VLA-COLLECT: TRAINING VISION LANGUAGE ACTION MODELS WITH FEW-SHOT DATA VIA COLLECTIVE ACTION FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models unify perception, language, and action and achieve strong performance across robotic tasks, but high-fidelity dexterous manipulation often still relies on costly supervised fine-tuning (SFT), pushing deployment into a few-shot regime. With few demonstrations, a policy can overfit to narrow behaviors and encounter unfamiliar states during execution. When self-generated rollouts are reused for training, low-quality or repetitive trajectories can reinforce execution errors and limited coverage. We propose Collective Action Fine-Tuning (CoAct), a collective-learning paradigm for interactive few-shot VLA post-training using limited initial expert demonstrations, additional rollouts, and binary task-completion feedback. CoAct maintains a diverse policy pool with two mechanisms: WeightCoAct encourages action-head discrepancy, while DataCoAct assigns persistent expert-specific visual views to broaden the experience generated by the pool. CoAct further introduces Collective Trajectory Curation (CoTC) to select successful rollouts and favor experience beyond the initial demonstration coverage, then distills curated trajectories into a unified student policy. Experiments on LIBERO, LIBERO-Plus, and RoboTwin, complemented by five real-world tasks, show improved few-shot performance and zero-shot transfer.

1 INTRODUCTION

Vision-Language-Action (VLA) models have made remarkable progress in robot learning by unifying perception, language, and action in a single policy model, achieving strong performance in various robot tasks (Brohan et al., 2023b;a; Kim et al., 2024; Zhang et al., 2025b; Zhou et al., 2025; Li et al., 2025b; Liu et al., 2025a; Fan et al., 2026; Liang et al., 2026; Li et al., 2026a; Ye et al., 2026; Yang et al., 2026a). However, for dexterous manipulation, pretraining alone is insufficient, where high performance still hinges on post-training a VLA policy with supervised fine-tuning (SFT) using task-relevant trajectories that capture subtle contact transitions, compliant interaction, and recovery behaviors (Kim et al., 2025; Zhao et al., 2023; Chi et al., 2024; Li et al., 2025c; Xiao et al., 2026; Yang et al., 2026b; Shi et al., 2026; Guo et al., 2026; Zhang et al., 2026a). On the other hand, a key barrier is the high cost of SFT data: dexterous demonstrations require careful teleoperation, perturbation trials, and long-horizon recovery, making large-scale collection difficult, expensive and time consuming (Brohan et al., 2023b; Zhao et al., 2023; Zheng et al., 2026; Zhang et al., 2026b;c; Li et al., 2026b). As a result, many application settings operate in a few-shot SFT regime, where only a small number of expert trajectories are available per task (Du et al., 2023; Lin et al., 2024; Li et al., 2025a; Sridhar et al., 2025). Thus, how to achieve reliable SFT of VLA models under limited data is both crucial for efficient and effective deployment and a challenging problem.

With limited trajectories, a policy may learn non-robust correlations and become brittle outside the demonstrated states. Small execution errors can then move it into unfamiliar states and compound over time (Ross et al., 2011). In rollout-augmented self-training, where the policy's executions become candidates for subsequent supervision, retaining low-quality or repetitive trajectories can reinforce execution errors and limited coverage.

To handle this issue, various data-efficient methods for VLA models have been explored. In-context learning (Ma et al., 2024; Sridhar et al., 2025; Yin et al., 2025) avoids weight updates but is limited

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.

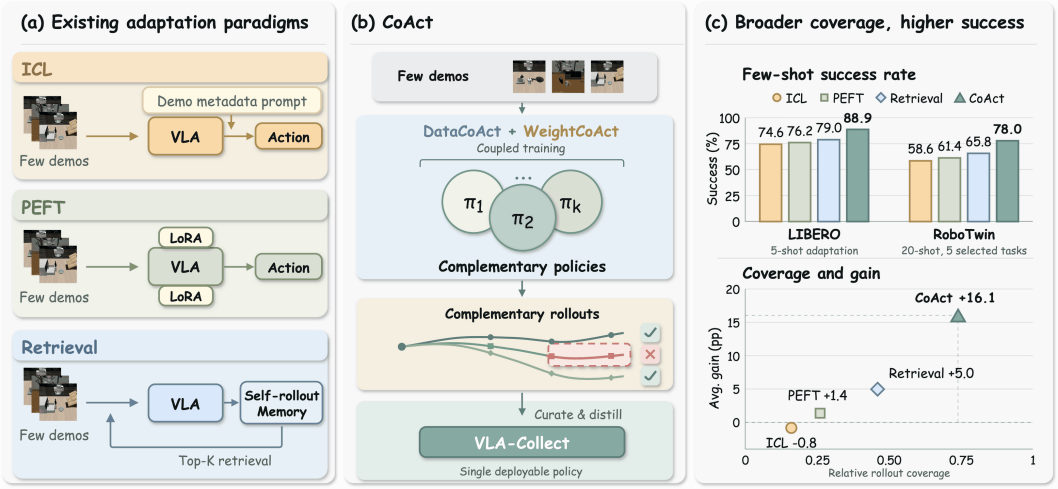


Figure 1: Overview of CoAct for few-shot VLA fine-tuning. ICL, PEFT, and retrieval-based training adapt one policy. CoAct trains complementary experts with WeightCoAct and DataCoAct, curates rollouts, and distills one deployment model. The lower panel shows dataset coverage and mean gains on LIBERO and RoboTwin; full dataset coverage is 1 (Appendix D.1).

by context size and struggles with long-horizon control. Parameter-efficient (Grover et al., 2025; Hu et al., 2022; 2024; Liang et al., 2022; Liu et al., 2023b) and specialized methods (Li et al., 2025a; Song et al., 2025) reduce trainable parameters but remain constrained by sparse dexterous data and can yield brittle policies. Meta-learning (Finn et al., 2017; Ghadirzadeh et al., 2021; Yu et al., 2018) and behavior retrieval (Du et al., 2023; Lin et al., 2024; Xie et al., 2025) make use of prior experience, yet depend on large, well-aligned past data and whether such priors cover the target task. These approaches improve adaptation under sparse demonstrations, but single-policy rollout-augmented training still draws its candidate supervision from one policy distribution. This motivates complementary experience generation when additional interactions and task-completion feedback are available.

We address this gap by exploring the core principle behind collective learning: under limited supervision, maintaining multiple complementary predictors can reduce overfitting and stabilize learning by preventing any single hypothesis from dominating. This motivates a shift from refining one policy model in isolation to collectively exploring and curating trajectories with multiple diverse VLA models, then consolidating their complementary skills into a single deployable model.

To this end, we propose Collective Action Fine-Tuning (CoAct), a collective learning paradigm for few-shot SFT of VLA models. CoAct works by enforcing multiple VLA models to learn from multiple distinct yet complementary views of few-shot VLA data, which thus allow distinct yet complementary trajectory generation and training. It induces diversity through two mechanisms: (1) Weight-Discrepancy CoAct (WeightCoAct), which encourages distinct policy parameters by separating expert action heads from a shared reference and from one another, so common deviations alone do not yield pairwise discrepancy; and (2) Data-Discrepancy CoAct (DataCoAct), which diversifies policy models by training them using different transformed views of the same demonstrations and curated rollouts. Together, they create various complementary "views", enabling multiple heterogeneous policy models to generate diverse, high-quality trajectories for robust few-shot post-training.

CoAct further combines collective exploration from diverse experts with Collective Trajectory Curation (CoTC) to improve the quality and coverage of rollout supervision. CoTC screens the joint expert rollout pool using task-completion feedback to exclude failed executions from subsequent supervision. Among successful executions, it uses observable trajectory statistics and novelty relative to the demonstrations to prioritize informative experience beyond their initial coverage. Finally, CoAct consolidates the curated trajectories into a unified student VLA model, termed "VLA-Collect", yielding a stable closed-loop post-training procedure. Figure 1 summarizes this contrast between single-policy few-shot adaptation and the proposed collective action fine-tuning framework.

The contributions of this work are fourfold. (1) We propose CoAct, a collective-learning-inspired method for few-shot SFT of VLA models that leverages multiple complementary experts for rollout generation and curation, and consolidates them into a unified student policy, termed VLA-Collect. (2) We design WeightCoAct and DataCoAct to counter expert homogenization in few-shot training through explicit separation of expert action heads and persistent visual views, encouraging complementary policies for broader rollout coverage. (3) We develop CoTC to construct reliable collective supervision by filtering unsuccessful rollouts and prioritizing informative experience beyond the initial demonstration coverage. (4) We conduct extensive experiments across diverse tasks and robot configurations, demonstrating consistent improvements in robustness and generalization over standard few-shot post-training baselines.

2 Related Works

2.1 Vision-Language-Action Models

Vision-Language-Action (VLA) models are the dominant paradigm for robot learning across diverse tasks, unifying multimodal observations and language instructions into a single policy to generate robot actions. Early milestones (e.g., RT-1/RT-2 (Brohan et al., 2023b;a), ACT (Zhao et al., 2023)) are built on imitation learning from teleoperation. Subsequent works expand along two core lines: tokenized action prediction with VLM backbones (e.g., OpenVLA (Kim et al., 2024)), and continuous action generation through diffusion (e.g., Diffusion Policy (Chi et al., 2024), RDT (Liu et al., 2025b)). Recent advances further integrate VLM semantic grounding with precise control, including the π_0 series (Black et al., 2026; Shukor et al., 2025; Intelligence et al., 2025; 2026) and hierarchical frameworks decoupling abstract reasoning from motor control (e.g., Hi-robot (Shi et al., 2025), Gr00t (NVIDIA et al., 2025)). Despite these advances, fine-tuning leading VLA models still requires large demonstration datasets for each task, especially for dexterous manipulation involving frequent contact. This makes adapting pretrained VLAs with few demonstrations a critical open challenge, which motivates our work.

2.2 Training with Limited Data

Learning with limited data, widely studied to address high annotation costs and few-shot pretrained model adaptation (Wang et al., 2020; Gharoun et al., 2024; Zhang et al., 2025a; Yang et al., 2022), faces amplified challenges in VLA post-training: trajectories for each task require costly expert teleoperation, and interactive policy learning inherently suffers from overfitting, covariate shift, and compounding execution errors as the policy shapes its visited state distribution. Existing methods for VLA adaptation with limited data fall into three core lines: (1) In-context learning, which avoids weight updates but struggles with control over long horizons (Ma et al., 2024; Sridhar et al., 2025; Yin et al., 2025); (2) Parameter-efficient fine-tuning (PEFT, e.g., LoRA), which mitigates overfitting by reducing trainable parameters (Grover et al., 2025; Hu et al., 2022; 2024); (3) Meta-learning/behavior retrieval, which leverages prior experience but relies on large datasets aligned with the target task (Finn et al., 2017; Ghadirzadeh et al., 2021; Du et al., 2023; Xie et al., 2025). When these adaptation methods collect rollouts and retrain a single policy, candidate supervision remains limited by that policy's behavioral coverage.

2.3 Collective Learning

Collective learning leverages multiple complementary models to boost generalization and sample efficiency under limited supervision. Its core paradigm, co-training first proposed by Blum & Mitchell (1998) (with later extensions (SUN & JIN, 2011; Yu et al., 2011)), learns complementary hypotheses via complementarity across data views, outperforming single predictors in settings with limited data. Extensions to deep learning include co-training with multiple networks, generative co-training (e.g., GenCo (Cui et al., 2021)), and ensemble sequential decision making (Qiao et al., 2018; Foerster et al., 2016). These works suggest that maintaining multiple hypotheses and exploiting complementary views can stabilize learning when supervision is scarce or biased. We use this principle to train VLA-Collect: CoAct couples heterogeneous experience generation with curation of successful trajectories and distills the resulting supervision into a single deployment policy. Appendix B.7 compares its design with related paradigms.

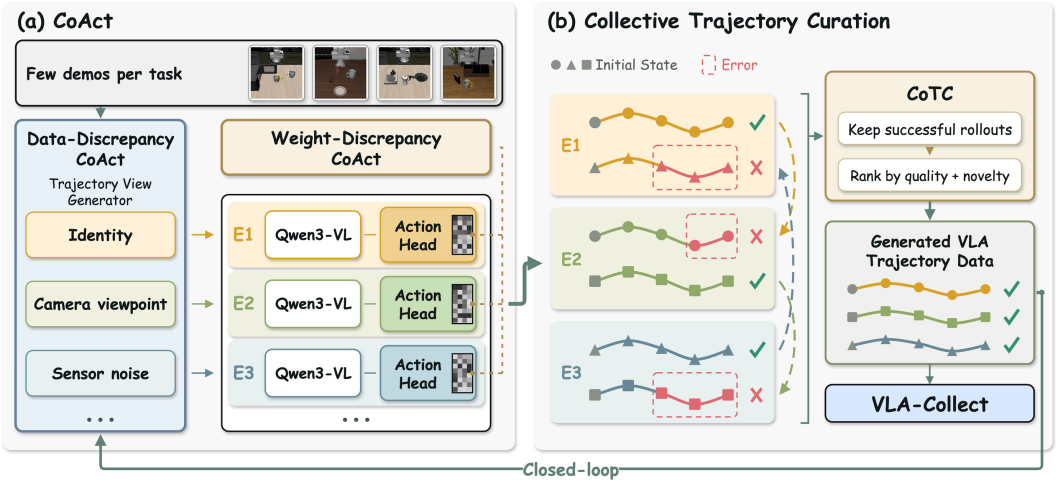


Figure 2: VLA-Collect consolidates collective training experience into a single deployment policy. CoAct encourages complementary experts through WeightCoAct and DataCoAct, then uses Collective Trajectory Curation (CoTC) to select successful rollouts. The curated trajectories and initial demonstrations support further expert adaptation and train VLA-Collect through behavior cloning.

3 VLA-COLLECT

VLA-Collect is a single deployable VLA policy trained through Collective Action Fine-Tuning (CoAct; Fig. 2). Its training separates the generation of experience from its consolidation: complementary experts collect trajectories, and a student learns from their curated executions. WeightCoAct and DataCoAct encourage expert complementarity, while Collective Trajectory Curation (CoTC) selects successful experience for further expert adaptation and final student training.

3.1 PROBLEM SETTING

A VLA policy π_θ predicts actions from visual observations and a language instruction. Starting from a pretrained model and a small set $\mathcal{D}_T$ of expert trajectories paired with instructions, we adapt the policy by supervised fine-tuning. We denote the standard objective for behavior cloning on a dataset $\mathcal{D}$ by $\mathcal{L}_{\text{SFT}}(\theta; \mathcal{D})$, using the model's imitation loss on paired observations and actions. Our setting permits additional environment interactions and binary feedback on task completion, but no new expert action annotations or corrective demonstrations. Few-shot therefore refers to the initial expert demonstrations; Appendix A separately reports interaction and retention budgets.

Few-shot policies may overfit narrow behaviors and encounter unfamiliar states; reusing their rollouts for supervision can reinforce coverage gaps and execution errors. CoAct improves VLA-Collect through collective experience while retaining one policy at deployment.

Curation improves reliability but cannot supply behaviors absent from a single policy's rollout pool. Conversely, unfiltered exploration can introduce failures into supervision. CoAct couples complementary experts that expand candidate behaviors with curation of successful training targets.

3.2 WEIGHTCOACT: ENCOURAGING POLICY HETEROGENEITY

Training multiple experts on the same scarce demonstrations can reproduce similar policy hypotheses. WeightCoAct encourages separation of expert action heads from a fixed anchor and from one another alongside the imitation objective. We initialize $K \geq 2$ experts from a pretrained VLA model; θ_k includes the trainable backbone and action head, with θ_k^a denoting the latter. For action

heads u, v, define their distance and the pool discrepancy, with $\Theta^a = \{\theta_k^a\}_{k=1}^K$ and fixed anchor $\bar{\theta}^a$:

$$d_{\mathrm{act}}(u, v) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \frac{\|u_p - v_p\|_2}{\sqrt{N_p}},$$
$$\mathcal{R}_{\mathrm{W}}(\Theta^a) = \frac{1}{K} \sum_{k=1}^{K} d_{\mathrm{act}}(\theta_k^a, \bar{\theta}^a) + \frac{2}{K(K-1)} \sum_{k<j} d_{\mathrm{act}}(\theta_k^a, \theta_j^a). \tag{1}$$

Here $\mathcal{P}$ indexes action head tensors and N_p is the size of tensor p. The two terms average anchor distances and distances over all distinct expert pairs. Identical experts receive zero pairwise discrepancy even when far from the anchor, so the second term explicitly discourages parameter collapse. Parameter separation need not diversify behavior; DataCoAct adds complementary visual views.

3.3 DataCoAct: Specializing Experts through Visual Views

DataCoAct makes visual experience a persistent source of expert specialization. A Trajectory View Generator applies transformations $\mathcal{T}_m$ to visual observations while preserving instructions, action labels, and trajectory timing. The views cover original observations and variations in camera viewpoint, illumination, scene appearance, and sensor noise (Appendix B.5). Expert k retains its assigned configuration $m(k)$ throughout training, adapting its visual representations and action predictions to that observation condition. Whereas pooling all views asks one predictor to accommodate their variation, persistent assignment gives each expert a distinct observation distribution. Their rollouts test whether this specialization yields complementary task solutions.

At round i, the training set is $\mathcal{D}^{(i)} = \mathcal{D}_T \cup \mathcal{D}_R^{(i)}$, where $\mathcal{D}_R^{(i)}$ contains the curated rollouts available for that round and $\mathcal{D}_R^{(1)} = \varnothing$. The joint expert objective is

$$\min_{\{\theta_k\}_{k=1}^K} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{\mathrm{SFT}}\Big(\theta_k; \mathcal{T}_{m(k)}(\mathcal{D}^{(i)})\Big) - \lambda \mathcal{R}_{\mathrm{W}}(\Theta^a). \tag{2}$$

The imitation term updates both the backbone and action head for task fidelity under each assigned view. With weight λ, WeightCoAct encourages separation from the anchor and between experts, backpropagating each pairwise distance through both action heads while keeping the anchor fixed. Together, the two mechanisms encourage heterogeneous policies that can expand rollout coverage.

3.4 CoTC: Constructing Reliable Collective Supervision

Broader exploration alone does not ensure useful supervision: expert rollouts may fail or repeat behaviors already covered by the demonstrations. CoTC therefore separates task validity from the informativeness of successful experience. For each task, let $\mathcal{D}_{R,\mathrm{raw}}$ be the joint expert rollout pool and $c(\tau) \in \{0, 1\}$ the task completion label of trajectory τ. CoTC first excludes unsuccessful executions, then ranks successful ones using recorded gripper motion, trajectory length, and novelty in feature space relative to the seed demonstrations. Using $q(\tau)$ (Appendix B.6), CoTC selects

$$\mathcal{D}_R = \mathrm{Top}_{b,q}(\{(\tau, l) \in \mathcal{D}_{R,\mathrm{raw}} : c(\tau) = 1\}), \tag{3}$$

where $\mathrm{Top}_{b,q}$ retains the successful trajectories with the highest scores under the round's retention budget b. The success gate controls task validity, and novelty favors experience beyond the initial demonstration coverage. Novelty is considered only among successful executions, so visual difference alone cannot admit a failed rollout. Here, novelty measures departure from the seed demonstrations; it does not by itself establish task difficulty or improvement over earlier collection rounds. Scores determine selection only; retained trajectories use the standard, unweighted imitation loss.

In simulation, completion comes from the benchmark success condition; in real-world experiments, it is supplied by a human completion judgment. All ranking statistics use recorded visual observations and robot proprioception, without dense rewards, hidden simulator states, or corrective actions. By operating on the joint expert pool, CoTC turns complementary executions into shared supervision for expert adaptation and the VLA-Collect student.

3.5 Training and Deploying VLA-Collect

To train VLA-Collect, CoAct alternates expert adaptation, rollout collection, and trajectory curation. Each round updates the expert backbones and action heads using Eq. (2). When a collection phase follows, the experts execute tasks, and Eq. (3) selects experience for the next round's training data. The seed demonstrations remain available in every round, the reference action head $\bar{\theta}^a$ stays fixed, and each expert retains its view assignment. The resulting feedback allows the experts to learn from successful experience beyond the initial demonstrations while filtering failed executions before they become training targets. Collection ends before the final training round, following Appendix A. This curated experience supervises subsequent experts and the VLA-Collect student.

Finally, we train the VLA-Collect student π_ϕ on the final dataset of demonstrations and rollouts using $\mathcal{L}_{\mathrm{SFT}}$, the objective for behavior cloning. This distillation transfers the experts' executed behaviors through paired observations and actions. At deployment, only VLA-Collect predicts actions, so execution requires neither an active expert pool nor aggregation of multiple action predictions.

4 Experiments

We evaluate VLA-Collect in simulation and real-world manipulation, then analyze the contributions of CoAct and the supervision it produces. We report success rates (%); detailed protocols, interaction budgets, and supplementary results appear in Appendices B, A, and C.

4.1 Simulation Experiments

4.1.1 LIBERO and LIBERO-Plus

We evaluate VLA-Collect with the Qwen-OFT 2B backbone on LIBERO (Liu et al., 2023a), using five initial expert demonstrations per task. CoAct additionally uses 100 candidate simulator rollouts per task and binary feedback on task completion, without additional expert action annotations (Appendices B.1 and A). Table 1 compares representative VLA backbones and adaptation baselines using Qwen-OFT. VLA-Collect achieves the best performance across all four suites, with an average success rate of 88.9%. It exceeds Qwen-OFT by 13.7 percentage points and VLA-Retrieval, our single-policy self-training baseline, by 9.9 points. It also achieves a higher reported success rate than larger baselines such as $\pi_{0.5}$ under the evaluated few-shot protocols.

Table 1: LIBERO success rates (%) with five initial demonstrations per task. Methods using Qwen-OFT share a 2B backbone. Superscripts show gains over Qwen-OFT in percentage points (pp).

Algorithm	Size	Spatial	Object	Goal	Long	Avg.
OpenVLA (Kim et al., 2024)	7B	14.6	32.2	43.0	1.2	22.8
OpenVLA-OFT (Kim et al., 2025)	7B	73.4	81.6	47.4	63.4	66.5
π_0 (Black et al., 2026)	3B	63.2	45.8	49.0	16.4	43.6
π_0-FAST (Pertsch et al., 2025)	3B	79.0	89.2	69.4	33.2	67.7
$\pi_{0.5}$ (Intelligence et al., 2025)	3B	88.2	86.8	79.0	60.4	78.6
GR00T-N1 (NVIDIA et al., 2025)	3B	60.0	75.0	40.8	38.2	53.5
Qwen-OFT (StarVLA Community, 2026)	2B	87.0	76.0	70.4	67.2	75.2
VLA-ICL (Sridhar et al., 2025)	2B	86.4	75.2	70.8	65.8	74.6
VLA-PEFT (Hu et al., 2022)	2B	87.8	78.4	72.0	66.6	76.2
VLA-Retrieval (Lin et al., 2024)	2B	89.0	83.6	74.8	68.4	79.0
VLA-Collect	2B	**93.8**$^{\uparrow 6.8}$	**95.0**$^{\uparrow 19.0}$	**86.6**$^{\uparrow 16.2}$	**80.0**$^{\uparrow 12.8}$	**88.9**$^{\uparrow 13.7}$

We evaluate zero-shot transfer to LIBERO-Plus (Fei et al., 2025) without LIBERO-Plus demonstrations, rollouts, or fine-tuning. Distribution shift lowers performance across methods (Table 2), while VLA-Collect retains the best average success rate of 54.3%, exceeding Qwen-OFT by 13.1 points and VLA-Retrieval by 10.7 points. These comparisons establish performance under the evaluated protocols; the controls with matched interaction and retention budgets used to assess CoAct's contributions are analyzed in Section 4.3.1 and Appendix A.1.

Table 2: Zero-shot success rates (%) on LIBERO-Plus for policies trained on LIBERO. Superscripts show gains (pp) over Qwen-OFT.

Algorithm	Size	Spatial	Object	Goal	Long	Avg.
OpenVLA (Kim et al., 2024)	7B	6.5	8.6	10.8	0.4	6.6
OpenVLA-OFT (Kim et al., 2025)	7B	44.6	51.8	21.7	29.4	36.9
π_0 (Black et al., 2026)	3B	40.6	27.8	22.6	7.8	24.7
π_0-FAST (Pertsch et al., 2025)	3B	50.7	54.2	32.1	15.8	38.2
$\pi_{0.5}$ (Intelligence et al., 2025)	3B	56.6	52.7	36.5	28.8	43.7
GR00T-N1 (NVIDIA et al., 2025)	3B	34.5	42.1	17.4	15.8	27.5
Qwen-OFT (StarVLA Community, 2026)	2B	58.1	45.2	29.5	31.9	41.2
VLA-ICL (Sridhar et al., 2025)	2B	57.6	44.6	29.8	31.0	40.8
VLA-PEFT (Hu et al., 2022)	2B	58.6	46.6	30.4	31.6	41.8
VLA-Retrieval (Lin et al., 2024)	2B	59.6	49.8	32.2	32.6	43.6
VLA-Collect	2B	**67.4**$^{\uparrow 9.3}$	**60.8**$^{\uparrow 15.6}$	**45.9**$^{\uparrow 16.4}$	**43.1**$^{\uparrow 11.2}$	**54.3**$^{\uparrow 13.1}$

4.1.2 RoboTwin

We further evaluate VLA-Collect on RoboTwin (Chen et al., 2025). Starting from Qwen-OFT with a Qwen3-VL-2B backbone trained on 20 clean demonstrations for each of all 50 tasks, we apply CoAct only to five tasks selected from those with initial success rates between 50% and 70%. Each policy is evaluated over 100 valid episodes per task in the clean environment with unseen instructions (Appendix B.8). Table 3 shows that VLA-Collect raises the mean success rate from 59.6% to 78.0%, a gain of 18.4 percentage points. VLA-Collect also outperforms VLA-Retrieval, our single-policy self-training baseline, on all five tasks, with a mean gain of 12.2 points.

Table 3: RoboTwin success rates (%) over 100 valid episodes per task. Avg. covers only the five selected tasks. Superscripts show gains (pp) over initial 20-shot SFT.

Method	Grab roller	Move playing card	Place can in basket	Press stapler	Stack three blocks	Avg.
Qwen-OFT (StarVLA Community, 2026)	70	60	61	54	53	59.6
VLA-ICL (Sridhar et al., 2025)	69	59	60	53	52	58.6
VLA-PEFT (Hu et al., 2022)	71	62	63	56	55	61.4
VLA-Retrieval (Lin et al., 2024)	75	66	67	61	60	65.8
VLA-Collect	**89**$^{\uparrow 19.0}$	**77**$^{\uparrow 17.0}$	**75**$^{\uparrow 14.0}$	**78**$^{\uparrow 24.0}$	**71**$^{\uparrow 18.0}$	**78.0**$^{\uparrow 18.4}$

4.2 Real-World Evaluation

We evaluate VLA-Collect and the baselines using the $\pi_{0.5}$ backbone on five physical tasks with ten initial demonstrations per task. VLA-Retrieval (single-policy self-training) and CoAct each use ten additional trajectories per task; CoAct shares this budget across its expert pool. CoTC uses human completion judgments; the full protocol is in Appendix B.3.

Table 4: Real-world success rates (%) over 20 trials per task. Avg. is the mean across five tasks. Superscripts show gains (pp) over VLA-Retrieval.

Method	Cloth folding	Table wiping	Pen insertion	Water pouring	Cube cleanup	Avg.
$\pi_{0.5}$ (Intelligence et al., 2025)	65	75	70	65	90	73.0
VLA-Retrieval	70	75	75	70	90	76.0
VLA-Collect	**75**$^{\uparrow 5.0}$	**90**$^{\uparrow 15.0}$	**85**$^{\uparrow 10.0}$	**80**$^{\uparrow 10.0}$	**100**$^{\uparrow 10.0}$	**86.0**$^{\uparrow 10.0}$

As shown in Table 4, VLA-Collect achieves an average success rate of 86.0%, exceeding VLA-Retrieval by 10.0 percentage points and the initial policy by 13.0 points. The improvement over VLA-Retrieval holds across all five tasks under matched initial demonstration budgets and matched

numbers of additional trajectories. These results provide preliminary evidence that collective trajectory generation and curation remain effective beyond simulation, using feedback on task completion and onboard observations. This study of five tasks does not establish broad real-world robustness or lower total collection cost than acquiring additional expert demonstrations.

4.3 Ablation and Analysis

4.3.1 Component Contributions and Student Distillation

We analyze CoAct using the Qwen-OFT 2B backbone and the LIBERO protocol (Appendix B.4). Table 5 shows that WeightCoAct and DataCoAct each improve performance on LIBERO and LIBERO-Plus, and their combination reaches 84.2% and 50.3%, respectively. Adding CoTC raises these results to 88.9% and 54.3%. Under matched interaction and retention budgets, single-policy self-training with CoTC achieves 81.4% on LIBERO, compared with 88.9% for full CoAct, a gain of 7.5 points (Appendix A.1). Together, these comparisons support the contributions of heterogeneous rollout generation and collective curation. The distilled VLA-Collect student also achieves 88.9% LIBERO success versus 85.1% for action ensembling of the same five experts, using one active policy at inference instead of five (Appendix C.3). This comparison supports retaining the benefits of collective experience in a single deployment policy.

Table 5: Component ablations: average success rates (%) on LIBERO and LIBERO-Plus. Superscripts show gains (pp) over the first row.

WeightCoAct	DataCoAct	CoTC	LIBERO Avg.	LIBERO-Plus Avg.
–	–	–	75.2	41.2
✓	–	–	$80.2^{\uparrow 5.0}$	$45.0^{\uparrow 3.8}$
–	✓	–	$81.4^{\uparrow 6.2}$	$46.9^{\uparrow 5.7}$
✓	✓	–	$84.2^{\uparrow 9.0}$	$50.3^{\uparrow 9.1}$
✓	✓	✓	**88.9**$^{\uparrow 13.7}$	**54.3**$^{\uparrow 13.1}$

4.3.2 Transferability of Collective Supervision

We test whether the collective supervision used to train VLA-Collect also benefits six additional VLA backbones (Appendix B.4). All six improve on both LIBERO and LIBERO-Plus when trained with the same dataset collected with CoAct (Figure 3). The gains show that the curated experience is useful across the evaluated model families, extending the benefit of collective supervision beyond the primary VLA-Collect backbone.

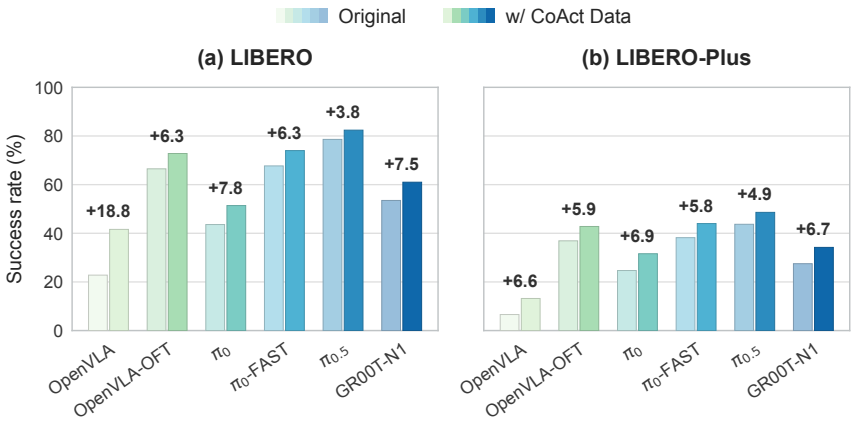


Figure 3: Data collected with CoAct improves all six evaluated backbones on LIBERO and LIBERO-Plus. Lighter bars show original models; darker bars show models trained with CoAct data. Annotations give gains in percentage points (pp). Bars show point estimates of success rate (%).

4.3.3 Qualitative Analysis of Rollout Coverage

Figure 4 compares supervision sources across ten LIBERO-Goal tasks. The selected Qwen-OFT videos occupy a concentrated band, whereas the CoAct pool spans a broader projected region. The two rightmost panels contain the same number of trajectories and frames, so their contrast is not simply a difference in sample count. Shared axes and a common projection support this comparison, while color locates each observation within its execution trajectory. The projection and the appearance sensitivity of CLIP limit this view to a descriptive comparison of successful rollouts, rather than a direct measure of behavioral coverage. Performance evidence comes from the controls under matched budgets (Appendix A.1).

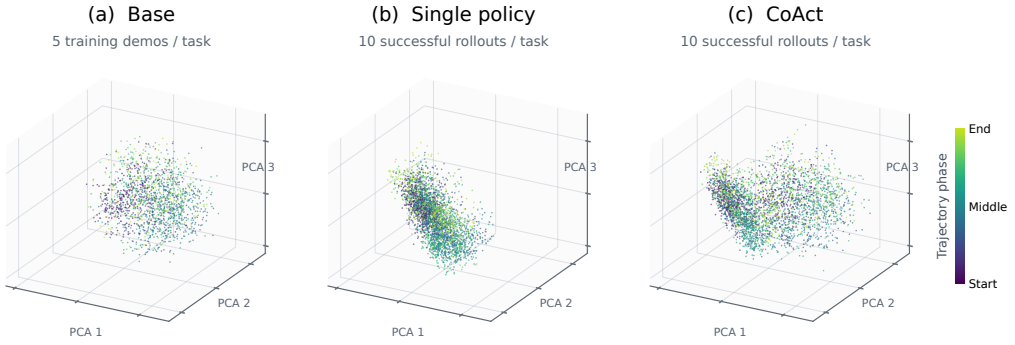


Figure 4: LIBERO-Goal distributions using 32 CLIP frames per trajectory and shared PCA; color denotes trajectory phase. The Qwen-OFT and CoAct panels each contain ten successful rollouts per task. See Appendix D.2.

Figure 5 shows the execution paths underlying supervision on plate pushing. Cyan denotes a single Qwen-OFT policy; purple and yellow denote CoAct experts. In this example, the single policy repeats similar approach directions and contact regions, whereas the expert pool exhibits more varied paths. This variation provides candidate experience for CoTC to curate before expert adaptation and student training. CoTC favors successful paths beyond the seed demonstration coverage. Appendix D presents additional task examples.

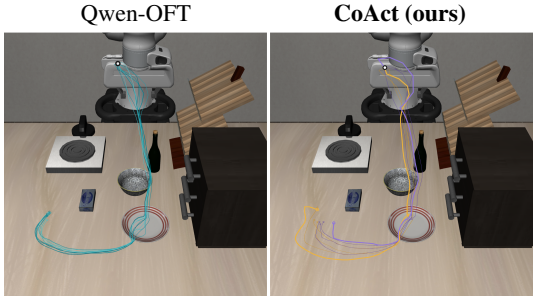


Figure 5: Rollouts for plate pushing.

5 Conclusion

We presented VLA-Collect, a VLA policy that consolidates collective experience into a single deployable model. Its CoAct training procedure diversifies experts through WeightCoAct and DataCoAct, curates their experience with Collective Trajectory Curation, and distills the retained behaviors into the student. With limited initial demonstrations and additional interactions, VLA-Collect improves few-shot performance and zero-shot transfer on LIBERO and LIBERO-Plus, with further evidence from five selected RoboTwin tasks and five real-world tasks. Comparisons under matched budgets support collective generation and curation, and student distillation retains these benefits in one deployable policy. The same curated data also improves six additional VLA backbones, supporting the reuse of collective supervision across different model families within the evaluated benchmarks. The evidence points to a practical direction for few-shot adaptation: improve which experience is collected and retained, then transfer it into an efficient deployment policy. Future work should examine longer collection horizons, broader task distributions, and noisier success feedback, while quantifying the full cost of collection and training.

AI USE STATEMENT

We used generative AI tools to assist with language editing and improve the clarity and readability of the manuscript. All AI-assisted content was reviewed and revised by the authors, who take full responsibility for the accuracy and integrity of the final manuscript.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pp. 92–100, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 1581130570. doi: 10.1145/279943.279962. URL https://doi.org/10.1145/279943.279962.

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, 2023a. URL https://arxiv.org/abs/2307.15818.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023b. URL https://arxiv.org/abs/2212.06817.

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

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, 2024. URL https://arxiv.org/abs/2303.04137.

Kaiwen Cui, Jiaxing Huang, Zhipeng Luo, Gongjie Zhang, Fangneng Zhan, and Shijian Lu. Genco: Generative co-training for generative adversarial networks with limited data, 2021. URL https://arxiv.org/abs/2110.01254.

Maximilian Du, Suraj Nair, Dorsa Sadigh, and Chelsea Finn. Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets. *arXiv preprint arXiv:2304.08742*, 2023.

Cunxin Fan, Xiaosong Jia, Yihang Sun, Yixiao Wang, Jianglan Wei, Ziyang Gong, Xiangyu Zhao, Masayoshi Tomizuka, Xue Yang, Junchi Yan, and Mingyu Ding. Interleave-VLA: Enhancing robot manipulation with image-text interleaved instructions. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ULTWUuGhC3.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Ji Li, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models, 2025. URL https://arxiv.org/abs/2510.13626.

Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pp. 357–368. PMLR, 2017.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning, 2016. URL https://arxiv.org/abs/1605.06676.

Ali Ghadirzadeh, Xi Chen, Petra Poklukar, Chelsea Finn, Mårten Björkman, and Danica Kragic. Bayesian meta-learning for few-shot policy adaptation across robotic platforms, 2021. URL https://arxiv.org/abs/2103.03697.

Hassan Gharoun, Fereshteh Momenifar, Fang Chen, and Amir H Gandomi. Meta-learning approaches for few-shot learning: A survey of recent advances. *ACM Computing Surveys*, 56(12): 1–41, 2024.

Shresth Grover, Akshay Gopalkrishnan, Bo Ai, Henrik I Christensen, Hao Su, and Xuanlin Li. Enhancing generalization in vision-language-action models by preserving pretrained representations. *arXiv preprint arXiv:2509.11417*, 2025.

Jianing Guo, Zhenhong Wu, Chang Tu, Yiyao Ma, Xiangqi Kong, Zhiqian Liu, Jiaming Ji, Shuning Zhang, Yuanpei Chen, Kai Chen, Qi Dou, Yaodong Yang, Xianglong Liu, Huijie Zhao, Weifeng Lv, and Simin Li. On robustness of vision-language-action model against multi-modal perturbations. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cS6xizdYD5.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Shengchao Hu, Wanru Zhao, Weixiong Lin, Li Shen, Ya Zhang, and Dacheng Tao. Prompt tuning with diffusion for few-shot pre-trained policy generalization. *arXiv preprint arXiv:2411.01168*, 2024.

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

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak,

Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=euMVC1DO4k.

Puhao Li, Yingying Wu, Ziheng Xi, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Jianan Wang, Song-Chun Zhu, Tengyu Liu, and Siyuan Huang. ControlVLA: Few-shot object-centric adaptation for pre-trained vision-language-action models. *arXiv preprint arXiv:2506.16211*, 2025a. URL https://arxiv.org/abs/2506.16211.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. *Advances in Neural Information Processing Systems*, 2025b.

Weiqi Li, Quande Zhang, Ruifeng Zhai, Liang Lin, and Guangrun Wang. Vla models are more generalizable than you think: Revisiting physical and spatial modeling. *arXiv preprint arXiv:2512.02902*, 2025c. doi: 10.48550/arXiv.2512.02902.

Ye Li, Yuan Meng, Zewen Sun, Kangye Ji, Chen Tang, Jiajun Fan, Xinzhu Ma, Shu-Tao Xia, Zhi Wang, and Wenwu Zhu. SP-VLA: A joint model scheduling and token pruning approach for VLA model acceleration. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=RwdGIIjPlC.

Anthony Liang, Ishika Singh, Karl Pertsch, and Jesse Thomason. Transformer adapters for robot learning. In *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022.

Wenqi Liang, Gan Sun, Yao He, Jiahua Dong, Suyan Dai, Ivan Laptev, Salman Khan, and Yang Cong. PixelVLA: Advancing pixel-level understanding in vision-language-action model. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7M6ryCABIc.

Li-Heng Lin, Yuchen Cui, Amber Xie, Tianyu Hua, and Dorsa Sadigh. Flowretrieval: Flow-guided data retrieval for few-shot imitation learning. *arXiv preprint arXiv:2408.16944*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning, 2023a. URL https://arxiv.org/abs/2306.03310.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. *Advances in Neural Information Processing Systems*, 2025a. doi: 10.48550/arXiv.2505.19789.

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, 2025b. URL https://arxiv.org/abs/2410.07864.

Zuxin Liu, Jesse Zhang, Kavosh Asadi, Yao Liu, Ding Zhao, Shoham Sabach, and Rasool Fakoor. Tail: Task-specific adapters for imitation learning with large pretrained models. *arXiv preprint arXiv:2310.05905*, 2023b.

Yecheng Jason Ma, Joey Hejna, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, et al. Vision language models are in-context value learners. In *The Thirteenth International Conference on Learning Representations*, 2024.

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. URL https://arxiv.org/abs/2503.14734.

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. URL https://arxiv.org/abs/2501.09747.

Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition, 2018. URL https://arxiv.org/abs/1803.05984.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

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*, 2026. URL https://openreview.net/forum?id=54U3XHf7qq.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. URL https://arxiv.org/abs/2502.19417.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Mingchen Song, Xiang Deng, Guoqiang Zhong, Qi Lv, Jia Wan, Yinchuan Li, Jianye Hao, and Weili Guan. Few-shot vision-language action-incremental policy learning. *arXiv preprint arXiv:2504.15517*, 2025. URL https://arxiv.org/abs/2504.15517.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. Ricl: Adding in-context adaptability to pre-trained vision-language-action models. *arXiv preprint arXiv:2508.02062*, 2025.

StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model developing, 2026. URL https://arxiv.org/abs/2604.05014.

SHILIANG SUN and FENG JIN. Robust co-training. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(07):1113–1126, 2011. doi: 10.1142/S0218001411008981. URL https://doi.org/10.1142/S0218001411008981.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi Fan, Guanya Shi, and Yuke Zhu. Self-improving vision-language-action models with data generation via residual RL. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eUGoqrZ6Ea.

Amber Xie, Rahul Chand, Dorsa Sadigh, and Joey Hejna. Data retrieval with importance weights for few-shot imitation learning. *arXiv preprint arXiv:2509.01657*, 2025.

Ganlin Yang, Tianyi Zhang, Haoran Hao, Weiyun Wang, Yibin Liu, Dehui Wang, Guanzhou Chen, Zijian Cai, Junting Chen, Weijie Su, Wengang Zhou, Yu Qiao, Jifeng Dai, Jiangmiao Pang, Gen Luo, Wenhai Wang, Yao Mu, and Zhi Hou. VLASER: Vision-language-action model with synergistic embodied reasoning. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=8xTDnj39Ti.

Shuai Yang, Hao Li, Bin Wang, Yilun Chen, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Vision-language-action instruction tuning: From understanding to manipulation. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=tsxwloasw5.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE transactions on knowledge and data engineering*, 35(9):8934–8954, 2022.

Jinhui Ye, Fangjing Wang, Ning Gao, Junqiu Yu, Yangkun Zhu, Bin Wang, Jinyu Zhang, Weiyang Jin, Yanwei Fu, Feng Zheng, Yilun Chen, and Jiangmiao Pang. ST4VLA: Spatially guided training for vision-language-action models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eKhOrQWAVJ.

Yida Yin, Zekai Wang, Yuvan Sharma, Dantong Niu, Trevor Darrell, and Roei Herzig. In-context learning enables robot action prediction in llms. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8972–8979. IEEE, 2025.

Shipeng Yu, Balaji Krishnapuram, Rómer Rosales, and R. Bharat Rao. Bayesian co-training. *J. Mach. Learn. Res.*, 12:2649–2680, November 2011. ISSN 1532-4435.

Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning, 2018. URL https://arxiv.org/abs/1802.01557.

Dan Zhang, Tao Feng, Lilong Xue, Yuandong Wang, Yuxiao Dong, and Jie Tang. Parameter-efficient fine-tuning for foundation models. *arXiv preprint arXiv:2501.13787*, 2025a.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. VLM4VLA: Revisiting vision-language models in vision-language-action models. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=tc2UsBeODW.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025b. doi: 10.48550/arXiv.2507.04447.

Yang Zhang, Chenwei Wang, Ouyang Lu, Yuan Zhao, Yunfei Ge, Zhenglong Sun, Xiu Li, Chi Zhang, Chenjia Bai, and Xuelong Li. Align-Then-stEer: Adapting the vision-language action models through unified latent guidance. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=T3i7Ifeatk.

Zhengshen Zhang, Hao Li, Yalun Dai, Zhengbang Zhu, Lei Zhou, Chenchen Liu, Dong Wang, Francis E. H. Tay, Sijin Chen, Ziwei Liu, Yuxiao Liu, Xinghang Li, and Pan Zhou. From spatial to actions: Grounding vision-language-action model in spatial foundation priors. In *International Conference on Learning Representations*, 2026c. URL https://openreview.net/forum?id=fzmittHfq3.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

Zewei Zhou, Tianhui Cai, Seth Z. Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 2025.

APPENDIX

A INTERACTION AND RETENTION BUDGETS

For the LIBERO experiments, CoAct and the controls for self-training across multiple rounds with matched budgets follow three training rounds with two phases of rollout collection. Table 6 summarizes the schedule for each task. The budget for candidate rollouts is shared across the expert pool.

Table 6: Training data and interaction budgets for each task. Retained rollouts denote the rollout trajectories included in each round's training set.

Stage	Expert demonstrations	Retained rollouts	Candidate rollouts
Round 1	5	0	50
Round 2	5	10	50
Round 3	5	25	0

The two collection phases use 100 candidate rollouts per task in total, with no further collection after the third training round. All controls with matched budgets use the same data retention schedule, training procedure for the final policy, and evaluation settings.

A.1 CONFIGURATIONS OF CONTROLS WITH MATCHED BUDGETS

All configurations in Table 7 use the Qwen-OFT 2B backbone and follow the schedule for each task in Table 6. The training procedure for the final policy and evaluation settings are shared across configurations.

Table 7: Comparisons under matched budgets on LIBERO. All variants use the Qwen-OFT 2B backbone, five initial demonstrations per task, and matched interaction and budgets for retained data. Variants without CoTC use random retention. CoAct experts use fixed views assigned to each expert. The superscript shows the gain (pp) over single-policy self-training with CoTC.

Configuration	CoTC	Success (%)
Single-policy self-training controls		
Original views	–	79.3
Pooled transformed views	–	80.5
Original views	✓	81.4
CoAct experts ($K = 5$)		
WeightCoAct + DataCoAct	–	84.2
VLA-Collect	✓	**88.9**$^{\uparrow 7.5}$

Single-policy self-training controls. We evaluate three configurations under matched budgets. The first trains on the original visual observations and uses random trajectory retention. The second pools the same five view configurations used by DataCoAct into the training data of one policy, also with random retention. These views comprise the identity view and transformations of camera viewpoint, illumination, scene appearance, and sensor noise. The third applies full CoAct's CoTC selection procedure to the rollout pool from one policy with original views, giving single-policy self-training with CoTC. These configurations obtain 79.3%, 80.5%, and 81.4% success, respectively.

Controls with multiple experts. Both configurations with multiple experts use $K = 5$ experts with WeightCoAct and DataCoAct. Each expert is assigned one fixed view configuration throughout training. The configuration without CoTC uses random retention and achieves 84.2%. Full CoAct applies CoTC to the combined rollout pool of the experts and produces the VLA-Collect student, which achieves 88.9% success. Both configurations share the total budget for candidate rollouts across the expert pool and use the same data retention schedule.

VLA-Retrieval reference. VLA-Retrieval is the original single-policy self-training baseline: one round of rollout collection, retrieval, and retraining yields 79.0%. The separate control with multiple rounds and random retention achieves 79.3%. Table 7 compares these modified self-training configurations under CoAct's interaction and retention budgets; the control with CoTC reaches 81.4%.

B Experimental Protocols

B.1 Benchmarks and Evaluation Metrics

We evaluate VLA-Collect on LIBERO (Liu et al., 2023a) with five initial expert demonstrations per task. We report results on four suites: Spatial, Object, Goal, and Long. Success rates for each suite are averaged over tasks within each suite, and the overall average is computed over the four suites. To evaluate out-of-distribution generalization, policies trained on LIBERO are directly evaluated on LIBERO-Plus (Fei et al., 2025), without LIBERO-Plus demonstrations, rollouts, or fine-tuning.

For LIBERO, CoAct uses an additional budget of 100 candidate simulator rollouts per task. CoTC uses binary feedback on task completion from the benchmark, without additional expert action annotations or corrective demonstrations. The controls with one policy and matched budgets share CoAct's budgets for interaction and retained data, training procedure for the final policy, and evaluation settings. The schedule for each round and control configurations are specified in Appendix A. Multiple experts are used only for trajectory generation and screening during training; after distillation, VLA-Collect deploys a single student policy with the same cost at inference as its underlying backbone.

B.2 Baseline Models

We compare with OpenVLA (Kim et al., 2024), OpenVLA-OFT (Kim et al., 2025), π_0 (Black et al., 2026), π_0-FAST (Black et al., 2026; Pertsch et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025), GR00T-N1 (NVIDIA et al., 2025), Qwen-OFT (StarVLA Community, 2026; Bai et al., 2025; Kim et al., 2025), VLA-ICL (Ma et al., 2024; Sridhar et al., 2025; Yin et al., 2025), VLA-PEFT (Hu et al., 2022; Liang et al., 2022; Liu et al., 2023b), and VLA-Retrieval (Du et al., 2023; Lin et al., 2024; Xie et al., 2025). The comparison spans representative VLA backbones and adaptation baselines sharing Qwen-OFT 2B with VLA-Collect. VLA-Retrieval denotes our single-policy self-training baseline, which retrieves trajectories from its own rollout memory for retraining. Qwen-OFT, VLA-PEFT, and VLA-ICL use only the initial five demonstrations per task on LIBERO and LIBERO-Plus, without rollout supervision. Appendix A.1 distinguishes the original VLA-Retrieval configuration from its controls with matched budgets, including single-policy self-training with CoTC.

B.3 Real-World Protocol

The five real-world tasks are cloth folding, table wiping, pen insertion, water pouring, and cube cleanup (placing cubes into a bucket), covering manipulation of deformable objects, sustained surface contact, alignment, liquid transfer, and collection of multiple objects. All methods use the $\pi_{0.5}$ backbone and the same ten initial expert demonstrations per task. VLA-Retrieval and CoAct each collect ten additional physical trajectories per task and follow the same retraining protocol. For CoAct, this collection budget is shared across the entire expert pool. The real-world CoAct pipeline includes generation of distinct views for each expert and collective trajectory curation. CoTC uses human judgments of binary task completion, while signals for ranking trajectories are computed from recorded observations and robot proprioception. Each resulting policy is evaluated over 20 physical trials per task; the reported average is the mean success rate across the five tasks. Replays of VLA-Collect performing four of these tasks are visualized in Appendix F.

B.4 Protocols for Ablation and Data Transfer

All component ablations use the Qwen-OFT 2B backbone with five initial demonstrations per task and the same evaluation protocol. Table 5 starts from initial Qwen-OFT SFT (75.2%), before rollout self-training, and enables WeightCoAct, DataCoAct, and CoTC in the indicated combinations.

To test the transferability of supervision collected with CoAct, we use the curated dataset to train six additional VLA backbones: OpenVLA, OpenVLA-OFT, π_0, π_0-FAST, $\pi_{0.5}$, and GR00T-N1. Figure 3 reports their average success rates on LIBERO and LIBERO-Plus and the absolute gains over the corresponding original models. Appendix D describes the controlled scenes, initializations, instructions, and rollout budgets used for qualitative comparisons.

B.5 DataCoAct View Configurations

The LIBERO configuration uses $K = 5$ experts and $M = 5$ fixed view configurations, listed in Table 8. Each expert is assigned one view configuration throughout training: one expert receives the identity view, and four experts receive the transformed views. The transformations modify visual observations while retaining the action labels and trajectory timing. The control that pools views for one policy in Table 7 uses the same set of view configurations for one policy.

Table 8: Default DataCoAct view configurations. Each expert uses one fixed configuration.

View configuration	Description
Identity	Preserves the original visual observations.
Camera viewpoint	Varies the observation viewpoint to model changes in camera placement.
Illumination	Changes scene brightness and lighting conditions.
Scene appearance	Modifies background and tabletop textures.
Sensor noise	Applies image corruption that mimics sensor noise.

B.6 CoTC Scoring and Selection

This section specifies the ranking score in Eq. (3). For one task and collection round, let $\mathcal{S}$ contain the successful trajectories from all experts, and let $\mathcal{D}_T$ denote the seed demonstrations for that task. A trajectory τ contains T_τ executed control steps and observations indexed by $t = 0, \ldots, T_\tau$. Ranking is performed within $\mathcal{S}$, using a common control frequency and the same camera inputs for all trajectories of the task.

Motion and execution length. For gripper j, normalize its recorded aperture $w_{t,j}$ by its hardware limits: $u_{t,j} = (w_{t,j} - w_j^{\min})/(w_j^{\max} - w_j^{\min})$. With m grippers, define

$$G(\tau) = \frac{1}{m} \sum_{j=1}^{m} \sum_{t=1}^{T_\tau} |u_{t,j} - u_{t-1,j}|, \qquad L(\tau) = T_\tau. \tag{4}$$

Lower G favors less cumulative opening and closing; lower L favors shorter executions. These are preferences for economical execution among successful trajectories, not tests of whether an individual movement is necessary. Averaging over grippers gives the same definition for single-arm and dual-arm settings.

Visual novelty. The scoring encoder is a fixed copy of the initial VLA's visual encoder before expert diversification. It remains unchanged across experts and collection rounds; the expert backbones themselves remain trainable. At every recorded time step, apply the encoder's native image preprocessing to the original camera frames, without DataCoAct transformations. Mean-pool the final spatial visual tokens, after the encoder's standard output projection and before fusion with language, separately for each camera. Average these vectors equally over the policy's camera inputs and normalize the result to unit ℓ_2 length, obtaining $z_{\tau,t}$. A zero vector is left at zero. Apply the same procedure to all seed frames and define

$$N(\tau) = \frac{1}{T_\tau + 1} \sum_{t=0}^{T_\tau} \min_{\substack{\sigma \in \mathcal{D}_T \\ 0 \le s \le T_\sigma}} \left(1 - z_{\tau,t}^\top z_{\sigma,s}\right). \tag{5}$$

All recorded frames contribute, so no temporal sampling parameter is introduced. Larger N favors observations outside the seed demonstrations' visual coverage. This measures visual difference, including appearance variation, not greater task difficulty or behavioral diversity.

Normalization and selection. For $n = |\mathcal{S}| > 0$, convert each statistic x to its percentile within the current task's successful pool, assigning ties the same midpoint rank:

$$R_x(\tau) = \frac{1}{n} \sum_{\sigma \in \mathcal{S}} \left[\mathbf{1}\{x(\sigma) < x(\tau)\} + \frac{1}{2} \mathbf{1}\{x(\sigma) = x(\tau)\} \right]. \tag{6}$$

Orient all three statistics so that larger values are preferred and take their arithmetic mean:

$$q(\tau) = \frac{R_{-G}(\tau) + R_{-L}(\tau) + R_N(\tau)}{3}. \tag{7}$$

The three coefficients are fixed at $1/3$; there is no separately tuned novelty coefficient or score threshold. Percentile normalization removes the differences in units and numerical scales. A constant statistic gives every candidate $1/2$ and has no effect on their relative order. Keep the $\min(b, n)$ highest-scoring trajectories under the retention budget in Appendix A, breaking equal total scores by ascending collection identifier assigned before scoring. If $n = 0$, retain no rollout; if $n < b$, retain all successful rollouts without filling the quota with failures or duplicates. Scores affect selection only: retained trajectories use the standard imitation loss without score weighting.

B.7 Relation to Established Training Paradigms

CoAct coordinates complementary expert training, curation of the joint pool, and distillation for few-shot VLA post-training. Table 9 distinguishes this coordination from related training operations; the contribution concerns how these operations work together to construct supervision.

Table 9: Design distinctions between CoAct and related paradigms.

Related paradigm	CoAct design
Ensemble at inference	Experts generate complementary training trajectories that are distilled into one deployment policy. Their actions are not combined at inference.
Single-policy self-training (VLA-Retrieval)	Deliberately heterogeneous experts generate the candidate rollout pool, expanding the sources of experience beyond one policy distribution.
Data augmentation	Persistent views assigned to each expert encourage specialization across policies, rather than training one policy for invariance to all views.
Aggregation as in DAgger (Ross et al., 2011)	Task completion selects trajectories executed by the policies; no additional expert action relabeling is required beyond the initial demonstrations.
Trajectory filtering	CoTC filters the joint expert pool by task completion and ranks successful trajectories using observable statistics and novelty relative to the seed demonstrations.

B.8 RoboTwin Protocol

The initial RoboTwin policy uses Qwen3-VL-2B-Instruct with the Qwen-OFT MLP action head. Initial SFT jointly covers all 50 tasks with 20 complete clean demonstrations per task, giving 1,000 demonstrations in total. The demonstrations are selected from the clean pool of 50 trajectories per task using deterministic SHA-256 ordering with sampling seed 42. Both the backbone and action head are fine-tuned for 40,000 steps with global batch size 32, 2,000 warmup steps, and training seed 42. Learning rates are 10^{-5} for the backbone and 10^{-4} for the action head, using AdamW and a cosine schedule with minimum learning rate 5×10^{-7}. These settings describe the initial SFT stage.

CoAct is subsequently applied only to the five tasks in Table 3. They are chosen from the eight tasks whose initial 20-shot success rates fall between 50% and 70%. Their task identifiers, in table column order, are `grab_roller`, `move_playingcard_away`, `place_can_basket`, `press_stapler`, and `stack_blocks_three`. Thus the initial policy is trained across all 50 tasks, whereas the collective post-training and evaluation cover only this subset.

The baseline uses the `aloha-agilex` embodiment with head, left-wrist, and right-wrist RGB images, each resized to 224×224, without state input. It predicts 14-dimensional absolute joint and gripper actions in chunks of 50 steps, with action ensembling disabled. Evaluation uses the

clean environment with unseen language instructions, using 100 valid episodes per task and method. The initial-policy evaluation uses seed 0 and screens initializations for expert feasibility; 100 valid episodes therefore do not mean the first 100 candidate environment seeds. The reported average is the arithmetic mean over the five tasks, equivalently 298 successes out of 500 episodes for initial SFT and 390 out of 500 for VLA-Collect. These results measure performance on the selected subset; the LIBERO interaction schedule in Appendix A does not specify the RoboTwin CoAct budget.

B.9 Compute Resources

All experiments were conducted on a single internal compute node equipped with 8 NVIDIA H20 GPUs. We used the same hardware configuration for all compared methods, including the Qwen-OFT baseline, few-shot adaptation baselines, and VLA-Collect. The main VLA-Collect run includes training of the expert pool, rollout generation, trajectory curation with CoTC, and final student distillation, whereas the final deployed policy is a single student VLA with the same cost at inference as the underlying backbone. Since elapsed time can vary with cluster load, data loading, and simulator scheduling, we report the hardware configuration used for all experiments and will release detailed running scripts and logs when the code is made public.

C Supplementary Experiments

C.1 Repeated Evaluations and Results by Task

We evaluate VLA-Collect over 10 independent evaluation runs, with 50 trials per task in each run. All uncertainty estimates in this section are the sample standard deviation across evaluation runs, not the standard error or a confidence interval. They quantify evaluation variability and do not measure variability from independent training seeds or resampling the initial demonstrations. For each run, success for each suite is the mean over the ten tasks in that suite, and overall LIBERO success is the mean over the four suites. We report each metric's mean and sample standard deviation over ten runs; aggregate standard deviations are not averages of per-task standard deviations.

Table 10 reports all 40 task results; Figure 6 visualizes the means. VLA-Collect achieves an overall LIBERO success rate of $89.2 \pm 0.2\%$ across repeated evaluations. The 88.9% result in Table 1 is the original result from a single evaluation. We retain that reporting convention in the original comparisons and report the statistics from repeated evaluations separately here.

Table 10: VLA-Collect success rates (%) on all 40 LIBERO tasks. Entries are mean $\pm$ sample standard deviation across 10 evaluation runs, each with 50 trials per task. Task identifiers t_0–t_9 refer to the task indices within each suite.

Task	Spatial	Object	Goal	Long
t_0	96.4 ± 1.6	98.4 ± 0.8	83.0 ± 1.4	80.4 ± 0.8
t_1	96.6 ± 1.0	88.4 ± 0.8	97.8 ± 2.2	87.6 ± 1.3
t_2	100.0 ± 0.0	97.0 ± 1.1	89.0 ± 1.7	94.4 ± 0.8
t_3	99.4 ± 1.3	100.0 ± 0.0	87.2 ± 1.0	83.2 ± 1.7
t_4	93.8 ± 1.5	98.6 ± 1.3	98.6 ± 1.0	72.2 ± 0.6
t_5	75.8 ± 0.6	90.2 ± 0.6	52.0 ± 0.0	90.4 ± 0.8
t_6	95.6 ± 1.6	94.6 ± 1.0	82.6 ± 1.0	64.4 ± 2.1
t_7	92.2 ± 0.6	98.4 ± 0.8	100.0 ± 0.0	85.2 ± 1.4
t_8	93.0 ± 1.4	94.6 ± 1.0	95.0 ± 1.4	57.6 ± 1.3
t_9	96.6 ± 1.3	94.0 ± 0.9	84.2 ± 0.6	88.8 ± 1.4
Suite avg.	93.9 ± 0.4	95.4 ± 0.3	86.9 ± 0.3	80.4 ± 0.3

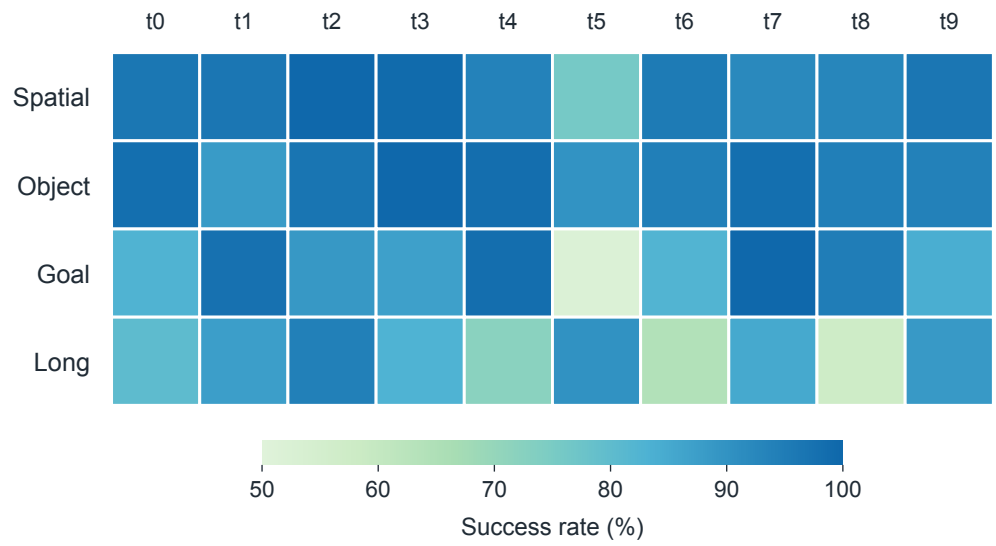


Figure 6: Mean success rates (%) for all 40 LIBERO tasks across 10 evaluation runs (50 trials per task per run). Colors use a shared 50–100% scale. Task indices t_0–t_9 refer to different tasks in each suite. Table 10 reports means and sample standard deviations.

C.2 Repeated Component Ablations

We repeat the evaluation of all component configurations in Table 5 using the same protocol as Appendix C.1. Table 11 reports means over evaluation runs and sample standard deviations on LIBERO and LIBERO-Plus. The full model has the highest mean success rate on both benchmarks. The configuration using only DataCoAct below is distinct from the single-policy self-training control with CoTC in Table 7, although both have a LIBERO mean of 81.4%.

Table 11: Repeated evaluation of component ablations. All variants use the Qwen-OFT 2B backbone and five initial demonstrations per task. Results are success rates (%), reported as mean ± sample standard deviation across 10 evaluation runs. Superscripts show mean gains (pp) over the first row.

WeightCoAct	DataCoAct	CoTC	LIBERO	LIBERO-Plus
–	–	–	75.2 ± 0.3	41.2 ± 0.8
✓	–	–	$80.2 \pm 0.4^{\uparrow 5.0}$	$45.0 \pm 0.7^{\uparrow 3.8}$
–	✓	–	$81.4 \pm 0.5^{\uparrow 6.2}$	$46.9 \pm 0.9^{\uparrow 5.7}$
✓	✓	–	$84.2 \pm 0.4^{\uparrow 9.0}$	$50.3 \pm 0.6^{\uparrow 9.1}$
✓	✓	✓	$\mathbf{89.2 \pm 0.2}^{\uparrow 14.0}$	$\mathbf{54.3 \pm 0.7}^{\uparrow 13.1}$

C.3 Student Distillation versus Expert Action Ensembling

We compare the distilled VLA-Collect student with an action ensemble constructed from the same five experts and used at inference. At each control step, the ensemble averages the continuous action predictions and uses majority voting for the gripper command. As shown in Table 12, the student improves LIBERO success by 3.8 percentage points while using one active policy at inference instead of five. This comparison supports consolidation through trajectories over this specific rule for aggregating actions; it does not measure inference latency or establish superiority over other ensemble methods.

Table 12: Student distillation versus action ensembling from the same five experts. Entries are point estimates of success rate (%). Active policies count the policies used at each inference step. The superscript shows the gain (pp) over the expert action ensemble.

Deployment strategy	LIBERO success (%)	Active policies
Expert Action Ensemble	85.1	5
VLA-Collect student	**88.9**$^{\uparrow 3.8}$	1

C.4 Sensitivity to the Size of the Expert Pool

The main configuration uses $K = 5$ experts. Table 13 evaluates $K \in \{3, 5, 7, 9\}$, and Figure 7(a) shows the trend. Increasing the pool from three to five experts improves success by 3.1 percentage points, whereas seven and nine experts improve over five by 0.8 and 1.2 points, respectively. These results show diminishing accuracy gains in the tested range and motivate the default pool size. They do not establish an optimum under matched total training compute.

Table 13: Sensitivity to the size of the expert pool on LIBERO. The default $K = 5$ configuration is highlighted. Entries are point estimates of success rate (%).

Number of experts K	3	5 (default)	7	9
LIBERO success (%)	85.8	**88.9**	89.7	90.1

C.5 Demonstration Budget and Applicability Range

We vary the number of initial demonstrations per task while keeping the backbone, budget for rollout interactions, and evaluation protocol fixed. Table 14 reports results for one, three, five, ten, and fifteen demonstrations per task; Figure 7(b) compares the two methods. VLA-Collect provides its largest gains at three to five demonstrations; the gain decreases as the initial SFT policy approaches saturation with ten to fifteen demonstrations.

Table 14: LIBERO success rates (%) as the initial demonstration budget varies. Superscripts show gains (pp) over initial SFT at the same demonstration budget. "Unstable" indicates that no numerical success rate is available.

Demonstrations per task	Initial SFT	VLA-Collect
1	46.2	Unstable
3	68.7	81.6$^{\uparrow 12.9}$
5	75.2	88.9$^{\uparrow 13.7}$
10	86.5	91.3$^{\uparrow 4.8}$
15	93.2	95.4$^{\uparrow 2.2}$

In the setting with one demonstration, the initial policy produces too few successful trajectories under the fixed interaction budget, and CoAct training does not yield a stable VLA-Collect result. No numerical VLA-Collect success rate is reported for this setting. This failure case indicates that CoAct requires a sufficiently capable initial policy to supply useful successful rollouts within the collection budget. This LIBERO sweep did not test twenty or fifty demonstrations.

D Qualitative Rollout Visualization

D.1 Dataset-Referenced Rollout Coverage

The reference for relative rollout coverage is the complete demonstration dataset for the evaluated LIBERO tasks, before selecting the few-shot training subset. Coverage of this reference dataset by itself is normalized to one. The quantity describes how much of the reference trajectory repertoire

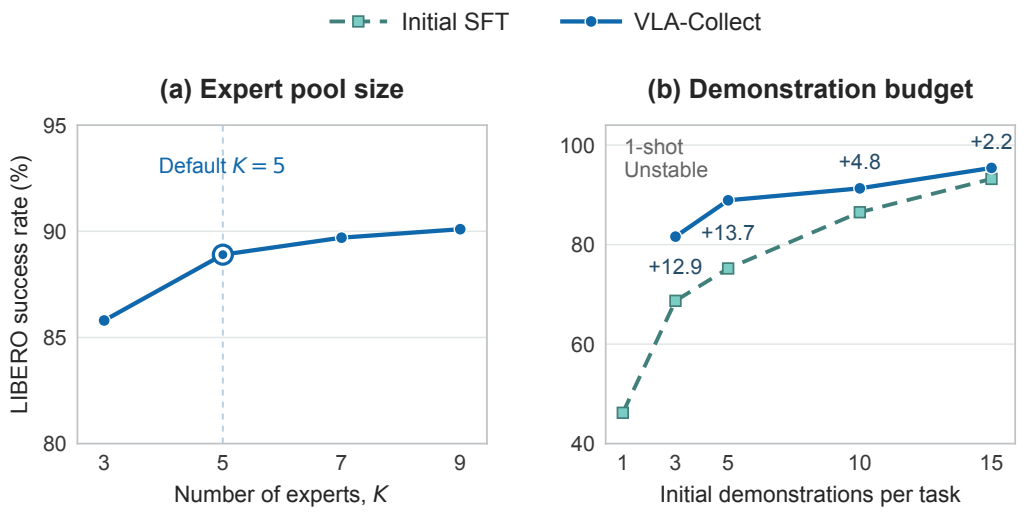


Figure 7: Sensitivity to the expert pool size and initial demonstration budget on LIBERO. (a) Success as the expert pool grows, with the default $K = 5$ highlighted. (b) Success under different demonstration budgets; annotations show gains over initial SFT in percentage points. VLA-Collect was unstable with one demonstration, so no numerical point is plotted there. Markers are reported point estimates and segments connect tested settings; no uncertainty estimates are available. The panels use different vertical scales. Corresponding values appear in Tables 13 and 14.

is represented by a collection of successful rollouts; its denominator is neither the number of rollout attempts nor the size of the few-shot subset.

We specify a reproducible proxy through occupancy of reference trajectory cells. For a trajectory τ, sample up to 32 distinct frames uniformly over its duration, encode them with frozen CLIP ViT-B/32 (Radford et al., 2021), normalize each frame embedding, and average them. Normalize this average to obtain a trajectory representation $h(\tau) \in \mathbb{R}^{512}$. For task t, let $\mathcal{D}_t$ be the complete reference collection and $Z_t = \{z_{t,1}, \ldots, z_{t,N_t}\}$ its distinct trajectory representations. Each reference representation defines a cell by nearest-neighbor assignment in this original feature space. For the successful rollout set $\mathcal{R}_{m,t}$ of method m, define

$$\begin{aligned} a_t(\tau) &= \arg\min_{1 \le j \le N_t} \|h(\tau) - z_{t,j}\|_2, \\ \mathrm{Cov}(\mathcal{R}_m; \mathcal{D}) &= \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{\left|\{a_t(\tau) : \tau \in \mathcal{R}_{m,t}\}\right|}{N_t}, \end{aligned} \tag{8}$$

where $\mathcal{T}$ is the evaluated task set and ties are resolved by a fixed reference ordering. An empty successful set has zero coverage. Multiple rollouts assigned to the same reference cell count only once, and tasks receive equal weight. Every distinct reference representation is assigned to itself, so $\mathrm{Cov}(\mathcal{D}; \mathcal{D}) = 1$ exactly. For example, occupying 20 of 50 reference cells gives coverage 0.4 for that task. Comparisons using this proxy require the same reference data, encoder, frame sampling rule, and candidate rollout budget across methods.

This definition measures occupancy relative to a finite dataset. A nearest reference assignment does not establish close trajectory imitation, and visual features averaged over time do not preserve action dynamics or temporal order. The statistic therefore does not measure the volume of physically reachable states or reward exploration outside the reference repertoire. Neither PCA coordinates nor the apparent area or volume of the point clouds in Figure 4 enter Equation 8.

In Figure 1, the horizontal coordinates $(0.16, 0.26, 0.46, 0.74)$ report the normalized dataset coverage for ICL, PEFT, Retrieval, and CoAct, respectively. The vertical coordinate is computed from

the reported success rates:

$$\Delta_m = \frac{1}{2}\big[(S_{m,\text{LIBERO}} - 75.2) + (S_{m,\text{RoboTwin}} - 59.6)\big], \tag{9}$$

where success rates are expressed in percent and Δ_m is in percentage points. The benchmarks receive equal weight; RoboTwin refers to the five selected tasks. The resulting gains are -0.8, 1.4, 5.0, and 16.05 percentage points, with the last displayed as 16.1. The coverage reference is LIBERO alone, so the horizontal axis is not a mean over the two benchmarks. This overview compares dataset coverage with performance gains; evidence for the benefits of collective generation and curation comes from the comparisons under matched budgets in Appendix A.1.

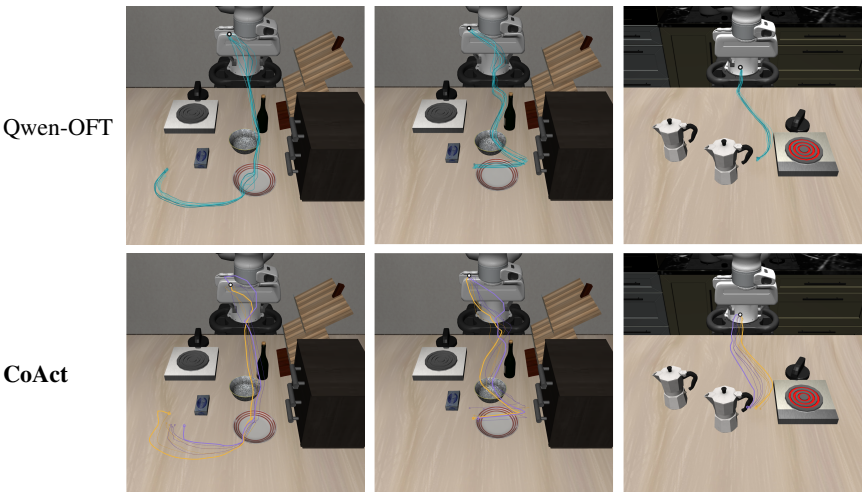
D.2 Visual Feature Distributions on LIBERO-Goal

Figure 4 compares visual feature distributions across all ten LIBERO-Goal tasks. Per task, the Base panel contains five training demonstrations, the Single Policy panel contains ten successful rollouts from the five-shot Qwen-OFT policy, and the CoAct panel contains ten successful rollouts from the CoAct expert pool. Thus, each rollout panel contains 100 successful trajectories across the ten tasks.

We sample 32 actual frames uniformly over each video, without interpolation, encode each frame with frozen CLIP ViT-B/32 (Radford et al., 2021), and normalize its 512-dimensional embedding to unit length. A single PCA is fitted to frames from the unique displayed videos across all three panels, and the first three components are shown with common axis limits, camera, point size, and opacity. Color denotes trajectory phase: normalized time within each execution. Shared videos contribute once to the PCA fit, and temporally adjacent frames are not independent trials.

The two rollout panels have equal numbers of trajectories and sampled frames; this does not establish matched training or interaction budgets. CLIP captures appearance as well as task progress, so a wider projected distribution need not imply wider behavioral coverage. We use the figure to describe the visual features of successful rollouts; the controls in Appendix A.1 provide the separate performance comparison under matched budgets.

D.3 Task-Level Trajectory Examples



(a) Plate pushing (b) Drawer opening (c) Pot grasping

Figure 8: Qualitative rollouts from Qwen-OFT (cyan, top) and CoAct experts (purple and yellow, bottom). Panels (b, c) show eight end-effector trajectories each; thicker curves mark CoTC selections.

For drawer opening and pot grasping in Figure 8, scene, object layout, initialization, instruction, and rollout budget are fixed across methods. The pool offers more varied paths for CoTC selection; Appendix A.1 provides the performance comparison under matched budgets.

E GENERATED LIBERO-LONG EXAMPLES

We show one generated episode of median length for each of the ten LIBERO-Long tasks. Each grid contains ten uniformly sampled frames spanning 5%–95% of the episode.

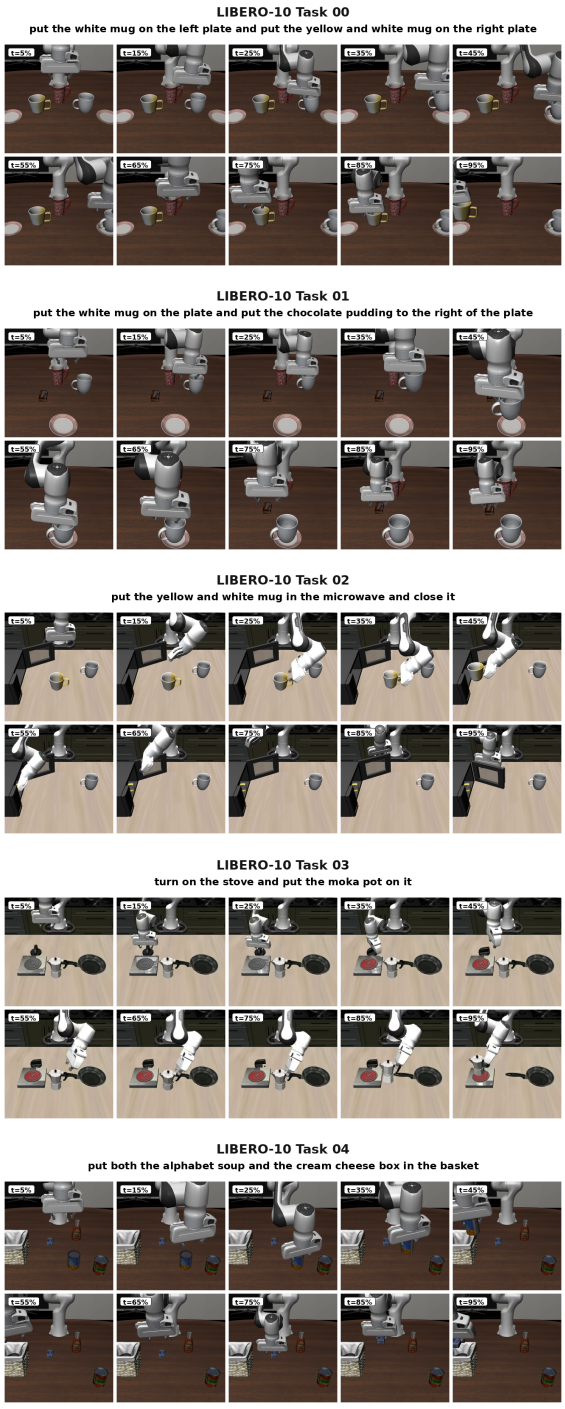


Figure 9: Generated dataset visualization for LIBERO-Long, tasks 0–4.

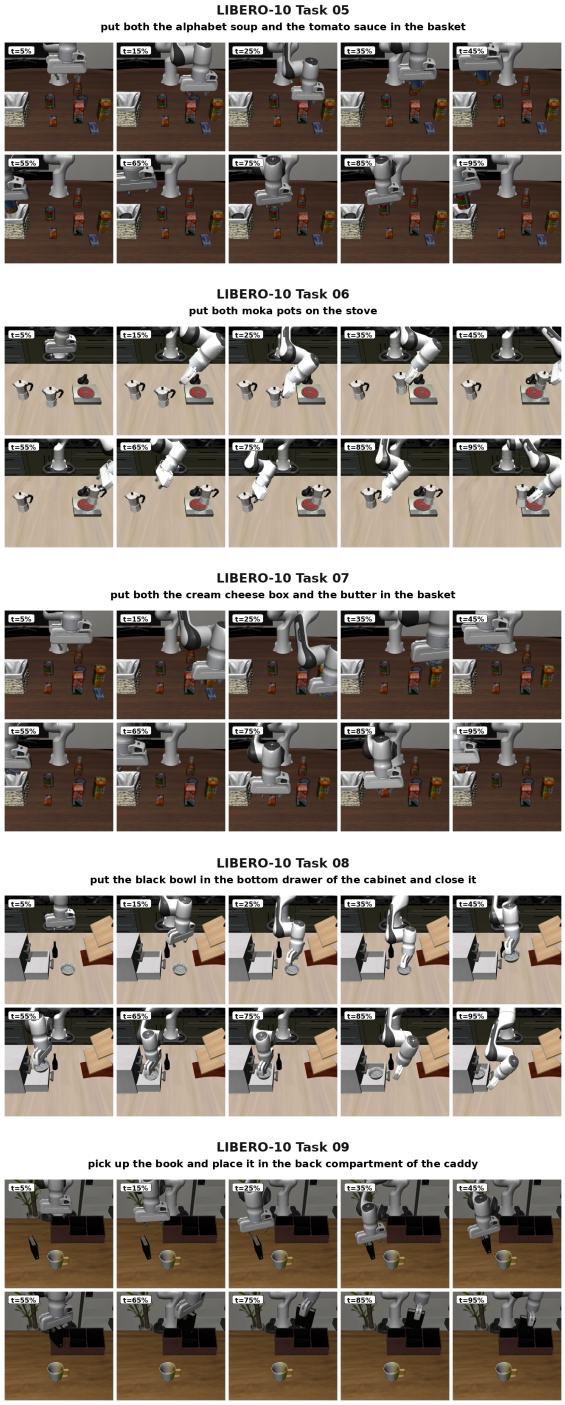


Figure 10: Generated dataset visualization for LIBERO-Long, tasks 5–9.

F Real-World Replays of VLA-Collect

We present replay visualizations of VLA-Collect performing four real-world tasks: table wiping, pen insertion, water pouring, and cube cleanup. Each grid shows one task execution using ten uniformly spaced frames, including the first and last frames, from the same recorded episode and global camera. Frames are read from left to right across the top row, then from left to right across the bottom row. These replays illustrate VLA-Collect's task execution; quantitative success rates are reported in Table 4.

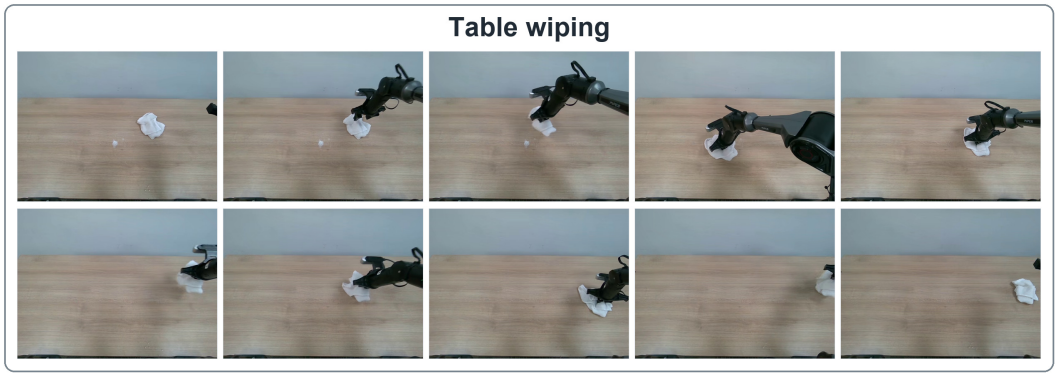


Figure 11: Table wiping. Ten frames from a VLA-Collect replay show towel contact and sweeping motions across the tabletop.

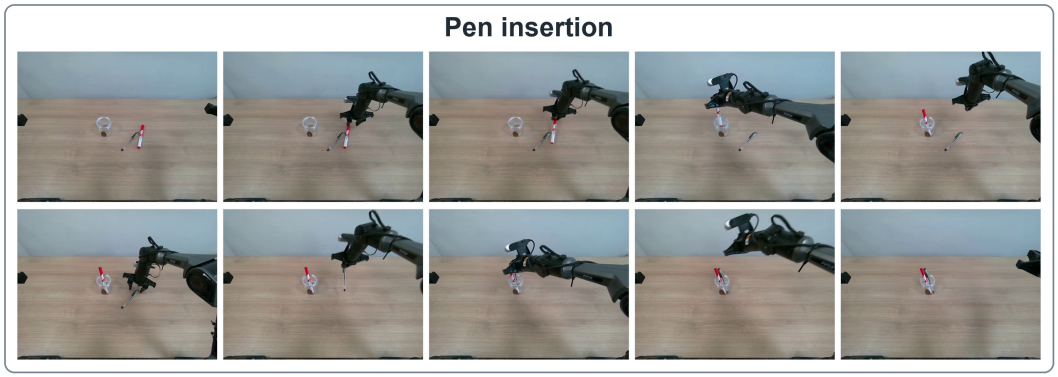


Figure 12: Pen insertion. Ten frames from a VLA-Collect replay show sequential placement of two pens into a holder.

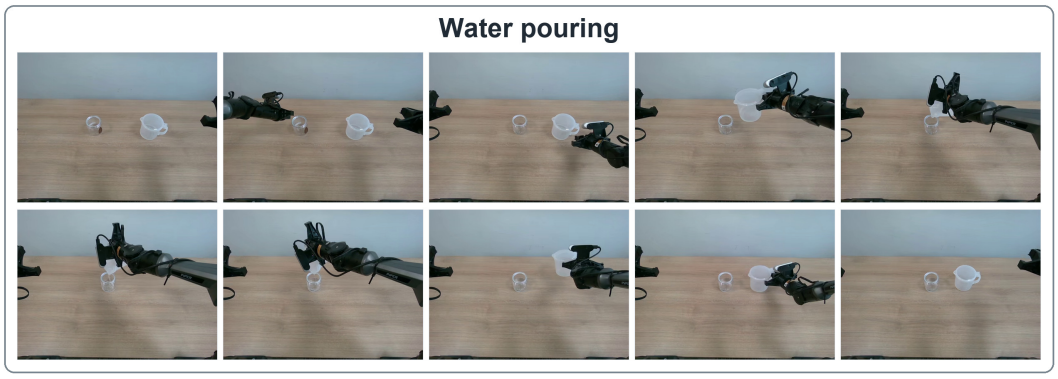


Figure 13: Water pouring. Ten frames from a VLA-Collect replay show grasping a pouring vessel, transferring water into a receiving cup, and returning the vessel to the table.

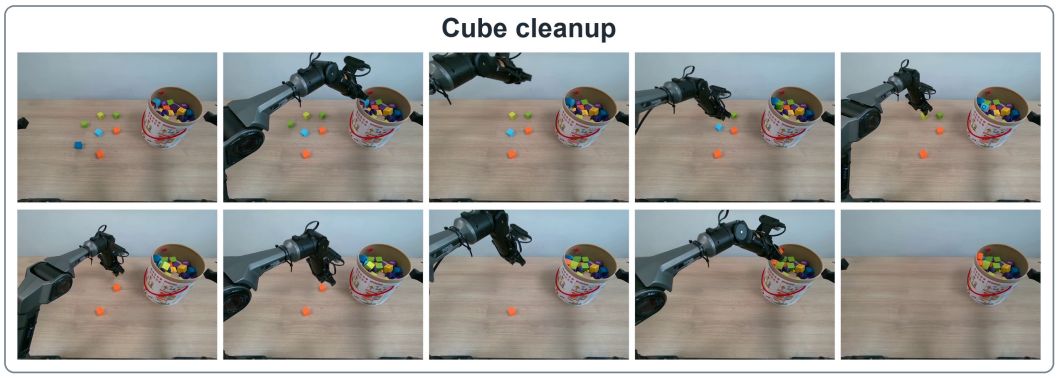


Figure 14: Cube cleanup. Ten frames from a VLA-Collect replay show the robot repeatedly picking up scattered cubes and placing them in a bucket.