EvoVLA: Self-Evolving Vision-Language-Action Model

Zeting Liu[1,2*], Zida Yang[1,3*], Zeyu Zhang[1*,†], and Hao Tang[1,4‡]

[1]School of Computer Science, Peking University, Beijing, China
[2]College of Automation, Qingdao University, Qingdao, China
[3]Guanghua School of Management, Peking University, Beijing, China
[4]Beijing Academy of Artificial Intelligence, Beijing, China
[*]Equal contribution. [†]Project lead. [‡]Corresponding author: bjdxtanghao@gmail.com.

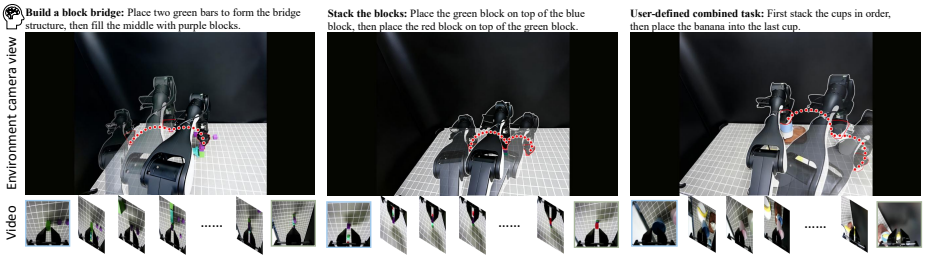

Fig. 1: Overview of EvoVLA on long-horizon and multi-stage robotic manipulation tasks. Our framework suppresses stage hallucination for robust multi-stage execution. We evaluate on Discoverse-L with three simulation tasks—**Bridge**, **Jujube-Cup**, and **Stack**—and on real robots with Sim2Real transfer on the same three tasks plus an on-robot **Insert (OOD)** task. Across these settings, EvoVLA improves success rates and sample efficiency while reducing stage hallucination.

Abstract. Vision-Language-Action (VLA) policies fine-tuned with RL and frozen VLM reward evaluators can produce observations that *look* correct without physically completing the task—a phenomenon we formalize as **stage hallucination**. To suppress it, we propose **EvoVLA**, a self-supervised framework with three synergistic modules: **Stage-Aligned Reward (SAR)** penalizes near-miss states via counterfactual hard negatives in CLIP-based scoring; **Pose-Based Object Exploration (POE)** grounds curiosity in relative gripper–object pose; and **Long-Horizon Memory** uses selective context retrieval with gated fusion to stabilize credit assignment. We also introduce **Discoverse-L**, a long-horizon benchmark (3 tasks, 18–74 stages) with a formal Hallucination Rate metric. Under matched backbones and budgets, EvoVLA achieves 69.2% success (+10.2 over OpenVLA-OFT), 1.5× sample efficiency, and reduces hallucination from 38.5% to 14.8%; real-robot deployment attains 54.6% (+11.0) across four tasks, confirming Sim2Real transfer.

Keywords: Vision-Language-Action Model · Long-horizon Manipulation · Self-supervised Reinforcement Learning · Robotics

1 Introduction

Vision-Language-Action (VLA) models unify perception, language, and control in a single backbone, enabling generalist robot policies [2,7,20,32,50]. A common recipe for adapting these models is reinforcement learning (RL) fine-tuning with a frozen vision-language model (VLM) as an external reward evaluator—for example, scoring each observation against a text description of the target state via CLIP image-text similarity. This setup works for short-horizon skills, but when tasks span dozens of semantically distinct stages, a failure mode emerges that we term **stage hallucination**: the policy learns to produce observations that score highly on the VLM evaluator *without actually completing the task*—e.g., hovering the gripper above a block may be scored as "object grasped" while making no physical contact. The root cause is that appearance-based evaluators cannot distinguish genuine completion from visually similar near-miss states, so the policy converges to deceptive local optima with inflated success despite zero real task progress.

Why does this problem persist? We trace it to two gaps that current methods leave open. **(1) Reward signal gap.** Standard VLM-based rewards compare images against only positive/negative text descriptions, leaving a wide margin for exploitation. Without counterfactual reward signals that penalize near-miss states, policies can discover visual shortcuts. Self-supervised curiosity [4,30] can provide denser feedback but relies on pixel-level novelty that collapses in cluttered manipulation scenes. **(2) Memory fragility.** Long-horizon tasks require persistent stage tracking, yet retaining and leveraging task history remains an open challenge for VLA-based manipulation [37], and compressing history can blur stage-specific details, amplifying credit mis-assignment.

These observations motivate **EvoVLA** (Figure 2), a framework that directly counteracts stage hallucination through three synergistic modules. **Stage-Aligned Reward (SAR)** introduces counterfactual hard negatives—textual descriptions of near-miss states—into CLIP-based evaluation, penalizing the policy for producing "looks-correct-but-isn't" observations. **Pose-Based Object Exploration (POE)** replaces fragile pixel-based curiosity with intrinsic rewards grounded in the relative gripper-to-object pose, ensuring exploration is driven by task-relevant geometric changes rather than visual distractors. **Long-Horizon Memory** selects Top-K history items via attention, fuses them through a learned gate, and modulates intrinsic shaping so that progress signals remain stable across extended episodes. The key insight is that *no single component suffices*: SAR corrects the reward landscape but cannot guide exploration; POE provides geometrically grounded exploration but needs stable temporal context; Memory stabilizes learning but requires accurate rewards to gate. Their synergy is what suppresses hallucination end-to-end.

We do not introduce a new policy backbone or increase model scale. Instead, we initialize the policy from OpenVLA-OFT and keep the policy architecture fixed across all comparisons, while fine-tuning the policy with PPO. The CLIP-based evaluator remains frozen and is used only as an external reward source.

We evaluate EvoVLA on **Discoverse-L**, a long-horizon manipulation benchmark we construct with three multi-stage tasks (18–74 stages) on the AIRBOT-Play platform, providing the community with ground-truth stage annotations for systematic study of hallucination and memory in long-horizon manipulation.
Contributions.

- **Problem formalization.** We formally define **stage hallucination**—a failure mode where VLA policies exploit VLM reward evaluators in long-horizon RL—and trace its root causes to reward signal gap and memory fragility. To our knowledge, this is the first systematic study of stage-level hallucination in long-horizon VLA reinforcement fine-tuning with VLM-based rewards.
- **Method.** We propose **EvoVLA**, which counters hallucination through three synergistic modules: SAR (counterfactual reward correction), POE (pose-grounded exploration), and Long-Horizon Memory (selective context with gated shaping). No single module suffices; removing any one forfeits $\geq$2.4 success points (Table 2).
- **Benchmark and empirical results.** We introduce **Discoverse-L** (3 tasks, 18–74 stages) with ground-truth stage annotations and the Hallucination Rate metric. Under matched backbone and training budget, EvoVLA achieves **69.2%** sim success (+10.2), **54.6%** real-robot success (+11.0), 1.5× sample efficiency, and reduces HR from 38.5% to **14.8%**.

2 Related Work

VLA Models and RL Fine-Tuning. VLA models [2, 3, 7, 8, 20, 23, 32, 46, 50] unify perception, language, and control into end-to-end robot policies, with recent work on optimized fine-tuning [19] improving efficiency via action chunking and continuous regression. Related embodied vision-language systems, including 3D scene understanding and navigation models [15–17, 24, 40, 41], further highlight the importance of reasoning and memory in interactive environments, but they do not directly address long-horizon VLA reinforcement fine-tuning under VLM-based reward evaluators. RL-based post-training [6, 10, 11, 13, 14, 21, 22, 25, 35, 39, 48] further improves VLAs beyond imitation, but depends critically on reward quality. A common practice is to use VLM-based evaluators (e.g., CLIP image-text similarity) as reward functions, yet this introduces the stage hallucination problem we study: policies exploit coarse appearance-level reward signals without physical task completion. To our knowledge, existing works do not explicitly diagnose this failure mode in long-horizon VLA RL or provide a targeted mitigation mechanism. EvoVLA provides a systematic characterization of stage hallucination and a principled framework to suppress it.
Intrinsic Motivation and Self-Supervised RL. Self-supervised RL generates dense intrinsic rewards to alleviate sparse-reward bottlenecks. ICM [30] and RND [4] define curiosity via prediction errors, Plan2Explore [36] uses trajectory ensemble disagreement, and RaMP [5] enables zero-shot transfer with random features. However, all these methods operate on pixel-level representations and

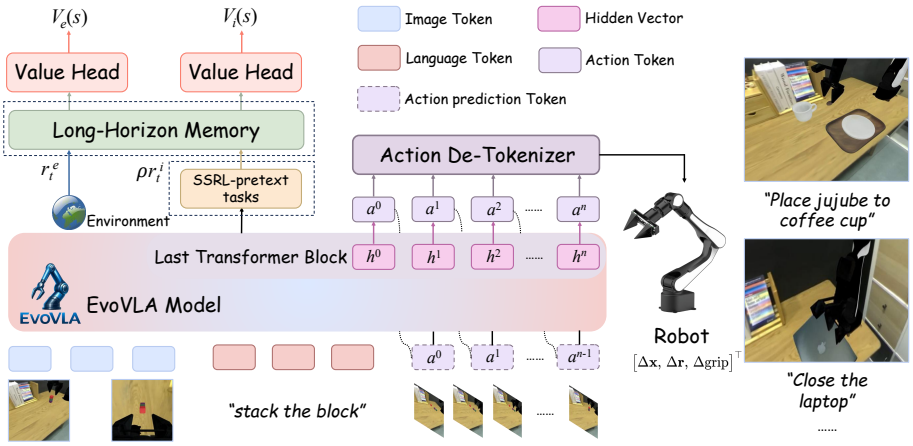

Fig. 2: EvoVLA overview. Built on OpenVLA-OFT backbone, EvoVLA integrates three modules: Stage-Aligned Reward (SAR) with hard negatives and temporal smoothing, Pose-Based Object Exploration (POE) via world models, and Long-Horizon Memory with context selection and gated fusion. The framework couples with Discoverse-L for training and deploys to real robots.

are brittle in cluttered manipulation scenes where lighting, camera motion, and irrelevant objects dominate novelty signals. EvoVLA introduces *pose-grounded* curiosity that operates in the task-relevant geometric space of relative gripper-object poses, providing robust intrinsic signals immune to visual distractors.

Reward Shaping and Memory for Long-Horizon Tasks. Potential-based shaping [12, 27] adjusts returns without changing optimal policies. VLM-based reward generation [26, 42, 44, 49] leverages pretrained vision-language models as reward sources, but typically uses only positive-negative contrast without counterfactual near-miss detection. Memory-augmented systems including Reflexion [38] and JARVIS-1 [45] address long-horizon reasoning but rely on non-end-to-end frameworks or short context windows. Recent memory-augmented VLA systems, such as MemoryVLA [37], demonstrate the value of incorporating perceptual-cognitive history for manipulation. EvoVLA differs in that its memory is explicitly tied to reward-level modulation: it retrieves Top-K stage-relevant history items and gates intrinsic progress rewards to stabilize credit assignment in long-horizon RL. Unlike generic contrastive regularization, SAR hard negatives are stage-conditional and directly coupled to online stage gating and PPO return shaping.

3 EvoVLA: Method

3.1 Overview

We first formalize the core problem before describing EvoVLA's architecture.

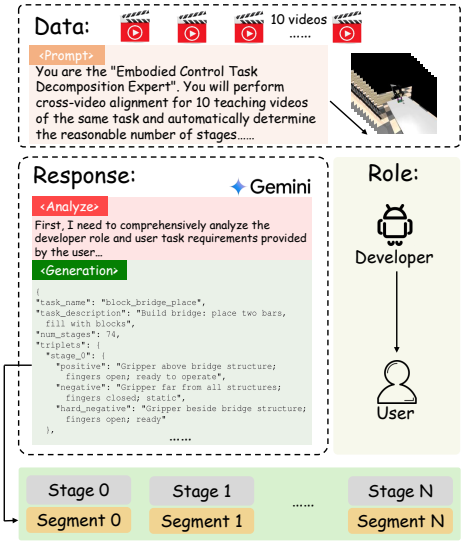

Fig. 3: EvoVLA Data Engine. Aligned with Discoverse-L and the video-driven stage discovery pipeline to close the data–reward–policy loop.

Stage Hallucination: Problem Definition. Consider a VLA policy π fine-tuned via RL with a VLM-based reward evaluator (e.g., CLIP image-text similarity). For a multi-stage task with K stages, the evaluator computes a score $u_k(t) = \langle \phi_{\text{img}}(o_t), \phi_{\text{text}}(T_k^+) \rangle$ comparing the current observation o_t to a positive text description T_k^+ of stage k completion. *Stage hallucination* occurs when π produces observations o_t that achieve high evaluator scores $u_k(t) > \theta$ while failing to satisfy the actual physical completion criterion $c_k(t) = 0$. This arises because appearance-based evaluators cannot distinguish genuine completion from visually similar near-miss states (e.g., "gripper hovering above object" vs. "gripper grasping object"). Hallucination causes the policy to converge to deceptive local optima that inflate reported progress without real task completion.

EvoVLA Pipeline. EvoVLA counteracts hallucination by correcting the reward landscape, grounding exploration, and stabilizing memory. It couples a task-aligned data pipeline (Figure 3) with long-horizon control modules (Figure 2). Discoverse-L supplies three AIRBOT-Play tasks with 50 scripted rollouts each; we store trajectories in RLDS, compute task-aligned normalization, and run a multi-step prompting workflow with a VLM (Gemini 2.5 Pro [9] or open-source alternatives such as Llama-3-Vision/Qwen2-VL; prompt templates and validation details are provided in the supplementary material) to produce unified stage dictionaries of positive, negative, and hard-negative predicates. The same semantics supervise simulation training and Sim2Real deployment; for unseen tasks we regenerate the dictionary from 50 teleoperated demonstrations.

The architecture centers on three interacting modules. SAR scores image-text triplets with CLIP [33] contrast and temporal smoothing, delivering stage-

faithful dense rewards. POE grounds curiosity in relative gripper-to-object poses via paired forward/inverse world models to reduce spurious novelty from visual distractors. Long-Horizon Memory selects salient history tokens via attention-based context selection and gates intrinsic shaping so progress signals remain stable. SAR, POE, and Long-Horizon Memory plug into an OpenVLA-OFT policy initialized from pretrained weights; the policy is fine-tuned with PPO together with trainable auxiliary modules (world models, memory gates), while the CLIP reward encoder stays frozen.

Training follows a modular recipe: optionally warm-start with supervised learning on the rollouts, then apply PPO [35] with task-aligned normalization while SAR, POE, and memory remain active. The following subsections detail each component.

Notation. We denote stage index by $k \in \{1, \ldots, K\}$, active stage by κ_t, current observation by o_t, and stage triplets by (T_k^+, T_k^-, T_k^{h-}). Reward terms follow Eq. (1): r_t^e (extrinsic), $r_t^i = r_t^{\text{stage}} + r_t^{\text{cur}} + r_t^{\text{prog}}$ (intrinsic, comprising SAR, POE curiosity, and memory-gated progress).

3.2 Self-Supervised Reinforcement Learning

The central design principle of EvoVLA is to replace sparse, hackable external rewards with dense intrinsic signals that are semantically faithful to task progress. Unlike prior curiosity methods [4, 30] that operate on raw pixels, our intrinsic rewards are grounded in (i) stage-aligned contrastive evaluation and (ii) geometric pose structure. The combined reward at each timestep interpolates extrinsic and intrinsic terms:

$$\tilde{r}_t = (1 - \rho)\, r_t^e + \rho\, r_t^i. \tag{1}$$

where r_t^e is the sparse external task reward, $r_t^i = r_t^{\text{stage}} + r_t^{\text{cur}} + r_t^{\text{prog}}$ is the aggregate intrinsic reward (components detailed below), and ρ is the intrinsic weight (we use $\rho = 0.6$, selected via sensitivity analysis in the supplementary material).

Stage-Aligned Reward. We evaluate stage completion through image-text contrastive learning, producing dense reward signals to guide policy learning. The key insight is that standard positive-negative contrastive learning is insufficient, so we need hard negatives that capture near-miss states to prevent policies from exploiting visual shortcuts.

Triplet text generation. For each stage k, an automated prompting workflow (e.g., Gemini 2.5 Pro [9] or Llama-3-Vision; templates and prompt compatibility are provided in the supplementary material) generates a stage dictionary providing a triplet: positive T_k^+ (completion state), mutually-exclusive negative T_k^-, and counterfactual hard negative T_k^{h-} (near-miss). The hard negative uses spatial/contact predicates (e.g., "grasping non-target object") rather than visual features (e.g., "grasping blue cup"), ensuring robustness to appearance variations.

Image-text contrastive scoring. Given VLM encoder ϕ (CLIP [33]), we compute stage-aligned scores by comparing image observations against the triplet

text descriptions:

$$u_k(t) = \sigma\Big(\tau\,[s_k^+(t) - \max\{s_k^-(t), s_k^h(t)\}]\Big), \quad u_k(t) \in [0,1]. \tag{2}$$

where $s_k^+(t) = \langle \phi_{\text{img}}(o_t), \phi_{\text{text}}(T_k^+)\rangle$ (similarly for $s_k^-(t), s_k^h(t)$), τ is the temperature parameter, and σ is the sigmoid function.

Why hard negatives suppress hallucination. Without T_k^{h-}, policies can exploit the coarse "task done" vs. "task not started" distinction by producing observations with high $s_k^+(t)$ (e.g., hovering near the target) without physical completion. The $\max\{s_k^-(t), s_k^h(t)\}$ term in Eq. (2) penalizes near-miss states (high $s_k^h(t)$) even when $s_k^+(t)$ is high. Although the CLIP encoder is frozen, $u_k(t)$ shapes policy optimization through PPO returns built from Eq. (1), so observations matching hard negatives are consistently down-weighted. Prior work [42, 44] supports that image-text similarity captures spatial relations, and our ablation (Table 2) confirms SAR reduces hallucination by 15.1 points.

Temporal smoothing. To stabilize score fluctuations and prevent noisy reward signals, we maintain a running average $\bar{u}_k$ that is updated smoothly over time:

$$\bar{u}_k(t) = (1-\alpha)\bar{u}_k(t-1) + \alpha u_k(t). \tag{3}$$

where α is the smoothing coefficient. The stage reward is then computed as the progress in these smoothed scores:

$$r_t^{\text{stage}} = \bar{u}_{\kappa_t}(t) - \bar{u}_{\kappa_t}(t-1). \tag{4}$$

where κ_t is the current active stage. We advance to stage $\kappa_t + 1$ when a sliding window (size $m = 8$) of recent r^{stage} values exceeds a threshold, indicating sustained progress in the current stage (implementation details are in the supplementary material). This approach filters out transient score fluctuations while preventing premature transitions. To ensure semantic robustness, these stage dictionaries are subjected to automated validation filters, and stress tests (Supp. Tab. 3) confirm that gains depend on correct stage-conditional semantics rather than any specific VLM backend.

Pose-Based Object Exploration. Pixel-based curiosity suffers from spurious novelty, with exploration driven by irrelevant changes, such as lighting variations or camera motion, rather than task progress. POE addresses this by grounding curiosity in task-relevant interaction dynamics between the object and the parallel-jaw gripper, focusing on how the gripper's motion and contact evolve during manipulation.

Latent manipulation dynamics. We represent the manipulation state as $z_t = \psi(T_{\text{ee}}^{-1} T_{\text{obj}}) \in \mathbb{R}^6$, where T_{ee} is the end-effector pose, T_{obj} is the object pose, and ψ converts the relative transformation to a 6D vector in axis-angle plus translation representation (3D translation + 3D axis-angle rotation). We train forward and inverse models:

$$\hat{z}_{t+1} = f_\phi(z_t, a_t), \quad \hat{a}_t = g_\psi(z_t, z_{t+1}). \tag{5}$$

These lightweight MLP models (2×256 units) learn the dynamics of object manipulation without being distracted by visual nuisances.

Task-relevant intrinsic rewards. The curiosity reward encourages exploring states where the forward model makes large prediction errors, while the progress reward captures learning improvements:

$$\begin{aligned} r_t^{\text{cur}} &= \frac{\eta}{2}\|\text{sg}(\hat{z}_{t+1}) - z_{t+1}\|_2^2, \\ r_t^{\text{base}} &= \text{ReLU}\big(\overline{\mathcal{L}_F}(t-1) - \overline{\mathcal{L}_F}(t)\big). \end{aligned} \tag{6}$$

where $\text{sg}(\cdot)$ denotes stop-gradient (preventing policy from exploiting model errors), $\overline{\mathcal{L}_F}$ is the forward loss with temporal smoothing, and $\eta = 1.0$ is the curiosity scale. By focusing on geometric task structure, POE substantially reduces spurious exploration caused by lighting changes and camera motion. The base progress term r_t^{base} is forwarded to Long-Horizon Memory for memory-conditioned gating before contributing to the combined reward.

Role and scope of r_t^{cur}. The curiosity reward r_t^{cur} is *not* designed to be a standalone task reward—a relative pose alone is insufficient for complex multi-stage manipulation, which additionally requires correct contact forces, sequential ordering, and stable placement. Instead, r_t^{cur} serves as a targeted exploration bonus (weighted by $\rho = 0.6$, Eq. 1) that steers the policy toward geometrically novel gripper-object configurations, accelerating the discovery of task-relevant interactions. Task success is ultimately measured by the extrinsic reward r_t^e and the stage-aligned reward r_t^{stage} (which evaluates semantic completion via contact and spatial predicates). The ablation in Table 2 quantifies the complementary role: adding POE to the full model contributes +3.1 SR / -4.7 HR, confirming that geometric exploration is most effective when combined with stage-aligned evaluation and memory stabilization.

3.3 Long-Horizon Memory

Motivation. In multi-stage manipulation, the policy must track which stages are completed and adapt its behavior accordingly. Naive averaging or truncation of history can blur stage-specific details, causing the policy to lose track of progress and amplifying hallucination via credit mis-assignment. We address this by treating *memory as selective context*: at each step we retrieve the Top-K most relevant history items, fuse them with the current latent via a learned gate, and perform lightweight write-back with utility-guided eviction. This preserves phase-specific information while keeping the store compact.

Context Selection via Attention. We extract a latent representation $x_t \in \mathbb{R}^d$ from the OpenVLA-OFT backbone at each step (by pooling over the LLM's hidden states), and we maintain a memory store $\mathcal{M} = \{m_i \in \mathbb{R}^d\}_{i=1}^L$ with timestamps $\{\tau_i\}$, where L is the store capacity. To select relevant history, we

define query, key, and value projections:

$$\begin{aligned} q_t &= W_q x_t, \\ k_i &= W_k\big(m_i + \mathrm{PE}(\tau_i)\big), \\ v_i &= W_v m_i, \end{aligned} \tag{7}$$

where $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$ are learnable linear maps, and $\mathrm{PE}(\cdot)$ is a sinusoidal temporal positional encoding. Attention scores and the selected index set are computed as:

$$\begin{aligned} a_i &= \frac{\exp(\langle q_t, k_i\rangle/\sqrt{d})}{\sum_{j=1}^{L}\exp(\langle q_t, k_j\rangle/\sqrt{d})}, \\ \mathcal{S}_t &= \mathrm{TopK}\big(\{a_i\}_{i=1}^{L}, K\big), \end{aligned} \tag{8}$$

where K selects the most relevant history items. We prepend $\{v_i \mid i \in \mathcal{S}_t\}$ to the current sequence as independent context tokens, replacing static adjacent-averaging consolidation used in prior work.

Gated Fusion and Reward Modulation. Let $\hat{h}_t = \sum_{i\in\mathcal{S}_t} a_i v_i$ be the weighted context embedding. We fuse it with the current latent x_t via a learned gate:

$$\begin{aligned} g_t^{\mathrm{mem}} &= \sigma\big(w_g^\top[\hat{h}_t; x_t]\big), \\ \tilde{x}_t &= (1 - g_t^{\mathrm{mem}})x_t + g_t^{\mathrm{mem}}\hat{h}_t. \end{aligned} \tag{9}$$

where $w_g \in \mathbb{R}^{2d}$ is a learnable weight vector and $[\cdot;\cdot]$ denotes concatenation. The gate g_t^{mem} modulates how much history context influences the current representation. The memory gate also modulates the base progress reward r_t^{base}:

$$r_t^{\mathrm{prog}} = g_t^{\mathrm{mem}} \cdot r_t^{\mathrm{base}}, \tag{10}$$

suppressing spurious progress signals when history context indicates unstable manipulation patterns (e.g., repeated failures, oscillations). We block gradients through g_{t-1}^{mem} to prevent reward hacking. Memory update writes back the fused representation $\tilde{x}_t$ only to the selected items $\mathcal{S}_t$, and eviction removes the least useful entry based on a utility score combining usage frequency, recency, and redundancy with existing items.

POE Integration and Effect. In contrast to Titans-MAC [1] and MemoryVLA [37], Long-Horizon Memory uses selective context with gated fusion for reward-level modulation. The core mechanism operates on backbone latents x_t and history items m_i via the attention and gating mechanisms above. When combined with POE (full model, Table 2 row 5 vs. row 4 standalone), POE's geometric pose tracking provides complementary grounding that enhances memory's ability to distinguish task-relevant manipulation contexts from spurious visual changes, further improving robustness. This mechanism complements SAR temporal smoothing and aligns with the training objective $\mathcal{L}$ (11). Empirically, Long-Horizon Memory in standalone mode contributes +2.4 points of success and a 3.9-point hallucination reduction, with POE integration adding an additional +3.1 points, highlighting the value of both selective context and geometric grounding.

3.4 Training Objective

The full training objective combines policy gradient loss, value losses, and world model losses:

$$\mathcal{L} = \mathcal{L}_{\text{PPO}}(\tilde{r}) + \lambda_F \mathcal{L}_F + \lambda_I \mathcal{L}_I - \lambda_{\text{ent}} H(\pi). \tag{11}$$

where $\mathcal{L}_F$ and $\mathcal{L}_I$ are the MSE losses for the forward and inverse world models, $H(\pi)$ is policy entropy, and $\lambda_F, \lambda_I, \lambda_{\text{ent}}$ weight the auxiliary objectives (complete values are listed in the supplementary material). We use PPO [35] with dual critics V_e, V_i for the returns induced by r_t^e and r_t^i, fusing advantages in the same proportion as Eq. (1): $A_t = (1-\rho)A_t^e + \rho A_t^i$, via GAE [34]. World models and the memory-conditioned GRU are trained via separate optimizers with stop-gradient to prevent reward exploitation.

4 Discoverse-L Benchmark

Benchmark Overview. Discoverse-L builds on the DISCOVERSE simulator [18] with the AIRBOT-Play platform, offering 3 multi-stage manipulation tasks with varying difficulty: (1) *Bridge* (74 stages): place two bars to form a bridge structure, then fill with multiple blocks, the most complex task requiring precise multi-object coordination; (2) *Stack* (18 stages): stack three colored blocks in sequence, demanding accurate alignment; (3) *Jujube-Cup* (19 stages): place a jujube fruit into a cup and move the cup onto a plate, a relatively simpler setup with fewer contact-critical stages. Stage counts reflect our Gemini [9]-prompted dictionaries; the DISCOVERSE simulator provides fine-grained ground-truth annotations (e.g., 79 for Bridge) used only for validation of stage-discovery coverage. The 74-stage Bridge dictionary merges five simulator micro-adjustments into semantically equivalent steps (93.7% recall), so reward supervision remains faithful while prompts stay compact. The 74-stage dictionary is used for SAR supervision and stage gating, while simulator-native stage annotations are used only for evaluation diagnostics. We generate 50 simulator rollouts per task with randomized initial conditions; normalization and protocol details are provided in the supplementary material.

5 Experiments

5.1 Experimental Design and Evaluation

For comparability across methods on Discoverse-L, we use matched training budgets/protocols and the same task-aligned normalization setup. Behavior-cloning warm-start, when enabled in a setting, is applied consistently across compared methods before interactive PPO training. Evaluation uses 50 episodes per task with held-out seeds, identical camera setups, and fixed episode budgets. For EvoVLA and all reported ablations, simulator stage labels are never used for training and are used only for evaluation diagnostics (e.g., HR computation).

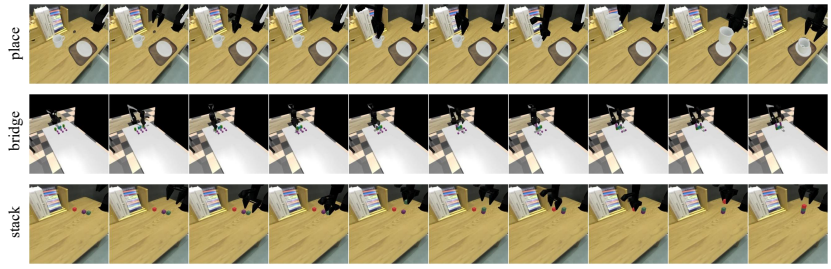

Fig. 4: Simulation rollouts. Each row shows a full episode for Jujube-Cup, Bridge, and Stack with dual third-person/wrist views; columns denote temporal progression.

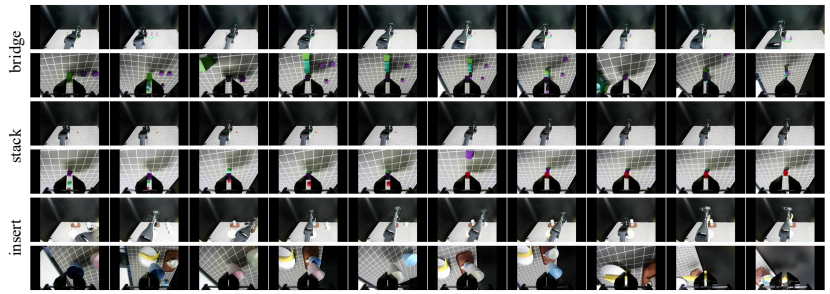

Fig. 5: Real-world rollouts. Each row shows stack, bridge, and insert executions with synchronized third-person and wrist cameras in the real world.

Implementation Details. We build upon OpenVLA-OFT [19], which extends OpenVLA [20] with parallel action decoding, action chunking, and L1 regression loss. The backbone uses multi-view ViT towers (SigLIP [47] + DINOv2 [29]) and Llama-2 7B [43]; SAR uses a separate frozen CLIP ViT-B/16 encoder [33]. Training uses 8 parallel DISCOVERSE environments, 2M timesteps per seed, and 3 random seeds; we set $\theta = 0.7$ from sensitivity analysis. Stage dictionaries follow the same video-driven workflow in simulation and on-robot settings, and task-aligned normalization is injected at initialization. Full hyperparameters, hardware, prompts, and filtering details are provided in the supplementary material.

Evaluation Metrics. *Success Rate (SR):* Percentage of episodes completing the task within 400 steps. We cap episodes at 400 to accommodate the most complex task (Bridge, 74 stages) while preventing indefinite rollouts; successful runs typically finish in 50–100 steps for simpler tasks (Stack, Jujube-Cup) and 150–250 steps for Bridge.

Sample Efficiency: Number of environment timesteps required to reach a 50% success rate.

Hallucination Rate (HR): The fraction of high VLM scores corresponding to failed actual completion. Formally, let $u_k(t) \in [0, 1]$ be the VLM score for stage k at time t, and $c_k(t) \in \{0, 1\}$ be the ground-truth completion indicator. Hallu-

Table 1: Discoverse-L benchmark (3 tasks; mean of 3 seeds, 50 evaluation episodes). All models use task-aligned normalization; EvoVLA adds SAR, POE, and Long-Horizon Memory to OpenVLA-OFT. The last row highlights improvements over OpenVLA-OFT.

Model	Bridge	Jujube-Cup	Stack	Avg.
Octo [8]	24.8	33.7	29.1	29.2
OpenVLA [20]	32.6	42.0	37.5	37.4
π_0 [2,32]	41.1	50.5	46.2	45.9
π_0-FAST [31,32]	47.4	56.9	52.8	52.4
OpenVLA-OFT [19]	54.1	63.5	59.4	59.0
EvoVLA (Ours)	**65.3**	**72.6**	**69.7**	**69.2**
Δ vs. OFT	**+11.2**	**+9.1**	**+10.3**	**+10.2**

Table 2: Component ablation. Progressive addition (averaged across 3 tasks, 3 seeds). Starting from OpenVLA-OFT baseline, we cumulatively add: hard negatives, temporal smoothing, Long Horizon Memory, and POE.

Configuration	SR (%)	HR (%)
OpenVLA-OFT (Baseline)	59.0	38.5
+ Hard Negatives (T^{h-})	61.8	31.2
+ Temporal Smoothing	63.7	23.4
+ Long-Horizon Memory	66.1	19.5
+ POE (full EvoVLA)	**69.2**	**14.8**

cination rate (HR) is:

$$\mathrm{HR} = \frac{\mathbb{E}\big[\mathbb{1}\{u_k(t) > \theta \wedge c_k(t) = 0\}\big]}{\mathbb{E}\big[\mathbb{1}\{u_k(t) > \theta\}\big]}. \tag{12}$$

where $u_k(t) > \theta$ (we use $\theta = 0.7$) indicates "high VLM score" and $c_k(t) = 0$ indicates failed completion, verified using task success signals from the DISCOVERSE simulator. The expectation averages over evaluation episodes, timesteps, and active stages. We emphasize that $c_k(t)$ is obtained solely at evaluation time from DISCOVERSE stage-event triggers to compute the metric; it is never used for training or model selection, ensuring methodological independence. Furthermore, our sensitivity analysis (Fig. 8) confirms that the HR ranking remains stable across threshold choices $\theta \in [0.65, 0.75]$, demonstrating that this metric is not an artifact of task-specific tuning.

Real-World Deployment. Beyond simulation, we validate EvoVLA on a physical AIRBOT-Play robot (details in Section 5.3).

5.2 Simulation Results

Baselines. We benchmark against state-of-the-art VLA backbones trained on Discoverse-L: Octo [8], OpenVLA [20], π_0 and π_0-FAST [31,32], and OpenVLA-OFT [19]. EvoVLA is obtained by augmenting the OpenVLA-OFT architecture with our SAR module, POE module, Long-Horizon Memory module, and task-aligned normalization to address stage hallucination. Because backbone family and training budgets are matched, observed gains are attributable to these added learning signals rather than increased model scale.

Main Simulation Results. Table 1 summarizes quantitative comparisons against the baselines. Starting from Octo (29.2%) and OpenVLA (37.4%), progressively stronger models achieve 45.9% (π_0), 52.4% (π_0-FAST), and 59.0% (OpenVLA-OFT). EvoVLA lifts average success to 69.2%, improving upon the strongest

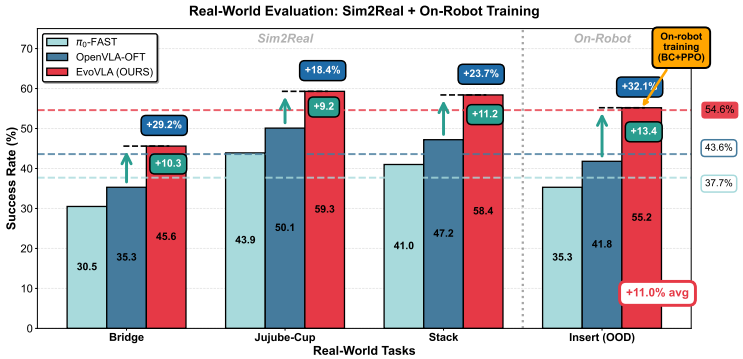

Fig. 6: Real-world evaluation. Green arrows: absolute gains over OpenVLA-OFT (+9.2–+13.4 points). Blue labels: relative gains (+18.4%–+32.1%). Horizontal dashed lines mark model averages (37.7%, 43.6%, 54.6%); the vertical dashed line splits Sim2Real transfers (left) from the on-robot Insert task (right).

baseline by +10.2 absolute points. Per-task gains are consistent: Bridge (+11.2), Jujube-Cup (+9.1), and Stack (+10.3), with the largest gain on the longest-horizon Bridge task. Figures 4 and 5 show representative simulation and real-world rollout sequences.

Sample Efficiency. EvoVLA reaches a 50% average success rate in approximately 6×10^5 environment steps (averaged over 3 seeds), whereas OpenVLA-OFT requires approximately 9×10^5 steps, demonstrating 1.5× better sample efficiency. This acceleration stems from dense intrinsic feedback: SAR provides stage-level guidance at every timestep, POE grounds exploration in task-relevant geometric structures, and Long-Horizon Memory stabilizes learning signals.

Hallucination Reduction. Simulation rollouts report a 14.8% Hallucination Rate versus 38.5% for OpenVLA-OFT, i.e., a 23.7-point drop (Table 2). The precise contribution of each module—hard negatives, temporal smoothing, Long-Horizon Memory, and POE—is analyzed in Section 5.4, so we defer further discussion there.

5.3 Real-World Evaluation

We deploy EvoVLA on a physical AIRBOT-Play robot (additional rollout snapshots in Figure 5) and compare against π_0-FAST [31,32] and OpenVLA-OFT [19] under matched real-world settings. Evaluation includes (i) direct Sim2Real transfer on the three Discoverse-L tasks and (ii) an unseen insertion task trained on-robot from 50 teleoperated demonstrations. In the on-robot setting, the policy is BC-warm-started and PPO-fine-tuned for roughly 5k steps; SAR (CLIP wrist-view queries), POE (AprilTag-based relative poses [28]), and memory gating remain active throughout training.

Main Real-World Results. Figure 6 shows that EvoVLA attains 54.6% average success across the three Sim2Real tasks plus the on-robot insertion task. Each model-task pair is evaluated over multiple batches of at least 20 valid

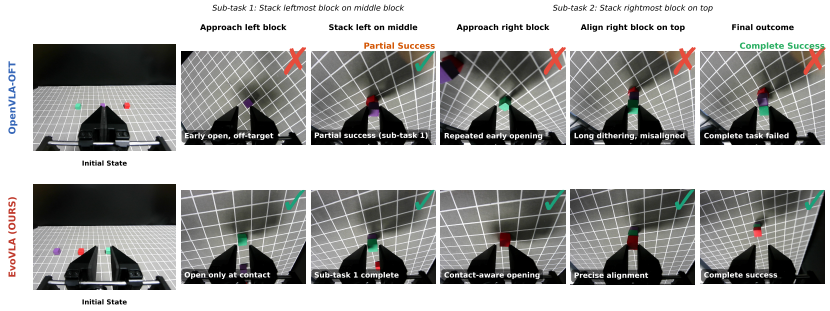

Fig. 7: Stack task qualitative comparison. Columns (1–5) cover sub-task 1 (left block onto middle) then sub-task 2 (right block on top). Top: OpenVLA-OFT (cross markers) opens before contact, dithers, misaligns, and drops the block. Bottom: EvoVLA (check marks) delays opening until contact, aligns within a few corrections, and leaves a stable stack.

physical trials. EvoVLA improves over OpenVLA-OFT on all four tasks by +9.2 to +13.4 points (+11.0 overall), with the largest gain on insertion (55.2% vs. 41.8%). Supplementary rollouts and qualitative comparisons further show fewer premature releases and more stable contact-aware placement. Figure 7 presents a qualitative comparison on the sequential block-stacking task. The baseline, OpenVLA-OFT (top row), exhibits suboptimal manipulation behaviors characterized by premature gripper release, oscillatory dithering, and misalignment, ultimately resulting in task failure. In contrast, EvoVLA (bottom row) demonstrates robust and precise control. By employing contact-aware execution, it delays the gripper release until physical contact, achieves accurate alignment with minimal corrective adjustments, and successfully completes a stable stack, highlighting its superior planning and execution capabilities.

5.4 Ablation Studies

Table 2 details cumulative gains: Hard Negatives (+2.8 SR/-7.3 HR), Smoothing (+1.9/-7.8), Memory (+2.4/-3.9), and POE (+3.1/-4.7) sum to +10.2 SR and -23.7 HR over OpenVLA-OFT. Ablating any component incurs $\geq$2.4 SR loss.

Isolated Component Analysis. Single modules achieve 63.7% (SAR), 61.4% (Memory), and 62.1% (POE) SR. POE offers exploration gains (+3.1 SR) with lower HR penalty than SAR (-5.7 vs. -15.1), confirming complementarity. Other hyperparameters are robust near $\rho = 0.6$ (Supp. Sec. 8.2). As shown in Fig. 8, the Success Rate remains stable (68.7%–69.2%) within a robust range of ± 0.05 around $\theta = 0.7$, while the Hallucination Rate decreases monotonically. This threshold optimally balances false-positive and false-negative rates while maintaining peak success.

Scaling with Task Complexity. Gains scale with horizon: Bridge (+11.2 SR, 74 stages) > Stack (+10.3) > Jujube-Cup (+9.1), as longer tasks increase hallucination risks mitigated by SAR. Real-world gains remain substantial (+9.2 to +13.4), though ordering varies due to Sim2Real noise.

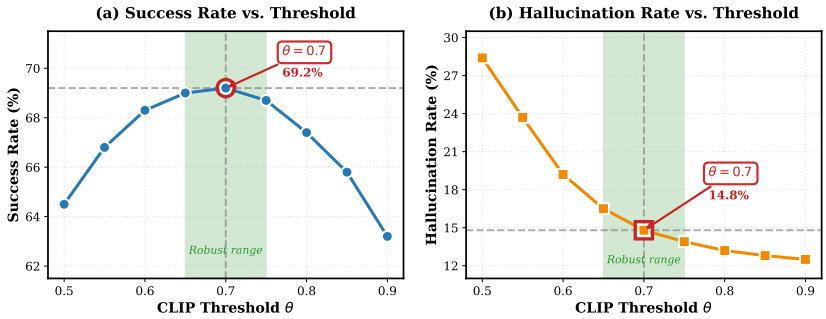

Fig. 8: CLIP threshold sensitivity for EvoVLA (full model) averaged across three Discoverse-L tasks and three seeds. (a) Success Rate remains stable within ±0.5 points around $\theta = 0.7$ (68.7–69.2%). (b) Hallucination Rate decreases monotonically with higher thresholds; $\theta = 0.7$ balances false-positive and false-negative rates while maintaining peak success.

Synergistic Attribution & Failure Diagnosis. Non-cumulative ablations (Supp. Tab. 2) confirm our modules synergistically cut the closed-loop hallucination failure, with POE outperforming pixel-based RND (see Supp. Sec. 6 for details).

6 Conclusion

We formalized stage hallucination–where policies exploit coarse VLM cues without actual completion–and showed it stems from coarse reward contrast and lossy memory compression. EvoVLA resolves this via three synergistic modules (SAR, POE, Memory) while keeping the policy architecture fixed and the CLIP evaluator frozen. Ablations confirm all three are jointly necessary (removing any forfeits $\geq$2.4 SR). EvoVLA achieves 69.2% sim success (+10.2), 1.5× sample efficiency, 14.8% HR (vs. 38.5%), and 54.6% real-robot success across four tasks. We release the Discoverse-L benchmark and HR metric to support future work. Beyond the immediate contributions, our findings suggest that designing stage-aware reward signals and selective memory mechanisms is a promising direction for improving the reliability of VLA policies in long-horizon manipulation, and we hope the tools introduced here facilitate further research on hallucination-aware robot learning.

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities, Peking University.

References

1. Behrouz, A., Zhong, P., Mirrokni, V.: Titans: Learning to memorize at test time. arXiv preprint arXiv:2501.00663 (2025). https://doi.org/10.48550/arXiv.2501.00663, https://arxiv.org/abs/2501.00663
2. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0: A Vision-Language-Action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024). https://doi.org/10.48550/arXiv.2410.24164
3. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: RT-1: Robotics transformer for real-world control at scale. In: Proceedings of Robotics: Science and Systems (RSS) (2023). https://doi.org/10.15607/RSS.2023.XIX.025
4. Burda, Y., Edwards, H., Storkey, A., Klimov, O.: Exploration by random network distillation. In: International Conference on Learning Representations (ICLR) (2019)
5. Chen, B., Zhu, C., Agrawal, P., Zhang, K., Gupta, A.: Self-supervised reinforcement learning that transfers using random features. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 36, pp. 56411–56436. Neural Information Processing Systems Foundation, Inc. (NeurIPS) (2023). https://doi.org/10.52202/075280-2461
6. Chen, Y., Tian, S., Liu, S., Zhou, Y., Li, H., Zhao, D.: ConRFT: A reinforced fine-tuning method for VLA models via consistency policy. In: Proceedings of Robotics: Science and Systems (RSS) (2025). https://doi.org/10.15607/RSS.2025.XXI.019
7. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: PaLM-E: An embodied multimodal language model. In: Proceedings of the International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 202, pp. 8469–8488. PMLR (2023), https://proceedings.mlr.press/v202/driess23a.html
8. Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., Luo, J., et al.: Octo: An open-source generalist robot policy. In: Proceedings of Robotics: Science and Systems (RSS). Robotics: Science and Systems Foundation (2024). https://doi.org/10.15607/rss.2024.xx.090, project: https://octo-models.github.io/, Code: https://github.com/octo-models/octo
9. Google DeepMind: Gemini 2.5 pro model documentation. https://ai.google.dev/gemini-api/docs (2025), accessed: 2025-11-08
10. Guo, Y., Zhang, J., Chen, X., Ji, X., Wang, Y.J., Hu, Y., Chen, J.: Improving Vision-Language-Action model with online reinforcement learning. In: IEEE International Conference on Robotics and Automation. pp. 15665–15672. IEEE (2025). https://doi.org/10.1109/ICRA55743.2025.11127299, arXiv:2501.16664
11. Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al.: Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905 (2018). https://doi.org/10.48550/arXiv.1812.05905
12. Harutyunyan, A., Devlin, S., Vrancx, P., Nowé, A.: Expressing arbitrary reward functions as potential-based advice. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29, pp. 2652–2658 (2015). https://doi.org/10.1609/aaai.v29i1.9628

13. Hu, J., Hendrix, R., Farhadi, A., Kembhavi, A., Martín-Martín, R., Stone, P., Zeng, K.H., Ehsan, K.: FLaRe: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 3617–3624. Atlanta, USA (May 2025). https://doi.org/10.1109/ICRA55743.2025.11127934
14. Huang, D., Fang, Z., Zhang, T., Li, Y., Zhao, L., Xia, C.: CO-RFT: Efficient fine-tuning of Vision-Language-Action models through chunked offline reinforcement learning. arXiv preprint arXiv:2508.02219 (2025). https://doi.org/10.48550/arXiv.2508.02219
15. Huang, T., Zhang, Z., Tang, H.: 3D-R1: Enhancing reasoning in 3D VLMs for unified scene understanding. arXiv preprint arXiv:2507.23478 (2025). https://doi.org/10.48550/arXiv.2507.23478
16. Huang, T., Zhang, Z., Wang, Y., Tang, H.: 3D CoCa: Contrastive learners are 3D captioners. In: International Conference on 3D Vision (3DV). pp. 1771–1780 (2026). https://doi.org/10.1109/3DV69130.2026.00168
17. Huang, T., Zhang, Z., Zhang, R., Zhao, Y.: DC-Scene: Data-centric learning for 3D scene understanding. arXiv preprint arXiv:2505.15232 (2025). https://doi.org/10.48550/arXiv.2505.15232
18. Jia, Y., Wang, G., Dong, Y., Wu, J., Zeng, Y., Lin, H., Wang, Z., Ge, H., Gu, W., Ding, K., et al.: DISCOVERSE: Efficient robot simulation in complex high-fidelity environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6272–6279. IEEE (2025). https://doi.org/10.1109/IROS60139.2025.11247559, https://github.com/TATP-233/DISCOVERSE, code available at the URL above
19. Kim, M.J., Finn, C., Liang, P.: Fine-tuning Vision-Language-Action models: Optimizing speed and success. In: Proceedings of Robotics: Science and Systems (RSS). Los Angeles, USA (2025). https://doi.org/10.15607/RSS.2025.XXI.017
20. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P.R., et al.: OpenVLA: An open-source Vision-Language-Action model. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 270, pp. 2679–2713. PMLR (2025), https://proceedings.mlr.press/v270/kim25c.html, code available at https://github.com/openvla/openvla
21. Li, H., Zuo, Y., Yu, J., Zhang, Y., Yang, Z., Zhang, K., Zhu, X., Zhang, Y., Chen, T., Cui, G., et al.: SimpleVLA-RL: Scaling VLA training via reinforcement learning. arXiv preprint arXiv:2509.09674 (2025). https://doi.org/10.48550/arXiv.2509.09674
22. Li, H., Ding, P., Suo, R., Wang, Y., Ge, Z., Zang, D., Yu, K., Sun, M., Zhang, H., Wang, D., et al.: VLA-RFT: Vision-Language-Action reinforcement fine-tuning with verified rewards in world simulators. arXiv preprint arXiv:2510.00406 (2025). https://doi.org/10.48550/arXiv.2510.00406
23. Liu, J., Liu, M., Wang, Z., An, P., Li, X., Zhou, K., Yang, S., Zhang, R., Guo, Y., Zhang, S.: RoboMamba: Efficient Vision-Language-Action model for robotic reasoning and manipulation. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 37, pp. 40085–40110. Neural Information Processing Systems Foundation, Inc. (NeurIPS) (2024). https://doi.org/10.52202/079017-1266
24. Liu, Q., Huang, T., Zhang, Z., Tang, H.: Nav-R1: Reasoning and navigation in embodied scenes. arXiv preprint arXiv:2509.10884 (2025). https://doi.org/10.48550/arXiv.2509.10884

25. Lu, G., Guo, W., Zhang, C., Zhou, Y., Jiang, H., Gao, Z., Tang, Y., Wang, Z.: VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning. arXiv preprint arXiv:2505.18719 (2025). https://doi.org/10.48550/arXiv.2505.18719
26. Luu, T.M., Lee, D., Lee, Y., Yoo, C.D.: Policy learning from large vision-language model feedback without reward modeling. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 11685–11692. IEEE (2025). https://doi.org/10.1109/IROS60139.2025.11246902
27. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. In: International Conference on Machine Learning. vol. 99, pp. 278–287 (1999)
28. Olson, E.: AprilTag: A robust and flexible visual fiducial system. In: IEEE International Conference on Robotics and Automation. pp. 3400–3407. IEEE (2011). https://doi.org/10.1109/ICRA.2011.5979561
29. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: DINOv2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2024), https://openreview.net/forum?id=a68SUt6zFt
30. Pathak, D., Agrawal, P., Efros, A.A., Darrell, T.: Curiosity-driven exploration by self-supervised prediction. In: Proceedings of the International Conference on Machine Learning (ICML). vol. 70, pp. 2778–2787. PMLR (2017)
31. Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: FAST: Efficient action tokenization for Vision-Language-Action models. In: Proceedings of Robotics: Science and Systems (RSS) (2025). https://doi.org/10.15607/RSS.2025.XXI.012
32. Physical Intelligence: OpenPI: Open-source π-series Vision-Language-Action policies. https://github.com/Physical-Intelligence/openpi (2025), accessed: 2025-11-08
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (2021)
34. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. In: International Conference on Learning Representations (ICLR) (2016)
35. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017). https://doi.org/10.48550/arXiv.1707.06347
36. Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., Pathak, D.: Planning to explore via self-supervised world models. In: Proceedings of the International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 119, pp. 8583–8592. PMLR (2020)
37. Shi, H., Xie, B., Liu, Y., Sun, L., Liu, F., Wang, T., Zhou, E., Fan, H., Zhang, X., Huang, G.: MemoryVLA: Perceptual-cognitive memory in Vision-Language-Action models for robotic manipulation. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=54U3XHf7qq, iCLR 2026 poster
38. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 36, pp. 8634–8652. Neural Information Processing

Systems Foundation, Inc. (NeurIPS) (2023). https://doi.org/10.52202/075280-0377

39. Shu, J., Lin, Z., Wang, Y.: RFTF: Reinforcement fine-tuning for embodied agents with temporal feedback. arXiv preprint arXiv:2505.19767 (2025). https://doi.org/10.48550/arXiv.2505.19767
40. Song, Z., Ouyang, G., Fang, M., Na, H., Shi, Z., Chen, Z., Fu, Y., Zhang, Z., Jiang, S., Fang, M., Chen, L., Chen, X.: Hazards in daily life? Enabling robots to proactively detect and resolve anomalies. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 7399–7415 (2025). https://doi.org/10.18653/v1/2025.naacl-long.379
41. Song, Z., Ouyang, G., Li, M., Ji, Y., Wang, C., Xu, Z., Zhang, Z., Zhang, X., Jiang, Q., Ji, F., Chen, Z., Li, Z., Chen, X.: ManipLVM-R1: Reinforcement learning for reasoning in embodied manipulation with Large Vision-Language Models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 18558–18566 (2026). https://doi.org/10.1609/aaai.v40i22.38922
42. Sontakke, S.A., Zhang, J., Arnold, S.M.R., Pertsch, K., Biyik, E., Sadigh, D., Finn, C., Itti, L.: RoboCLIP: One demonstration is enough to learn robot policies. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 36, pp. 55681–55693 (2023). https://doi.org/10.52202/075280-2430
43. Touvron, H., Martin, L., Stone, K.R., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023). https://doi.org/10.48550/arXiv.2307.09288
44. Venkataraman, S., Wang, Y., Wang, Z., Ravie, N.S., Erickson, Z., Held, D.: Real-world offline reinforcement learning from Vision-Language Model feedback. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 13452–13459. IEEE (2025). https://doi.org/10.1109/IROS60139.2025.11246918
45. Wang, Z., Cai, S., Liu, A., Jin, Y., Hou, J., Zhang, B., Lin, H., He, Z., Zheng, Z., Yang, Y., et al.: JARVIS-1: Open-world multi-task agents with memory-augmented multimodal language models. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**, 1894–1907 (2025). https://doi.org/10.1109/TPAMI.2024.3511593
46. Ye, A., Zhang, Z., Wang, B., Wang, X., Zhang, D., Zhu, Z.: VLA-R1: Enhancing reasoning in Vision-Language-Action models. arXiv preprint arXiv:2510.01623 (2025). https://doi.org/10.48550/arXiv.2510.01623
47. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: SigLIP: Sigmoid loss for language image pre-training. In: IEEE International Conference on Computer Vision. pp. 11941–11952. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.01100
48. Zhang, H., Zhuang, Z., Zhao, H., Ding, P., Lu, H., Wang, D.: ReinboT: Amplifying robot visual-language manipulation with reinforcement learning. In: Proceedings of the 42nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 267, pp. 77254–77271. PMLR (2025), https://proceedings.mlr.press/v267/zhang25dr.html
49. Zhao, Y., Yuan, J., Xu, Z., Hao, X., Zhang, X., Wu, K., Che, Z., Liu, C.H., Tang, J.: Training-free generation of temporally consistent rewards from VLMs. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 8133–8143. IEEE (2025). https://doi.org/10.1109/ICCV51701.2025.00762

50. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P.R., Salazar, G., Ryoo, M.S., Reymann, K., Rao, K., Pertsch, K., Mordatch, I., Michalewski, H., Lu, Y., Levine, S., Lee, L., Lee, T.W.E., Leal, I., Kuang, Y., Kalashnikov, D., Julian, R., Joshi, N.J., Irpan, A., Ichter, B., Hsu, J., Herzog, A., Hausman, K., Gopalakrishnan, K., Fu, C., Florence, P., Finn, C., Dubey, K.A., Driess, D., Ding, T., Choromanski, K.M., Chen, X., Chebotar, Y., Carbajal, J., Brown, N., Brohan, A., Arenas, M.G., Han, K.: RT-2: Vision-Language-Action models transfer web knowledge to robotic control. In: Proceedings of the 7th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 229, pp. 2165–2183. PMLR (2023), https://proceedings.mlr.press/v229/zitkovich23a.html