GROUNDING ATTENTION IN MOTION: PROPRIOCEPTION-GUIDED VISUAL TOKEN PRUNING WITH ADAPTIVE LAYER SELECTION FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models benefit from the perceptual generalization of pretrained vision-language backbones, but repeatedly processing hundreds of visual tokens makes closed-loop inference costly. Visual token pruning can reduce this cost, yet existing methods typically rank tokens using only vision-language interactions and prune at a fixed backbone layer, overlooking both the robot's current state and the input-dependent stability of token rankings. We introduce PROPPRUNE, a training-free framework that adapts both *which* visual tokens are retained and *where* they are pruned. A *proprio-guided importance score* uses each instruction token's attention to the proprioceptive state to weight its attention to visual tokens, grounding visual relevance in the current embodied context. An *adaptive layer selector* then measures the Jaccard similarity between top-K visual-token sets from adjacent shallow layers and defers pruning to a later layer when the shallow ranking is unstable. Both components use attention maps already produced by the pretrained VLA, requiring no fine-tuning, auxiliary model, or additional forward pass. On LIBERO, PROPPRUNE achieves the highest average success rate among the evaluated pruning baselines at both 50% and 25% visual-token retention, remaining within 0.7 and 1.3 points of the full-token model while reducing latency by 16.0% and 34.9% and FLOPs by 38.1% and 53.6%, respectively. On LIBERO-Plus, it improves over FastV under seven controlled distribution shifts and remains within 0.26 points of the full-token reference. On two real-world tasks, 50% retention matches full-token success while reducing latency by 19.1%.

1 INTRODUCTION

Vision-Language-Action (VLA) models have emerged as a unified paradigm for language-conditioned robotic manipulation, coupling large vision-language model (VLM) backbones with policy heads to predict task-conditioned action representations from raw observations and natural-language instructions (Brohan et al., 2023; Li et al., 2024; Kim et al., 2024; Black et al., 2024; Bjorck et al., 2025). This architecture brings pretrained visual and linguistic representations into robot control and has become an increasingly prominent direction in robotic manipulation.

This generality, however, comes with substantial inference cost. A VLA policy requires repeated forward passes through its large vision-language backbone throughout a rollout, causing computational costs to accumulate across policy executions and making the backbone a major inference bottleneck. Recent efficient-VLA studies therefore focus on reducing latency and increasing action-generation throughput (Kim et al., 2025; Xu et al., 2026; Yang et al., 2026). Reducing this repeated computation is important for the practical deployment of large-scale VLA policies.

Visual token reduction offers a direct way to lower this cost by removing or reusing redundant visual information. Prior work on VLMs has explored attention-based or relevance-based pruning, token selection and merging, and diversity-preserving subset selection (Chen et al., 2024; Shang et al., 2025; Zhang et al., 2024; Yang et al., 2025; Alvar et al., 2025), while recent VLA studies have

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.

considered visual-token selection and temporal reuse (Xu et al., 2026; Yang et al., 2026). Together, these studies demonstrate that substantial visual redundancy can be removed or reused to reduce inference computation.

Despite this progress, current visual-token reduction methods do not fully exploit the embodied and input-dependent signals available during VLA inference. First, the relevance of a visual region depends not only on the image and instruction but also on the robot's current state: as the robot moves through a task, the visual evidence needed for its next decision can change. Second, adjacent shallow layers may disagree on which visual tokens are important for a given observation, whereas existing pruning methods apply reduction at the same preselected layer (Chen et al., 2024; Alvar et al., 2025; Yang et al., 2026). These observations motivate a VLA pruning strategy that incorporates the robot's state into visual-token ranking and delays pruning when adjacent shallow layers disagree on which tokens should be retained.

To address these limitations, we introduce PROPPRUNE, a training-free visual token pruning framework with two components. First, a *proprio-guided importance score* measures how strongly each instruction token attends to the proprioceptive state and uses these values to weight its attention to visual tokens. This produces a visual-token ranking that incorporates the robot's current state. Second, an *adaptive layer selector* compares the top-K token sets obtained at two adjacent shallow layers. When their Jaccard similarity is sufficiently high, pruning is applied at the shallow probe layer; otherwise, pruning is deferred to a later layer. Both components use attention maps already produced by the pretrained VLA and require no fine-tuning, auxiliary model, or additional forward pass.

Our main contributions are:

- We introduce a proprio-guided importance score that grounds visual token selection in the robot's current state by coupling language-to-proprioception and language-to-vision attention from the pretrained VLA.
- We propose adaptive pruning-layer selection that uses the similarity between top-K visual-token sets at adjacent shallow layers as a ranking-stability signal, pruning early when the ranking is stable and deferring pruning when it is not.
- We demonstrate a favorable accuracy–efficiency trade-off across in-distribution LIBERO tasks, seven categories of controlled distribution shift in LIBERO-Plus, and two real-world manipulation tasks. PROPPRUNE achieves the highest average success rate among the evaluated pruning baselines at both 50% and 25% retention, with particularly large gains on long-horizon tasks.

2 RELATED WORK

2.1 VISION-LANGUAGE-ACTION MODELS

Vision-Language-Action (VLA) models extend pretrained vision-language backbones into closed-loop robotic policies that predict action representations conditioned on image observations and natural-language instructions. RT-2 (Brohan et al., 2023) first demonstrated that co-fine-tuning a web-scale VLM with robot trajectories transfers semantic and commonsense knowledge into manipulation behaviour. OpenVLA (Kim et al., 2024) scales this paradigm with a 7B-parameter backbone trained on the Open X-Embodiment dataset, while π_0 (Black et al., 2024) couples a VLM with a flow-matching action expert to support high-frequency dexterous control. More recent variants such as OpenVLA-OFT combine action chunking with parallel decoding to generate an entire action chunk in a single forward pass, substantially increasing action-generation throughput.

2.2 VISUAL TOKEN REDUCTION FOR VLMS

Visual-token reduction has been studied extensively for general VLM inference. Existing approaches broadly include attention- or relevance-based pruning, encoder-side token selection and merging, and diversity-preserving subset selection (Chen et al., 2024; Zhang et al., 2024; Shang et al., 2025; Yang et al., 2025; Alvar et al., 2025). These approaches reduce redundant visual rep-

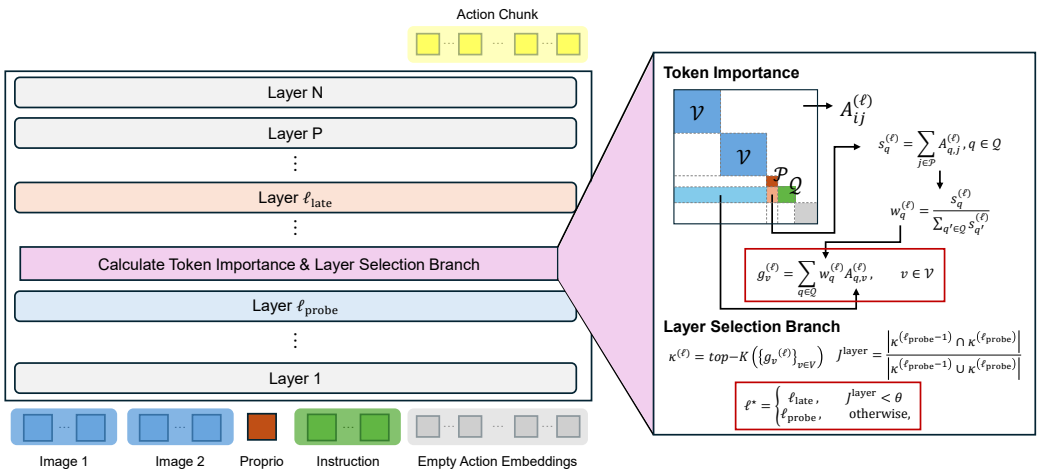


Figure 1: Overview of PROPPRUNE. Given the pretrained VLA token stream, our proprio-guided scoring mechanism first weights language-context tokens by their attention to the current proprioceptive state and then aggregates their attention to visual tokens to obtain token-importance scores. For each policy inference, we compare the top-K visual-token sets produced by two adjacent shallow layers. When the adjacent-layer Jaccard similarity is high, visual tokens are pruned at the probe layer; when their Jaccard similarity is low, indicating that the shallow-layer selection is not yet stable, pruning is deferred to a deeper layer. The framework is applied entirely at inference time without modifying or fine-tuning the pretrained model.

resentations either before or within the language backbone to lower inference cost while preserving information needed for downstream multimodal tasks.

These approaches are developed and evaluated primarily for image and video understanding, visual question answering, and multimodal reasoning. These tasks emphasize semantic understanding and reasoning, whereas VLA policies must also retain fine-grained spatial cues such as object boundaries, relative positions, and interaction regions to predict precise actions. Consequently, pruning criteria designed for general VLM tasks may not preserve all of the visual information required for precise action prediction.

2.3 EFFICIENT VLA INFERENCE

Recent work on efficient VLA inference has targeted redundancy across different components of the VLA pipeline. VLA-Cache exploits temporal continuity by identifying visual tokens that change little between adjacent frames and reusing their cached key-value representations (Xu et al., 2026). EfficientVLA combines language-layer pruning, task-aware selection of a compact and diverse set of visual tokens, and feature reuse in a diffusion-based action head (Yang et al., 2026). Together, these methods reduce inference cost through temporal reuse and structured reduction across language, vision, and action modules.

Despite this progress, the use of robot-specific inputs in visual-token scoring and the choice of where within the backbone to apply token reduction remain underexplored. Building on this direction, our work studies visual-token pruning in a pretrained VLA backbone by incorporating proprioceptive input into importance scoring and using the Jaccard similarity between top-K selections from two adjacent shallow layers to select the pruning layer.

3 METHOD

We present PROPPRUNE, a training-free framework that prunes visual tokens in a pretrained VLA by combining (i) a proprio-guided importance score that exploits the embodied state, and (ii) adaptive

pruning-layer selection that responds to instability in shallow-layer token rankings. The framework operates entirely at inference time and modifies neither the backbone weights nor the input pipeline.

3.1 Problem Setup

We consider a pretrained VLA transformer that jointly represents visual observations, proprioceptive input, and language instructions as tokens. Let $\mathcal{V}$ and $\mathcal{P}$ denote the index sets of visual and proprioceptive tokens, and let $\mathcal{Q}$ denote the instruction tokens used to construct the importance score. We use L for the total sequence length, including any special or architecture-specific tokens. The method assumes that the backbone exposes intermediate attention maps, represents proprioception within the transformer, and allows visual tokens to be removed between layers. At each transformer layer ℓ, let $A^{(\ell)} \in \mathbb{R}^{L\times L}$ denote the head-averaged self-attention map for one input sequence, with $A^{(\ell)}_{ij}$ giving the attention weight from query token i to key token j.

3.2 Proprio-Guided Token Importance

The importance score combines language-to-proprioception and language-to-vision attention from the pretrained model. We first measure how strongly each instruction token attends to the proprioceptive input and then use this value to weight its attention to each visual token. Visual tokens receive higher scores when they are attended to by instruction tokens that also attend strongly to proprioception.

At transformer layer ℓ, we first measure the attention from each instruction token to the proprioceptive tokens:

$$s^{(\ell)}_q \;=\; \sum_{j\in\mathcal{P}} A^{(\ell)}_{q,\,j}, \quad q \in \mathcal{Q}. \tag{1}$$

We then normalise these values across the instruction tokens:

$$w^{(\ell)}_q \;=\; \frac{s^{(\ell)}_q}{\sum_{q'\in\mathcal{Q}} s^{(\ell)}_{q'}}. \tag{2}$$

We use these weights to aggregate language-to-vision attention and obtain an importance score for each visual token:

$$g^{(\ell)}_v \;=\; \sum_{q\in\mathcal{Q}} w^{(\ell)}_q A^{(\ell)}_{q,\,v}, \quad v \in \mathcal{V}. \tag{3}$$

At the selected pruning layer $\ell^\star$, only the top-K visual tokens ranked by $g^{(\ell^\star)}_v$ are passed to subsequent layers.

All scores are computed from attention maps already produced during inference. The procedure introduces no trainable parameters, auxiliary networks, or fine-tuning.

3.3 Adaptive Pruning-Layer Selection

Prior token-pruning methods have shown that visual tokens can be reduced after only the first few transformer layers while retaining strong performance (Chen et al., 2024; Yang et al., 2026). We therefore choose a shallow probe layer ℓ_{probe} within this region so that pruning there preserves most of the computational benefit. The visual-token ranking at such an early layer, however, may still change as the representations are refined. Figure 2 examines this behavior by comparing the top-K visual-token sets obtained from every pair of adjacent layers. The earliest transitions exhibit lower mean Jaccard similarity and greater across-episode variation, whereas most later transitions maintain higher Jaccard values. To avoid prematurely discarding tokens under an unstable early ranking, we prune at ℓ_{probe} only when the adjacent-layer Jaccard similarity is sufficiently high. When the similarity is low, we preserve all visual tokens and defer pruning to a later layer ℓ_{late} chosen from the high-Jaccard region.

Using the proprio-guided score in Eq. (3), we define the top-K visual-token index set at layer ℓ as

$$\mathcal{K}^{(\ell)} \;=\; \text{top-}K\big(\{g^{(\ell)}_v\}_{v\in\mathcal{V}}\big). \tag{4}$$

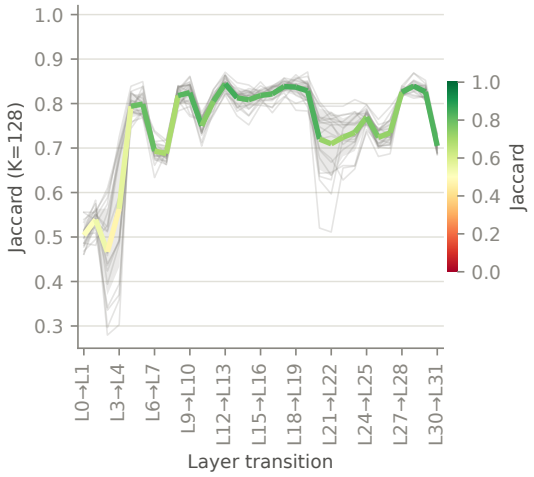


Figure 2: Adjacent-layer Jaccard similarity of the proprio-guided top-K visual-token sets.

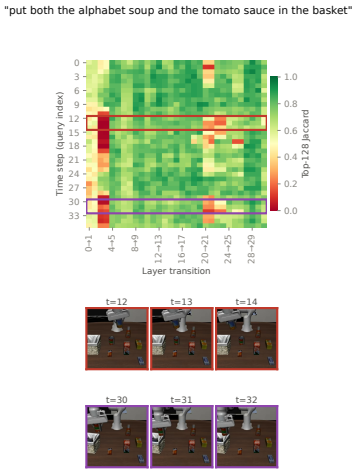


Figure 3: Within-episode Top-128 Jaccard similarity for a representative LIBERO-Long rollout. The highlighted intervals mark a fine-grained manipulation interval (red) and a transition to a new task stage (purple), both exhibiting low overlap in the shallow probe region.

At ℓ_{probe}, we measure the similarity of its selected token set to that of the preceding layer using

$$J^{\text{layer}} = \frac{\left|\mathcal{K}^{(\ell_{\text{probe}}-1)} \cap \mathcal{K}^{(\ell_{\text{probe}})}\right|}{\left|\mathcal{K}^{(\ell_{\text{probe}}-1)} \cup \mathcal{K}^{(\ell_{\text{probe}})}\right|}. \tag{5}$$

A low J^{layer} indicates that the visual-token ranking remains unstable across the adjacent layers. Pruning at ℓ_{probe} under this instability could irreversibly remove tokens that become important after further layer processing. We therefore use a threshold θ to determine whether to prune at ℓ_{probe} or defer pruning to ℓ_{late}:

$$\ell^{\star} = \begin{cases} \ell_{\text{late}}, & J^{\text{layer}} < \theta, \\ \ell_{\text{probe}}, & \text{otherwise}, \end{cases} \tag{6}$$

When $J^{\text{layer}} \geq \theta$, we prune at ℓ_{probe} using $\mathcal{K}^{(\ell_{\text{probe}})}$. Otherwise, all visual tokens are retained through ℓ_{late}, where the scores are recomputed and pruning uses $\mathcal{K}^{(\ell_{\text{late}})}$. The two sets used to compute J^{layer} are produced while reaching ℓ_{probe}, so the gate requires no additional forward pass.

Figure 3 provides an interpretable view of when the adaptive layer selector defers pruning. Low Jaccard values at the probe transition occur at semantically meaningful points rather than uniformly throughout the rollout. The highlighted intervals correspond to a fine-grained manipulation interval ($t = 12$–14) and a transition to a new task stage ($t = 30$–32), where the policy must adjust its visual focus to the current interaction or subgoal. At these moments, the low Jaccard similarity between adjacent-layer top-K sets suggests that visual-token relevance is still being refined across layers, making an early commitment to the shallow ranking less reliable. The adaptive layer selector therefore preserves the full visual sequence until ℓ_{late}, allowing the representations to mature before pruning. This behavior supports adaptive layer selection and makes deferral interpretable in terms of changing control demands throughout the rollout. How often pruning is deferred for each suite and retention budget is reported in Appendix A.

Algorithm 1 summarises the complete inference procedure. The pruning decision is made independently for each policy inference and maintains no state across inferences.

Algorithm 1 Inference with PROPPRUNE

Input: Token sequence and index sets $\mathcal{V}, \mathcal{P}, \mathcal{Q}$
Parameters: K, θ, ℓ_{probe}, ℓ_{late}
Output: Action prediction

1: Run to ℓ_{probe} and compute $\mathcal{K}^{(\ell_{\text{probe}}-1)}$ and $\mathcal{K}^{(\ell_{\text{probe}})}$.
2: Compute J^{layer} via Eq. (5).
3: **if** $J^{\text{layer}} < \theta$ **then**
4: Run to ℓ_{late}; set $\mathcal{K}^{\star} \leftarrow \mathcal{K}^{(\ell_{\text{late}})}$.
5: **else**
6: $\mathcal{K}^{\star} \leftarrow \mathcal{K}^{(\ell_{\text{probe}})}$.
7: **end if**
8: Retain $\mathcal{K}^{\star}$, run the remaining layers, and return the action.

4 EXPERIMENTAL SETUP

4.1 BENCHMARKS AND METRICS

We evaluate on the four LIBERO suites (Liu et al., 2023): Spatial, Object, Goal, and Long. Each suite contains 10 tasks that respectively emphasize spatial relations, object variation, goal variation, and long-horizon composition. Following the standard protocol, we run 50 rollouts per task with varied initial conditions and report success rate (SR).

We further evaluate robustness on LIBERO-Plus (Fei et al., 2025), which applies controlled shifts in camera viewpoint, robot initial state, language, lighting, background, sensor noise, and object layout. We run one rollout for each of its 10,030 perturbed instances and report category-wise and overall SR. All methods use the publicly released OpenVLA-OFT+ checkpoint without further adaptation.

For both simulated and real-world evaluations, we report latency and FLOPs. Latency is the mean wall-clock time per policy inference with batch size 1, averaged over rollout steps. FLOPs are computed from the sequence length processed at each transformer layer and averaged over the same inferences, thereby accounting for input-dependent deferral to ℓ_{late}.

4.2 BASELINES

All methods use the 7B OpenVLA-OFT architecture (Kim et al., 2025), which augments OpenVLA (Kim et al., 2024) with continuous actions, action chunking, and parallel decoding. The model processes a third-person image and a wrist-camera image, yielding 512 visual tokens, together with a proprioceptive token and the instruction tokens. It predicts an eight-step chunk of seven-dimensional actions in parallel. We use the standard OpenVLA-OFT checkpoints for LIBERO and the public OpenVLA-OFT+ checkpoint for LIBERO-Plus.

We compare PROPPRUNE with three training-free visual-token pruning baselines and one caching-based VLA acceleration method:

- **VLA-Cache** (Xu et al., 2026) identifies visual tokens that change little between consecutive observations and reuses their cached representations.
- **FastV** (Chen et al., 2024) ranks visual tokens using shallow-layer attention and retains the top-K tokens.
- **DivPrune** (Alvar et al., 2025) selects a subset that maximises feature diversity among the retained visual tokens.
- **EfficientVLA** (Yang et al., 2026) selects task-relevant visual tokens while promoting feature diversity. We use its visual-token selection component without its language-layer pruning or diffusion-head caching components.

All token selectors are implemented in the same backbone and evaluated on identical episodes under the same inference configuration. We evaluate two retention budgets fixed throughout each rollout:

Table 1: Accuracy–efficiency comparison of the original OpenVLA-OFT model and the evaluated VLA acceleration methods on the four LIBERO suites. Latency and FLOPs are averaged across suites. At each visual-token retention budget, bold marks the best success rate among pruning methods.

	Success rate (%)					Efficiency	
Method	Spatial	Object	Goal	Long	Avg.	Latency (ms)	FLOPs (T)
Original Model							
OpenVLA-OFT	99.2	98.2	97.2	95.6	97.6	77.32	4.013
Caching Method							
VLA-Cache	99.0	97.8	97.6	93.0	96.9	70.42	3.179
50% Visual-Token Retention ($K = 256$)							
FastV	98.2	**98.4**	97.0	87.2	95.2	60.21	2.507
DivPrune	96.8	78.2	93.6	64.6	83.3	58.59	2.290
EfficientVLA	96.6	74.6	92.2	58.0	80.4	63.53	2.453
PROPPRUNE	**98.8**	97.8	**97.2**	**93.8**	**96.9**	64.92	2.483
25% Visual-Token Retention ($K = 128$)							
FastV	98.2	**98.4**	95.8	82.4	93.7	46.43	1.765
DivPrune	81.2	58.6	83.6	36.0	64.9	45.16	1.444
EfficientVLA	87.0	29.8	79.6	7.8	51.1	49.77	1.684
PROPPRUNE	**98.6**	97.0	**96.4**	**93.2**	**96.3**	50.31	1.864

$K = 256$ (50%) and $K = 128$ (25%). The original model with all 512 visual tokens serves as the reference.

4.3 METHOD CONFIGURATION

We set $\ell_{\text{probe}} = 3$, $\ell_{\text{late}} = 6$, and $\theta = 0.35$. At each policy inference, the gate compares the top-K sets from layers 2 and 3. Pruning occurs at layer 3 when their Jaccard similarity is at least θ and is otherwise deferred to layer 6. We use the same configuration across all task suites and retention budgets.

4.4 REAL-WORLD SETUP

We fine-tune OpenVLA-OFT on demonstrations collected for two language-conditioned tabletop tasks. We then evaluate the resulting checkpoint with all 512 visual tokens as the full-token reference and with PROPPRUNE at 50% and 25% retention. All configurations use the same fine-tuned weights and inference setup. Both tasks use an external camera for the global workspace and a wrist-mounted camera for local object–gripper interactions, matching the model's dual-view input. We conduct 20 trials per task for each configuration. The tasks involve different target objects and manipulation sequences; representative observations are shown in Figure 4. Further details of the physical setup are given in Appendix D.

5 RESULTS

5.1 MAIN RESULTS

Table 1 summarizes the trade-off between task success and inference efficiency. At 50% retention, PROPPRUNE achieves an average SR of 96.9%, trailing the original model by only 0.7 points and outperforming the strongest pruning baseline, FastV, by 1.7 points. At 25% retention, it achieves 96.3% SR, exceeding FastV by 2.6 points while remaining within 1.3 points of the original model. The gains are largest on LIBERO-Long, where PROPPRUNE outperforms FastV by 6.6 and 10.8 points at 50% and 25% retention, respectively, demonstrating the effectiveness of adaptive layer selection on long-horizon tasks (Table 3).

DivPrune and EfficientVLA achieve lower computational costs, but their average SRs fall to 83.3% and 80.4% at 50% retention and to 64.9% and 51.1% at 25% retention, with the largest degrada-

Table 2: Robustness comparison on LIBERO-Plus across seven perturbation dimensions. All methods use the publicly released OpenVLA-OFT+ checkpoint. Overall denotes success rate across all perturbed tasks rather than the unweighted mean of category scores. Bold marks the best success rate among pruning methods in each category.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
OpenVLA-OFT+ (full tokens)	96.05	28.25	87.23	95.90	97.28	93.45	76.90	82.15
FastV	93.83	26.13	83.03	95.05	**97.05**	93.93	**75.18**	80.60
PROPPRUNE	**94.80**	**27.38**	**85.70**	**96.15**	96.75	**95.83**	**76.65**	**81.89**

Table 3: Sensitivity to the adjacent-layer Jaccard threshold θ at 50% visual-token retention. $\theta = 0$ always prunes at ℓ_{probe}, and $\theta \approx 1$ always defers pruning to ℓ_{late}. FLOPs are averaged over the four LIBERO suites. Bold marks the threshold used in the main experiments.

θ	Spatial	Object	Goal	Long	Avg.	FLOPs (T)
0 (always ℓ_{probe})	98.4	83.8	97.4	83.6	90.8	2.477
0.20	98.6	91.8	97.4	92.2	95.0	2.479
0.35	**98.8**	**97.8**	**97.2**	**93.8**	**96.9**	**2.483**
0.50	98.8	97.8	98.2	92.6	96.9	2.492
$\approx$1 (always ℓ_{late})	99.4	97.8	98.6	92.8	97.1	2.582

tion on LIBERO-Object and LIBERO-Long. VLA-Cache preserves a 96.9% average SR, matching PROPPRUNE at 50% retention, but reduces latency and FLOPs by only 8.9% and 20.8%, compared with 16.0% and 38.1% for PROPPRUNE. At 25% retention, PROPPRUNE increases these reductions to 34.9% and 53.6% while maintaining 96.3% SR. FastV is modestly faster because it always prunes at a fixed shallow layer, whereas adaptive deferral provides substantially higher task success, particularly on LIBERO-Long.

5.2 ROBUSTNESS ON LIBERO-PLUS

Table 2 extends the comparison beyond the in-distribution LIBERO suites. Whereas the original benchmark primarily measures task completion under familiar rendering and embodiment conditions, LIBERO-Plus tests performance under controlled distribution shifts. We report the full-token OpenVLA-OFT+ reference and the two pruning methods evaluated in this setting. The camera and robot columns capture geometric and embodiment changes; language measures instruction variation; light, background, and noise cover appearance and sensing changes; and layout captures changes in object placement. The overall score is computed over all benchmark tasks rather than as an unweighted mean of the seven category scores.

PROPPRUNE achieves an overall success rate of 81.89%, improving over FastV by 1.29 points and remaining 0.26 points below the full-token reference. It outperforms FastV in six of the seven perturbation categories, with the largest gains on language, noise, and layout shifts at 2.67, 1.90, and 1.47 points, respectively.

5.3 ABLATION STUDY

Table 3 varies the adjacent-layer stability threshold while fixing visual-token retention at 50%. Always pruning at the probe layer ($\theta = 0$) reduces the average SR to 90.8%, whereas always deferring to ℓ_{late} ($\theta \approx 1$) reaches 97.1% but requires 2.582 T FLOPs. With $\theta = 0.35$, the gate achieves 96.9% at 2.483 T FLOPs, close to the cost of always pruning early, so we use it in all experiments. Appendix B further isolates the contribution of proprioceptive guidance.

5.4 REAL-WORLD RESULTS

Figure 4 illustrates the two physical manipulation tasks, pairing an external view of the workspace with wrist-camera observations of local object–gripper interactions. As shown in Table 4, PROPPRUNE at 50% retention matches the full-token reference from the same fine-tuned OpenVLA-OFT

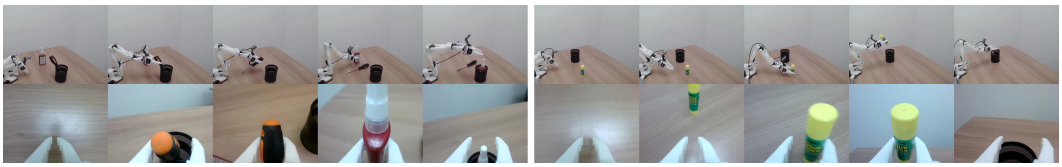


Figure 4: Representative observations from the two real-world manipulation tasks used to fine-tune and evaluate OpenVLA-OFT. Each column corresponds to one task; the top sequence shows the external view and the bottom sequence shows the corresponding wrist-camera observations.

Table 4: Real-world success and efficiency on two tabletop manipulation tasks. All configurations use the same fine-tuned OpenVLA-OFT checkpoint. Task columns report successful trials out of 20, and Overall SR is computed across both tasks. Latency and FLOPs are averaged across policy inferences.

Setting	Ret.	Task 1	Task 2	SR	Lat. (ms)	FLOPs (T)
Full tokens	100%	18/20	20/20	95.0%	57.60	3.9664
PROPPRUNE	50%	18/20	20/20	95.0%	46.62	2.5144
PROPPRUNE	25%	17/20	20/20	92.5%	40.05	1.9597

checkpoint with 38 successful trials out of 40, while reducing latency by 19.1% and FLOPs by 36.6%. At 25% retention, it succeeds in 37 of 40 trials and reduces latency and FLOPs by 30.5% and 50.6%, respectively.

6 CONCLUSION

We presented PROPPRUNE, a training-free framework that accelerates Vision-Language-Action inference by adapting both visual token selection and the layer at which pruning is applied. Its proprio-guided importance score conditions visual relevance on the robot's current state, while its adaptive layer selector defers pruning when the shallow token ranking remains unstable. Because both decisions are derived from attention maps already computed by the pretrained VLA, PROPPRUNE requires neither fine-tuning nor an auxiliary network. Across the original LIBERO suites, it provides the strongest accuracy–efficiency trade-off among the evaluated pruning methods, preserving success at aggressive retention budgets and yielding its largest gains on long-horizon tasks. Results on LIBERO-Plus and two physical manipulation tasks further show that these pruning decisions remain reliable under controlled distribution shifts and real-world execution. Together, these findings indicate that embodied context and layer-wise ranking stability provide effective internal signals for efficient VLA inference.

Limitations. Our experiments focus on the 7B OpenVLA-OFT model family, whose input sequence explicitly separates visual, language, and proprioceptive tokens. The extent to which the same attention signals and pruning policy transfer to other VLA architectures, fusion mechanisms, and action decoders remains to be established. Although we use one configuration across all evaluated task suites and retention budgets, the probe layer, deferred pruning layer, and stability threshold may still require architecture-specific selection. LIBERO-Plus provides broad coverage of controlled distribution shifts, but its perturbations are procedurally generated and use one rollout per perturbed instance; category-level averages therefore do not measure repeated-trial variance or reproduce the coupled perception, contact, and latency failures of physical systems. Finally, our real-world evaluation is limited to two tabletop tasks with 20 trials per task. Broader validation on diverse robots, longer-horizon tasks, and safety-critical or high-frequency control settings is needed to characterize the practical limits of the method.

AI USE STATEMENT

In this work, we used generative AI tools only to assist with translating the manuscript into English.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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*, pp. 19–35. Springer, 2024.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

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

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations*, volume 2024, pp. 26703–26721, 2024.

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

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867. IEEE, 2025.

Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. Vla-cache: Efficient vision-language-action manipulation via adaptive token caching. *Advances in Neural Information Processing Systems*, 38:164448–164473, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025.

Yantai Yang, Yuhao Wang, Zichen Wen, Luo Zhongwei, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. Efficientvla: Training-free acceleration and compression for vision-language-action models. *Advances in Neural Information Processing Systems*, 38:40891–40914, 2026.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

APPENDIX

A ADAPTIVE LAYER SELECTION ANALYSIS

We analyze how often pruning is deferred from layer 3 to layer 6 and how this decision varies across task suites and retention budgets. The late-deferral rate is the percentage of policy inferences for which the adjacent-layer Jaccard similarity falls below $\theta = 0.35$.

Table 5: Frequency of deferred pruning for each LIBERO suite and retention budget.

Suite	Late deferral (%)	
	$K = 256$	$K = 128$
Spatial	11.6	12.3
Object	14.0	24.0
Goal	7.6	10.5
Long	26.9	41.4
Overall	17.8	26.7

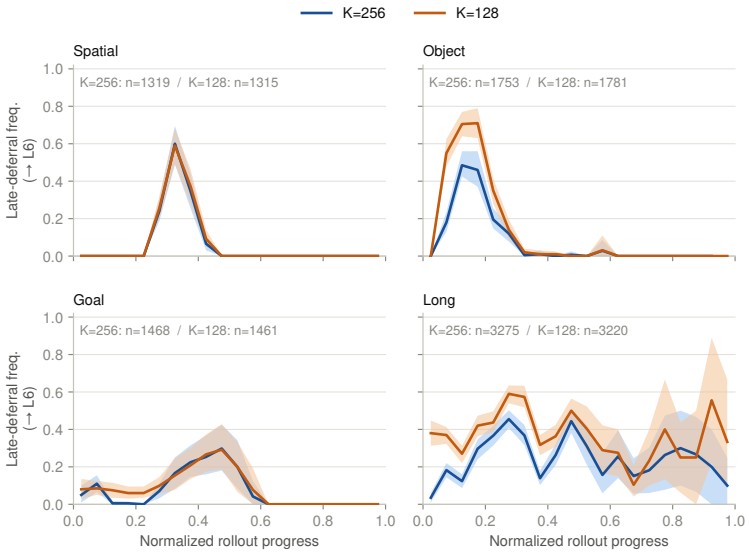


Figure 5: Late-deferral frequency over normalized rollout progress, reported separately for each LIBERO suite as the mean with a 95% bootstrap confidence interval. We avoid pooling suites because the same normalized rollout position can correspond to different task phases, which would average together misaligned suite-specific peaks.

Figure 5 shows that gate usage differs substantially across task structures. In the three non-Long suites, late deferral is concentrated in a transient window, peaking earliest for Object ($t \approx 0.15$), followed by Spatial ($t \approx 0.30$) and Goal ($t \approx 0.45$). These suite-dependent peaks are consistent with periods in which the shallow-layer token ranking remains unstable; outside the peak windows, the deferral rate falls close to zero. Long instead exhibits elevated deferral throughout much of the rollout, consistent with its long-horizon, multi-stage structure repeatedly requiring the policy to reassess relevant visual regions.

Across the four suites, $K = 128$ generally produces a higher deferral rate than $K = 256$ over normalized rollout progress (Table 5). This indicates that the gate falls back to the later layer more frequently under the tighter visual-token budget, adapting its computation to the increased instability of the smaller retained token set.

B VALIDATION OF PROPRIOCEPTIVE GUIDANCE

We isolate the contribution of proprioceptive guidance by comparing the complete method with a language-to-vision-only variant. Both methods use the same pretrained weights, retention budget, and adaptive layer-selection rule. The language-to-vision variant assigns equal weight to every instruction token, thereby removing proprioceptive weighting while preserving attention-based visual-token ranking.

Proprioceptive guidance improves the average success rate by 0.50 points at 50% retention and 0.25 points at 25% retention (Table 6). The improvement is concentrated on Long, where PROPPRUNE gains 1.8 and 1.2 points, respectively. Spatial slightly favors language-only scoring, Object is nearly unchanged, and Goal shows a small gain from proprioceptive guidance. The consistent Long improvement suggests that robot-state conditioning is most useful when relevant visual evidence changes repeatedly over a long-horizon task.

Table 6: Comparison of language-to-vision and proprio-guided scoring on LIBERO.

Scoring	Ret.	Spatial	Object	Goal	Long	Avg.
Language-only	50%	99.2	97.4	97.0	92.0	96.40
PROPPRUNE	50%	98.8	97.8	97.2	93.8	96.90
Language-only	25%	99.0	97.0	96.2	92.0	96.05
PROPPRUNE	25%	98.6	97.0	96.4	93.2	96.30

C DETAILED LIBERO-PLUS ROBUSTNESS RESULTS

We report suite-wise LIBERO-Plus success rates under seven perturbation dimensions in Table 7. The Avg. rows give the mean across the four suites for each dimension, while Overall denotes the aggregate benchmark score.

Table 7: Suite-wise success rate (%) on LIBERO-Plus across seven perturbation dimensions. Overall is reported for the aggregate result of each method.

Method	Suite	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
OpenVLA-OFT+	Spatial	98.40	30.00	93.80	98.60	97.70	85.80	96.40	
	Object	98.00	23.60	100.00	97.30	98.00	97.90	78.20	
	Goal	99.00	20.50	59.00	95.00	97.20	96.30	55.30	
	Long	88.80	38.90	96.10	92.70	96.20	93.80	77.60	
	Avg.	96.05	28.25	87.23	95.90	97.28	93.45	76.90	82.15
FastV	Spatial	97.90	24.90	91.30	97.30	100.00	90.00	96.40	
	Object	99.80	23.90	98.60	99.30	98.00	99.30	75.70	
	Goal	93.10	18.80	57.10	96.40	96.40	95.80	55.50	
	Long	84.50	36.90	85.10	87.20	93.80	90.60	73.10	
	Avg.	93.83	26.13	83.03	95.05	97.05	93.93	75.18	80.60
PROPPRUNE	Spatial	97.60	30.00	94.10	100.00	100.00	93.20	96.10	
	Object	98.70	23.60	99.70	98.70	99.20	99.10	76.90	
	Goal	92.70	19.30	57.10	93.90	94.00	95.20	56.70	
	Long	90.20	36.60	91.90	92.00	93.80	95.80	76.90	
	Avg.	94.80	27.38	85.70	96.15	96.75	95.83	76.65	81.89

Among the evaluated token-pruning methods, PROPPRUNE achieves the highest Overall score of 81.89, outperforming FastV by 1.29 points while remaining within 0.26 points of the full-token OpenVLA-OFT+ reference. It also exceeds FastV in six of the seven perturbation dimensions: camera, robot, language, lighting, sensor noise, and layout.

At the suite level, PROPPRUNE achieves higher seven-perturbation averages than FastV on Spatial, Object, and Long by 1.89, 0.19, and 3.71 points, respectively, with the largest margin appearing on Long.

D REAL-WORLD SETUP AND TASK-SPECIFIC FINE-TUNING

The physical evaluation uses OpenVLA-OFT as the base policy. We collect 50 demonstrations separately for each of two language-conditioned tabletop tasks and fine-tune a separate task-specific policy using LoRA. Each policy is trained for 80,000 optimization steps with a learning rate of 5×10^{-4}, a batch size of 8, and a LoRA scaling parameter of $\alpha = 32$. This adaptation is performed only to learn the physical task and is independent of visual-token pruning. Within each task, the corresponding fine-tuned weights are held fixed when evaluating the full-token reference and PROP-PRUNE at 50% and 25% visual-token retention. Consequently, the comparisons isolate the effect of changing the inference-time token budget.

Task 1 uses the instruction "Put the glue stick in the pen holder." Task 2 requires a longer manipulation sequence and uses the instruction "Remove the screwdriver from the pen holder, then place the super glue in the pen holder." The experiments are performed with an SO-101 robot arm. Each policy inference receives one image from an external camera observing the global workspace and one image from a wrist-mounted camera capturing local object–gripper interactions. This dual-view observation produces the 512 visual tokens used by OpenVLA-OFT. Policy inference is executed on an NVIDIA GeForce RTX 5090 GPU. For a given task, the full-token and pruned configurations use the same task-specific policy, observation interface, and evaluation procedure.

The evaluation contains 20 trials per task for each of three inference configurations: full tokens, 50% retention, and 25% retention. This yields 40 trials per configuration and 120 physical trials in total. Table 8 summarizes the confirmed training and evaluation setup.

Table 8: Task-specific training and physical evaluation setup.

Item	Confirmed setup
Base policy	OpenVLA-OFT
Robot	SO-101 robot arm
Task 1	Put the glue stick in the pen holder.
Task 2	Remove the screwdriver from the pen holder, then place the super glue in the pen holder.
Task-specific training	Separate LoRA fine-tuning for each task
Training data	50 demonstrations per task
Optimization	80,000 steps; learning rate 5×10^{-4}; batch size 8
LoRA configuration	Scaling parameter $\alpha = 32$
Visual observations	External and wrist-camera images
Inference hardware	NVIDIA GeForce RTX 5090 GPU
Compared settings	Full tokens, 50% retention, and 25% retention
Pruning-specific training	None
Model weights	Identical across retention settings within each task
Trials	20 per task and setting; 120 trials in total