CAPO: REAL-WORLD REINFORCEMENT LEARNING FOR VLA VIA CORRECTIVE ADJOINT POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

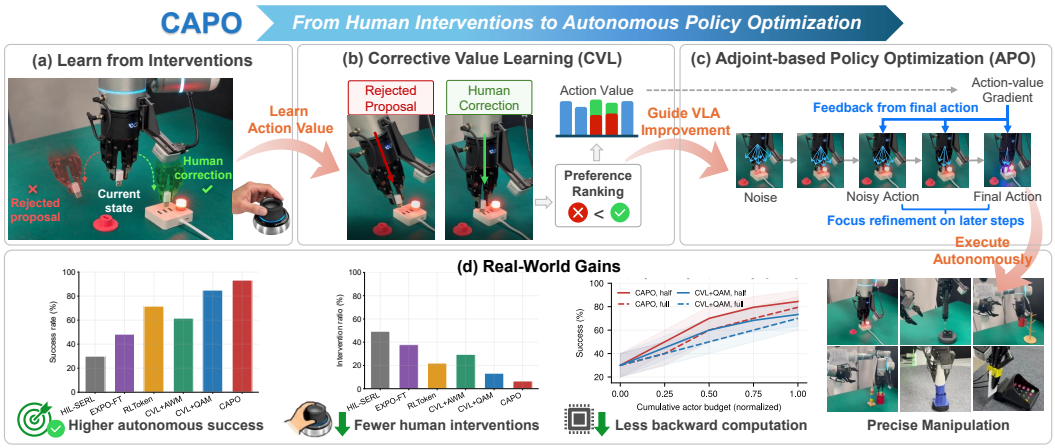


Figure 1: In this work, we propose CAPO, a real-world reinforcement learning framework for flow-based VLAs that *connects human interventions to policy optimization*. (a) Human interventions provide corrective experience, which (b) CAPO turns this corrective experience into informative action values, (c) then translates the learned action-value guidance into updates of the flow-based policy, (d) and achieves higher autonomous success and fewer human interventions.

ABSTRACT

Human-in-the-Loop Reinforcement Learning (HiL-RL) offers a practical route for adapting Vision-Language-Action (VLA) models through real-world interaction guided by human interventions. Yet effectively leveraging these interventions remains challenging, their corrective signals must first support informative value learning and then be translated into effective updates of the VLA's multi-step flow policy. To address these challenges, we propose **C**orrective **A**djoint **P**olicy **O**ptimization (**CAPO**), a real-world RL framework that couples Corrective Value Learning (CVL) with Adjoint-based Policy Optimization (APO). Specifically, CVL exploits human corrections both to learn action values from their observed outcomes and to provide comparative supervision against rejected policy proposals. Guided by the learned critic, APO propagates action-value gradients along the VLA's native flow trajectories and converts them into local policy updates, while preserving the original deployment sampler. Across six real-world manipulation tasks, CAPO achieves the highest mean autonomous success of 93.3% and the lowest intervention rate among the evaluated methods.

1 INTRODUCTION

Vision-language-action (VLA) models provide a strong foundation for robot manipulation, but improving their reliability in real-world tasks still requires learning through trial and error (Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025b; 2026). Human-in-the-Loop Reinforcement Learning (HiL-RL) makes this process practical by allowing operators to intervene when the policy makes mistakes or encounters difficult states (Chen et al., 2025b; Luo et al., 2025; Pan et al.,

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.

2026). Beyond recovering individual trials, the goal of HiL-RL is to learn from these interventions to improve the policy and reduce the need for future human assistance.

Achieving this goal requires bridging two connected steps: *learning action values from interventions*, and *translating value guidance into policy updates*. For the first step, existing RL methods learn action-value functions on executed transitions using temporal-difference (TD) objectives (Zhao et al., 2025; Wagenmaker et al., 2025). However, an intervention can additionally reveal a preference between corrective actions and rejected policy proposals, even though its outcome is unobserved. This preference supplies relative value information that TD supervision alone does not explicitly capture. For the second step, recent VLAs increasingly adopt flow-based policy that generate action chunks through multiple velocity predictions (Black et al., 2024; Intelligence et al., 2025b; 2026), making direct policy optimization challenging due to the numerical instability of backpropagating action-value feedback through the full generation process (Zhang et al., 2026). Existing approaches either avoid this difficulty by introducing auxiliary policy components (Xu et al., 2026), or use scalar weighting that ignores action-value gradients (Intelligence et al., 2025a; Yang et al., 2026). Local velocity supervision offers an alternative (Wang et al., 2026b;a), but constructing effective step-wise targets requires balancing faithful value propagation with stable and efficient policy updates.

Our key insight is to treat these two steps as a continuous learning process. A human intervention provides both the outcome of the executed corrective action and a preference over the proposal it replaces, offering complementary supervision for learning action values. The resulting value guidance can then be carried along the VLA's native flow trajectories, allowing feedback on a final action-chunk to supervise the intermediate velocity predictions that produced it. In this way, interventions inform not only which actions the policy should favor, but also how its generation process should be refined, extending their use beyond direct imitation of corrective actions.

To this end, we instantiate this insight as **C**orrective **A**djoint **P**olicy **O**ptimization (**CAPO**), a real-world RL framework that couples *Corrective Value Learning* (CVL) with *Adjoint-based Policy Optimization* (APO), as illustrated in Figure 1. CVL augments TD regression on executed transitions with within-state preference ranking between corrective actions and rejected proposals, providing relative value supervision without assigning outcomes to unexecuted actions. The resulting critic supplies action-value gradients that APO propagates backward along the VLA's native ODE trajectories and converts into local velocity targets for the action expert. To support stable online refinement, APO employs a reference policy that remains fixed within each actor phase and is refreshed between phases. It further restricts adjoint construction and velocity matching to a configurable terminal interval, reducing backward computation while retaining the complete forward generation process. At deployment, the updated VLA retains its original sampler, without an auxiliary actor or inference-time critic guidance. Across six real-world manipulation tasks spanning precise alignment and contact-rich manipulation, CAPO achieves the highest mean autonomous success among the evaluated methods at 93.3%, with a final-window intervention ratio of 6.9%.

Overall, our contributions are summarized:

- We propose CAPO, a real-world RL framework for flow-based VLAs that turns human corrective experience into autonomous policy optimization by connecting corrective value learning with value-guided action generation.
- We adapt adjoint-based policy optimization to translate the learned value gradients into VLA updates, with phase-wise reference regularization and terminal-interval matching to control update magnitude and computational cost.
- We validate CAPO on six real-world manipulation tasks, demonstrating improved autonomous success and reduced human intervention.

2 Related Work

Real-world reinforcement learning for VLAs. Online post-training adapts pretrained VLAs using robot experience (Guo et al., 2025; Chen et al., 2025a; Dong et al., 2026b; Yu et al., 2026). ALOE combines current-policy action evaluation with value-weighted updates (Yang et al., 2026), while RECAP uses advantage-conditioned training (Intelligence et al., 2025a). Other methods learn residual policies or actors conditioned on VLA representations (Yuan et al., 2025; Chen et al., 2026; Xu et al., 2026); EXPO-FT combines action editing and selection with VLA fine-tuning (Dong et al.,

2026c;a). Our CAPO directly updates the flow action expert, retaining the same policy architecture at deployment.

Learning from human interventions. HG-DAgger learns from human-gated corrections (Kelly et al., 2019). HIL-SERL and ConRFT combine demonstrations and online interaction with RL (Luo et al., 2025; Chen et al., 2025b), and SiLRI studies suboptimal interventions (Zhao et al., 2025). Concurrent with our work, PACT uses progress-based preference calibration (Liu et al., 2026), and VLA-Precision explicitly ranks state-aligned corrective and overwritten actions within critic learning (Su et al., 2026). Our CAPO shares the use of intervention-derived preferences but differs in how value learning is coupled to policy optimization: we supplement TD supervision with corrective–rejected action ranking and translate the resulting critic's action gradients into velocity-field targets along native ODE trajectories.

Value-guided flow optimization. Flow Q-Learning distills an auxiliary one-step actor (Park et al., 2025). FlowDPG, Q-VGM, and CGPO instead use value gradients to construct velocity supervision without full-sampler backpropagation (Shi et al., 2026; Wang et al., 2026b; Zhou et al., 2026; Ding et al., 2026). QAM adapts adjoint matching (Domingo i Enrich et al., 2025) to value-based RL through stochastic, behavior-regularized policy optimization (Li & Levine, 2026). Learning While Deploying further uses QAM for VLA adaptation (Wang et al., 2026a). Deterministic adjoint matching has been developed for image generation (Guo et al., 2026); we build on those ideas for online action-value-guided learning. The resulting design combines a learned terminal critic, a reference policy refreshed between actor phases, and intervention-derived supervision.

3 Preliminaries and Problem Formulation

Flow-based VLA. Given a context s containing visual observations, language instruction, and proprioception, a flow-based VLA generates an action chunk by integrating a velocity field (Intelligence et al., 2026; Lu et al., 2026):

$$x_0 \sim \mathcal{N}(0, I), \qquad \frac{\mathrm{d}x_\tau}{\mathrm{d}\tau} = v_\theta(x_\tau, \tau \mid s), \quad \tau \in [0, 1]. \tag{1}$$

The endpoint $A = x_1$ defines a sequence of H robot actions, inducing the policy $\pi_\theta(A \mid s)$. Here τ denotes flow time, distinct from environment decision time t. The ODE is deterministic conditional on (s, x_0); random initial noise still induces a distribution over actions.

Action-chunk value learning. For a fixed execution horizon H, A_t denotes the sequence actually executed at decision t. It produces reward $R_t = \sum_{j=0}^{H-1} \gamma^j r_{t,j}$ and successor s_{t+1}. Under a Markov context, the chunk-level action value satisfies

$$Q^\pi(s_t, A_t) = \mathbb{E}\left[R_t + (1 - d_t)\gamma^H \mathbb{E}_{A' \sim \pi(\cdot \mid s_{t+1})} Q^\pi(s_{t+1}, A') \,\middle|\, s_t, A_t\right], \tag{2}$$

where d_t is the terminal flag. Executed actions may come from different behavior policies, while the bootstrap evaluates continuation by π. The objective is to improve autonomous task return using demonstrations and human-assisted online experience. If a controller interrupts chunks or executes only a prefix, replay and critic inputs must use a consistent executed action unit; Appendix B.1 details this requirement.

4 Method

4.1 Framework Overview

As shown in Figure 2, CAPO starts from task-specific supervised fine-tuning of the VLA and then alternates human-assisted data collection, corrective value learning (CVL), and adjoint-based policy optimization (APO)-based action-expert updates. During online RL, all executed transitions are stored in $\mathcal{D}_{\text{exec}}$. When a intervention replaces a policy proposal at the same decision state, the aligned corrective–rejected pair is retained in $\mathcal{D}_{\text{inter}}$. Demonstrations and human interventions also form $\mathcal{D}_{\text{demo}}$ for actor imitation. During each learning round, CVL combines executed outcomes and corrective preferences to update the critic. We then freeze the critic parameters and use APO to

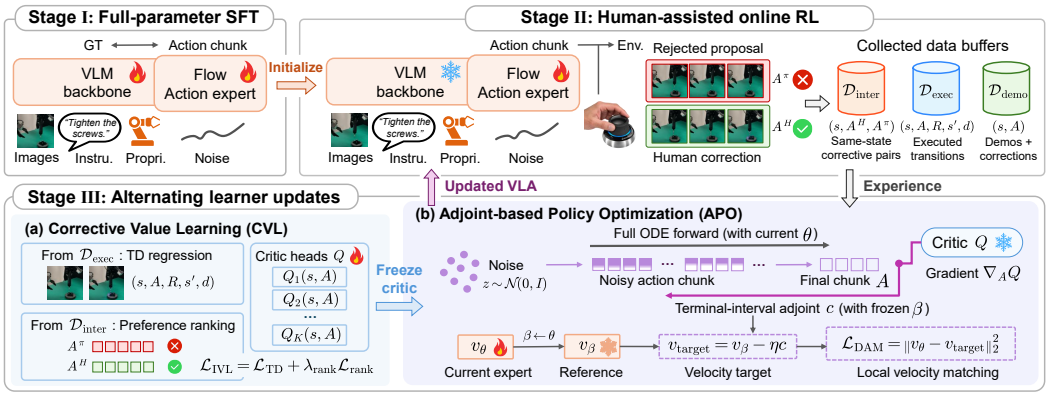


Figure 2: **Training pipeline of CAPO.** A task-adapted VLA collects human-assisted experience in three buffers: executed transitions ($\mathcal{D}_{\text{exec}}$), aligned corrective–rejected pairs ($\mathcal{D}_{\text{inter}}$), and imitation examples ($\mathcal{D}_{\text{demo}}$). CVL updates the critic using TD regression and preference ranking. With the critic frozen, APO constructs terminal-interval adjoints through a reference snapshot from the actor at each phase, then updates the action expert with detached velocity targets.

translate its action-value gradients into updates of the native flow policy. The updated VLA collects the next round of experience. Algorithm 1 summarizes this procedure, and Appendix B.1 specifies replay alignment and action semantics.

4.2 Corrective Value Learning

To learn more informative action values from limited corrective experience, we propose *Corrective Value Learning* (CVL). CVL combines TD regression on executed transitions with local preferences between human interventions and overwritten policy proposals, using both the outcomes of executed actions and the comparisons revealed by interventions.

Learning from executed outcomes. The critic evaluates the return from an action followed by autonomous continuation under the current VLA. Accordingly, we bootstrap with current-policy samples, including when the recorded action was executed by a human. Following action-level policy evaluation (Yang et al., 2026), we hold π_θ fixed during each critic phase and train K online heads Q_{ϕ_k} with Polyak-averaged targets $Q_{\bar{\phi}_k}$. For an executed transition $(s_t, A_t, R_t, s_{t+1}, d_t)$, define

$$\begin{aligned} \bar{Q}(s, A) &= \frac{1}{K}\sum_{k=1}^{K} Q_{\bar{\phi}_k}(s, A), \\ y_t &= R_t + (1 - d_t)\gamma^H \mathbb{E}_{A' \sim \pi_\theta(\cdot|s_{t+1})} \bar{Q}(s_{t+1}, A'), \end{aligned} \tag{3}$$

where d_t indicates termination and the expectation is estimated using fresh current-policy samples. Each online head regresses toward the same detached target:

$$\mathcal{L}_{\text{TD}} = \mathbb{E}_{\mathcal{D}_{\text{exec}}}\left[\frac{1}{K}\sum_{k=1}^{K}\left(Q_{\phi_k}(s_t, A_t) - \text{sg}[y_t]\right)^2\right], \tag{4}$$

where sg denotes stop-gradient. The recorded action must correspond to the observed reward and successor. Thus, when a human overwrites a proposal, the transition supervises the executed correction; it provides no observed outcome for the rejected proposal.

Learning from corrective comparisons. An intervention additionally reveals a local preference between the correction and the rejected proposal. CVL uses this comparison to provide relative value supervision for the unexecuted action. Let $(s_i^I, A_i^H, A_i^\pi) \in \mathcal{D}_{\text{inter}}$ denote a pair consisting of an accepted human correction A_i^H and an overwritten proposal A_i^π, aligned in decision state, action representation, and horizon. We minimize

$$\mathcal{L}_{\text{rank}} = \mathbb{E}_{\mathcal{D}_{\text{inter}}}\left[\frac{1}{K}\sum_{k=1}^{K}\left[m_i + Q_{\phi_k}(s_i^I, A_i^\pi) - Q_{\phi_k}(s_i^I, A_i^H)\right]_+^2\right], \tag{5}$$

where $[z]_+ = \max(z, 0)$ and $m_i \geq 0$ is the preference margin. Comparisons are computed within each head before averaging the losses. Both value branches receive gradients when the margin is violated, and the loss vanishes once the correction is ranked above the proposal by the prescribed margin. This makes fuller use of each human intervention by providing value supervision for both the correction and the rejected proposal.

CVL objective. We combine outcome-based regression and corrective preference supervision:

$$\mathcal{L}_{\mathrm{CVL}} = \mathcal{L}_{\mathrm{TD}} + \lambda_{\mathrm{rank}} \mathcal{L}_{\mathrm{rank}}, \qquad \lambda_{\mathrm{rank}} \geq 0. \tag{6}$$

TD regression grounds values in observed rewards and autonomous continuation, while preference ranking constrains relative action values at intervention states. The coefficient λ_{rank} balances these complementary signals. Notably, the ranking objective treats accepted corrections as locally preferable to their paired proposals, without requiring globally optimal human behavior. Appendix A.2 discusses this assumption. The resulting critic provides action-value gradients for the APO updates described next. Appendix A.3 relates corrective comparisons to APO's terminal action gradients.

4.3 Adjoint-based Policy Optimization

To support stable VLA improvement while efficiently exploiting critic action gradients, we propose *Adjoint-based Policy Optimization* (APO), inspired by deterministic adjoint formulations (Guo et al., 2026). APO converts terminal critic gradients into detached velocity supervision along native ODE trajectories, with reference regularization to moderate policy changes and terminal-interval matching to control backward computation. During each actor phase, we freeze the critic parameters and use $\underline{Q}_\phi(s, A) = \min_k Q_{\phi_k}(s, A)$ as the terminal value objective.

Reference-regularized optimization. An imperfect critic can favor actions outside well-supported regions of the data. To discourage excessive policy changes, we penalize velocity deviations from a reference field v_β. At the start of each actor phase, we copy $\beta \leftarrow \theta$ and hold the reference fixed throughout that phase. APO learns a behavior reference from replay (Li & Levine, 2026); our reference instead tracks successive snapshots of the VLA, allowing the regularization center to evolve with policy optimization.

With $u_\theta = v_\theta - v_\beta$, we formulate the reference-regularized control objective as

$$\mathcal{J}_s(u) = \mathbb{E}_{x_0}\left[\int_0^1 \frac{\|u(x_\tau, \tau \mid s)\|_2^2}{2\eta}\,\mathrm{d}\tau - \alpha_Q \underline{Q}_\phi(s, x_1)\right], \tag{7}$$

subject to $\dot{x}_\tau = v_\beta(x_\tau, \tau \mid s) + u(x_\tau, \tau \mid s)$, $x_0 \sim \mathcal{N}(0, I)$, and $\eta, \alpha_Q > 0$. The terminal term favors high-value action chunks, while the running cost penalizes velocity changes within the actor phase. This velocity-space regularization does not impose an exact KL constraint on the action distribution or bound cumulative drift across phases. The native action expert represents u_θ through its parameter updates, without an additional policy module.

From terminal gradients to velocity supervision. The critic evaluates completed action chunks, whereas the actor predicts velocities at intermediate flow states. To connect these quantities, we first generate a complete trajectory using the current field v_θ, without retaining the sampler's parameter-gradient graph. Unlike the stochastic QAM (Dong et al., 2026d; Ghanem & Ghogho, 2026), these candidate trajectories follow the VLA's native ODE without a diffusion-noise schedule.

At the endpoint $A = x_1$, the critic's action gradient initializes a lean adjoint, which is integrated backward along the generated path:

$$c_1 = -\alpha_Q \left.\nabla_A \underline{Q}_\phi(s, A)\right|_{A=x_1}, \qquad \frac{\mathrm{d}c_\tau}{\mathrm{d}\tau} = -[\nabla_x v_\beta(x_\tau, \tau \mid s)]^\top c_\tau. \tag{8}$$

The reference-field Jacobian is evaluated at the current-policy trajectory states. For the quadratic control cost, the first-order control condition $u = -\eta c$ motivates the velocity target

$$v_\tau^{\mathrm{tar}} = v_\beta(x_\tau, \tau \mid s) - \eta c_\tau. \tag{9}$$

This construction transports endpoint value guidance to intermediate velocities. It defines a matching surrogate rather than the exact parameter gradient of Eq. (7) away from self-consistent matching; Appendix A.4 derives the corresponding stationarity conditions.

Controlling backward computation. Although APO avoids full-sampler parameter backpropagation, constructing its targets still requires reference-field vector–Jacobian products along the trajectory. We therefore introduce terminal-interval matching (Guo et al., 2026), restricting adjoint integration and velocity matching to $[\tau_c, 1]$ while retaining complete forward generation. For a fixed trajectory and reference, the adjoint targets within this interval remain identical to those obtained by full matching. This design is motivated by the hypothesis that adapting a pretrained VLA can rely primarily on late-time action refinement, while behavior-cloning anchor provides supervision across the full flow-time interval. We empirically evaluate the resulting computation–performance trade-off, with $\tau_c = 0$ recovering full matching.

APO objective. Given the detached candidate states and adjoint-derived targets, we optimize

$$\mathcal{L}_{\mathrm{APO}} = \mathbb{E}_{\substack{s\sim\mathcal{D}_{\mathrm{exec}},\, x_0\sim\mathcal{N}(0,I) \\ \tau\sim\mathcal{U}[\tau_c,1]}} \left[\left\| v_\theta(\mathrm{sg}[x_\tau], \tau \mid s) - \mathrm{sg}[v_\tau^{\mathrm{tar}}] \right\|_2^2 \right]. \tag{10}$$

Candidate trajectories and targets are regenerated as the actor changes. For this loss, actor parameter gradients pass only through local velocity evaluations; critic action gradients and reference-field vector–Jacobian products remain part of target construction.

We additionally retain a standard flow-matching behavior-cloning anchor on demonstrations and accepted corrections in $\mathcal{D}_{\mathrm{demo}}$. For $(s, A^B) \sim \mathcal{D}_{\mathrm{demo}}$, $z \sim \mathcal{N}(0, I)$, and $\tau \sim \mathcal{U}[0, 1]$, let $\widetilde{x}_\tau = (1-\tau)z + \tau A^B$. The complete actor objective is

$$\begin{aligned} \mathcal{L}_{\mathrm{BC}} &= \mathbb{E}_{s,A^B,z,\tau} \left[\| v_\theta(\widetilde{x}_\tau, \tau \mid s) - (A^B - z) \|_2^2 \right], \\ \mathcal{L}_{\mathrm{actor}} &= \mathcal{L}_{\mathrm{APO}} + \lambda_{\mathrm{BC}} \mathcal{L}_{\mathrm{BC}}, \qquad \lambda_{\mathrm{BC}} > 0. \end{aligned} \tag{11}$$

The anchor supplies direct action supervision throughout training and samples the full flow-time interval, independently of the APO cutoff. At deployment, the updated VLA retains its original full ODE sampler without auxiliary actors. Notably, determinism refers to the flow dynamics; the policy remains stochastic through its initial noise.

5 Experiments

The experiments address four questions: (1) Does CAPO improve autonomous success and reduce human intervention? (Section 5.2) (2) Does CVL improve value discrimination and policy learning? (Section 5.3) (3) How does APO, including reference refresh and matching interval, affect performance and training cost? (Section 5.4) (4) Is CAPO robust to environmental changes and unreliable intervention preferences? (Section 5.5) We first describe the experimental setup, then examine each question.

5.1 Experimental Setup

Tasks, hardware, training, and evaluation. As shown in Figure 3, we conduct experiments on six real-world tasks spanning precise alignment and contact-rich manipulation: USB Insertion, Peg-in-Hole, Block Stacking, Screw Tightening, Card Swiping, and Cup Hanging. The setup uses Franka FR3/Rizon 4 robots, D405/D435 cameras, SpaceMouse interventions, and a task-adapted $\pi_{0.5}$ backbone. Each task uses 30–50 demonstrations and 40–100 minutes of active collection. We report success every 10 minutes, computed over the 10 most recent completed task attempts in each task/run; task budgets and success criteria are detailed in Appendix C. Final scores use the final-budget checkpoint, counting timeouts and safety aborts as failures. The intervention metric is the final-window fraction of human-controlled primitive steps.

Baselines. Comparisons include frozen SFT, HIL-SERL (Luo et al., 2025), RLToken (Xu et al., 2026), and EXPO-FT (Dong et al., 2026a). CVL+AWM (Xue et al., 2025) and CVL+QAM (Li & Levine, 2026) share our critic and demo-only BC to control the actor comparison. All online variants receive equal active-collection budgets; full-system baselines retain their own architectures. The APO default uses a phase snapshot and $\tau_c = 0.5$. The illustrative solver has 10 forward intervals for every cutoff. The confirmed 40/40/20 replay mixture and training-time prefix KV cache are specified in Appendices B.1 and B.4. Appendix B.2 consolidates the training configuration.

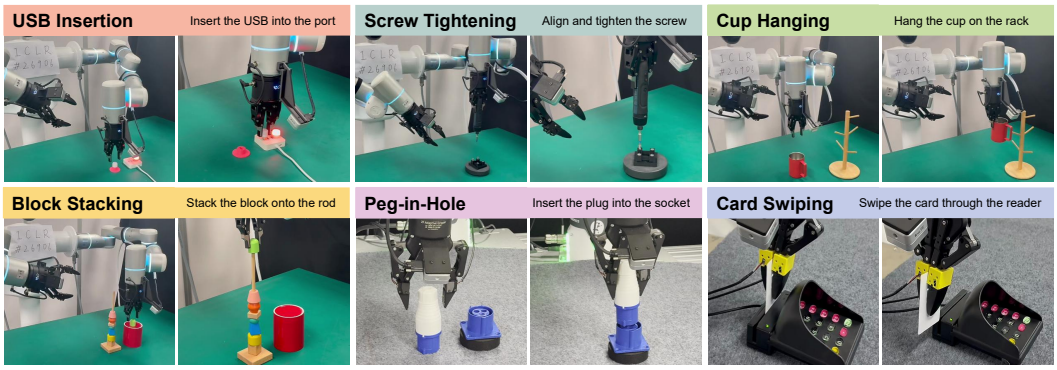


Figure 3: **Six real-world manipulation tasks.** Task settings are detailed in Appendix C.

5.2 Autonomous Performance and Human Intervention

Autonomous task success. As shown in Figure 4, CAPO achieves an average final success rate of 93.3%, exceeding CVL+QAM by 8.3 percentage points and RLToken by 21.7 points. Both CAPO and CVL+QAM reach 100% on Cup Hanging, leaving little room to distinguish them on this task. On Block Stacking and Screw Tightening, however, CAPO reaches 80% and 90%, respectively, compared with 70% and 80% for CVL+QAM. These tasks require multiple stages to succeed within one execution, so an unresolved action error can prevent completion. A plausible explanation is that regularized critic-guided updates help refine imperfect actions while retaining useful pretrained behavior. Since CVL+QAM uses the same corrective preference supervision, the remaining gap motivates a closer examination of the actor update.

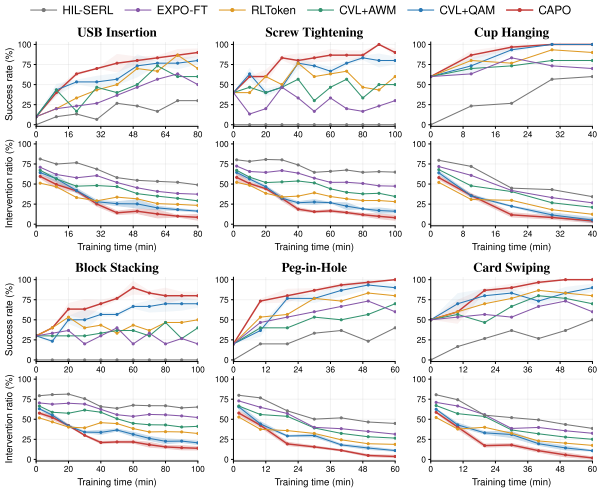
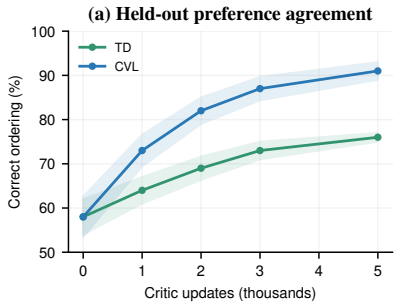


Figure 4: Learning curves for six manipulation tasks. Columns denote tasks; rows report autonomous success and intervention ratios.

Human intervention. The success gains are accompanied by lower reliance on human assistance during collection. On Block Stacking and Screw Tightening, CAPO's final-window intervention ratios fall to 14.5% and 8.3%, compared with 20.6% and 16.1% for CVL+QAM. On the remaining tasks, they range from 2.4% to 8.7%. One possible explanation is that corrections become reusable learning signals: the critic learns to distinguish a correction from the proposal it replaced, while the actor learns both from the corrective action and from subsequent value guidance. This can reduce the need to correct similar mistakes repeatedly.

5.3 Corrective Value Learning

Value discrimination and gradient utility. We compare TD-only and CVL critics on shared replay with a frozen VLA. CVL reaches 91% agreement on held-out corrective–rejected pairs, versus 76% for TD-only (Figure 5 (a)), while normalized Monte Carlo MAE decreases from 0.190 to 0.140 (Figure 5 (b)). Because APO relies on action gradients, we further compare small gradient-directed action perturbations with matched-norm random perturbations, followed by independent autonomous continuations. The return advantage over random perturbations is 0.037 ± 0.006 for CVL, compared with 0.008 ± 0.004 for TD-only. Together, the lower value error and larger return



(b) Value and gradient diagnostics

Critic	MC MAE ↓	$\Delta G_{\nabla - r}$ ↑
TD	$.190 \pm .010$	$.008 \pm .004$
CVL	$.140 \pm .007$	$.037 \pm .006$

(c) Ranking × corrective BC

Rank	Corr. BC	Success ↑	Interv. ↓
Off	Off	73.3 ± 1.9	21.9 ± 1.5
Off	On	81.7 ± 2.5	12.9 ± 0.9
On	Off	82.8 ± 1.7	14.9 ± 1.0
On	On	$\mathbf{93.3} \pm 1.7$	$\mathbf{6.9} \pm 0.8$

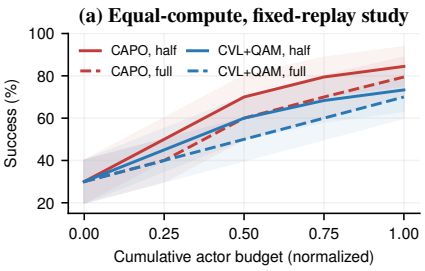
Figure 5: CVL diagnostics and ablations. (a) Agreement on 100 held-out pairs per run. (b) Normalized return error and return difference between critic-gradient and matched-norm random action perturbations. (c) Success and final-window intervention percentages.

advantage suggest that corrective preference supervision improves both value estimation and local action-update directions, providing more useful terminal guidance for APO.

Complementary uses of corrections. We cross preference ranking with corrective-action BC while retaining demonstration BC and APO. As shown in Figure 5 (c), when online corrections are excluded from BC, ranking raises final-window success from 73.3% to 82.8% and reduces intervention from 21.9% to 14.9%. This improvement highlights CVL's ability to exploit interventions through value supervision without directly imitating corrective actions. With corrective BC enabled, ranking further improves success from 81.7% to 93.3%, while reducing intervention from 12.9% to 6.9%. The gains under both settings show that corrective preferences provide useful supervision beyond learning the accepted action alone.

5.4 Adjoint-based Policy Optimization

Actor quality under a computation budget. Figure 6 reports online actor comparisons alongside a separate fixed-replay, equal-compute diagnostic. Under equal update counts, full APO and full QAM have similar costs, 1.00 and 1.02; restricting the adjoint interval reduces these to 0.68 and 0.70. The QAM suffix control is essential: shorter integration is not exclusive to deterministic matching. At the same cutoff, APO reaches 93.3% versus 83.3% for QAM, a comparison of the complete actor designs, including their different references. AWM is cheaper per update but achieves lower success. The fixed-replay curves additionally compare policy quality at matched cumulative actor cost, allowing cheaper updates to use more optimization steps. Half-interval APO makes faster progress than full matching over the plotted budget.



(b) Online actor comparison

Actor	Success (%)	Cost	Mem.
AWM	61.7 ± 10.0	.48	.91
QAM (full)	85.0 ± 8.3	1.02	1.03
QAM (half)	83.3 ± 8.3	.70	1.01
APO (full)	95.0 ± 3.3	1.00	1.00
APO (half)	93.3 ± 5.0	.68	.98

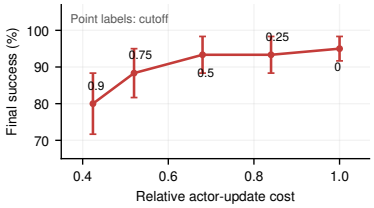
Figure 6: Actor quality and computation. (a) A separate fixed-replay study with a frozen CVL critic; one budget unit equals the cost of 1,000 full-APO updates. (b) Online results at equal update counts; all actors share CVL and BC. "Half" uses $\tau_c = 0.5$ with full forward generation.

Reference refresh and update magnitude. As illustrated in Figure 7, a phase-wise snapshot reaches 93.3% success, compared with 81.7% for a fixed initial reference and 88.3% for a phase-wise EMA. To examine the regularization mechanism, we measure changes in generated actions on fixed held-out states with shared initial noise. The fixed reference gives the smallest phase-to-phase change, but also limits improvement. Refreshing after every optimizer step allows larger

changes and reaches 86.7%, whereas a reference held fixed within each phase supports refinement with smaller changes. The comparison favors controlling the regularization timescale rather than minimizing all policy movement.

Terminal-interval matching. Moving the cutoff from 0 to 0.5 reduces update cost by 32%, while success changes from 95.0% to 93.3%. The full forward path and terminal critic gradient remain available, and retained adjoint targets are unchanged for a fixed path and reference. Moderate truncation can therefore concentrate computation on late-time refinement. At $\tau_c = 0.9$, however, success drops to 80.0%: the cheapest update omits useful supervision.

(a) Matching interval trade-off



(b) Reference refresh at $\tau_c = 0.5$

Reference	Success (%)	Action change
Fixed initial	81.7 ± 8.3	$.035 \pm .005$
Phase EMA	88.3 ± 8.3	$.050 \pm .005$
Phase snapshot	93.3 ± 5.0	$.065 \pm .005$
Per-step snapshot	86.7 ± 6.7	$.170 \pm .020$

Figure 7: APO interval and reference ablations. (a) Success versus relative update cost, with cutoffs annotated and all 10 forward steps retained. (b) Final success and the 95th percentile of normalized action RMS change across adjacent actor phases on fixed probes.

5.5 Robustness

Environmental changes. We evaluate frozen checkpoints under one environmental change at a time, without adaptation. As shown in Figure 8, CAPO reaches 83.3%, 86.7%, and 88.3% under layout shifts, lighting changes, and distractors, respectively. Layout shifts cause the largest drop for both methods, suggesting greater sensitivity to robot–target geometry than to the tested appearance changes. Overall, CAPO retains its advantage under environmental variation.

Reliability of intervention preferences. Success remains 91.7% with 10% preference-label corruption and 85.0% with 20% corruption, but falls to 75.0% at 30%, below the TD+APO baseline of 81.7%. Only ranking labels are flipped; executed TD transitions and corrective BC actions remain unchanged. The baseline is unaffected because it does not use preference labels. Thus, CVL tolerates moderate label noise, but sufficiently unreliable preferences can outweigh its benefits and undermine outcome-based learning, making pair alignment and filtering important.

(a) Environmental robustness

CAPO	Orig.	Layout	Light	Distr.
CAPO	93.3	83.3	86.7	88.3
CVL+QAM	85.0	70.0	76.7	78.3
RLToken	71.7	56.7	61.7	63.3
CVL+AWM	61.7	46.7	51.7	51.7
EXPO-FT	48.3	35.0	38.3	40.0
HIL-SERL	30.0	23.3	23.3	23.3

(b) Preference reliability

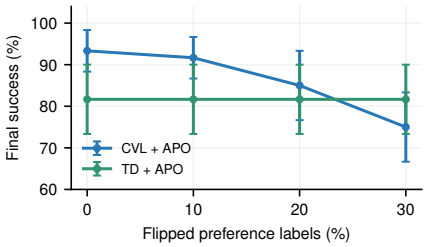


Figure 8: Environmental and preference robustness. (a) Six-task mean success (%). (b) Corrupted training preference labels; TD transitions, corrective BC, and held-out labels stay clean.

6 Conclusion

In this work, we presented CAPO, a real-world reinforcement learning framework for flow-based VLAs that turns human interventions into value-guided updates of the native flow policy. CAPO combines Corrective Value Learning, which learns from both executed corrections and their comparisons with rejected policy proposals, with Adjoint-based Policy Optimization, which propagates critic gradients along native flow trajectories to construct local velocity supervision. CAPO achieves the highest mean autonomous success and the lowest intervention rate across six real-world tasks.

REFERENCES

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

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, et al. π_{rl}: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 8, 2025a.

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.

Zhongxi Chen, Yifan Han, Yanming Shao, Huanming Liu, Congsheng Xu, Xiaoyu Chen, Yao Mu, and Wenzhao Lian. Bora: Bridging offline reinforcement learning and online residual adaptation for real-world dexterous vla models. *arXiv preprint arXiv:2605.30226*, 2026.

Shutong Ding, Zejia Zhong, Zhongyi Wang, Ke Hu, Bikang Pan, Jingya Wang, and Ye Shi. Sample-efficient diffusion-based reinforcement learning with critic guidance. *arXiv preprint arXiv:2605.30056*, 2026.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, volume 2025, pp. 53791–53846, 2025.

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*, 2026a.

Perry Dong, Kuo-Han Hung, Dorsa Sadigh, and Chelsea Finn. Reinforcement learning for real-time vision-language-action policies. *arXiv preprint arXiv:2609.18207*, 2026b.

Perry Dong, Qiyang Li, Dorsa Sadigh, and Chelsea Finn. Expo: Stable reinforcement learning with expressive policies. In *International Conference on Learning Representations*, volume 2026, pp. 131379–131394, 2026c.

Yonghoon Dong, Kyungmin Lee, Changyeon Kim, Jaehyuk Kim, and Jinwoo Shin. Trust region q adjoint matching. *arXiv preprint arXiv:2605.27079*, 2026d.

Abdelghani Ghanem and Mounir Ghogho. Entropy-regularized adjoint matching for offline reinforcement learning. *arXiv preprint arXiv:2605.06156*, 2026.

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)*, pp. 15665–15672. IEEE, 2025.

Zhengyi Guo, Jiayuan Sheng, David D Yao, and Wenpin Tang. Improved techniques for fine-tuning flow models via adjoint matching: a deterministic control pipeline. *arXiv preprint arXiv:2605.06583*, 2026.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 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. IEEE, 2019.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Qiyang Li and Sergey Levine. Q-learning with adjoint matching. *arXiv preprint arXiv:2601.14234*, 2026.

Zeyi Liu, Guangyao Liu, Yinuo Qu, Yuquan Xue, Bofang Jia, Chunhua Yang, Weihua Gui, Keke Huang, and Ziwei Wang. Preference-calibrated human-in-the-loop reinforcement learning for robotic manipulation. *arXiv preprint arXiv:2606.03949*, 2026.

Tianyi Lu, Hui Zhang, Zijie Diao, Junke Wang, Shengqi Xu, Xingyao Lin, Guojin Zhong, Ziyi Ye, Peng Wang, Zuxuan Wu, et al. Thinkingvla: Interleaved vision and language reasoning for robotic manipulation. *arXiv preprint arXiv:2606.17937*, 2026.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025.

Mingjie Pan, Siyuan Feng, Qinglin Zhang, Xinchen Li, Jianheng Song, Chendi Qu, Yi Wang, Chuankang Li, Ziyu Xiong, Zhi Chen, et al. Sop: A scalable online post-training system for vision-language-action models. *arXiv preprint arXiv:2601.03044*, 2026.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. *arXiv preprint arXiv:2502.02538*, 2025.

Kexin Shi, Junyao Shi, Poorvi Hebbar, Zhuoluo Zhao, Tarun Amarnath, Yifan Su, Shikhar Bahl, and Deepak Pathak. Flowdpg: Deterministic policy gradient on flow matching policies for real-world manipulation. In *RSS 2026 Workshop on Diffusion for Robot Learning*, 2026.

Chenyu Su, Zhaolong Shen, Yuan Qian, Chen Qian, Rui Zhang, Feng Yan, Weixing Chen, Fei Zhang, Jiamin Wang, Shuang Cong, et al. Vla-precision: Asymmetric co-bootstrapping for efficient real-world online rl of vision-language-action models. *arXiv preprint arXiv:2609.04355*, 2026.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.

Yi Wang, Xinchen Li, Pengwei Xie, Pu Yang, Buqing Nie, Yunuo Cai, Qinglin Zhang, Chendi Qu, Jeffrey Wu, Jianheng Song, et al. Learning while deploying: Fleet-scale reinforcement learning for generalist robot policies. *arXiv preprint arXiv:2605.00416*, 2026a.

Ziqian Wang, Yitian Liu, Xingjian Mao, Minqian Wang, and Yao Mu. Q-vgm: Q-value-gradient matching for off-policy reinforcement learning of flow-matching vla. *arXiv preprint arXiv:2606.08015*, 2026b.

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.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025.

Rushuai Yang, Hecheng Wang, Zhichao Wu, Chiming Liu, Xiaohan Yan, Xuan Du, Shuoyu Yue, Chuheng Zhang, Yunlong Wang, Yongcheng Liu, et al. Aloe: Action-level off-policy evaluation for vision-language-action model post-training. *arXiv preprint arXiv:2602.12691*, 2026.

Raymond Yu, William Huey, Mustafa Mukadam, Anusha Nagabandi, and Abhishek Gupta. Support-constrained rl enables real-world policy improvement without real-world experience. *arXiv preprint arXiv:2606.27475*, 2026.

Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Zhang Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. In *International Conference on Learning Representations*, volume 2025, pp. 28129–28164, 2025.

Yixian Zhang, Shu-ang Yu, Tonghe Zhang, Mo Guang, Haojia Hui, Kaiwen Long, Yu Wang, Chao Yu, and Wenbo Ding. Sac flow: Sample-efficient reinforcement learning of flow-based policies via velocity-reparameterized sequential modeling. In *International Conference on Learning Representations*, volume 2026, pp. 121983–122016, 2026.

Yinuo Zhao, Huiqian Jin, Lechun Jiang, Xinyi Zhang, Kun Wu, Pei Ren, Zhiyuan Xu, Zhengping Che, Lei Sun, Dapeng Wu, et al. Real-world reinforcement learning from suboptimal interventions. *arXiv preprint arXiv:2512.24288*, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.

Zhiyuan Zhou, Andy Peng, Charles Xu, Qiyang Li, Tobias Springenberg, Kevin Frans, and Sergey Levine. Test-time gradient guidance of flow policies in reinforcement learning. *arXiv preprint arXiv:2606.11087*, 2026.

A THEORETICAL ANALYSIS OF CVL AND APO

This appendix analyzes the two learning stages of CAPO: evaluating robot chunks from human-assisted replay, and updating the VLA action expert using that critic. The conditioning input s contains the visual observations, language instruction, and proprioception. It is held fixed during one action generation; x_τ is an intermediate *action chunk*, not a robot state. Thus, flow time $\tau \in [0, 1]$ must be distinguished from the H environment steps whose return the critic predicts.

A.1 EXECUTED CHUNKS AND AUTONOMOUS VLA CONTINUATION

Fix the current VLA $\pi = \pi_\theta$ during an CVL critic phase. In Eq. (2), $Q^\pi(s, A)$ evaluates executing the recorded chunk A and then returning control to π; it does not evaluate continued human assistance. Assume bounded rewards, $0 \le \gamma < 1$, a Markov-sufficient context, and the committed action semantics in Appendix B.1. Conditioned on (s, A), executed replay must preserve the environment law of (R, s', d). Its marginal distribution over contexts and actions may mix historical VLAs, demonstrations, and human corrections. Filtering TD transitions by their future success need not preserve this conditional law.

Lemma 1 (CVL targets autonomous continuation)**.** *If the target-head mean in Eq. (3) equals Q^π, then, for both autonomous and human-executed chunks,*

$$\mathbb{E}[y_t - Q^\pi(s_t, A_t) \mid s_t, A_t] = 0. \tag{12}$$

More generally, if $|\bar{Q} - Q^\pi| \le \epsilon_{\text{tar}}$ on the continuation state–action support, the absolute conditional mean residual is at most $\gamma^H \epsilon_{\text{tar}}$. These statements also hold for an unbiased finite-sample estimate using fresh continuation actions from π.

Proof. Condition on the executed (s_t, A_t) and average first over the fresh $A' \sim \pi(\cdot \mid s_{t+1})$, then over the recorded reward, successor, and termination. Subtracting Eq. (2) gives

$$\begin{aligned} &\mathbb{E}[y_t - Q^\pi(s_t, A_t) \mid s_t, A_t] \\ &\quad = \gamma^H \mathbb{E}\left[(1 - d_t)\mathbb{E}_{A' \sim \pi(\cdot|s_{t+1})}(\bar{Q} - Q^\pi)(s_{t+1}, A') \,\middle|\, s_t, A_t\right]. \end{aligned} \tag{13}$$

The right-hand side is zero at exact targets, and its absolute value is at most $\gamma^H \epsilon_{\text{tar}}$ otherwise. Unbiased action sampling preserves this expectation. □

This is the action-chunk policy-evaluation interpretation used in Section 4.2, following (Yang et al., 2026). The actual next replay action need not come from π: it is not used in the bootstrap. Conversely, when a correction overwrites a proposal, this argument applies only to the executed correction. The rejected chunk has no observed (R, s', d) and therefore receives comparison supervision, not a fictitious TD transition.

A.2 COMPATIBILITY OF CVL WITH THE AUTONOMOUS CRITIC

For a retained pair i, define its *autonomous-continuation* value gap $\Delta_i^\pi = Q^\pi(s_i^I, A_i^H) - Q^\pi(s_i^I, A_i^\pi)$. The relevant preference assumption is

$$\Delta_i^\pi \ge m_i \quad \text{almost surely over retained pairs.} \tag{14}$$

Both chunks are evaluated at the same decision context, over the same executed horizon, and with the same VLA continuation. This is a local assumption about accepted corrections, not global optimality of the human. For example, a correction that succeeds only if the operator continues to assist need not satisfy Eq. (14). A takeover alone cannot establish it.

Proposition 2 (CVL has an autonomous-value-compatible fixed point)**.** *Assume Lemma 1, Eq. (14), representability of Q^π by all heads, and finite differentiable expectations. If every online and target head equals Q^π, both the expected TD semi-gradient and the ranking gradient in Eq. (6) vanish, for any $\lambda_{\text{rank}} \ge 0$.*

Proof. Let ϕ collect all online critic parameters, including any shared backbone. For each head, the conditional TD contribution is

$$\mathbb{E}\left[\frac{2}{K}(Q_{\phi_k}(s_t, A_t) - y_t)\nabla_\phi Q_{\phi_k}(s_t, A_t) \,\middle|\, s_t, A_t\right] = 0,$$

because the online derivative is fixed at the conditioned input and the residual has zero conditional mean. For each preference pair, set $\xi_{ik} = [m_i + Q_{\phi_k}(s_i^I, A_i^\pi) - Q_{\phi_k}(s_i^I, A_i^H)]_+$. Its ranking contribution is

$$\frac{2}{K}\xi_{ik}\nabla_\phi\big(Q_{\phi_k}(s_i^I, A_i^\pi) - Q_{\phi_k}(s_i^I, A_i^H)\big).$$

At Q^π, Eq. (14) makes $\xi_{ik} = 0$; the squared hinge also has zero derivative at its boundary. Averaging pairs and summing heads proves the claim, including with shared parameters. □

A.3 From corrective comparisons to terminal action guidance

APO uses $\nabla_A \underline{Q}_\phi$ at completed VLA chunks, so value ordering alone is not the whole requirement. The following local statement describes what CVL's squared-hinge loss can imply about that gradient. Let $\Delta A_i = A_i^H - A_i^\pi$, measured in the normalized action coordinates used by the critic, and retain ξ_{ik} from the preceding proof.

Proposition 3 (Correction-directed terminal gradients)**.** *Suppose each head has an L_Q-Lipschitz action gradient on the straight segment from A_i^π to A_i^H, and the minimizing head k_i^* is unique at (s_i^I, A_i^π). Then*

$$\left\langle \nabla_A \underline{Q}_\phi(s_i^I, A_i^\pi), \Delta A_i \right\rangle \geq m_i - \xi_{ik_i^*} - \frac{L_Q}{2}\|\Delta A_i\|_2^2. \tag{15}$$

Consequently, under a uniform L_Q and finite expectations over the same pair distribution as Eq. (5),

$$\mathbb{E}_i\left[\left\langle \nabla_A \underline{Q}_\phi(s_i^I, A_i^\pi), \Delta A_i \right\rangle\right] \geq \mathbb{E}_i[m_i] - \sqrt{K\mathcal{L}_{\text{rank}}} - \frac{L_Q}{2}\mathbb{E}_i\|\Delta A_i\|_2^2. \tag{16}$$

Proof. Fix the head k_i^* active at the rejected chunk. Smoothness of that head on the segment gives

$$Q_{\phi_{k_i^*}}(s_i^I, A_i^H) - Q_{\phi_{k_i^*}}(s_i^I, A_i^\pi) \leq \left\langle \nabla_A Q_{\phi_{k_i^*}}(s_i^I, A_i^\pi), \Delta A_i \right\rangle + \frac{L_Q}{2}\|\Delta A_i\|_2^2.$$

The left-hand side is at least $m_i - \xi_{ik_i^*}$ by the definition of the hinge residual. Uniqueness of the active head identifies its gradient with $\nabla_A \underline{Q}_\phi$ at A_i^π, proving Eq. (15). Finally, $\mathbb{E}_i \xi_{ik_i^*} \leq (\mathbb{E}_i \sum_k \xi_{ik}^2)^{1/2} = \sqrt{K\mathcal{L}_{\text{rank}}}$. □

When the right-hand side of Eq. (15) is positive, the learned terminal gradient has a positive projection toward the paired correction. This links CVL to the action gradient initializing APO, rather than to imitation of the whole human trajectory. The statement is local to aligned pairs: it does not identify the true Q^π gradient, apply to every new VLA proposal, or guarantee a feasible finite robot motion along the interpolating segment. Those distinctions motivate the separate value and gradient diagnostics in Section 5.3.

A.4 APO as an update of the native VLA action expert

Fix one actor phase: the CVL critic $q(s, A) = \underline{Q}_\phi(s, A)$, the copied VLA reference v_β, and the replay-context distribution μ are all held fixed. Define

$$\mathcal{J}_\mu(\theta) = \mathbb{E}_{s\sim\mu}\mathcal{J}_s(u_\theta), \qquad u_\theta = v_\theta - v_\beta, \qquad r_\tau = u_\theta(x_\tau, \tau \mid s) + \eta c_\tau, \tag{17}$$

where x_τ is generated by the current VLA and c_τ is exactly the lean adjoint in Eq. (8) on that same path. Expectations below include $s \sim \mu$ and $x_0 \sim \mathcal{N}(0, I)$. This objective concerns critic-guided chunk generation at replay contexts; it does not differentiate the robot's visitation distribution or its environment dynamics.

Assume the action-generation ODE is well posed, the fields are continuously differentiable in the variables used below, and differentiation and expectation can be interchanged. For $q = \min_k Q_{\phi_k}$, the classical argument requires a locally unique minimizing head at almost every sampled endpoint and integrable derivatives. Hard-min ties without a common derivative are outside this smooth statement. If the generated chunk is projected to an executed prefix $A = Px_1$, replace the terminal gradient below by $-\alpha_Q P^\top \nabla_A q(s, Px_1)$, as in Appendix B.1.

Why the lean adjoint is not the full feedback derivative. The quadratic Hamiltonian corresponding to Eq. (7) is $\mathcal{H} = \|u\|_2^2/(2\eta) + p^\top(v_\beta + u)$, with pointwise control condition $u/\eta + p = 0$. However, the VLA parameterizes a feedback field $u_\theta(x, \tau \mid s)$, not independently chosen controls at each flow state. Its exact feedback costate p satisfies

$$\dot{p}_\tau = -(\nabla_x v_\beta)^\top p_\tau - (\nabla_x u_\theta)^\top (p_\tau + u_\theta/\eta), \qquad p_1 = -\alpha_Q \nabla_A q(s, x_1). \tag{18}$$

All fields are evaluated on the current VLA path. The lean adjoint c omits the second term and generally differs from p. This specializes the deterministic adjoint construction of (Guo et al., 2026) to the frozen-CVL-critic, phase-wise VLA update used here.

Lemma 4 (Exact action-expert parameter gradient)**.** *Let $D_\theta u_\theta$ denote the local parameter derivative with x_τ, s, and β held fixed. Under the preceding assumptions,*

$$\nabla_\theta \mathcal{J}_\mu = \mathbb{E} \int_0^1 [D_\theta u_\theta(x_\tau, \tau \mid s)]^\top (u_\theta(x_\tau, \tau \mid s)/\eta + p_\tau)\, \mathrm{d}\tau. \tag{19}$$

Proof. For a parameter perturbation h, the path variation z_τ obeys $\dot{z}_\tau = (\nabla_x v_\beta + \nabla_x u_\theta) z_\tau + D_\theta u_\theta h$ and $z_0 = 0$. Differentiating the phase objective gives

$$\delta \mathcal{J}_\mu[h] = \mathbb{E}\left[\int_0^1 \frac{u_\theta^\top}{\eta} (\nabla_x u_\theta z_\tau + D_\theta u_\theta h)\, \mathrm{d}\tau + p_1^\top z_1 \right].$$

Integrating $\mathrm{d}(p_\tau^\top z_\tau)/\mathrm{d}\tau$ and substituting Eq. (18) cancels the terms containing $\nabla_x u_\theta z_\tau$. The remaining expression is $\mathbb{E} \int_0^1 (u_\theta/\eta + p_\tau)^\top D_\theta u_\theta h\, \mathrm{d}\tau$. Since h is arbitrary, Eq. (19) follows. □

B Implementation Details

B.1 Replay and parameter-update boundaries

Algorithm 1 gives the complete training schedule. We maintain three logical replay views for outcome, comparison, and imitation supervision. TD learning uses executed transitions from both $\mathcal{D}_{\text{exec}}$ and $\mathcal{D}_{\text{demo}}$, while $\mathcal{D}_{\text{demo}}$ additionally provides action labels for behavior cloning:

Replay view	Contents	Use
$\mathcal{D}_{\text{exec}}$	Executed transitions, including failures	TD learning; APO contexts
$\mathcal{D}_{\text{inter}}$	Same-state correction–proposal pairs	Per-head preference ranking
$\mathcal{D}_{\text{demo}}$	Demonstration and accepted-correction transitions with imitation labels	TD learning; flow-matching BC

The three views need not be disjoint. An executed corrective action can contribute a TD transition, a correction–proposal pair, and a BC example. Its rejected proposal receives comparative value supervision but is not assigned the reward or successor state of the executed correction. For a valid preference pair, the corrective action and rejected proposal must correspond to the same decision state, execution horizon, and action representation.

The critic action must also match what was actually executed. If only a prefix of a generated action chunk is committed, we use the same projection $A = Px_1$ for TD learning and preference ranking, and initialize the APO terminal adjoint with

$$-\alpha_Q P^\top \nabla_A \underline{Q}_\phi(s, Px_1).$$

Prefix-only imitation labels are masked accordingly. Likewise, variable-duration actions use their recorded execution duration in the discounted target rather than assigning a full-chunk outcome to an interrupted proposal.

Critic updates. The VLA policy is fixed during critic updates, and fresh continuation actions are sampled from the current policy to construct Eq. (3). The target-head mean is used for TD regression, whereas preference ranking is computed independently within each online critic head. Gradients propagate through both the corrective and rejected-action values in each valid pair. If an update contains no valid preference pairs, the ranking term is omitted for that update.

APO updates. During APO updates, critic parameters are frozen and the reference policy is copied from the current VLA at the beginning of each APO phase and held fixed throughout that phase. The minimum over online critic heads provides the terminal value objective used for policy guidance. Although the critic and reference parameters are frozen, their required input derivatives remain available: APO differentiates the critic with respect to the terminal action and evaluates reference vector–Jacobian products with respect to the intermediate flow state. Candidate trajectories and adjoint-derived targets are detached before local velocity matching, so actor parameter gradients pass only through the online velocity evaluations and the independent BC objective.

Current-policy trajectories and APO targets are regenerated after policy updates. BC uses an independently sampled batch of demonstrations and accepted corrections. Replay sampling proportions, preference-pair construction, BC mixture, critic/APO update ratios, and reference-refresh frequency are reported in Table 1.

B.2 Hyperparameters and Training Configuration

Tables 1 and 2 provide a detailed configuration for CAPO.

Table 1: CAPO hyperparameters.

Setting	**Actual value**
Primitive discount γ (chunk discount γ^H)	0.99
Number of critic heads K	2
Policy samples per TD target	1
Target retention ρ in $\bar{\phi} \leftarrow \rho\bar{\phi} + (1-\rho)\phi$	0.995
Preference margin m_i	0.1
Ranking weight λ_{rank}	10
APO control scale η	1.0
Terminal value scale α_Q	1.0
BC weight λ_{BC}	0.5
APO/BC flow-time samples per context	5/1
Critic warm-up steps	200
Replay mixture	40/40/20
Batch sizes	256

Table 2: Stage-specific optimization.

Setting	**SFT**	**CVL critic**	**APO actor**
Optimizer	AdamW	AdamW	AdamW
Learning rate	5×10^{-5}	3×10^{-5}	2×10^{-5}
Betas/momentum; epsilon	$(0.9, 0.95)$; 10^{-8}	$(0.9, 0.95)$; 10^{-8}	$(0.9, 0.95)$; 10^{-8}
Weight decay	10^{-10}	10^{-10}	10^{-10}
Learning-rate schedule	warm-up 10,000 steps, then constant 5×10^{-5}	warm-up 100; cosine to 3×10^{-6} at 20,000 Q steps	warm-up 1,000; cosine to 2.5×10^{-6} at 10,000 actor steps

B.3 Finite-step APO targets

For explicit Euler, let $0 = \tau_0 < \cdots < \tau_L = 1$, $h_\ell = \tau_{\ell+1} - \tau_\ell$, and $\tau_c = \tau_{\ell_c}$. Generate the entire current-VLA path without retaining the solver's parameter graph. With $c_L = -\alpha_Q \nabla_A \underline{Q}_\phi(s, x_L)$,

$$x_{\ell+1} = x_\ell + h_\ell v_\theta(x_\ell, \tau_\ell \mid s), \tag{20}$$

$$c_\ell = c_{\ell+1} + h_\ell [\nabla_x v_\beta(x_\ell, \tau_\ell \mid s)]^\top c_{\ell+1}, \tag{21}$$

$$v_\ell^{\text{tar}} = v_\beta(x_\ell, \tau_\ell \mid s) - \eta c_{\ell+1}. \tag{22}$$

The costate is indexed at $\ell + 1$ because interval ℓ changes $x_{\ell+1}$. Indeed, for $\mathcal{J}_\mu^h = \mathbb{E}[\sum_\ell h_\ell \|u_\ell\|^2/(2\eta) - \alpha_Q q(s, x_L)]$, the exact feedback recursion and parameter derivative are

$$p_\ell = p_{\ell+1} + h_\ell(\nabla_x v_{\beta,\ell})^\top p_{\ell+1} + h_\ell(\nabla_x u_\ell)^\top (p_{\ell+1} + u_\ell/\eta), \tag{23}$$

$$\nabla_\theta \mathcal{J}_\mu^h = \mathbb{E} \sum_\ell h_\ell (D_\theta u_\ell)^\top (u_\ell/\eta + p_{\ell+1}). \tag{24}$$

These follow by differentiating each stage cost and Euler transition. Setting $u_\ell = -\eta c_{\ell+1}$ makes $p_\ell = c_\ell$ by backward induction and the parameter derivative zero. At a freshly copied reference, the full-grid detached matching gradient equals $2\eta \nabla_\theta \mathcal{J}_\mu^h$ before adding BC.

Reference VJPs are computed on detached states without forming dense Jacobians. On a uniform 10-interval grid, cutoffs 0.25 and 0.75 are off-grid: their actual rounding or partial-interval treatment must be included rather than silently changing the forward sampler.

Algorithm 1: Alternating CVL and APO

1. **Initialize.** Pretrained π_θ, critics $\{Q_{\phi_k}\}_{k=1}^K$, targets $\bar{\phi}_k \leftarrow \phi_k$, demonstration buffers $\mathcal{D}_{\text{exec}}$ and $\mathcal{D}_{\text{demo}}$, and $\mathcal{D}_{\text{inter}} \leftarrow \emptyset$.
2. **Collect.** Execute π_θ with human interventions. Store executed transitions in $\mathcal{D}_{\text{exec}}$, aligned corrective–rejected pairs in $\mathcal{D}_{\text{inter}}$, and accepted corrections in $\mathcal{D}_{\text{demo}}$.
3. **Critic phase (CVL).** Fix π_θ. At each critic step:
   (a) Sample transitions from $\mathcal{D}_{\text{exec}}$ and pairs from $\mathcal{D}_{\text{inter}}$.
   (b) Construct detached TD targets with fresh current-policy actions (Eq. (3)).
   (c) Minimize $\mathcal{L}_{\text{CVL}} = \mathcal{L}_{\text{TD}} + \lambda_{\text{rank}} \mathcal{L}_{\text{rank}}$.
   (d) Update $\bar{\phi}_k \leftarrow \rho \bar{\phi}_k + (1-\rho)\phi_k$.
4. **Actor phase (APO).** Freeze critics; set $\beta \leftarrow \theta$ and keep it fixed. At each actor step:
   (a) Sample $s \sim \mathcal{D}_{\text{exec}}$ and $x_0 \sim \mathcal{N}(0, I)$; generate a full ODE trajectory with v_θ.
   (b) Integrate backward over $[\tau_c, 1]$:
   $$c_1 = -\alpha_Q \nabla_A \underline{Q}_\phi(s, x_1), \qquad \dot{c}_\tau = -[\nabla_x v_\beta(x_\tau, \tau \mid s)]^\top c_\tau.$$
   (c) Form detached targets $v_\tau^{\text{tar}} = v_\beta(x_\tau, \tau \mid s) - \eta c_\tau$.
   (d) Minimize $\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{APO}} + \lambda_{\text{BC}} \mathcal{L}_{\text{BC}}$, using APO on $[\tau_c, 1]$ and BC from $\mathcal{D}_B$ on $[0, 1]$.
5. **Repeat and deploy.** Repeat Steps 2–4; deploy π_θ with its original full ODE sampler and no critic queries.

B.4 Training-time prefix KV caching

We use batch-local prefix key–value (KV) caching during online RL. After SFT, we freeze the vision–language prefix and update the action expert and its action/time projections. For each sampled microbatch, we encode the conditioning prefix once and detach its layer-wise KV tensors, attention mask, and position offsets. The attention mask prevents prefix tokens from attending to action tokens. We retain the native $\pi_{0.5}$ tokenization and suffix construction; no action-state or flow-time-dependent feature is included in the cache.

The cached prefix is shared by the current and reference action experts for full forward sampling, reference-adjoint computation, and local velocity matching on that microbatch. Only the action-expert parameters are copied at reference refresh, so both experts use the same frozen prefix. At each flow step, action embeddings, time conditioning, and all suffix attention features are recomputed. The independent BC microbatch uses its own prefix cache, as does a successor-context microbatch used for TD targets. Caches are released after the microbatch; no persistent replay-level KV storage is maintained.

Prefix tensors are constants during differentiation, but action-dependent gradient paths remain intact. In particular, APO computes the terminal critic gradient $\nabla_A \underline{Q}_\phi(s, A)$ and reference vector–Jacobian

products $[\nabla_x v_\beta(x_\tau, \tau \mid s)]^\top c$. The candidate trajectory and matching targets are detached, while local velocity evaluations and BC retain gradients to the online action expert. The critic is trained with its own encoder and does not update the frozen VLA prefix.

Controlled actor comparisons use the same prefix-freezing boundary, cache scope, and precision. Actor-update timing includes prefix construction and the complete update computation. Prefix caching removes repeated context encoding within an update; terminal-interval matching reduces reference-adjoint computation. Neither changes the full forward sampling grid.

C Tasks and Experimental Protocols

C.1 Tasks, initialization, and human-assisted collection

Table 3 retains the task goals and active-collection budgets shown in the source figures. The setup lists Franka FR3/Rizon 4 robots, D405/D435 cameras, and SpaceMouse intervention. Each task uses 30–50 demonstrations.

Table 3: Task goals, reported budgets, and demonstration counts.

Task	Goal	Minutes	Demos
USB Insertion	Insert USB into its port.	80	40
Screw Tightening	Align and tighten the screw.	100	50
Cup Hanging	Hang the cup on the rack.	40	30
Block Stacking	Stack the block onto the rod.	100	40
Peg-in-Hole	Insert the plug into the socket.	60	30
Card Swiping	Swipe the card through the reader.	60	30

C.2 Task success and intervention

An autonomous success is a completed attempt with no within-attempt human action correction, timeout, or safety abort. Demonstrations and resets before an attempt do not make training autonomous.

For primitive steps k in the final collection window W_{rj}, with h_k indicating human control,

$$I_{rj}^{\text{step}} = \frac{\sum_{k \in W_{rj}} h_k}{|W_{rj}|}. \tag{25}$$

C.3 Testing whether IVL provides useful guidance

We evaluate whether IVL learns informative value guidance through three complementary analyses: value discrimination, action-gradient utility, and the contribution of preference supervision beyond corrective imitation. All evaluations use a frozen VLA policy and shared replay data to isolate the effect of critic learning.

Value discrimination and calibration. We compare a TD-only critic with CVL using held-out corrective–rejected pairs collected from intervention states. For each pair, the critic is evaluated on the corrective action and the rejected policy proposal at the same state. Preference accuracy is computed as

$$\text{Acc}_r = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{q_{\text{eval}}(s_i, A_i^H) > q_{\text{eval}}(s_i, A_i^\pi)\}, \tag{26}$$

where N is the number of held-out intervention pairs.

We further measure value estimation error using normalized Monte Carlo returns:

$$\text{NMAE} = \frac{1}{NZ} \sum_i |q_{\text{eval}}(s_i, A_i) - \widehat{Q}^\pi(s_i, A_i)|, \tag{27}$$

where Monte Carlo targets are obtained by executing the evaluated action and then continuing with the frozen VLA policy. The rejected proposal is never assigned the return of the corrective action, since it was not executed in the environment. All held-out pairs are split by intervention event to avoid overlap between training and evaluation.

Action-gradient utility. Because APO uses critic action gradients, we evaluate whether these gradients provide useful local improvement directions. For each evaluated action chunk, we compare a small perturbation along the critic-gradient direction with a random perturbation of the same norm. Both perturbed actions are executed under independent VLA continuations, and we measure the return difference:

$$\Delta G_{\nabla - r} = \mathbb{E}[G(A^{\nabla}) - G(A^{r})]. \tag{28}$$

The random perturbation is matched in action space after clipping and projection, ensuring that the comparison measures the direction rather than the perturbation magnitude. A larger $\Delta G_{\nabla - r}$ indicates that the learned critic gradient provides more useful local guidance. This diagnostic evaluates gradient utility rather than merely critic ranking accuracy.

Value guidance beyond corrective BC. We perform a 2×2 ablation over preference ranking and corrective behavior cloning while keeping demonstration BC and APO active. The ranking component is removed by setting $\lambda_{\text{rank}} = 0$, whereas corrective BC is removed by excluding accepted human corrections from the imitation buffer.

This design separates two ways of exploiting interventions: direct imitation of corrected actions and value supervision from correction–proposal comparisons. The comparison tests whether preference supervision provides additional information beyond reproducing accepted corrective actions.

C.4 Testing how APO uses the guidance

Comparison controls. We evaluate APO at both the full-system level and the actor-update level. Full-system comparisons use the same active-collection budget and task evaluation protocol across methods. Actor-level comparisons use the same critic, replay data, action interface, initialization, and behavior-cloning objective whenever possible, allowing the effect of the policy update mechanism to be isolated.

Equal updates versus equal compute. We report two complementary comparisons. The online evaluation compares methods with the same number of actor updates. A separate fixed-replay experiment compares policies under matched actor-update computation: each method receives the same replay data and the same critic guidance, while the number of optimization steps is adjusted according to measured update cost.

The measured actor-update cost includes forward action generation, value-gradient computation, reference vector–Jacobian products, velocity matching, BC computation, and optimizer updates. The reported reduction therefore reflects actor optimization cost rather than complete training time or robot interaction time.

Reference-change diagnostic. To analyze the effect of reference refresh, we evaluate generated action changes across adjacent actor phases. For identical contexts and initial flow noise, we compute

$$D_{p,i} = \frac{\|N_j(A_{p+1}(s_i, z_i)) - N_j(A_p(s_i, z_i))\|_2}{\sqrt{d_j}}, \tag{29}$$

where $N_j(\cdot)$ denotes task-specific action normalization and d_j is the action dimension.

The reported metric is the 95th percentile of action changes over the evaluation set. This measures how the reference strategy affects update magnitude; it is not interpreted as a hard trust region or a safety guarantee.

Matching-interval sweep. We evaluate terminal-interval matching with different cutoff values while keeping the forward generation process unchanged. For each cutoff, APO constructs adjoint targets only within the selected terminal interval, whereas full forward sampling and behavior cloning remain unchanged.

The sweep evaluates the trade-off between actor-update computation and task performance. The full-matching setting ($\tau_c = 0$) serves as the computation baseline, while larger cutoffs reduce backward matching computation.

D Additional Experiments

Due to limited real-world robot interaction budgets, the following diagnostic experiments are conducted with a single run per setting. The reported results are intended to provide qualitative validation of the proposed mechanisms.

D.1 Correction Imitation Baseline

To examine whether the improvement mainly comes from directly imitating human corrections, we compare CAPO with a correction-only behavioral cloning baseline. The baseline uses the same collected human corrective actions as CAPO but removes the corrective value ranking and adjoint-based policy optimization. All methods use identical demonstrations and human-assisted interaction budgets.

Table 4: Correction-only imitation baseline on three representative manipulation tasks.

Method	USB Insertion	Screw Tightening	Block Stacking	Average
SFT	65.0	60.0	50.0	58.3
Correction BC	80.0	70.0	65.0	71.7
CAPO	95.0	90.0	85.0	90.0

The correction-only baseline already improves over SFT, confirming that human interventions provide useful behavioral supervision. However, CAPO further improves autonomous success, suggesting that exploiting intervention data through value learning and policy optimization provides additional benefits beyond direct imitation.

D.2 Validation of Corrective Preference Assumption

CVL assumes that a human correction is locally preferable to the overwritten policy proposal under autonomous continuation. To examine this assumption, we evaluate the learned critic on held-out corrective–rejected pairs and measure whether the learned value ordering agrees with the human intervention preference.

Table 5: Validation of corrective preference ordering.

Task	Preference Agreement (%)	Margin Satisfied (%)	Reversed Pairs (%)
USB Insertion	91	84	9
Screw Tightening	87	79	13
Block Stacking	85	76	15

The results indicate that most human corrections are consistent with the autonomous continuation value assumption, while a non-negligible fraction of interventions violate the assumed ordering. This suggests that future improvements may benefit from confidence-weighted preference learning or filtering unreliable intervention pairs.

D.3 Lean Adjoint Approximation Analysis

APO uses a lean adjoint that removes the policy Jacobian term from the exact feedback costate computation. To quantify the approximation error, we compare the velocity update direction from the lean adjoint with the exact feedback gradient on sampled trajectories.

Table 6: Cosine similarity between lean-adjoint and exact feedback gradients.

Task	Gradient Cosine Similarity
USB Insertion	0.94
Screw Tightening	0.91
Block Stacking	0.89

The high cosine similarity suggests that the lean adjoint preserves the major optimization direction on the tested tasks. The slightly lower alignment on Block Stacking indicates that more complex contact-rich manipulation may introduce larger approximation errors.

D.4 Reference Refresh Interval Ablation

The reference policy in APO controls update magnitude within each actor phase. We evaluate different reference refresh intervals while keeping the total number of actor updates fixed.

Table 7: Reference refresh interval ablation.

Refresh Interval	USB Insertion	Screw Tightening
250 updates	90.0	85.0
500 updates	95.0	90.0
1000 updates	90.0	80.0
2000 updates	85.0	80.0

The results suggest that an intermediate refresh interval provides a balance between excessive policy drift and overly conservative updates.

E Limitations and Future Work

Despite the effectiveness of CAPO in improving real-world VLA adaptation, several limitations remain. First, CAPO currently relies on human corrections as informative preference signals. However, human interventions are not always optimal: as the policy improves during training, the autonomous policy may eventually outperform human operators in certain states, making some human corrections suboptimal or overly conservative. Future work could investigate learning paradigms that explicitly model suboptimal human behaviors, rather than treating all interventions as reliable preferences, enabling the policy to benefit from human experience while avoiding unnecessary imitation of inferior actions.

Second, CAPO mainly exploits corrective experiences by learning values from accepted corrections and rejected proposals. A promising direction is to further explore negative-aware flow tuning methods (NFT) (Zheng et al., 2026), which could directly utilize failed or undesirable action trajectories as informative signals for refining flow-based policies. Such approaches may provide richer supervision beyond pairwise corrective preferences and improve policy refinement efficiency.

Finally, similar to many reinforcement learning methods, CAPO may face challenges in long-horizon tasks where delayed rewards and complex credit assignment make value estimation difficult. Future work could explore hierarchical solutions, such as segment-level Q-learning or policy/value distillation, to decompose long trajectories into more manageable sub-problems and improve scalability to complex multi-stage manipulation tasks.

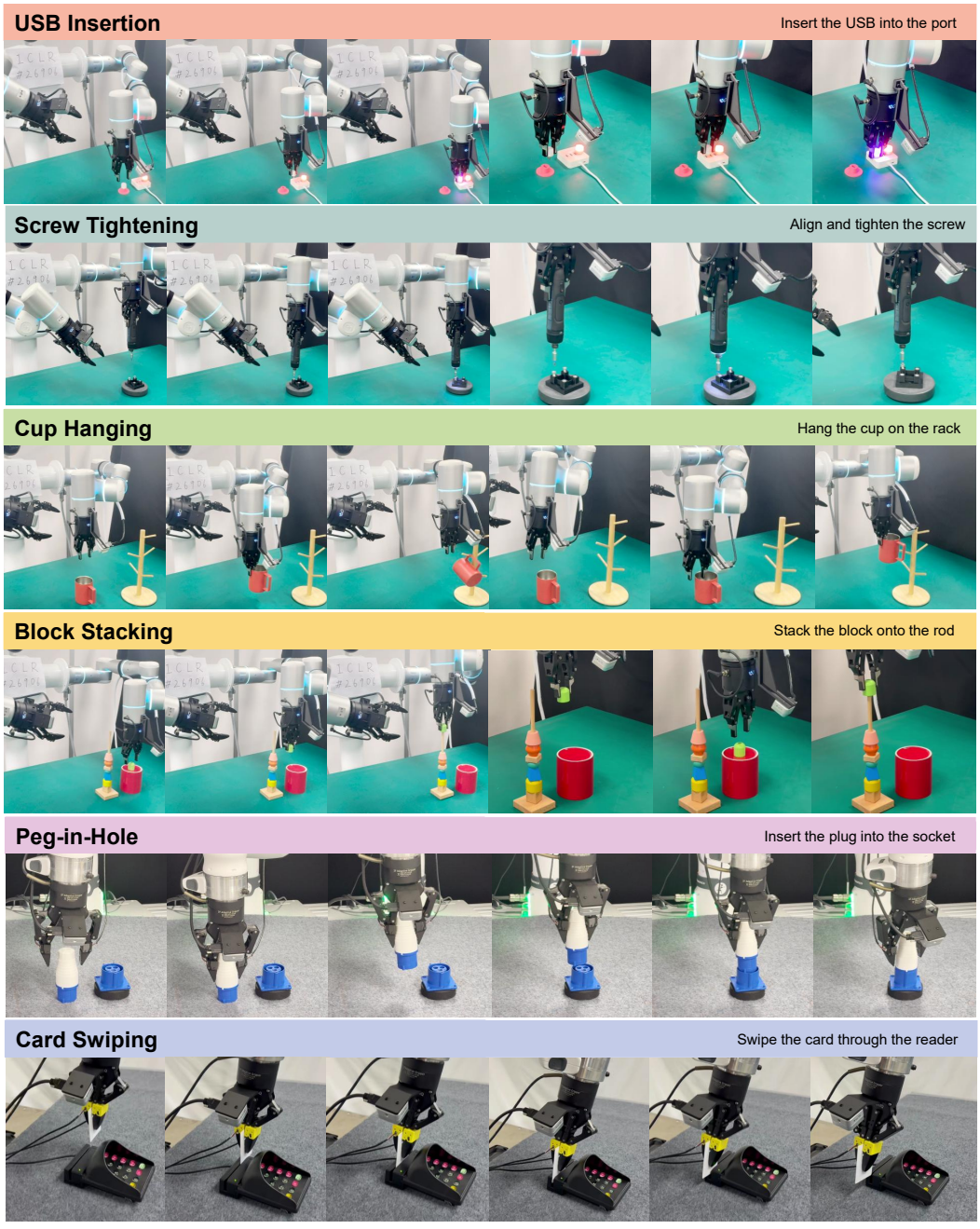


Figure 9: Six real-world tasks.