DART-VLA: DYNAMIC ACTION-RELEVANCE TOKEN PRUNING FOR EFFICIENT VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models map multimodal observations and instructions to continuous control actions. However, the large number of visual tokens imposes substantial computational overhead on the language-model backbone, hindering real-time closed-loop inference. Existing VLA pruning methods primarily rely on semantic attention or static importance estimates, leaving the potential of action signals for guiding token-level pruning largely underexplored. Our key insight is that a visual token can be better valued by its influence on the predicted action, as this directly reflects its contribution to the decision process. Based on this insight, we propose DART-VLA, a dynamic action-relevance token pruning framework for efficient vision-language-action models. DART-VLA backpropagates gradients from the action outputs to estimate each visual token's influence on action prediction and prunes tokens accordingly. DART-VLA first adapts the overall retention budget to the current motion state. Considering the distinct functional roles of the main and wrist views, we further allocate the token budget between them based on end-effector motion and independently retain the tokens most relevant to action prediction within each view. DART-VLA achieves a 97.40% average task success rate while delivering a 1.26× inference speedup. Moreover, at a comparable inference speedup, DART-VLA outperforms the strongest pruning baseline by 1.90 % in average task success.

1 INTRODUCTION

Vision-Language-Action (VLA) models jointly encode visual observations, language instructions, and robot states to generate executable actions, providing a promising paradigm for general-purpose embodied perception, decision-making, and control (Brohan et al., 2023; Zitkovich et al., 2023; Kim et al., 2025b; Black et al., 2025a). To capture sufficient environmental context, mainstream VLA models typically encode observations from main and wrist-mounted cameras into dense visual tokens, which are jointly processed with language and state tokens by the language-model backbone (Kim et al., 2025a; Pei et al., 2026). Because visual tokens dominate the input sequence and are repeatedly processed at every step of closed-loop control, they incur substantial computational overhead, increase action latency, and limit control frequency (Xu et al., 2025; Li et al., 2026; Yang et al., 2025b). Given the strict latency requirements of robotic control,reducing the cost of visual-token processing is therefore critical to efficient VLA inference.

Token pruning is a promising approach of alleviating this bottleneck by shortening vision tokens and reducing backbone computation (Li et al., 2026; Pei et al., 2026; Yang et al., 2025b; Liu et al., 2025b; Wang et al., 2026; Cheng et al., 2026). Existing VLA pruning methods typically select tokens based on semantic relevance, attention strength, inter-frame visual variation, or geometric priors (Xu et al., 2025; Li et al., 2026; Pei et al., 2026; Yang et al., 2025b; Liu et al., 2025b; Wang et al., 2026; Cheng et al., 2026). Even when action information is incorporated, it is primarily used to determine pruning schedules, adjust retention ratios, or construct attention-based proxy scores, rather than to directly quantify each visual token's influence on action outputs (Li et al., 2026; Pei et al., 2026; Liu et al., 2025b; Wang et al., 2026; Cheng et al., 2026). This misalignment between the pruning objective and action generation can discard control-critical visual evidence under tight token budgets,

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

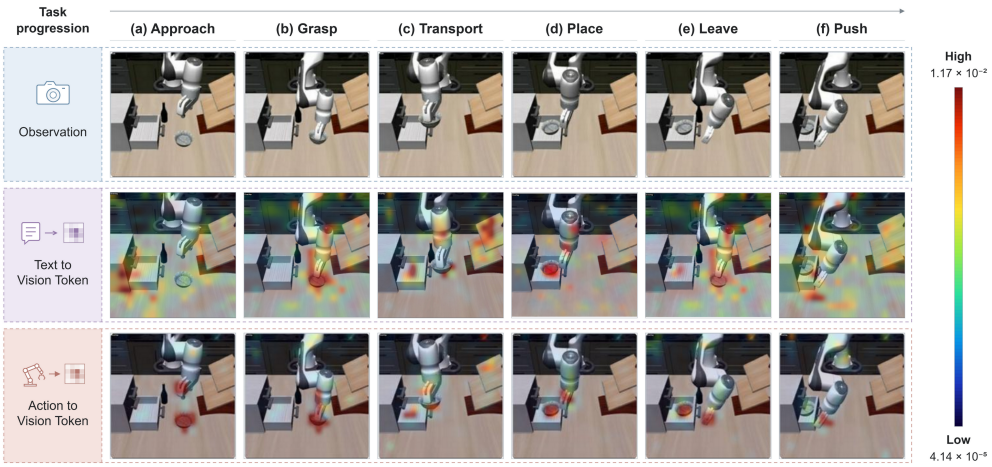


Figure 1: Visual-token relevance during manipulation. Action-gradient relevance highlights interaction regions more selectively than text-to-vision attention. Warmer colors indicate higher scores.

leading to substantial task-performance degradation (Pei et al., 2026; Liu et al., 2025b; Wang et al., 2026; Cheng et al., 2026).

To examine whether action outputs can guide visual-token selection, we compare text-to-vision attention with action-gradient relevance across manipulation stages (Figure 1). In these examples, text-to-vision attention assigns high scores to spatially dispersed regions without consistently prioritizing interaction-relevant cues. Action-gradient relevance more selectively highlights the end effector, manipulated object, and interaction sites, with high-scoring regions shifting as manipulation progresses. These observations suggest that text-to-vision attention alone may be insufficient to identify control-critical visual tokens, motivating a pruning criterion that incorporates the local sensitivity of action predictions to visual information.

Motivated by this observation, we propose DART-VLA, a training-free dynamic action-relevance token-pruning framework that directly leverages action outputs for efficient VLA inference. The framework uses action gradients to estimate the local sensitivity of the current action prediction to each visual token, thereby prioritizing visual evidence critical to action generation. Specifically, we introduce Action-Relative Dynamic Budgeting (AR-DB), which compares the current motion magnitude with episode-level historical statistics to determine the total retention budget. It retains fewer tokens during visually redundant coarse-motion phases while preserving sufficient visual evidence for fine-grained manipulation. Given this budget, we further introduce Action-Motion-Conditioned View Allocation(AMC-VA), which distributes tokens between the main-camera and wrist-camera views according to the current end-effector motion, coordinating their complementary global and local cues across manipulation stages.

Our main contributions are summarized as follows:

1. We propose DART-VLA, a training-free, action-gradient-guided visual-token pruning framework that directly selects tokens using the sensitivity of action outputs to visual token.
2. We develop Action-Relative Dynamic Budgeting (AR-DB), which adjusts the total retention budget by comparing the current planned-motion magnitude with its historical mean, balancing coarse-motion efficiency and fine-manipulation accuracy.
3. Given this budget, we further introduce Action-Motion-Conditioned View Allocation, which dynamically distributes tokens between the main-camera and wrist-camera views according to the current end-effector motion, coordinating their complementary global and local cues to mitigate pruning-induced performance degradation.
4. Across four LIBERO suites, DART-VLA achieves 97.40% average success with a 1.26× speedup on OpenVLA-OFT, exceeding the strongest pruning baseline by 1.90 % at comparable speedup. On OpenPI's $\pi_{0.5}$, it achieves 97.35% success with a 1.15× speedup and a 0.55-point gain over the strongest pruning baseline.

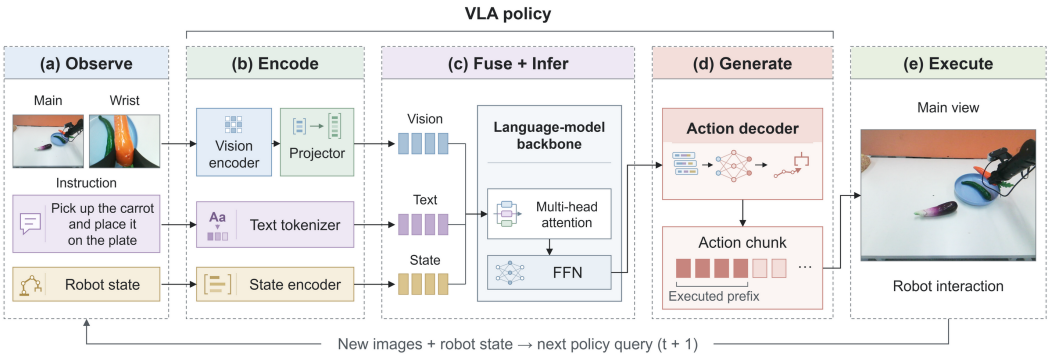


Figure 2: Closed-loop VLA inference. Main- and wrist-camera observations are encoded and projected into visual tokens, then fused with language and robot-state tokens for action prediction. Execution yields updated observations and robot state for the next policy query.

2 Related Work and Background

2.1 Vision-Language-Action Models

Vision-Language-Action (VLA) models map visual observations, language instructions, and robot states to executable actions. RT-1 introduced scalable Transformer-based action prediction, while RT-2 transferred web-scale vision-language knowledge to robotic control (Brohan et al., 2023; Zitkovich et al., 2023). Octo and OpenVLA further advanced cross-embodiment generalist policies and open-source VLA modeling (Ghosh et al., 2024; Kim et al., 2025b), while recent models support continuous control through diffusion- or flow-based decoders and action chunking (Liu et al., 2025a; Li et al., 2024a; Black et al., 2025a;b; Kim et al., 2025a). Complementary work explores multimodal pretraining and prompting, cross-robot datasets, video pretraining, 3D representations, and action tokenization (Driess et al., 2023; Jiang et al., 2023; Open X-Embodiment Collaboration, 2024; Li et al., 2024b; Wu et al., 2024; Zhen et al., 2024; Pertsch et al., 2025; Qu et al., 2025). Despite architectural differences, these models largely share the closed-loop pipeline in Figure 2: multi-view images are encoded and projected into visual tokens, optionally selected, and fused with language and state tokens for processing by the language-model backbone. An action decoder maps the resulting hidden states to discrete actions or continuous action chunks, whose execution produces updated observations and robot states for the next query. This repeated processing motivates pruning after visual-feature projection and before language-model backbone inference, reducing the multimodal input length while leaving the vision encoder unchanged (Kim et al., 2025a;b).

2.2 VLA Inference Acceleration and Visual Token Pruning

Visual-token reduction targets visual or semantic redundancy through selection, merging, and pruning in vision Transformers (Rao et al., 2021; Yin et al., 2022; Bolya et al., 2023) and LVLMs (Chen et al., 2024; Shang et al., 2025; Zhang et al., 2025; Yang et al., 2025a; Xing et al., 2025), without explicit alignment with action prediction. For VLA models, VLA-Cache reuses key–value representations of temporally stable tokens (Xu et al., 2025), while SP-VLA and ADP adapt pruning using motion histories and action trajectories, respectively (Li et al., 2026; Pei et al., 2026). EfficientVLA dynamically compresses tokens, whereas VLA-Pruner scores them using action-decoder attention (Yang et al., 2025b; Liu et al., 2025b). SpecPrune-VLA and VLA-IAP employ self-speculative pruning and interaction-aware selection, respectively (Wang et al., 2026; Cheng et al., 2026). Nevertheless, existing methods mainly use action signals to control pruning schedules or budgets, or rely on attention-based proxies for token selection, rather than directly measuring each token's local sensitivity to the predicted action. DART-VLA instead derives action-relevance scores from the local sensitivity of predicted actions to individual visual tokens, aligning visual-token selection more directly with action generation.

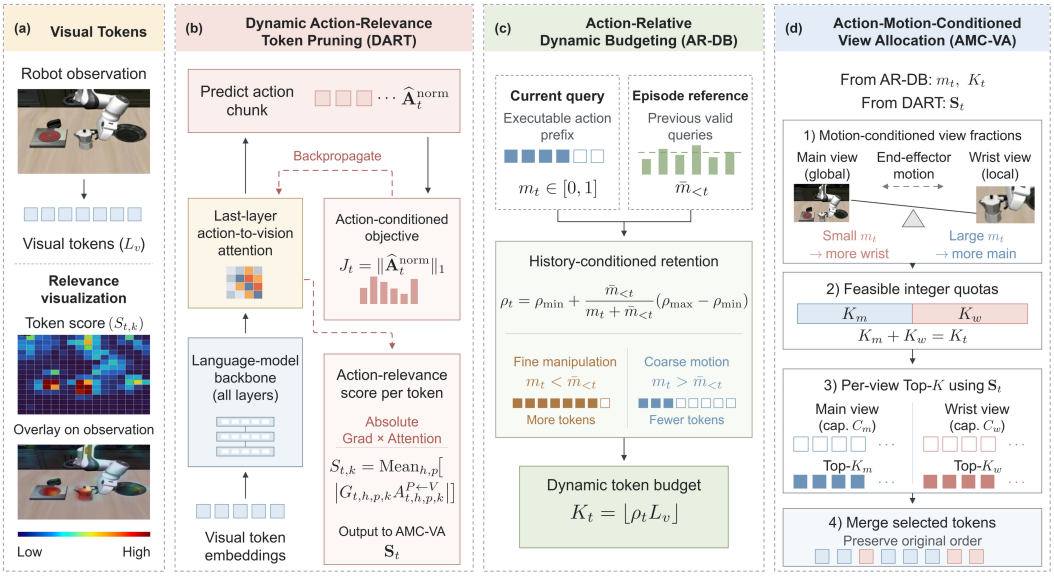


Figure 3: Overview of DART-VLA. At each full-scoring query, Grad×Attention scores visual tokens by action relevance, AR-DB determines the retention budget, and AMC-VA allocates it across views; the resulting indices are applied to subsequent pruned queries.

3 METHOD

This section presents DART-VLA, a training-free, action-gradient-guided visual-token pruning framework. At each periodic full-scoring query, DART-VLA constructs a differentiable objective from the predicted action chunk and backpropagates action gradients to compute an action-relevance score for each visual token. AR-DB compares the current planned-motion magnitude with the mean of prior valid motion magnitudes in the episode to determine the total retention budget, while AMC-VA allocates this budget between the main and wrist views according to the current end-effector motion. DART-VLA therefore determines which tokens to retain, how many to retain, and how to distribute them across views, reducing inference overhead in the language-model backbone while preserving action-relevant visual information. We first describe action-relevance scoring, token selection, and temporal reuse.

3.1 DYNAMIC ACTION-RELEVANCE TOKEN PRUNING (DART)

As shown in Figure 1, text-to-vision attention can emphasize visually salient regions without directly assessing their relevance to the current action, potentially diverting limited token budgets from control-critical cues. VLA models should therefore use actions to guide visual-token pruning, prioritizing evidence relevant to the current control decision. DART addresses this mismatch through action-gradient scoring, which incorporates local action sensitivity and more selectively highlights the end effector, manipulated object, and interaction sites across manipulation stages. This better aligns token selection with action prediction, providing a more targeted criterion for preserving control-relevant visual evidence.

Action-Conditioned Objective. As illustrated in the action-relevance scoring stage of Figure 3, DART aggregates the predicted action chunk into a scalar objective and backpropagates it to the final-layer action-to-vision attention, yielding a relevance score for each visual token. At a full-scoring query t, projected visual tokens $\mathbf{X}_{v,t} \in \mathbb{R}^{L_v \times d}$ yield a normalized action chunk of length L_a and dimension d_a. We aggregate the chunk into a differentiable scalar for relevance estimation:

$$\widehat{\mathbf{A}}_t^{\text{norm}} = [\widehat{\mathbf{a}}_{t,1}^{\text{norm}}; \ldots; \widehat{\mathbf{a}}_{t,L_a}^{\text{norm}}] \in \mathbb{R}^{L_a \times d_a}, \qquad J_t = \|\widehat{\mathbf{A}}_t^{\text{norm}}\|_1 = \sum_{i=1}^{L_a} \sum_{j=1}^{d_a} |\widehat{a}_{t,i,j}^{\text{norm}}|.$$

Here, L_v and d are the visual-token count and embedding dimension, respectively, and $\widehat{\mathbf{a}}_{t,i}^{\text{norm}}$ is the i-th normalized action. During scoring, gradients are backpropagated only to estimate token relevance, while all model parameters remain frozen without any training or updates.

Action-Relevance Scoring. From the final language-model layer, we extract the post-softmax attention from action-prediction queries P to visual-token keys V:

$$\mathbf{A}_t^{P\leftarrow V} \in \mathbb{R}^{N_h \times N_a \times L_v}.$$

Here, $P \leftarrow V$ denotes information flow from visual tokens to action-prediction positions; N_h and N_a denote the number of attention heads and the number of action-prediction positions, respectively. The element $A_{t,h,p,k}^{P\leftarrow V}$ represents the attention weight connecting action position p to visual token k in head h. Using the scalar objective J_t, we define

$$G_{t,h,p,k} = \frac{\partial J_t}{\partial A_{t,h,p,k}^{P\leftarrow V}}, \qquad Q_{t,h,p,k} = \left| G_{t,h,p,k} A_{t,h,p,k}^{P\leftarrow V} \right|.$$

The gradient measures how the scalar action objective responds to a small change in an attention weight. Multiplying its magnitude by the corresponding attention weight combines local sensitivity with connection strength. Under the same small proportional weakening of a connection, a higher score indicates a larger magnitude of the first-order change in the objective. Taking absolute values captures sensitivity in either direction and prevents cancellation during aggregation. The resulting score provides a local estimate of action relevance for token selection, rather than a causal attribution of the effects of token removal.

For each sample, we independently average relevance scores over attention heads and action-prediction positions:

$$S_{t,k} = \frac{1}{N_h N_a} \sum_{h=1}^{N_h} \sum_{p=1}^{N_a} Q_{t,h,p,k}.$$

The vector $\mathbf{S}_t = [S_{t,1}, \ldots, S_{t,L_v}]$ is termed the action-relevance score vector, where $S_{t,k}$ measures the local relevance of visual token k to the action prediction at query t.

Token Selection and Temporal Reuse. AR-DB determines the total number of visual tokens to retain, and AMC-VA distributes this budget between the main and wrist views. Within each view, we rank the corresponding entries of the action-relevance score vector $\mathbf{S}_t$ and retain the highest-scoring tokens up to the view-specific token quota determined by AMC-VA. The selected indices are then merged in their original visual-token order, and the corresponding positional indices are preserved when constructing the reduced multimodal sequence.

To amortize the scoring overhead, DART-VLA performs full scoring periodically and uses pruned inference between refreshes. Each full-scoring query predicts the current action using all visual tokens before updating the relevance scores, retention budget, and selected indices. These updates apply only to subsequent pruned queries. At each pruned query, the latest images are independently encoded and projected, and the stored indices select the current visual embeddings passed to the language-model backbone. Visual features are therefore recomputed at each policy query, while the retention budget and selected indices remain fixed between refreshes. The next full-scoring query recomputes the selection using the latest observation and action prediction, allowing it to adapt as manipulation progresses. The refresh interval is measured in policy queries, and both the selection state and motion history are reset at episode boundaries.

3.2 Action-Relative Dynamic Budgeting (AR-DB)

A fixed retention ratio cannot accommodate the changing visual-information requirements across manipulation stages. AR-DB therefore adapts the number of retained visual tokens to the current planned motion, assigning smaller budgets to coarse movements and larger budgets to fine-grained manipulation. AR-DB determines how many tokens to retain, while DART identifies which tokens are most action-relevant.

Planned-Motion Magnitude. At each full-scoring query t, we measure motion using the executable prefix of the predicted action chunk, i.e., the actions scheduled before the next policy query. The

root-mean-square magnitude of the clipped translation and rotation components, excluding the gripper command, defines a dimensionless motion descriptor $m_t \in [0, 1]$. Let n_t denote the number of actions in the executable prefix, and let $C_{t,p,j} \in [-1, 1]$ denote the clipped pose increment of the p-th action along dimension j. We compute

$$m_t = \sqrt{\frac{1}{6n_t} \sum_{p=1}^{n_t} \sum_{j=1}^{6} C_{t,p,j}^2}.$$

This descriptor serves as a proxy for the current manipulation stage.

History-Conditioned Budget Adaptation. Let $\bar{m}_{<t}$ denote the mean of the valid motion magnitudes from previous full-scoring queries in the current episode. AR-DB compares m_t with this historical reference and computes the visual-token retention ratio as

$$\rho_t = \rho_{\min} + \frac{\bar{m}_{<t}}{m_t + \bar{m}_{<t}} \left(\rho_{\max} - \rho_{\min}\right),$$

where $\rho_{\min}$ and $\rho_{\max}$ are the predefined lower and upper retention bounds, respectively. The resulting token budget is $K_t = \lfloor \rho_t L_v \rfloor$, where L_v is the total number of visual tokens. A smaller m_t relative to the episode history increases K_t to preserve visual evidence for fine manipulation, whereas a larger m_t reduces the budget during coarse motion. We choose $\rho_{\min}$ such that $K_t \geq 2$ whenever both views are available, enabling AMC-VA to retain at least one token from each view.

If no valid historical reference is available or both motion magnitudes are zero, we set $\rho_t = \rho_{\max}$. The updated budget is applied only to subsequent pruned queries. After computing the budget, the current valid motion magnitude is added to the episode history, which is reset at the beginning of each episode.

3.3 Action-Motion-Conditioned View Allocation (AMC-VA)

As illustrated in Figure 3, at each full-scoring query, AMC-VA distributes the total retention budget K_t produced by AR-DB between the main and wrist views according to the planned-motion magnitude m_t. The main view provides global workspace context, whereas the wrist view captures local interaction details. A fixed split cannot adapt to changing visual requirements, while joint ranking may concentrate the budget in a single view. AMC-VA therefore shifts the allocation toward the main view as motion increases and toward the wrist view as it decreases, converts the resulting continuous fractions into feasible integer quotas that respect view capacities and preserve K_t, and retains the most action-relevant tokens within each view.

Motion-Conditioned View Allocation. Because the two views contain the same number of candidate visual tokens, their budget fractions depend only on m_t. Let $\tau > 0$ control the sensitivity of the allocation to motion. We compute

$$\pi_m = \frac{\exp(m_t/\tau)}{\exp(m_t/\tau) + \exp((1 - m_t)/\tau)}, \qquad \pi_w = 1 - \pi_m,$$

where π_m and π_w denote the budget fractions assigned to the main and wrist views, respectively. When $m_t = 0.5$, the budget is divided equally between the two views. A larger m_t shifts the allocation toward the main view, whereas a smaller m_t favors the wrist view; τ controls the strength of this shift. If m_t is unavailable, we set $m_t = 0.5$.

Integer Quotas and Token Selection. We obtain the view-specific token quotas by setting

$$K_m = \mathrm{round}(\pi_m K_t), \qquad K_w = K_t - K_m.$$

We adjust the quotas to respect view capacities and retain at least one token per view while preserving $K_m + K_w = K_t$. We then rank tokens independently within each view using the corresponding entries of the action-relevance score vector $\mathbf{S}_t$, retaining the top-K_m main-view tokens and top-K_w wrist-view tokens. This respects the motion-conditioned allocation while prioritizing action-relevant visual evidence within each view's assigned quota. The selected indices are merged in their original visual-token order, with positional indices preserved, and reused to select the current visual embeddings in subsequent pruned queries until the next full-scoring update.

Table 1: Performance and efficiency comparison of visual-token pruning methods on OpenVLA-OFT and OpenPI backbones

Method	Success Rate ↑					FLOPs	Latency	Control Freq	Speed
	Spatial	Object	Goal	Long	Average	(T) ↓	(ms) ↓	(Hz) ↑	up
OpenVLA-OFT	98.0%	97.2%	97.0%	95.2%	96.85%	3.97	68.45	83.06	1.00×
VLA-Cache	97.80%	97.80%	96.2%	96.2%	97.0%	3.07	62.37	86.38	1.09×
ADP	97.4%	98.2%	96.60%	94.6%	96.7%	3.04	57.53	92.6	1.18×
SP-VLA	97.60%	96.30%	93.70%	94.40%	95.50%	2.874	53.48	94.95	1.27×
DART-VLA	**98.0%**	98.0%	**97.0%**	**96.6%**	**97.40%**	3.06	54.06	94.72	1.26×
OpenPI($\pi_{0.5}$)	97.8%	98.4%	97.6%	94.0%	96.95%	2.046	74.63	79.72	1.00×
VLA-Cache	97.2%	98.8%	97.8%	93.4%	96.8%	1.510	70.01	85.16	1.06×
ADP	96.8%	97.8%	97.2%	90.0%	95.45%	1.487	62.32	93.63	1.17×
SP-VLA	96.4%	97.6%	96.8%	90.4%	95.30%	1.496	63.20	90.20	1.15×
DART-VLA	**98.2%**	**99.0%**	97.6%	**94.6%**	**97.35%**	1.491	64.68	90.81	1.15×

4 Experiment Results

4.1 Experiments Setup

Evaluation Setup. We evaluate DART-VLA on all four LIBERO suites—Spatial, Object, Goal, and Long (Liu et al., 2023)—following the official protocol, with ten tasks per suite and 50 rollouts per task (2,000 per configuration). We use OpenVLA-OFT (Kim et al., 2025a) and OpenPI's $\pi_{0.5}$ (Black et al., 2025b;a) with their native action heads and decoding protocols. OpenVLA-OFT uses a jointly fine-tuned checkpoint, while $\pi_{0.5}$ uses the same `pi05_libero` checkpoint across suites. Baselines include the corresponding unpruned backbone, VLA-Cache (Xu et al., 2025), ADP (Pei et al., 2026), and SP-VLA (Li et al., 2026). Within each backbone, all methods share checkpoints, observations, rollout counts, and hardware: an NVIDIA GeForce RTX 4090 GPU with 24 GB memory. We report per-suite and average success rates, average visual-token retention, FLOPs, inference latency, control frequency, and speedup over the unpruned backbone.

Implementation Details. Unless otherwise specified, we use last-layer Grad×Attention scoring and set the AMC-VA temperature to $\tau = 1.0$. The AR-DB bounds $(\rho_{\min}, \rho_{\max})$ are $(0.25, 0.50)$ for OpenVLA-OFT and $(0.70, 0.75)$ for $\pi_{0.5}$, with full scoring every two and three policy queries, respectively. These bounds apply to pruned queries; full-scoring queries retain all visual tokens. Motion descriptors use executable prefixes of eight and five actions, respectively. For $\pi_{0.5}$, scoring is performed at the final denoising step using an objective over the first five actions and seven control dimensions. Token-selection states and motion histories are reset at the beginning of each episode.

4.2 Main Results

OpenVLA-OFT. We evaluate DART-VLA on OpenVLA-OFT across all four LIBERO suites, with AR-DB dynamically retaining 25%–50% of visual tokens in pruned queries. As shown in Table 1, DART-VLA achieves the highest average success rate of 97.40%, exceeding the unpruned backbone and VLA-Cache by 0.55 and 0.40 percentage points, respectively. Relative to the unpruned backbone, it reduces FLOPs by 22.9% to 3.06 T and latency by 21.0% to 54.06 ms, with a reported speedup of 1.26×. At latency comparable to SP-VLA (54.06 vs. 53.48 ms), DART-VLA improves average success by 1.90 percentage points, offering a better performance–efficiency trade-off.

OpenPI ($\pi_{0.5}$). To assess applicability to a second backbone, we evaluate DART-VLA on OpenPI's $\pi_{0.5}$ using the same four suites, with AR-DB dynamically retaining 70%–75% of visual tokens in pruned queries. DART-VLA achieves the highest average success rate of 97.35%, improving over the unpruned backbone and VLA-Cache by 0.40 and 0.55 percentage points, respectively. Meanwhile, FLOPs decrease by 27.1% to 1.491 T and latency by 13.3% to 64.68 ms relative to the unpruned backbone, yielding a 1.15× speedup. Compared with SP-VLA at the same reported speedup, DART-VLA achieves 2.05 percentage points higher average success, extending the observed benefits to this backbone.

4.3 COMPONENT ABLATION

Table 2: Ablation experiment. **(a)** Component-wise ablation of DART-VLA. **(b)** Ablation of visual-token scoring criteria. **(c)** Ablation of gradient-propagation depth. **(d)** Sensitivity to the visual-token pruning ratio. SR denotes average success rate; Lat. and Freq. denote latency and control frequency.

Method	SR (%)↑	FLOPs (T)↓	Lat. (ms)↓	Freq. (Hz)↑	Speed up
OpenVLA-OFT	96.85	3.97	68.45	83.06	1.00
OFT+DART	96.75	3.240	56.53	92.75	1.18
+ AR-DB	97.20	3.238	56.81	91.17	1.18
+ AMC-VA	97.15	3.18	55.92	92.48	1.20
DART-VLA	**97.40**	**3.06**	**54.06**	**94.72**	**1.26**

(a)

Method	SR (%)↑	FLOPs (T)↓	Lat. (ms)↓	Freq. (Hz)↑	Speed up
OpenPI($\pi_{0.5}$)	96.95	2.046	74.63	79.72	1.00
Random	42.7	1.491	64.72	88.63	1.15
Gradient only	96.85	1.490	64.76	88.93	1.15
Attention only	95.05	1.491	64.58	89.02	1.15
Grad×Attn	**97.35**	1.491	64.68	**90.81**	**1.15**

(b)

Method	SR (%)↑	FLOPs (T)↓	Lat. (ms)↓	Freq. (Hz)↑	Speed up
OpenPI($\pi_{0.5}$)	96.95	2.046	74.63	79.72	1.00
4 layers	97.30	1.520	66.02	87.65	1.12
3 layers	96.8	1.513	65.40	89.42	1.14
2 layers	97.05	1.496	65.12	89.87	1.14
1 layer	**97.35**	**1.491**	**64.68**	**90.81**	**1.15**

(c)

Method	SR (%)↑	FLOPs (T)↓	Lat. (ms)↓	Freq. (Hz)↑	Speed up
OpenVLA-OFT	96.85	3.97	68.45	83.06	1.00
25% pruning	96.90	3.486	59.76	88.72	1.13
50% pruning	97.20	3.004	54.05	94.78	1.25
75% pruning	97.10	2.879	49.75	99.00	1.36
90% pruning	93.60	**2.641**	**45.60**	**104.6**	**1.50**

(d)

Component contributions. To assess the individual and combined effects of AR-DB and AMC-VA, we add them separately and jointly to DART on OpenVLA-OFT across all four LIBERO suites. In pruned queries, DART and DART+AMC-VA retain 50% of visual tokens, while variants with AR-DB dynamically retain 25%–50%. Table 2(a) shows that DART alone reduces FLOPs to 3.240 T and latency to 56.53 ms, with a 0.10-percentage-point decrease in success relative to the unpruned backbone. AR-DB improves success by 0.45 percentage points with nearly unchanged FLOPs, whereas AMC-VA improves success by 0.40 percentage points and reduces latency to 55.92 ms. Their combination improves success by 0.65 percentage points over DART alone while achieving the lowest FLOPs and latency among the ablated variants, supporting their complementary contributions.

4.4 ANALYSIS

Token-scoring criterion. To isolate the effect of token scoring, we vary only the ranking criterion on OpenPI's $\pi_{0.5}$ across all four LIBERO suites, with 2,000 rollouts per configuration. All criteria share the same AR-DB rule with retention bounds of 70%–75% for pruned queries and the refresh settings in Section 4.1. As shown in Table 2(b), random selection achieves 42.70% success, compared with 95.05% for attention-only, 96.85% for gradient-only, and 97.35% for Grad×Attention. With nearly identical FLOPs and latency among the non-random criteria, these results support combining attention magnitude with action sensitivity for token ranking.

Gradient-propagation depth. To examine whether deeper relevance gradients improve token selection, we vary propagation depth from one to four using the same $\pi_{0.5}$ checkpoint, dynamic-budget rule, and rollout protocol as the scoring ablation. Depth denotes the number of final language-model Transformer layers included in relevance-gradient computation. Table 2(c) shows no consistent success improvement with increasing depth: one layer achieves 97.35%, compared with 96.80%–97.30% for deeper variants. Using one instead of four layers reduces FLOPs from 1.520 T to 1.491 T and latency from 66.02 ms to 64.68 ms, supporting our default choice of one-layer propagation.

Pruning-ratio sensitivity. To assess the effect of pruning strength, we evaluate DART-VLA on the jointly fine-tuned OpenVLA-OFT checkpoint across all four LIBERO suites, using fixed pruning ratios of 25%, 50%, 75%, and 90% in pruned queries and 2,000 rollouts per configuration. Grad×Attention scoring, one-layer gradient propagation, and all non-budget settings remain fixed. Table 2(d) shows 96.90%–97.20% success at 25%–75% pruning. At 75%, success is 97.10% with a 1.36× speedup; 90% increases speedup to 1.50× but reduces success to 93.60%. Thus, 75%

Table 3: Comparison of success rate on real robot tasks.

Method	Success Rate ↑					FLOPs (T) ↓	Latency (ms) ↓	Control Freq (Hz) ↑	Speed up
	Task1	Task2	Task3	Task4	Average				
OpenPI($\pi_{0.5}$)	92%	88%	85%	78%	85.75%	2.032	78.44	41.85	1.00×
+ DART-VLA	94%	91%	82%	83%	87.5%	1.372	60.26	52.72	1.30×



Figure 4: Real world experiments. We conduct the experiments on a single-arm ALOHA platform.

yields the largest speedup among the tested ratios without an observed loss in success relative to the unpruned backbone.

4.5 Results on Real Robot

Evaluation Setup. We evaluate DART-VLA on a single-arm ALOHA platform with fixed external and wrist-mounted RGB cameras. The policy receives both views, a natural-language instruction, and a seven-dimensional state comprising six joint positions and the gripper state, and predicts joint-space action chunks with gripper commands. We compare OpenPI's $\pi_{0.5}$ with and without DART-VLA on the four tasks in Table 3, using the same checkpoint, observations, and action-execution settings. AR-DB dynamically retains 50%–75% of visual tokens in pruned queries, with full scoring every three policy queries. Full-scoring queries retain all visual tokens, and the AMC-VA temperature is $\tau = 1.0$. As in simulation, policy inference runs on a server workstation equipped with an NVIDIA GeForce RTX 4090 GPU.

Main Results. To assess performance beyond simulation, we examine task success and inference efficiency under this real-robot configuration, with representative task sequences shown in Figure 4. Table 3 shows that average success increases from 85.75% to 87.50%, a gain of 1.75 percentage points. FLOPs and inference latency decrease by 32.5% and 23.2%, respectively, yielding a 1.30× speedup, while control frequency rises from 41.85 to 52.72 Hz. Task4 shows the largest success gain, from 78% to 83%, while Task1 and Task2 improve by 2 and 3 percentage points, respectively. However, success on Task3 decreases from 85% to 82%, indicating that the aggregate improvement does not extend uniformly to every task.

5 Conclusion

In this work, we present DART-VLA, a training-free framework for efficient VLA inference through action-aware visual-token pruning. DART-VLA directly estimates each visual token's contribution to action prediction using gradients backpropagated from action outputs, with action-to-vision attention further refining the relevance scores. It additionally adapts token allocation across camera views and dynamically adjusts the global retention budget according to motion state and manipulation phase. Extensive experiments demonstrate that DART-VLA substantially reduces visual-token computation and inference latency while preserving task performance, highlighting the effectiveness of action signals for efficient VLA deployment.

AI USE STATEMENT

Generative AI tools were not used to develop the research ideas or methods, implement the research code, conduct experiments, or analyze the results. The authors take full responsibility for the research process, the accuracy of the reported findings, and the final content of this manuscript.

REFERENCES

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems*, 2025b. URL https://www.roboticsproceedings.org/rss21/p010.html.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT but Faster. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S Ryoo, Grecia Salazar, Pannag R Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale. In *Robotics: Science and Systems*, 2023. URL https://roboticsproceedings.org/rss19/p025.html.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 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*, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/10478_ECCV_2024_paper.php.

Jintao Cheng, Haozhe Wang, Weibin Li, Gang Wang, Yipu Zhang, Xiaoyu Tang, Jin Wu, Xieyuanli Chen, Yunhui Liu, and Wei Zhang. VLA-IAP: Training-Free Visual Token Pruning via Interaction Alignment for Vision-Language-Action Models, 2026. URL https://arxiv.org/abs/2603.22991.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An Embodied Multimodal Language Model. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp.

8469–8488. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems*, 2024. URL https://www.roboticsproceedings.org/rss20/p090.html.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot Manipulation with Multimodal Prompts. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14975–15022. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/jiang23b.html.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Robotics: Science and Systems*, 2025a. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. CogACT: A Foundational Vision-Language-Action Model for Synergizing Cognition and Action in Robotic Manipulation, 2024a. URL https://arxiv.org/abs/2411.19650.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-Language Foundation Models as Effective Robot Imitators. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=lFYj0oibGR.

Ye Li, Yuan Meng, Zewen Sun, Kangye Ji, Chen Tang, Jiajun Fan, Xinzhu Ma, Shutao Xia, Zhi Wang, and Wenwu Zhu. SP-VLA: A Joint Model Scheduling and Token Pruning Approach for VLA Model Acceleration. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2506.12723.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A Diffusion Foundation Model for Bimanual Manipulation. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=yAzN4tz7oI.

Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. Bridging the Semantic-Action Gap in Visual Token Pruning for Efficient VLA Inference, 2025b. URL https://arxiv.org/abs/2511.16449.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *IEEE International Conference on Robotics and Automation*, 2024. URL https://arxiv.org/abs/2310.08864.

Xiaohuan Pei, Yuxing Chen, Siyu Xu, Yunke Wang, Yuheng Shi, and Chang Xu. Action-Aware Dynamic Pruning for Efficient Vision-Language-Action Manipulation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.22093.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In *Robotics: Science and Systems*, 2025. URL https://www.roboticsproceedings.org/rss21/p012.html.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Models. In *Robotics: Science and Systems*, 2025. URL https://www.roboticsproceedings.org/rss21/p011.html.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/747d3443e319a22747fbb873e8b2f9f2-Abstract.html.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. In *IEEE/CVF International Conference on Computer Vision*, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Shang_LLaVA-PruMerge_Adaptive_Token_Reduction_for_Efficient_Large_Multimodal_Models_ICCV_2025_paper.html.

Hanzhen Wang, Jiaming Xu, Yushun Xiang, Jiayi Pan, Yongkang Zhou, Yong-Lu Li, and Guohao Dai. SpecPrune-VLA: Accelerating Vision-Language-Action Models via Action-Aware Self-Speculative Pruning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.05614.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing Large-Scale Video Generative Pre-Training for Visual Robot Manipulation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NxoFmGgWC9.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. PyramidDrop: Accelerating Your Large Vision-Language Models via Pyramid Visual Redundancy Reduction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2410.17247.

Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. VLA-Cache: Efficient Vision-Language-Action Manipulation via Adaptive Token Caching. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f062da1973ac9ac61fc6d44dd7fa309f-Abstract-Conference.html.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer Is Better but Not Necessary in Vision-Language Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_VisionZip_Longer_is_Better_but_Not_Necessary_in_Vision_Language_CVPR_2025_paper.html.

Yantai Yang, Yuhao Wang, Zichen Wen, Zhongwei Luo, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. EfficientVLA: Training-Free Acceleration and Compression for Vision-Language-Action Models. In *Advances in Neural Information Processing Systems*, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3a2ef31a1e45908901adc0ca853a8faf-Abstract-Conference.html.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive Tokens for Efficient Vision Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Yin_A-ViT_Adaptive_Tokens_for_Efficient_Vision_Transformer_CVPR_2022_paper.html.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=80faIPZ67S.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3D-VLA: A 3D Vision-Language-Action Generative World Model. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61229–61245. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhen24a.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.