BEYOND POLICY PATCHES: DISTILLING RL SPECIALISTS INTO VISION-LANGUAGE-ACTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies can handle noncritical stages of long-horizon manipulation yet fail during short precision-critical phases. Lightweight reinforcement learning (RL) specialists can address these failures without directly applying RL to the full VLA, but retain auxiliary controllers and switching mechanisms at deployment. We study how to transfer their capabilities into a single flow-matching VLA. Direct imitation of specialist rollouts can reproduce action jitter and leave the VLA unprepared for states encountered during its own execution. We introduce REPAIR, a value-guided distillation framework that combines offline initialization with on-policy refinement. At each critical-phase observation, REPAIR queries the corresponding specialist for multiple action candidates and weights them using its critic to construct a supervision distribution. A shared weighted flow-matching objective trains the VLA first on observations from specialist-assisted rollouts, then on observations collected through its own execution. Specialists provide supervision without taking control during on-policy collection and are removed at deployment. Across six real-robot tasks with 30 evaluation trials per task, REPAIR achieves 100% success on all six tasks, with lower mean end-effector jerk than specialist-assisted execution on every task. Additional visual results are available at https://repair-vla.github.io/.

1 INTRODUCTION

Vision-language-action (VLA) policies provide a general foundation for robotic manipulation (Black et al., 2024; Intelligence et al., 2025; Zheng et al., 2026; Bjorck et al., 2025), yet reliable long-horizon execution can be limited by short precision-critical phases (Fan et al., 2025). A policy may reliably approach an object, yet fail during contact-rich insertion, alignment, fastening, or assembly. In these phases, millimeter-scale errors, hesitation, and repeated corrections can determine the outcome of the entire episode. Collecting more human demonstrations is not always sufficient: demonstrations may be inconsistent precisely where the required behavior is most exact, while supervised fine-tuning (SFT) provides no task-reward signal for improving those demonstrated actions.

Reinforcement learning (RL) offers a way beyond these limitations by improving behavior through interaction and task feedback (Lei et al., 2025; Chen et al., 2026; Jiang et al., 2025). However, directly applying RL to the VLA's action expert can lead to unstable policy updates (Chen et al., 2025b; Wang et al., 2026). Full-task RL introduces a separate difficulty in long-horizon settings, where sparse terminal rewards must assign credit to decisions made many steps earlier (Zhou et al., 2026). An alternative is to freeze the VLA and train a lightweight MLP specialist only for each precision-critical phase (Xu et al., 2026). This avoids RL updates to the large action expert and, with phase-local start conditions and success feedback, shortens the credit-assignment horizon. The learning advantage comes with deployment costs: a phase-specific design adds controllers as more critical skills are introduced, and handoffs between the VLA and specialists can introduce action discontinuities. The specialist's own execution can also be jittery, making it unstable to deploy (Yu et al., 2026b).

Local specialists thus simplify RL training but complicate deployment. This raises a question: *Can this local RL expertise instead repair the VLA itself, without retaining auxiliary controllers at deployment?* A natural approach is to distill the specialist's executed actions into the VLA. Prior work shows that RL-generated demonstrations can outperform human data (Xu et al., 2024; Yang et al., 2025; Xiao et al., 2025). Yet successful trajectories alone leave two problems unresolved. First,

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.

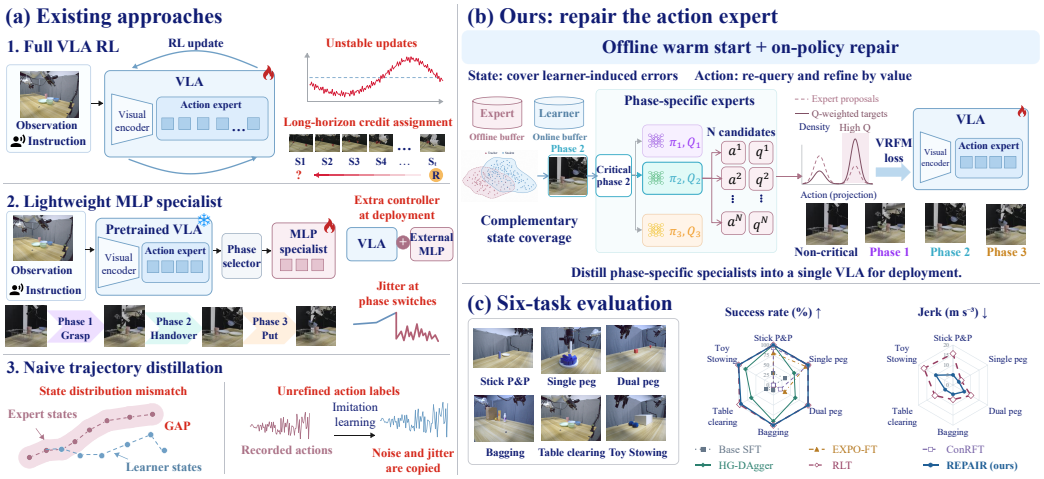


Figure 1: Overview of REPAIR. Local RL expertise provides value-guided supervision for offline initialization and on-policy repair. The VLA acts without specialist takeover during online distillation and is deployed as a single policy.

their action labels record one realization of the specialist's behavior, including stochastic fluctuations and switching artifacts. Direct imitation treats these variations as targets alongside useful corrections, and can reproduce jitter even after the specialist is removed (Section 3). The recorded actions leave unused the expert's ability to propose alternatives and its critic's ability to evaluate them. Second, the VLA induces a different state distribution from specialist-assisted execution. Under purely offline distillation, small deviations can therefore lead to states without appropriate supervision and compound into task failure (Ross et al., 2011). Effective transfer requires constructing suitable expert action targets at the states the learner actually encounters.

To address these challenges, we introduce REPAIR, a value-guided on-policy distillation framework that transfers local RL expertise into a single VLA. At each critical-phase observation, REPAIR re-queries the relevant specialist for multiple action candidates and uses group-normalized critic scores to construct a value-refined expert action distribution. A weighted flow-matching objective trains the VLA on these alternatives, rather than on fixed executed-action labels. The same supervision operator supports both stages of learning: an offline warm start uses observations from specialist-assisted rollouts to establish the critical skills, and on-policy repair applies the operator to observations collected by the VLA itself. During this second stage, specialists supply action targets without taking control, so supervision follows errors arising under the learner's own execution. Imitation of replayed VLA actions supports retention of noncritical behavior, using the student's stored predictions during on-policy repair. Only the repaired VLA is deployed, with no auxiliary specialist or controller switching.

The contributions are:

- We empirically study specialist deployment and trajectory distillation in long-horizon manipulation. Our analysis shows that successful critical-phase takeover can still produce jittery execution, and identifies inherited action fluctuations and teacher–learner state-distribution mismatch as limitations of direct trajectory distillation.
- We introduce REPAIR, a unified framework that absorbs local RL expertise into the VLA's action expert. Critic-weighted expert candidates define a value-refined supervision distribution, with the same flow-matching objective supporting offline initialization and on-policy repair on learner-visited states, without specialist takeover during online collection.
- Across six real-robot tasks with 30 evaluation trials each, REPAIR achieves 100% success on all six tasks. It deploys a single VLA without auxiliary controllers and reduces mean end-effector jerk on successful trajectories by 29.8–67.5% relative to RL specialist takeover.

2 Related Work

Real-world RL and VLA post-training. On-policy RL approaches (Jiang et al., 2026; Chen et al., 2025a; Liu et al., 2025), improve VLA performance beyond supervised fine-tuning (SFT). However, these methods are largely evaluated in simulation, because parallel rollouts are not available in real-world settings. To improve sample efficiency, RLPD (Ball et al., 2023) reuses offline data during online learning, while HIL-SERL (Luo et al., 2025) combines demonstrations and human corrections for real-robot training. ConRFT (Chen et al., 2025b) extends this direction to VLAs. Direct value optimization of expressive action generators can nevertheless destabilize training. Residual (Yuan et al., 2025) and action-editing methods (Dong et al., 2026), ease this optimization burden by training lightweight correction policies. However, a smaller controller does not by itself shorten the temporal credit-assignment horizon. RL Token (RLT) (Xu et al., 2026) concentrates online learning on precision-critical phases, where a lightweight MLP actor takes over from the base VLA, localizing credit assignment to short task segments. This design retains auxiliary controllers and policy handoffs at deployment. These limitations motivate transferring local RL expertise into a single VLA that executes the entire task without auxiliary controllers.

Distilling Knowledge from data and policy priors. Supervised fine-tuning (SFT) on human teleoperation data is a common way to transfer task knowledge into pretrained robot policies (Black et al., 2024). HG-DAgger (Kelly et al., 2019) extends demonstration learning by allowing humans to take over during policy execution and aggregating their corrective actions for subsequent policy updates. Learned policies provide another source of supervision beyond human demonstrations. RLDG (Xu et al., 2024) trains task-specific RL experts and uses their rollouts to fine-tune generalist policies. Beyond Human Demonstrations (Yang et al., 2025) uses diffusion-based RL to generate smooth, low-variance trajectories for VLA training. PLD (Xiao et al., 2025) learns residual specialists that probe generalist failure regions, collects recovery trajectories through hybrid rollouts aligned with the generalist's state distribution, and distills the curated data into the generalist with SFT. In these methods, knowledge transfer relies on action labels recorded during data collection. REPAIR instead queries the specialist actor and critic to construct supervision at each critical-phase observation: it resamples multiple action chunks and weights them by their relative predicted values to form a value-refined target distribution. The same supervision rule supports offline initialization and subsequent on-policy distillation on learner-visited states, where specialists provide action proposals and value-based preferences without taking control.

3 Analysis of Specialist Deployment and Distillation

Can a lightweight MLP serve as a deployed specialist? We examine this question in two peg-insertion settings: short-horizon single-arm insertion and a longer-horizon task culminating in peg assembly. We use a lightweight MLP specialist trained with RL Token (RLT), and compare two control scopes: *full-task takeover*, where the specialist controls the entire episode, and *critical-phase takeover*, where the base VLA executes the preceding stages and the specialist controls only the final assembly phase. Our preliminary experiments show comparable success for the two variants in the short-horizon setting. In the longer-horizon setting, full-task takeover performs poorly, whereas critical-phase takeover achieves high success but exhibits pronounced action jitter (see Figure 2(c)). These observations expose a practical tension: restricting the specialist's scope makes the local skill effective, but retaining it as a deployed patch does not yield satisfactory end-to-end control. They motivate repairing the VLA's own action expert so that the same action generator handles both the preceding stages and precision-critical behavior.

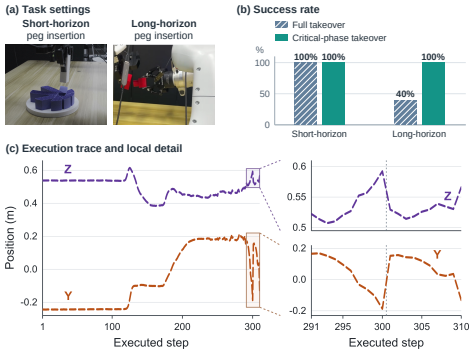


Figure 2: **Specialist deployment across task horizons.** (a) Peg-insertion settings. (b) Success rates for full and critical-phase takeover. (c) Position traces with local magnification.

Can the specialist's competence be absorbed into the VLA? Direct fine-tuning on 10 specialist trajectories reduces success from 100% under critical-phase takeover to 60% (18/30 trials). The student reproduces the specialist's jitter, copying incidental action fluctuations along with useful corrections. A successful rollout records one way of completing the task, but does not indicate whether its local action variations were necessary or avoidable. To examine state coverage, we compare the teacher's and offline-distilled student's RL-token representations using PCA (Figure 3). The projected distributions overlap, while student observations also extend beyond the teacher's main region. Once the student's own actions lead away from the recorded trajectories, it must recover from deviations for which the dataset may contain few corrective examples. Together, the inherited jitter and coverage mismatch suggest overfitting to a limited set of executions. Reliable transfer therefore requires refining the action targets and extending supervision to learner-visited states.

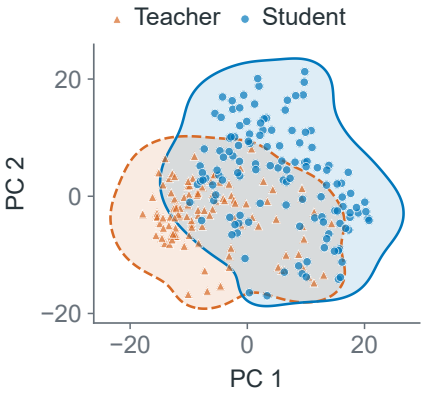


Figure 3: **State coverage after offline distillation.** PCA of RL-token observations. Shading marks 95% KDE regions.

4 From Local Specialists to Action-Expert Repair

REPAIR repairs the VLA's action expert using value-guided supervision from temporary phase-specific RL specialists. We first define the repair problem and its supervision requirements, then outline the framework before detailing supervision construction and the two distillation stages. The same supervision mechanism supports offline initialization and on-policy repair; the two stages differ in where observations are collected.

4.1 Problem Setup and Supervision Requirements

Motivated by Section 3, we study how to repair a pretrained VLA's action expert using temporary phase-specific RL specialists. We consider a family of long-horizon manipulation tasks represented as partially observed Markov decision processes. At time k, the policy receives observation o_k (images and proprioception), a language instruction ℓ, and predicts an action chunk $a_k \in \mathcal{A}^H$ of horizon H. The student $\pi_\theta(a \mid o, \ell)$ is the pretrained VLA to be repaired. A task contains $C \geq 1$ precision-critical phases. Let $g(o, \ell) \in \{0, \ldots, C\}$ denote the experimenter-supplied training-time phase annotation, where 0 indicates a noncritical observation. Each critical phase $c \in \{1, \ldots, C\}$ has a lightweight MLP specialist $\pi_{E,c}(a \mid o, \ell)$ and a corresponding critic $Q_{E,c}(o, \ell, a)$. For an observation with $g(o, \ell) = c > 0$, this pair supplies the specialist supervision.

Repair objective. We seek a single deployed VLA that completes the entire task with high success, efficient execution, and smooth control under a fixed real-robot interaction budget. The specialists are available as sources of training data and supervision, but are absent at deployment. Repair should improve behavior across the critical phases while retaining the VLA's competence in the remaining stages.

Two support gaps. Let ρ_E denote the observation distribution in successful specialist or hybrid rollouts and d^{π_θ} the distribution induced when the student acts alone. Offline trajectory distillation minimizes imitation error under ρ_E, but deployment error accumulates under d^{π_θ}. We call this mismatch the *state-support gap*. At a fixed state, the student may also fail to propose a useful recovery action, and a single executed expert action does not reveal the set of high-value alternatives. We call this the *action-support gap*.

Supervision requirements. These gaps call for supervision that covers the states encountered by the VLA when acting independently and provides useful expert action alternatives at those states. The supervision should distinguish actions by their expected contribution to task success, rather than treat every recorded or sampled expert action as equally desirable. The resulting problem is to construct

an expert-informed action supervision distribution on learner-induced states that enables the VLA itself to acquire the critical skill and correct its execution errors.

4.2 Framework Overview

Figure 4 organizes REPAIR into SFT initialization, phase-specific specialist training, and unified offline-to-on-policy distillation. In panel (a), human demonstrations $\mathcal{D}_H$ are used to fine-tune a pretrained VLA, yielding the base policy $\pi_{0.5}$ that initializes the student π_θ and supports specialist training.

Panel (b) prepares the local experts using RL Token (RLT) (Xu et al., 2026). With $\pi_{0.5}$ frozen, we train the lightweight actor $\pi_{E,c}$ and critic $Q_{E,c}$ for each critical phase through robot interaction with human interventions. The base policy handles noncritical stages, while each specialist learns its local skill. The trained RL actors are subsequently held fixed, and specialist-assisted rollouts provide observations and phase annotations for the offline buffer $\mathcal{D}_E$. Specialist-training details are deferred to Appendix B.2.

Panel (c) connects both observation sources to the same value-refined supervision mechanism. At a critical observation, the phase annotation selects one specialist, which proposes multiple action candidates; its corresponding critic determines their relative weights in the distillation target. The VLA's action expert learns from these targets in critical phases and imitates stored VLA actions in noncritical phases. Offline distillation first draws observations from $\mathcal{D}_E$ to establish the critical skills, generating expert actions anew instead of treating executed actions as fixed labels.

After this warm start, student-only rollouts in panel (c) supply observations to $\mathcal{D}_\pi$. The specialists provide action candidates and value estimates at these learner-induced states without taking control. Both stages use the same supervision rule and training objective. Only the repaired VLA is retained at deployment.

4.3 Value-Guided Offline-to-On-Policy Distillation

Value-refined expert targets. For each critical observation, we write π_E and Q_E for the specialist and critic selected by $g(o, \ell)$, suppressing the phase subscript in the following equations. To meet the requirements in Section 4.1, we construct a value-tilted expert distribution

$$p_E^Q(a \mid o, \ell) \propto \pi_E(a \mid o, \ell) \exp\left(\frac{A_E(o, \ell, a)}{\tau}\right), \tag{1}$$

where A_E is a within-state relative value and τ controls how sharply the critic emphasizes high-value expert actions. REPAIR approximates this distribution with a finite candidate set and trains the VLA by weighted flow matching.

For a critical observation (o, ℓ), draw N action chunks

$$a_i \sim \pi_E(\cdot \mid o, \ell), \qquad q_i = Q_E(o, \ell, a_i), \qquad i = 1, \ldots, N. \tag{2}$$

To remove arbitrary per-state shifts and scales in the critic, compute group-relative scores

$$\widehat{A}_i = \frac{q_i - \bar{q}}{\max(\sigma_q, \epsilon)}, \qquad w_i = \frac{\exp(\widehat{A}_i/\tau)}{\sum_{j=1}^N \exp(\widehat{A}_j/\tau)}, \tag{3}$$

where $\bar{q}$ and σ_q are the mean and standard deviation across candidates, and ϵ stabilizes the denominator. The weights emphasize high-value alternatives within the selected specialist's candidate set.

Unified distillation objective. Let $\mathcal{L}_{\mathrm{FM}}(\theta; o, \ell, a)$ denote the VLA's standard flow-matching loss for target action chunk a (Lipman et al., 2023). We combine the candidate losses using their critic-derived weights:

$$\mathcal{L}_{\mathrm{VRFM}}(\theta; o, \ell) = \mathbb{E}_{\{a_i\} \sim \pi_E} \left[\sum_{i=1}^N w_i \, \mathcal{L}_{\mathrm{FM}}(\theta; o, \ell, a_i) \right]. \tag{4}$$

For noncritical observations, $\mathcal{L}_{\mathrm{FM}}$ imitates the stored action a^{r}: recorded VLA actions offline and student predictions online. For replay sampling distribution ρ, the unified objective is

$$\mathcal{L}(\theta; \rho) = \mathbb{E}_{(o,\ell,a^{\mathrm{r}}) \sim \rho}[\mathbb{I}\{g(o, \ell) > 0\}\mathcal{L}_{\mathrm{VRFM}} + \mathbb{I}\{g(o, \ell) = 0\}\mathcal{L}_{\mathrm{FM}}]. \tag{5}$$

Noncritical supervision becomes self-distillation on student rollouts.

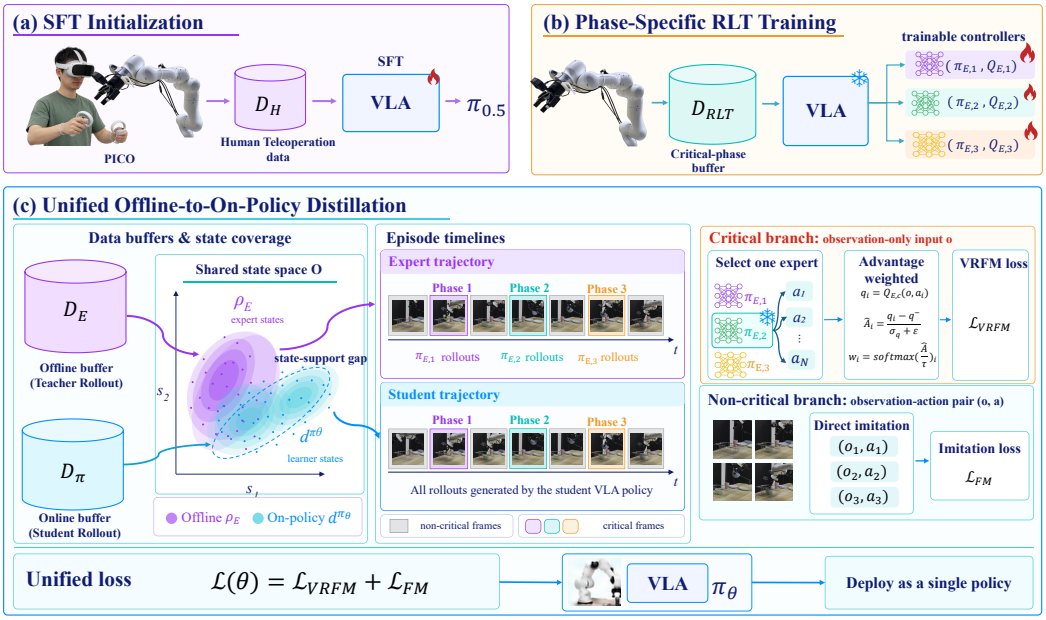


Figure 4: **Training framework of REPAIR.** (a) Human demonstrations initialize the base VLA $\pi_{0.5}$. (b) With this VLA frozen, critical-phase RL trains multiple specialist actors and their critics. (c) Offline observations from $\mathcal{D}_E$ and on-policy observations from $\mathcal{D}_\pi$ feed a shared value-refined supervision mechanism. For each critical observation, the selected phase specialist generates the action candidates and its critic determines their relative weights. The VLA learns from these targets in critical phases and replayed VLA actions in noncritical phases. Only the VLA is deployed.

From offline initialization to on-policy repair. After offline training on $\mathcal{D}_E$, student-only rollouts populate $\mathcal{D}_\pi$ and are added to the same replay. The critics are frozen offline but updated during on-policy repair to mitigate value-estimation errors on student-visited states. Equation 5 applies throughout with fixed RL actors, extending repair to learner-induced errors without expert takeover and yielding a single VLA for deployment. Hyperparameters are provided in Appendix B.1.

5 Main Experiments

Our main experiments evaluate whether REPAIR repairs long-horizon task execution while retaining a single VLA at deployment. We compare task success and motion smoothness across six real-robot tasks: two short-horizon tasks and four long-horizon tasks.

Tasks and evaluation protocol. Figure 5 illustrates the task sequences. The two short-horizon tasks are **Stick Pick-and-Place**, which requires grasping a stick and placing it on a plate, and **Single-Arm Peg Insertion**, which starts with a held peg and requires aligning and inserting it into a fixture. The four long-horizon tasks involve multiple sequential subgoals: **Dual-Arm Peg Insertion** requires picking up the socket and peg with separate arms, aligning the parts, and completing insertion; **Grocery Bagging** requires loading a beverage bottle, a chip container, and a biscuit packet into a bag before lifting it by its handles; **Bimanual Table Clearing** requires transferring two bowls to a draining basket, with each bowl grasped, handed over between arms, and placed in the basket; and **Toy Stowing in a Drawer** requires opening a drawer, placing two toys inside in sequence, and fully closing it. Each task contains one or more precision-critical phases. We evaluate each method for 30 trials on each task. Appendix C.1 specifies the task configurations, critical phases, success criteria, and representative failure modes.

Compared methods. We compare REPAIR with five baselines: **Base SFT**, the VLA fine-tuned on human demonstrations without subsequent RL adaptation; **RLT** (Xu et al., 2026), which retains the base VLA outside critical phases and uses the corresponding RLT specialist within each critical phase; **EXPO-FT** (Dong et al., 2026), which fine-tunes the VLA via RL using a learned residual policy

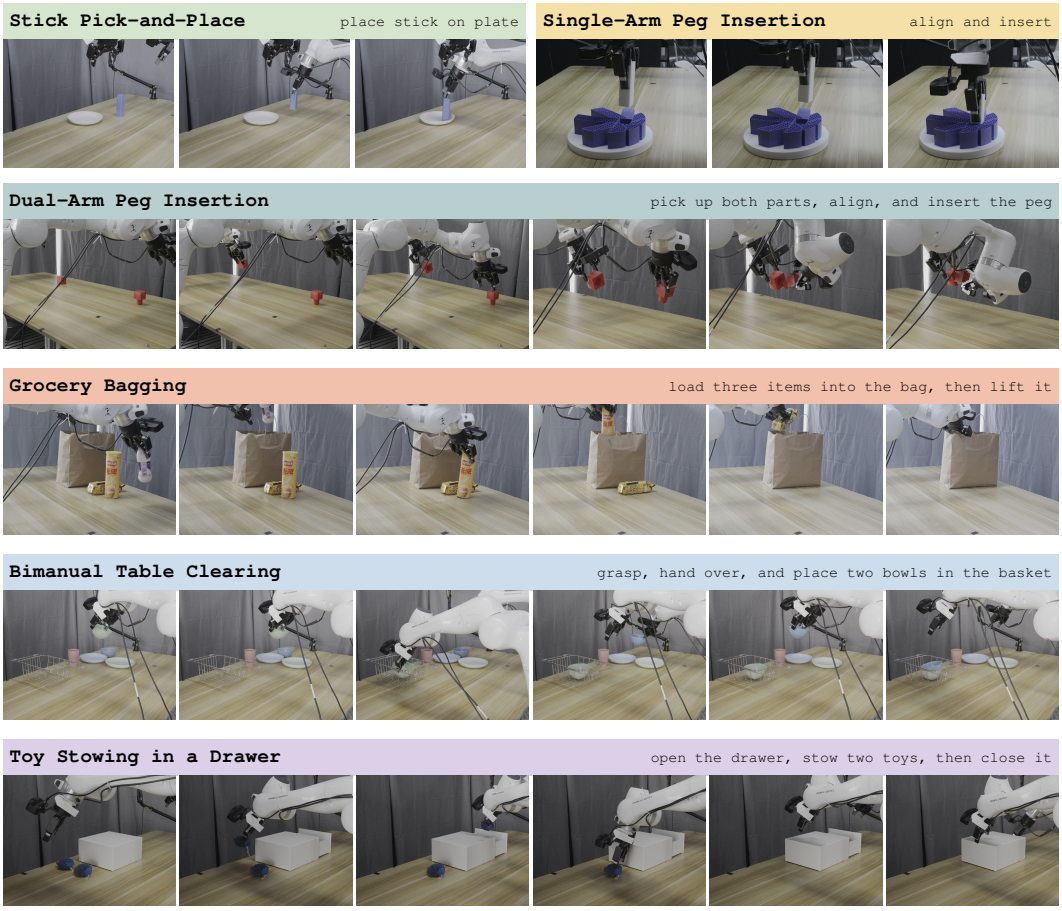


Figure 5: The tasks in our experiments.

to refine VLA-generated action chunks, with Q-guided selection; **ConRFT** (Chen et al., 2025b), which adapts the VLA through offline-to-online reinforcement learning with a consistency-policy objective; and **HG-DAgger** (Kelly et al., 2019), which aggregates human takeover demonstrations for supervised policy updates.

Training budgets. For SFT initialization, all methods use 50 human-teleoperated trajectories per short-horizon task and 100 per long-horizon task (Appendix C.2). To ensure a fair comparison, we allocate the same adaptation rollout budget of 50 trajectories per task to all methods, excluding human demonstrations. Each online baseline uses this budget entirely for training, while REPAIR distributes it across specialist training and distillation. Specifically, we first train an RLT teacher, denoted RLT*, using 30 online trajectories per task. We then freeze this checkpoint and collect 10 additional specialist-assisted trajectories for offline distillation, followed by 10 trajectories generated by the student VLA for on-policy distillation. Thus, including specialist training and data collection for both distillation stages, REPAIR uses the same total of 50 adaptation trajectories per task as the baselines. After distillation, REPAIR deploys only the repaired VLA, without specialist takeover or critic-based action selection.

Metrics. **Success rate (SR)** is the percentage of evaluation episodes that satisfy the complete task's success criterion before timeout or a safety abort. **Jerk** measures motion smoothness through the time derivative of end-effector acceleration. For an episode with T control steps, let $\mathbf{x}_t^{(r)} \in \mathbb{R}^3$ be the measured Cartesian position of arm r at time $t\Delta t$, with R active arms. We use the mean magnitude

of the third-order finite difference:

$$J = \frac{1}{R(T-2)} \sum_{r=1}^{R} \sum_{t=3}^{T} \left\| \frac{\mathbf{x}_t^{(r)} - 3\mathbf{x}_{t-1}^{(r)} + 3\mathbf{x}_{t-2}^{(r)} - \mathbf{x}_{t-3}^{(r)}}{\Delta t^3} \right\|_2 . \tag{6}$$

Jerk is computed over each complete successful episode and reported as the mean $\pm$ standard deviation across successful evaluation trials, in $\mathrm{m\,s^{-3}}$.

Main comparison. Table 1 shows that REPAIR matches the observed task success of specialist takeover while deploying only a single VLA. Both REPAIR and RLT achieve 30/30 successes on each of the six tasks. RLT converges early in training, with the 30-trajectory RLT* teacher performing comparably to the 50-trajectory RLT baseline. We therefore omit a separate entry for RLT* in Table 1. The advantage over the other adaptation baselines is most pronounced on long-horizon tasks. EXPO-FT and ConRFT achieve 80.0–100.0% success on the two short-horizon tasks, but neither succeeds on Bagging, Table Clearing, or Drawer under the reported training budgets. HG-DAgger is more competitive on these tasks, yet its aggregate success rate across the four long-horizon tasks is 74.2%, compared with 100.0% for REPAIR, a gap of 25.8 percentage points. Alongside this high task success, REPAIR reduces mean jerk on successful trajectories by 29.8–67.5% relative to RLT across all six tasks. Its jerk values remain close to those of HG-DAgger, which has slightly lower means on four tasks, indicating that the benefit over this imitation baseline lies primarily in higher task success while maintaining a similar level of motion smoothness.

Table 1: Main results on six real-robot tasks (30 trials per method per task). SR: success rate (%, higher is better). Jerk: mean $\pm$ standard deviation over successful trials ($\mathrm{m\,s^{-3}}$, lower is better), with identical sampling and filtering across methods. Dashes denote no successful trials.

	Stick P&P		Single peg		Dual peg		Bagging		Table clearing		Drawer	
Method	SR ↑	Jerk ↓	SR ↑	Jerk ↓	SR ↑	Jerk ↓	SR ↑	Jerk ↓	SR ↑	Jerk ↓	SR ↑	Jerk ↓
Base SFT ($\pi_{0.5}$)	30.0	4.2 ± 1.2	33.3	3.3 ± 0.5	0.0	–	13.3	4.7 ± 0.9	20.0	4.7 ± 0.8	0.0	–
RLT	100.0	15.7 ± 2.0	100.0	4.7 ± 0.8	100.0	10.3 ± 0.9	100.0	7.2 ± 0.8	100.0	10.4 ± 2.5	100.0	16.7 ± 5.1
EXPO-FT	80.0	13.5 ± 0.8	93.3	4.2 ± 1.0	33.3	9.0 ± 1.3	0.0	–	0.0	–	0.0	–
ConRFT	93.3	12.8 ± 2.3	100.0	3.9 ± 0.6	20.0	7.1 ± 2.4	0.0	–	0.0	–	0.0	–
HG-DAgger	100.0	5.9 ± 1.2	60.0	3.1 ± 0.6	63.3	6.3 ± 1.1	90.0	4.4 ± 0.8	63.3	4.4 ± 0.6	80.0	10.1 ± 1.8
REPAIR (ours)	100.0	5.1 ± 1.0	100.0	3.3 ± 0.7	100.0	6.5 ± 1.6	100.0	4.6 ± 0.9	100.0	4.5 ± 0.7	100.0	9.9 ± 3.9

We further compare episode lengths on Dual-Arm Peg Insertion to assess whether smoother execution compromises efficiency (Figure 6). Across 30 successful trials per method, REPAIR reduces mean episode length by 4.7%, from 249.7 ± 21.9 to 238.0 ± 9.6 (mean $\pm$ SD). Together with 36.9% lower mean jerk (6.5 versus $10.3\,\mathrm{m\,s^{-3}}$), this indicates smoother execution with shorter, less variable episodes. At a common control timestep, shorter episodes would also imply faster completion. Fewer alignment corrections during controller handoffs or specialist execution may contribute, although this mechanism remains unverified.

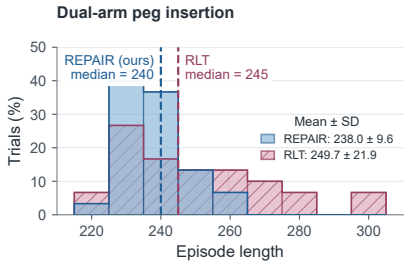


Figure 6: Dual-arm peg insertion episode lengths (30 trials per method; 100% success).

6 Analysis and Ablations

We ablate two key design choices in REPAIR: which actions the VLA learns from and which states it visits during distillation. The evaluation protocol and metrics follow Section 5.

RQ1: What supervision should be transferred from a specialist? To isolate the effect of supervision from state coverage and online interaction, we first study this question in an offline setting with a fixed expert buffer $\mathcal{D}_E$. We compare executed specialist actions, one re-queried expert action ($N = 1$), eight candidates with uniform weights, and the same candidate sets with critic-based weights, keeping Base SFT initialization, observation sampling, noncritical replay supervision, and

optimizer-update budgets fixed. Executed-action cloning follows the trajectory-supervision principle shared by RLDG (Xu et al., 2024) and PLD (Xiao et al., 2025): SFT on recorded specialist actions. It achieves 60.0% success with the highest mean jerk, 11.1 (Table 2). Consistent with Section 3, this suggests that limited-data cloning can overfit abrupt, aggressive RLT corrections alongside useful task behavior. Re-querying one action improves success to 66.7% and lowers jerk to 9.2, while uniform supervision over eight candidates reaches 76.7% and 7.28, respectively, suggesting a benefit from broader expert action support. Critic weighting of the same eight candidates further improves success to 90.0% and lowers jerk to 6.7. Relative to executed-action cloning, this is a gain of 30.0 percentage points in success rate and a 39.6% reduction in mean jerk on successful trials, supporting the use of multiple expert alternatives with value-based preferences.

To further evaluate value-refined action supervision during on-policy repair, we initialize all variants from the same VLA checkpoint obtained using REPAIR's offline training recipe. We compare value weighting with online critic updates, value weighting with a frozen critic, and uniform weighting of expert candidates. The latter two variants initially preserve the offline policy's performance but gradually degrade with continued training. Online critic updates progressively improve success to 100% and sustain this level. These trends suggest that adapting the critic to student-visited states helps maintain effective supervision during on-policy repair.

Table 2: **Which actions to learn.** Supervision targets on fixed specialist states (Dual Peg; 30 trials). SR in %; jerk: mean ± SD over successful trials ($\mathrm{m\,s^{-3}}$).

Action targets	N	SR ↑	Jerk ↓
Executed actions	–	60.0	11.1 ± 1.1
Re-queried action	1	66.7	9.2 ± 0.8
Uniform candidates	8	76.7	7.28 ± 1.3
Value-weighted candidates	8	90.0	6.7 ± 1.1

Table 3: **Which states to visit.** Specialist versus student rollouts under fixed action supervision (30 trials per task; SR in %).

	Rollouts		SR ↑	
Variant	Specialist	Student	Dual peg	Table clearing
Offline	10	0	90.0	73.3
	20	0	90.0	80.0
	30	0	93.3	83.3
	50	0	100.0	90.0
REPAIR	10	10	100.0	100.0

RQ2: Where do the gains from on-policy distillation come from? To test whether on-policy gains simply reflect more data, we compare larger offline specialist buffers with on-policy distillation on Dual Peg and Table Clearing (Table 3). All variants use 100 human demonstrations per task for SFT initialization and the same multi-candidate value-weighted supervision. At 20 total rollouts, offline distillation achieves 90.0% and 80.0% success on Dual Peg and Table Clearing, respectively, compared with 100.0% on both tasks for REPAIR's 10 specialist plus 10 student rollouts. This contrast supports a benefit from labeling learner-visited states beyond increasing trajectory count. Expanding offline buffers from 10 to 50 rollouts raises success from 90.0% to 100.0% and from 73.3% to 90.0%, respectively. The largest offline buffer thus matches on-policy performance on Dual Peg but remains 10 percentage points lower on Table Clearing, using 40 extra specialist rollouts versus 10 student rollouts. The pattern is consistent with a coverage-based explanation: larger specialist buffers may cover more recovery states, whereas on-policy distillation directly labels deviations produced as the student's execution errors accumulate.

7 Conclusion

We presented REPAIR, which distills local RL expertise into a single VLA using critic-weighted expert action candidates. A shared flow-matching objective supports offline initialization and on-policy refinement on learner-visited states, with specialists providing supervision without takeover. Across six real-robot tasks, REPAIR achieves 100% success on all six tasks, matching the observed success rates of specialist takeover while reducing mean end-effector jerk on successful trajectories by 29.8–67.5%. Ablations support multiple expert candidates with critic-based weighting, and data-scaling results favor on-policy supervision at matched rollout counts on the analyzed tasks. Training still relies on phase annotations, and critic rankings on learner states need direct validation. These findings support transferring specialists' critical skills into the VLA while removing auxiliary controllers at deployment.

Ethics Statement

The proposed system uses online robot interaction and may involve human interventions during specialist training. Experiments should therefore use workspace limits, emergency stops, force or torque thresholds where available, and an explicit protocol for reporting safety interventions and hardware incidents. No personal data are required for the proposed study.

Reproducibility Statement

We report the evaluation protocol, metrics, and training budgets in Section 5. Appendices A–C document the robot setup, specialist training procedure, distillation hyperparameters, baseline implementations, and task definitions.

AI Use Statement

Generative AI tools were used to assist with the implementation, debugging, and refinement of experimental code, as well as to improve the grammar of the manuscript. All AI-assisted code was reviewed and tested by the authors, and all AI-assisted edits to the manuscript were manually verified.

References

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1577–1594, 2023.

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

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

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Xiang Li, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, Tiejun Huang, Yu Wang, and Chao Yu. π_{RL}: Online RL fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025a. doi: 10.48550/arXiv.2510.25889.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. ConRFT: A reinforced fine-tuning method for VLA models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025b.

Yuhui Chen, Haoran Li, Zhennan Jiang, Yuxing Qin, Yuxuan Wan, Weiheng Liu, and Dongbin Zhao. Posterior optimization with clipped objective for bridging efficiency and stability in generative policy learning. *arXiv preprint arXiv:2604.01860*, 2026.

Perry Dong, Kuo-Han Hung, Tian Gao, Dorsa Sadigh, and Chelsea Finn. Expo-ft: Sample-efficient reinforcement learning finetuning for vision-language-action models. *arXiv preprint arXiv:2605.25477*, 2026.

Yiguo Fan, Shuanghao Bai, Xinyang Tong, Pengxiang Ding, Yuyang Zhu, Hongchao Lu, Fengqi Dai, Wei Zhao, Yang Liu, Siteng Huang, Zhaoxin Fan, Badong Chen, and Donglin Wang. Long-vla: Unleashing long-horizon capability of vision language action model for robot manipulation. 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. 2018–2037. PMLR, 27–30 Sep 2025.

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, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Zhennan Jiang, Kai Liu, Yuxin Qin, Shuai Tian, Yupeng Zheng, Mingcai Zhou, Chao Yu, Haoran Li, and Dongbin Zhao. World4rl: Diffusion world models for policy refinement with reinforcement learning for robotic manipulation. *arXiv preprint arXiv:2509.19080*, 2025.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, Yu Wang, Haoran Li, Chao Yu, and Dongbin Zhao. Wovr: World models as reliable simulators for post-training vla policies with rl. *arXiv preprint arXiv:2602.13977*, 2026.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083, 2019. doi: 10.1109/ICRA.2019.8793698.

Kun Lei, Huanyu Li, Dongjie Yu, Zhenyu Wei, Lingxiao Guo, Zhennan Jiang, Ziyu Wang, Shiyu Liang, and Huazhe Xu. RL-100: Performant robotic manipulation with real-world reinforcement learning. *arXiv preprint arXiv:2510.14830*, 2025.

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

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*, 38:97121–97151, 2025.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2025.

Stephane Ross, Geoffrey Gordon, and Drew 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, 2011.

Ziqian Wang, Yitian Liu, Xingjian Mao, Minqian Wang, and Yao Mu. Q-VGM: Q-value-gradient matching for offline-to-online reinforcement learning of flow-matching VLA. *arXiv preprint arXiv:2606.08015*, 2026.

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. *arXiv preprint arXiv:2511.00091*, 2025.

Charles Xu, Qiyang Li, Jianlan Luo, and Sergey Levine. RLDG: Robotic generalist policy distillation via reinforcement learning. *arXiv preprint arXiv:2412.09858*, 2024.

Charles Xu, Jost Tobias Springenberg, Michael Equi, Ali Amin, Adnan Esmail, Sergey Levine, and Liyiming Ke. RL Token: Bootstrapping online RL with vision-language-action models. *arXiv preprint arXiv:2604.23073*, 2026.

Rushuai Yang, Hangxing Wei, Ran Zhang, Zhiyuan Feng, Xiaoyu Chen, Tong Li, Chuheng Zhang, Li Zhao, Jiang Bian, Xiu Su, and Yi Chen. Beyond human demonstrations: Diffusion-based reinforcement learning to generate data for VLA training. *arXiv preprint arXiv:2509.19752*, 2025.

Chao Yu, Yuanqing Wang, Zhen Guo, Hao Lin, Si Xu, Hongzhi Zang, Quanlu Zhang, Yongji Wu, Chunyang Zhu, Junhao Hu, Zixiao Huang, Mingjie Wei, Yuqing Xie, Ke Yang, Bo Dai, Zhexuan Xu, Jiakun Du, Xiangyuan Wang, Xu Fu, Letong Shi, Zhihao Liu, Kang Chen, Weilin Liu, Gang Liu, Boxun Li, Jianlei Yang, Zhi Yang, Guohao Dai, and Yu Wang. RLinf: Flexible and efficient Large-Scale reinforcement learning via Macro-to-Micro flow transformation. In *20th USENIX Symposium on Operating Systems Design and Implementation (OSDI 26)*, pp. 829–846, Seattle, WA, July 2026a. USENIX Association. ISBN 978-1-939133-55-7.

Dongjie Yu, Kun Lei, Zhennan Jiang, Jia Pan, and Huazhe Xu. Beyond action residuals: Real-world robot policy steering via bottleneck latent reinforcement learning. *arXiv preprint arXiv:2605.19919*, 2026b.

Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. In *International Conference on Learning Representations*, 2025.

Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, et al. Rlinf-vla: A unified and efficient framework for vla+ rl training. *arXiv preprint arXiv:2510.06710*, 2025.

Hongzhi Zang, Shu'ang Yu, Hao Lin, Tianxing Zhou, Zefang Huang, Zhen Guo, Xin Xu, Jiakai Zhou, Yuze Sheng, Shizhe Zhang, Feng Gao, Wenhao Tang, Yufeng Yue, Quanlu Zhang, Xinlei Chen, Chao Yu, and Yu Wang. Rlinf-user: A unified and extensible system for real-world online policy learning in embodied ai. *arXiv preprint arXiv:2602.07837*, 2026.

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

Tianxing Zhou, Haojia Ao, Haoyang Lu, Guangyan Chen, Zichen Zhou, Te Cui, Chao Yu, and Yufeng Yue. Human-guided online reward adaptation for real-robot arm manipulation. *IEEE Robotics and Automation Letters*, 11(8):9072–9079, 2026. doi: 10.1109/LRA.2026.3699150.

A Franka Robot Setup

We conduct real-world experiments on a bimanual platform with two 7-DoF Franka arms, each equipped with a parallel-jaw gripper. The policy receives one third-person RGB view, two wrist-mounted RGB views, the language instruction, and the proprioceptive state of both arms. For each arm, the state consists of the TCP position, a 6D rotation representation, and the gripper state, resulting in a 20-D bimanual state

$$\boldsymbol{s}_t = [\boldsymbol{p}_t^L, \boldsymbol{r}_{6D,t}^L, g_t^L, \boldsymbol{p}_t^R, \boldsymbol{r}_{6D,t}^R, g_t^R].$$

The policy predicts 20-D bimanual actions with the same per-arm layout. While the robot interface executes absolute TCP waypoints, we train $\pi_{0.5}$ using body-frame SE(3) relative actions,

$$\Delta \boldsymbol{T}_{t,k}^i = (\boldsymbol{T}_t^i)^{-1} \boldsymbol{T}_{t+k}^i, \qquad i \in \{L, R\},$$

and recover absolute TCP targets before execution. We use an action horizon of $H = 10$, and execute the bimanual policy at $10\,\mathrm{Hz}$.

B Implementation Details

B.1 REPAIR

We implement REPAIR with RLinf (Yu et al., 2026a; Zang et al., 2025; 2026). During distillation, we update the VLA's action expert while keeping the backbone, RL-token encoder, and specialist actors frozen. The RLT critics are frozen during offline distillation and updated alongside the VLA during online repair to mitigate Q-value overestimation at previously unseen states. A frozen copy of $\pi_{0.5}$ supplies the specialists' features and reference actions. Candidate scores use the minimum of the two target critics, and replay minibatches contain equal numbers of critical and noncritical observations. Table 4 lists the distillation hyperparameters.

Table 4: Distillation hyperparameters for REPAIR. Batch size counts observations before candidate expansion.

Setting	Value
Specialist candidates per critical observation	8
Q-weight temperature τ	0.1
Reference dropout probability	0.5
Optimizer	AdamW
Learning rate / schedule	5×10^{-6} / constant
AdamW (β_1, β_2) / ϵ	$(0.9, 0.95)$ / 10^{-8}
Weight decay	0.01
Global batch size	64
Optimizer steps per learner iteration	1
Computation precision	BF16

B.2 RLT

RLT trains phase-specific actors and critics while keeping the base VLA $\pi_{0.5}$ and its RL-token encoder frozen.

Architecture. Each critical phase $c \in \{1, \ldots, C\}$ has an actor $\pi_{E,c}$ taking $[\mathrm{vec}(\bar{a}_t), z_t, p_t]$ and twin critics taking $[z_t, p_t, \mathrm{vec}(a_t)]$. Both use three 512-unit hidden layers with tanh activations; the critics additionally use layer normalization. Specialist and reference chunks have horizon $H = 10$ and share the replay action coordinates. The actor samples

$$\mathrm{vec}(a_t) = \tanh(\mu_c(\bar{a}_t, z_t, p_t) + \sigma\epsilon)\,, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{7}$$

Phase-local replay. The phase annotation activates the frozen VLA for $g = 0$ and specialist c for $g = c > 0$, with human overrides allowed. Replay stores executed actions and intervention indicators from successful and failed critical-phase segments. Each segment terminates on local success or failure, with reward one for local success and zero otherwise. Later phase outcomes do not alter earlier rewards, and noncritical segments are excluded.

Critic update. For $s_t = (z_t, p_t)$, phase-termination indicator d_t, and next action $a' \sim \pi_{E,c}(\cdot \mid s_{t+H}, \bar{a}_{t+H})$, the chunk TD target is

$$y_t = \sum_{j=0}^{H-1} \gamma^j r_{t+j} + (1 - d_t)\gamma^H \min_{k \in \{1,2\}} Q^-_{c,k}(s_{t+H}, a'), \qquad \gamma = 0.96, \tag{8}$$

where $Q^-_{c,k}$ denotes a target critic. Both critics regress to y_t on executed replay chunks, and target parameters use an exponential moving average with coefficient $\eta = 0.005$ after each critic step. Neither the TD target nor the actor objective includes an entropy term.

Actor update and intervention supervision. The actor maximizes the first critic with a behavior-cloning (BC) penalty. The target $b_{t,j}$ is the executed action at steps containing human intervention and the frozen VLA reference action otherwise. With validity mask $m_{t,j}$ and actor sample $\tilde{a}_t$ from Equation 7, the loss is

$$\mathcal{L}_{\text{actor}} = -\lambda_Q \, \mathbb{E}[Q_{c,1}(s_t, \tilde{a}_t)] + \lambda_{\text{BC}} \frac{\sum_{t,j} m_{t,j} \|\tilde{a}_{t,j} - b_{t,j}\|_2^2}{d_a \sum_{t,j} m_{t,j}}, \qquad \lambda_Q = \lambda_{\text{BC}} = 1. \tag{9}$$

Short terminal chunks use repeat-last padding. Reference dropout zeros the entire actor reference input with probability 0.5 during actor updates and training rollouts; BC targets and next-action sampling for the critic target retain the full reference.

Table 5: RLT training hyperparameters.

Setting	Value
RL-token	2048
proprioceptive dimensions	20
Action dimension	20
chunk horizon	10
Actor / critic hidden widths	$(512, 512, 512)$ / $(512, 512, 512)$
Reference dropout probability	0.5
Q / BC objective weights	1 / 1
Discount / target-update coefficient	0.96 / 0.005
Actor / critic learning rate	10^{-4} / 10^{-4} (constant)
Actor / critic gradient-norm clipping	10 / 10
Minibatch size	256
Critic-to-actor update ratio	4:1

B.3 Other Baselines

EXPO-FT. Following the official implementation (Dong et al., 2026), we train a lightweight edit policy over VLA action chunks and select the highest-Q candidate from the base and edited actions. The VLA is updated with its original supervised objective on replay data, including human corrections.

ConRFT. Following the official implementation (Chen et al., 2025b), we initialize the consistency policy and critic on offline demonstrations using behavior cloning and calibrated Q-learning. Online training retains the consistency-based actor objective and uses demonstrations, policy rollouts, and human interventions.

C Task Descriptions and Demonstration Budgets

C.1 Task Descriptions

All six tasks use the Franka FR3 platform with Robotiq grippers described in Appendix A. The descriptions below specify the initial conditions, task sequence, success criteria, precision-critical phases, and representative failure modes. Task names follow Section 5.

Single-Arm Peg Insertion. A white rectangular peg must be inserted into a matching opening in an irregularly shaped fixture secured to the table. The peg starts in the gripper, with horizontal and vertical offsets from the opening of approximately 20 cm or less. The initial configuration is fixed across episodes. The robot approaches the opening, aligns the peg, and advances it to full insertion depth. Success requires the peg to be fully seated at the bottom of the opening within the allotted time. The critical phase spans initial alignment through completed insertion, covering nearly the entire task. Precise positional and angular alignment must be maintained throughout: misalignment can prevent entry or obstruct advancement after partial insertion. Typical failures are failure to enter the opening and incomplete insertion before timeout (Figure 7).

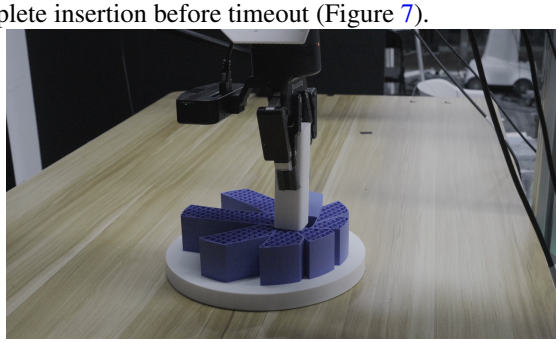
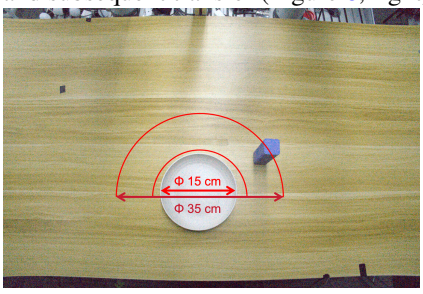


Figure 7: Single-Arm Peg Insertion failure. Misalignment prevents the peg from entering the opening and completing insertion within the allotted time.

Stick Pick-and-Place. The robot grasps a purple rectangular stick and places it inside a white circular plate fixed to the table. The stick's initial position is randomized within the marked semi-annular region around the plate (Figure 8, left). Its location determines the executing arm: the left arm handles the left side and the right arm the right side. The selected arm grasps and lifts the stick, transports it to the plate, and releases it inside. Success requires both pickup and placement. The critical phase is grasp acquisition, from final approach through gripper closure and initial lifting. This combines spatially conditioned arm selection with accurate gripper–object alignment and a stable grasp across randomized positions. The predominant failure is an insecure grasp that prevents lifting and subsequent transfer (Figure 8, right).



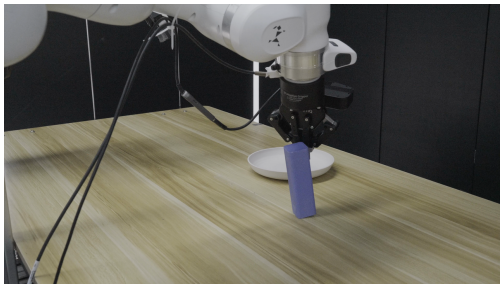


Figure 8: Stick Pick-and-Place. Left: the red boundaries mark the object initialization region. Right: the gripper fails to establish a secure grasp. Dimensions and region annotations are retained from the supplied task illustration.

Dual-Arm Peg Insertion. The robot assembles two complementary components, one with a protruding peg and the other with a matching socket. The left arm first grasps and lifts the socket, followed by the right arm grasping and lifting the peg. The robot then brings the components together, aligns them, and performs insertion while holding both. Success requires completed assembly within the allotted time. The critical phase comprises relative alignment and insertion after both grasps. Motion of either arm changes the relative pose, so this phase requires coordinated positional and angular control while maintaining stable grasps. Failures include repeated unsuccessful insertion attempts until timeout (Figure 9) and premature release by the right gripper, causing the peg component to fall before engagement.

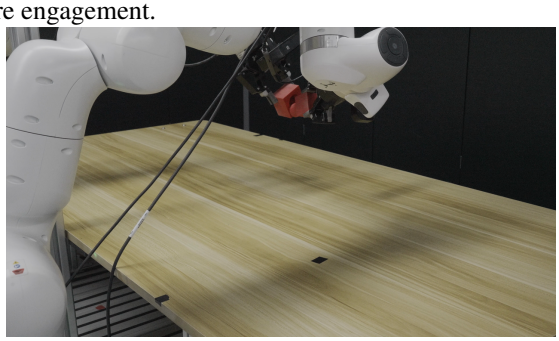


Figure 9: Dual-Arm Peg Insertion failure. Misalignment between the held components prevents the peg from entering the socket despite repeated attempts.

Toy Stowing in a Drawer. Two plush toys start beside a drawer unit secured to the tabletop. The right arm opens the drawer, the left arm picks up and places the toys inside sequentially, and the right arm closes the drawer. Success requires both toys to be contained inside and the drawer to be fully closed. There are two critical phases: placement of the second toy by the left arm and final drawer closure by the right arm. The second placement must deposit the toy inside the drawer, while closure must complete the drawer's full travel. Typical failures are depositing a toy outside the drawer and leaving the drawer partially open (Figure 10).

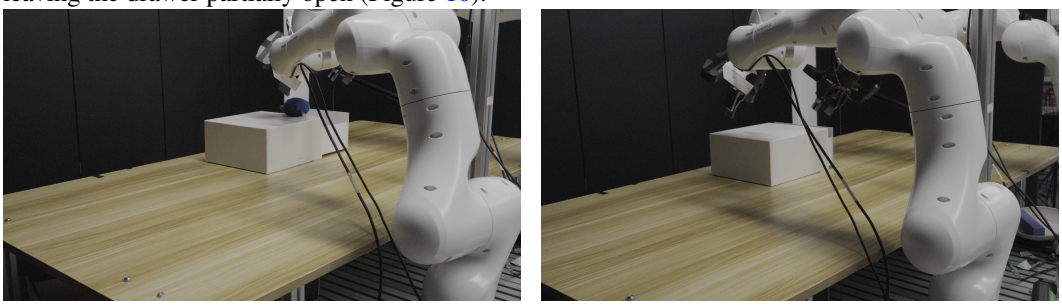
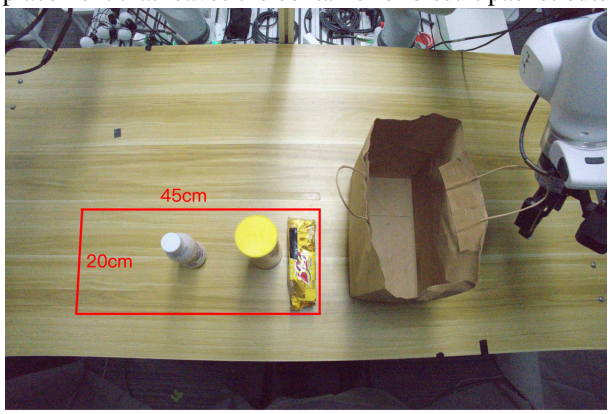


Figure 10: Toy Stowing in a Drawer failures. Left: the toy is deposited on top of the drawer housing. Right: the drawer remains partially open after the closing attempt.

Grocery Bagging. The robot packs a beverage bottle, a cylindrical chip container, and a biscuit packet into a bag, then lifts it by both handles. The bag starts at a fixed location; object positions are randomized within the annotated region in Figure 11 (top), with the biscuit packet lying flat. The left arm raises and holds the left handle while the right arm places the bottle, chip container, and biscuit packet into the bag in sequence. The right arm then grasps the right handle, and both arms lift the loaded bag. Success requires all three objects inside the bag and lifting by both handles. The three critical phases are grasping the chip container, placing it into the bag, and placing the biscuit packet. The cylindrical container requires precise two-finger grasping, and placements require alignment with a bag opening that may deform after unintended contact. Failures include an insecure chip-container grasp and inaccurate placement that leaves the container or biscuit packet outside the bag.



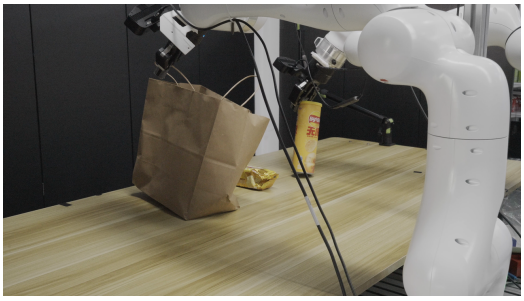
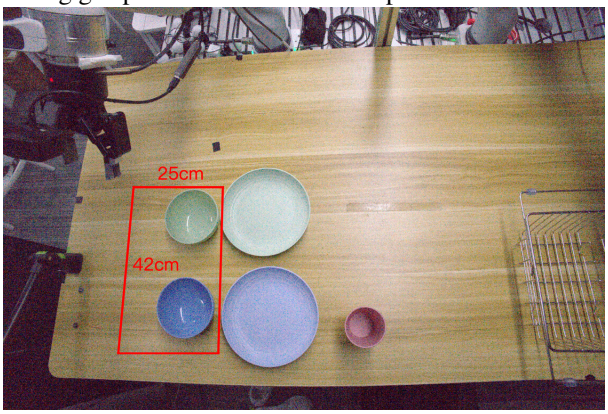




Figure 11: Grocery Bagging setup and failures. Top: the red region defines randomized object initialization. Bottom left: the right gripper fails to secure the chip container. Bottom right: inaccurate alignment prevents placing the container through the bag opening.

Bimanual Table Clearing. Two bowls must be transferred from the table to a dish-drying basket through sequential handovers. The setup also includes two plates and a cup. The bowls start at randomized positions within the marked region in Figure 12 (top); the other items and basket remain fixed. The right arm picks up the green bowl and hands it to the left arm for placement in the basket. The sequence is repeated with the blue bowl, which is placed on top of the green bowl. Success requires both bowls to be stacked inside the basket. The two critical phases are grasping the blue bowl and transferring it from right to left. During handover, the left gripper must securely engage the rim before the right gripper releases it. Failures include missing the blue bowl during pickup and a shallow or missed receiving grasp that lets the bowl fall upon release.



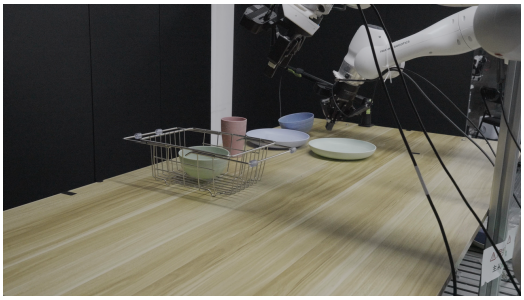




Figure 12: Bimanual Table Clearing setup and failures. Top: the red region defines bowl initialization. Bottom left: the right gripper misses the blue bowl's rim. Bottom right: the left gripper fails to secure the bowl during handover, causing it to fall when released.

C.2 Demonstration Scaling and SFT Budgets

Increasing the number of human demonstrations does not necessarily resolve the difficult behaviors in these tasks. Table 6 examines whether increasing teleoperation data resolves failures on two difficult long-horizon tasks using human-only SFT, without specialist distillation. Dual-Arm Peg Insertion remains at 0% success with 50, 100, 200, and 300 demonstrations: even a sixfold increase yields no successful evaluation episodes under the tested SFT setup. This task's difficulty therefore persists despite substantial data expansion. Bimanual Table Clearing benefits from additional data, but with diminishing observed gains: 100, 150, and 300 demonstrations yield 20.0%, 26.7%, and 30.0% success. The first 50 additional trajectories add 6.7 percentage points, whereas the next 150 add only 3.3 points.

These results motivate keeping demonstration budgets fixed within each task-horizon group instead of continually expanding datasets for difficult tasks. For SFT initialization in the main experiments, we use 50 human-teleoperated trajectories per short-horizon task and 100 per long-horizon task. The budget is identical across methods for each task, with subsequent adaptation addressing the remaining precision-critical failures. Both tasks in Table 6 are long-horizon tasks; buffer sizes other than 100 trajectories are supplementary scaling controls.

Table 6: Human-only SFT scaling on two long-horizon tasks (30 evaluation trials per setting). Trajectory counts refer to human teleoperation demonstrations. The main experiments use 100 demonstrations for each of these tasks; other buffer sizes are supplementary controls. SR is reported in %; dashes denote untested settings.

Human trajectories	Dual peg SR ↑	Table clearing SR ↑
50	0.0	–
100	0.0	20.0
150	–	26.7
200	0.0	–
300	0.0	30.0