonstrations, optimizer, global batch, action and execution horizons, seed, and evaluation protocol are fixed. The matched LIBERO/LIBERO-Plus tokenizer comparisons and component ablations use 30K policy training steps. Full LIBERO contains 2,000 rollouts, full LIBERO-Plus 10,030, and complete VLA-Arena 3,400.

B Real-World Experiments

Hardware setup. We evaluate ActionPiece on a real-world platform comprising two Franka Research 3 robot arms, each equipped with a Robotiq 2F-85 gripper. Three Intel RealSense D435 cameras provide a head-mounted view and left- and right-wrist views. Figure A1 shows the experimental platform.

Tasks and demonstrations. We consider test-tube insertion in two configurations and collect 1,000 demonstrations for each. In the *single-arm* task, a test tube initially stands upright on the table. Each arm must independently pick up and insert a tube into the rack, giving two insertions per trial. We report both the success rate of the first insertion and the full-task success rate, which requires both insertions to succeed. Figure A2 illustrates the right- and left-arm sequences using the head camera and the corresponding wrist camera. In the *bimanual* task, the tube initially lies horizontally on the table. One arm picks it up and hands it to the other arm, which then inserts it into the rack. This configuration is particularly challenging during the inter-arm handover, when the receiving gripper must align with and secure the tube before the other arm releases it. Figure A3 illustrates the sequence from the head camera, with red boxes highlighting the tube and active manipulation regions. The 2,000 demonstrations cover tube and rack placements distributed across the tabletop, with variation in both position and orientation. We choose this task because it requires precise grasping and alignment across these varied spatial configurations. Similar grasping, transport, and insertion motions recur, while their execution requires adjustments to the specific tube and rack poses. Distorting, amplifying, or attenuating these adjustments during tokenization can compromise grasping or insertion accuracy, directly connecting the task to the motivation in Section 3.

Evaluation and results. We compare ActionPiece with $\pi_{0.5}$, FAST, and plain RVQ over 50 trials per method for each task. Table 5 reports the success rates. ActionPiece achieves 78% success on

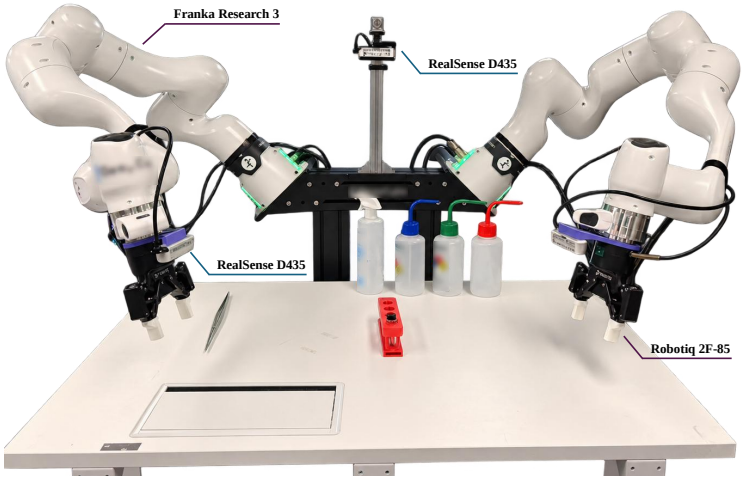

Figure A1: **Real-world experimental platform.** Two Franka Research 3 arms with Robotiq 2F-85 grippers and three RealSense D435 cameras, positioned at the head and the two wrists.

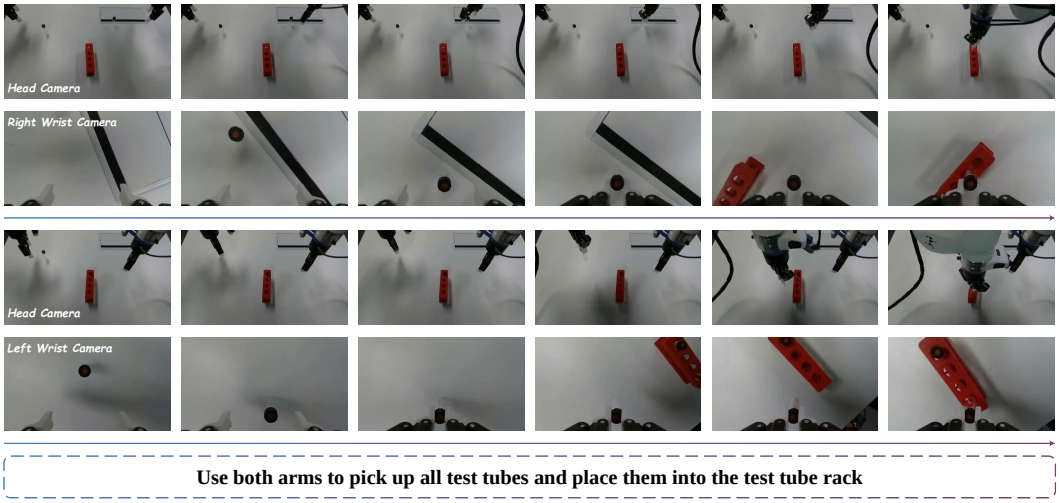

Figure A2: **Single-arm test-tube insertion sequences executed by ActionPiece.** The upper two rows show right-arm manipulation from the head and right-wrist cameras; the lower two rows show left-arm manipulation from the head and left-wrist cameras. Each arm grasps an upright tube and inserts it into the rack. Frames progress from left to right within each sequence. Full-task success requires both insertions to succeed.

the first single-arm insertion and 58% when both arms must complete their respective insertions, compared with 72% and 48% for $\pi_{0.5}$. Its bimanual success rate is 32%, exceeding $\pi_{0.5}$ by 10 percentage points. Relative to plain RVQ, the gains in full-task success are 22 percentage points for the single-arm configuration and 18 for the bimanual configuration. ActionPiece leads under all three criteria, extending its performance advantage to physical execution. These results support the benefit of preserving physical action relationships in real-world manipulation requiring precise grasping, alignment, and inter-arm coordination.

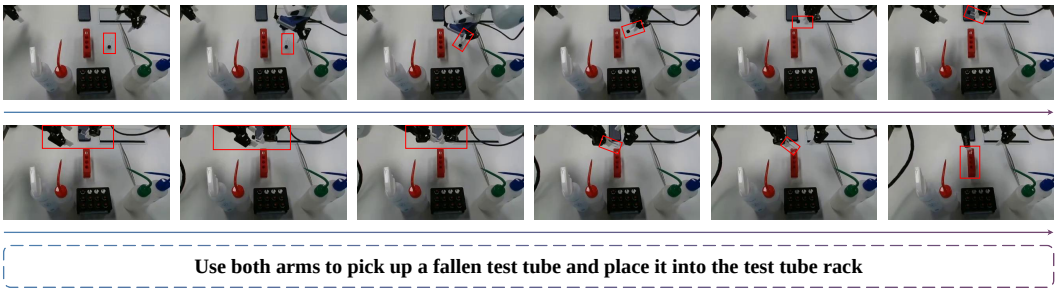

Figure A3: **Bimanual test-tube handover and insertion executed by ActionPiece.** Head-camera frames show one arm picking up a tube lying on the table, transferring it to the other arm, and the receiving arm inserting it into the rack. Frames progress from left to right across the upper row and then the lower row. The handover requires coordinated positioning, grasping, and release between the two arms. Red boxes highlight the test tube and active manipulation regions.

Table 5: **Real-world test-tube insertion results.** Success rates (%) over 50 trials per method for each task, with 1,000 collected demonstrations per task. Single-arm results distinguish the first insertion from full-task success (both insertions). Best results are bold.

Task	ActionPiece	$\pi_{0.5}$	FAST	Plain RVQ
Single-arm: first insertion	**78**	72	64	58
Single-arm: both insertions	**58**	48	42	36
Bimanual	**32**	22	18	14

C Alternative Training Objectives

We compare alternative objectives added to the standard RVQ tokenizer to examine how different supervision affects reconstruction, physical structure, and policy success. Table 6 reports measurements for these variants and the component ablations, using Qwen3-VL-4B at 30K policy training steps.

SIGReg. We apply Sketched Isotropic Gaussian Regularization (SIGReg) from LeJEPA (Balestriero & LeCun, 2025) to encoder features, encouraging their distribution to match an isotropic Gaussian.

Temporal. This objective matches the temporal changes of reconstructed actions to those of demonstrations. For translation and gripper commands, it compares first differences between adjacent time steps in the reconstructed and target sequences. For rotation, it compares the relative quaternion rotations between adjacent steps. Thus, it supervises the demonstrated pattern of change, including its magnitude and direction, rather than penalizing motion itself.

Neighborhood. This objective attracts physically similar actions in representation space. For each anchor i, we select the ten physically nearest actions in the minibatch, denoted by $\mathcal{N}_{10}(i)$. With larger nonnegative weights w_{ij} for physically closer neighbors, the objective is

$$\mathcal{L}_{\text{nei}} = \frac{\sum_i \sum_{j \in \mathcal{N}_{10}(i)} w_{ij}\, d^2_{\text{lat}}(i,j)}{\sum_i \sum_{j \in \mathcal{N}_{10}(i)} w_{ij}}, \tag{14}$$

where d_{lat} is the distance between action representations. Neighborhood supervises attraction to nearby actions; physical rank preservation additionally compares a nearby action with a farther one to supervise their relative order (Section 4.3).

These objectives target different properties of the codec. Temporal supervises changes within a decoded action chunk, Neighborhood attracts representations across chunks, and physical rank preservation supervises relative distances. The results illustrate why both reconstruction and neighborhood measurements are useful: the Neighborhood variant improves policy success over standard RVQ while its PRC decreases, whereas ActionPiece improves both PRC and success. ActionPiece achieves

Table 6: **Training-objective variants of standard RVQ.** Each "+" row adds only the named objective to standard RVQ. PRP and QR denote physical rank preservation and quantization regularization. Policies use Qwen3-VL-4B with 30K training steps. Success rates are in percent; MSE is scaled by 10^4. Best values are bold, determined before rounding.

Variant	MSE ↓	PRC ↑	LIBERO Overall ↑	LIBERO-Plus Overall ↑
Standard RVQ	4.06	0.902	90.9	60.4
+ SIGReg	3.77	0.950	93.3	66.5
+ Neighborhood	5.09	0.889	94.3	63.5
+ Temporal	3.83	0.924	94.8	66.1
+ PRP	3.63	0.947	93.8	65.4
+ QR	3.92	0.916	92.9	62.4
ActionPiece (PRP + QR)	**3.32**	**0.953**	**94.8**	**68.8**

the lowest MSE, highest PRC, and highest overall success on both benchmarks among the variants shown.

D Implementation Details

ActionPiece configuration. We use the same ActionPiece configuration for LIBERO, LIBERO-Plus, VLA-Arena, and SimplerEnv, as summarized in Table 7. With S latent slots and Q residual quantization levels, each slot produces Q code indices, giving $n = SQ$ output tokens. Our configuration uses $S = 16$ and $Q = 1$, hence $n = 16$. Other tokenizers use the configurations specified in their original papers or released codebases. We set $\lambda = 0.25$ in Eq. (7), assigning weights of 0.25 and 0.75 to encoder and quantized feature distances, respectively.

Table 7: **ActionPiece configuration and objective weights.**

Setting	Value
Action chunk length H	8
Output tokens n	16
Latent slots S	16
RVQ levels Q	1
Codebook size K	512
Reconstruction MSE weight	1
Commitment weight λ_{vq}	0.25
PRP weight λ_r	6.25×10^{-4}
QR weight λ_q	6.25×10^{-4}

D.1 Differentiating Codeword Selection

With $Q = 1$, each of the 16 latent slots selects one codeword. Let $u_{i,s}$ be the projected encoder feature and $\Delta_{i,s,c} = \|u_{i,s} - e_c\|_2^2$ its squared distance to codebook entry c. Hard assignment selects a code; to differentiate the representation distance, we use a straight-through estimator with soft codeword probabilities (van den Oord et al., 2017; Jang et al., 2016). Let $e_{i,s,\text{hard}}$ be the selected code contribution after the quantizer output projection, and let $e_{i,s,\text{soft}}$ be the output projection of the soft codebook average. Here sg denotes stop-gradient. With $s_{i,s} = \max\{\text{sg}[\text{Std}_c(\Delta_{i,s,c})], 10^{-6}\}$, we form

$$\pi_{i,s,c} = \text{softmax}_c\left(-\frac{\Delta_{i,s,c}}{s_{i,s}\tau_q}\right), \qquad \widetilde{e}_{i,s} = e_{i,s,\text{soft}} + \text{sg}[e_{i,s,\text{hard}} - e_{i,s,\text{soft}}]\,. \tag{15}$$

We use $\tau_q = 1$ and sample no Gumbel noise. Thus $\widetilde{e}$ has the selected hard code contribution as its forward value and the soft assignment for gradient computation. The reconstruction objective uses

an identity straight-through estimator, and the codebook is updated by exponential moving average (EMA).

D.2 Multiple Quantization Levels

In our experiments, $Q = 1$ and $Q = 2$ yield similar downstream policy performance. We use $Q = 1$ for the final ActionPiece configuration. For $Q > 1$, we apply the same differentiable codeword selection at each residual level. Let $q_{i,s}^{(r)} = \sum_{\ell=1}^{r} \widetilde{e}_{i,s}^{(\ell)}$ be the cumulative contribution after the corresponding output projections, and let $\bar{q}_{i,s}^{(r)}$ be normalized using LayerNorm followed by unit ℓ_2 normalization. We extend Eq. (6) by averaging over the partial reconstructions at all residual levels:

$$d^2_{\text{quant}}(i,j) = \frac{1}{Q}\sum_{r=1}^{Q}\frac{1}{S}\sum_{s=1}^{S}\left\|\bar{q}_{i,s}^{(r)} - \bar{q}_{j,s}^{(r)}\right\|_2^2. \tag{16}$$

At $Q = 1$, this expression reduces to Eq. (6).

D.3 Physical-Distance Weights and Normalization

Training-set scales. Let $p = (p_x, p_y, p_z)$ denote the translational command components and $r = (r_x, r_y, r_z)$ the three rotational components in the action representation used by the tokenizer. We compute their marginal standard deviations over training action time steps and set

$$s_p = (\sigma_{p_x}, \sigma_{p_y}, \sigma_{p_z}), \qquad s_R = \sqrt{\sigma_{r_x}^2 + \sigma_{r_y}^2 + \sigma_{r_z}^2}. \tag{17}$$

Here s_p is a three-dimensional vector, and division by s_p is elementwise; s_R is a scalar. Rotational components are expressed in radians when computing these statistics. Rotations are mapped to valid elements of $\mathrm{SO}(3)$ before distance computation. The rotational distance itself is the shortest geodesic angle between the corresponding rotation matrices, $\theta = d_{\mathrm{SO}(3)}(R, R') \in [0, \pi]$, rather than a componentwise difference between rotational coordinates. All scales are computed from training data and held fixed during subsequent training and evaluation; they are not learnable parameters.

Weights and averaging. The implementation averages the normalized squared translation differences over the three coordinates. Accordingly, the weights in Eq. (5), which writes translation as a squared Euclidean norm, are

$$\alpha_p = \frac{1}{3}, \qquad \alpha_R = 1, \qquad \alpha_g = \frac{1}{4}. \tag{18}$$

The factor $1/3$ averages the three normalized squared differences in x, y, and z. The rotation term uses unit weight after normalization by s_R. In our experiments, binary gripper states are encoded as $\{-1, +1\}$, so the weight $1/4$ assigns unit cost to an open–closed mismatch:

$$\frac{1}{4}(g - g')^2 = \begin{cases} 0, & g = g', \\ 1, & g \neq g', \end{cases} \qquad g, g' \in \{-1, +1\}. \tag{19}$$

Thus, the gripper term in Eq. (5) directly measures binary state disagreement without an additional scale parameter. Explicitly, the single-arm distance with binary gripper commands is

$$\delta_{\text{phys}}(a, a') = \frac{1}{3}\sum_{k\in\{x,y,z\}}\left(\frac{p_k - p'_k}{\sigma_{p_k}}\right)^2 + \left(\frac{\theta}{s_R}\right)^2 + \frac{1}{4}\,|g - g'|^2. \tag{20}$$

For two chunks of length H, we average corresponding time-step distances:

$$d_{\text{phys}}(A, A') = \frac{1}{H}\sum_{t=1}^{H}\delta_{\text{phys}}(a_t, a'_t). \tag{21}$$

Continuous gripper commands. We map continuous widths to $\tilde{g} = 2(g - g_{\min})/(g_{\max} - g_{\min}) - 1$, where $g_{\min}$ and $g_{\max}$ are the calibrated fully closed and fully open widths. The same gripper term becomes $\frac{1}{4}|\tilde{g} - \tilde{g}'|^2 = ((g - g')/(g_{\max} - g_{\min}))^2$; binary commands are its endpoint case.

Table 8: **Full VLA-Arena results** (Zhang et al., 2025). SR is success rate (%); CC is cumulative cost (lower is better). Avg equally weights the 11 suites. For each task and difficulty level, the highest SR across the six models is bold, including ties; the same rule applies to Avg. CC values are not bold.

Task	Metric	$\pi_{0.5}$			GR00T-N1.6			Qwen-GR00T			Evo-Depth			Motus			**ActionPiece**		
		L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2
Safety																			
Static obstacles	SR ↑	90.0	62.0	40.0	72.0	30.0	14.0	**91.0**	18.0	5.0	88.0	66.0	48.0	64.0	67.0	41.0	**91.0**	**83.0**	**57.0**
	CC ↓	0.0	33.3	76.6	0.0	11.2	38.4	0.0	9.1	29.6	0.0	8.7	14.9	0.0	72.0	124.3	0.0	14.5	30.3
Cautious grasp	SR ↑	50.0	14.0	0.0	16.0	2.0	0.0	77.0	8.0	2.0	**78.0**	24.0	0.0	62.0	**25.0**	**11.0**	77.0	18.0	1.0
	CC ↓	5.0	5.5	1.2	9.6	41.2	10.4	2.4	98.2	11.1	0.0	11.4	26.3	3.9	91.0	9.8	1.1	19.3	12.4
Hazard avoidance	SR ↑	58.0	30.0	36.0	64.0	4.0	10.0	67.0	15.0	19.0	40.0	0.0	14.0	65.0	**34.0**	21.0	**68.0**	22.0	**46.0**
	CC ↓	7.1	15.0	14.5	6.1	20.2	17.9	7.0	19.3	19.5	7.4	25.9	7.9	15.6	40.2	51.3	0.7	4.0	1.7
State preservation	SR ↑	58.0	56.0	54.0	66.0	50.0	38.0	86.0	56.0	47.0	88.0	66.0	56.0	63.0	63.0	**72.0**	**94.0**	**80.0**	64.0
	CC ↓	0.0	5.6	20.8	0.0	5.0	10.4	0.0	5.6	15.7	0.0	2.3	5.1	0.0	6.3	16.0	0.0	6.9	25.3
Dynamic obstacles	SR ↑	50.0	44.0	22.0	74.0	50.0	2.0	81.0	56.0	3.0	**82.0**	60.0	6.0	58.0	33.0	11.0	**82.0**	**76.0**	**27.0**
	CC ↓	2.4	8.8	5.7	5.7	7.3	56.8	6.0	8.3	2.7	0.0	3.4	12.7	6.5	47.9	7.2	4.6	21.2	31.0
Distractor																			
Static distractors	SR ↑	88.0	16.0	16.0	46.0	**32.0**	6.0	91.0	6.0	2.0	**94.0**	20.0	**24.0**	69.0	19.0	13.0	92.0	21.0	22.0
Dynamic distractors	SR ↑	80.0	66.0	**54.0**	70.0	**72.0**	18.0	**92.0**	49.0	25.0	86.0	60.0	32.0	77.0	51.0	26.0	90.0	63.0	43.0
Extrapolation																			
Prep. combinations	SR ↑	62.0	**24.0**	**6.0**	48.0	0.0	0.0	51.0	1.0	0.0	66.0	0.0	0.0	41.0	7.0	1.0	**69.0**	15.0	0.0
Task workflows	SR ↑	38.0	20.0	**22.0**	42.0	0.0	0.0	51.0	3.0	9.0	32.0	0.0	0.0	38.0	25.0	14.0	**64.0**	**33.0**	21.0
Unseen objects	SR ↑	48.0	60.0	20.0	26.0	18.0	16.0	63.0	46.0	26.0	**78.0**	52.0	4.0	63.0	**65.0**	19.0	**78.0**	59.0	**43.0**
Long Horizon																			
Long horizon	SR ↑	85.0	0.0	0.0	29.0	2.0	0.0	96.0	0.0	0.0	93.0	0.0	0.0	61.0	**4.0**	**1.0**	**99.0**	0.0	0.0
Avg	SR ↑	64.3	35.6	24.5	50.3	23.6	9.5	76.9	23.5	12.5	75.0	31.6	16.7	60.1	35.7	20.9	**82.2**	**42.7**	**29.5**

Dual-arm extension. For dual-arm actions, each arm has its own translation and rotation scales, computed from its respective training action components. Continuous gripper width differences are normalized by each arm's calibrated range $g^{(b)}_{\max} - g^{(b)}_{\min}$, for $b \in \{L, R\}$. The per-step distance averages the two arms:

$$\delta^{\text{dual}}_{\text{phys}}(a, a') = \frac{1}{2} \sum_{b \in \{L,R\}} \left[\frac{1}{3} \left\| \frac{p^{(b)} - p'^{(b)}}{s^{(b)}_p} \right\|^2_2 + \left(\frac{\theta^{(b)}}{s^{(b)}_R} \right)^2 + \left(\frac{g^{(b)} - g'^{(b)}}{g^{(b)}_{\max} - g^{(b)}_{\min}} \right)^2 \right], \tag{22}$$

where $\theta^{(b)} = d_{\text{SO}(3)}(R^{(b)}, R'^{(b)})$. Chunk distances then use the same temporal average as above.

D.4 Physical-Neighbor Selection

For each anchor in the training batch, we exclude the anchor itself and rank the remaining action chunks by increasing physical distance. We select the action at the 5th percentile ($q_{0.05}$) as the positive (near) example and the action at the 95th percentile ($q_{0.95}$) as the negative (far) example. These percentiles are computed separately for each anchor over the other actions in its minibatch. Both physical rank preservation and quantization regularization use this same pair. The former compares representation distances D_{ij}, while the latter compares codeword probabilities using Jensen–Shannon divergence. We use natural logarithms without additional normalization for the divergence. The discrete action vocabulary and decoding procedure are unchanged by the two training objectives.

E Full VLA-Arena Results

Table 8 provides the complete VLA-Arena results, including success rates (SR) and cumulative costs (CC) for safety tasks. Lower CC indicates lower cumulative safety cost.