CRL-VLA: CONTINUAL VISION–LANGUAGE–ACTION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Lifelong learning is critical for embodied agents in open-world environments, where reinforcement learning fine-tuning has emerged as an important paradigm to enable Vision-Language-Action (VLA) models to master dexterous manipulation through environmental interaction. Thus, Continual Reinforcement Learning (CRL) is a promising pathway for deploying VLAs in lifelong robotic scenarios, yet balancing stability (retaining old skills) and plasticity (learning new ones) remains a formidable challenge. We introduce CRL-VLA, a framework for continual post-training of VLA models with theoretical performance-change bounds. The bounds relate old- and new-task return changes to goal-conditioned advantage magnitude and policy divergence. Motivated by this sensitivity analysis, the revised CRL-VLA formulation combines a frozen retention action critic, a trainable adaptation action critic, and policy regularization for retention and adaptation. Evaluations in simulation and on physical robots show that CRL-VLA supports continual adaptation across diverse VLA RL post-training paradigms and robot embodiments while maintaining robust task performance.

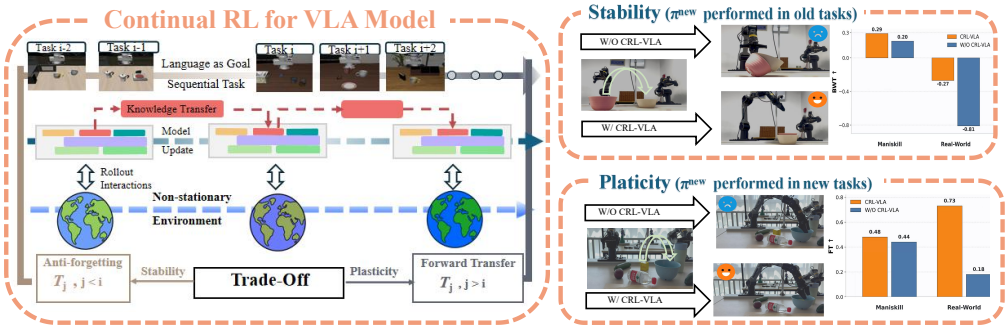

Figure 1: CRL-VLA is a continual learning framework for VLA that treats the goal-conditioned advantage magnitude as a key factor for the trade-off between stability and plasticity.

Reinforcement Learning (RL) post-training has emerged as a promising approach for improving Vision–Language–Action (VLA) policies through interaction and reward feedback (Physical Intelligence et al., 2025; Guo et al., 2025). By optimizing policies through interaction, RL post-training enables adaptive, goal-directed behaviors beyond what supervised learning can provide. Sustaining such adaptability requires lifelong learning across non-stationary task streams. However, in large-scale settings, Continual Reinforcement Learning (CRL) aims to equip agents with the ability to acquire new skills over time without catastrophic forgetting (Khetarpal et al., 2022). This problem fundamentally requires balancing scalability, stability, and plasticity under non-stationary task distributions (Pan et al., 2025). Despite its importance, CRL for VLA remains an open question, particularly in robotic manipulation scenarios involving multi-modal observations and long-horizon dynamics (Pan et al., 2025).

Purely sequential learning can retain strong plasticity on new tasks, but it often suffers from catastrophic forgetting. Existing CRL methods therefore improve stability through regularization or rehearsal, yet these strategies face limitations when applied to modern VLA architectures. Experience

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.

replay can mitigate forgetting in continual RL (Rolnick et al., 2019). Its computational trade-offs depend on the replay budget and learning setting; related evidence from continual language-model pretraining also highlights this dependence (Abbes et al., 2026). Parameter-importance regularization, including elastic weight consolidation, constrains changes to weights important for previous tasks (Kirkpatrick et al., 2017; Kutalev and Lapina, 2021). Applying these methods to large VLA models motivates attention to the cost of estimating and storing importance weights. KL regularization constrains changes relative to a reference policy (Korbak et al., 2022), and on-policy RL can exhibit an implicit bias toward smaller KL changes (Shenfeld et al., 2026). These findings motivate studying complementary value-consistency signals for continual VLA adaptation. Parameter-efficient adaptation can preserve pretrained multimodal capabilities: VLM2VLA represents actions as natural language and uses LoRA to limit changes to the VLM backbone (Hancock et al., 2026). Preserving such capabilities during continual RL post-training remains a distinct challenge.

We study continual VLA adaptation through goal-conditioned performance-change bounds and a framework that combines value retention with policy regularization. Our contributions are summarized as follows:

- We derive unified goal-conditioned performance-change bounds that characterize how advantage magnitude and policy divergence jointly bound return changes on old and new tasks, providing a theoretical perspective on the stability–plasticity trade-off.
- We propose **CRL-VLA**, a dual-critic framework that retains historical values, learns current-task values, and regularizes the policy on both historical and current-task data.
- Experiments in simulation and on real robots provide evidence that CRL-VLA improves retention or final performance across the evaluated VLA RL post-training settings, spanning online and offline learning and on- and off-policy methods.

2 Related work

2.1 Reinforcement Learning for VLA Models

VLAs have gained interest as unified robotic manipulation policies, with recent work exploring online RL to address SFT limitations and improve generalization and long-horizon reasoning. SimpleVLA-RL (Li et al., 2026a) builds upon OpenVLA-OFT and GRPO, showing that online RL substantially improves long-horizon planning under demonstration scarcity. RIPT-VLA (Tan et al., 2025) applies the REINFORCE leave-one-out estimator. ARFM (Zhang et al., 2026) enhances VLA action models via adaptive offline RL, balancing signal and variance to improve generalization and robustness. RLinf-VLA (Zang et al., 2025) implements a novel hybrid fine-grained pipeline allocation mode scaling up VLA training. EXPO-FT enables sample-efficient online reinforcement learning of pretrained vision-language-action models on real robots by combining value-guided action-chunk optimization with human interventions to improve task reliability Dong et al. (2026). WAV brings offline model-based reinforcement learning to VLA systems by jointly learning a world model and a trajectory value function from collected data, then using value-guided latent inference to generate actions that address the limited long-horizon reasoning of direct action prediction Li et al. (2026b). In contrast, our work aims to investigate the knowledge transfer of VLA and their ability to learn new tasks stably and continuously.

2.2 Continual Reinforcement Learning

Continual reinforcement learning generally falls into three paradigms. Regularization-based methods constrain updates to important parameters using Fisher information (Kirkpatrick et al., 2017) or path integrals (Zenke et al., 2017), while KL penalties constrain changes relative to a reference policy (Korbak et al., 2022). On-policy RL can also exhibit an implicit bias toward smaller KL changes (Shenfeld et al., 2026). Replay-based strategies (Rolnick et al., 2019) mitigate forgetting by interleaving past experience. Continual World provides a robotic benchmark for evaluating retention and forward transfer (Wołczyk et al., 2021). Architecture-based approaches (Xu and Zhu, 2018) prevent interference by expanding network capacity, a strategy often impractical for large-scale pretrained models. Prior work transfers policy knowledge through distillation (Schwarz et al., 2018; Rusu et al., 2016a). Our focus is historical action-value consistency during language-conditioned

continual adaptation. Unlike these approaches, our work focuses on the VLA for general visual language manipulation tasks.

2.3 Continual Learning for VLA Models

Recent work (Yadav et al., 2026) improves policy stability under task variations, providing useful insights for subsequent lifelong learning approaches, although it is evaluated in a non-continual setting. Recently, several works have extended these paradigms to the VLA domain. Stellar-VLA (Wu et al., 2025) introduces a learned task–skill knowledge space and knowledge-guided expert routing for continual imitation learning. Info-VLA (Zhao et al., 2026) preserves historical vision–language–action alignment through replay-anchor contrastive learning and cross-modal mutual-information constraints. Concurrent work by Hu et al. (2026) systematically evaluates continual RL for pretrained VLAs and finds that sequential fine-tuning with LoRA can retain strong performance and zero-shot generalization. Unlike these works, we aim to investigate a continuous post-training framework for VLA and explicitly model the stability-plasticity trade-off between new and old tasks.

3 Preliminaries

We study continual post-training of VLA policies in a goal-conditioned reinforcement learning setting. Each task is formulated as a goal-conditioned MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{G}, P, r, \gamma \rangle$, where $s \in \mathcal{S}$ is the state, $g \in \mathcal{G}$ is a language instruction, and $a \in \mathcal{A}$ is a robotic control command. For a given policy π and goal g, the action-value function is

$$Q^\pi(s, a, g) = \mathbb{E}_{\tau \sim \pi, g}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, g) \,\middle|\, s_0 = s, a_0 = a \right].$$

It measures the expected cumulative return after taking action a in state s and subsequently following π. The state-value function $V^\pi(s, g)$ is the expected return from state s when actions follow π, equivalently

$$V^\pi(s, g) = \mathbb{E}_{a' \sim \pi(\cdot|s,g)}[Q^\pi(s, a', g)].$$

The advantage function is the difference between the action-value and state-value functions:

$$A^\pi(s, a, g) = Q^\pi(s, a, g) - V^\pi(s, g).$$

Thus, an action's advantage is its Q-value relative to the state-value baseline at the same state and goal. We also write $A^\pi_g(s, a) = A^\pi(s, a, g)$. For a measurable set $B \subseteq \mathcal{S}$, the discounted state occupancy measure is $d^\pi_g(B) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t \in B \mid g, \pi)$, and the corresponding state-action occupancy measure is $\rho^\pi_g(ds, da) = d^\pi_g(ds)\pi(da \mid s, g)$. These measures describe the states and actions visited by the goal-conditioned VLA policy $\pi(a \mid s, g)$. For initial-state distribution μ_g, the expected return can be written directly in terms of Q:

$$J_g(\pi) = \mathbb{E}_{s_0 \sim \mu_g,\, a_0 \sim \pi(\cdot|s_0,g)}[Q^\pi(s_0, a_0, g)].$$

Performance on old tasks under distribution $p_{\text{old}}(g)$ is $J_{\text{old}}(\pi) = \mathbb{E}_{g \sim p_{\text{old}}}[J_g(\pi)]$. New policy's performance π^{new} on new tasks under distribution $p_{\text{new}}(g)$ is $J_{\text{new}}(\pi^{\text{new}}) = \mathbb{E}_{g \sim p_{\text{new}}}[J_g(\pi^{\text{new}})]$. A pretrained VLA policy π_θ is adapted sequentially to a task stream $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$. At stage k, the agent receives offline data and/or collects online experience for $\mathcal{T}_k$, and updates to π^k using the corresponding post-training procedure. The available historical data and model state are specified separately for each implementation.

4 Methodology

CRL-VLA separates historical value retention from current-task value learning. We describe the action-value formulation and its backend-specific variants.

4.1 Motivation: Performance Sensitivity

At stage k, the policy learns task $\mathcal{T}_k$ while retaining earlier skills. For a reference policy π^{old}, let $M_g = \sup_{s,a} |A^{\pi^{\text{old}}}(s, a, g)|$, and let M_{old} and M_{new} denote its root-mean-square values over the

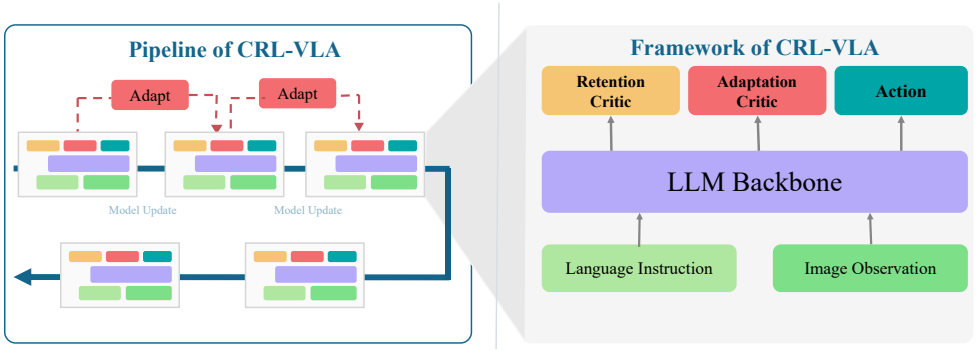

Figure 2: The double-critic framework and iterative training pipeline of CRL-VLA. The Retention Critic preserves knowledge for old tasks, while the Adaptation Critic estimates values for learning new tasks.

historical and current goal distributions p_{old} and p_{new}. Define nonnegative divergence quantities by

$$\begin{aligned} D_{\text{old}}^2 &= 2\mathbb{E}_{g\sim p_{\text{old}},\, s\sim d_g^{\pi^{\text{old}}}} D_{\text{KL}}\big(\pi^{\text{new}}(\cdot \mid s,g)\|\pi^{\text{old}}(\cdot \mid s,g)\big)\,, \\ D_{\text{new}}^2 &= 2\mathbb{E}_{g\sim p_{\text{new}},\, s\sim d_g^{\pi^{\text{new}}}} D_{\text{KL}}\big(\pi^{\text{new}}(\cdot \mid s,g)\|\pi^{\text{old}}(\cdot \mid s,g)\big)\,. \end{aligned} \tag{1}$$

Theorem 1 (Performance Sensitivity)**.** *For each goal, compare the complete executed-action policies, including planning, editing, or candidate selection, in the same MDP with the same initial-state distribution. With bounded rewards,* $0 \le \gamma < 1$*, finite RMS magnitudes* $M_{\text{old}}, M_{\text{new}}$*, and finite expected KL divergences defined above,*

$$\begin{aligned} |J_{\text{old}}(\pi^{\text{new}}) - J_{\text{old}}(\pi^{\text{old}})| &\le \frac{M_{\text{old}} D_{\text{old}}}{(1-\gamma)^2}, \\ J_{\text{new}}(\pi^{\text{new}}) - J_{\text{new}}(\pi^{\text{old}}) &\le \frac{M_{\text{new}} D_{\text{new}}}{1-\gamma}. \end{aligned} \tag{2}$$

The proof is provided in Appendix B.2. These bounds motivate considering value information alongside policy divergence. They characterize sensitivity, not guaranteed improvement: preserving sampled Q predictions does not directly minimize the true advantage magnitude. We therefore combine historical value supervision with current-task value learning and policy regularization.

4.2 Retention and Adaptation Critics

Let $z_\phi(s,g)$ encode the observation, robot state, and instruction, and let $\pi_{\phi,\omega}$ be the VLA policy. Policy parameters are carried forward from stage $k-1$. At stage $k>1$, we copy the preceding adaptation critic into a frozen teacher, including both its encoder and value head, and initialize the trainable student from the same checkpoint.

$$\begin{aligned} Q_k^{\text{ret}}(s,a,g) &= q_{\bar\psi_k}\left([z_{\bar\phi_k}(s,g),a]\right), \\ Q_k^{\text{adapt}}(s,a,g) &= q_{\psi_k}\left([z_{\phi_k}(s,g),a]\right), \qquad (\bar\phi_k,\bar\psi_k) = (\phi_{k-1},\psi_{k-1}). \end{aligned} \tag{3}$$

For historical memory $\mathcal{M}_{k-1}$ and current-task data $\mathcal{D}_k$, the student minimizes

$$\begin{aligned} \mathcal{L}_{\text{ret}} &= \mathbb{E}_{(s,a,g)\sim\mathcal{M}_{k-1}}\left[\left(Q_k^{\text{adapt}}(s,a,g) - \text{sg}\left[Q_k^{\text{ret}}(s,a,g)\right]\right)^2\right], \\ \mathcal{L}_{\text{adapt}} &= \mathbb{E}_{(s,a,g,y^{(b)})\sim\mathcal{D}_k}\left[\left(Q_k^{\text{adapt}}(s,a,g) - \text{sg}\left[y^{(b)}\right]\right)^2\right]. \end{aligned} \tag{4}$$

Here $y^{(b)}$ is a return or bootstrapped target supplied by backend b, and sg stops gradients. Both losses update the student encoder and head (ϕ,ψ), linking value supervision to the policy's shared

Table 1: Backend-specific mechanisms. Each variant has its own policy update; the complete explicit-KL objective is not shared by all backends.

Backend	Historical retention	Current-task learning
RIPT-VLA	Full-critic value distillation and historical policy KL	RLOO-based proximal updates and Monte Carlo value regression
WAV	Historical action rehearsal and value supervision through a frozen expert	Action and value flow supervision
Expo-FT	Frozen and adaptive critics jointly rank action candidates	TD learning, edit-policy updates, and successful-trajectory BC

representation. Historical targets use the current stage's frozen teacher and are refreshed at each stage transition. The student combines historical and current supervision; it is not assumed to equal the current policy's true Q-function on every task. The overall framework is shown in Fig. 2.

4.3 Policy Optimization and Backend Variants

At each stage, freeze a complete policy reference $\pi_k^- \leftarrow \pi^{k-1}$. With $\theta = (\phi, \omega)$, define separate penalties on historical and current state–goal distributions:

$$\mathcal{R}_x = \mathbb{E}_{(s,g)\sim\mu_{x,k}} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s,g) \| \pi_k^-(\cdot \mid s,g)\big), \quad x \in \{\mathrm{old}, \mathrm{new}\}. \tag{5}$$

These KL penalties require suitable common support. Here $\mu_{\mathrm{old},k}$ comes from memory and $\mu_{\mathrm{new},k}$ from current-task batches. The former limits historical behavioral drift; the latter regulates adaptation jointly with the actor objective. Both use the same stage-fixed reference, distinct from any refreshed rollout policy. These empirical distributions need not match the occupancies in Eq. 1. For the backend's actor-only loss $\mathcal{L}_\pi^{(b)}$ (Appendix C.2), the objective is

$$\mathcal{L}_{\mathrm{CRL}} = \mathcal{L}_\pi^{(b)} + \beta\mathcal{L}_{\mathrm{ret}} + \eta\mathcal{L}_{\mathrm{adapt}} + \alpha_{\mathrm{old}}\mathcal{R}_{\mathrm{old}} + \alpha_{\mathrm{new}}\mathcal{R}_{\mathrm{new}}, \tag{6}$$

with nonnegative weights. Value losses update (ϕ, ψ), whereas actor and KL losses update (ϕ, ω). Critic evaluators and their state features are fixed during actor optimization; action-input gradients may still pass to the actor. Teachers, policy references, and regression targets receive no gradients. Each iteration uses current-task minibatches together with historical minibatches when $k > 1$. A capacity-B reservoir ingests each distinct current-task record once and is committed at stage end; the historical memory used for training remains fixed within the stage. The policy and critic are then saved for the next stage. At $k = 1$, $\mathcal{L}_{\mathrm{ret}} = \mathcal{R}_{\mathrm{old}} = 0$ and $\pi_1^- = \pi^0$. Algorithm 1, actor objectives, target construction, and KL estimators appear in Appendix C.

Backend variants. The backends share the retention–adaptation principle but use distinct policy updates (Table 1). RIPT-VLA uses reverse-direction historical KL and batch-level proximal control, with $\alpha_{\mathrm{new}} = 0$; clipping is not the explicit stage-reference KL. WAV and Expo-FT use flow supervision and candidate selection, respectively, rather than the full objective in Eq. 6. Their targets and update rules are specified in Appendix C.

5 Experiments

We organize the evaluation around five questions: **(1)** how does CRL-VLA balance retention of earlier tasks with learning new tasks; **(2)** how does performance change under longer task sequences, visual disturbances; **(3)** how applicable is CRL-VLA across different learning settings, embodiments, and computational constraints; **(4)** and **(5)** how does CRL-VLA perform in real-world robotic environments? We first present the simulation protocols and overall comparisons, then examine longer sequences, applicability, and computational cost. Real-world evaluations follow, with ablations and mechanism analysis placed last.

5.1 Experimental Setup

Tasks and protocols. The simulation record comprises language-conditioned task sequences from LIBERO (Liu et al., 2023) and ManiSkill (Mu et al., 2021). The original LIBERO studies include

Table 2: Five-task LIBERO-Spatial results (%). Arrows indicate the preferred direction.

Type	Method	AVG ↑	NBT ↓	FWT ↑
Continual RL	SL (Hu et al., 2026)	81.2±0.4	0.3±0.5	3.9±1.5
	EWC (Kirkpatrick et al., 2017)	66.1±0.9	0.7±1.7	1.5±0.3
	ER (Rolnick et al., 2019)	80.2±0.5	0.6±1.1	−2.3±0.1
	DER (Buzzega et al., 2020)	73.4±1.3	4.7±1.3	0.7±0.9
	DWE (Hu et al., 2026)	79.6±0.9	0.0±0.0	0.0±0.0
	SLCA (Zhang et al., 2023)	69.9±0.7	−0.6±2.0	1.5±0.3
	RETAIN (Yadav et al., 2026)	66.0±0.7	2.9±1.4	1.4±1.4
Continual Imitation	Stellar-VLA (Wu et al., 2025)	17.1±3.6	5.2±9.9	−25.0±9.5
	Info-VLA (Zhao et al., 2026)	20.4±1.4	13.5±5.4	−29.7±4.1
Joint Training	MTL (Oracle) (Hu et al., 2026)	85.8±0.2	–	–
Ours	CRL-VLA	81.6±1.7	4.2±1.0	0.13±3.1

Dashes indicate that the complete initial checkpoint vector $S_{0,j}$ was not recorded.

three single-task stages in Spatial, two multi-task stages in Object, and a perturbed ten-task sequence. ManiSkill uses two stages containing multiple object-placement commands under sparse rewards. The documented task instructions and visual transformations appear in Appendix E and Appendix E.6. These settings differ in task composition and perturbations; their scores are interpreted within each protocol.

Backbones and optimization. The original simulation experiments use OpenVLA-OFT (Kim et al., 2025) on LIBERO and the flow-based $\pi_{0.5}$ (Black et al., 2025) on ManiSkill, with PPO as described in Appendix E.4. A separate five-task LIBERO-Spatial reference protocol follows Hu et al. (2026), using OpenVLA-OFT, LoRA, and GRPO (Appendix E.3). Sharing a task order does not make these PPO and GRPO experiments a controlled method comparison. For real-world setting, we use the Expo-FT (Dong et al., 2026), and WAV (Li et al., 2026c).

Baselines. The baselines include Sequential Learning (SL) (Liu et al., 2023), Experience Replay (ER) (Rolnick et al., 2019), Learning Without Forgetting (LWF) (Li and Hoiem, 2018), Multi-Task Learning (MTL) (Liu et al., 2023), Sequential Fine-Tuning (Seq. FT) (Hu et al., 2026), Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017), Expert Replay (Rolnick et al., 2019), Dark Experience Replay (DER) (Buzzega et al., 2020), Dynamic Weight Expansion (DWE) (Hu et al., 2026), SLCA (Layered LR) (Zhang et al., 2023), RETAIN (Yadav et al., 2026), Multi-Task (Oracle) (Hu et al., 2026), Stellar-VLA (Wu et al., 2025), Info-VLA (Zhao et al., 2026).

Evaluation metrics. Final average success (FAR) measures aggregate performance after the sequence; backward transfer (BWT) and forgetting (F) measure changes on earlier tasks (Wołczyk et al., 2021). We retain the original FT metric and separately report the reference protocol's AVG, NBT, and FWT. For the same success matrix, AVG equals FAR and NBT equals $-\mathrm{BWT}$. FWT uses the initial model's zero-shot task scores, whereas FT uses a separately trained single-task reference; the two are not interchangeable. Definitions are given in Appendix E.5.

5.2 Overall Continual Learning Performance

We first conduct continual learning experiments on LIBERO-Spatial using OpenVLA-OFT, LoRA, and GRPO (Appendix E.3), with one task learned at each stage, to investigate whether CRL-VLA achieves better stability. We compare continual reinforcement learning and continual imitation learning methods. The results show that continual RL baselines achieve higher final average success rates overall, while most exhibit limited forgetting, suggesting their potential to balance new-task acquisition with previous-task retention.

Table 3: Single task under visual disturbance: single-task stages on LIBERO-Spatial. Task settings are given in Appendix E.2. The onlien RL configuration is specified in Table 9.

Scenario	Benchmark	Type	Method	FAR ↑	BWT ↑	FT ↑
Single-task	LIBERO-Spatial	Baseline	SL (Liu et al., 2023)	0.00	-0.62	-0.50
			MTL (Liu et al., 2023)	<u>0.96</u>	-	-
			ER (Rolnick et al., 2019)	0.60	-0.51	**-0.06**
			LWF (Li and Hoiem, 2018)	0.67	<u>-0.50</u>	**-0.06**
		Ours	CRL-VLA	**0.98**	**-0.02**	**-0.06**

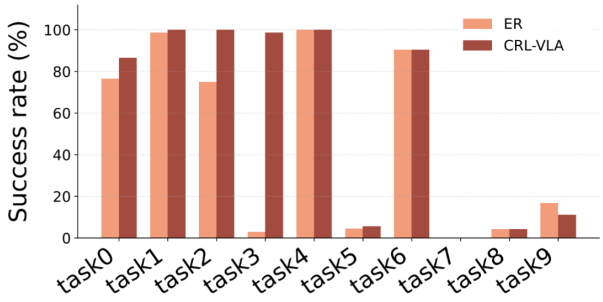

Figure 3: Original ten-task LIBERO-Long comparison under visual disturbance. The translation perturbation is specified in Appendix E.6. This is a single task sequence.

5.3 Robustness to Visual Disturbances

Building on the preceding experiments, we further investigate whether CRL-VLA can maintain task performance and retain previously acquired knowledge under visual disturbances. We compare it with established continual learning baselines across single-task stages and a longer task sequence.

(1) Single-task continual learning. On LIBERO-Spatial, each stage introduces one task, with translation, rotation, and color jitter applied to visual inputs. CRL-VLA achieves a final average success rate of 0.98, compared with 0.67 for LWF and 0.60 for ER. Its BWT of -0.02 also indicates substantially less forgetting than LWF (-0.50) and ER (-0.51). These results show that CRL-VLA maintains high final task performance while mitigating forgetting under the evaluated visual disturbances. **(2) Longer task sequence.** We further compare CRL-VLA with ER on a ten-task sequence under visual disturbances to examine its performance over an extended continual learning process. As shown in Figure 3, CRL-VLA achieves higher overall success in this setting, indicating that it maintains task performance and knowledge retention over the longer perturbed task sequence.

5.4 Applicability and Computational Cost

ManiSkill with a flow-based VLA. Figure 4 reports the original $\pi_{0.5}$ comparison on ManiSkill under sparse rewards, where CRL-VLA shows stronger BWT and FT than the plotted baselines. This extends the empirical record to another model–environment configuration. Since both the backbone and environment change relative to LIBERO, it does not independently identify an architecture effect or establish robustness to reward sparsity.

Resource measurements. On an A100 GPU, CRL-VLA and ER take 3132.9 and 3688.8 seconds per training stage, with peak GPU memory usage of 32.40 and 31.75 GB, respectively. CRL-VLA stores 30 successful trajectories from the preceding task in a 13.58 MB buffer (Appendix E.11).

5.5 Real-World Evaluation across Learning Settings and Embodiments

We evaluate CRL-VLA through three complementary real-world studies: online continual learning with Expo, offline learning over a longer five-task WAV sequence, and a two-task WAV evaluation

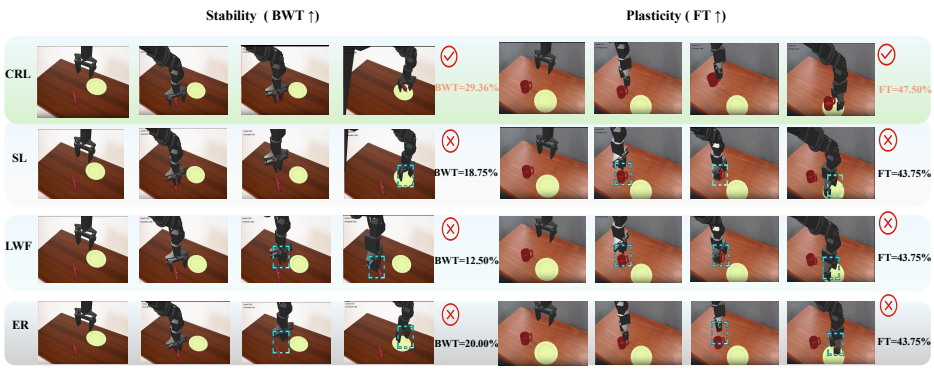

Figure 4: Original continual-learning metrics for $\pi_{0.5}$ on ManiSkill with sparse rewards. The two stages contain multiple object-placement commands, as listed in Appendix E.1.

on Piper. These studies examine the framework across learning regimes, sequence lengths, and robot embodiments. Each uses a backend-specific retention–adaptation implementation, as distinguished in Section 4.3 and detailed in the corresponding appendices.

Online continual reinforcement learning in real world. We evaluate an online off-policy instantiation of CRL-VLA in a three-task setting on XArm using Expo-FT (Dong et al., 2026). The task score is the mean of grasp (progress) and placement (success) rates, with ten trials per observed checkpoint–task pair. The updated CRL-VLA record gives final task scores of 0.25, 0.35, and 0.45, yielding FAR 0.350. Across both earlier tasks, BWT is -0.075 and forgetting is 0.125. The baseline's recorded BWT and forgetting cover only Task 2; on that same task, CRL-VLA gives -0.100 and 0.100, compared with the reported baseline values of -0.200 and 0.200. Full-sequence and single-task retention estimates should not be compared directly. The checkpoint-level data and scoring rule are provided in Appendix E.8.

Table 4: Real-world continual-learning metrics for online Expo on xArm, offline WAV over five tasks on xArm, and WAV on Piper.

Task	Type	Method	FAR ↑	BWT ↑	F ↓
Expo-FT (xArm)	**Baseline**	ER	0.13±0.19	-0.20	0.20
	Ours	CRL-VLA	0.35±0.08	-0.075	0.125
WAV (xArm)	**Baseline**	ER	0.34±0.19	$0.00^{\dagger}$	$0.10^{\dagger}$
	Ours	CRL-VLA	0.58±0.36	0.13	0.13
WAV (Piper)	**Baseline**	ER	0.18	-0.82	0.82
	Ours	CRL-VLA	0.73	-0.27	0.27

Offline continual reinforcement learning over a longer task sequence. We next extend the evaluation to offline learning and a five-task real-world sequence using WAV (Li et al., 2026c). Both methods are evaluated on all tasks seen so far after each stage, with ten trials per checkpoint–task pair (Appendix E.9). At the fifth checkpoint, CRL-VLA achieves 58% mean final task success versus 34% for the baseline, and 82% versus 60% mean grasp success (Table 4). The task sequence and complete checkpoint matrices are provided in Fig. 8 and Appendix E.9. The overall gain coexists with local forgetting: Task-1 success falls from 50% after stage four to 0% at the final checkpoint. Final-success forgetting is 12.5 percentage points, compared with $10.0^{\dagger}$ for the baseline. Thus, this five-task offline evaluation shows improved overall final performance without eliminating individual-task failure. These are single-run estimates; the different tasks and learning regime also preclude attributing differences from Expo solely to sequence length.

Cross-embodiment applicability and retention on Piper. Finally, we examine applicability to another robot embodiment using a two-task WAV-based model-based RL and VLA setup on Piper (Li et al., 2026c). The sequence comprises bowl organization followed by cube picking (Ap-

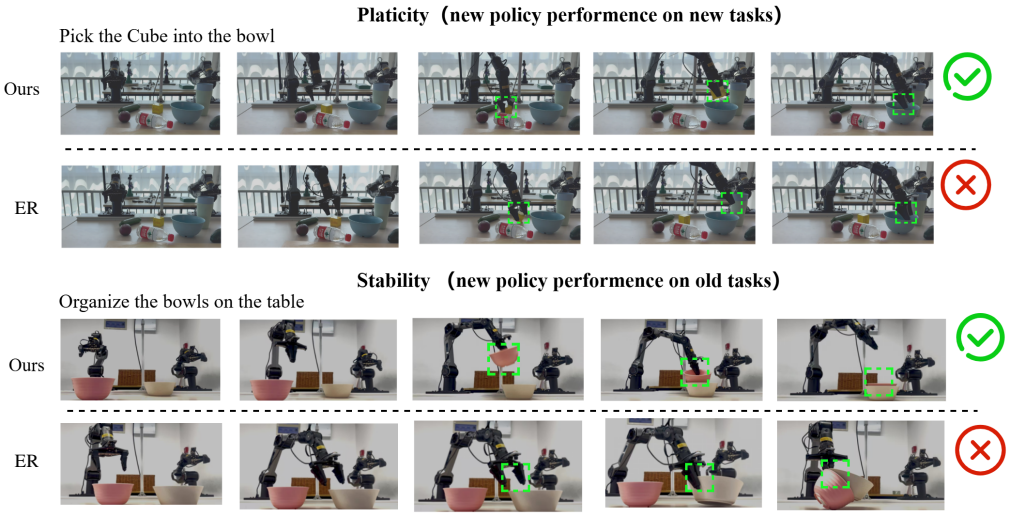

Figure 5: Representative final-checkpoint Piper rollouts (WAV-based): newly learned cube picking (top two rows) and retained bowl organization (bottom two rows). See Table 4 for aggregate results.

pendix E.10). At the final checkpoint, CRL-VLA improves FAR from 0.1818 to 0.7273 and BWT from -0.8182 to -0.2727, while reducing forgetting from 0.8182 to 0.2727 (Table 4). Figure 5 illustrates executions on the newly learned and earlier tasks. This comparison extends the framework evaluation to Piper and supports better retention than ER in this setting. Cross-embodiment applicability here means training and evaluating the framework on different robots; it does not imply zero-shot transfer of an unchanged policy.

5.6 Ablations and Mechanism Analysis

Figure 6 reports retention and adaptation coefficient sweeps for the PPO configuration on Task Suite-4. The highest recorded FAR is 0.98 at retention weight 0.01 and 0.99 at adaptation weight $\eta = 1$. Tables 5 and 6 in appendix provide the full values, including a single policy-KL coefficient sweep.

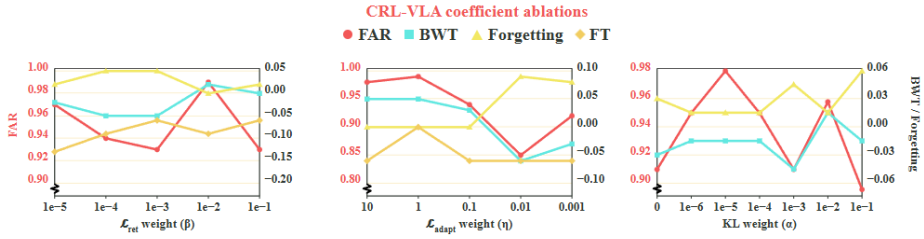

Figure 6: CRL-VLA ablation study: retention loss (left) adaptation loss (middle), and KL loss(right). Tabulated values appear in Appendix D.

6 Conclusion and Future Work

In this work, we study continual learning for VLAs and analyze how goal-conditioned advantage magnitude and policy divergence bound performance changes. The **CRL-VLA** framework combines value retention and adaptation with backend-specific policy regularization. Experiments in simulation and on real robots demonstrate strong performance and robustness across diverse VLA RL post-training settings and robot embodiments.

REFERENCES

Istabrak Abbes, Gopeshh Subbaraj, Matthew Riemer, Nizar Islah, Tsuguchika Tabaru, Hiroaki Kingetsu, Sarath Chandar, and Irina Rish. Revisiting Replay and Gradient Alignment for Continual Pre-Training of Large Language Models. In *Conference on Lifelong Learning Agents*, volume 330 of *Proceedings of Machine Learning Research*, pages 465–486. PMLR, 2026. URL https://proceedings.mlr.press/v330/abbes26a.html.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained Policy Optimization. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR, 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40. PMLR, 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning: A Strong, Simple Baseline. In *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html.

Perry Dong, Kuo-Han Hung, Tian Gao, Dorsa Sadigh, and Chelsea Finn. EXPO-FT: Sample-Efficient Reinforcement Learning Finetuning for Vision-Language-Action Models, 2026. URL https://arxiv.org/abs/2605.25477. arXiv preprint; no journal volume or page range assigned to this version.

Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving Vision-Language-Action Model with Online Reinforcement Learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15665–15672. IEEE, 2025. doi: 10.1109/icra55743.2025.11127299. URL https://doi.org/10.1109/icra55743.2025.11127299.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as Language: Fine-Tuning VLMs into VLAs Without Catastrophic Forgetting. In *International Conference on Learning Representations*, volume 2026, pages 75333–75360, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7a0f8055c838df8e62329a76c7c6403d-Paper-Conference.pdf.

Jiaheng Hu, Jay Shim, Chen Tang, Yoonchang Sung, Bo Liu, Peter Stone, and Roberto Martín-Martín. Simple Recipe Works: Vision-Language-Action Models are Natural Continual Learners with Reinforcement Learning, 2026. URL https://arxiv.org/abs/2603.11653. arXiv preprint; no journal volume or page range assigned to this version.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002. URL https://www.cs.utexas.edu/~shivaram/readings/b2hd-KakadeLangford2002.html.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards Continual Reinforcement Learning: A Review and Perspectives. *Journal of Artificial Intelligence Research*, 75: 1401–1476, 2022. doi: 10.1613/jair.1.13673. URL https://doi.org/10.1613/jair.1.13673.

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. arXiv preprint; no journal volume or page range assigned to this version.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://doi.org/10.1073/pnas.1611835114.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1083–1091, Abu Dhabi, United Arab Emirates, dec 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.77. URL https://aclanthology.org/2022.findings-emnlp.77/.

Alexey Kutalev and Alisa Lapina. Stabilizing Elastic Weight Consolidation method in practical ML tasks and using weight importances for neural network pruning, 2021. URL https://arxiv.org/abs/2109.10021. arXiv preprint; no journal volume or page range assigned to this version.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, Dehui Wang, Dingxiang Luo, Yuchen Fan, Youbang Sun, Jia Zeng, Jiangmiao Pang, Shanghang Zhang, Yu Wang, Yao Mu, Bowen Zhou, and Ning Ding. SimpleVLA-RL: Scaling VLA Training via Reinforcement Learning. In *International Conference on Learning Representations*, volume 2026, pages 125846–125869, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/cbfbcb4da14235bd69b134070898ae9d-Paper-Conference.pdf.

Runze Li, Hongyin Zhang, Junxi Jin, Qixin Zeng, Zifeng Zhuang, Yiqi Tang, Shangke Lyu, and Donglin Wang. World-value-action model: Implicit planning for vision-language-action systems, 2026b. URL https://arxiv.org/abs/2604.14732.

Runze Li, Hongyin Zhang, Junxi Jin, Qixin Zeng, Zifeng Zhuang, Yiqi Tang, Shangke Lyu, and Donglin Wang. World-Value-Action Model: Implicit Planning for Vision-Language-Action Systems, 2026c. URL https://arxiv.org/abs/2604.14732. arXiv preprint; no journal volume or page range assigned to this version.

Zhizhong Li and Derek Hoiem. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. doi: 10.1109/tpami.2017.2773081. URL https://doi.org/10.1109/tpami.2017.2773081.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 44776–44791. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-1939. URL https://doi.org/10.52202/075280-1939.

Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. ManiSkill: Generalizable Manipulation Skill Benchmark with Large-Scale Demonstrations. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/eda80a3d5b344bc40f3bc04f65b7a357-Paper-round2.pdf. Page range not supplied by the consulted proceedings record.

Chaofan Pan, Xin Yang, Yanhua Li, Wei Wei, Tianrui Li, Bo An, and Jiye Liang. A Survey of Continual Reinforcement Learning, 2025. URL https://arxiv.org/abs/2506.21872. arXiv preprint; no journal volume or page range assigned to this version.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a VLA That Learns From Experience, 2025. URL https://arxiv.org/abs/2511.14759. arXiv preprint; no journal volume or page range assigned to this version.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Paper.pdf. Page range not supplied by the consulted proceedings record.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy Distillation. In *International Conference on Learning Representations*, 2016a. URL https://arxiv.org/abs/1511.06295. Page range not supplied by the consulted proceedings record.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks, 2016b. URL https://arxiv.org/abs/1606.04671. arXiv preprint; no journal volume or page range assigned to this version.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL https://arxiv.org/abs/1707.06347. arXiv preprint; no journal volume or page range assigned to this version.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & Compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4528–4537. PMLR, 2018. URL https://proceedings.mlr.press/v80/schwarz18a.html.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's Razor: Why Online Reinforcement Learning Forgets Less. In *International Conference on Learning Representations*, volume 2026, pages 59839–59864, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/618c95f4557c15b253fb0e6f548ea0c0-Paper-Conference.pdf.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive Post-Training for Vision-Language-Action Models, 2025. URL https://arxiv.org/abs/2505.17016. arXiv preprint; no journal volume or page range assigned to this version.

Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual World: A Robotic Benchmark For Continual Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 28496–28510. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ef8446f35513a8d6aa2308357a268a7e-Paper.pdf.

Yuxuan Wu, Guangming Wang, Zhiheng Yang, Tianchen Deng, Maoqing Yao, Brian Sheil, and Hesheng Wang. Continually Evolving Skill Knowledge in Vision Language Action Model, 2025.

URL https://arxiv.org/abs/2511.18085. arXiv preprint; no journal volume or page range assigned to this version.

Ju Xu and Zhanxing Zhu. Reinforced Continual Learning. In *Advances in Neural Information Processing Systems*, volume 31, pages 907–916. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/cee631121c2ec9232f3a2f028ad5c89b-Paper.pdf.

Yajat Yadav, Zhiyuan Zhou, Andrew Wagenmaker, Karl Pertsch, and Sergey Levine. Robust Fine-tuning of Vision-Language-Action Robot Policies via Parameter Merging. In *International Conference on Learning Representations*, volume 2026, pages 56385–56417, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5c7465e4a236b56226ca221f3d16bf2b-Paper-Conference.pdf.

Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Peihong Wang, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, Zhihao Liu, Kang Chen, Wenhao Tang, Quanlu Zhang, Weinan Zhang, Chao Yu, and Yu Wang. RLinf-VLA: A Unified and Efficient Framework for Reinforcement Learning of Vision-Language-Action Models, 2025. URL https://arxiv.org/abs/2510.06710. arXiv preprint; no journal volume or page range assigned to this version.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR, 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow Learner with Classifier Alignment for Continual Learning on a Pre-trained Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19148–19158, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_SLCA_Slow_Learner_with_Classifier_Alignment_for_Continual_Learning_on_ICCV_2023_paper.html.

Hongyin Zhang, Shiyuan Zhang, Junxi Jin, Qixin Zeng, Yifan Qiao, Hongchao Lu, and Donglin Wang. Balancing Signal and Variance: Adaptive Offline RL Post-Training for VLA Flow Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18755–18763. Association for the Advancement of Artificial Intelligence (AAAI), 2026. doi: 10.1609/aaai.v40i22.38944. URL https://doi.org/10.1609/aaai.v40i22.38944.

Libang Zhao, Qixin Zeng, Hongyin Zhang, and Donglin Wang. Information-Theoretic Constraints for Continual Vision-Language-Action Alignment, 2026. URL https://arxiv.org/abs/2603.13335. arXiv preprint; no journal volume or page range assigned to this version.

A Appendix

B Theoretical Analysis and Proofs

B.1 Preliminaries and Definitions

Definition 1 (Mixture Occupancy Measures)**.** Given a goal distribution $p(g)$, we define the mixture occupancy measure as:

$$d_p^\pi(B) \triangleq \mathbb{E}_{g\sim p}[d_g^\pi(B)],$$

where $B \subseteq \mathcal{S}$ is measurable and d_g^π is the discounted state occupancy measure defined in the preliminaries. The corresponding state-action occupancy is $\rho_g^\pi(ds, da) = d_g^\pi(ds)\pi(da \mid s, g)$.

Definition 2 (Advantage Magnitude)**.** Fix the reference policy π^{old}. As in Section 4.1,

$$M_g \triangleq \sup_{s\in\mathcal{S},\, a\in\mathcal{A}} |A_g^{\pi^{\text{old}}}(s, a)|. \tag{7}$$

The aggregate magnitudes are

$$M_{\text{old}} = \sqrt{\mathbb{E}_{g\sim p_{\text{old}}} M_g^2}, \qquad M_{\text{new}} = \sqrt{\mathbb{E}_{g\sim p_{\text{new}}} M_g^2}.$$

B.2 Proof of Theorem 1

Under the assumptions of Theorem 1, we use the goal-conditioned Performance Difference Lemma (PDL) (Kakade and Langford, 2002). The policies are compared within the same goal-conditioned MDP and initial-state distribution. Sums below denote integrals in continuous spaces.

Lemma 1 (Goal-Conditioned PDL)**.** *For any policies $\pi^{\text{new}}, \pi^{\text{old}}$ and goal g, the performance difference satisfies:*

$$J_g(\pi^{\text{new}}) - J_g(\pi^{\text{old}}) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_g^{\pi^{\text{new}}}}\left[\mathbb{E}_{a\sim\pi^{\text{new}}(\cdot|s,g)}[A_g^{\pi^{\text{old}}}(s, a)]\right].$$

B.2.1 Proof of Stability Bound

We aim to bound the performance degradation on old tasks: $|J_{\text{old}}(\pi^{\text{new}}) - J_{\text{old}}(\pi^{\text{old}})|$. Fix a goal $g \sim p_{\text{old}}$. Let $f_g(s) \triangleq \mathbb{E}_{a\sim\pi^{\text{new}}(\cdot|s,g)}[A_g^{\pi^{\text{old}}}(s, a)]$. Using Lemma 1, we decompose the expectation over $d_g^{\pi^{\text{new}}}$:

$$J_g(\pi^{\text{new}}) - J_g(\pi^{\text{old}}) = \frac{1}{1-\gamma}\left(\underbrace{\mathbb{E}_{s\sim d_g^{\pi^{\text{old}}}}[f_g(s)]}_{\text{(I) Action Mismatch}} + \underbrace{\int f_g(s)\big(d_g^{\pi^{\text{new}}} - d_g^{\pi^{\text{old}}}\big)(ds)}_{\text{(II) Occupancy Mismatch}}\right).$$

Bounding the two terms. The global definition gives $\sup_s |f_g(s)| \le M_g$. Define

$$t_g = \mathbb{E}_{s\sim d_g^{\pi^{\text{old}}}} D_{\text{TV}}\big(\pi^{\text{new}}(\cdot \mid s, g), \pi^{\text{old}}(\cdot \mid s, g)\big).$$

Since $\mathbb{E}_{a\sim\pi^{\text{old}}} A_g^{\pi^{\text{old}}}(s, a) = 0$, the action-mismatch term satisfies

$$|(\text{I})| = \left|\mathbb{E}_{s\sim d_g^{\pi^{\text{old}}}} \int (\pi^{\text{new}} - \pi^{\text{old}})(da \mid s, g) A_g^{\pi^{\text{old}}}(s, a)\right| \le 2M_g t_g.$$

The discounted occupancy bound (Achiam et al., 2017) yields

$$2D_{\text{TV}}\left(d_g^{\pi^{\text{new}}}, d_g^{\pi^{\text{old}}}\right) \le \frac{2\gamma}{1-\gamma} t_g, \qquad |(\text{II})| \le \frac{2\gamma}{1-\gamma} M_g t_g.$$

Combining both terms gives

$$|J_g(\pi^{\text{new}}) - J_g(\pi^{\text{old}})| \le \frac{2M_g t_g}{(1-\gamma)^2}.$$

Let K_g^{old} denote the expected KL divergence in the definition of D_{old}, conditional on g. Pinsker's and Jensen's inequalities give $t_g \leq \sqrt{K_g^{\text{old}}/2}$. Thus

$$|J_g(\pi^{\text{new}}) - J_g(\pi^{\text{old}})| \leq \frac{M_g\sqrt{2K_g^{\text{old}}}}{(1-\gamma)^2}.$$

Finally, the triangle inequality and Cauchy–Schwarz over goals give

$$\begin{aligned}
|J_{\text{old}}(\pi^{\text{new}}) - J_{\text{old}}(\pi^{\text{old}})| &\leq \frac{\mathbb{E}_{g\sim p_{\text{old}}}[M_g\sqrt{2K_g^{\text{old}}}]}{(1-\gamma)^2} \\
&\leq \frac{\sqrt{\mathbb{E}_{g\sim p_{\text{old}}} M_g^2}\,\sqrt{2\mathbb{E}_{g\sim p_{\text{old}}} K_g^{\text{old}}}}{(1-\gamma)^2} \\
&= \frac{M_{\text{old}} D_{\text{old}}}{(1-\gamma)^2}.
\end{aligned}$$

B.2.2 Proof of Plasticity Bound

For new tasks, we consider the performance gain: $J_{\text{new}}(\pi^{\text{new}}) - J_{\text{new}}(\pi^{\text{old}})$. Using Lemma 1 with π^{old} as the anchor, for a goal $g \sim p_{\text{new}}$:

$$J_g(\pi^{\text{new}}) - J_g(\pi^{\text{old}}) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_g^{\pi^{\text{new}}}}\left[\mathbb{E}_{a\sim\pi^{\text{new}}}[A_g^{\pi^{\text{old}}}(s,a)]\right].$$

Since $V^{\pi^{\text{old}}}(s,g) = \mathbb{E}_{a\sim\pi^{\text{old}}(\cdot|s,g)}[Q^{\pi^{\text{old}}}(s,a,g)]$, $\mathbb{E}_{a\sim\pi^{\text{old}}(\cdot|s,g)}[A_g^{\pi^{\text{old}}}(s,a)] = 0$. We can therefore rewrite the inner expectation:

$$\mathbb{E}_{a\sim\pi^{\text{new}}}[A_g^{\pi^{\text{old}}}] = \int (\pi^{\text{new}} - \pi^{\text{old}})(da \mid s,g) A_g^{\pi^{\text{old}}}(s,a).$$

By Hölder's inequality:

$$\left|\mathbb{E}_{a\sim\pi^{\text{new}}}[A_g^{\pi^{\text{old}}}]\right| \leq 2D_{\text{TV}}(\pi^{\text{new}}\|\pi^{\text{old}}) \cdot \sup_a \left|A_g^{\pi^{\text{old}}}(s,a)\right|.$$

The supremum over all actions is bounded by M_g. Applying Pinsker's inequality and substituting into the PDL gives

$$J_g(\pi^{\text{new}}) - J_g(\pi^{\text{old}}) \leq \frac{M_g}{1-\gamma}\mathbb{E}_{s\sim d_g^{\pi^{\text{new}}}}\left[\sqrt{2D_{\text{KL}}(\pi^{\text{new}}(\cdot \mid s,g)\|\pi^{\text{old}}(\cdot \mid s,g))}\right].$$

Let K_g^{new} be the expected KL conditional on g under $d_g^{\pi^{\text{new}}}$. Jensen's inequality over states followed by Cauchy–Schwarz over goals yields

$$\begin{aligned}
J_{\text{new}}(\pi^{\text{new}}) - J_{\text{new}}(\pi^{\text{old}}) &\leq \frac{\mathbb{E}_{g\sim p_{\text{new}}}[M_g\sqrt{2K_g^{\text{new}}}]}{1-\gamma} \\
&\leq \frac{\sqrt{\mathbb{E}_{g\sim p_{\text{new}}} M_g^2}\,\sqrt{2\mathbb{E}_{g\sim p_{\text{new}}} K_g^{\text{new}}}}{1-\gamma} \\
&= \frac{M_{\text{new}} D_{\text{new}}}{1-\gamma}.
\end{aligned}$$

□

The different horizon factors in Theorem 1 arise from measuring divergence under the reference-policy and updated-policy occupancies, respectively. The new-task upper bound does not imply a positive gain. By Pinsker's inequality, the same bounds hold with reversed KL under the same state distributions whenever the corresponding expectations are finite.

C Implementation Details and Backend-Specific Instantiations

C.1 Overview

Table 1 summarizes the mechanisms in the main text. We first provide the action-value targets and training loop, then specify each backend's estimators, memory, and policy updates.

C.2 Actor Objectives and Training Procedure

Online policy optimization. For trajectories collected by a stochastic rollout policy π_b, use the return target $y_t^{\text{return}} = \sum_{j=t}^{T-1} \gamma^{j-t} r_j$ for $\mathcal{L}_{\text{adapt}}$. The clipped actor objective (Schulman et al., 2017) is

$$\mathcal{L}_\pi^{\text{on}} = -\mathbb{E}_{\mathcal{B}_{\text{new}}} \min\Big\{\rho_t \widehat{A}_t,\ \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\widehat{A}_t\Big\}, \quad \rho_t = \frac{\pi_\theta(a_t \mid s_t, g)}{\pi_b(a_t \mid s_t, g)}. \tag{8}$$

Here $\widehat{A}_t$ estimates the rollout policy's advantage. For a fixed adaptation-critic snapshot $\widetilde{Q}$, a critic-based estimate uses $\text{sg}[\widetilde{Q}(s_t, a_t, g) - \mathbb{E}_{a' \sim \pi_b} \widetilde{Q}(s_t, a', g)]$; the RIPT-VLA backend uses the rollout-return estimator detailed in Appendix C.3. In this backend, PPO clipping and KL monitoring control current-task updates, so $\alpha_{\text{new}} = 0$ in Eq. 6. This proximal control uses the batch rollout policy π_b, whereas historical regularization uses the stage-fixed π_k^-. Clipping does not implement the explicit stage-reference KL penalty.

Offline and online off-policy learning. For a transition (s, a, r, s', d, g), a representative action-value interface is

$$\begin{aligned} y^{\text{TD}} &= r + \gamma(1-d)\, \mathbb{E}_{a' \sim \pi_b(\cdot|s',g)} Q^{\text{tgt}}(s', a', g), \\ \mathcal{L}_\pi^{\text{off}} &= -\mathbb{E}_{\substack{(s,g)\sim\mu_{\text{new},k} \\ a\sim\pi_\theta(\cdot|s,g)}} \widetilde{Q}(s, a, g). \end{aligned} \tag{9}$$

Here π_b and $Q^{\text{tgt}} = \widetilde{Q}$ are update-entry policy and adaptation-critic snapshots; the target evaluator is distinct from the historical teacher. Actor gradients pass through the action input while the evaluator parameters and state features are fixed. For action chunks, rewards and discounting use the executed chunk duration. This action-value interface does not directly describe WAV flow supervision or Expo-FT candidate selection; their actual objectives are specified in Appendices C.4 and C.5.

Stage transitions and memory. Algorithm 1 specifies the action-value training loop. The stage-reference policy and complete retention teacher stay fixed throughout a stage, while rollout policies and adaptation evaluators may be refreshed between updates. Actor optimization holds evaluator parameters and state features fixed, retaining action-input gradients where applicable. All regression targets are stopped.

A capacity-B reservoir for the next stage is initialized from $\mathcal{M}_{k-1}$. Each previously unseen transition or distinct offline record is ingested once, using a cumulative count across stages; repeated minibatch samples are not new records. Historical training samples continue to use the unchanged $\mathcal{M}_{k-1}$ until the new reservoir is committed at stage end. Reference Q labels are recomputed or re-cached from the new teacher at the next stage. Backend-specific memory and policy-regularization variants are stated in their respective subsections.

C.3 On-Policy Instantiation

The RIPT-VLA-based instantiation uses the OpenVLA-OFT policy interface (Tan et al., 2025). Its critic architecture and initialization follow Section 4.2; Monte Carlo targets use the return in Appendix C.2. The two value losses are given in Eq. 4.

Rollout-based advantage estimation. For $m > 1$ independent rollouts from the same initial observation–goal context c under π_b, with returns $R_1, \ldots, R_m$, the leave-one-out estimate is

$$\widehat{A}_i^{\text{RLOO}} = R_i - \frac{1}{m-1} \sum_{j \neq i} R_j. \tag{10}$$

Algorithm 1 CRL-VLA: Continual Vision–Language–Action Learning

Require: Pretrained VLA π^0; tasks $\{\mathcal{T}_k\}_{k=1}^K$; backend b; memory capacity B.
Initialize adaptation head ψ, memory $\mathcal{M}_0 = \emptyset$, and reservoir count $n = 0$.
for $k = 1, \ldots, K$ **do**
  Set the stage reference and initialize critics as in Sections 4.2–4.3.
  Initialize next-stage reservoir $\mathcal{M}_{\text{next}} \leftarrow \mathcal{M}_{k-1}$.
  for each training iteration on $\mathcal{T}_k$ **do**
    Obtain current-task data by rollout or sampling its offline/replay dataset $\mathcal{M}_{k-1}$.
    Ingest each previously unseen data record once into $\mathcal{M}_{\text{next}}$; update n.
    Draw $\mathcal{B}_{\text{new}}$ by rollout; draw historical $\mathcal{B}_{\text{old}}$ by $\mathcal{M}_{k-1}$ only if $k > 1$.
    Compute backend targets $y^{(b)}$ and actor estimates.
    Compute $\mathcal{L}_{\text{adapt}}$; if $k > 1$, evaluate the teacher and compute $\mathcal{L}_{\text{ret}}$.
    Compute the actor objective and active policy regularizers.
    Update (ϕ, ω, ψ) using Eq. 6.
  end for
  Save π^k and Q_k^{adapt}; commit $\mathcal{M}_k \leftarrow \mathcal{M}_{\text{next}}$.
end for
return Policies $\{\pi^k\}_{k=1}^K$ and their critic snapshots.

RLOO estimates advantage using sampled returns and a baseline formed from the other rollouts (Ahmadian et al., 2024; Tan et al., 2025). In a rollout-level formulation, let u_i denote the sampled outcome, $Q(c, u) = \mathbb{E}[R \mid c, u]$, and $V(c) = \mathbb{E}[R \mid c]$. Independence gives $\mathbb{E}[\widehat{A}_i^{\text{RLOO}} \mid c, u_i] = Q(c, u_i) - V(c)$. Thus, RLOO supplies a Monte Carlo estimate of the rollout-level $Q - V$ difference without using a learned critic as its baseline. This identity applies to independent groups before any outcome-based filtering.

Policy objective. The RIPT actor uses the clipped surrogate in Eq. 8, with sample index i, rollout-level likelihood ratios, and the group advantages above. Historical policy regularization is evaluated on the retained previous-task buffer $\mathcal{B}_{k-1}$:

$$\mathcal{R}_{\text{prev}}^{\text{RIPT}} = \mathbb{E}_{(s,g)\sim\mathcal{B}_{k-1}} \overline{D}_{\text{KL}}\big(\pi_k^-(\cdot \mid s, g) \,\|\, \pi_\theta(\cdot \mid s, g)\big) . \tag{11}$$

Here $\overline{D}_{\text{KL}}$ sums analytic Laplace KL over action dimensions and averages over chunk positions. This backend uses the reverse direction of Eq. 5. Writing $\mathcal{L}_\pi^{\text{RIPT}}$ for its proximal actor loss, the specialization is

$$\mathcal{L}_{\text{RIPT}} = \mathcal{L}_\pi^{\text{RIPT}} + \eta\mathcal{L}_{\text{adapt}} + \beta\mathcal{L}_{\text{ret}} + \alpha\mathcal{R}_{\text{prev}}^{\text{RIPT}}. \tag{12}$$

Historical expectations in this instantiation use $\mathcal{B}_{k-1}$, which is replaced by retained task-k records after each stage. This specifies the previous-task memory variant of the training loop.

Policy-penalty estimation. With common support, the KL in Eq. 5 can be computed analytically or estimated with actions sampled from the current policy; averaging log-ratios over stored historical actions estimates a different expectation. Under the theorem's exact occupancy distributions, the population penalties equal $D_{\text{old}}^2/2$ and $D_{\text{new}}^2/2$; empirical replay and rollout distributions need not coincide with them.

C.4 Offline Instantiation

Value and action branches. The offline instantiation builds on WAV, which couples a video-language backbone with action and value experts. At each task transition, the trainable model is initialized from the preceding checkpoint, and a separate historical value expert is copied from that checkpoint and frozen. Both value experts attend to the current backbone features, while the action expert also attends to the trainable value expert's features. The value branches predict return-related quantities from video-language features, a value-noise input, and a flow timestep.

Current and historical supervision. Each training iteration combines current-task data $\mathcal{D}_k$ with a historical partition $\mathcal{D}_{\text{hist}}$. In the five-task offline sequence, this historical partition contains data from all preceding tasks. Action learning uses flow-matching supervision on both partitions. Let $\mathcal{L}_{\text{act,new}}$ and $\mathcal{L}_{\text{act,hist}}$ denote the corresponding action-velocity regression losses.

Value supervision uses the return labels provided with the data. Let ℓ_{val} denote the weighted value-regression loss in flow coordinates, including the return-label transformation and noise conditioning. With v_ψ the trainable value expert and $v_{\bar{\psi}_k}$ the frozen historical expert, the value objectives are

$$\begin{aligned}\mathcal{L}_{\text{val,new}} &= \mathbb{E}_{(s,g,G)\sim\mathcal{D}_k}\ell_{\text{val}}(v_\psi, z_\phi(s,g), G),\\ \mathcal{L}_{\text{val,hist}} &= \mathbb{E}_{(s,g,G)\sim\mathcal{D}_{\text{hist}}}\ell_{\text{val}}(v_{\bar{\psi}_k}, z_\phi(s,g), G).\end{aligned} \tag{13}$$

Historical supervision preserves compatibility between the evolving representation and the fixed value expert using the historical return labels. Current-task supervision trains the adaptation expert and the shared representation on new data.

Joint optimization. The offline objective combines the four terms:

$$\mathcal{L}_{\text{WAV}} = \mathcal{L}_{\text{act,new}} + \mathcal{L}_{\text{act,hist}} + c_{\text{ret}}\mathcal{L}_{\text{val,hist}} + c_{\text{adapt}}\mathcal{L}_{\text{val,new}}. \tag{14}$$

The value coefficients are adaptively scaled relative to the behavior-cloning (BC) loss and treated as constants during backpropagation. The historical value loss updates the shared backbone through the frozen expert, while the current value loss updates the trainable expert and backbone. The action losses update the action pathway and its attached conditioning paths. At the first stage, training uses only the current-task action and value objectives.

C.5 Online Off-Policy Instantiation

Critic branches and stage transition. The online instantiation builds on Expo-FT. A language-conditioned VLA and an edit policy generate candidate action chunks, which are evaluated by action-value ensembles using images and robot state. At a task transition, the adaptation encoder, state projection, and target Q ensemble are copied into a frozen retention branch. The adaptation branch is initialized from the same snapshot and continues learning from current-stage interaction. The two branches encode observations independently.

Retention through action selection. Let $\mathcal{C}(s,g)$ be the candidate action set, with a denoting the portion of a chunk used for execution and value evaluation. For a shared sampled subset $\mathcal{I}$ of ensemble heads, define

$$\begin{aligned}q_k^{\text{ret}}(s,a) &= \min_{j\in\mathcal{I}} Q_{k,j}^{\text{ret}}(s,a),\\ q_k^{\text{adapt}}(s,a) &= \min_{j\in\mathcal{I}} Q_{k,j}^{\text{adapt,tgt}}(s,a).\end{aligned} \tag{15}$$

The executed candidate is selected by

$$a_k^\star(s,g) = \arg\max_{a\in\mathcal{C}(s,g)} \left[q_k^{\text{ret}}(s,a) + q_k^{\text{adapt}}(s,a)\right]. \tag{16}$$

The retention branch contributes a fixed historical evaluation, while the adaptation branch supplies current-task value information. At the first stage, candidate selection uses only the adaptation branch. A single candidate is executed directly without re-ranking.

Value and policy updates. The combined selection rule also chooses the next action for the TD update. The numerical target uses the adaptation target ensemble:

$$y = r_{\text{chunk}} + \gamma^H m_{\text{nt}} \min_{j\in\mathcal{J}} Q_{k,j}^{\text{adapt,tgt}}\left(s', a_k^\star(s',g)\right), \tag{17}$$

where r_{chunk} is the discounted reward accumulated over the executed chunk, H is its executed horizon, m_{nt} is the nonterminal mask, and $\mathcal{J}$ is the head subset used by the TD target evaluator. The online adaptation ensemble minimizes squared TD error over valid replay windows, and its Q target parameters are updated by an exponential moving average.

The edit policy is optimized using the mean online adaptation Q value together with entropy regularization. The VLA is updated by behavior cloning on successful current-task trajectories; its update is skipped when no eligible success batch is available. This backend retains historical information through candidate selection, without a historical replay loss or an old-task behavior-cloning term.

D Additional Validation and Ablations

Figure 6 visualizes retention and adaptation loss coefficient sweeps for the PPO configuration in Table 9. The tables below report these values and the single policy-KL coefficient sweep.

Table 5: Comparison of continual learning metrics under different retention loss and policy KL coefficients, where all experiments are conducted on the Task-4 benchmark.

Parameter Value	FAR (↑)	BWT (↑)	Forgetting (↓)	FT (↑)
Retention				
0.00001	0.97	-0.02	0.02	-0.13
0.0001	0.94	-0.05	0.05	-0.09
0.001	0.93	-0.05	0.05	**-0.06**
0.01	**0.98**	**0.02**	**0.00**	-0.09
0.1	0.93	0.00	0.02	**-0.06**
KL				
0	0.91	-0.03	0.03	**-0.03**
0.000001	0.95	-0.02	**0.02**	-0.09
0.00001	**0.98**	-0.02	**0.02**	-0.06
0.0001	0.95	-0.02	**0.02**	-0.11
0.001	0.91	-0.05	0.05	**-0.03**
0.01	0.96	**0.02**	**0.02**	-0.06
0.1	0.90	-0.02	0.06	-0.09

Table 6: Effect of the adaptation loss coefficient on performance and stability. All experiments are conducted on the Task-4 benchmark.

Parameter Coef	FAR (↑)	BWT (↑)	Forgetting (↓)	FT (↑)
Adaptation				
10	0.98	**0.05**	**0.00**	-0.06
1	**0.99**	**0.05**	**0.00**	**0.00**
0.1	0.94	0.03	**0.00**	-0.06
0.01	0.85	-0.06	0.09	-0.06
0.001	0.92	-0.03	0.08	-0.06

E TASK AND EXPERIMENT DETAILS

To examine performance across task groupings of different sizes, we construct multiple benchmarks by randomly sampling tasks from a shared task pool. Each benchmark contains a different number of object manipulation tasks, serving as fixed comparison settings with varying task complexity and combinatorial difficulty. These settings provide descriptive evidence about behavior across task compositions, rather than an isolated test of robustness or scalability.

Specifically, we evaluate our method on a subset of the **LIBERO** benchmark suite, consisting of tasks randomly sampled from the original collection to encompass various task cardinalities. We denote these benchmarks as **Task-1** through **Task-4**. Each benchmark comprises multiple task sets defined by natural language directives, primarily focusing on robotic pick-and-place behaviors. This setup allows us to compare instruction-conditioned continual-learning behavior across several fixed task compositions.

We next detail the language instructions used for each benchmark. Each task consists of one or more natural language directives.

E.1 TASK SUITE-0 MANISKILL BENCHMARK

- *Task 0*: "put carrot on plate; put kitchen shovel on plate; put bread on plate; put plastic bottle on plate; put up can on plate; put zucchini on plate; put ketchup bottle on plate; put watering can on plate; put pipe on plate; put toy bear on plate; put fast food cup on plate; put plant on plate; put banana on plate; put hamburger on plate; put golf ball on plate; put BBQ sauce on plate."
- *Task 1*: "put travel cup on plate; put pepper on plate; put nonstop can on plate; put potato on plate; put baguette on plate; put champagne glass on plate; put kitchen spoon on plate; put onion on plate; put cup on plate."

E.2 TASK SUITE-1 LIBERO-SPATIAL BENCHMARK

- *Task 0*: "pick up the black bowl from table center and place it on the plate"
- *Task 1*: "pick up the black bowl next to the ramekin and place it on the plate"
- *Task 2*: "pick up the black bowl on the cookie box and place it on the plate"

E.3 LIBERO-SPATIAL PROTOCOL

To enable a controlled comparison with Hu et al. (2026), we specify a separate matched LIBERO-Spatial experiment. This protocol is not pooled with our three-task perturbed LIBERO-Spatial experiment. It uses an OpenVLA-OFT base model initialized by supervised fine-tuning on 10 in-domain demonstrations, for which the source protocol reports an initial average training-task success of 56.9%. The sequence contains five tasks, with 10,240 online episodes per task and a maximum episode length of 512. The published baseline results in Table 2 use three independent seeds and report mean $\pm$ standard error.

Training tasks and order. The five language-conditioned tasks are evaluated in the following fixed order:

1. "pick up the black bowl between the plate and the ramekin and place it on the plate";
2. "pick up the black bowl next to the ramekin and place it on the plate";
3. "pick up the black bowl from table center and place it on the plate";
4. "pick up the black bowl on the cookie box and place it on the plate";
5. "pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate."

Imitation-based continual VLA runs. We additionally evaluate Stellar-VLA (Wu et al., 2025) and Info-VLA (Zhao et al., 2026) using the same five training-task order. Table 2 reports the aggregate metrics. Tasks 0–4 are the training tasks listed above.

Table 7: Shared hyperparameters for the matched LIBERO-Spatial protocol, reproduced from Hu et al. (2026). These settings apply to the source-reported baseline rows in Table 2.

Parameter	Value	Parameter	Value
Algorithm	GRPO	Optimizer	AdamW
Learning rate	2×10^{-5}	Global batch size	8192
AdamW β_1 / β_2	0.9 / 0.999	AdamW ϵ	10^{-5}
Gradient clip norm	1.0	Discount γ	0.99
GAE λ	0.95	Clip ratio (low/high)	0.20 / 0.28
KL coefficient β	0.0	Entropy bonus	0.0
Rollout epochs	16	Group size	8
LoRA rank	32	Episodes per task	10,240
Episode length	512	Number of training tasks	5

Table 8: Method-specific hyperparameters for the source-reported matched-protocol baselines (Hu et al., 2026). Sequential Fine-Tuning, Dynamic Weight Expansion, and Multi-Task Training use no additional method-specific hyperparameters.

Method	Parameter	Value
EWC	Regularization coefficient λ; Fisher samples	10^6; 65,536
Expert Replay	Replay loss weight; stored trajectories; replay batch size	0.03; 10; 8192
Dark Experience Replay	Replay loss weight; stored trajectories; replay batch size	0.03; 10; 8192
SLCA (Layered LR)	Slow learning rate; fast learning rate	4×10^{-6}; 4×10^{-5}
RETAIN	Merge coefficient λ	0.5

E.4 Baselines

Baselines in the original CRL-VLA experiments. **1) Sequential Learning (SL)** (Liu et al., 2023) fine-tunes the VLA on each new task without an explicit retention mechanism. **2) Multi-Task Learning (MTL)** (Liu et al., 2023) learns from new and old tasks jointly. **3) Learning Without Forgetting (LWF)** (Li and Hoiem, 2018) distills predictions from the previous model while learning the current task. **4) Experience Replay (ER)** (Rolnick et al., 2019) interleaves stored historical examples with current-task data. **1) Sequential Fine-Tuning** is the same no-retention sequential baseline described above. **2) Elastic Weight Consolidation (EWC)** penalizes changes to parameters estimated to be important for earlier tasks using Fisher information (Kirkpatrick et al., 2017). **3) Expert Replay** replays stored expert trajectories through an auxiliary behavior-cloning loss (Rolnick et al., 2019). **4) Dark Experience Replay (DER)** stores historical model outputs and regularizes current outputs toward those stored targets (Buzzega et al., 2020). **5) Dynamic Weight Expansion** allocates a task-specific LoRA adapter for each task, instantiating the parameter-isolation principle (Rusu et al., 2016b). **6) SLCA (Layered LR)** applies a lower learning rate to the pretrained representation and a higher learning rate to the action head; this matched-protocol adaptation uses the layered-rate component of SLCA (Zhang et al., 2023). **7) RETAIN** merges each task's parameter update back into the base model with a discount coefficient (Yadav et al., 2026). **8) Multi-Task (Oracle)** jointly trains on all five tasks and therefore serves as a non-continual upper reference rather than a deployable CRL method. **1) Stellar-VLA** is a knowledge-driven continual imitation-learning framework. It jointly learns task representations and a task–skill knowledge space, then routes through experts using learned knowledge relations and semantic embeddings (Wu et al., 2025). **2) Info-VLA** preserves cross-modal structure using a frozen-teacher replay anchor for contrastive alignment together with a mutual-information objective between visual and language representations (Zhao et al., 2026).

E.5 Evaluation Metrics Formulas

Final Performance. The Final Average Return (**FAR**) is computed as:

$$\mathrm{FAR} := \frac{1}{T} \sum_{i=1}^{T} R_{T,i},$$

which represents the average success rate of the VLA across all tasks after training.

Table 9: Training hyperparameter configuration.

Parameter	Value
General	
LoRA Rank	32
Gradient Accumulation Steps	1
PPO Epochs	1
PPO Clip Range	0.2
PPO Clip High	0.2
Max Step Batch Size	2
Learning Rate (LoRA modules)	1.0×10^{-4}
Learning Rate (Action head)	5.0×10^{-5}
Weight Decay	1.0×10^{-4}
Gradient Clip Norm (model)	1.0
Gradient Clip Norm (header)	1.0
Total Steps M	12
Eval Interval $I_{interval}$	1
Update Times N	10
CRL-VLA Weight	
Q-Function Consistency ($\mathcal{L}^{Q}_{\mathrm{GCV}}$)	0.01
KL Divergence Scale (λ_{KL})	1.0×10^{-6}
Q Loss Coef ($\mathcal{L}^{Q}_{\mathrm{MC}}$)	1.0
Retention Critic (GCV)	Enabled

Stability and Forgetting. Backward Transfer (**BWT**) measures how learning a new task affects previous tasks:

$$\mathrm{BWT} := \frac{1}{T-1} \sum_{i=1}^{T-1} \left(R_{T,i} - R_{i,i} \right),$$

where a negative BWT indicates catastrophic forgetting.

Forgetting (**F**) quantifies the degradation from peak to final performance:

$$F_i := \max_{k \geq i} R_{k,i} - R_{T,i}, \quad \mathrm{F} := \frac{1}{T-1} \sum_{i=1}^{T-1} F_i.$$

Plasticity and Forward Transfer. Forward Transfer (**FT**) reflects the influence of prior knowledge on new tasks:

$$\mathrm{FT} := \frac{1}{T-1} \sum_{i=2}^{T} \left(R_{i-1,i} - b_i \right),$$

where b_i is the baseline success rate of task $\mathcal{T}_i$ obtained by training from scratch on task i without prior continual pre-training, i.e., before learning previous tasks. Positive FT indicates beneficial forward knowledge transfer.

Matched-protocol metrics. For the comparison in Table 2, let $S_{i,j} \in [0, 1]$ denote success on task j after completing training through task i, and let $S_{0,j}$ denote the initial base model's success on task j before continual training (Hu et al., 2026). Training Average Final Success (**AVG**) is

$$\mathrm{AVG} := \frac{1}{T} \sum_{j=1}^{T} S_{T,j}.$$

Its change from the initial checkpoint is

$$\Delta\mathrm{AVG} := \frac{1}{T} \sum_{j=1}^{T} \left(S_{T,j} - S_{0,j} \right).$$

Negative Backward Transfer (**NBT**) measures degradation relative to performance immediately after each task was learned:

$$\mathrm{NBT} := \frac{1}{T-1} \sum_{j=1}^{T-1} \left(S_{j,j} - S_{T,j} \right).$$

Lower NBT indicates better retention. With an identical success matrix, $\mathrm{NBT} = -\mathrm{BWT}$. Forward Transfer (**FWT**) measures the change in success on a future task before it is trained, relative to the initial base model:

$$\mathrm{FWT} := \frac{1}{T-1} \sum_{j=2}^{T} \left(S_{j-1,j} - S_{0,j} \right).$$

Positive FWT indicates beneficial transfer to unseen tasks. Unlike our FT above, this definition does not use a separately trained single-task reference b_j. Values in Table 2 are multiplied by 100 and reported as percentages.

E.6 Visual Perturbation

Libero-Long

- **Perturbation type:** Random translation
- **Applied to:** Third-person view and wrist camera view
- **Translation intensity:** 0.1 (i.e., up to 10% of image width/height)
- **Implementation:** Pixels exposed after translation are filled with black

Libero-Spatial and Libero-Object

- **Perturbation types:** Random translation, rotation, and color jitter
- **Translation intensity:** 0.2 (up to 20% of image width/height)
- **Rotation:** Randomly sampled between 1° and 19°, maximum 20°
- **Color jitter:** Random changes in saturation, brightness, contrast, and sharpness
- **Applied to:** All visual inputs (third-person and wrist camera views)

E.7 Training Procedure and Hyperparameters

For the OpenVLA-OFT simulation implementation (Kim et al., 2025), we adopt the official pre-trained checkpoint and apply LoRA adapters for task adaptation. This implementation builds on the RIPT-VLA codebase (Tan et al., 2025). Appendix C.3 specifies the on-policy CRL instantiation. Training is performed with a LoRA rank of 32, and the detailed hyperparameter configuration is summarized in Tab. 9. Unless otherwise stated, all hyperparameters are kept consistent across experiments to ensure fair comparison. All experiments are executed on the following hardware platform: **CPU:** Intel(R) Xeon(R) Platinum 8358 @ 2.60GHz; **GPU:** NVIDIA A100-SXM4-80GB.

E.8 THREE-TASK ONLINE CONTINUAL LEARNING WITH EXPO

The first real-world study instantiates CRL-VLA in an online off-policy setting based on the Expo-FT interface (Dong et al., 2026). It tests application of the retention–adaptation framework during robot interaction over a three-task sequence. This appendix documents the implementation and the coverage of the available evaluation records.

Robot platform and task stream. The online experiments use an xArm manipulation setup with RGB observations, robot proprioception, and action chunks executed under a supervised operator interface. The initial VLA is a $\pi_0.5$-style policy first adapted by supervised fine-tuning on task demonstrations. The evaluated sequence contains three stages, in the order: eggplant into a cup (T1), eggplant into a box (T2), and a blackboard eraser into a box (T3). Each stage follows the reset–attempt–label protocol and is trained until its interaction budget is reached. Figure 7 illustrates manipulation phases in the setup; the task order used for the quantitative results is the one specified here.

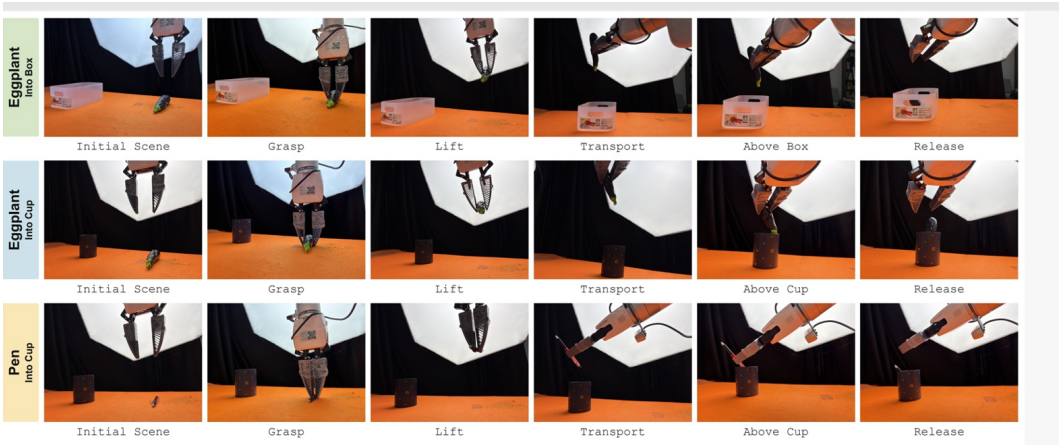

Figure 7: Illustrative manipulation phases in the Expo setup: initial scene, grasp, lift, transport, above-container, and release. Rows show eggplant-to-box, eggplant-to-cup, and pen-to-cup examples, not the evaluated task order. The pen-to-cup example illustrates the setup; quantitative Task 3 is eraser-to-box.

Data collection and evaluation protocol. Training is online and off-policy. During each attempt, the policy proposes action chunks, the robot executes them with optional human takeover for safety and recovery, and the resulting transitions are inserted into the current-stage replay buffer. Sparse success labels are supplied by the operator at the end of each attempt. The updated CRL-VLA evaluation record contains all six observed checkpoint–task pairs in the three-stage sequence, with ten trials per pair. The earlier baseline record contains Task 2 at checkpoint 2 and Tasks 1–3 at checkpoint 3.

For checkpoint k and task i, let $P_{k,i}$ be the grasp (progress) rate and $S_{k,i}$ the complete placement (success) rate, each divided by all ten trials in that cell. The task score is $R_{k,i} = (P_{k,i} + S_{k,i})/2$. Thus, a grasp without placement earns half credit, and a complete placement earns full credit. Table 10 preserves both rates and their combined scores; untested tasks remain missing, not zero.

Using the combined scores, FAR is $(0.25 + 0.35 + 0.45)/3 = 0.350$. For the online CRL-VLA row in Table 4, the $\pm$ term is the population standard deviation across the three final task scores, $\sqrt{\sum_{i=1}^{3}(R_{3,i} - \mathrm{FAR})^2/3} = 0.08165$, and MAD is $\sum_{i=1}^{3} |R_{3,i} - \mathrm{FAR}|/3 = 0.06667$. These describe variation across tasks, not uncertainty across independent training seeds. BWT over both earlier tasks is $[(0.25 - 0.30) + (0.35 - 0.45)]/2 = -0.075$; forgetting is $[(0.40 - 0.25) + (0.45 - 0.35)]/2 = 0.125$. The baseline's previously reported retention metrics remain restricted to Task 2; on this matched subset, CRL-VLA has BWT -0.100 and forgetting 0.100. No forward-transfer estimate is inferred from missing future-task evaluations.

Table 10: CRL-VLA online evaluation rates and combined scores (%). CP1–CP3 correspond to ours1–ours3 in the source record. Dashes denote untested tasks.

Task	Measure	CP1	CP2	CP3
T1: eggplant into cup	Grasp (progress)	50	70	50
	Placement (success)	10	10	0
	Combined score	30	40	25
T2: eggplant into box	Grasp (progress)	–	70	60
	Placement (success)	–	20	10
	Combined score	–	45	35
T3: eraser into box	Grasp (progress)	–	–	60
	Placement (success)	–	–	30
	Combined score	–	–	45

The critic construction, candidate selection, TD target, and actor updates are specified in Appendix C.5.

E.9 Five-Task Offline Real-World Continual Sequence

This experiment extends the real-world evaluation from online interaction to offline continual learning with the WAV backbone (Li et al., 2026c), and studies a longer sequence of five manipulation instructions. It assesses performance in this five-task setting rather than isolating sequence length as the only changing factor. In the supplied task order, T1 places an eggplant in a cup, T2 places a pen in a cup, T3 places a bottled milk drink in a box, T4 pours tea from a teapot into a fair cup, and T5 pours tea from the fair cup into a teacup (Fig. 8). The baseline is WAV trained offline without the continual-learning component; the comparison applies CRL-VLA to the WAV backbone. The training hyperparameters are listed in Table 11. After stage k, checkpoint CPk is tested on tasks T1 through Tk. Each observed checkpoint–task cell contains ten trials, giving 150 trials per method across the triangular evaluation matrix. Untested future tasks are shown as dashes, not zeroes. A successful grasp counts toward progress success; only a complete task execution counts toward final success. Both rates use all ten trials in the cell as denominator.

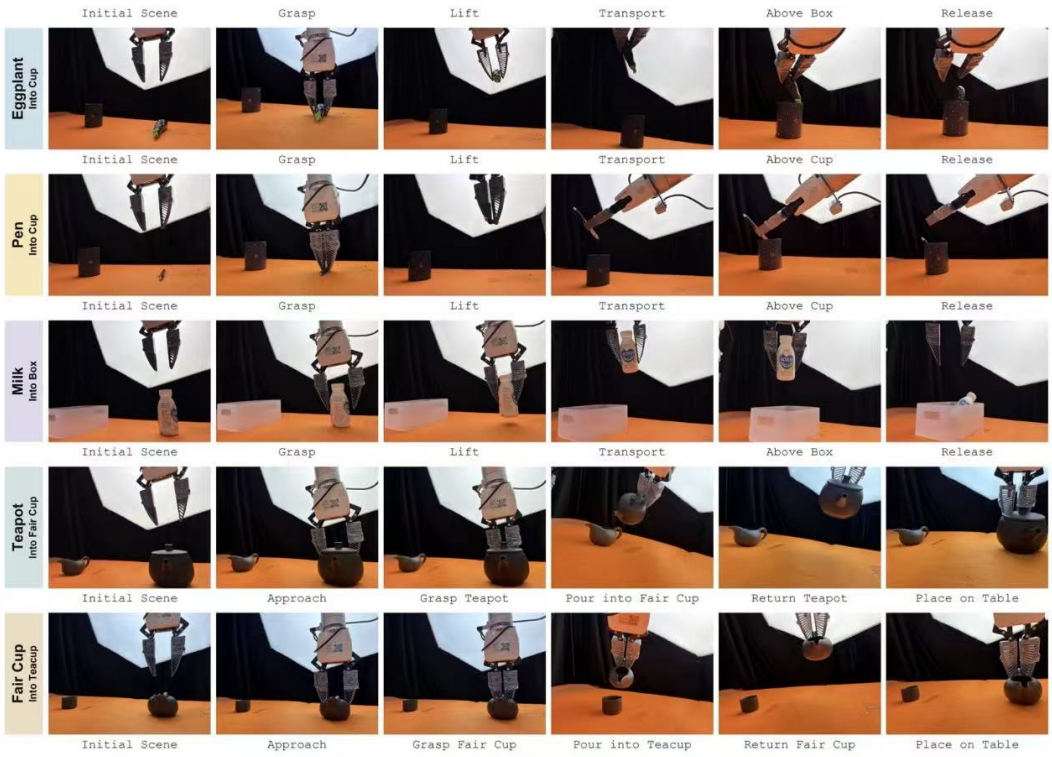

Figure 8: Task order and illustrative execution phases for the five-task offline real-world sequence. From top to bottom: eggplant into cup, pen into cup, milk drink into box, pouring from a teapot into a fair cup, and pouring from the fair cup into a teacup. The T1 montage labels its fifth phase "Above Box," although the destination shown is a cup. The panels document task procedures, not quantitative outcomes.

Tables 12 and 13 reproduce the two stage matrices from the evaluation records. The baseline CP4 source record contains three overwritten trial marks: two for T3 and one for T4. Affected point estimates are marked with †. The fifth-checkpoint task rates and their FAR means do not depend on those ambiguous CP4 marks. The supplied evaluation sheets do not record independent training seeds or uncertainty estimates; all comparisons here are descriptive.

Table 11: Training hyperparameter configuration for WAV.

Parameter	Value
General	
Training steps	20,000
Learning rate	5×10^{-5}
Batch size	128
Gradient accumulation	1
Optimizer	AdamW
Adam betas	$\beta_1 = 0.9,\ \beta_2 = 0.95$
Epsilon	1×10^{-8}
Weight decay	1×10^{-5}
LR scheduler	Constant with warmup
Warmup steps	1,000
Gradient clipping	1.0
CRL-specific Loss Weights	
Value-loss coefficients ($c_{\mathrm{ret}}, c_{\mathrm{adapt}}$)	Adaptive relative to BC loss

Table 12: Final task success in the five-task offline real-world sequence (%). Each reported cell is based on ten trials. † denotes a value transcribed from ambiguous, overwritten trial marks.

Method	Checkpoint	T1	T2	T3	T4	T5
ER (Li et al., 2026c)	CP1	30	–	–	–	–
	CP2	20	10	–	–	–
	CP3	20	0	10	–	–
	CP4	0	0	10†	80†	–
	CP5	30	0	50	50	40
CRL-VLA	CP1	30	–	–	–	–
	CP2	30	20	–	–	–
	CP3	40	10	50	–	–
	CP4	50	0	100	80	–
	CP5	0	40	100	90	60

Table 13: Grasp (progress) success in the same five-task sequence (%). Each reported cell is based on ten trials; the † convention matches Table 12.

Method	Checkpoint	T1	T2	T3	T4	T5
ER (Li et al., 2026c)	CP1	90	–	–	–	–
	CP2	100	40	–	–	–
	CP3	100	90	20	–	–
	CP4	100	50	50†	80†	–
	CP5	90	60	60	50	40
CRL-VLA	CP1	90	–	–	–	–
	CP2	100	70	–	–	–
	CP3	90	40	50	–	–
	CP4	80	50	100	90	–
	CP5	100	60	100	90	60

E.10 Cross-Embodiment Evaluation on Piper

To evaluate cross-embodiment applicability and retention against ER, we use a two-task WAV-based setup on Piper. The sequence is bowl organization followed by cube picking. We collected approximately 100 successful demonstrations for each task. The demonstrations were collected using

a real-world Piper robot under visually clean but distinct background conditions. Each trajectory consists of RGB observations from two wrist-mounted cameras with resolution $240 \times 320 \times 3$, RGB observations from a third-person overhead camera with resolution $240 \times 424 \times 3$, and a 14-dimensional robot joint-state vector.

The Piper comparison evaluates CRL-VLA against ER using the offline WAV backend (Li et al., 2026c). Appendix C.4 specifies its frozen value expert and historical/current-task flow-supervision objectives.

Evaluation role. Table 4 reports the available final-checkpoint metrics, and Fig. 5 illustrates executions of both tasks. This experiment evaluates forgetting relative to ER on Piper. Its cross-embodiment role is framework applicability through platform-specific training and evaluation, rather than direct transfer of an unchanged policy from xArm. The task set, sequence length, and comparator differ from the five-task offline study, so the two WAV experiments address complementary questions.

E.11 Computational Cost and Buffer Scalability

Table 14 reports profiling measurements for the PPO configuration with a 30-trajectory buffer. CRL-VLA has lower recorded per-stage and rollout-generation times than ER, with peak GPU memory of 32.40 versus 31.75 GB. These measurements characterize this configuration on the reported hardware.

Table 14: Computational profiling of the PPO configuration and ER on an NVIDIA A100-SXM4-80GB GPU. Relative change uses ER as the reference; no timing uncertainty is available.

Metric	CRL-VLA	ER	Relative Change
Per-stage wall-clock time (sec)	3132.9	3688.8	$\downarrow 15.1\%$
Rollout generation time (sec)	450.5	495.3	$\downarrow 9.1\%$
Peak GPU memory (GB)	32.40	31.75	Comparable

The profiled PPO buffer keeps 30 recent successful trajectories from the immediately preceding task and discards older ones after each stage, with a reported footprint of 13.58 MB. These measurements apply to the preceding-task buffer configuration rather than the historical reservoir in Algorithm 1.

E.12 RL vs. SFT in Continual VLA

We further compare RL-based and SFT-based adaptation in the continual VLA setting. While SFT directly optimizes imitation objectives, RL post-training optimizes task returns and can better account for sequential performance changes across tasks. As shown in Table 15, RL-based adaptation achieves comparable forward transfer to SFT in this recorded setting, with higher backward transfer and lower forgetting. These results provide setting-specific empirical motivation for focusing on RL post-training in continual VLA learning.

Table 15: Comparison between SFT-based and RL-based adaptation in continual VLA.

Method	FT $\uparrow$	BWT $\uparrow$	F $\downarrow$
SFT-based adaptation	0.2660	-0.5980	0.5980
RL-based adaptation	0.2667	0.0000	0.1333