PARL-VLA: Pruning-Aware On-Policy Reinforcement Learning for Vision-Language-Action Model

Ruiyan Xu[1], Jiashu Lv[3], Sixu Lin[1], Ruixing Jin[1], Shuliang He[1], and Guiliang Liu[*1,2]

[1] School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[2] Shenzhen Loop Area Institute, Shenzhen, China
[3] Peking University, Beijing, China

Abstract. Token pruning has emerged as an essential technique for enabling efficient and resilient inference for foundation models, showing great potential in accelerating the Reinforcement Learning (RL) optimization of Vision–Language–Action (VLA) Models. However, our in-depth analysis indicates that directly transferring prior pruning strategies to RL is non-trivial. Inference-time pruning introduces a train–test discrepancy, whereas dynamically adjusting the pruning configuration leads to inconsistent policy rollout and improvement. Therefore, we propose PARL-VLA, a pruning-aware RL framework that enforces exploration-exploitation pruning consistency by recording the rollout pruning configuration and replaying it during policy updates, enabling end-to-end co-adaptation between token pruning and policy optimization. PARL-VLA further trains across a spectrum of token budgets to learn a single policy that remains reliable under different information budgets at test time. On LIBERO and LIBERO-Plus, PARL-VLA discards 55% of visual tokens, speeds up rollout policy forward by about 1.3× and policy updates by about 1.7×, and improves LIBERO-Plus robustness by up to 6.6 points (3.2 on average) without sacrificing in-distribution success. On RoboTwin2.0, PARL-VLA achieves 46.9% real-robot success over 405 trials, improving over RLinf at 37.0%.

Keywords: Vision-Language-Action · Reinforcement Learning · Visual Token Pruning · Robustness

1 Introduction

Vision-Language-Action (VLA) models [1,12,17,18,30,35] have recently emerged as generalist foundation models for robotic manipulation, capable of translating visual observations and language instructions into executable actions across diverse tasks and environments. To improve model dexterity and precision, prior

* Corresponding author.

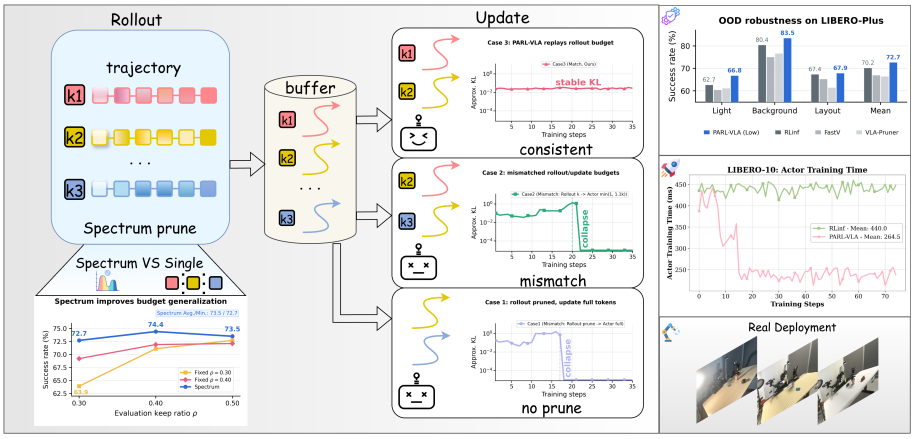


Fig. 1: Overview of PARL-VLA. Our pruning-aware RL loop enables co-adaptation between token selection and policy optimization through exploration exploitation pruning consistency. Here k_1, k_2, k_3 are sampled visual-token budgets, and spectrum pruning trains over multiple keep ratios rather than one fixed budget. The update panels show stable KL when the rollout pruning configuration is replayed, but collapse under mismatched or no-pruning updates.

work [20, 37, 45] applied Reinforcement Learning (RL) during post-training, allowing agents to learn from direct environmental interactions. However, RL training typically incurs substantial computational and sample complexity, as it requires large-scale environment interactions for effective policy optimization [36]. This challenge is particularly significant for foundation models such as VLAs, owing to their heavy computational overhead and the high inference cost associated with large-scale architectures [29].

Striving for sample-efficient post-training for VLA models, model token pruning appears as a promising path toward lightweight policy optimization. Recent studies have shown that a substantial portion of visual tokens are often redundant or even distracting for decision making [19]. By locating and discarding these redundant patch tokens, pruning shortens the attention sequence and reduces the computational cost of transformer inference. This technique has proven effective for enhancing the efficiency of both vision-language models [3, 43, 46–49] and VLA models [14, 23, 27]. These properties make inference-time pruning a natural choice to accelerate RL optimization, where candidate policies are repeatedly evaluated in the environment to collect rollout trajectories.

In this study, we explore the integration of pruning methods into on-policy reinforcement learning (RL) algorithms for VLA post-training. However, since RL involves iterative policy evaluation and updates, directly applying conventional pruning techniques often falls short of maintaining stable policy improvement, which can in turn hinder the convergence toward an optimal policy. For example, applying pruning only during evaluation introduces a train–test mismatch,

as the policy is never trained on pruned observations. Conversely, if pruning is applied during rollouts but the policy updates recompute log-probabilities under a different pruning configuration, the likelihood ratios used by GRPO no longer align with the rollout distribution [33]. In our experiments, such inconsistencies can lead to severe training instability or collapse. Notably, even a modest mismatch in pruning budgets between rollout and update phases results in a persistent performance degradation.

These findings highlight a key design principle for incorporating pruning into VLA-RL: pruning and policy optimization must be coupled to enable co-adaptation within the RL loop. Based on this principle, we propose PARL-VLA, a pruning-aware RL framework that applies pruning in both rollouts and on-policy updates. For each rollout sample, we record the rollout pruning configuration and replay it when recomputing action log-probabilities during updates, keeping likelihood ratios aligned with the rollout regime. We train the pruning mechanism end-to-end with straight-through gradients and train across a spectrum of token budgets using a curriculum that shifts from denser observations to more aggressive pruning over training.

We evaluate PARL-VLA on LIBERO [25] and its perturbed benchmark LIBERO-Plus [9]. PARL-VLA improves robustness under lighting, background, and layout perturbations by up to 6.6 percentage points while discarding 55% of visual tokens, reducing wall-clock cost without sacrificing success.

Overall, we summarize our contributions as follows:

- We identify and characterize rollout-learn pruning mismatch in on-policy VLA-RL, showing that naive pruning can cause training collapse or consistent performance degradation.
- We propose PARL-VLA, a pruning-aware RL framework that enforces exploration exploitation pruning consistency and enables co-adaptation between token pruning and on-policy RL optimization.
- We show that training across a spectrum of token budgets improves robustness under visual perturbations and reduces the cost of data collection and learning.
- We validate PARL-VLA on LIBERO, LIBERO-Plus, and RoboTwin2.0, demonstrating up to 6.6 points robustness improvement with 55% visual token reduction, about 1.3× faster rollout policy forward and 1.7× faster policy updates during training, and 46.9% real-robot success over 405 trials.

2 Related Work

Reinforcement Learning for VLA. Supervised fine-tuning on demonstrations [31, 38] is a common adaptation strategy for VLA models, yet it can struggle under distribution shift and limited coverage of offline data. Recent work explores reinforcement learning as a post-training stage that improves task success and generalization, including scalable online RL frameworks [20, 28], offline RL with action chunking [10], and interactive fine-tuning with sparse success feedback [37]. Several policy optimization variants further improve stability and

efficiency for VLA post-training [6, 8], and complementary directions incorporate consistency objectives or human guidance [5, 15]. Other frameworks study verified rewards in simulators [21], sim–real co-training [34], systematic factors governing VLA sim-to-real generalization [16], and RL-driven data generation and distillation back into VLA policies [39]. System support for large-scale RL training has also been studied [45], and concurrent analyses examine when RL improves VLA generalization [26]. Related work on generalist policy distillation via RL provides further evidence that RL can improve both state coverage and action distributions [41]. This line of work focuses on better policy optimization, while our work studies how to incorporate token pruning into on-policy training without introducing exploration-exploitation mismatch.

Token Pruning for VLA. Visual token pruning accelerates multimodal transformers by reducing the number of visual patch tokens processed by attention. Token reduction has also been studied for vision transformers through token reorganization and token merging [2, 24]. A large line of work focuses on training-free inference-time token pruning and sparsification for vision–language models, including FastV [3], SparseVLM [48], Fit-and-Prune [44], and [CLS]-guided pruning [47]. Other inference-time approaches reduce multimodal context via dynamic sparsification or token reduction operators [11, 22, 32, 40]. Complementary approaches learn lightweight token compression modules for multimodal LLMs [49] or introduce differentiable pruning during VLA adaptation [14]. VLA-specific efficiency methods must preserve fine-grained visuomotor cues and often incorporate temporal structure or action-related signals, as in VLA-Pruner [27] and SP-VLA [23]. Related directions explore training-free acceleration and compression [43], adaptive token caching [42], and adaptive computation mechanisms [46]. DTP shows that pruning task-irrelevant tokens can also improve manipulation success and robustness [19]. Most existing pruning methods target inference or supervised adaptation, whereas our work integrates pruning into on-policy RL training and enforces exploration-exploitation pruning consistency.

3 Problem Formulation

Partially Observable Markov Decision Process. We model the task as an instruction-conditioned partially observable Markov decision process (POMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, \Omega, r, \gamma, p_0)$. At the beginning of each episode, an instruction $x \sim p(x)$ specifies the task. The environment initializes from $s_0 \sim p_0$, evolves according to $s_{t+1} \sim \mathcal{T}(\cdot \mid s_t, a_t)$, and emits observations $o_t \sim \Omega(\cdot \mid s_t, x)$. In our setting, the agent observes $o_t \triangleq (I_t, x)$, where I_t is an image observation and x remains fixed within the episode. A finite-horizon trajectory is $\tau = (s_0, o_0, a_0, \ldots, s_{T-1}, o_{T-1}, a_{T-1})$, and the episodic return is $R(\tau) = \sum_{t=0}^{T-1} r(s_t, a_t)$.

Token Pruning. We consider a pruning-aware policy $a_t \sim \pi_{\theta,\phi}(\cdot \mid o_t)$. At control step t, a VLA encoder maps I_t to N visual patch tokens and tokenizes the instruction into L language tokens. Let $V_t \in \mathbb{R}^{N \times d}$ denote the visual tokens and $X \in \mathbb{R}^{L \times d}$ denote the instruction tokens (we omit the time index for X

since x is episode-level). We introduce a rollout-conditioned visual token budget $k \in \{1, \ldots, N\}$ and a selector perturbation ϵ_t used for stochastic hard selection. A pruning module $\mathcal{P}_\phi$ produces a size-k pruned visual sequence:

$$\tilde{V}_t = \mathcal{P}_\phi(V_t, X, k, \epsilon_t), \qquad |\tilde{V}_t| = k. \tag{1}$$

The VLA policy then conditions on the pruned token sequence

$$\tilde{H}_t = [\text{BOS}, \tilde{V}_t, X, U_t], \tag{2}$$

where BOS is the beginning-of-sequence token and U_t denotes additional backbone-specific tokens such as action and prompt tokens. Importantly, rollouts and actor updates apply pruning under the same recorded configuration (k, ϵ_t) for each sample, which avoids exploration-exploitation pruning mismatch during policy updates. We jointly optimize the policy parameters θ and pruning module parameters ϕ under a budget distribution $p(k)$:

$$\max_{\theta,\phi} \mathbb{E}_{x\sim p(x),\, k\sim p(k)} \Big[\mathbb{E}_{\tau\sim\pi_{\theta,\phi}(\cdot\,;x,k)}[R(\tau)] \Big] \quad \text{s.t.} \quad |\tilde{V}_t| = k,\ \forall t. \tag{3}$$

Compared with fixed-ratio pruning, sampling $k \sim p(k)$ exposes training to varying information density and yields a single policy that is robust across token budgets at test time. In practice, we sample (k, ϵ_t) during rollouts and record them in trajectories. This adds only a few scalars per transition and enables actor learning to replay the rollout pruning configuration during policy updates.

4 Effective Token Pruning for VLA-RL

Unlike post-hoc inference pruning, PARL-VLA enforces exploration-exploitation pruning consistency by replaying the recorded rollout pruning configuration during learning, enabling co-adaptation between token selection and policy optimization under GRPO objectives. We further train over a spectrum of token budgets to improve robustness across information budgets at test time, while reducing wall-clock cost in both data collection and learning.

4.1 GRPO

VLA Backbone. We instantiate PARL-VLA on OpenVLA-OFT [17] in our RL training stack. Images are encoded into patch-level visual tokens and concatenated with instruction tokens to form the multimodal context. While OpenVLA-OFT can incorporate proprioceptive state inputs, we follow the SimpleVLA-RL LIBERO setting and do not condition on them for efficiency. The transformer backbone predicts chunked action tokens, whose log-probabilities are used for policy-gradient updates. Executed continuous controls are recovered by de-tokenization and unnormalization. PARL-VLA is inserted before the language backbone: only visual patch tokens are pruned, while special tokens and non-visual tokens are preserved.

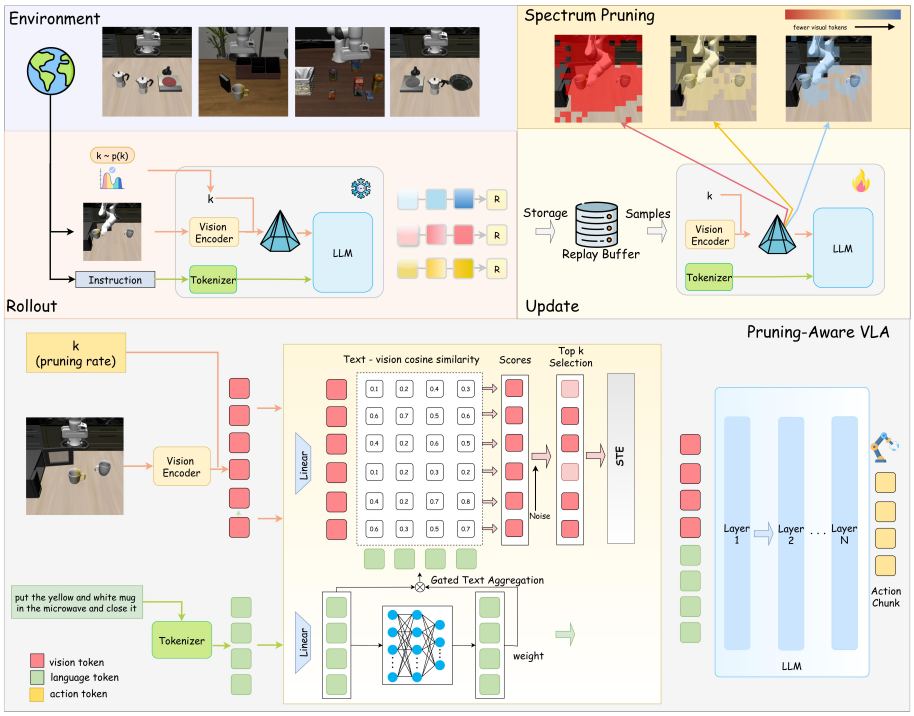


Fig. 2: Training Pipeline. PARL-VLA integrates visual token pruning into both rollouts and GRPO updates, and trains a single policy across token budgets.

Group Relative Policy Optimization. For each instruction x, rollout generates a trajectory group $\{\tau_i\}_{i=1}^{G}$ and computes a scalar trajectory score R_i from environment rewards. GRPO [33] computes group-relative advantages:

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G}) + \epsilon}. \tag{4}$$

Policy updates use a clipped surrogate on action tokens:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}\Big[\min\Big(r_{i,t}(\theta)\hat{A}_i, \text{clip}(r_{i,t}(\theta), 1-\epsilon_l, 1+\epsilon_h)\hat{A}_i\Big)\Big], \tag{5}$$

where

$$r_{i,t}(\theta) = \exp(\log \pi_\theta(a_{i,t}|s_{i,t}) - \log \pi_{\theta_{\text{old}}}(a_{i,t}|s_{i,t})), \tag{6}$$

and the expectation is taken over valid action tokens (excluding padding and post-termination positions). In PARL-VLA, we attach pruning metadata such as the token budget k to each trajectory sample, and actor updates recompute log-probabilities under the same pruning configuration used during rollout.

4.2 Spectrum Pruning

We prune visual patch tokens to a budget k before the language backbone, retaining only the tokens that are most relevant to the instruction. Following FastV and SparseVLM [3, 48], we accelerate computation by dropping tokens from the input sequence rather than sparsifying model weights. This keeps the transformer backbone unchanged: it still performs dense attention over the retained tokens, and it can directly leverage fused-attention kernels such as FlashAttention [7] and standard distributed training setups.

Spectrum Training. During rollouts, we sample a token budget $k \sim p(k)$ by first sampling a keep ratio $\rho = k/N$, where N is the number of visual patch tokens, and then setting $k = \lfloor \rho N \rfloor$. The spectrum distribution is a discretized uniform distribution over keep-ratio intervals. We use a loose-to-aggressive curriculum: $\rho \in [0.75, 1.0]$ for training progress $[0, 0.1)$, $\rho \in [0.5, 0.75]$ for $[0.1, 0.2)$, and $\rho \in [0.25, 0.5]$ afterward. For each rollout sample, we store the sampled k and selector perturbation ϵ_t in the trajectory and reuse this configuration when recomputing $\log \pi_{\theta,\phi}(a_t \mid I_t, x; k, \epsilon_t)$ during GRPO updates, which keeps likelihood ratios aligned with the rollout regime. Training under multiple budgets forces the policy to act with less visual evidence and discourages brittle reliance on nuisance visual cues. This yields a single policy that remains stable across evaluation budgets and improves robustness under distribution shift.

Text-conditioned Scoring. Given visual tokens $V_t \in \mathbb{R}^{N \times d}$ and instruction tokens $X_t \in \mathbb{R}^{L \times d}$, we compute a relevance score for each visual token using normalized similarity. We first project both modalities into a shared projection space and L_2-normalize:

$$q_{t,\ell} = \mathrm{norm}(W_q x_{t,\ell}), \quad k_{t,n} = \mathrm{norm}(W_k v_{t,n}), \tag{7}$$

where $\mathrm{norm}(\cdot)$ denotes L_2 normalization and $x_{t,\ell}$ and $v_{t,n}$ are tokens from X_t and V_t. To aggregate instruction tokens, we use a lightweight gated pooling that suppresses padding and low-salience tokens:

$$g_{t,\ell} = \sigma(f_{\mathrm{gate}}(x_{t,\ell})), \quad \bar{q}_t = \frac{\sum_{\ell=1}^{L} g_{t,\ell} q_{t,\ell}}{\sum_{\ell=1}^{L} g_{t,\ell} + \epsilon}. \tag{8}$$

The final score is cosine similarity between the pooled instruction query and each visual key:

$$s_{t,n} = \bar{q}_t^{\top} k_{t,n}. \tag{9}$$

Stochastic Hard Top-k with Straight-through Gradients. We use a perturb-and-select rule in the spirit of Gumbel-style [13] token selection: during learning we add a small random perturbation to token scores and then apply hard Top-k to physically shorten the visual sequence,

$$\mathcal{S}_t = \mathrm{TopK}(s_{t,1:N} + \epsilon_{t,1:N}, k), \qquad \epsilon_{t,n} \sim \mathcal{U}(-\sigma_t/2, \sigma_t/2), \qquad \tilde{V}_t = \{v_{t,n} \mid n \in \mathcal{S}_t\}. \tag{10}$$

We optimize the scoring network with a straight-through estimator, using a soft relaxation (temperature-controlled softmax) only to define gradients while

keeping the forward selection discrete. The perturbation is used as a training-time exploration mechanism.

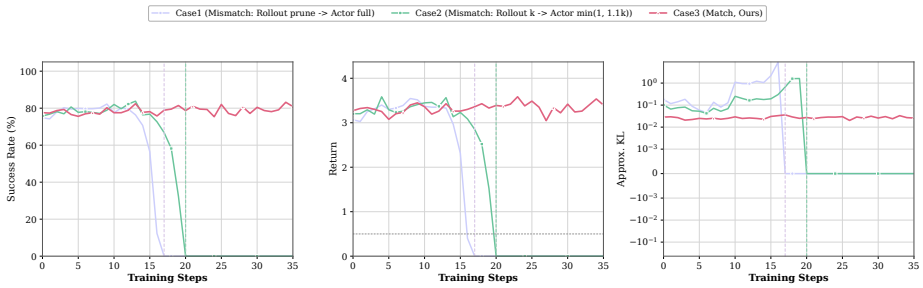


Fig. 3: Rollout-learn Pruning Consistency. We compare three on-policy training variants that differ only in how pruning is handled when recomputing action log-probabilities during policy updates. **Case 1:** rollouts use pruned observations, while updates use the full visual token set. **Case 2:** rollouts and updates both prune, but updates use a different token budget than rollouts. **Case 3 PARL-VLA:** updates replay the recorded rollout pruning configuration for each sample.

5 Experiments

We conduct experiments in simulation and on a real robot to characterize prune-aware VLA-RL in terms of robustness, efficiency, optimization stability, and budget generalization. We organize our evaluation around four questions:

- **Q1: Robustness.** Does training with visual token pruning improve robustness under visual perturbations in simulation and on a real robot?
- **Q2: Efficiency.** How does pruning affect the compute cost of rollout policy inference and policy updates, and what performance–efficiency trade-off does it induce?
- **Q3: Exploration-exploitation pruning consistency.** Is it necessary to replay the rollout pruning configuration during actor updates to ensure stable learning under on-policy optimization?
- **Q4: Spectrum vs. single budget.** Does training over a spectrum of token budgets improve generalization across evaluation budgets compared to fixed-budget training?

We answer Q1 and Q2 in the main results and address Q3 and Q4 via targeted ablations.

5.1 Experimental Setup

Benchmarks. We evaluate all methods on LIBERO [25], training on the LIBERO task suites (LIBERO-10, Spatial, Object, and Goal) and reporting in-distribution

performance on LIBERO as well as out-of-distribution generalization on LIBERO-Plus [9]. LIBERO comprises five suites spanning 130 tasks, including *Spatial* (varying object placements), *Object* (swapping objects in a box), *Goal* (diverse goals in a fixed environment), and *Long* (long-horizon tasks across scenes). LIBERO-Plus augments LIBERO with seven perturbation types, resulting in 10,030 task variants for stress-testing robustness under distribution shift. We follow the official LIBERO-Plus protocol and focus on three visual perturbation categories: lighting, background textures, and object layout.

Baselines. We compare: (i) RLinf [45], a full-token baseline without pruning, (ii) evaluation-only pruning baselines that train with full tokens and prune only at evaluation, including FastV [3] and VLA-Pruner [27], and (iii) PARL-VLA, which applies pruning during both rollout and learning with matched exploration-exploitation pruning configurations. Unless otherwise noted, we report PARL-VLA at three token budgets, denoted Low, Medium, and High. These are evaluation-time pruning settings rather than the spectrum training distribution: Low, Medium, and High denote evaluation-time keep ratios of 30%, 50%, and 70%, respectively, corresponding to pruning rates of 70%, 50%, and 30%.

Evaluation Metrics. We use success rate as the primary metric throughout. For robustness evaluation, we report success on LIBERO-Plus perturbations (light, background, and layout) relative to in-distribution LIBERO evaluation. For efficiency, we report the token keep ratio and wall-clock time for rollout generation, actor updates, and evaluation; when environment simulation dominates end-to-end time, we also report policy-forward latency to isolate model-side compute. We also report episodic return and the approximate KL logged by GRPO as indicators of optimization stability.

Implementation Details. All experiments are implemented in the RLinf [45] training framework, which orchestrates distributed rollouts and FSDP-based actor updates. We use OpenVLA-OFT [17] as the base VLA policy and optimize it with GRPO. Following the SimpleVLA-RL LIBERO setting, we do not condition on proprioceptive inputs. Unless stated otherwise, the policy receives a single image observation. For efficiency benchmarking, we use the multi-view setting with an additional wrist camera to increase the number of visual tokens and better isolate model-side speedups from environment time. When prompt-level reward filtering is enabled by RLinf, we mark the filtering threshold in plots. All runs are configured for 100 epochs; we report results using the checkpoint saved at epoch 75 for all methods to ensure a consistent training budget across comparisons. This checkpoint lies in the main sparsity regime of our keep-ratio curriculum (the $\rho \in [0.25, 0.5]$ phase), therefore the reported results reflect the target sparse operating range.

5.2 Main Results

Robustness under Visual Perturbations Quantitative results. To answer Q1, Tab. 1 reports success on LIBERO-Plus under lighting, background, and layout shifts. At the High token budget, PARL-VLA improves the overall mean success from 70.2 to 74.0 and improves the Goal suite from 66.7 to

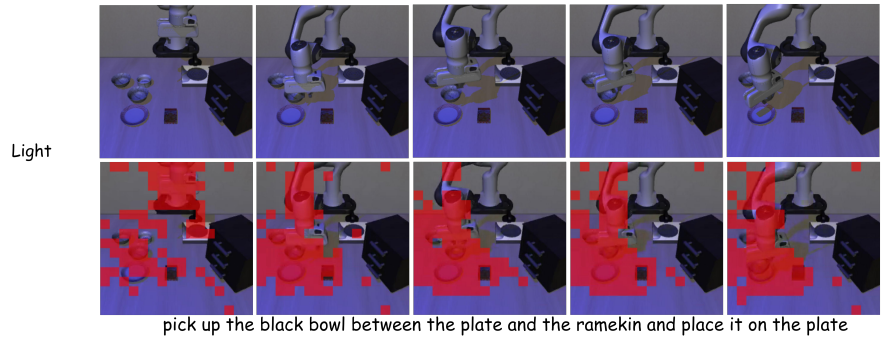


Fig. 4: Robustness visualization. Top: observations under perturbations. Bottom: retained patch tokens in red.

73.5. Lighting is the most challenging perturbation for RLinf, and PARL-VLA improves the light-category average from 62.7 to 68.7. Evaluation-only pruning does not improve the overall mean robustness over RLinf, with FastV and VLA-Pruner reaching 67.0 and 66.5. These results indicate that pruning improves robustness when it is integrated into on-policy training rather than applied only at evaluation.

Qualitative Visualization. Fig. 4 visualizes retained patch tokens under representative perturbations. Across perturbations, retained tokens place substantial emphasis on the robot end-effector and task-relevant objects, while allocating fewer tokens to regions dominated by perturbation-specific appearance changes. These examples illustrate that pruning can preserve action-relevant evidence under a range of visual shifts.

In-Distribution Performance on LIBERO Tab. 2 reports in-distribution success when training and testing on LIBERO. Across Low, Medium, and High token budgets, PARL-VLA matches RLinf, achieving 94.3–94.9 average success compared to 94.4 for RLinf, while evaluation-only pruning baselines achieve lower success. Together with the robustness gains on LIBERO-Plus, these results suggest that improved robustness does not require sacrificing in-distribution task success.

Training Acceleration To answer Q2, we quantify the wall-clock efficiency gains from pruning during on-policy training. Fig. 5 shows that PARL-VLA removes 55% of visual tokens on average, yielding about 1.30× faster rollout policy forward and 1.66–1.74× faster policy updates. Combining rollout generation and policy updates yields a 1.21–1.23× end-to-end speedup per training step in our setup. This end-to-end speedup is smaller than the model-side gains because environment stepping and visual encoding dominate total rollout time. Tab. 4 shows that this acceleration does not come at the expense of task success: PARL-VLA matches RLinf across suites after accelerated training.

Table 1: Out-of-Distribution Robustness on LIBERO-Plus. Success rate (%) under three perturbation categories (light conditions, background textures, and object layout changes). Avg. is the mean over the task suites; Mean averages over perturbation categories. PARL-VLA (Low/Medium/High) denotes evaluation-time keep ratios of 30%/50%/70%, respectively.

Pert.	Method	Spatial	Object	Goal	Long	Avg.
Light	RLinf [45]	83.9	44.1	63.8	59.1	62.7
	FastV [3]	71.9	38.7	74.6	56.9	60.5
	VLA-Pruner [27]	77.7	47.5	62.4	57.3	61.2
	PARL-VLA (Low)	**86.6**	44.1	74.9	61.7	66.8
	PARL-VLA (Medium)	84.9	49.8	**76.3**	57.3	67.1
	PARL-VLA (High)	82.9	**51.2**	75.6	**65.0**	**68.7**
Background	RLinf [45]	91.9	81.0	79.0	69.6	80.4
	FastV [3]	82.2	69.8	78.3	70.2	75.1
	VLA-Pruner [27]	84.5	79.4	75.8	67.1	76.7
	PARL-VLA (Low)	95.3	**82.3**	83.3	**73.0**	83.5
	PARL-VLA (Medium)	**96.5**	**82.3**	85.8	71.6	**84.1**
	PARL-VLA (High)	93.0	81.0	**86.1**	70.2	82.6
Layout	RLinf [45]	87.0	62.0	57.2	63.5	67.4
	FastV [3]	80.8	61.8	58.4	60.3	65.3
	VLA-Pruner [27]	73.5	58.6	56.2	57.7	61.5
	PARL-VLA (Low)	89.4	60.3	**59.1**	62.8	67.9
	PARL-VLA (Medium)	90.1	**64.5**	58.6	64.1	69.3
	PARL-VLA (High)	**93.0**	63.5	58.8	**67.3**	**70.6**
Mean	RLinf [45]	87.6	62.4	66.7	64.1	70.2
	FastV [3]	78.3	56.8	70.4	62.5	67.0
	VLA-Pruner [27]	78.6	61.8	64.8	60.7	66.5
	PARL-VLA (Low)	90.4	62.2	72.4	65.8	72.7
	PARL-VLA (Medium)	**90.5**	**65.5**	**73.6**	64.3	73.5
	PARL-VLA (High)	89.6	65.2	73.5	**67.5**	**74.0**

5.3 Ablation

Exploration-Exploitation Pruning Consistency To answer Q3, we test whether policy updates must replay the rollout pruning configuration to keep likelihood ratios aligned between exploration and exploitation. We compare three cases that differ only in how pruning is handled when recomputing action log-probabilities: using full tokens during updates, using a different pruning budget during updates, or replaying the recorded rollout configuration for each sample. Fig. 3 shows that replaying the rollout configuration is required for stable learning, keeping the approximate KL near 0.01. In contrast, both mismatch settings produce much larger KL, rising from around 0.1 early in training and exceeding 1 later. This corresponds to rapid collapse when updates use full tokens and a persistent performance gap when the update pruning configuration differs from rollout. As performance degrades and rollouts concentrate on low-return trajectories, the relative-advantage signal becomes weak and often near-zero, reducing the effective learning signal and further destabilizing updates.

Effect of Spectrum Training. To answer Q4, we study whether training across a spectrum of token budgets yields a single policy that remains reliable under different evaluation budgets. Tab. 3 compares fixed-budget training and

Table 2: In-distribution Results on LIBERO. Success rate (%) on the four LIBERO suites when training and testing on LIBERO.

Method	Spatial	Object	Goal	Long	Avg.
RLinf [45]	**97.5**	96.8	94.5	88.9	94.4
FastV [3]	94.6	94.0	93.5	87.5	92.4
VLA-Pruner [27]	92.9	94.4	92.1	87.7	91.8
PARL-VLA (Low)	96.6	97.2	**95.0**	**90.7**	**94.9**
PARL-VLA (Medium)	96.2	**97.8**	94.6	90.1	94.7
PARL-VLA (High)	97.0	96.4	93.8	90.0	94.3

Table 3: Effect of Spectrum Training. Success rate (%) averaged over suites and visual perturbations on LIBERO-Plus. Only the distribution of training keep ratios differs.

Training	Eval keep ratio ρ			Avg.	Min.
	0.30	0.40	0.50		
Fixed $\rho = 0.30$	63.9	71.1	72.7	69.2	63.9
Fixed $\rho = 0.40$	69.2	71.9	72.1	71.1	69.2
Spectrum	**72.7**	**74.4**	**73.5**	**73.5**	**72.7**

Table 4: Acceleration without Performance Drop. Evaluation success rate (%) after accelerated training across LIBERO suites.

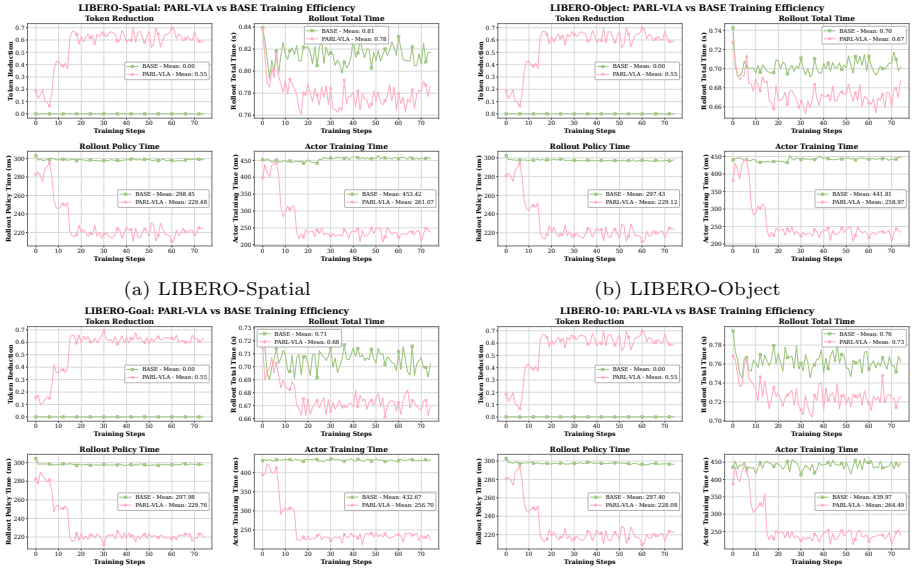
Method	Spatial	Long	Goal	Object
RLinf [45]	96.1	91.2	96.7	97.5
FastV [3]	63.5	87.0	93.8	66.7
VLA-Pruner [27]	94.3	91.3	97.4	92.0
PARL-VLA	95.8	92.5	96.2	96.6

spectrum training. Spectrum training improves generalization across evaluation budgets, increasing both average and worst-case success on LIBERO-Plus. Compared to Fixed $\rho = 0.40$, the best fixed-budget setting, spectrum training improves the average success from 71.1 to 73.5 and raises the worst-case success from 69.2 to 72.7. The largest gain occurs at the most aggressive evaluation budget $\rho = 0.30$, where spectrum training reaches 72.7 versus 63.9 for Fixed $\rho = 0.30$ and 69.2 for Fixed $\rho = 0.40$. These results suggest that fixed-budget training over-specializes to a single information level, whereas varying budgets during training regularizes the policy to operate under varying degrees of visual information.

5.4 Real-World Experiments

RoboTwin2.0 Setup. To further answer Q1 in a real-world setting, we evaluate on an AgileX Piper robotic arm shown in Fig. 6 using RoboTwin2.0 with three manipulation tasks: Place empty cup, Beat block hammer, and Pick dual bottles. Following the SimpleVLA-RL protocol, each policy acts from a single RGB observation and is executed open-loop at the control frequency used by the benchmark. To stress robustness to visual shift, we test three perturbation types: lighting changes (L), background texture changes (B), and object layout changes (O). For each task–perturbation pair, we evaluate a 3×3 initial-state grid with five repeats, giving 45 trials per pair and 405 real-robot trials per method. All methods use the same grid and success criterion: completing the task within the horizon without human intervention.

Results. As shown in Tab. 5, the real-robot evaluation confirms that PARL-VLA achieves the highest overall success rate, reaching 46.9% over 405 trials



(a) LIBERO-Spatial (b) LIBERO-Object

(c) LIBERO-Goal (d) LIBERO-10

Fig. 5: Training Acceleration. Wall-clock training efficiency of PARL-VLA compared with the full-token RLinf baseline. Across LIBERO suites, PARL-VLA reduces visual tokens during on-policy training and speeds up both rollout generation and actor updates, yielding consistent end-to-end acceleration.

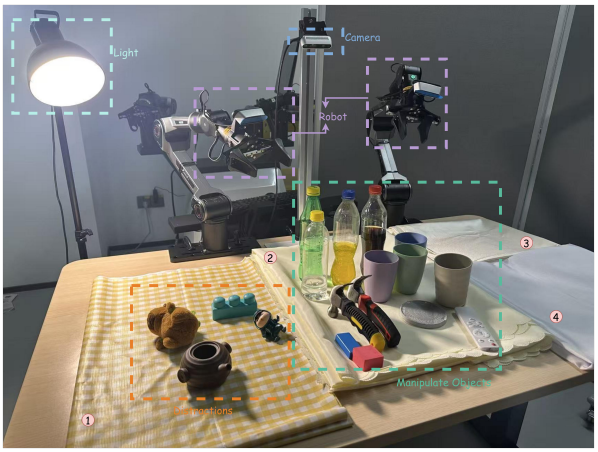


Fig. 6: Real-robot Platform. We evaluate on an AgileX Piper robotic arm following the SimpleVLA-RL [20] real-world protocol.

with a 95% Wilson interval of 42.1–51.8%. It outperforms the full-token RLinf baseline at 37.0% (32.5–41.8), VLA-Pruner at 26.7% (22.6–31.2), and FastV at 21.2% (17.5–25.5). The gains are consistent across tasks: compared with RLinf,

Table 5: RoboTwin2.0 [4] Results in Simulation and on a Real Robot. Top: success rates in domain-randomized simulation. Bottom: real-robot evaluation with 45 trials per task–perturbation pair, aggregated over lighting/background/object-layout perturbations; parentheses denote 95% Wilson intervals.

Simulation (success %)				
Method	**Place empty cup**	**Beat block hammer**	**Pick dual bottles**	**Avg.**
RLinf [45]	93.4	78.9	64.8	79.0
FastV [3]	89.9	76.6	62.7	76.4
VLA-Pruner [27]	91.4	77.8	61.3	76.8
PARL-VLA	**94.1**	**81.6**	**68.0**	**81.2**

Real-World Tests (%, 95% Wilson CI)				
Method	**Place empty cup**	**Beat block hammer**	**Pick dual bottles**	**Avg.**
RLinf [45]	60.7 (52.3/68.6)	23.7 (17.3/31.5)	26.7 (19.9/34.7)	37.0 (32.5/41.8)
FastV [3]	37.8 (30.0/46.2)	12.6 (8.0/19.2)	13.3 (8.6/20.1)	21.2 (17.5/25.5)
VLA-Pruner [27]	45.9 (37.7/54.3)	16.3 (11.0/23.4)	17.8 (12.2/25.1)	26.7 (22.6/31.2)
PARL-VLA	**77.0** (69.3/83.3)	**30.4** (23.2/38.6)	**33.3** (25.9/41.6)	**46.9** (42.1/51.8)

PARL-VLA improves Place empty cup from 60.7% to 77.0%, Beat block hammer from 23.7% to 30.4%, and Pick dual bottles from 26.7% to 33.3%. Together with the domain-randomized simulation results, where PARL-VLA also matches or exceeds the full-token RLinf baseline (81.2% versus 79.0%), these results indicate that pruning-aware training improves robustness under visual perturbations while remaining deployable on physical hardware.

6 Conclusion

We study how to integrate visual token pruning into on-policy reinforcement learning for vision-language-action policies. Our experiments show that naive pruning can fail due to exploration-exploitation pruning mismatch, where trajectory collection and policy optimization operate under different pruned observations and yield inconsistent likelihood ratios. PARL-VLA addresses this by recording the rollout pruning configuration and replaying it during updates, jointly training the pruning mechanism with the policy, and training across a spectrum of token budgets. On LIBERO-Plus, PARL-VLA improves robustness under visual perturbations, increasing the overall mean success from 70.2 to 74.0 and the Goal suite from 66.7 to 73.6, while matching RLinf in-distribution on LIBERO. On RoboTwin2.0, PARL-VLA achieves 46.9% real-robot success over 405 trials, improving over RLinf at 37.0%. Pruning removes 55% of visual tokens on average during training and yields about 1.3× faster rollout policy forward and up to 1.74× faster policy updates. These results support a practical design principle for efficient VLA-RL: pruning should be part of the RL loop to enable co-adaptation while maintaining on-policy consistency.

Acknowledgements

This work is supported in part by Shenzhen Science and Technology Program under grant KJZD20240903104008012, Shenzhen Science and Technology Program under grant ZDCY20250901113000001, CUHK-CUHK(SZ)-GDSTC Joint Collaboration Fund No. 2025A0505000053, and GuangDong Key Laboratory of Big Data Computing (2021B1212040002).

References

1. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022)
3. Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In: European Conference on Computer Vision. pp. 19–35. Springer (2024)
4. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088 (2025)
5. Chen, Y., Tian, S., Liu, S., Zhou, Y., Li, H., Zhao, D.: Conrft: A reinforced fine-tuning method for vla models via consistency policy. arXiv preprint arXiv:2502.05450 (2025)
6. Chen, Z., Niu, R., Kong, H., Wang, Q., Xing, Q., Fan, Z.: Tgrpo: Fine-tuning vision-language-action model via trajectory-wise group relative policy optimization. arXiv preprint arXiv:2506.08440 (2025)
7. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023)
8. Fei, S., Wang, S., Ji, L., Li, A., Zhang, S., Liu, L., Hou, J., Gong, J., Zhao, X., Qiu, X.: Srpo: Self-referential policy optimization for vision-language-action models. arXiv preprint arXiv:2511.15605 (2025)
9. Fei, S., Wang, S., Shi, J., Dai, Z., Cai, J., Qian, P., Ji, L., He, X., Zhang, S., Fei, Z., et al.: Libero-plus: In-depth robustness analysis of vision-language-action models. arXiv preprint arXiv:2510.13626 (2025)
10. 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)
11. Huang, W., Zhai, Z., Shen, Y., Cao, S., Zhao, F., Xu, X., Ye, Z., Hu, Y., Lin, S.: Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. arXiv preprint arXiv:2412.00876 (2024)
12. Hung, C.Y., Sun, Q., Hong, P., Zadeh, A., Li, C., Tan, U., Majumder, N., Poria, S., et al.: Nora: A small open-sourced generalist vision language action model for embodied tasks. arXiv preprint arXiv:2504.19854 (2025)
13. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)

14. Jiang, T., Jiang, X., Ma, Y., Wen, X., Li, B., Zhan, K., Jia, P., Liu, Y., Sun, S., Lang, X.: The better you learn, the smarter you prune: Towards efficient vision-language-action models via differentiable token pruning. arXiv preprint arXiv:2509.12594 (2025)
15. Jin, P., Wang, Q., Sun, G., Cai, Z., He, P., You, Y.: Dual-actor fine-tuning of vla models: A talk-and-tweak human-in-the-loop approach. arXiv preprint arXiv:2509.13774 (2025)
16. Jin, R., Zhu, Z., Ouyang, R., Xu, S., Yue, B., Wu, Z., Liu, G.: Grounding sim-to-real generalization in dexterous manipulation: An empirical study with vision-language-action models. arXiv preprint arXiv:2603.22876 (2026)
17. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645 (2025)
18. Li, C., Wen, J., Peng, Y., Peng, Y., Zhu, Y.: Pointvla: Injecting the 3d world into vision-language-action models. IEEE Robotics and Automation Letters **11**(3), 2506–2513 (2026)
19. Li, C., Liu, J., Li, B., Gao, B., Yuan, Y., He, Y., Li, Y., Tang, J.: Dtp: A simple yet effective distracting token pruning framework for vision-language action models. arXiv preprint arXiv:2601.16065 (2026)
20. 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)
21. 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)
22. Li, W., Yuan, Y., Liu, J., Tang, D., Wang, S., Qin, J., Zhu, J., Zhang, L.: Tokenpacker: Efficient visual projector for multimodal llm. International Journal of Computer Vision **133**(10), 6794–6812 (2025)
23. Li, Y., Meng, Y., Sun, Z., Ji, K., Tang, C., Fan, J., Ma, X., Xia, S., Wang, Z., Zhu, W.: Sp-vla: A joint model scheduling and token pruning approach for vla model acceleration. arXiv preprint arXiv:2506.12723 (2025)
24. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: Expediting vision transformers via token reorganizations. arXiv preprint arXiv:2202.07800 (2022)
25. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)
26. Liu, J., Gao, F., Wei, B., Chen, X., Liao, Q., Wu, Y., Yu, C., Wang, Y.: What can rl bring to vla generalization? an empirical study. arXiv preprint arXiv:2505.19789 (2025)
27. Liu, Z., Chen, Y., Cai, H., Lin, T., Yang, S., Liu, Z., Zhao, B.: Vla-pruner: Temporal-aware dual-level visual token pruning for efficient vision-language-action inference. arXiv preprint arXiv:2511.16449 (2025)
28. 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)
29. Ma, Y., Song, Z., Zhuang, Y., Hao, J., King, I.: A survey on vision-language-action models for embodied ai. arXiv preprint arXiv:2405.14093 (2024)
30. 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. arXiv preprint arXiv:2501.09747 (2025)

31. Qu, D., Song, H., Chen, Q., Yao, Y., Ye, X., Ding, Y., Wang, Z., Gu, J., Zhao, B., Wang, D., et al.: Spatialvla: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830 (2025)
32. Shang, Y., Cai, M., Xu, B., Lee, Y.J., Yan, Y.: Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22857–22867 (2025)
33. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
34. Shi, L., Chen, S., Gao, F., Chen, Y., Chen, K., Zhang, T., Zhang, H., Zhang, W., Yu, C., Wang, Y.: Rlinf-co: Reinforcement learning-based sim-real co-training for vla models. arXiv preprint arXiv:2602.12628 (2026)
35. Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., et al.: Smolvla: A vision-language-action model for affordable and efficient robotics. arXiv preprint arXiv:2506.01844 (2025)
36. Sutton, R.S., Barto, A.G., et al.: Reinforcement learning: An introduction, vol. 1
37. Tan, S., Dou, K., Zhao, Y., Krähenbühl, P.: Interactive post-training for vision-language-action models. arXiv preprint arXiv:2505.17016 (2025)
38. Wang, Y., Ding, P., Li, L., Cui, C., Ge, Z., Tong, X., Song, W., Zhao, H., Zhao, W., Hou, P., et al.: Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. arXiv preprint arXiv:2509.09372 (2025)
39. Xiao, W., Lin, H., Peng, A., Xue, H., He, T., Xie, Y., Hu, F., Wu, J., Luo, Z., Fan, L., et al.: Self-improving vision-language-action models with data generation via residual rl. arXiv preprint arXiv:2511.00091 (2025)
40. Xing, L., Huang, Q., Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., et al.: Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. arXiv preprint arXiv:2410.17247 (2024)
41. Xu, C., Li, Q., Luo, J., Levine, S.: Rldg: Robotic generalist policy distillation via reinforcement learning. arXiv preprint arXiv:2412.09858 (2024)
42. Xu, S., Wang, Y., Xia, C., Zhu, D., Huang, T., Xu, C.: Vla-cache: Towards efficient vision-language-action model via adaptive token caching in robotic manipulation. arXiv e-prints pp. arXiv–2502 (2025)
43. Yang, Y., Wang, Y., Wen, Z., Zhongwei, L., Zou, C., Zhang, Z., Wen, C., Zhang, L.: Efficientvla: Training-free acceleration and compression for vision-language-action models. arXiv preprint arXiv:2506.10100 (2025)
44. Ye, W., Wu, Q., Lin, W., Zhou, Y.: Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 22128–22136 (2025)
45. Yu, C., Wang, Y., Guo, Z., Lin, H., Xu, S., Zang, H., Zhang, Q., Wu, Y., Zhu, C., Hu, J., et al.: Rlinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation. arXiv preprint arXiv:2509.15965 (2025)
46. Yu, W., Wang, T., Li, F., Li, J., Zhu, L.: Acˆ 2-vla: Action-context-aware adaptive computation in vision-language-action models for efficient robotic manipulation. arXiv preprint arXiv:2601.19634 (2026)
47. Zhang, Q., Cheng, A., Lu, M., Zhuo, Z., Wang, M., Cao, J., Guo, S., She, Q., Zhang, S.: [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. arXiv e-prints pp. arXiv–2412 (2024)
48. Zhang, Y., Fan, C.K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., et al.: Sparsevlm: Visual token sparsifica-

tion for efficient vision-language model inference. arXiv preprint arXiv:2410.04417 (2024)
49. Zhu, J., Zhu, Y., Lu, X., Yan, W., Li, D., Liu, K., Fu, X., Zha, Z.J.: Visionselector: End-to-end learnable visual token compression for efficient multimodal llms. arXiv preprint arXiv:2510.16598 (2025)