SAFE-Pruner: Semantic Attention–Guided Future-Aware Token Pruning for Efficient Vision-Language-Action Manipulation

Shilin Ma[1], Chubin Zhang[1], Changyuan Wang[1], Yuji Wang[1], Yue Wu[1], Zixuan Wang[1], Jingqi Tian[1], Zheng Zhu[2], and Yansong Tang[1†]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[2] GigaAI, China
{msl21@mails, tang.yansong@sz}.tsinghua.edu.cn

Abstract. Real-time inference of vision–language–action (VLA) models is essential for robotic control. While visual token pruning has shown strong potential for accelerating inference, most existing methods mainly base pruning decisions on shallow-layer cues and risk discarding visual information required by deep layers. To address this issue, we propose **SAFE-Pruner**, a plug-and-play pruning framework that incorporates attention cues of future layers into pruning decisions. Specifically, we identify semantic attention consistency, the tendency that VLA models concentrate their attention probability mass on the same semantic entity across control timesteps. Based on this observation, we design a forward-looking strategy to forecast the token saliency in deep layers, which prevents the premature removal of critical tokens and leads to more stable acceleration. We further introduce a reference timestep refresh strategy that triggers updates upon attention shifts, thereby improving forecasting accuracy and pruning reliability. Extensive experiments across diverse evaluation settings demonstrate that our method achieves up to 1.89× speedup with a minimal degradation in success rate of less than 1.5%, while outperforming state-of-the-art methods by up to 1.9%.

Keywords: Vision-Language-Action Model · Token Pruning · Inference Acceleration

1 Introduction

Vision-Language-Action (VLA) models [4,5,19,20,22,32] have become a promising route toward general and flexible robotic manipulation, building on the advances in pretrained vision-language backbones [2,3,18] and large-scale robotic datasets [12,15,17,28]. By unifying visual perception, language understanding and downstream generation in a single framework, these models enable end-to-end embodied control across diverse tasks and environments. However, the heavy computational overhead and high inference latency severely hinder their application for real-time robotic control, especially in dynamic scenarios where timely

† Corresponding author

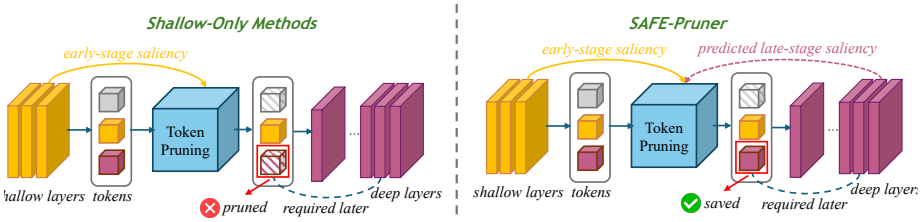


Fig. 1: Comparison of SAFE-Pruner and shallow-only token pruning methods. Shallow-only methods (left) rely on early-stage saliency, risking premature pruning of tokens critical for deep layers. SAFE-Pruner (right) comprehensively considers saliency across inference stages by predicting late-stage importance, enabling correct retention of important tokens and achieving a better accuracy-efficiency balance.

responsiveness is critical. Thus, developing lightweight acceleration methods for VLA models is an urgent and critical research challenge for enabling real-world robotic perception and control.

While traditional acceleration techniques such as distillation [13,46], quantization [27,35,38] and layer pruning [43] provide efficiency gains for VLA models, their requirement for retraining or architectural modifications severely limits their generalization capability. Recently, visual token pruning [1, 8, 25, 37, 44] has become a promising direction, which reduces computation by discarding visually redundant tokens based on attention probabilities [8, 44] or similarity scores [1,42]. Some methods further integrate VLA-specific properties into pruning decisions to enhance stability in long-horizon control [31,37].

Despite these advances, current token pruning methods for VLA models face a short-sighted selection issue. Most existing approaches [1,8,25,44] make pruning decisions using signals extracted from early intra-timestep stages like shallow-layer attention, and then permanently discard the remaining tokens, which fails to fully account for their downstream utility within the same control timestep. In VLA models, producing a single action typically involves multi-stage computation like layer-wise feature refinement from shallow to deep Transformer layers; as a result, tokens that appear less salient early on may later become informative for deep-layer reasoning and prematurely pruning such tokens can therefore cause significant performance degradation. While recent works [25, 36] attempt to address this issue, they mainly estimate late-stage token saliency from visual-token subsets already pruned in early layers, where some critical tokens may have been irreversibly removed.

To address the above limitations, we propose SAFE-Pruner, a plug-and-play framework that incorporates token saliency of different inference stages into pruning decisions to avoid premature discarding of critical tokens. Our goal is to predict intra-timestep future reasoning cues during early pruning stages based on historical frames. To achieve this goal, we first analyze how the attention scores of VLA models evolve across different frames within the same rollout. Specifically, we identify a phenomenon of semantic attention consistency: when

performing the same task across different timesteps, the model predominantly attends to patches with consistent semantic information, such as target objects to be grasped, even when their spatial locations change. Based on this phenomenon, we propose a future attention forecast strategy that uses late-stage inference information from historical frames to predict future reasoning cues of the current frame, thereby avoiding the premature pruning of tokens that will be attended to in late stages and leading to consistent performance gains across diverse benchmarks and tasks. While semantic attention consistency is well-maintained most of the time, we still observe distinct changes in attention patterns at the boundaries between two significantly different subtasks, which degrade the prediction accuracy. In addition, long-horizon forecasting may accumulate drift over time. To address these issues, we propose an adaptive key timestep selection strategy based on early-stage attention changes, which refreshes the reference when necessary to maintain consistency between the current and reference timesteps and avoid interference from outdated information. Fig. 1 provides a visual comparison of our method against conventional shallow-only approaches.

We conduct extensive and well-designed experiments to validate the effectiveness and generalization capability of our method. Specifically, we implement our approach alongside state-of-the-art open-source methods [1, 8, 25, 37, 44] across four distinct VLA architectures [19,20,22,32] and benchmarks [23,24]. The experimental results demonstrate that our method outperforms existing approaches in both inference acceleration and task performance.

Our main contributions can be summarized as follows:

- We identify and validate the semantic attention consistency phenomenon of VLA models in continuous manipulation, revealing a critical attention regularity for pruning decision-making;
- We propose a future-aware token pruning framework with an attention prediction mechanism based on semantic consistency and adaptive key timestep selection, mitigating the short-sighted pruning problem of existing methods;
- We conduct comprehensive evaluations in diverse settings, demonstrating that our method achieves up to 1.89× speedup with a negligible drop in success rate below 1.5%, while outperforming state-of-the-art methods by up to 1.9%.

2 Related Work

Vision-Language-Action Models (VLA). Vision-language-action (VLA) models aim to unify visual perception, language understanding, and action generation within a single policy for embodied manipulation [6,7,14,34]. A common recipe is to leverage a pre-trained vision-language model (VLM) [9,33] and scale learning with large collections of robot demonstrations [7, 12, 30], which enables open-vocabulary instruction following and improved generalization across tasks, objects, and environments. Recent work further explores different action decoding formulations, which include autoregressive policies [20,41], action-chunk prediction [19, 45], and diffusion-style policy heads [10, 22], as well as data-centric

strategies such as multi-task training [4, 32]. Together, these advances establish VLA as a promising paradigm for general-purpose manipulation, while also bringing increased computational demands due to dense visual tokens and long-horizon, closed-loop inference [39].

Acceleration for VLA Models. Improving the inference efficiency of VLA models is important for real-time robotic control, where latency can directly affect closed-loop stability and task success. Acceleration efforts span both system- and model-level directions, comprising both general-purpose techniques and VLA-specialized optimizations. On the system side, prior work improves throughput using mixed-precision inference [29], operator/kernel optimizations [11], and better KV-cache management [21]. On the model side, researchers study lightweight architectures [13,43], as well as training-free methods [40] that adaptively reduce computation. In particular, visual token pruning [8,25] is attractive in VLA settings because visual tokens typically dominate sequence length and attention cost. By selecting a compact subset of informative patches guided by attention features [8, 31, 36] or similarity signals [1], these methods can yield substantial speedups with limited performance degradation. Nevertheless, pruning at high ratios remains challenging, as token importance may shift across layers and different stages, motivating approaches that better anticipate downstream cues and preserve task-critical visual evidence to achieve better performance.

3 Method

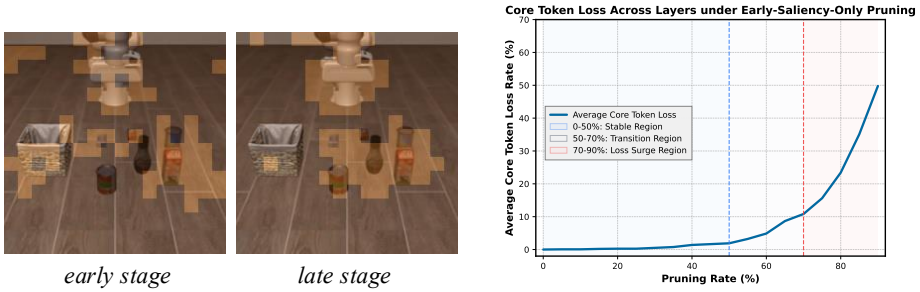
3.1 Preliminary: Attention-based Token Saliency

Current works [8,25,37,44] typically utilize the cross-attention weights to identify the important tokens and guide the pruning. Let ℓ_s denote the layer at which token pruning is performed, $\mathbf{H}_{\text{vis}}^{\ell_s} \in \mathbb{R}^{L_{\text{vis}} \times D}$ and $\mathbf{H}_{\text{query}}^{\ell_s} \in \mathbb{R}^{L_{\text{query}} \times D}$ represent the hidden state matrices of visual tokens and query tokens respectively, where L_{vis} and L_{query} denote the number of visual and query tokens, and D is the dimension of hidden states. In this work, we adopt action tokens as query tokens. Via standard multi-head projection and reshaping, we derive query matrices $\mathbf{Q}^{\ell_s} \in \mathbb{R}^{N_h \times L_{\text{query}} \times d}$ from query tokens and key matrices $\mathbf{K}^{\ell_s} \in \mathbb{R}^{N_h \times L_{\text{vis}} \times d}$ from visual tokens, with N_h being the number of attention heads and d the dimension of each head. The scaled dot-product attention $\mathbf{A}^{\ell_s}$ from query tokens to visual tokens is calculated as

$$\mathbf{A}^{\ell_s} = Softmax\left(\frac{\mathbf{Q}^{\ell_s}\left(\mathbf{K}^{\ell_s}\right)^{\top}}{\sqrt{d}}\right) \in \mathbb{R}^{N_h \times L_{\text{query}} \times L_{\text{vis}}}. \tag{1}$$

Each entry $A_{h,t,v}^{\ell_s}$ measures how strongly the t-th query token attends to the v-th visual token under head h. We aggregate attention across heads and query tokens to obtain a per-visual-token saliency score $\mathcal{S}^{\ell_s}$:

$$\mathcal{S}^{\ell_s} = \frac{1}{N_h\, L_{\text{query}}} \sum_{h=1}^{N_h} \sum_{t=1}^{L_{\text{query}}} \mathbf{A}_{h,t,v}^{\ell_s} \in \mathbb{R}^{L_{\text{vis}}}. \tag{2}$$



(a) Saliency of different stages **(b)** Core token loss rate

Fig. 2: Attention pattern differences and core token loss under early-saliency-only pruning in OpenVLA-OFT. (a) Visualization of a coarse-to-fine attention focusing regime, where early layers show broad attention and late layers focus on critical regions. (b) Core token loss rate under early-saliency pruning, rising sharply at aggressive pruning levels.

Based on the score, we only keep the top-K highest-saliency visual tokens and prune the rest to focus computation on evidence most relevant to current task. It should be noted that the pruning operation is irreversible, meaning that only this subset of visual tokens is available for the subsequent inference phase.

3.2 Visual Attention Dynamics in VLA Models

The generation of action tokens at a given control timestep typically consists of multiple distinct stages. For instance, autoregressive models involve prefill and decoding stages, single-forward models perform inference across different depth of Transformer blocks, and two-stage models include separate semantic understanding and action generation phases. To achieve favorable acceleration efficiency, pruning is typically performed in the early stage of VLA inference. This raises a critical question: *Will early-stage pruning prematurely discard visual tokens that the model will focus on in the late stages?* To investigate this issue, we thoroughly analyze how the attention patterns evolve across **intra-timestep stages** and **control timesteps**. Our key insights can be summarized as follows.
Varied concentration across intra-timestep stages. As illustrated in Fig. 2, we visualize the attention pattern differences between early and late stages based on the single-forward model OpenVLA-OFT [19]. Early attention patterns are scattered and broad, while later attention patterns become more locally concentrated and focused, reflecting a coarse-to-fine attention regime. We further quantify how this discrepancy affects pruning. At pruning rates above 70%, the probability that early-saliency-only pruning removes core tokens (the top 10% most important tokens) of other layers rises sharply, causing substantial performance degradation in conventional pruning methods.
Semantic consistency across control timesteps. Although attention patterns differ across intra-timestep stages, we observe that attention patterns at

Fig. 3: Visualization of semantic attention consistency and subtask transitions. We visualize key attention tokens (orange) across five frames of a robotic pick-and-place episode. For most steps, attention remains anchored to task-relevant semantic entities rather than spatial positions, demonstrating semantic consistency. An abrupt shift in attention occurs at Frame 13, which coincides with the subtask transition. This temporal evolution highlights both the stability of semantic attention within a subtask and the abrupt shift at the boundary between two distinct subtasks.

the same stage across different control timesteps, especially the more concentrated late-stage attention, exhibit striking semantic consistency. As depicted in Fig. 3, the gripper moves from the middle to the right to grasp the soup and then to the left to place it; despite this large displacement, attention remains highly concentrated around the gripper region in every frame. This suggests that VLA models consistently attend to patches sharing the same task-critical *semantics* throughout the episode, which in turn enables the future attention prediction strategy based on historical frames. It is worth noting that prior work [25,36] has reported substantial overlap in attention patterns between adjacent timesteps, while our observation extends this finding to longer horizons and reveals a more general form of semantic consistency.

Sudden changes at subtask transitions. Although semantic attention consistency holds for most steps, we sometimes observe abrupt shifts in attention at subtask transitions. For example, in Fig. 3, frame 13 attends more to the basket compared with frame 12. This change corresponds to the transition from the "pick up" subtask to the "place" subtask, where the task-critical semantic entity changes from the object to the target box.

These observations jointly motivate our overall pruning framework. The varied attention patterns across intra-timestep stages demonstrate that effective pruning must consider the saliency of different stages to avoid the premature discarding of critical tokens of the late stage. However, since pruning is executed in the early stage, the late-stage saliency is unavailable at the pruning moment. To resolve this, we leverage the observed semantic consistency across control timesteps to predict the current step's late-stage saliency using information from prior steps, as introduced in Sec. 3.3. Furthermore, to mitigate unreliable predictions caused by the observed sudden attention shifts during subtask transitions as well as accumulated drift, we propose an adaptive reference timestep selection strategy, as introduced in Sec. 3.4. The pipeline of our full approach is illustrated in Fig. 4.

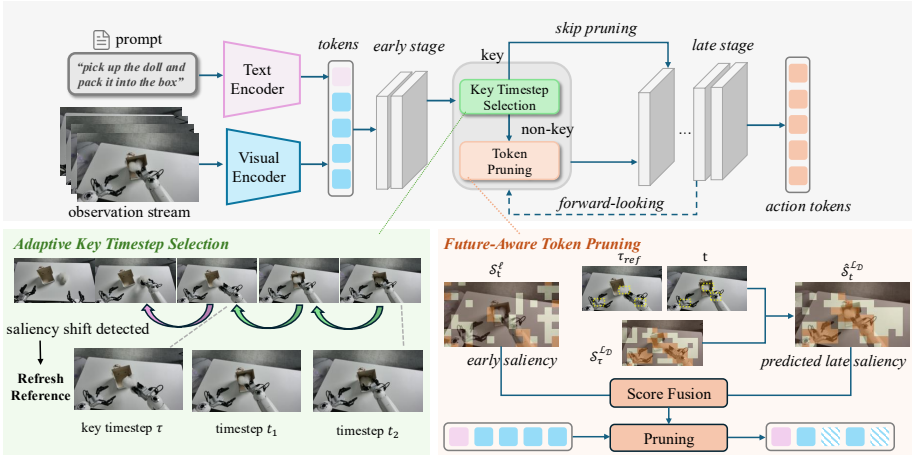


Fig. 4: The overall framework of SAFE-Pruner. Given a text prompt and a sequence of visual frames, multimodal tokens are produced by text and visual encoders. A saliency-based keyframe detection module identifies frames with significant saliency shifts. For keyframes, pruning is skipped; for non-keyframes, our future-aware token pruning module predicts late-stage token saliency from historical frames, fuses it with early-stage saliency scores, and removes non-critical tokens. This forward-looking strategy incorporates pruning impact on future stages into decision, achieving a superior accuracy–efficiency balance in VLA inference.

3.3 Future-Aware Token Pruning

Inspired by the semantic attention consistency introduced in Sec. 3.2, we propose a forward-looking strategy that *forecasts* token importance in the late stages.

Consider a VLA task with a total of T control timesteps. For each control timestep, the model generates the corresponding action tokens via forward propagation through L Transformer layers. To extract representative timesteps for saliency prediction, we first select a subset of timesteps $\mathcal{T}_{\text{key}}$ as **key timesteps**. For each key timestep $\tau \in \mathcal{T}_{\text{key}}$, we do not prune at any layer and cache the full-token saliency vector $\{\mathcal{S}_\tau^\ell, \ell = 0, 1, \ldots, L-1\}$ for every layer l.

For each non-key timestep $t \notin \mathcal{T}_{\text{key}}$, we perform pruning at the shallow layer ℓ_s. We first calculate $\mathcal{S}_t^{\ell_s}$ as early-stage saliency. To predict the late stage saliency that we cannot directly compute at ℓ_s, we take the most recent key timestep τ_{ref} as reference timestep, which is defined as:

$$\tau_{ref} = \max\{\tau \in \mathcal{T}_{\text{key}} : \tau < t\}. \tag{3}$$

Afterwards, we identify the semantically closest visual token at the reference key timestep τ_{ref} for each visual token at the current timestep t, by minimizing the cosine distance between the corresponding tokens. Based on this visual token correspondence, we transfer the precomputed saliency scores $\mathcal{S}_{\tau_{ref}}^\ell$ of every layer l from τ_{ref} to t, denoted as $\{\hat{\mathcal{S}}_t^\ell, \ell = 0, 1, \ldots, L-1\}$.

To better approximate the importance of tokens in the late stages, we predict saliency vectors for a set of deeper layers $\mathcal{L}_d$ and aggregate them by averaging:

$$\hat{\mathcal{S}}_t^{\mathcal{L}_d} = \frac{1}{|\mathcal{L}_d|} \sum_{\ell \in \mathcal{L}_d} \hat{\mathcal{S}}_t^{\ell} \in \mathbb{R}^{L_{\text{vis}}}. \tag{4}$$

We then fuse the shallow-layer saliency and the deeper-layer predicted saliency to generate the final token importance scores:

$$\mathbf{s}_t = (1-\lambda)\, \mathcal{S}_t^{\ell_s} + \lambda\, \hat{\mathcal{S}}_t^{\mathcal{L}_d}, \qquad \lambda \in [0,1]. \tag{5}$$

Based on $\mathbf{s}_t$, we select the top-K visual tokens with the highest importance scores for retention. By jointly considering token importance in both current and future inference stages during pruning, our approach effectively mitigates the risk of prematurely discarding tokens that become critical in later stages, thereby achieving a better balance between performance and efficiency.

3.4 Adaptive Key Timestep Selection

The key timestep set $\mathcal{T}_{\text{key}}$ directly determines the quality of our future-saliency forecast and therefore plays a crucial role in the overall pruning performance. A straightforward baseline is to sample key timesteps at a fixed temporal interval. However, as discussed in Sec. 3.2, attention may change abruptly at subtask transitions; in such cases, an interval-based strategy can miss the transition point and propagate an outdated saliency prior across multiple consecutive steps, leading to suboptimal pruning performance. We argue that the validity of the saliency prior is more correlated with whether two control timesteps belong to the *same subtask* than with their temporal distance. Motivated by this observation, we adaptively refresh key timesteps by detecting attention shifts via the cosine similarity of saliency maps.

We use the shallow-layer saliency vector $\mathcal{S}_t^{\ell_s} \in \mathbb{R}^{L_{\text{vis}}}$ as a lightweight proxy of the current attention pattern and compare it with the forecasted saliency vector in the same layer. We compute the cosine similarity:

$$\kappa_t = \frac{\left\langle \mathcal{S}_t^{\ell_s},\, \hat{\mathcal{S}}_t^{\ell_s} \right\rangle}{\left\| \mathcal{S}_t^{\ell_s} \right\|_2 \left\| \hat{\mathcal{S}}_t^{\ell_s} \right\|_2} \in [-1, 1]. \tag{6}$$

When $\kappa_t < \gamma$ (where γ is a threshold), we detect an attention shift and refresh the key timestep:

$$t \in \mathcal{T}_{\text{key}} \iff \kappa_t < \gamma. \tag{7}$$

At such a key timestep, we trigger full-token inference and update the stored saliency prior for subsequent forecasting; otherwise we reuse the latest key timestep τ_{ref} for saliency transfer and forecasting, followed by the token pruning.

Overall, at each timestep t, SAFE-Pruner: (i) computes shallow saliency $\mathcal{S}_t^{\ell_s}$; (ii) forecasts layer-wise saliency from the latest keyframe cache via Eq. (4); (iii)

decides whether t is a keyframe using Eqs. (6)–(7) in current shallow layer. If $t \in \mathcal{T}_{\text{key}}$, we *disable pruning* and run full-token inference to directly output the action $\mathbf{a}_t$, while computing reliable saliency maps in different layers and updating the keyframe cache $\{\mathcal{S}_\tau^\ell\}$ for subsequent forecasting. If $t \notin \mathcal{T}_{\text{key}}$, we skip the expensive full-token computation and instead rely on the forecasted deep-layer saliency $\hat{\mathcal{S}}_t^{\mathcal{L}_d}$ (Eq. (4)) together with the shallow metrics $\mathcal{S}_t^{\ell_s}$ to fuse scores and select tokens using Eq. (5), and run the original VLA policy on the pruned token set to output the action $\mathbf{a}_t$. This yields a training-free, plug-and-play pruning pipeline that refreshes priors *only when necessary* while preserving the accuracy benefits of future-aware saliency.

4 Experiments

4.1 Experiment Setups

Foundation Models. We implement our method on four VLA backbones with different architectures and decoding paradigms to test generalization. Specifically, we use: (1) OpenVLA [20], a token-based VLA that predicts discretized actions conditioned on a fused vision encoder and an LLM backbone; (2) OpenVLA-OFT [19], which replaces step-by-step decoding with parallel decoding plus action chunking and continuous-action regression; (3) CogACT [22], a componentized VLA that separates vision/language reasoning from a specialized diffusion-based action module; and (4) $\pi_{0.5}$ [32], which couples a VLM backbone with a flow-matching action expert. These models cover the mainstream architectural families in modern VLAs, and consistent gains across them verify our method's cross-architecture generalization.

Comparison Methods. We compare our approach against the unaccelerated vanilla model as primary baseline, together with a set of representative state-of-the-art open-source token pruning methods. These include VLM acceleration methods like FastV [8], SparseVLM [44], and DivPrune [1], as well as recent efficiency methods developed specifically for VLA models: VLA-Cache [37] and VLA-Pruner [25]. For all baseline methods, we conduct thorough hyperparameter tuning under the same experimental setup to obtain their best possible performance, ensuring a fair and convincing comparison.

Evaluation Metrics. We use three evaluation metrics, success rate, FLOPs and latency, to jointly assess task performance and inference efficiency. These metrics are widely adopted in current token pruning works [25, 37], allowing us to comprehensively and fairly evaluate the performance-efficiency trade-off of different methods. For latency evaluation, following VLA-Cache [37], we report the per-timestep inference latency of the VLM backbone. Unless otherwise specified, all latency measurements are conducted on an NVIDIA RTX 4090 GPU. The reported latency is averaged over all evaluated control timesteps, including both key and non-key timesteps, and incorporates all additional computational overhead introduced by our method.

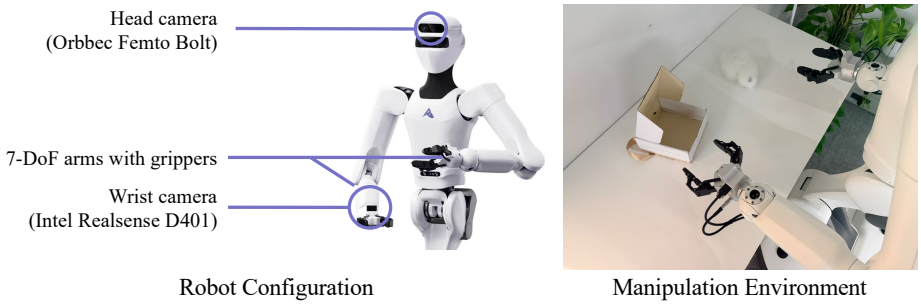


Fig. 5: Visualization of the real-world experiment setup. Left: The Astribot S1 dual-arm platform, equipped with 7-DoF arms, grippers, and wrist-mounted and head-mounted RGB cameras for visual perception. Right: The manipulation environment used to evaluate task performance across diverse robotic scenarios.

4.2 Evaluation Benchmarks.

LIBERO [24]. We adopt the LIBERO benchmark as our primary evaluation suite, which assesses the generalization capability of VLA models across four manipulation dimensions: Spatial reasoning, Object-centric interaction, Goal-directed execution, and Long-horizon planning. Structured into four distinct suites, LIBERO features 10 unique manipulation tasks per suite, with 50 evaluation episodes allocated to each task—resulting in 2000 episodes. We conduct experiments on OpenVLA, OpenVLA-OFT and $\pi_{0.5}$ in this benchmark. To maintain consistent and strict experimental comparability with prior VLA work, we adhere to the standardized evaluation protocol and publicly released official weights for all baseline models.

SIMPLER [23]. We evaluate our method on the SIMPLER simulator to assess its robustness in bridging the simulation-to-reality gap. This simulator provides two complementary settings for a Google robot arm: Visual Matching and Variant Aggregation, each encompassing four distinct manipulation tasks. We select CogACT [22] as our base model for this benchmark, as it provides official pre-trained weights tailored for SIMPLER, enabling us to reliably quantify the performance-efficiency trade-offs of different methods.

Real Robot Experiment. We conduct real-world evaluations using the Astribot S1 [16] dual-arm platform. With its chassis and torso immobilized, control is strictly limited to the 14-DoF arms and grippers. Visual observations are provided by two wrist-mounted RGB cameras alongside a head-mounted camera that actively tracks the workspace center. The detailed configuration is visualized in Fig. 5. Using 200 to 1,000 expert demonstrations collected via VR teleoperation, we fine-tune task-specific $\pi_{0.5}$ models for three distinct scenarios designed to test a spectrum of robotic capabilities. These include *Pick and Place* to assess basic manipulation across 10 different objects, *Throw Basketball* to test small-object dexterity, and *Pack Doll*—a long-horizon, multi-stage task requiring precise geometric coordination to pick up a doll, place it inside a box, and

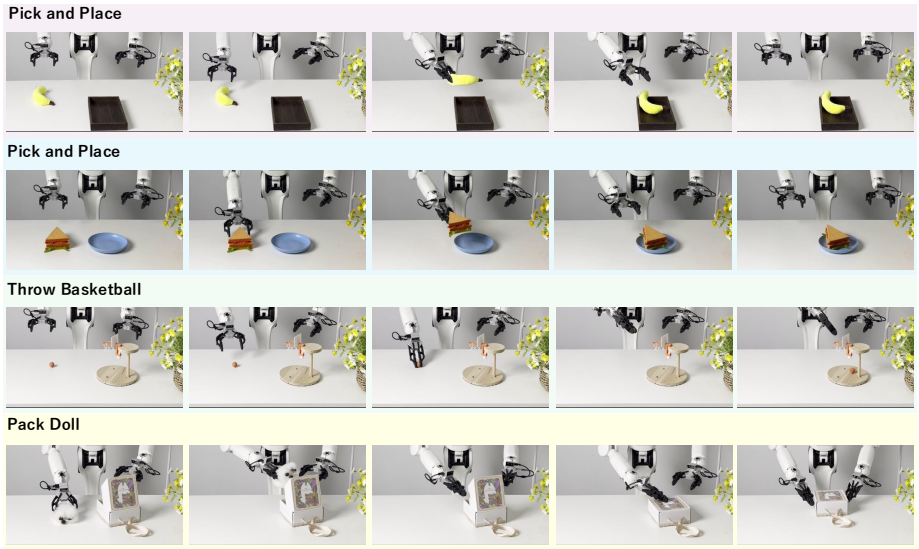


Fig. 6: Real-world robotic task scenarios for evaluation. Visualization of three distinct task scenarios on the Astribot S1 platform: Pick and Place task for basic manipulation, Throw Basketball for small-object dexterity, and Pack Doll for long-horizon multi-stage coordination, each designed to assess different robotic capabilities.

secure the lid. The task setups are visualized in Fig. 6. Each trained model is evaluated across 200 trials to comprehensively measure its performance.

4.3 Main Results

Results on LIBERO. The results of the LIBERO experiment are listed in Tab. 1, which demonstrates the broad applicability and significant superiority of our method. Due to differences in model architecture and size, visual token pruning methods exhibit varying acceleration effects across different models. Nevertheless, our method consistently achieves an excellent performance-efficiency balance across diverse model architectures when compared to the unaccelerated vanilla model—incurring only a negligible performance drop of no more than 1.5% in average success rate while reducing computational cost and latency by up to 60% and 47%, respectively. Furthermore, by integrating future attention cues into pruning decisions, our method maintains consistently superior performance over existing methods even under higher acceleration settings. This is evident in the OpenVLA-OFT [19] experiment: while achieving substantially better acceleration in terms of FLOPs from 2.126 T to 1.722 T and backbone latency from 50.24 ms to 37.22 ms, our method still achieves the highest success rate, which is 1.9% higher than the best performance of existing methods. These results confirm the superiority of our method in both efficiency and performance across different architectures of mainstream VLA models. We further

Table 1: Comparison on LIBERO benchmarks. Best results among acceleration methods are highlighted in **bold**. Our method consistently achieves the best trade-off between efficiency and performance across models with different architectures, delivering the lowest FLOPs and latency while maintaining top success rates.

Model	Method	Success Rate(%) ↑					FLOPs (T)↓	Latency (ms)↓
		Spatial	Object	Goal	Long	Average		
OpenVLA	Vanilla	84.6	86.6	78.2	53.2	75.7	1.862	48.24
	FastV	81.8	70.6	71.8	44.6	67.2	0.811	34.63
	SparseVLM	80.6	72.4	68.2	46.6	67.0	0.876	34.32
	DivPrune	78.6	62.8	66.2	43.2	62.7	0.832	37.18
	VLA-Cache	78.2	71.2	70.6	45.8	66.5	0.941	35.21
	VLA-Pruner	82.0	84.4	**77.8**	**52.6**	74.2	0.793	36.47
	Ours	**82.2**	**84.8**	77.6	52.0	**74.2**	**0.742**	**32.18**
OpenVLA-OFT	Vanilla	98.4	98.2	96.4	94.2	96.8	3.970	70.25
	FastV	97.8	92.8	95.6	91.8	94.5	2.141	50.24
	SparseVLM	**98.6**	91.8	94.8	92.0	94.3	2.289	52.15
	DivPrune	97.2	90.0	93.4	89.6	92.6	2.126	53.87
	VLA-Cache	97.0	92.2	94.6	91.0	93.7	2.730	58.14
	VLA-Pruner	97.4	93.0	94.0	92.0	94.1	2.234	54.53
	Ours	98.0	**98.0**	**96.2**	**93.4**	**96.4**	**1.722**	**37.22**
$\pi_{0.5}$	Vanilla	98.2	98.0	98.0	91.8	96.5	2.115	35.28
	FastV	96.4	96.8	95.6	89.0	94.5	1.612	28.64
	SparseVLM	94.6	97.2	94.8	89.2	94.0	1.713	29.03
	DivPrune	93.8	96.6	92.4	88.0	92.7	1.576	30.55
	VLA-Cache	94.8	96.0	**96.4**	88.2	93.9	1.632	28.14
	VLA-Pruner	95.8	97.2	95.6	88.0	94.2	1.583	32.64
	Ours	**97.0**	**98.8**	96.0	**90.2**	**95.5**	**1.482**	**24.33**

evaluate the compatibility of SAFE-Pruner with HiF8 [26] quantization, as reported in Tab. 2. The combined framework reduces FLOPs by 56.4% and GPU memory by 38.1% relative to the vanilla OpenVLA-OFT model, while maintaining a comparable success rate. These results demonstrate that SAFE-Pruner is compatible with quantization methods, enabling simultaneous improvements in computational and memory efficiency.

Results on SIMPLER. We evaluate the sim-to-real robustness and efficiency of our method on the SIMPLER benchmark (Tab. 3), focusing on the CogACT backbone across two domain-shift settings: Visual Matching (VM) and Variant Aggregation (VA). Overall, our approach achieves state-of-the-art inference acceleration while preserving task performance comparable to the unpruned CogACT baseline, demonstrating its effectiveness in realistic robotic scenarios. In the VM setting, our method attains an average success rate of 74.5%, nearly identical to the 74.8% success rate achieved by the vanilla CogACT, while reducing FLOPs to just 37.4% of the baseline and delivering a 1.73× speedup. This outperforms all competing methods: FastV delivers only a 1.21× speedup at 42.0%

Table 2: Compatibility with quantization. Combining SAFE-Pruner with HiF8 jointly reduces computational cost and GPU memory while maintaining performance comparable to the vanilla OpenVLA-OFT model.

Method	Success Rate (%) ↑					FLOPs (T) ↓	GPU Memory (GB) ↓
	Spatial	Object	Goal	Long	Average		
Vanilla	98.4	98.2	96.4	94.2	96.8	3.970	17.83
Ours	98.0	98.0	96.2	93.4	96.4	1.722	18.31
Ours + HiF8	98.8	98.6	95.8	93.4	96.6	1.729	11.03

Table 3: Comparison on SIMPLER benchmarks. Best results among acceleration methods are highlighted in **bold**. The results demonstrate that our approach yields the highest speedup and lowest computational cost, while maintaining a success rate competitive with or better than the baseline and other acceleration methods.

SIMPLER	Method	Success Rate(%) ↑					FLOPs (%)↓	Speed up↑
		Pick	Move	Drawer	Apple	Average		
Visual Matching	CogACT	91.3	85.0	71.8	50.9	74.8	100.0	1.00×
	Random	9.7	20.4	53.5	0.0	20.9	58.5	1.20×
	FastV	**92.6**	81.4	69.8	**52.4**	74.1	42.0	1.21×
	VLA-Cache	92.0	83.3	70.5	51.6	74.4	80.1	1.38×
	Ours	91.3	**83.8**	**70.8**	51.9	**74.5**	**37.4**	**1.73**×
Variant Aggregation	CogACT	89.6	80.8	28.3	46.6	61.3	100.0	1.00×
	Random	4.0	16.1	15.6	0.0	8.9	58.5	1.20×
	FastV	91.4	78.6	27.6	**50.6**	62.1	42.0	1.19×
	VLA-Cache	91.7	**79.3**	**32.5**	45.8	**62.3**	82.6	1.37×
	Ours	**92.1**	79.2	28.0	48.1	61.9	**36.2**	**1.67**×

FLOPs, while VLA-Cache offers a modest 1.38× speedup at 80.1% FLOPs. In the VA setting, our method maintains an average success rate of 61.9%, which is on par with the baseline 61.3%, and achieves a 1.67× speedup with 36.2% FLOPs retention, outperforming the 1.37× speedup of VLA-Cache and the marginal acceleration of FastV. These results confirm that our token pruning strategy generalizes to sim-to-real scenarios, making it suitable for deployment on real robotic systems with strict latency constraints. In some settings, the accelerated model slightly outperforms the vanilla baseline, a phenomenon also observed in prior work [37]. We assume that this is because our method filters out tokens irrelevant to the task, thus making the model more concentrated on the useful information and improving the success rate.

Results on Real Robot. We further validate our method in real-world deployment on the Astribot S1 dual-arm platform. We conduct a comprehensive evaluation of **FastV** and **Ours**, reporting task success rate together with computational cost and VLM backbone inference latency measured on an NVIDIA RTX 3090. The results are shown in Tab. 4. Compared with the unpruned $\pi_{0.5}$

Table 4: Comparison on real robot manipulation tasks. 'Latency' refers to the inference latency of the VLM backbone, tested on an NVIDIA RTX 3090 GPU.

Method	Success Rate ↑				FLOPs	Latency
	Pick and Place	*Throw Basketball*	*Pack Doll*	*Average*	(T)↓	(ms)↓
$\pi_{0.5}$	**94%**	77%	**73%**	**81.3%**	2.264	80.36
+ **FastV**	82%	69%	56%	69.0%	1.857	56.93
+ **Ours**	90%	**81%**	67%	79.3%	**1.543**	**43.56**

Table 5: Ablation Studies. The ablation results verify the effectiveness of each proposed component, showing that both the forecasting mechanism and the keyframe selection strategy contribute to improved performance–efficiency balance, and their combination yields the best overall results.

Method	Success Rate(%) ↑					FLOPs	Latency
	Spatial	Object	Goal	Long	Average	(T)↓	(ms)↓
Vanilla	98.4	98.2	96.4	94.2	96.8	3.970	70.25
w/o forecast	97.8	93.0	95.8	91.6	94.5	2.141	50.31
w/o adaptivity	**98.2**	96.2	96.0	92.6	95.8	2.236	51.98
Ours	98.0	**98.0**	**96.2**	**93.4**	**96.4**	**1.722**	**37.22**

baseline, both pruning methods substantially reduce inference cost and latency, decreasing FLOPs from 2.264 T to 1.857 T with FastV and to 1.543 T with our method, while reducing latency from 80.36 ms to 56.93 ms and 43.56 ms, respectively. This efficiency gain comes with a drop in average success rate from 81.3% to 69.0% for FastV and to 79.3% for our method, showing that our approach preserves performance better. Overall, our method achieves a consistently better accuracy–efficiency trade-off than FastV. The gains are most pronounced on *Throw Basketball* from 69% to 81%, which requires precise grasping of small objects. We also observe improvements from 82% to 90% on *Pick and Place* and from 56% to 67% on the long-horizon *Pack Doll* task. These results indicate that incorporating future-aware attention cues yields more reliable pruning decisions in real robotic control, preserving task-critical visual evidence while reducing inference cost.

4.4 Analysis

To validate the effectiveness of each component in our method, we perform systematic ablation studies using OpenVLA-OFT [19] on the LIBERO benchmarks [24], evaluating four experimental configurations: (1) the unaccelerated model (vanilla), (2) token pruning only using shallow attention without future state forecasting (w/o forecast), (3) future-layer prediction guided pruning with fixed-interval keyframe selection (w/o adaptivity) and (4) our complete pipeline

(Ours). As presented in Tab. 5, the vanilla baseline achieves a high average success rate of 96.8% but introduces heavy computational costs with 3.970 T FLOPs and 70.25 ms latency; although w/o forecast delivers obvious acceleration, it suffers from a clear performance drop to 94.5% due to the premature discarding of task-relevant tokens. By incorporating future-layer information to assist token pruning, w/o adaptivity alleviates accuracy degradation and boosts the average success rate to 95.8%, but it fails to account for attention shift at subtask boundaries, and full-token inference on fixed-interval keyframes limits further acceleration. In contrast, our full approach incorporates an adaptive key-timestep selection mechanism, which improves prediction robustness while reducing the number of key timesteps. As a result, it achieves the highest average success rate of 96.4%, along with the lowest FLOPs of 1.722 T and latency of 37.22 ms among acceleration variants, demonstrating that future forecasting and adaptive reference refreshing jointly balance performance and inference efficiency.

5 Conclusion

We present SAFE-Pruner, a training-free, plug-and-play token pruning framework that leverages the observed phenomenon, semantic attention consistency, to forecast late-stage token saliency from historical keyframes and thus avoid premature removal of task-critical visual evidence. By fusing shallow-stage cues with predicted deep-stage saliency and adaptively selecting key timesteps, our method successfully preserves downstream reasoning while substantially reducing computation and inference delay. Experiments across multiple VLA backbones, simulation benchmarks and real-robot trials demonstrate that SAFE-Pruner consistently improves the accuracy–efficiency trade-off, achieving up to 1.89× speedup with negligible loss in task success. We believe SAFE-Pruner not only offers a practical path toward low-latency VLA inference for real-time robotic control, but also provides new insights into the attention dynamics of VLA models.

Acknowledgements

This work was supported by Shenzhen Science and Technology Program under Grant JCYJ20240813111903006.

References

1. Alvar, S.R., Singh, G., Akbari, M., Zhang, Y.: Divprune: Diversity-based visual token pruning for large multimodal models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 9392–9401 (2025)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)

3. Beyer, L., Steiner, A., Pinto, A.S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., et al.: Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726 (2024)
4. 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)
5. 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)
6. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M.G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.W.E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., Zitkovich, B.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818 (2023)
7. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2023)
8. 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)
9. Chen, X., Djolonga, J., Padlewski, P., Mustafa, B., Changpinyo, S., Wu, J., Ruiz, C.R., Goodman, S., Wang, X., Tay, Y., Shakeri, S., Dehghani, M., Salz, D., Lucic, M., Tschannen, M., Nagrani, A., Hu, H., Joshi, M., Pang, B., Montgomery, C., Pietrzyk, P., Ritter, M., Piergiovanni, A., Minderer, M., Pavetic, F., Waters, A., Li, G., Alabdulmohsin, I., Beyer, L., Amelot, J., Lee, K., Steiner, A.P., Li, Y., Keysers, D., Arnab, A., Xu, Y., Rong, K., Kolesnikov, A., Seyedhosseini, M., Angelova, A., Zhai, X., Houlsby, N., Soricut, R.: Pali-x: On scaling up a multilingual vision and language model. arXiv preprint arXiv:2305.18565 (2023)
10. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research **44**(10-11), 1684–1704 (2025)
11. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in neural information processing systems **35**, 16344–16359 (2022)
12. Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., Finn, C.: Robonet: Large-scale multi-robot learning. arXiv preprint arXiv:1910.11215 (2020)
13. Dong, S., Fu, C., Gao, H., Zhang, Y.F., Yan, C., Wu, C., Liu, X., Shen, Y., Huo, J., Jiang, D., Cao, H., Gao, Y., Sun, X., He, R., Shan, C.: Vita-vla: Efficiently teaching vision-language models to act via action expert distillation. arXiv preprint arXiv:2510.09607 (2025)
14. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K.,

Zeng, A., Mordatch, I., Florence, P.: Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)
15. Ebert, F., Yang, Y., Schmeckpeper, K., Bucher, B., Georgakis, G., Daniilidis, K., Finn, C., Levine, S.: Bridge data: Boosting generalization of robotic skills with cross-domain datasets. arXiv preprint arXiv:2109.13396 (2021)
16. Gao, G., Wang, J., Zuo, J., Jiang, J., Zhang, J., Zeng, X., Zhu, Y., Ma, L., Chen, K., Sheng, M., et al.: Towards human-level intelligence via human-like whole-body manipulation. arXiv preprint arXiv:2507.17141 (2025)
17. Kalashnikov, D., Varley, J., Chebotar, Y., Swanson, B., Jonschkowski, R., Finn, C., Levine, S., Hausman, K.: Mt-opt: Continuous multi-task robotic reinforcement learning at scale. arXiv preprint arXiv:2104.08212 (2021)
18. Karamcheti, S., Nair, S., Balakrishna, A., Liang, P., Kollar, T., Sadigh, D.: Prismatic vlms: Investigating the design space of visually-conditioned language models. In: Forty-first International Conference on Machine Learning (2024)
19. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645 (2025)
20. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
21. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J.E., Zhang, H., Stoica, I.: Efficient memory management for large language model serving with pagedattention. arXiv preprint arXiv:2309.06180 (2023)
22. Li, Q., Liang, Y., Wang, Z., Luo, L., Chen, X., Liao, M., Wei, F., Deng, Y., Xu, S., Zhang, Y., et al.: Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. arXiv preprint arXiv:2411.19650 (2024)
23. Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., Levine, S., Wu, J., Finn, C., Su, H., Vuong, Q., Xiao, T.: Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941 (2024)
24. 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)
25. 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)
26. Luo, Y., Zhang, Z., Wu, R., Liu, H., Jin, Y., Zheng, K., Wang, M., He, Z., Hu, G., Chen, L., et al.: Ascend hifloat8 format for deep learning. arXiv preprint arXiv:2409.16626 (2024)
27. Lv, Z., Fan, Z., Tian, Q., Zhang, W., Zhuang, Y.: Enhancing post-training quantization via future activation awareness. arXiv preprint arXiv:2602.02538 (2026)
28. Mees, O., Hermann, L., Rosete-Beas, E., Burgard, W.: Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. IEEE Robotics and Automation Letters **7**(3), 7327–7334 (2022)
29. Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., Wu, H.: Mixed precision training. In: International Conference on Learning Representations (ICLR) (2018)
30. O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In: 2024 IEEE

International Conference on Robotics and Automation (ICRA). pp. 6892–6903. IEEE (2024)
31. Pei, X., Chen, Y., Xu, S., Wang, Y., Shi, Y., Xu, C.: Action-aware dynamic pruning for efficient vision-language-action manipulation. arXiv preprint arXiv:2509.22093 (2025)
32. Physical Intelligence, Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (2021)
34. Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S.G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J.T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., de Freitas, N.: A generalist agent. arXiv preprint arXiv:2205.06175 (2022)
35. Wang, C., Wang, Z., Xu, X., Tang, Y., Zhou, J., Lu, J.: Q-vlm: Post-training quantization for large vision-language models. Advances in Neural Information Processing Systems **37**, 114553–114573 (2024)
36. Wang, H., Xu, J., Pan, J., Zhou, Y., Dai, G.: Specprune-vla: Accelerating vision-language-action models via action-aware self-speculative pruning. arXiv preprint arXiv:2509.05614 (2025)
37. Xu, S., Wang, Y., Xia, C., Zhu, D., Huang, T., Xu, C.: Vla-cache: Efficient vision-language-action manipulation via adaptive token caching. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
38. Xu, Y., Yang, Y., Fan, Z., Liu, Y., Li, Y., Li, B., Zhang, Z.: Qvla: Not all channels are equal in vision-language-action model's quantization. arXiv preprint arXiv:2602.03782 (2026)
39. 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)
40. Yue, Y., Wang, Y., Kang, B., Han, Y., Wang, S., Song, S., Feng, J., Huang, G.: Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. Advances in Neural Information Processing Systems **37**, 56619–56643 (2024)
41. Zhang, C., Wang, J., Gao, Z., Su, Y., Dai, T., Zhou, C., Lu, J., Tang, Y.: Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos. arXiv preprint arXiv:2601.04061 (2026)
42. Zhang, Q., Liu, M., Li, L., Lu, M., Zhang, Y., Pan, J., She, Q., Zhang, S.: Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. arXiv preprint arXiv:2506.10967 (2025)
43. Zhang, R., Dong, M., Zhang, Y., Heng, L., Chi, X., Dai, G., Du, L., Du, Y., Zhang, S.: Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. arXiv preprint arXiv:2503.20384 (2025)

44. 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 sparsification for efficient vision-language model inference. arXiv preprint arXiv:2410.04417 (2024)
45. Zhao, T.Z., Kumar, V., Levine, S., Finn, C.: Learning fine-grained bimanual manipulation with low-cost hardware. arXiv preprint arXiv:2304.13705 (2023)
46. Zhu, Y., Ma, S., Wang, H., Li, A., Jing, Y., Tang, Y., Chen, L., Lu, J., Zhou, J.: Varestorer: One-step var distillation for real-world image super-resolution. arXiv preprint arXiv:2604.21450 (2026)