ROAD-VLA: ROBUST ONLINE ADAPTATION VIA SELF-DISTILLATION FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Online reinforcement learning can adapt vision-language-action (VLA) models to new deployment conditions, but scalar advantages provide coarse supervision for their high-dimensional, structured action predictions. We propose ROAD-VLA, an advantage-guided self-distillation framework that turns rollout feedback into a structured teacher directly in the policy's native action space. After each on-policy rollout, ROAD-VLA takes a proximal mirror-descent step from the rollout prediction, moving toward better-than-expected executed behavior and away from worse-than-expected behavior. To improve reliability under distribution shift, a frozen reference critic is incorporated only when its advantage estimate agrees in sign with the online critic. This yields a unified construction for discrete action tokens, continuous action chunks, and flow-matching velocity predictions, without requiring an external expert. We derive a conditional policy-improvement bound under explicit calibration and stability assumptions, providing a common theoretical interpretation across all three action representations. Across ManiSkill environments spanning visual, compositional, and execution shifts, ROAD-VLA improves OpenVLA-7B and OpenVLA-OFT over PPO and outperforms language-guided and external-teacher distillation baselines. We further demonstrate the same mechanism with the flow-matching $\pi_{0.5}$ policy on LIBERO. Across these settings, ROAD-VLA enables faster online adaptation while maintaining higher policy entropy and lower late-stage variance than PPO.

1 INTRODUCTION

Vision-language-action (VLA) models have emerged as a promising paradigm for general-purpose robotic manipulation, mapping visual observations and language instructions directly to low-level actions through large-scale pretraining on diverse demonstrations (Firoozi et al., 2025; Ma et al., 2026). Models such as RT-2 (Zitkovich et al., 2023a), OpenVLA (Kim et al., 2024), and π_0 (Black et al., 2024) have demonstrated broad generalization across tasks and environments, while more recent systems including $\pi_{0.5}$ (Black et al., 2025) and GR00T N1 (Bjorck et al., 2025) extend this capability across increasingly diverse tasks, environments, and robot embodiments. Yet reliable deployment remains challenging: robots inevitably encounter visual changes, novel object configurations, sensor noise, and execution disturbances that differ from pretraining. Effective post-training must therefore adapt a pretrained policy through interaction while preserving its general capabilities.

Reinforcement learning (RL) provides a natural framework for this setting because it can adapt policies from task feedback without additional expert demonstrations. Recent work has explored PPO (Schulman et al., 2017), DPO (Rafailov et al., 2023; Zhang et al., 2024b), and GRPO (Guo et al., 2025) for adapting large pretrained models, and exploration can improve policies beyond the support of their original demonstrations (Liu et al., 2026b). However, there is a mismatch between how RL assigns credit and how VLAs represent actions. Policy-gradient methods typically reduce rollout feedback to a scalar advantage, whereas a VLA may predict multiple action tokens (Kim et al., 2024), a continuous action chunk (Kim et al., 2025), or a flow field (Black et al., 2025). A single scalar therefore provides only coarse supervision for a high-dimensional structured prediction. For example, for a 7-DoF arm producing a 16-step action chunk, the same scalar advantage weights the entire chunk without indicating which joints or temporal components most contributed to success

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

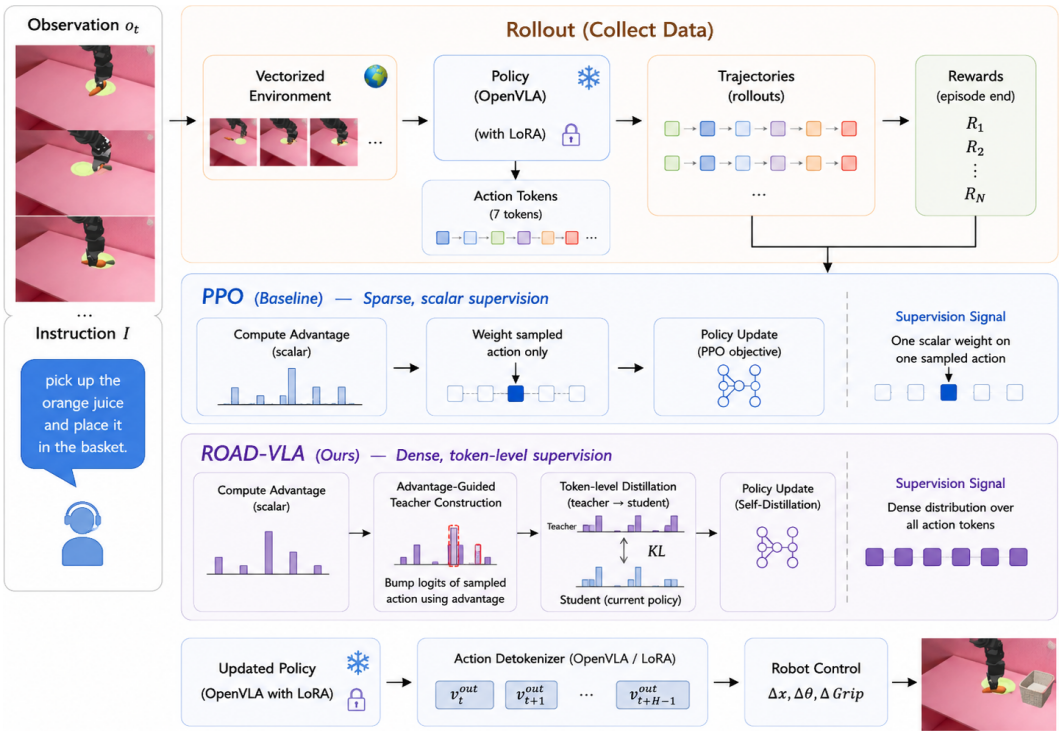


Figure 1: Overview of ROAD-VLA, illustrated with OpenVLA for clarity. PPO uses the scalar advantage to weight the sampled action, whereas ROAD-VLA converts the agreement-gated advantage into a proximal teacher and distills the student toward it, yielding dense supervision over the action tokens. The same formulation extends to continuous action chunks and flow-matching velocity predictions without requiring an external expert.

or failure. The key challenge is thus to convert coarse outcome feedback into structured supervision in the policy's action-prediction space.

Self-distillation (Hinton et al., 2015) offers a natural way to provide such supervision. Recent work on LLM post-training shows that a privileged version of the model can serve as a teacher, converting outcome-level feedback into denser step-wise targets (Agarwal et al., 2024; Zhao et al., 2026; Penaloza et al., 2026). For VLA adaptation, however, constructing a useful teacher is non-trivial. Privileged demonstrations, retrieved successful actions, or high-level textual plans can describe desirable behavior, but they do not directly specify how the policy's native action prediction should change. This gap is particularly important for low-level control, where language-level guidance may not translate reliably into precise action updates (Kim et al., 2024; Brohan et al., 2022; Zhang et al., 2024a). An effective teacher should therefore provide supervision directly in the policy's native action space.

We propose ROAD-VLA (Figure 1), which turns the scalar rollout advantage into a structured teacher in the policy's native action space. After each on-policy rollout, ROAD-VLA uses the executed action and its signed advantage to move the rollout prediction toward better-than-expected behavior and away from worse-than-expected behavior. A proximal penalty keeps this teacher close to the rollout policy, yielding a single mirror-descent-style step whose geometry matches the action head. The construction therefore applies naturally to discrete action tokens through a KL-based logit shift, continuous action chunks through a Euclidean update, and flow-matching policies through a velocity-space update. To improve robustness under distribution shift, we combine the online and frozen reference critics only when their advantage estimates agree in sign; otherwise, we fall back to the online critic. The resulting teacher converts scalar rollout feedback into dense supervision over the full action prediction without an external expert.

ROAD-VLA augments the standard PPO objective with this native-space distillation loss. PPO retains its usual likelihood-based policy update, while self-distillation provides a structured target derived from the same rollout feedback; setting the distillation weight to zero recovers PPO exactly. We further derive a conditional policy-improvement lower bound under explicit teacher-calibration, distillation-stability, and bounded-advantage conditions. The analysis characterizes how advantage-calibrated teacher displacement, student–teacher mismatch, and policy distribution shift jointly affect improvement, providing a common interpretation across discrete, continuous, and flow-matching action heads.

We evaluate ROAD-VLA on seven ManiSkill (Mu et al., 2021) manipulation environments spanning three axes of distribution shift: visual robustness (unseen backgrounds, dynamic textures, and sensor noise), compositional reasoning (disambiguation among unseen distractors), and execution robustness (geometry shifts and mid-episode state perturbations). With OpenVLA-7B, ROAD-VLA improves average success over PPO in both in-distribution and out-of-distribution settings, and outperforms both language-privileged teachers and VLA-OPD (Zhong et al., 2026), an on-policy distillation method that relies on an external expert. Training analyses show faster online adaptation, sustained policy entropy, and lower late-stage variance than PPO. To test whether the mechanism depends on tokenized actions, we further evaluate continuous OpenVLA-OFT and the flow-matching $\pi_{0.5}$ policy, with the latter tested on LIBERO (Liu et al., 2023).

Our contributions are threefold: (i) we introduce ROAD-VLA, an advantage-guided self-distillation framework that constructs a proximal teacher directly in the policy's native action space, yielding a unified objective for tokenized, continuous, and flow-matching action heads without external teachers or additional demonstrations; (ii) we derive a conditional policy-improvement bound that characterizes when native-space teacher matching translates into improved policy performance; and (iii) we validate ROAD-VLA across diverse distribution shifts, action heads, and benchmarks using OpenVLA-7B, OpenVLA-OFT, and $\pi_{0.5}$.

2 Related Work

Vision–Language–Action Models. Vision–language–action (VLA) models unify visual perception, language understanding, and robot control, with architectures spanning discrete action tokens (Brohan et al., 2022; Zitkovich et al., 2023b; O'Neill et al., 2024; Kim et al., 2024), continuous action chunks (Kim et al., 2025), and diffusion- or flow-based action representations (Octo Model Team et al., 2024; Black et al., 2024; 2025; Bjorck et al., 2025). Despite strong pretrained capabilities, these policies remain vulnerable to deployment distribution shifts. We study online adaptation across these heterogeneous action representations using a unified advantage-guided self-distillation framework.

Online RL for VLA Adaptation. Online RL enables adaptation from environment interaction without additional expert demonstrations. PPO has recently shown strong performance for VLA adaptation (Schulman et al., 2017; Liu et al., 2026b), while advantage-weighted methods such as AWR, AWAC, and CRR (Peng et al., 2019; Nair et al., 2020; Wang et al., 2020) use scalar advantages to emphasize better actions. However, such signals remain coarse for structured VLA outputs. ROAD-VLA instead converts the advantage into a bounded proximal teacher directly in the policy's native action space.

Distillation and Privileged Guidance. Distillation has been widely used for transferring predictive and policy knowledge (Hinton et al., 2015; Rusu et al., 2015; Li & Hoiem, 2016), and recent self-distillation methods use privileged or on-policy teachers to densify sparse feedback (Shenfeld et al., 2026; Zhao et al., 2026). In robotics, privileged language or vision-language guidance can provide textual feedback or reward shaping (Ma et al., 2024; Liang et al., 2023; Yu et al., 2023), while VLA-specific methods such as ActDistill (Ye et al., 2025) and VLA-OPD (Zhong et al., 2026) use action-space teachers. ROAD-VLA differs by constructing its teacher directly from the current policy and its advantage estimates, without privileged language or an external expert.

3 BACKGROUND

3.1 VISION–LANGUAGE–ACTION MODELS

We consider pretrained vision–language–action (VLA) policies that map an observation history $o_{t-H+1:t}$ and language instruction l to robot actions. Modern VLAs differ primarily in how they represent and predict actions. Autoregressive models such as OpenVLA (Kim et al., 2024) discretize continuous actions into tokens and predict them with next-token classification. Continuous models such as OpenVLA-OFT (Kim et al., 2025) and Octo (Octo Model Team et al., 2024) directly regress action chunks, while flow-matching models such as π_0 (Black et al., 2024) and $\pi_{0.5}$ (Black et al., 2025) generate actions through a learned velocity field. ROAD-VLA operates in each policy's native prediction space and is therefore agnostic to the specific action parameterization.

3.2 ONLINE VLA ADAPTATION

We study online adaptation of a pretrained VLA policy π_{θ_0} to a target environment $\mathcal{M}^{\mathrm{tar}}$, modeled as a language-conditioned POMDP $\mathcal{M}^{\mathrm{tar}} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \mathcal{L}, p(s_0), \gamma)$. Let

$$h_t = (o_{t-H+1:t}, l)$$

denote the policy context at step t. The policy produces $a_t \sim \pi_\theta(\cdot \mid h_t)$, where the underlying prediction may be represented by discrete tokens, continuous action chunks, or flow variables. Our goal is to adapt θ_0 through interaction with $\mathcal{M}^{\mathrm{tar}}$ without requiring expert demonstrations.

A standard approach is PPO (Schulman et al., 2017), which optimizes

$$\mathcal{L}_{\mathrm{PPO}}(\theta) = -\mathbb{E}_t\Big[\min\Big(r_t(\theta)\hat{A}_t,\ \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t\Big)\Big], \tag{1}$$

where

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid h_t)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid h_t)}, \tag{2}$$

$\hat{A}_t$ is the estimated advantage, and ϵ is the clipping threshold. Recent work has found PPO to be a strong baseline for online VLA adaptation (Liu et al., 2026b). However, its learning signal remains coarse: sparse outcome feedback is compressed into a scalar advantage, which provides limited guidance for high-dimensional, structured VLA outputs such as action tokens, continuous action chunks, or flow trajectories. This raises a central question: *how can sparse rollout feedback be converted into a denser, more informative target in the policy's native action-prediction space?*

4 METHOD

Figure 1 summarizes ROAD-VLA. Given an on-policy rollout, we use training-time feedback to construct a local teacher from the rollout policy's own prediction. Section 4.1 introduces the self-distillation framework. We then consider text-based privileged guidance (Section 4.2.1) before introducing our advantage-guided teacher in Section 4.2.2. Section 4.3 gives a conditional policy-improvement analysis.

4.1 SELF-DISTILLATION

Policy-gradient methods compress rollout feedback into a scalar advantage, which provides limited guidance on how a structured VLA prediction should change. We instead use rollout feedback to construct a local target from the rollout policy's own prediction and train the policy to match it.

Let $\pi_{\mathrm{old}} = \pi_{\theta_{\mathrm{old}}}$ denote the rollout policy and

$$\hat{\tau} = \big((o_1, \hat{a}_1), \ldots, (o_T, \hat{a}_T)\big) \sim \pi_{\mathrm{old}}(\cdot \mid l).$$

With $h_t = (o_{t-H+1:t}, l)$, let $y_t^{\mathrm{old}} = f_{\theta_{\mathrm{old}}}(h_t)$ denote the policy's native action prediction, e.g., a categorical token distribution, a continuous action chunk, or a flow velocity. Training-time information $\mathcal{I}_t$ derived from the rollout is used to construct a fixed local teacher

$$\tilde{y}_t = \mathcal{T}(y_t^{\mathrm{old}}, \mathcal{I}_t). \tag{3}$$

The student is then optimized to match $\tilde{y}_t$ under a divergence appropriate to the prediction space. The central question is therefore how to choose $\mathcal{I}_t$ and turn it into a useful local teacher.

4.2 Design of Privileged Information

The quality of self-distillation depends on the information used to construct the teacher. A natural choice is privileged guidance about successful behavior, such as retrieved actions, spatial relations, or task plans. We therefore ask whether such information can provide sufficiently precise supervision for correcting VLA action predictions.

4.2.1 Text-guided $\mathcal{I}$

We first exploit the natural-language interface of VLAs and construct privileged information in text form. We consider three variants: ***MCTS PI***, which retrieves visually similar successful trajectories and verbalizes their actions; ***RelSpatial PI***, which provides egocentric spatial relations between the end-effector and task-relevant objects; and ***Plan+RelSpatial PI***, which augments these relations with a step-by-step task plan. In each case, a teacher conditioned on the privileged text supervises a hint-free student through token-level distillation; prompt templates and training details are provided in Appendix D. Although expressive, text guidance introduces a text-to-action bottleneck: the policy must translate qualitative descriptions, coordinates, or plans into precise changes in its action predictions. This motivates constructing privileged supervision directly from rollout outcomes in the policy's native action-prediction space. We evaluate these alternatives empirically in Section 5.3.3.

4.2.2 Advantage-Guided Teacher Construction

From rollout feedback to a local teacher. An on-policy rollout indicates whether an executed action was better or worse than expected. ROAD-VLA uses this feedback to perturb the rollout policy's own prediction toward positively advantaged behavior and away from negatively advantaged behavior, directly in the policy's native prediction space.

Reliable advantage signal. Let $\hat{A}_t^{\text{int}}$ be the online advantage and $\hat{A}_t^{\text{ref}}$ the estimate from a frozen reference critic. We first match their scales and use the reference estimate only when the two critics agree on sign:

$$\tilde{A}_t^{\text{ref}} = \mu_{\text{int}} + \frac{\sigma_{\text{int}}}{\sigma_{\text{ref}} + \varepsilon}(\hat{A}_t^{\text{ref}} - \mu_{\text{ref}}), \qquad \hat{A}_t^{\text{mix}} = \hat{A}_t^{\text{int}} + \alpha g_t(\tilde{A}_t^{\text{ref}} - \hat{A}_t^{\text{int}}), \tag{4}$$

where $g_t = \mathbf{1}[\text{sign}(\hat{A}_t^{\text{int}}) = \text{sign}(\tilde{A}_t^{\text{ref}})]$. We then set

$$\omega_t = \text{clip}(\hat{A}_t^{\text{mix}}, -c, c), \qquad \alpha = 0.5, \quad c = 2.0. \tag{5}$$

We keep ω_t signed and unstandardized so that its zero point remains the critic baseline.

Proximal teacher. Let y_t^{old} be the rollout prediction and let δ_t denote the local direction associated with the executed action or chunk. We define

$$r_t(y) = \eta\omega_t\langle\delta_t, y - y_t^{\text{old}}\rangle, \qquad y_t^\star = \arg\max_{y\in\mathcal{Y}} \left\{r_t(y) - D_\Phi(y, y_t^{\text{old}})\right\}, \tag{6}$$

where D_Φ is a Bregman divergence chosen to match the action head. The first term moves the teacher in the advantage-supported direction, while the second keeps it proximal to the rollout prediction. The resulting closed forms are

$$q_{t,k}^\star = \text{softmax}\left(z_{t,k}^{\text{old}} + \eta\omega_t e_{\hat{a}_{t,k}}\right), \tag{7}$$

$$\mathbf{a}_t^\star = \mathbf{a}_t^{\text{old}} + \eta\omega_t\left(\hat{\mathbf{a}}_t - \mathbf{a}_t^{\text{old}}\right), \tag{8}$$

$$v_t^\star = v_t^{\text{old}} + \eta\omega_t\left(v_t^{\text{FM}} - v_t^{\text{old}}\right), \qquad v_t^{\text{FM}} = \epsilon - \hat{\mathbf{a}}_t, \tag{9}$$

for discrete tokens, continuous regression, and flow matching, respectively. For flow matching, $x_s = s\epsilon + (1-s)\hat{\mathbf{a}}_t$ is the noised executed chunk. Positive ω_t moves the teacher toward the executed behavior and negative ω_t moves it away.

For chunking policies, t indexes one policy decision: the full chunk $\hat{\mathbf{a}}_t \in \mathbb{R}^{Cd_a}$ receives one PPO advantage, and the same ω_t modifies the entire chunk prediction jointly. Exact proximal derivations and implementation details are given in Appendix A.1.

Training objective. The teacher is built once from θ_{old} and held fixed during the student update. Let $\widehat{D}_t$ denote the empirical native divergence: teacher-forced forward KL for autoregressive tokens in the default model, half squared Euclidean distance for regression, and the sampled flow-matching loss for flow heads. Any distillation temperature is part of this empirical divergence and does not change the teacher construction. We optimize

$$\mathcal{L}_{\text{AGD}}(\theta) = \mathbb{E}_{\hat{\tau}}\left[\frac{1}{T}\sum_{t=1}^{T}\widehat{D}_t\left(\text{sg}(y_t^\star), y_t^\theta\right)\right], \qquad \mathcal{L}_{\text{ROAD}} = \mathcal{L}_{\text{PPO}} + \lambda\mathcal{L}_{\text{AGD}}, \quad \lambda = 0.5. \tag{10}$$

PPO uses advantage as a scalar policy-gradient signal, whereas AGD converts the same rollout feedback into a structured local target. Setting $\lambda = 0$ recovers PPO.

4.3 Theoretical Results

Let $D_t(\cdot,\cdot)$ denote the population counterpart of the native divergence used by each head, with rollout and teacher-construction randomness understood in the expectation. Let $\pi_t^\star$ denote the corresponding teacher behavior after marginalizing this construction randomness. Precise head-specific definitions are given in Appendix A.5.

Theorem 1 (Conditional policy-improvement bound). Let

$$Z_\gamma = \sum_{t=1}^{T}\gamma^{t-1}, \qquad w_t = \frac{\gamma^{t-1}}{Z_\gamma}, \qquad J_\gamma(\pi) = \frac{1}{Z_\gamma}\mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^{t-1}\rho_t\right],$$

where γ is the decision-level discount used by PPO, and let $A_t^{\pi_{\text{old}}}$ be the corresponding time-dependent discounted advantage. Assume, for contexts in the support of $d_t^{\pi_{\text{old}}}$,

$$\mathbb{E}_{a\sim\pi_t^\star}\left[A_t^{\pi_{\text{old}}}(h_t,a)\right] \geq \beta D_t(y_t^\star, y_t^{\text{old}}) - \epsilon_{\text{cal}}, \qquad \beta > 0, \tag{11}$$

$$\left|\mathbb{E}_{a\sim\pi_t^\star}[A_t^{\pi_{\text{old}}}(h_t,a)] - \mathbb{E}_{a\sim\pi_{\theta'}}[A_t^{\pi_{\text{old}}}(h_t,a)]\right| \leq L\sqrt{2D_t(y_t^\star, y_t^{\theta'})}. \tag{12}$$

If $|A_t^{\pi_{\text{old}}}(h_t,a)| \leq B_A$, then for any updated student $\pi_{\theta'}$,

$$\begin{aligned} J_\gamma(\pi_{\theta'}) \geq J_\gamma(\pi_{\text{old}}) + \sum_{t=1}^{T} w_t \mathbb{E}_{h_t\sim d_t^{\pi_{\text{old}}}}\left[\beta D_t(y_t^\star, y_t^{\text{old}}) - L\sqrt{2D_t(y_t^\star, y_t^{\theta'})}\right] \\ - \epsilon_{\text{cal}} - \Delta_{\text{occ}}, \end{aligned} \tag{13}$$

where $\Delta_{\text{occ}} = 2B_A\sum_{t=1}^{T} w_t\,\text{TV}\left(d_t^{\pi_{\theta'}}, d_t^{\pi_{\text{old}}}\right)$. The assumptions have a simple interpretation. The first requires the teacher displacement to be aligned, on average, with true rollout-policy improvement; the second requires a student that is close to the teacher in the native prediction space to have similar decoded advantage. The bounded-advantage condition follows from bounded finite-horizon rewards. The theorem is therefore conditional rather than an unconditional improvement guarantee. The full definitions, intuition, and proof are given in Appendix A.5.

5 Experiments

We use OpenVLA-7B as our primary model to evaluate whether advantage-guided online self-distillation improves RL-based adaptation and generalization beyond PPO, and to analyze the mechanisms underlying these gains, including a comparison with privileged language guidance. Extensions to continuous OpenVLA-OFT and flow-matching $\pi_{0.5}$ are reported in Appendix F.1.

5.1 Overall Experimental Setup and Baselines

Testing environments. We evaluate ROAD-VLA on two benchmarks, detailed in Appendix E.1. Our primary benchmark comprises seven ManiSkill3 (Mu et al., 2021) tasks with distribution shifts along three axes: **visual robustness** (VR), **compositional reasoning** (CR), and **execution robustness** (ER). VR perturbs appearance while preserving task semantics and geometry; CR requires

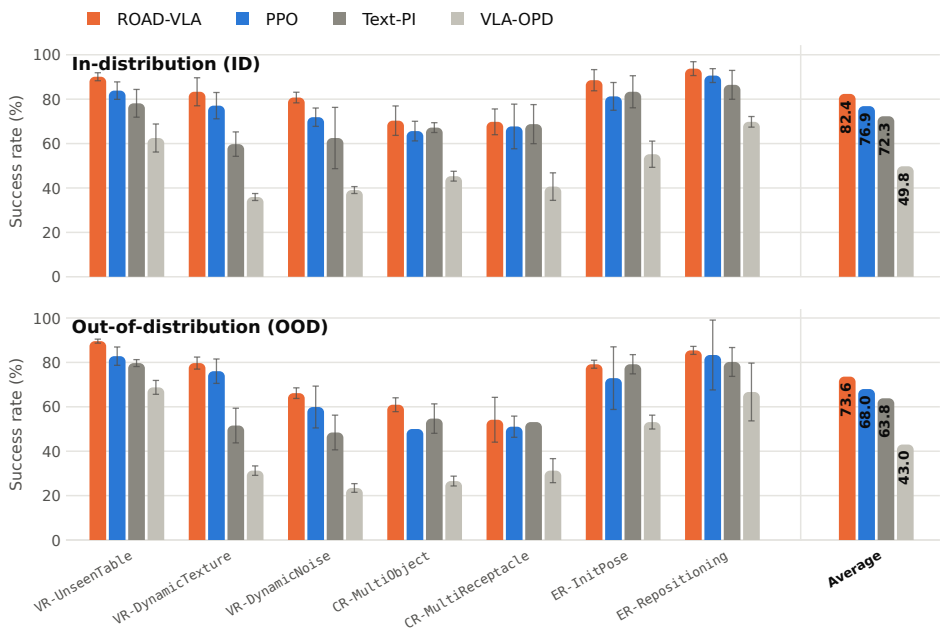


Figure 2: Task success rates (%) for OpenVLA-7B across seven ManiSkill evaluation environments. Results are reported as mean and standard deviation over 3 random seeds. The environments are grouped by visual robustness, compositional reasoning, and execution robustness. Numerical results are provided in Appendix F.2.

identifying the instruction-relevant object under unseen object compositions; and ER perturbs the state at initialization or during execution.

Our main experiments use ManiSkill3 with OpenVLA-7B (Kim et al., 2024). To test generality beyond discrete action tokens, Appendix F.1 extends ROAD-VLA to continuous OpenVLA-OFT (Kim et al., 2025) on ManiSkill and flow-matching $\pi_{0.5}$ (Black et al., 2025) on LIBERO (Liu et al., 2023).

Baselines. Our primary baseline is PPO, a standard approach for on-policy VLA adaptation (Liu et al., 2026b). On the primary benchmark, we further compare against two distillation baselines. *Text-PI* replaces the advantage-guided teacher with privileged language guidance while retaining the same self-distillation framework. We use MCTS PI as its main variant due to its greater stability; alternative text-guided methods, including RelSpatial PI, are analyzed in Section 5.3.3, with details in Appendix D. *VLA-OPD* (Zhong et al., 2026) instead distills from an external expert teacher. All methods start from the same warm-up checkpoint and use identical rollout buffers, rewards, and optimization settings (Appendix E.1.2). We report mean $\pm$ standard deviation over three random seeds.

5.2 Core Evaluation: Robustness to Distribution Shift

Figure 2 summarizes in-distribution (ID) and out-of-distribution (OOD) success across the seven evaluation environments. ROAD-VLA achieves the strongest overall performance, reaching 82.36% ID and 73.59% OOD success. Compared with PPO, ROAD-VLA improves all 14 environment–split comparisons, raising the average from 76.86% to 82.36% ID and from 68.00% to 73.59% OOD, gains of $+5.50$ and $+5.59$ percentage points, respectively. The gains span all three shift categories, with the largest OOD improvement on `CR-MultiObject` ($+10.94$ points). This consistent improvement shows that advantage-guided self-distillation provides a useful learning signal beyond the standard PPO objective, including under distribution shift.

Comparisons with alternative distillation strategies further show that the gains do not follow from distillation alone. Text-PI uses the same self-distillation framework but replaces advantage-guided targets with privileged language guidance. Despite this additional information, it reaches only

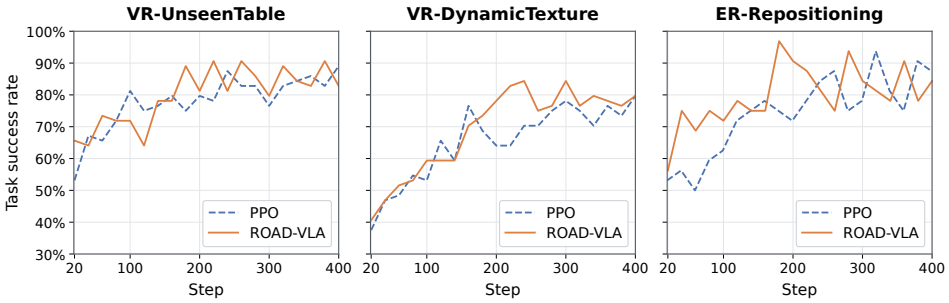


Figure 3: Online OOD adaptation trajectories. ROAD-VLA generally reaches high success earlier than PPO while maintaining competitive or stronger late-stage performance across representative visual and execution shifts.

72.30% ID and 63.84% OOD, below PPO on average. VLA-OPD (Zhong et al., 2026), which instead distills from an external expert teacher, performs substantially lower at 49.77% ID and 43.01% OOD. These results distinguish ROAD-VLA from both language-guided self-distillation and external-teacher distillation: rather than introducing a separate source of supervision, ROAD-VLA converts online rollout feedback into a bounded, state-dependent teacher directly in the policy's action-prediction space. The results therefore highlight *how* the teacher is constructed, from the policy's own prediction and observed advantage, as a key factor in effective online VLA adaptation.

5.3 Understanding ROAD-VLA

5.3.1 Adaptation Dynamics

We first examine how ROAD-VLA changes the online adaptation dynamics relative to PPO. Figure 3 tracks OOD success throughout training on representative distribution shifts. Starting from the same warm-up checkpoint, both methods improve over the course of online adaptation, but ROAD-VLA reaches strong performance earlier and maintains comparable or higher success later in training. The effect is particularly clear on `VR-UnseenTable` and `VR-DynamicTexture`, where ROAD-VLA reaches high success substantially earlier than PPO. Similar behavior is observed on `ER-Repositioning`, despite the different source of distribution shift.

These trajectories suggest that the gain from ROAD-VLA is not limited to a better final policy. By turning rollout feedback into an explicit target in the policy's prediction space, ROAD-VLA provides a more direct adaptation signal alongside the PPO objective, accelerating learning without destabilizing late-stage performance.

5.3.2 How Does Advantage Guidance Shape Optimization?

To better understand the optimization behavior of ROAD-VLA, we track policy update magnitude, the mean advantage weight used for distillation, and agreement between the online and reference critics (Figure 4).

Sustained policy adaptation. Figure 4(a) shows that ROAD-VLA and PPO follow similar early optimization dynamics, with a sharp initial increase in update magnitude followed by gradual stabilization. ROAD-VLA maintains slightly larger updates later in training, indicating that the auxiliary objective continues to provide an adaptation signal even as PPO updates diminish.

Evolving advantage guidance. Figure 4(b) shows that the mean advantage weight shifts from predominantly negative to positive as training progresses. This transition indicates that the distillation signal evolves with the policy rather than imposing a fixed teacher throughout adaptation. Different environments exhibit different transition rates, reflecting the state-dependent nature of the guidance.

Selective reference guidance. Figure 4(c) shows that the online and reference critics maintain substantial, but imperfect, agreement throughout training. ROAD-VLA uses the mixed advantage when their signs agree and falls back to the online estimate otherwise. This allows the reference

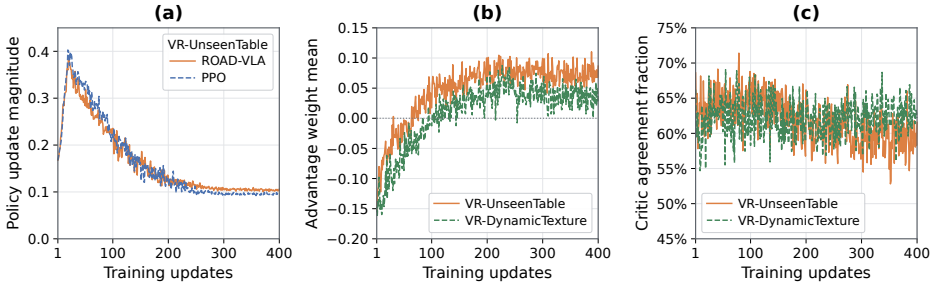


Figure 4: Optimization dynamics of ROAD-VLA. (a) ROAD-VLA maintains slightly larger late-stage policy updates than PPO. (b) Advantage weights evolve from negative to predominantly positive as adaptation progresses. (c) Online and reference critics retain substantial but incomplete agreement, enabling selective use of reference guidance.

critic to contribute when its signal is consistent with the current critic while limiting its influence under disagreement.

5.3.3 Why Not Distill Privileged Language?

Table 1 reveals a clear hierarchy in privileged supervision. Plan- and spatial-based text guidance performs poorly, with the best variant reaching only 32.81% ID and 25.52% OOD; combining both cues further degrades performance. This is consistent with a text-to-action bottleneck: VLA foundation models are largely trained to map relatively simple instructions to actions, making richer privileged descriptions difficult to translate into precise control updates. MCTS PI performs substantially better (78.13% ID, 79.69% OOD) by grounding its guidance in successful action trajectories. Still, direct action-space guidance performs best, reaching 90.10% ID and 89.58% OOD. These results suggest that privileged supervision becomes more effective as it is aligned more directly with the policy's action-prediction space.

Table 1: Ablation of privileged-information (PI) designs on `VR-UnseenTable`. Success rates (%) are reported as mean ± standard deviation over 3 seeds.

Guidance	**ID**	**OOD**
GlobalPlan	32.81 ± 6.81	25.52 ± 5.02
RelSpatial	31.25 ± 2.71	23.44 ± 3.12
GlobalPlan + RelSpatial	18.75 ± 4.69	19.27 ± 5.02
MCTS PI	78.13 ± 6.25	79.69 ± 1.56
Action-guided	$\mathbf{90.10 \pm 1.80}$	$\mathbf{89.58 \pm 0.90}$

6 Conclusion

In this work, we introduced ROAD-VLA, an advantage-guided self-distillation framework for online VLA adaptation. Rather than relying on language-level privileged guidance, ROAD-VLA converts scalar rollout advantages into dense supervision directly in the policy's native action space. This yields a unified adaptation mechanism for discrete tokens, continuous action chunks, and flow-matching policies. Across robotic manipulation benchmarks, ROAD-VLA consistently improves over PPO in both in-distribution and out-of-distribution settings, demonstrating more effective and stable online adaptation.

Several limitations remain. Our experiments focus primarily on simulated pick-and-place tasks, and future work should evaluate ROAD-VLA on physical robots, longer-horizon behaviors, and broader task distributions. The current framework also relies on a frozen reference critic whose reliability may deteriorate under severe distribution shift, motivating critic-free, adaptive, or uncertainty-aware teacher construction.

REPRODUCIBILITY STATEMENT

We provide full experimental details in Appendix E, including model initialization, training and evaluation protocols, environment settings, hyperparameters, baselines, and implementation details for each VLA backbone. Additional derivations and algorithmic details are provided in the appendices where relevant. Code and configuration files accompanying the submission reproduce the training and evaluation pipelines used in our experiments. All comparisons use matched experimental settings, and results are reported over multiple random seeds as specified in the paper. Code is available at `https://anonymous.4open.science/r/ROAD-1726`.

ETHICS STATEMENT

This work studies online adaptation of vision–language–action models for robotic manipulation. Our experiments are conducted in simulated environments using existing models and benchmarks, and involve neither human subjects nor newly collected personal data. As with other learning-based robotic systems, real-world deployment may introduce safety risks when model predictions are unreliable or deployment conditions differ substantially from training. Appropriate validation, monitoring, and task-specific safety mechanisms should therefore be used before physical deployment. We do not identify additional ethical concerns specific to the proposed method beyond those generally associated with large pretrained models and autonomous robotic systems.

AI USE STATEMENT

Generative AI tools were used to support coding and implementation, aid and polish the writing and presentation of the paper, and assist with retrieval and discovery of relevant literature. All technical claims, citations, experimental results, code changes, and final manuscript content were reviewed and verified by the authors.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

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 *Proceedings of The 9th Conference on Robot Learning*, Proceedings of Machine Learning Research. PMLR, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 202, 2023.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, 44(5): 701–739, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pp. 614–629, 2016.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International conference on robotics and automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Changyu Liu, Yiyang Liu, Taowen Wang, Qiao Zhuang, James Chenhao Liang, Wenhao Yang, Renjing Xu, Qifan Wang, Dongfang Liu, and Cheng Han. On-the-fly vla adaptation via test-time reinforcement learning. *arXiv preprint arXiv:2601.06748*, 2026a.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can RL bring to VLA generalization? an empirical study. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026b.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, et al. Eureka: Human-level reward design via coding large language models. In *International conference on learning Representations*, volume 2024, pp. 26516–26560, 2024.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision–language–action models for embodied ai. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

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. *arXiv preprint arXiv:2107.14483*, 2021.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR) Journal*, 2024.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33:7768–7778, 2020.

Wencheng Ye, Tianshi Wang, Lei Zhu, Fengling Li, and Guoli Yang. Actdistill: General action-guided self-derived distillation for efficient vision-language-action models. *arXiv preprint arXiv:2511.18082*, 2025.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis. In *Proceedings of The 7th Conference on Robot Learning*, pp. 374–404, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Hongyin Zhang, Shuo Zhang, Junxi Jin, Qixin Zeng, Runze Li, and Donglin Wang. Robustvla: Robustness-aware reinforcement post-training for vision-language-action models. *arXiv preprint arXiv:2511.01331*, 2025.

Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. Hirt: Enhancing robotic control with hierarchical robot transformers. *arXiv preprint arXiv:2410.05273*, 2024a.

Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Siwei Han, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. Grape: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024b.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Zhide Zhong, Haodong Yan, Junfeng Li, Junjie He, Tianran Zhang, and Haoang Li. Vla-opd: Bridging offline sft and online rl for vision-language-action models via on-policy distillation. *arXiv preprint arXiv:2603.26666*, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023a.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023b.

APPENDIX CONTENTS FOR
ROAD-VLA: Robust Online Adaptation via Self-Distillation for Vision-Language-Action Models

A Detailed Theoretical Results **15**

A.1 Additional details of the ROAD-VLA objective 15

A.2 Discrete-token teacher 15

A.3 Euclidean teachers 16

A.4 Flow-matching teacher 16

A.5 Assumptions and proof of Theorem 1 17

B ROAD-VLA Pseudocode **18**

C Additional Related Work **19**

D Text-Guided Privileged Context **19**

E Detailed Experimental Settings **21**

E.1 OpenVLA and OpenVLA-OFT on ManiSkill: Environment and Implementation Details 21

E.2 $\pi_{0.5}$ on LIBERO: Environment and Implementation Details 23

F Additional Analyses and Ablation Studies **25**

F.1 Extending to Continuous and Flow-Matching Heads 25

F.2 Core Numerical Results 26

F.3 Training Dynamics on LIBERO 26

F.4 Failure Mode Analysis 27

F.5 Ablation of Reference Guidance and Distillation 28

F.6 Sensitivity to Reference Critic Checkpoint 28

F.7 Signed Advantage Weight 29

F.8 Computational Cost 29

G Additional Visualizations **30**

H Limitation and Future Work **30**

A Detailed Theoretical Results

A.1 Additional details of the ROAD-VLA objective

This appendix gives the derivations behind Section 4.2.2 and the proof of Theorem 1.

Decision index and action chunks. Throughout ROAD-VLA, t indexes a policy decision. For a chunking policy, one decision produces the full chunk $\hat{\mathbf{a}}_t = (\hat{a}_{t,0}, \dots, \hat{a}_{t,C-1}) \in \mathbb{R}^{Cd_a}$. PPO assigns one advantage to that decision, and ROAD-VLA uses the corresponding scalar ω_t for the entire chunk. We do not introduce separate credit assignment for the C primitive controls inside the chunk.

Why ω_t is not standardized. The critic baseline already defines the relevant zero point: positive advantage means better than expected and negative advantage means worse than expected. Re-centering by a batch mean can change this sign interpretation, while per-batch rescaling can make the teacher step vary sharply under sparse rewards. We therefore keep the signed advantage and bound only its magnitude by c; η controls the teacher step size.

Bregman geometry and optimality. For a differentiable strictly convex Φ on a convex prediction domain $\mathcal{Y}$,

$$D_\Phi(y, y') = \Phi(y) - \Phi(y') - \langle \nabla\Phi(y'), y - y' \rangle.$$

Writing $u_t = \eta\omega_t\delta_t$, Eq. 6 is

$$y_t^\star = \arg\max_{y\in\mathcal{Y}} \left\{ \langle u_t, y - y_t^{\mathrm{old}} \rangle - D_\Phi(y, y_t^{\mathrm{old}}) \right\}. \tag{14}$$

Its constrained first-order condition is

$$0 \in \nabla\Phi(y_t^\star) - \nabla\Phi(y_t^{\mathrm{old}}) - u_t + N_{\mathcal{Y}}(y_t^\star), \tag{15}$$

where $N_{\mathcal{Y}}$ is the normal cone of $\mathcal{Y}$. For unconstrained Euclidean predictions this reduces to an additive step; for categorical distributions on the simplex it yields the exponential tilt derived below.

Since $y = y_t^{\mathrm{old}}$ has objective value zero,

$$r_t(y_t^\star) \geq D_\Phi(y_t^\star, y_t^{\mathrm{old}}). \tag{16}$$

This inequality concerns the shaped proximal objective only; it does not by itself imply improvement in environment return. Assumption (A1) is the condition connecting the teacher displacement to true advantage.

Anchoring the teacher. All quantities entering the teacher are computed from the rollout policy θ_{old} and frozen during the student update. This is particularly important for negative ω_t: a fixed teacher gives a non-negative distillation objective with a fixed minimizer, whereas rebuilding the target from the current student can reduce to a negatively weighted regression objective with no finite minimum.

A.2 Discrete-token teacher

Fix one token position. Let $p^{\mathrm{old}} \in \Delta(\mathcal{V})$ be the rollout categorical distribution and let $\hat{a}$ be the sampled token. With negative entropy, $D_\Phi(q, p^{\mathrm{old}}) = \mathrm{KL}(q\|p^{\mathrm{old}})$, and Eq. 14 becomes

$$q^\star = \arg\max_{q\in\Delta(\mathcal{V})} \left\{ \eta\omega\langle e_{\hat{a}}, q - p^{\mathrm{old}} \rangle - \mathrm{KL}(q\|p^{\mathrm{old}}) \right\}. \tag{17}$$

The constant term does not affect the optimizer. Using a multiplier λ for $\sum_v q(v) = 1$, stationarity gives

$$\eta\omega\mathbf{1}[v = \hat{a}] - \left(\log\frac{q(v)}{p^{\mathrm{old}}(v)} + 1 \right) + \lambda = 0.$$

Hence $q(v) \propto p^{\mathrm{old}}(v)\exp(\eta\omega\mathbf{1}[v = \hat{a}])$, so

$$q^\star = \mathrm{softmax}\left(z^{\mathrm{old}} + \eta\omega e_{\hat{a}}\right), \qquad p^{\mathrm{old}} = \mathrm{softmax}(z^{\mathrm{old}}). \tag{18}$$

Thus positive ω shifts mass toward the sampled token and negative ω shifts it away.

For an autoregressive action with K tokens, the same decision-level ω_t is used at every position. For a realized rollout action $\hat{a}_t$, define, for every prefix $a_{<k}$,

$$q^\star_{t,k}(\cdot \mid a_{<k}, h_t, \hat{a}_t) = \mathrm{softmax}\left(z^{\mathrm{old}}_{t,k}(a_{<k}, h_t) + \eta\omega_t e_{\hat{a}_{t,k}}\right). \tag{19}$$

These conditionals define a valid autoregressive teacher. The implementation evaluates them only on the sampled rollout prefixes and minimizes

$$\widehat{D}^{\mathrm{tok}}_t = \frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}\left(q^\star_{t,k}(\cdot \mid \hat{a}_{t,<k}, h_t, \hat{a}_t) \| p^\theta_{t,k}(\cdot \mid \hat{a}_{t,<k}, h_t)\right). \tag{20}$$

For a tokenized action chunk, the K tokens jointly encode the full chunk and share one chunk-level ω_t. The teacher-forced loss above is not, in general, identical to the sequence-level KL between two autoregressive policies; Theorem 1 therefore keeps the corresponding behavioral relation explicit in Assumption (A2). If a distillation temperature is used in the implementation, it is applied inside this empirical divergence and does not alter the proximal teacher in Eq. 18.

A.3 Euclidean teachers

For regression and local flow predictions, $\Phi(y) = \frac{1}{2}\|y\|_2^2$, so $D_\Phi(y, y') = \frac{1}{2}\|y - y'\|_2^2$. Equation 14 becomes

$$y^\star_t = \arg\max_y \left\{\eta\omega_t\langle\delta_t, y - y^{\mathrm{old}}_t\rangle - \frac{1}{2}\|y - y^{\mathrm{old}}_t\|_2^2\right\}, \tag{21}$$

whose unique solution is

$$y^\star_t = y^{\mathrm{old}}_t + \eta\omega_t\delta_t. \tag{22}$$

At the solution,

$$r_t(y^\star_t) = \eta^2\omega_t^2\|\delta_t\|_2^2, \qquad D_\Phi(y^\star_t, y^{\mathrm{old}}_t) = \frac{1}{2}\eta^2\omega_t^2\|\delta_t\|_2^2. \tag{23}$$

For a regression chunk, $\delta_t = \hat{\mathbf{a}}_t - \mathbf{a}^{\mathrm{old}}_t$, hence

$$\mathbf{a}^\star_t = \mathbf{a}^{\mathrm{old}}_t + \eta\omega_t(\hat{\mathbf{a}}_t - \mathbf{a}^{\mathrm{old}}_t).$$

The whole Cd_a-dimensional prediction is updated by the same chunk-level advantage. Because $|\omega_t| \le c$, choosing $\eta \le 1/c$ prevents a positive-advantage Euclidean teacher from overshooting the executed target; for negative advantage, its displacement in the opposite direction is at most $\|\delta_t\|_2$.

A.4 Flow-matching teacher

For an executed chunk $\hat{\mathbf{a}}_t \in \mathbb{R}^{Cd_a}$, draw $\epsilon \sim \mathcal{N}(0, I)$ and flow time $s \in (0, 1]$, and form

$$x_s = s\epsilon + (1 - s)\hat{\mathbf{a}}_t, \qquad v^{\mathrm{FM}}_t = \epsilon - \hat{\mathbf{a}}_t. \tag{24}$$

At this flow state, the rollout policy predicts

$$v^{\mathrm{old}}_t = v^{\theta_{\mathrm{old}}}(x_s, s \mid h_t),$$

and the ROAD-VLA teacher is

$$v^\star_t = v^{\mathrm{old}}_t + \eta\omega_t\left(v^{\mathrm{FM}}_t - v^{\mathrm{old}}_t\right). \tag{25}$$

The sampled pair (ϵ, s) is held fixed during the student update, and the empirical distillation loss is

$$\widehat{D}^{\mathrm{flow}}_t = \frac{1}{2}\left\|v^\star_t - v^\theta(x_s, s \mid h_t)\right\|_2^2. \tag{26}$$

Its population counterpart is

$$D^{\mathrm{flow}}_t = \mathbb{E}_{\epsilon,s}\left[\frac{1}{2}\left\|v^\star_t - v^\theta(x_s, s \mid h_t)\right\|_2^2\right], \tag{27}$$

so Eq. 26 is a one-sample Monte Carlo estimate of the native divergence used in the theoretical analysis.

A.5 Assumptions and proof of Theorem 1

Population quantities. To keep the main text compact, D_t suppresses the randomness used to construct the teacher. At a fixed context h_t, let ξ_t collect the rollout action or chunk, its advantage-derived weight, and any auxiliary flow noise. If $d_t(\cdot, \cdot; \xi_t)$ denotes the corresponding per-sample native divergence, define

$$D_t(y_t^\star, y_t^\theta) = \mathbb{E}_{\xi_t}\left[d_t(y_t^\star, y_t^\theta; \xi_t)\right]. \tag{28}$$

For token heads, d_t is Eq. 20; for regression it is half squared distance on the full chunk; and for flow it is the sampled velocity-space loss in Eq. 26.

Let $\pi_t^\star(\cdot \mid h_t, \xi_t)$ denote the behavior decoded from the corresponding teacher for a realized construction ξ_t. The main text uses $\pi_t^\star(\cdot \mid h_t)$ as shorthand for the resulting mixture after marginalizing ξ_t. We write

$$\bar{A}_t^\star(h_t) = \mathbb{E}_{\xi_t}\mathbb{E}_{a\sim\pi_t^\star(\cdot|h_t,\xi_t)}\left[A_t^{\pi_{\text{old}}}(h_t, a)\right], \qquad \bar{A}_t^{\theta'}(h_t) = \mathbb{E}_{a\sim\pi_{\theta'}(\cdot|h_t)}\left[A_t^{\pi_{\text{old}}}(h_t, a)\right]. \tag{29}$$

For regression, the teacher uses the same fixed exploration mechanism as the policy; for flow, it is the sampler induced by the teacher velocity field above.

Assumption intuition. The theorem is deliberately conditional.

(A1) Teacher calibration requires

$$\bar{A}_t^\star(h_t) \geq \beta D_t(y_t^\star, y_t^{\text{old}}) - \epsilon_{\text{cal}}.$$

Intuitively, moving the teacher away from the rollout prediction should correspond, on average, to an improving direction under the true rollout-policy advantage. The proximal construction alone does not guarantee this: in particular, moving away from a negative-advantage sample need not always point toward a better action.

(A2) Distillation stability requires

$$\left|\bar{A}_t^\star(h_t) - \bar{A}_t^{\theta'}(h_t)\right| \leq L\sqrt{2D_t(y_t^\star, y_t^{\theta'})}.$$

Intuitively, if the student is close to the teacher in the native prediction space, their decoded behaviors should have similar expected advantage. For regression this is a Lipschitz-type condition on decoded chunk behavior; for flow it also includes stability of the sampler. For token policies, bounded advantage together with full sequence-level KL would give an analogous bound by Pinsker's inequality, but the implemented teacher-forced KL uses rollout prefixes, so we keep (A2) explicit.

(A3) Bounded advantage requires $|A_t^{\pi_{\text{old}}}(h_t, a)| \leq B_A$. This is a mild condition for bounded finite-horizon rewards. If $|\rho_t| \leq R_{\max}$ and $\gamma < 1$, for example, one may take $B_A \leq 2R_{\max}/(1-\gamma)$; a finite-horizon bound can be made tighter by truncating the geometric sum. If the only nonzero reward is a binary terminal success indicator, then $B_A \leq 1$.

These assumptions serve as sufficient conditions for the bound and make explicit the conditions under which advantage-guided native-space distillation translates into policy improvement.

Proof. For h_t in the support of $d_t^{\pi_{\text{old}}}$, (A1) and (A2) imply

$$\bar{A}_t^{\theta'}(h_t) \geq \beta D_t(y_t^\star, y_t^{\text{old}}) - L\sqrt{2D_t(y_t^\star, y_t^{\theta'})} - \epsilon_{\text{cal}}. \tag{30}$$

Define $F_t(h) = \bar{A}_t^{\theta'}(h)$. By (A3), $|F_t(h)| \leq B_A$. Using $\text{TV}(p, q) = \frac{1}{2}\|p - q\|_1$,

$$\mathbb{E}_{h_t\sim d_t^{\pi_{\theta'}}} F_t(h_t) \geq \mathbb{E}_{h_t\sim d_t^{\pi_{\text{old}}}} F_t(h_t) - 2B_A\,\text{TV}\left(d_t^{\pi_{\theta'}}, d_t^{\pi_{\text{old}}}\right). \tag{31}$$

Let

$$Z_\gamma = \sum_{t=1}^{T}\gamma^{t-1}, \qquad w_t = \frac{\gamma^{t-1}}{Z_\gamma}, \qquad J_\gamma(\pi) = \frac{1}{Z_\gamma}\mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^{t-1}\rho_t\right].$$

Here t indexes policy decisions and γ is the same decision-level discount used by PPO/GAE. The finite-horizon discounted performance-difference identity gives

$$J_\gamma(\pi_{\theta'}) - J_\gamma(\pi_{\text{old}}) = \sum_{t=1}^{T} w_t \mathbb{E}_{h_t\sim d_t^{\pi_{\theta'}}} F_t(h_t). \tag{32}$$

Substituting Eqs. 30 and 31 into Eq. 32 yields

$$J_\gamma(\pi_{\theta'}) \geq J_\gamma(\pi_{\text{old}}) + \sum_{t=1}^{T} w_t \mathbb{E}_{h_t \sim d_t^{\pi_{\text{old}}}} \left[\beta D_t(y_t^\star, y_t^{\text{old}}) - L\sqrt{2D_t(y_t^\star, y_t^{\theta'})} \right] - \epsilon_{\text{cal}} - 2B_A \sum_{t=1}^{T} w_t \, \text{TV}\left(d_t^{\pi_{\theta'}}, d_t^{\pi_{\text{old}}}\right), \tag{33}$$

which is Eq. 13. This proves Theorem 1.

B ROAD-VLA PSEUDOCODE

Algorithm 1 ROAD-VLA: Advantage-Guided Online Self-Distillation

Require: Policy π_θ with online critic V; frozen reference critic V^{ref}; teacher step size η; distillation weight λ; mixing weight α; advantage clip c

Ensure: Adapted policy π_θ

for each training iteration **do**
  $\theta_{\text{old}} \leftarrow \theta$
  // 1. Collect on-policy rollout
  Sample $\hat{\tau} = \{(h_t, \hat{a}_t, r_t)\}_{t=1}^{T} \sim \pi_{\theta_{\text{old}}}$
  Compute $\{\hat{A}_t^{\text{int}}\}_{t=1}^{T}$ from V and $\{\hat{A}_t^{\text{ref}}\}_{t=1}^{T}$ from V^{ref}
  Match the reference-advantage scale: $\tilde{A}_t^{\text{ref}} = \mu_{\text{int}} + \frac{\sigma_{\text{int}}}{\sigma_{\text{ref}} + \varepsilon}\left(\hat{A}_t^{\text{ref}} - \mu_{\text{ref}}\right)$
  for $t = 1, \ldots, T$ **do**
    // 2. Agreement-gated advantage
    if $\text{sign}(\hat{A}_t^{\text{int}}) = \text{sign}(\tilde{A}_t^{\text{ref}})$ **then**
      $\hat{A}_t^{\text{mix}} = (1-\alpha)\hat{A}_t^{\text{int}} + \alpha\tilde{A}_t^{\text{ref}}$
    else
      $\hat{A}_t^{\text{mix}} = \hat{A}_t^{\text{int}}$
    end if
    $\omega_t = \text{clip}(\hat{A}_t^{\text{mix}}, -c, c)$
    // 3. Construct fixed teacher
    if token head **then**
      for $k = 1, \ldots, K$ **do**
        $q_{t,k}^\star = \text{softmax}\left(z_{t,k}^{\text{old}} + \eta\omega_t e_{\hat{a}_{t,k}}\right)$
      end for
    else if regression head **then**
      $\mathbf{a}_t^\star = \mathbf{a}_t^{\text{old}} + \eta\omega_t\left(\hat{\mathbf{a}}_t - \mathbf{a}_t^{\text{old}}\right)$
    else if flow-matching head **then**
      Draw one (ϵ, s) and set $x_s = s\epsilon + (1-s)\hat{\mathbf{a}}_t$
      $v_t^{\text{FM}} = \epsilon - \hat{\mathbf{a}}_t$
      $v_t^\star = v_t^{\text{old}} + \eta\omega_t\left(v_t^{\text{FM}} - v_t^{\text{old}}\right)$
    end if
  end for
  // 4. PPO + native-space distillation
  $\mathcal{L}_{\text{AGD}} = \frac{1}{T}\sum_{t=1}^{T} \widehat{D}_t\left(\text{sg}(y_t^\star), y_t^\theta\right)$
  where $\widehat{D}_t$ is forward KL for tokens, $\frac{1}{2}\|\mathbf{a}_t^\star - \mathbf{a}_t^\theta\|_2^2$ for regression, and $\frac{1}{2}\|v_t^\star - v^\theta(x_s, s \mid h_t)\|_2^2$ for flow matching
  $\mathcal{L}_{\text{ROAD}} = \mathcal{L}_{\text{PPO}} + \lambda\mathcal{L}_{\text{AGD}}$
  Update policy/value parameters using $\mathcal{L}_{\text{ROAD}}$, keeping all teacher targets fixed
end for

C Additional Related Work

Vision–Language–Action Models. VLA models have emerged as a general framework for language-conditioned robot control. RT-1 demonstrated scalable Transformer policies trained on diverse robot data (Brohan et al., 2022), while RT-2 and RT-X connected vision-language pretraining with cross-embodiment control using discrete action tokens (Zitkovich et al., 2023b; O'Neill et al., 2024). PaLM-E extended this paradigm toward embodied multimodal reasoning with continuous sensor inputs (Driess et al., 2023).

Recent open-source VLAs span multiple action representations. OpenVLA uses autoregressive discrete action tokens (Kim et al., 2024), while OpenVLA-OFT predicts continuous action chunks (Kim et al., 2025). Other systems use diffusion- or flow-based action representations, including Octo (Octo Model Team et al., 2024), π_0 (Black et al., 2024), $\pi_{0.5}$ (Black et al., 2025), and GR00T N1 (Bjorck et al., 2025). OpenVLA combines pretrained visual and language components (Zhai et al., 2023; Karamcheti et al., 2024; Oquab et al., 2024) with autoregressive action prediction. Despite strong pretrained capabilities, VLA policies remain susceptible to distribution shifts at deployment. Our work focuses on adapting these different action representations from sparse reward feedback through a common advantage-guided self-distillation mechanism.

Online Adaptation and Reinforcement Learning. Online RL provides a natural mechanism for adapting pretrained VLA policies through environment interaction. PPO (Schulman et al., 2017) is widely used for policy optimization, and recent work demonstrates its effectiveness for VLA adaptation (Liu et al., 2026b). However, policy-gradient methods typically use advantages as scalar weights, which provides only coarse supervision for structured predictions such as multi-token actions, continuous chunks, and flow fields.

Related advantage-weighted methods include AWR (Peng et al., 2019), AWAC (Nair et al., 2020), and CRR (Wang et al., 2020). These methods emphasize regression toward higher-advantage actions, but do not explicitly construct a proximal target around the current policy prediction. Recent VLA-specific approaches introduce additional adaptation signals or regularization, including robustness-aware regularization in RobustVLA (Zhang et al., 2025) and task-progress feedback in TT-VLA (Liu et al., 2026a). In contrast, ROAD-VLA directly transforms the policy's own advantage estimates into a bounded mirror-descent update, producing a proximal teacher in the native action space. This construction is shared across discrete, continuous, and flow-matching action heads.

Distillation and Privileged Guidance. Knowledge distillation (Hinton et al., 2015) and policy distillation (Rusu et al., 2015) transfer predictive or behavioral knowledge from one model to another, and related ideas have also been used for continual adaptation (Li & Hoiem, 2016). More recently, self-distillation methods for LLM post-training show that privileged or on-policy teachers can transform sparse outcome feedback into denser supervision (Shenfeld et al., 2026; Zhao et al., 2026).

In robotics, privileged supervision often takes the form of language- or vision-language-based feedback and reward shaping (Ma et al., 2024; Liang et al., 2023; Yu et al., 2023). Such guidance can express useful high-level information, but may lack the spatial and temporal precision required for low-level control. VLA-specific approaches instead use action-space teachers. ActDistill (Ye et al., 2025) employs action-guided distillation for model compression, while VLA-OPD (Zhong et al., 2026) performs on-policy distillation during RL using an external expert. ROAD-VLA differs in that its teacher is constructed directly from the current policy prediction and its advantage estimates, without requiring privileged language or an external expert.

D Text-Guided Privileged Context

Shared setup. All text-guided privileged-information (PI) experiments are conducted with **OpenVLA**, following its standard prompt format:

```
In: What action should the robot take to {task}?\nOut:
```

The teacher shares the same OpenVLA parameters as the student but receives an additional text hint I_t during training. Both teacher and student evaluate the executed action sequence under teacher forcing.

Let $z^S_{t,k}$ and $z^T_{t,k}$ denote the student and teacher logits for the k-th action token. With distillation temperature $\tau_{\mathrm{KD}} = 2$, define

$$p^{X,\tau_{\mathrm{KD}}}_{t,k} = \mathrm{softmax}\left(\frac{z^X_{t,k}}{\tau_{\mathrm{KD}}}\right), \qquad X \in \{S, T\}. \tag{34}$$

We use the teacher-to-student forward KL

$$\ell_t = \frac{\tau^2_{\mathrm{KD}}}{K} \sum_{k=1}^{K} \mathrm{KL}\left(p^{T,\tau_{\mathrm{KD}}}_{t,k} \| p^{S,\tau_{\mathrm{KD}}}_{t,k}\right). \tag{35}$$

Here, τ_{KD} only softens the distributions used for distillation and is unrelated to the ROAD-VLA proximal step, which is controlled by $\eta\omega_t$. Privileged hints are never used during rollout or evaluation.

MCTS PI: Retrieved action hints. MCTS PI retrieves successful actions from the policy's own interaction history, without using simulator state. Each rollout worker maintains a buffer of successful state–action pairs (e_i, a_i), where e_i is an L_2-normalized, mean-pooled OpenVLA visual feature and a_i is the executed action. When an episode first succeeds, its transitions up to success are added to the buffer, which is capped at 2×10^4 entries.

At timestep t, the current feature e_t retrieves the stored action with maximum cosine similarity,

$$q_t = \max_i \cos(e_t, e_i).$$

Current-rollout transitions are inserted only after retrieval to avoid self-matching, and no hint is used if the buffer is empty or $q_t < 0$.

For a 7-DoF action $a_i = [dx, dy, dz, d\mathrm{roll}, d\mathrm{pitch}, d\mathrm{yaw}, g]$, the retrieved action is verbalized as

```
Hint (q={q_t}): move ({dx}, {dy}, {dz}), rotate ({droll},
{dpitch}, {dyaw}), gripper {open|close}
```

with values rounded to two decimal places. The hint is appended directly to the OpenVLA instruction.

RelSpatial PI: Egocentric spatial state. RelSpatial PI uses privileged simulator state expressed relative to the end-effector. We extract the end-effector position p_g, source-object position p_s, target-object position p_t, grasp status, and gripper state. Relative displacements are mapped to qualitative directions:

$$+x/-x \mapsto \text{right/left}, \qquad +y/-y \mapsto \text{ahead/behind}, \qquad +z/-z \mapsto \text{above/below}.$$

An axis is reported only when $|\Delta| \geq 0.2\,\mathrm{cm}$; otherwise it is described as *aligned*. No absolute coordinates or exact distances are provided.

The hint has the form

```
{src} is {dir(p_s − p_g)} end-effector, {grasped|not grasped}.
{tgt} is {dir(p_t − p_g)} end-effector. {src} to {tgt}: move
{dir(p_t − p_s)}. end-effector {open|closed}
```

and is inserted into a dedicated reference field:

```
In: What action should the robot take to {task}?\n
Reference: {I_t}\nOut:
```

Plan+RelSpatial PI. Plan+RelSpatial PI prepends a fixed high-level manipulation plan to the RelSpatial description:

```
Here is a possible reference plan to achieve the instruction
"{task}": 1. Approach the object. 2. Align for grasping.
3. Grasp and lift the object. 4. Move toward the target.
5. Align for placement. 6. Place and retract. Current
state: {RelSpatial description}
```

Training procedure. OpenVLA always acts using the standard hint-free prompt. Privileged text is used only during optimization. For RelSpatial and Plan+RelSpatial, we first perform the standard PPO update, then distill on the selected successful rollout transitions using Eq. 35. For MCTS PI, PPO and text-guided distillation are optimized jointly over the collected rollout transitions.

E Detailed Experimental Settings

E.1 OpenVLA and OpenVLA-OFT on ManiSkill: Environment and Implementation Details

In this section, we provide the technical specifications for the evaluation suite used in Section 5. All environments are obtained from the RL4VLA repo [1], which provides a standardized interface for reinforcement learning with vision-language-action models. Observations consist of RGB images from a front-view camera and the robot's proprioceptive state, while the action space is discretized into $K = 7$ tokens per step following the OpenVLA specification.

Table 2: Mapping of main paper environment aliases to technical identifiers.

Category	Main Paper Alias	Simulator Environment ID
Visual Robustness	VR-UnseenTable	PutOnPlateInScene25VisionImage-v1
	VR-DynamicTexture	PutOnPlateInScene25VisionTexture05-v1
	VR-DynamicNoise	PutOnPlateInScene25VisionWhole05-v1
Compositional Reasoning	CR-MultiObject	PutOnPlateInScene25MultiCarrot-v1
	CR-MultiReceptacle	PutOnPlateInScene25MultiPlate-v1
Execution Robustness	ER-InitPose	PutOnPlateInScene25EEPose-v1
	ER-Repositioning	PutOnPlateInScene25PositionChangeTo-v1

E.1.1 Environment Perturbation Protocols

To systematically probe the robustness of ROAD-VLA, we implemented the following perturbations across our axes of evaluation:

Visual Robustness. These environments test the model's invariance to low-level visual shifts.

- **Unseen Table:** Replaces the training table surface with diverse, unseen textures and materials (e.g., wood, marble, metallic) not present in the warm-up data.
- **Dynamic Texture:** A random distractor texture is overlaid on the object, receptacle, and robot arm. This texture is re-cropped and resized at **every timestep** to create non-stationary visual noise. We evaluate at the strong intensity levels: *strong* ($\alpha = 0.5$).
- **Dynamic Noise:** Similar to dynamic textures, but the flickering noise is overlaid on the entire image frame, simulating extreme sensor interference or low-light artifacts at intensities of 0.5.

Compositional Reasoning. These environments test semantic grounding and the ability to handle distractors.

[1] https://github.com/gen-robot/RL4VLA

- **Multi-Object/Receptacle:** Spawns multiple distractor objects (e.g., additional carrots or plates) that were not present during the zero-shot task phase. The model must correctly identify the target specified in the instruction l despite visual similarity.

Execution Robustness. These environments test temporal adaptation and dynamic error correction.

- **Init Pose:** Randomizes the initial end-effector (EE) pose outside the standard training distribution, requiring the model to generalize its reach-to-grasp trajectories.
- **Repositioning (Mid-episode):** To test for closed-loop visual feedback, the target object is teleported to a new random position at the 5th timestep of the episode. This requires the model to break its initial plan and adapt to the updated state mid-execution.

E.1.2 OpenVLA and OpenVLA-OFT Implementation Details

Model and framework. All ManiSkill experiments are built on RLinf and start from the checkpoints its OpenVLA and OpenVLA-OFT ManiSkill recipes start from, so that PPO and ROAD-VLA share one initialization and differ only in the update rule. For OpenVLA-7B, the student is the ManiSkill warm-up checkpoint on which the RLinf OpenVLA PPO model is based, `gen-robot/openvla-7b-rlvla-warmup`[2], a supervised warm-up of OpenVLA-7B on 140 planner-collected trajectories; it predicts one 7-DoF action per policy call as 7 discrete tokens over 256 bins, from a single 224×224 image and the instruction, with no proprioception. For OpenVLA-OFT-7B, the student is the RLinf ManiSkill base `RLinf/RLinf-OpenVLAOFT-ManiSkill-Base-Main`[3] with its supervised `RLinf-OpenVLAOFT-ManiSkill-Base-Lora`[4] adapter merged into the weights); it predicts a chunk of $C = 8$ actions (56 tokens) per policy call in one parallel-decoding pass, and the environment executes the whole chunk before the next observation. In both cases a fresh rank-32 LoRA ($\alpha = 32$, no dropout) on the attention and MLP projections of the vision encoder, projector and language backbone, together with a value head on the final hidden state, are the only trained parameters; the base weights are frozen. Exploration is the categorical sampling of the PPO baseline at temperature 1.0, reduced to 0.6 for evaluation, so both methods see the same rollout distribution at initialization.

Advantage estimation. Advantages use GAE ($\gamma = 0.99$, $\lambda_{\text{GAE}} = 0.95$) over environment steps from the student's value head; for OpenVLA-OFT one reward (the sum over the chunk) and one value per policy call are used, treating the chunk as a single macro-action, and the PPO ratio is the joint likelihood ratio of the whole chunk (56 action dimensions). The intrinsic and reference estimates are mixed and gated as in Section 4.2.2 with $\alpha = 0.5$, $c = 2.0$ and the agreement gate; ω_t is computed from the unnormalized mixed advantage. As for $\pi_{0.5}$, the reference critic is the value head of the PPO checkpoint released with RLinf for the same backbone (`RLinf/RLinf-OpenVLA-PPO-ManiSkill3-25ood`[5] for OpenVLA and `RLinf/RLinf-OpenVLAOFT-PPO-ManiSkill3-25ood`[6] for OpenVLA-OFT), which is shared by every seed and condition; the reference advantage is a second GAE pass over the frozen critic's values. Training and reference critics are never updated from OOD data: the reference is frozen and the student critic sees only training-split rollouts.

Teacher. The teacher is the student's own action-token distribution with the executed token's logit shifted by $\eta\,\omega_t$ ($\eta = 0.5$), at temperature $\tau_{\text{KD}} = 2$; the distillation term is the forward KL from teacher to student summed over the tokens of a policy call, scaled by τ_{KD}^2 and added with $\lambda = 0.5$. A transition's distillation gradient is dropped once its PPO ratio has already moved past the clip range in the direction of ω_t, so distillation cannot push a transition beyond what the clipped surrogate would allow.

[2] `https://huggingface.co/gen-robot/openvla-7b-rlvla-warmup`
[3] `https://huggingface.co/RLinf/RLinf-OpenVLAOFT-ManiSkill-Base-Main`
[4] `https://huggingface.co/RLinf/RLinf-OpenVLAOFT-ManiSkill-Base-Lora`
[5] `https://huggingface.co/RLinf/RLinf-OpenVLA-PPO-ManiSkill3-25ood`
[6] `https://huggingface.co/RLinf/RLinf-OpenVLAOFT-PPO-ManiSkill3-25ood`

Optimization. Both methods use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01) with a policy learning rate of 1×10^{-4} and a value-head learning rate of 3×10^{-3}, one PPO epoch per rollout, clip ratio 0.2. OpenVLA follows the RLinf OpenVLA recipe (cosine schedule with 160 warm-up optimizer steps, gradient clipping at 1.0); OpenVLA-OFT follows the RLinf OFT recipe (constant learning rate, gradient clipping at 10).

Protocol. Each run is 400 rollout–update iterations on two GPUs (H100 or H200). Episodes last at most 80 steps and terminate on success. Every 20 iterations the policy is evaluated at temperature 0.6 on 32 fixed trajectories of the training split (ID) and 32 of the held-out split (OOD) of the condition's environment; the seed fixes the evaluation initial states, so ID and OOD numbers within a run are paired. The PPO-OFT baseline uses the RLinf OFT recipe unchanged (128 environments, global batch 640, upper clip ratio 0.28) See Table 3.

Table 3: Hyperparameters of the ManiSkill experiments. A single value applies to both backbones.

Hyperparameter	**OpenVLA**	**OpenVLA-OFT**
Shared (PPO and ROAD-VLA)		
Student initialization	RLinf/RL4VLA warm-up	RLinf Base-Main + Base-LoRA
Actions per policy call (C)	1	8
Parallel environments	16	16
Rollout length (env. steps)	80	80
Episode length (env. steps)	80	
Iterations	400	
Global batch (transitions)	160	
Optimizer steps per iteration	8	
PPO epochs per rollout	1	
Clip ratio	0.2	
GAE γ / λ_{GAE}	0.99 / 0.95	
Entropy coefficient	0.01	
LoRA rank / α	32 / 32	
Policy / value learning rate	1×10^{-4} / 3×10^{-3}	
LR schedule	cosine, 160 warm-up steps	constant
Gradient clipping	1.0	10.0
AdamW β_1 / β_2 / wd	0.9 / 0.999 / 0.01	
Sampling temperature (train / eval)	1.0 / 0.6	
Evaluation trajectories (ID / OOD)	32 / 32, every 20 iterations	
ROAD-VLA only		
Distillation coefficient (λ)	0.5	
Teacher step size (η)	0.5	
Advantage clip (c)	2.0	
Distillation temperature (τ_{KD})	2.0	
Divergence	forward KL	
Ratio gate on distillation	True	
Positive-only ω_t	False	
Reference advantage	GAE over frozen critic	
Reference mixing (α) / agreement gate	0.5 / True	

E.2 $\pi_{0.5}$ ON LIBERO: ENVIRONMENT AND IMPLEMENTATION DETAILS

Model and framework. We use the $\pi_{0.5}$ LIBERO SFT checkpoint released with RLinf (`RLinf/RLinf-Pi05-LIBERO-SFT`[7]), which predicts chunks of $C = 5$ actions with $K = 5$ Euler denoising steps and 32-dimensional zero-padded actions, of which the first 7 are sent to the simulator, and which takes no proprioceptive input, following the openpi and π_{RL} configuration. The original PaliGemma weights are frozen; we train the 300M-parameter action expert, LoRA adapters (rank 32, scaling factor 64, Gaussian initialization, no dropout) inserted into the attention and MLP projections of PaliGemma, and a value head on the VLM output. Within each forward

[7] `https://huggingface.co/RLinf/RLinf-Pi05-LIBERO-SFT`

Table 4: Hyperparameters of PPO and ROAD-VLA on LIBERO-Goal with $\pi_{0.5}$. Shared values apply to both methods; training seeds differ.

Hyperparameter	$\pi_{0.5}$
Shared (PPO and ROAD-VLA)	
Student initialization	RLinf/RLinf-Pi05-LIBERO-SFT
Actions per policy call / denoising steps	5 / 5
Parallel training environments	32
Rollout horizon / episode limit (env. steps)	320 / 320
Training iterations	200
Global batch (action-chunk samples)	128
Micro batch per GPU / accumulation steps	4 / 8
Optimizer steps per iteration / PPO epochs	16 / 1
PPO clip ratio / value clip	0.2 / 0.2
GAE γ / λ_{GAE}	0.99 / 0.95
PPO advantage normalization	True
Entropy coefficient / KL penalty coefficient	0 / 0
LoRA rank / α_{LoRA}	32 / 64
Policy / value learning rate	2×10^{-6} / 1×10^{-5}
LR schedule / warm-up steps	Constant / 0
Gradient clipping	1.0
AdamW β_1 / β_2 / weight decay	0.9 / 0.95 / 0.01
AdamW ϵ	10^{-8}
Training sampler / noise level	Flow-SDE / 0.3
Evaluation sampler	Seeded Flow-ODE
ROAD-VLA only	
Distillation coefficient (λ)	0.5
Teacher step size (η) / advantage clip (c)	0.5 / 2.0
Distillation objective	$\frac{1}{2}\,\mathrm{MSE}(v_\theta, v^\star)$
Reference advantage	$\hat{R}_t^{\mathrm{on}} - V_{\mathrm{ref}}(s_t)$
Reference mixing (α) / agreement gate	0.5 / True
Teacher anchor	Frozen rollout-policy velocity
Noise–time draws per transition	1, fixed within each update
Distillation support	5 executed actions $\times$ 7 dimensions

pass, the prefix key-value cache is computed once per observation and reused across the K denoising steps. Exploration during rollout uses the Flow-SDE noise injection of the PPO baseline with the same noise level, so that ROAD-VLA and PPO see identical rollout distributions at initialization and differ only in the update rule. The PPO term of Eq. 10 is the π_{RL} objective unchanged, including its per-denoising-step likelihood; the distillation term is added with $\lambda = 0.5$ and evaluates the action expert at the stored flow states only.

Advantage estimation. Advantages are estimated at the chunk level with GAE ($\gamma = 0.99$, $\lambda_{\mathrm{GAE}} = 0.95$) from the value head on the VLM output. The intrinsic and reference estimates are mixed and gated as in Section 4.2.2 with $\alpha = 0.5$ and $c = 2.0$; ω_t is computed from the unnormalized mixed advantage, while the PPO term normalizes its own advantages as in the baseline. The mixing weight $\alpha = 0.5$ and the agreement gate are the defaults of Section 4.2.2.

Teacher. For each executed chunk we draw $M = 1$ pair (ϵ, s), fixed for the duration of the update (Appendix A.4), and use $\eta = 0.5$.

Evaluation splits. In-distribution success is measured on the unperturbed LIBERO-Goal suite (10 tasks), using the same 64 task–initial-state pairs as the perturbed splits. Out-of-distribution success is measured on LIBERO-Pro through RLinf's evaluation entry point with `LIBERO_TYPE=pro` and one perturbation active at a time, selected with `LIBERO_PERTURBATION`. Table 5 maps the aliases of the main paper to these settings.

Protocol. Both methods use the rollout and optimization settings of `libero_goal_ppo_openpi_pi05.yaml`; ROAD-VLA changes only the `algorithm`

Table 5: Mapping of $\pi_{0.5}$ evaluation aliases to LIBERO-Pro perturbation settings.

Category	Main Paper Alias	LIBERO-Pro perturbation	`LIBERO_PERTURBATION`
In-distribution	`ID`	None (original LIBERO-Goal)	—
Visual Robustness	`VR-ObjectAttr`	Object attributes (color, texture, size)	`object`
Compositional Reasoning	`CR-TargetSwap`	Target object swapped in the instruction	`task`
Execution Robustness	`ER-ObjectPos`	Initial object positions	`swap`

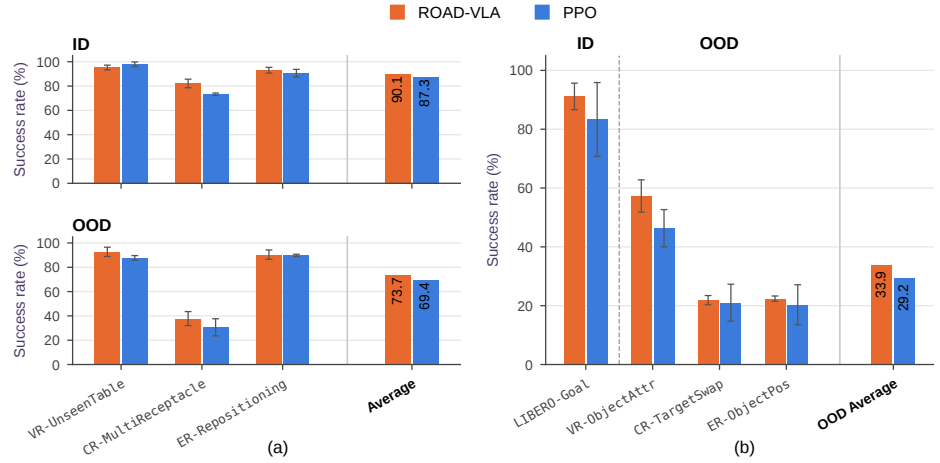


Figure 5: Success rates (%) of PPO and ROAD-VLA, reported as mean ± standard deviation over 3 seeds. (a) OpenVLA-OFT on ManiSkill. (b) $\pi_{0.5}$ on LIBERO, including the in-distribution `LIBERO-Goal` suite and three LIBERO-Pro OOD perturbations. Average denotes the mean over the evaluation environments shown. Numerical results are provided in Appendix F.2.

block. Training runs 32 parallel environments with one rollout epoch of 320 environment steps per iteration for 200 iterations, i.e. about 2.05M environment steps per run. Each method is trained with seeds 1 and 3. Checkpoints are evaluated every 25 iterations on 64 episodes per split.

F Additional Analyses and Ablation Studies

F.1 Extending to Continuous and Flow-Matching Heads

To test whether ROAD-VLA generalizes beyond discrete action tokens, we extend it to two different action representations: continuous regression with OpenVLA-OFT and flow matching with $\pi_{0.5}$.

F.1.1 Direct Regression (OpenVLA-OFT)

We first evaluate whether ROAD-VLA transfers from discrete action tokens to continuous action prediction (Figure 5a). OpenVLA-OFT (Kim et al., 2025) replaces autoregressive token decoding with a parallel regression head. For computational efficiency, we evaluate one environment per distribution-shift axis. Table 8 shows gains in five of six ID/OOD comparisons: ROAD-VLA improves average success from 87.29% to 90.12% ID and from 69.37% to 73.65% OOD, with the largest gains on `CR-MultiReceptacle` (+8.79 ID and +7.19 OOD points). The only decrease is on `VR-UnseenTable` ID (−2.71 points), where PPO already reaches 97.92%. These results show that the advantage-guided teacher also applies to continuous action chunks.

F.1.2 Flow Matching ($\pi_{0.5}$)

We further evaluate ROAD-VLA on the flow-matching $\pi_{0.5}$ policy (Black et al., 2025). This setting changes not only the action representation, but also the benchmark and RL framework: $\pi_{0.5}$ uses a

PaliGemma backbone with a separate 300M-parameter flow-matching action expert and is evaluated on LIBERO rather than ManiSkill.

Model and framework. We fine-tune $\pi_{0.5}$ within RLinf from the released LIBERO SFT checkpoint. ROAD-VLA uses the same PPO setup as the π_{RL} baseline and adds only the advantage-guided velocity-space distillation objective, with all other training settings shared (Appendix E.2). The flow teacher is constructed from the rollout velocity prediction using Eq. 9.

Training and evaluation. Both methods are trained on the unperturbed LIBERO-Goal suite and evaluated on the same suite (ID) and on three LIBERO-Pro perturbations (OOD), one per shift axis: `VR-ObjectAttr`, `CR-TargetSwap`, and `ER-ObjectPos` (Appendix E.2).

Results. Table 9 shows that ROAD-VLA improves ID success from 83.33% to 91.15% and outperforms PPO on all three OOD perturbations, with the largest gain on `VR-ObjectAttr` ($+10.94$ points). Together with the OpenVLA-OFT results, this shows that the advantage-guided teacher extends beyond discrete token distributions to both continuous regression and generative flow-matching policies.

For flow matching, the teacher remains local to the rollout prediction:

$$\|v_t^\star - v_t^{\mathrm{old}}\|_2 \le \eta c \|v_t^{\mathrm{FM}} - v_t^{\mathrm{old}}\|_2,$$

since $|\omega_t| \le c$, including for negative advantages (Appendix A.1).

F.2 Core Numerical Results

F.2.1 OpenVLA-7B

Tables 6 and 7 report the task success rates of Text-PI, VLA-OPD, PPO, and ROAD-VLA across the ID and OOD evaluation environments using OpenVLA-7B as backbone, respectively.

Table 6: Comparison of ID task success rates (%) between Text-PI, VLA-OPD, PPO, and ROAD-VLA across evaluation environments using the OpenVLA-7B backbone. All results are reported as mean $\pm$ standard deviation over 3 random seeds. The superior result in each row is bolded.

Backbone: OpenVLA-7B — ID					
Category	**Environment Name**	**Text-PI**	**VLA-OPD**	**PPO**	**ROAD-VLA**
Visual Robustness	`VR-UnseenTable`	78.13 ± 6.25	62.50 ± 6.31	83.85 ± 3.93	$\mathbf{90.10 \pm 1.80}$
	`VR-DynamicTexture`	59.75 ± 5.50	35.93 ± 1.57	77.08 ± 5.92	$\mathbf{83.33 \pm 6.31}$
	`VR-DynamicNoise`	62.50 ± 13.8	39.06 ± 1.56	71.88 ± 4.13	$\mathbf{80.73 \pm 2.39}$
Compositional Reasoning	`CR-MultiObject`	67.19 ± 2.21	45.31 ± 2.21	65.62 ± 4.42	$\mathbf{70.31 \pm 6.63}$
	`CR-MultiReceptacle`	68.75 ± 8.80	40.62 ± 6.20	67.71 ± 10.05	$\mathbf{69.79 \pm 5.80}$
Execution Robustness	`ER-InitPose`	83.33 ± 7.22	55.20 ± 5.91	81.25 ± 6.25	$\mathbf{88.54 \pm 4.77}$
	`ER-Repositioning`	86.45 ± 6.50	69.79 ± 2.39	90.62 ± 3.12	$\mathbf{93.75 \pm 3.12}$
Average		72.30	49.77	76.86	**82.36**

F.2.2 OpenVLA-OFT-7B

Tables 8 report the task success rates of PPO, and ROAD-VLA across the ID and OOD evaluation environments using OpenVLA-OFT as backbone, respectively.

F.2.3 $\pi_{0.5}$

Tables 9 report the task success rates of PPO, and ROAD-VLA across the ID and OOD evaluation environments using $\pi_{0.5}$ as backbone, respectively.

F.3 Training Dynamics on LIBERO

Figure 6 shows the evolution of task success during adaptation on LIBERO-Goal and three LIBERO-Pro shifts. ROAD-VLA generally maintains higher success than PPO throughout training, with the

Table 7: Comparison of OOD task success rates (%) between Text-PI, VLA-OPD, PPO, and ROAD-VLA across evaluation environments using the OpenVLA-7B backbone. All results are reported as mean ± standard deviation over 3 random seeds. The superior result in each row is bolded.

Backbone: OpenVLA-7B — OOD					
Category	**Environment Name**	**Text-PI**	**VLA-OPD**	**PPO**	**ROAD-VLA**
Visual Robustness	`VR-UnseenTable`	79.69 ± 1.56	68.75 ± 3.13	82.81 ± 4.13	$\mathbf{89.58 \pm 0.90}$
	`VR-DynamicTexture`	51.56 ± 7.81	31.25 ± 2.13	76.04 ± 5.49	$\mathbf{79.69 \pm 2.71}$
	`VR-DynamicNoise`	48.43 ± 7.81	23.44 ± 1.96	59.90 ± 9.42	$\mathbf{66.15 \pm 2.39}$
Compositional Reasoning	`CR-MultiObject`	54.69 ± 6.63	26.56 ± 2.21	50.00 ± 0.00	$\mathbf{60.94 \pm 3.12}$
	`CR-MultiReceptacle`	53.12 ± 0.00	31.25 ± 5.41	51.04 ± 4.77	$\mathbf{54.17 \pm 10.09}$
Execution Robustness	`ER-InitPose`	$\mathbf{79.17 \pm 4.33}$	53.12 ± 3.12	72.91 ± 14.10	$\mathbf{79.17 \pm 1.80}$
	`ER-Repositioning`	80.20 ± 6.50	66.67 ± 13.01	83.33 ± 15.73	$\mathbf{85.41 \pm 1.80}$
Average		63.84	43.01	68.00	**73.59**

Table 8: Success rates (%) between PPO and ROAD-VLA using the OpenVLA-OFT-7B backbone across evaluation environments. All results are reported as mean ± standard deviation over 3 random seeds. The superior result in each row is bolded.

Backbone: OpenVLA-OFT					
Category	**Task Name**	**ID**		**OOD**	
		PPO	**ROAD-VLA**	**PPO**	**ROAD-VLA**
Visual Robustness	`VR-UnseenTable`	$\mathbf{97.92 \pm 1.91}$	95.21 ± 2.01	87.71 ± 1.91	$\mathbf{92.71 \pm 3.82}$
Compositional Reasoning	`CR-MultiReceptacle`	73.33 ± 0.95	$\mathbf{82.12 \pm 3.54}$	30.62 ± 7.04	$\mathbf{37.81 \pm 5.75}$
Execution Robustness	`ER-Repositioning`	90.62 ± 3.12	$\mathbf{93.04 \pm 2.37}$	89.79 ± 0.95	$\mathbf{90.42 \pm 3.82}$
Average		87.29	**90.12**	69.37	**73.65**

clearest improvements on the ID, `ObjectAttr`, and `ObjectPos` settings. On `TargetSwap`, the two methods are closer and their relative performance varies across checkpoints, indicating a more challenging compositional shift. Overall, the curves suggest that ROAD-VLA improves adaptation across several OOD conditions while retaining strong in-distribution performance.

F.4 Failure Mode Analysis

To better understand model behavior, we examine the dominant failure patterns under representative OOD settings and compare how ROAD-VLA alters these outcomes.

In execution robustness tasks, in particular `ER-Repositioning`, PPO commonly exhibits a grasp-then-drift behavior: after a successful grasp, the policy moves toward the original training-distribution plate location rather than the shifted target, leading to object drop or task failure. As illustrated in Figure 9, this reflects reliance on a fixed spatial prior rather than continuous visual correction. ROAD-VLA reduces this failure pattern by maintaining stronger alignment with updated task cues, enabling successful object placement under shifted configurations.

In visual robustness tasks, specifically `VR-DynamicTexture`, the dominant error is perception-induced no-grasp. PPO frequently fails to localize the object under unseen textures, resulting in repeated unsuccessful approach attempts and eventual timeout. By contrast, ROAD-VLA demonstrates more reliable visual grounding, achieving successful grasps and task completion despite altered appearances. At the final checkpoint, ROAD-VLA maintains a 2.0% advantage (76.0% vs. 74.0%), with the gap reaching up to 7.3% during mid-training. This suggests improved robustness to appearance-level perturbations, though perceptual invariance remains a challenge in highly shifted scenes.

Overall, the qualitative comparisons reveal that ROAD-VLA not only improves final success rates, but also changes the underlying failure dynamics from rigid prior-driven execution and unstable perception toward more adaptive, task-aligned behavior.

Table 9: $\pi_{0.5}$ fine-tuned on LIBERO-Goal and evaluated on the unperturbed LIBERO-Goal suite (ID) and three LIBERO-Pro perturbations (OOD), one per axis. Success rates (%) are mean $\pm$ standard deviation over 3 seeds; the superior method in each row is bolded. The OOD average is computed over the three perturbations.

Backbone: $\pi_{0.5}$				
Split	**Category**	**Suite / Perturbation**	**PPO**	**ROAD-VLA**
ID	Unperturbed	`LIBERO-Goal`	83.33 ± 12.53	$\mathbf{91.15 \pm 4.51}$
OOD	**Visual Robustness**	`VR-ObjectAttr`	46.35 ± 6.32	$\mathbf{57.29 \pm 5.49}$
	Compositional Reasoning	`CR-TargetSwap`	21.35 ± 6.26	$\mathbf{21.88 \pm 1.57}$
	Execution Robustness	`ER-ObjectPos`	20.31 ± 6.81	$\mathbf{22.40 \pm 0.90}$
OOD Average			29.34 ± 5.89	$\mathbf{33.86 \pm 2.09}$

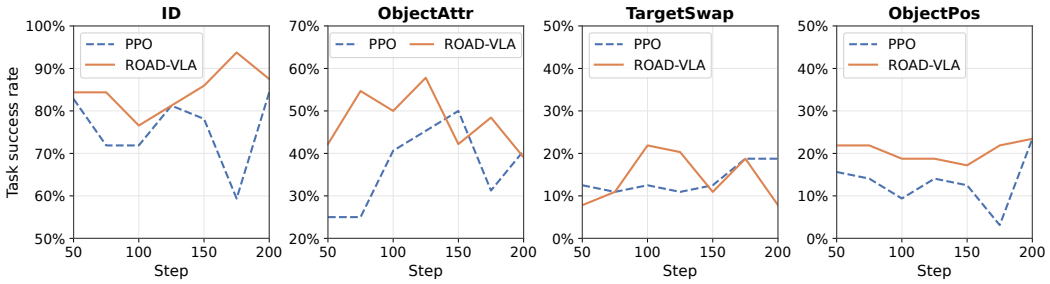


Figure 6: Task success rate over training on the original LIBERO-Goal tasks (`ID`) and three LIBERO-PRO shifts: object attributes (`ObjectAttr`), target swap (`TargetSwap`), and object position (`ObjectPos`).

F.5 Ablation of Reference Guidance and Distillation

Figure 7 compares several design choices in ROAD-VLA. The default setting with $\alpha = 0.5$ achieves the strongest overall training trajectory and finishes at approximately the same high success level as the reference-only setting ($\alpha = 1$). Removing the sign-agreement gate produces fast intermediate gains, but performance drops noticeably late in training, indicating that unfiltered reference guidance can become unreliable as the policy evolves. Increasing the distillation weight to $\lambda = 1$ also underperforms the default, suggesting that overly strong teacher matching can limit adaptation. Overall, these results support the default configuration, which combines moderate reference mixing with agreement-gated guidance and a balanced distillation weight.

F.6 Sensitivity to Reference Critic Checkpoint

In all main experiments, ROAD-VLA uses the released RLinf PPO checkpoint as the reference critic. To study the sensitivity of ROAD-VLA to the choice and training stage of the reference critic, we additionally conduct an online ablation in Figure 8, where we train PPO and freeze reference critics at different stages of online training (episodes 39, 79, 119, 159, and 165). We evaluate these reference critics on `VR-UnseenTable` and `ER-Repositioning`. Overall, ROAD-VLA is relatively robust to the reference checkpoint, with variants from different training stages following similar learning trajectories and converging within a comparable performance range. Interestingly, the most fully trained critic does not consistently yield the best downstream performance: while later checkpoints perform strongly on visual robustness, an intermediate checkpoint (episode 79, approximately 50% of the online PPO training trajectory) provides a modest advantage under the stronger spatial and execution shift. These results suggest that ROAD-VLA does not require careful reference-checkpoint selection, supporting our use of the released RLinf PPO checkpoint as a fixed reference critic in the main experiments.

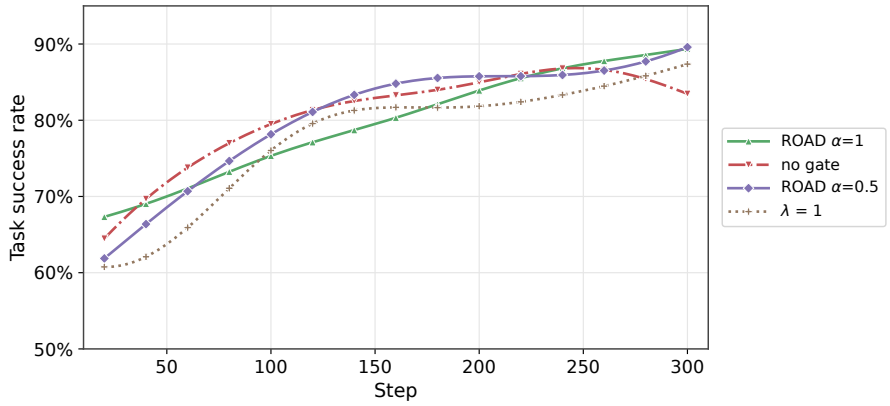


Figure 7: Training success under different ROAD-VLA configurations. We compare the default reference mixing ($\alpha = 0.5$), reference-only guidance ($\alpha = 1$), removal of the sign-agreement gate, and a larger distillation weight ($\lambda = 1$).

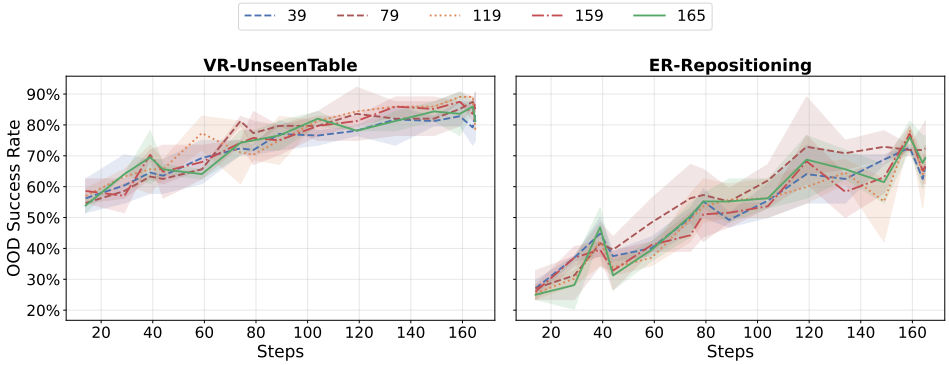


Figure 8: Tuning PPO warm-up checkpoint.

F.7 Signed Advantage Weight

Table 10: Ablation of signed versus positive-only advantage weighting on `CR-MultiObject`. All other hyperparameters are fixed. Results are mean ± standard deviation over three seeds.

Variant	ID Success (%)	OOD Success (%)
Positive-only ω_t	65.55 ± 8.12	57.02 ± 7.12
Signed ω_t	70.31 ± 6.63	60.94 ± 3.12

ROAD-VLA uses the signed advantage weight ω_t: positive advantages move the teacher toward the executed action, whereas negative advantages move it away. Table 10 compares this bidirectional update with a positive-only variant that suppresses negative-advantage distillation. Using signed weights improves ID success from 65.55% to 70.31% and OOD success from 57.02% to 60.94%, while also exhibiting lower variance across seeds. This supports the use of negative-advantage samples as corrective supervision rather than discarding them. We therefore use signed ω_t as the default throughout ROAD-VLA.

F.8 Computational Cost

We report the wall-clock training cost of ROAD-VLA relative to PPO for OpenVLA-7B and OpenVLA-OFT-7B on `ER-Repositioning`. For each backbone, PPO and ROAD-VLA use

the same training configuration and hardware setup with two NVIDIA H100 80GB GPUs, so the reported overhead isolates the additional computation introduced by ROAD-VLA.

Table 11: Computational cost of PPO and ROAD-VLA. Wall-clock time is measured using two NVIDIA H100 80GB GPUs. Overhead is reported relative to PPO with the same backbone.

Backbone	PPO (h)	ROAD-VLA (h)	Overhead
OpenVLA-7B	25.4	40.8	$1.61\times$
OpenVLA-OFT-7B	20.3	28.6	$1.41\times$

For OpenVLA-7B, PPO requires approximately **25.4 hours**, while ROAD-VLA requires **40.8 hours**, corresponding to a $1.61\times$ wall-clock overhead. For OpenVLA-OFT-7B, the corresponding training times are **20.3** and **28.6 hours**, yielding a $1.41\times$ overhead.

The additional cost comes from the distillation objective during policy optimization, including construction of the proximal teacher and evaluation of the corresponding native-space distillation loss.

G Additional Visualizations

Figures 9 and 10 show representative OOD rollouts. PPO fails under both execution and visual shifts, either drifting from the target or failing to complete the manipulation, while ROAD-VLA maintains task-directed behavior and succeeds. These examples qualitatively complement the robustness gains reported in the main results.

H Limitation and Future Work

Although ROAD-VLA improves online adaptation robustness over PPO in our evaluated settings, several limitations remain. First, our experiments are conducted on a focused set of language-conditioned manipulation tasks based on OpenVLA. While these tasks cover visual, compositional, and execution-level distribution shifts, they are still centered around pick-and-place style behaviors. Future work should evaluate ROAD-VLA on broader long-horizon, multi-stage, and multi-task manipulation settings.

Second, our evaluation is currently limited to simulation environments from the RL4VLA benchmark. Simulated perturbations allow controlled analysis of visual noise, object distractors, and execution shifts, but they may not fully capture real-world challenges such as calibration errors, lighting variation, contact dynamics, and hardware latency. Deploying ROAD-VLA on physical robots is an important next step.

Third, ROAD-VLA relies on advantage estimates and a reference PPO critic to construct the proximal teacher. Although the reference signal improves stability, its quality can affect the resulting teacher distribution. If the critic is poorly calibrated or becomes stale under large distribution shifts, the distillation target may become less reliable. Future work could explore critic-free teacher construction, uncertainty-aware advantage weighting, or adaptive mechanisms for updating the reference critic.

Fourth, the current method introduces several hyperparameters, including the distillation coefficient, logit perturbation scale, advantage clipping range, and reference critic mixing weight. While we provide a fixed configuration in our experiments, a more systematic sensitivity analysis is needed to understand how these choices affect stability and generalization. Future work could also investigate automatic tuning of the distillation strength based on policy divergence or critic agreement.

Finally, our current study focuses mainly on comparison with PPO. Additional baselines and ablations, such as text-guided distillation, uniform self-distillation, distillation without a reference critic, and different divergence objectives, would further clarify which components are most responsible for the observed gains. We believe that extending ROAD-VLA along these directions can lead to more reliable and scalable online adaptation for general-purpose VLA policies.

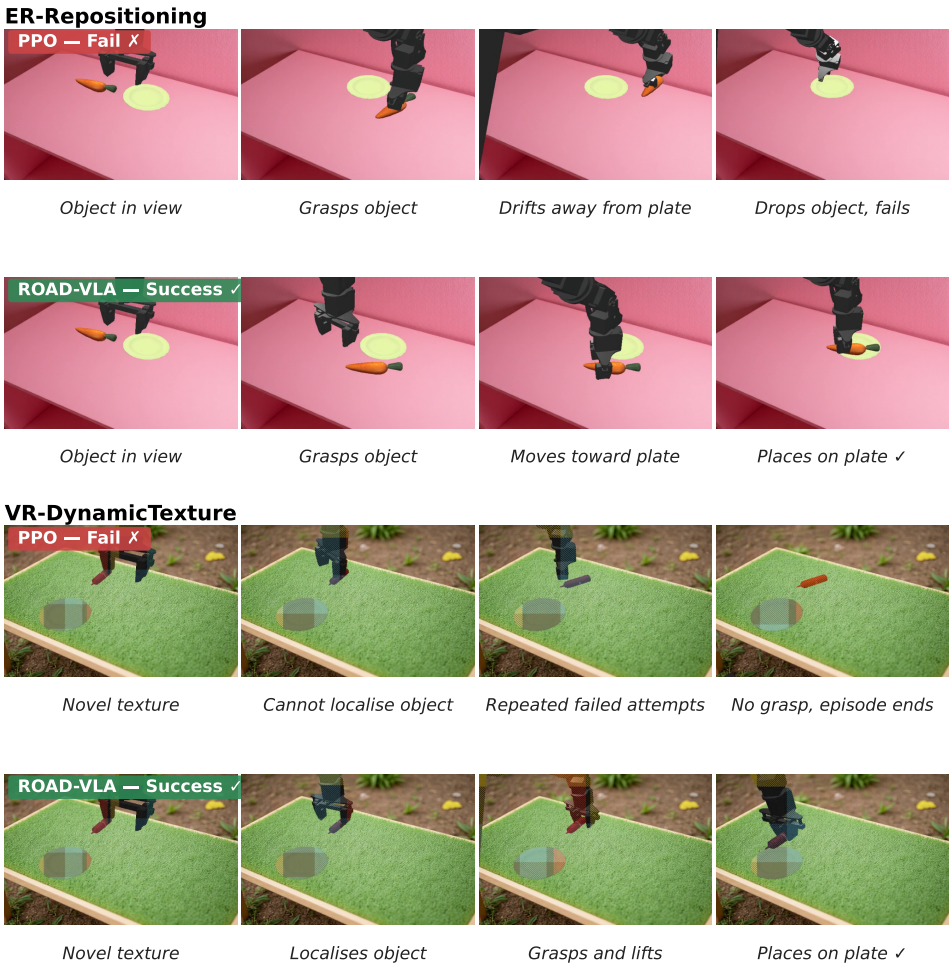


Figure 9: Qualitative rollout comparison under OOD conditions. In `ER-Repositioning`, PPO drifts toward the training-distribution receptacle location after grasping, while ROAD-VLA successfully adapts and places the object at the shifted target. In `VR-DynamicTexture`, PPO fails to reliably localize the object under unseen textures, whereas ROAD-VLA maintains stable perception and completes the task.

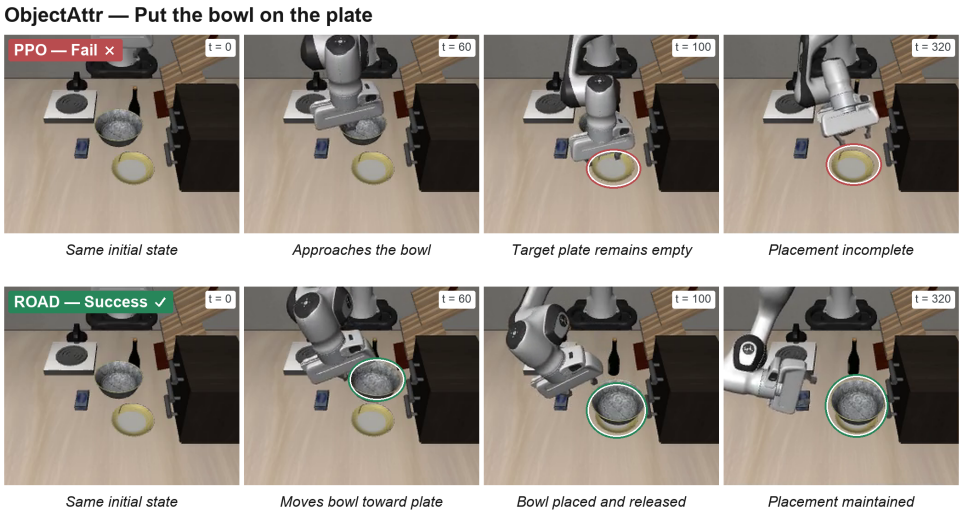


Figure 10: Qualitative rollout comparison under object-attribute shifts in `ObjectAttr`. Starting from the same initial state, PPO approaches the bowl but fails to place it on the target plate before the episode ends. ROAD-VLA successfully transfers and releases the bowl onto the plate, where it remains after the gripper retracts. Columns show matched environment steps $t \in \{0, 60, 100, 320\}$. Colored outlines highlight the relevant object and target regions.