# POLICY-FUNCTIONAL TOKEN PRUNING FOR EFFICIENT VISION-LANGUAGE-ACTION MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Visual token pruning can accelerate vision-language-action (VLA) inference, but accurate importance rankings do not necessarily yield an effective retained set. Our token-deletion diagnostics show that attention closely tracks how much removing a token changes an intermediate action-related representation, yet different influential tokens can change it in similar directions. In light of these observations, we introduce Policy-Functional Token Pruning (PFT-PRUNE), a training-free method that considers both individual influence and pairwise directional alignment. Specifically, at an early attention layer, we characterize each visual token by the local sensitivity of an action-related representation to its key, and use shared random projections to construct a compact policy-functional fingerprint (PFF). PFF energy provides the scalar priority, while fingerprint alignment identifies potential functional redundancy; Functional NMS uses both to select tokens under an exact budget before physical pruning. Across four LIBERO suites, PFT-PRUNE achieves 94.1% and 93.2% average success on $\pi_{0.5}$ and OpenVLA-OFT at 20% nominal visual-token retention, with inference speedups of $1.23\times$ and $1.59\times$, respectively. On four physical-robot tasks, it achieves a $1.22\times$ speedup at 50% nominal retention while preserving 95.9% of $\pi_{0.5}$'s average success rate.

## 1 INTRODUCTION

Vision-language-action (VLA) policies map camera observations, language instructions, and robot state to actions or short action chunks (Zitkovich et al., 2023; Kim et al., 2025b; Black et al., 2025b;a). Each inference step processes hundreds of visual tokens alongside nonvisual context through successive transformer layers. This repeated computation contributes substantial latency during closed-loop execution. Pruning visual tokens before later layers reduces the cost, but must preserve the objects, contact regions, and spatial relations needed to predict the next action.

Existing token pruning methods commonly select or merge tokens according to individual importance or feature similarity (Rao et al., 2021; Liang et al., 2022; Bolya et al., 2023; Chen et al., 2024; Wen et al., 2025; Zhang et al., 2025). Importance scores prioritize salient tokens, while feature similarity identifies redundancy between token representations. However, these criteria do not directly compare how removing different tokens changes an intermediate representation used for action prediction. Removing two tokens separately from the same attention interaction may produce similar changes in its action-related update, even when their features differ. Under a limited budget, selection should therefore consider not only individual importance, but also the similarity between these removal-induced changes.

Our analysis in Figure 1 supports this concern. Attention is a good indicator of how much removing a token changes an intermediate representation used for action prediction. Yet removing different influential tokens individually often changes that representation in similar ways. Thus, even an accurate importance ranking can overlook similarities between these changes. This motivates considering not only which tokens matter individually, but also whether they influence the representation in different ways.

Building on these observations, we introduce Policy-Functional Token Pruning (PFT-PRUNE), a training-free method for online visual-token selection. We use gradients at an early attention interaction to characterize how each visual token influences an intermediate representation used for action prediction. Shared random projections summarize this information as a policy-functional fingerprint



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.

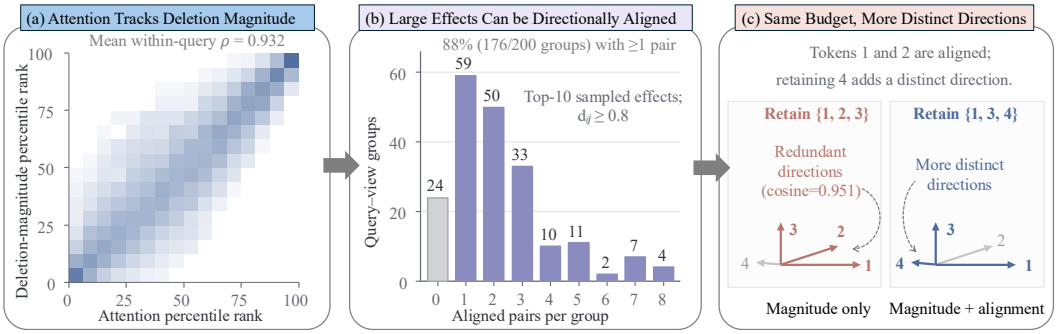



Figure 1: **Strong scalar ranking does not eliminate directional redundancy.** (a) Attention closely tracks exact deletion magnitude (mean within-query Spearman $\rho_S = 0.932$). (b) Among the ten largest sampled deletion effects, 176/200 query–view groups (88%) contain an aligned pair ($d_{ij} \geq 0.8$). (c) Illustration: at the same budget, accounting for alignment replaces a redundant high-magnitude token with one that preserves a distinct effect direction.

(PFF) for each token. These fingerprints guide token ranking and allow Functional NMS (Neubeck & Van Gool, 2006) to suppress candidates whose fingerprints are highly similar to those of higher-priority retained tokens within the same camera view. After filling any remaining budget, we physically remove unselected visual tokens and continue inference on the shortened sequence, preserving all nonvisual context.

We evaluate PFT-PRUNE on $\pi_{0.5}$ (Black et al., 2025a) and OpenVLA-OFT (Kim et al., 2025a) across four LIBERO suites. At 20% nominal visual-token retention, it achieves 94.1% and 93.2% average success, with inference speedups of $1.23\times$ and $1.59\times$ over the corresponding unpruned policies. On four physical-robot tasks with $\pi_{0.5}$, 50% retention yields 70.8% average success versus 73.8% without pruning, alongside a $1.22\times$ speedup. Ablations further support comparing token fingerprints rather than relying on individual importance rankings alone.

## 2 MOTIVATING OBSERVATIONS

Scalar importance scores are widely used to rank visual tokens for pruning. However, ranking tokens individually does not reveal whether different important tokens influence the policy computation in similar ways. So we ask two questions: *1. how much does removing each visual token change an action-related representation, and 2. do different influential tokens change it in similar directions?*

**Diagnostic Setting.** We analyze 100 fixed OpenVLA-OFT policy queries, with 25 from each of the four LIBERO suites (Kim et al., 2025a; Liu et al., 2023). Each query contains 512 visual tokens from the main and wrist cameras. Since exact token deletion requires a separate local recomputation for each tested token, we evaluate a fixed subset of 64 visual tokens for each query and reuse the same subset across all diagnostic comparisons. Appendix B.3 provides the sampling details.

**Action-Related Representation.** Consider an early attention operation with action tokens $\mathcal{A}$, visual tokens $\mathcal{V}$, and protected nonvisual tokens $\mathcal{P}$. For each action token $a \in \mathcal{A}$, let $q_a$ be its query, and let $k_j$ and $v_j$ be the key and value of token $j \in \mathcal{V} \cup \mathcal{P}$. Its attention output is $h_a = \sum_{j \in \mathcal{V} \cup \mathcal{P}} \alpha_{aj} v_j$, where $\alpha_{aj} = \mathrm{softmax}_j(q_a^\top k_j / \sqrt{d_k})$. We average these outputs across the action tokens to obtain:

$$F(K; x) := \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} h_a(K; x) \in \mathbb{R}^{d_F}, \tag{1}$$

where $K$ collects the visual-token keys and $x$ denotes the remaining inputs held fixed in this local computation. We use $F$ as the representation that we measure when visual tokens are removed.

**Exact Token Deletion.** To measure the effect of visual token $i$, we remove its key $k_i$ and value $v_i$ from the attention operation. We renormalize the attention weights over the remaining tokens and recompute $F$, while keeping all other inputs fixed. Let $F^{(-i)}$ denote the resulting representation. We define:

$$\Delta_i := F^{(-i)} - F, \qquad m_i := \|\Delta_i\|_2, \qquad d_{ij} := \frac{\Delta_i^\top \Delta_j}{\|\Delta_i\|_2 \|\Delta_j\|_2}. \tag{2}$$

Here, $\Delta_i$ is the change in $F$ caused by removing token $i$. Its magnitude $m_i$ measures the size of this change, while $d_{ij}$ measures the directional similarity between the changes caused by removing tokens $i$ and $j$. These exact deletions are used only for analysis.

**Observation 1: Attention Tracks Deletion Magnitude.** For each visual token, we average its attention weights across action tokens and attention heads to obtain a scalar attention score. For each query, we compute the Spearman correlation between this score and the deletion magnitude $m_i$ over the 64 sampled tokens. The mean correlation across the 100 queries is $0.932$ (Figure 1). Thus, attention already provides a strong indication of how much removing a token changes $F$.

**Observation 2: Large Deletion Effects Can Be Directionally Aligned.** A large deletion magnitude does not tell us whether another token causes a similar change in $F$. We therefore examine the main and wrist views of each query separately, giving 200 query–view groups. Among the ten sampled tokens with the largest deletion magnitudes in each group, 176 groups (88%) contain at least one pair with $d_{ij} \geq 0.8$ (Figure 1). Thus, tokens that individually cause large changes in $F$ can still produce strongly aligned changes.

Together, these observations suggest that token selection should consider not only how strongly each token affects $F$, but also whether different tokens produce similar changes in $F$. This motivates a functional view of token importance that captures both magnitude and direction of token effect.

## 3 Policy-Functional Token Pruning

Section 2 characterizes each visual token by the change $\Delta_i = F^{(-i)} - F$ caused by removing it. This gives both the magnitude and direction of the token's effect, but repeating the deletion for every visual token is too expensive for online pruning.

Rather than explicitly removing each visual token, PFT-Prune uses the local sensitivity of the same representation $F$ as an online proxy. Specifically, we differentiate $F$ with respect to each visual-token key and summarize the resulting sensitivity as a policy-functional fingerprint (PFF). PFF energy provides a scalar measure of local sensitivity, while the cosine between two PFFs measures their directional alignment. Functional NMS then uses these quantities to select visual tokens under a fixed budget. Figure 2 summarizes the method.

### 3.1 From Token Deletion to Local Sensitivity

We prune visual tokens independently within each view while keeping nonvisual tokens unchanged. For view $v$, let $\mathcal{V}_v$ denote its $N_v$ visual tokens and let $m_v$ be the number to retain. We select:

$$\mathcal{S}_v \subseteq \mathcal{V}_v, \qquad |\mathcal{S}_v| = m_v. \tag{3}$$

We next relate the exact deletion effects in Section 2 to visual-key derivatives. We first consider the same attention operation in the single-query, single-head case. Dropping the action-token and head indices, let $q$ denote the query, $\alpha_j$ the attention weight on token $j$, $v_j$ its value, and $o = \sum_j \alpha_j v_j$ the resulting attention output. Let $k_j$ denote the corresponding key entering the attention dot product, after any positional transformation when applicable.

**Proposition 3.1** (Local Deletion Geometry). *For token $i$ with $\alpha_i < 1$ and $q \neq 0$, let $o^{(-i)}$ be the attention output after removing token $i$ and renormalizing the remaining attention weights. Then*

$$o^{(-i)} - o = \frac{\alpha_i}{1-\alpha_i}(o - v_i), \qquad \frac{\partial o}{\partial k_i} = \frac{\alpha_i}{\sqrt{d_k}}(v_i - o)q^\top. \tag{4}$$

*and therefore*

$$o^{(-i)} - o = -\frac{\sqrt{d_k}}{(1-\alpha_i)\|q\|_2^2}\left(\frac{\partial o}{\partial k_i}\right) q. \tag{5}$$

The proposition shows that the deletion effect depends not only on the attention weight $\alpha_i$, but also on the residual direction $v_i - o$. Moreover, for two tokens with nonzero deletion effects, their key derivatives preserve the pairwise directional relationship identified in Section 2:

$$\cos_F\left(\frac{\partial o}{\partial k_i}, \frac{\partial o}{\partial k_j}\right) = \cos\Big(o^{(-i)} - o,\, o^{(-j)} - o\Big), \tag{6}$$

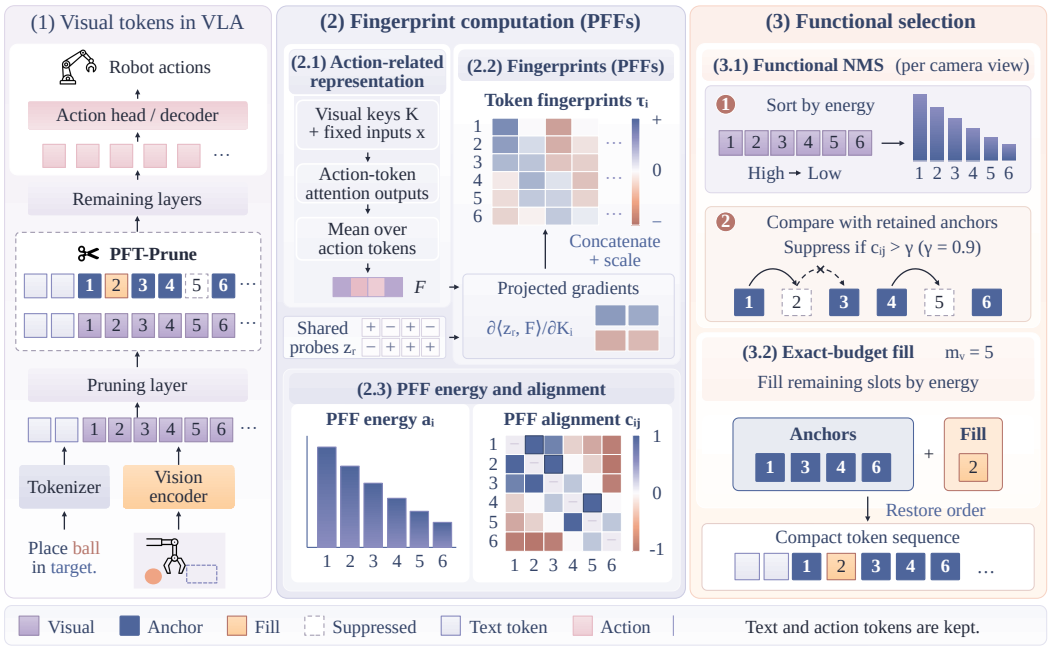



Figure 2: Overview of PFT-PRUNE. At an early attention layer, the action-related representation $F$ is computed from the attention outputs at the action tokens. Shared random probes compress each visual token's sensitivity to $F$ into a PFF. PFF energy ranks the visual tokens, while PFF alignment identifies tokens with similar effect directions. Functional NMS suppresses lower-priority aligned candidates, and exact-budget fill produces the final retained set. Suppressed candidates do not suppress later tokens.

where $\cos_F$ is cosine similarity between the two derivative matrices after flattening them. Thus, in the single-query, single-head case, comparing key derivatives gives exactly the same directional similarity as comparing the corresponding deletion effects. Appendix A.1 provides the derivation and discusses the scope of this result.

**From Exact Deletion to an Online Proxy.** Motivated by Proposition 3.1, we return to the multi-token, multi-head representation $F$ defined in Section 2. Rather than recomputing $F^{(-i)}$ for every visual token, PFT-PRUNE differentiates $F$ with respect to each visual-token key while keeping all other inputs and model parameters fixed. This provides a local proxy for the finite deletion effects without explicitly removing each token.

### 3.2 POLICY-FUNCTIONAL FINGERPRINTS

For the full representation $F$, let $K_i \in \mathbb{R}^{d_k}$ denote the visual key of token $i$ used for fingerprint construction. We characterize the local sensitivity of $F$ to this token by the key-side Jacobian:

$$J_i := \frac{\partial F(K; x)}{\partial K_i} \in \mathbb{R}^{d_F \times d_k}, \tag{7}$$

where $d_F$ is the dimension of $F$ and $d_k$ is the visual-key dimension. The Jacobian $J_i$ describes how small changes to token $i$'s key change the action-related representation $F$. Constructing and storing the full Jacobian for every visual token, however, is expensive. We therefore seek a compact representation that preserves the relative sensitivity and pairwise alignment of these Jacobians.

To construct this representation, we sample $R$ shared directions in the $d_F$-dimensional output space of $F$. For $r = 1, \ldots, R$, let:

$$z_r = \frac{1}{\sqrt{d_F}} \epsilon_r, \qquad \epsilon_r \in \{-1, +1\}^{d_F}, \tag{8}$$

where the entries of $\epsilon_r$ are independent Rademacher variables. The same $R$ probes are shared across all visual tokens. Each probe $z_r$ selects one direction in the output space of $F$. Rather than storing

the full Jacobian $J_i$, we measure how sensitive $F$ is to token $i$ along this direction:

$$g_i^{(r)} := \frac{\partial \langle z_r, F(K; x) \rangle}{\partial K_i} = J_i^\top z_r \in \mathbb{R}^{d_k}. \tag{9}$$

We concatenate the projected gradients from all $R$ shared directions to form the policy-functional fingerprint (PFF):

$$\tau_i := \frac{1}{\sqrt{R}} \operatorname{vec}\left(g_i^{(1)}, \ldots, g_i^{(R)}\right) \in \mathbb{R}^{Rd_k}. \tag{10}$$

Because all tokens use the same probes, their fingerprints are directly comparable. For each probe, a single reverse-mode vector–Jacobian product (Baydin et al., 2018) computes $g_i^{(r)}$ for all visual tokens jointly, without explicitly constructing the Jacobians $J_i$.

**Proposition 3.2** (Randomized Functional Sketch)**.** *For any visual tokens $i$ and $j$,*

$$\mathbb{E}\left[\tau_i^\top \tau_j\right] = \frac{1}{d_F} \langle J_i, J_j \rangle_F, \qquad \mathbb{E}\left[\|\tau_i\|_2^2\right] = \frac{1}{d_F} \|J_i\|_F^2. \tag{11}$$

Proposition 3.2 shows that PFF inner products estimate the Frobenius inner products between token Jacobians, while PFF energy estimates their squared Frobenius norms. Thus, the fingerprints preserve the quantities needed for token priority and pairwise comparison in expectation, using only $Rd_k$ values per token instead of $d_F d_k$. Appendix A.4 gives the proof and finite-probe analysis.

We therefore use PFF for token priority and normalized PFF similarity for pairwise alignment:

$$a_i := \|\tau_i\|_2^2, \qquad c_{ij} := \frac{\tau_i^\top \tau_j}{\|\tau_i\|_2 \|\tau_j\|_2 + \epsilon}, \tag{12}$$

where $\epsilon > 0$ avoids division by zero. The PFF energy $a_i$ measures the local sensitivity of $F$ to token $i$ and provides its scalar priority. The PFF alignment $c_{ij}$ measures whether tokens $i$ and $j$ induce similar local sensitivity directions. A large positive $c_{ij}$ indicates potential functional redundancy under a limited token budget.

### 3.3 Budgeted Functional Selection

We select tokens by combining PFF-energy priority with pairwise PFF alignment: high-energy tokens are considered first, while candidates strongly aligned with already retained tokens are suppressed. Selection is performed independently within each camera view.

For view $v$, we first sort its $N_v$ visual tokens by decreasing PFF energy:

$$(i_1, \ldots, i_{N_v}), \qquad a_{i_1} \geq \cdots \geq a_{i_{N_v}}. \tag{13}$$

We scan the tokens in this order, retaining $i_1$ as the first anchor. Each subsequent token is retained as a new anchor if its PFF alignment with every existing anchor is at most $\gamma$; otherwise, it is suppressed. Suppressed tokens do not suppress later candidates.

This procedure, which we call *Functional NMS*, favors high-energy tokens whose local effect directions are not strongly aligned with earlier retained tokens. Let $\mathcal{A}_v$ denote the anchors in their original energy order and $\mathcal{B}_v$ the suppressed tokens, also in energy order.

Functional NMS may produce more or fewer than the target budget $m_v$. If $|\mathcal{A}_v| \geq m_v$, we retain the first $m_v$ anchors. Otherwise, we retain all anchors and fill the remaining slots with the highest-energy tokens in $\mathcal{B}_v$. This exact-budget fill ensures $|\mathcal{S}_v| = m_v$.

**Proposition 3.3** (Budget and Anchor Separation)**.** *For any $0 \leq m_v \leq N_v$, the selection procedure returns $|\mathcal{S}_v| = m_v$. Moreover, any two distinct anchors $i, j \in \mathcal{A}_v$ satisfy $c_{ij} \leq \gamma$.*

The separation guarantee applies only to the anchors: exact-budget fill may reintroduce suppressed tokens when $|\mathcal{A}_v| < m_v$. Appendix A.5 provides the proof and matrix implementation.

After selection, we physically remove the unselected visual tokens and restore the retained visual and nonvisual tokens to their original sequence order. The remaining policy layers then operate on this shortened sequence. Appendix C provides the complete algorithm, physical gathering details, and complexity analysis.

Table 1: LIBERO success (%) and policy latency (ms). Within each policy and budget, bold marks the best value; underline marks the second-lowest distinct latency. Long denotes LIBERO-10.

| Method | $\pi_{0.5}$ | | | | | | OpenVLA-OFT | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Spatial | Object | Goal | Long | Avg.↑ | Lat.↓ | Spatial | Object | Goal | Long | Avg.↑ | Lat.↓ |
| *Retain 100% of visual tokens* | | | | | | | | | | | | |
| No pruning | 99.5 | 98.5 | 97.0 | 92.5 | 96.9 | 60.75 | 93.0 | 98.8 | 96.4 | 95.6 | 96.0 | 94.08 |
| *Retain 20% of visual tokens* | | | | | | | | | | | | |
| FastV | 80.0 | 82.0 | 67.0 | 49.0 | 69.5 | **47.23** | 94.0 | 89.2 | 96.0 | 89.2 | 92.1 | **56.76** |
| SparseVLM | 88.5 | 94.0 | 81.0 | 62.0 | 81.4 | 51.46 | **94.6** | 75.8 | **96.6** | 73.8 | 85.2 | 60.58 |
| DART | 61.0 | 15.0 | 34.5 | 19.0 | 32.4 | 49.52 | 82.8 | 73.0 | 93.2 | 63.8 | 78.2 | 59.12 |
| ADP | 32.5 | 62.5 | 40.5 | 31.5 | 41.8 | 51.75 | 88.8 | 81.6 | **96.6** | 88.4 | 88.9 | 70.35 |
| SpecPrune | 92.5 | 86.5 | 75.5 | 56.5 | 77.8 | 58.05 | 89.8 | **96.0** | 94.6 | 66.8 | 86.8 | 68.87 |
| PFT-PRUNE | **96.0** | **94.5** | **95.0** | **91.0** | **94.1** | 49.44 | **94.6** | 92.8 | 95.2 | **90.2** | **93.2** | 59.22 |
| *Retain 30% of visual tokens* | | | | | | | | | | | | |
| FastV | 96.0 | 94.5 | 93.0 | 80.0 | 90.9 | **50.98** | 92.4 | **96.0** | 96.4 | 90.0 | 93.7 | **60.22** |
| SparseVLM | 93.0 | 93.0 | 82.5 | 63.5 | 83.0 | 53.73 | **94.2** | 92.2 | 95.4 | 90.8 | 93.2 | 66.26 |
| DART | 91.5 | 87.5 | 90.0 | 71.5 | 85.1 | 54.78 | 87.6 | 84.6 | 96.2 | 74.2 | 85.7 | 64.07 |
| ADP | 89.0 | 85.0 | 74.0 | 77.5 | 81.4 | 53.23 | 88.8 | 81.6 | **96.6** | 88.4 | 88.9 | 70.35 |
| SpecPrune | 92.5 | 86.5 | 75.5 | 56.5 | 77.8 | 58.05 | 89.8 | **96.0** | 94.6 | 66.8 | 86.8 | 68.87 |
| PFT-PRUNE | **97.5** | **97.5** | **95.0** | **93.5** | **95.9** | 51.35 | **94.2** | 95.2 | 95.4 | **93.8** | **94.7** | 61.26 |
| *Retain 40% of visual tokens* | | | | | | | | | | | | |
| FastV | 97.0 | 96.5 | 91.5 | 85.5 | 92.6 | 53.50 | 93.8 | **97.8** | 97.2 | 90.8 | 94.9 | **70.33** |
| SparseVLM | 95.0 | 95.5 | 85.0 | 76.0 | 87.9 | 55.20 | 93.8 | 96.4 | 97.4 | **92.6** | **95.1** | 74.29 |
| DART | 93.5 | 92.5 | 86.0 | 69.0 | 85.3 | 56.78 | 90.8 | 93.4 | **97.6** | 85.6 | 91.9 | 75.31 |
| ADP | 95.0 | 86.0 | 83.5 | 85.5 | 87.5 | 54.24 | 91.6 | 93.2 | 96.6 | 90.6 | 93.0 | 72.27 |
| SpecPrune | 95.5 | 97.0 | 90.5 | 83.5 | 91.6 | 58.68 | 91.4 | **97.8** | 97.0 | 90.2 | 94.1 | 70.59 |
| PFT-PRUNE | **97.5** | **99.0** | **97.5** | **92.5** | **96.6** | **51.33** | **94.2** | 96.2 | 96.2 | **92.6** | 94.8 | 70.83 |
| *Retain 50% of visual tokens* | | | | | | | | | | | | |
| FastV | 97.0 | 97.0 | 95.5 | 88.0 | 94.4 | **54.10** | 94.2 | 97.6 | 97.2 | 92.6 | 95.4 | **72.23** |
| SparseVLM | 94.5 | 96.5 | 91.0 | 79.0 | 90.3 | 55.63 | **94.8** | 97.0 | **97.6** | 92.6 | **95.5** | 79.50 |
| DART | 92.5 | 91.0 | 89.0 | 77.5 | 87.5 | 57.64 | 91.2 | 92.6 | 96.8 | 81.0 | 90.4 | 75.64 |
| ADP | 96.0 | 92.0 | 90.5 | 88.5 | 91.8 | 55.50 | 93.2 | 96.2 | 97.2 | 91.6 | 94.6 | 78.09 |
| SpecPrune | 97.5 | 94.5 | 92.0 | 86.0 | 92.5 | 59.48 | 92.6 | **98.2** | 97.4 | 93.6 | **95.5** | 79.26 |
| PFT-PRUNE | **98.5** | **99.0** | **96.5** | **94.0** | **97.0** | 54.54 | 93.8 | 95.8 | 96.8 | **94.4** | 95.2 | 72.95 |

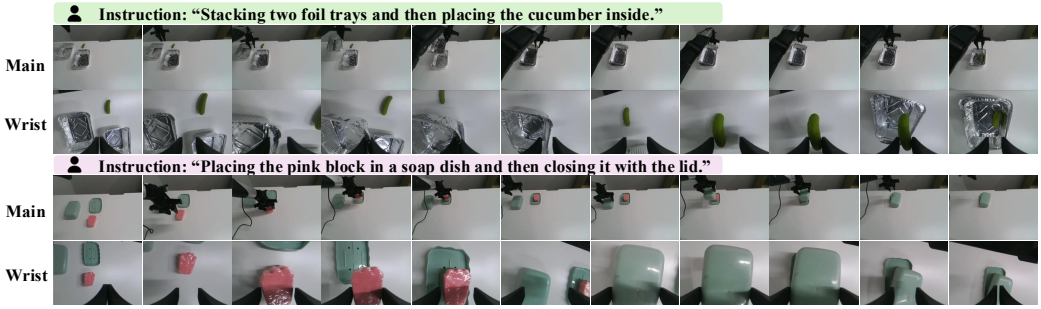



Figure 3: Representative physical-robot rollouts for the two long-horizon tasks.

## 4 EXPERIMENTS

**Experimental Setup.** We evaluate PFT-PRUNE on two VLA policies, $\pi_{0.5}$ and OpenVLA-OFT across LIBERO-Spatial, Object, Goal, and LIBERO-10, abbreviated as Long (Liu et al., 2023). For $\pi_{0.5}$, we run 20 episodes per task using the benchmark's fixed initial-state schedule; for OpenVLA-OFT, we use suite-specific checkpoints and 50 episodes per task. At 20%, 30%, 40%, and 50% visual-token retention, we compare FastV (Chen et al., 2024), SparseVLM (Zhang et al., 2025), DART (Wen et al., 2025), ADP (Pei et al., 2026), and SpecPrune (Wang et al., 2026). Appendix G details the baseline implementations and token accounting. We measure policy latency from observation, instruction, and robot-state inputs to the action or action-chunk output on a single 24 GB NVIDIA RTX 4090 GPU with batch size one. Reported PFT-PRUNE latency includes all fingerprint construction, selection, and physical-compaction costs, including the layer-1 proxy used by OpenVLA-OFT. Speedup is measured relative to the corresponding unpruned policy.

| Variant | Spatial | Object | Goal | Long | Avg.↑ | Lat.↓ |
|---|---|---|---|---|---|---|
| Full PFT-PRUNE | 94.6 | 92.8 | 95.2 | 90.2 | **93.2** | 59.22 |
| Native-key norm + cosine | 58.8 | 65.2 | 93.0 | 52.2 | 67.3 | 56.56 |
| Attention priority | 94.8 | 81.2 | 96.4 | 89.2 | 90.4 | 59.85 |
| PFF-Energy Top-K | 92.6 | 93.2 | 96.6 | 85.0 | 91.8 | 58.70 |
| PFF energy + native-key cosine | 91.4 | 93.2 | 96.4 | 83.2 | 91.1 | 59.32 |
| Value-side PFF | 92.0 | 78.4 | 94.8 | 87.0 | 88.0 | 58.85 |

Table 2: OpenVLA-OFT ablations at 20% token retention on LIBERO. Success and latency are reported in % and ms. Bold marks the best average success.

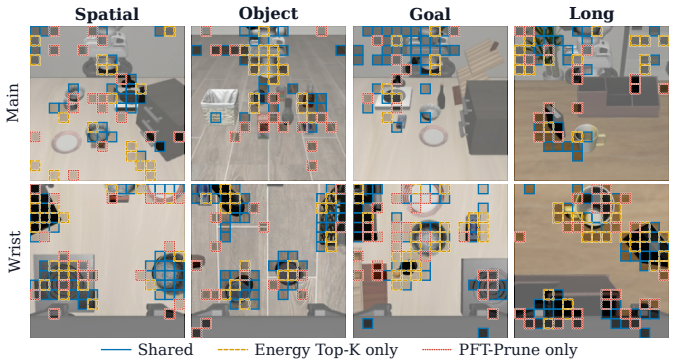



Figure 4: Functional alignment changes token selection. Representative OpenVLA-OFT queries from the four LIBERO suites at 20% retention compare PFF-Energy Top-K and PFT-PRUNE at fixed observation, PFF energy, and budget. Queries are selected independently of rollout outcome using median main/wrist selection overlap.

**Main Results on Simulation Experiments.** Table 1 shows that PFT-PRUNE attains the highest $\pi_{0.5}$ mean success at every tested budget. At 20% retention, it reaches 94.1% success with a 1.23× speedup. At 50%, it processes half the visual tokens with 97.0% success, close to the unpruned policy's 96.9%, and a 1.11× speedup. On OpenVLA-OFT, PFT-PRUNE leads average success at 20% and 30% retention. At 20%, it reaches 93.2% success with a 1.59× speedup.

**Real-World Experiments.** Physical evaluation uses an AgileX NERO 7-DoF arm and a task-specific $\pi_{0.5}$ policy with main- and wrist-camera inputs. Task 1 places a cucumber in a foil tray; Task 2 places a block in a target area. Task 3 stacks two trays before placing the cucumber inside; Task 4 places the block in a dish and closes its lid. We run 100 trials per task–method pair, with 50% retention for all pruning methods. Figure 3 shows representative long-horizon rollouts; Appendix E details the setup and protocol. Table 3 shows that PFT-PRUNE achieves the highest average success among the pruning methods at 70.8%, versus 73.8% without pruning. Its 124.84 ms policy latency corresponds to a 1.22× speedup over the unpruned policy.

Table 3: Physical-robot results at 50% retention over 100 trials per task–method pair. Success is in %; latency is in ms. Speedup is relative to unpruned $\pi_{0.5}$.

| Method | Task 1 | Task 2 | Task 3 | Task 4 | Avg.↑ | Lat.↓ | Speed.↑ |
|---|---|---|---|---|---|---|---|
| No pruning | 88.0 | 91.0 | 64.0 | 52.0 | 73.8 | 152.43 | 1.00 |
| FastV | 77.0 | 79.0 | 43.0 | 25.0 | 56.0 | **123.75** | **1.23** |
| ADP | 85.0 | **89.0** | 52.0 | 36.0 | 65.5 | 131.13 | 1.16 |
| SpecPrune | **89.0** | 87.0 | 56.0 | 44.0 | 69.0 | 132.59 | 1.15 |
| PFT-PRUNE | 86.0 | **89.0** | **61.0** | **47.0** | **70.8** | 124.84 | 1.22 |

## 5 ANALYSIS

**Ablation Study.** Table 2 tests PFF representation, priority, alignment, and differentiation interface on OpenVLA-OFT at 20% retention, fixing the protocol and per-view budgets ($\gamma = 0.90$ unless varied). Appendix D provides $\pi_{0.5}$ ablations and selection-example provenance. Native-key norm + cosine replaces both PFF signals, reducing mean success from 93.2% to 67.3%. Early-layer attention priority retains PFF cosine and achieves 90.4%, supporting attention as a useful ordering signal. PFF-Energy Top-K retains energy priority but omits Functional NMS. PFF energy + native-key cosine compares raw keys, not native-coordinate PFFs, fixing PFF-energy priority, threshold, retained-anchor rule, and exact-budget fill. Its success drop versus full PFT-PRUNE at similar latency gives $p = 1.5 \times 10^{-4}$ (exact two-sided McNemar test, matched episodes (McNemar, 1947)). Value-side PFFs achieve 88.0%, below key-side PFFs. Figure 4 compares selections across all four suites and both views at fixed PFF energy and budget.

**Functional Alignment is Most Active Under Tight Budgets.** We test whether exact-budget fill undoes Functional NMS (Section 3.3). Figure 5 analyzes 200 fixed $\pi_{0.5}$ queries, 50 per suite,

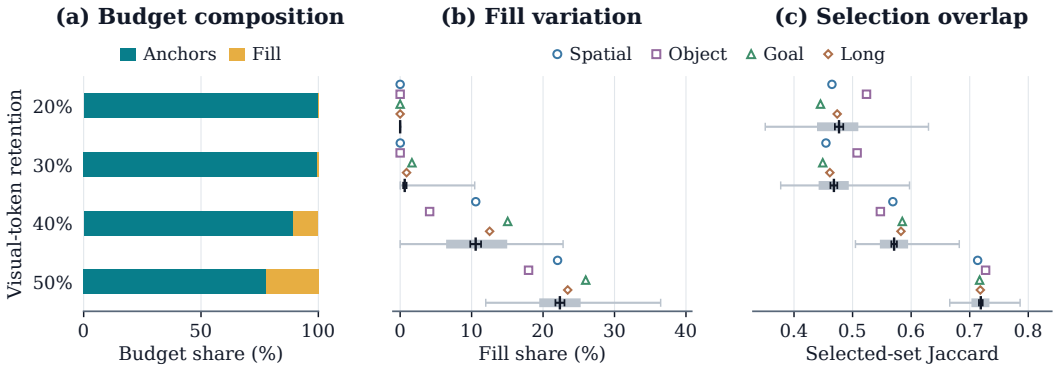



Figure 5: **Functional alignment is most active under tight budgets.** 200 fixed $\pi_{0.5}$ queries, 50 per suite. **(a)** Mean retained-anchor and fill shares. **(b,c)** Colored markers show suite means (vertically offset); gray whiskers and thick segments show full query ranges and middle 50%; black marks show aggregate means and 95% bootstrap confidence intervals. **(c)** Selected-set Jaccard between full PFT-PRUNE and PFF-Energy Top-K.

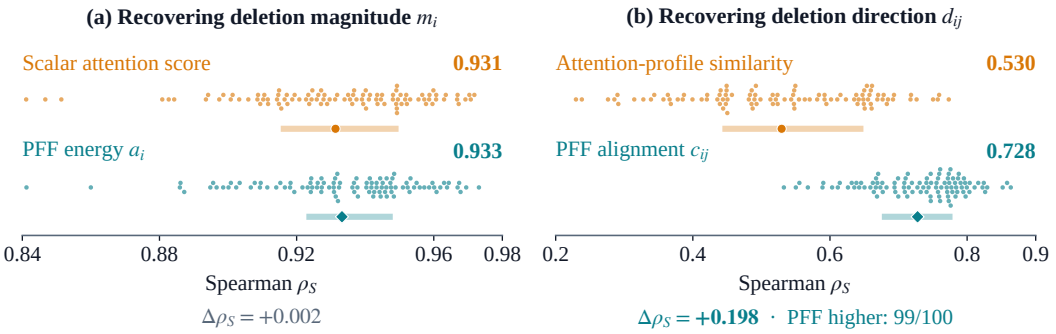



Figure 6: **PFFs recover deletion magnitude and better recover deletion direction.** On 100 OpenVLA-OFT queries with uniform token sampling, the scalar attention score and PFF energy show comparable agreement with $m_i$ (0.931 vs. 0.933). PFF alignment better matches $d_{ij}$ than attention-profile similarity (0.728 vs. 0.530; higher on 99/100 queries). Points show within-query Spearman correlations; markers show means and bars the middle 50%. Axes are zoomed.

holding fingerprints and energy order fixed between selectors. At 20% retention, no query requires fill: all retained tokens are anchors, preserving within-view separation. Mean selected-set Jaccard overlap with PFF-Energy Top-K is 0.477. At 50%, fill occupies 22.4% of the budget and overlap is 0.719. Thus, at larger budgets more suppressed candidates return and selection is closer to energy Top-K. Appendix C.5 details the protocol.

**Validating PFFs Against Exact Deletion Effects.** Having established when alignment changes selection, we test whether PFF energy $a_i$ and alignment $c_{ij}$ recover exact deletion magnitude $m_i$ and deletion-direction similarity $d_{ij}$. We reuse the 100 OpenVLA-OFT queries from Section 2, with 64 visual tokens per query sampled uniformly and independently of PFF energy.

**PFF Energy Recovers Deletion Magnitude.** The scalar attention score and PFF energy achieve mean within-query Spearman correlations of 0.931 and 0.933 with $m_i$, respectively (Figure 6a). Thus, PFF energy tracks deletion magnitude comparably to attention, its main benefit does not come from a stronger scalar ranking.

**PFF Alignment Recovers Deletion Direction.** Attention-profile similarity is the Pearson correlation between two tokens' action-query and head-wise attention profiles, whose averages give the scalar score (Section 2). Against $d_{ij}$ over sampled token pairs, its mean within-query Spearman correlation is 0.530, versus 0.728 for PFF alignment $c_{ij}$ (Figure 6b). PFF alignment is higher on 99/100 queries, with a mean gain of $+0.198$. Its additional information therefore concerns pairwise effect direction in the local action-related representation $F$, which scalar importance cannot express.

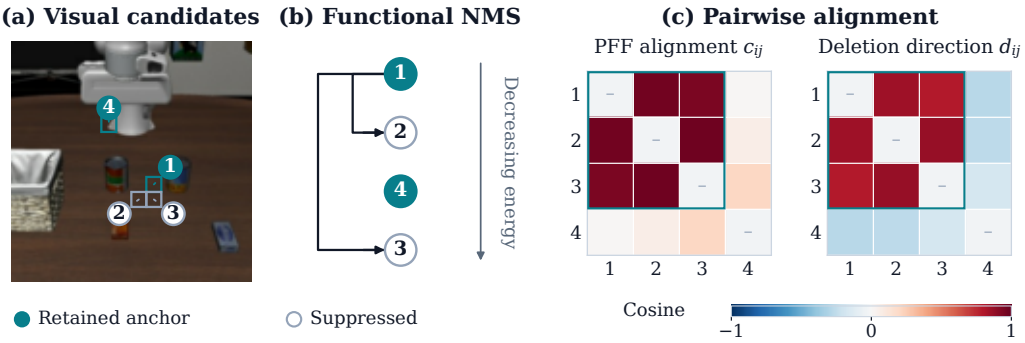



Figure 7: **Measured Functional NMS example.** Four token candidates from one camera view are shown. **(a)** IDs indicate their token locations. **(b)** Candidates are processed in decreasing PFF-energy order; anchor 1 suppresses candidates 2 and 3, while candidate 4 remains a retained anchor. **(c)** PFF alignment $c_{ij}$ and exact deletion-direction similarity $d_{ij}$ show consistent local grouping on the same cosine scale.

**A Measured Example of Functional NMS.** Figure 7 shows how the directional signal enters an actual selection decision. Candidates 2 and 3 are strongly aligned with the higher-energy anchor 1 and are therefore suppressed, while candidate 4 remains a distinct anchor. Their exact deletion-direction similarities show the same local grouping.

## 6 Related Work

**Vision-Language-Action Policies.** RT-2 and OpenVLA generate discrete action tokens (Zitkovich et al., 2023; Kim et al., 2025b), while $\pi_0$ and $\pi_{0.5}$ generate action chunks through flow matching (Black et al., 2025b;a; Lipman et al., 2023). OpenVLA-OFT adapts OpenVLA for efficient action-chunk prediction (Kim et al., 2025a). Inference optimizations also exploit reuse: VLA-Cache reuses computation for stable visual tokens (Xu et al., 2025), and DepthCache distributes depth-guided token merging across frames (Li et al., 2026b). EfficientVLA combines visual-token selection, language-layer pruning, and diffusion-head reuse (Yang et al., 2025).

**Visual-Token Reduction.** DynamicViT learns token selection, while EViT and Token Merging aggregate representations (Rao et al., 2021; Liang et al., 2022; Bolya et al., 2023). In vision-language models, FastV uses early-layer attention (Chen et al., 2024), SparseVLM uses language-conditioned relevance (Zhang et al., 2025), and DART removes tokens redundant with visual or textual pivots (Wen et al., 2025). ZOO-Prune combines zeroth-order sensitivity estimates with visual-feature diversity (Kim et al., 2026).

**VLA-Specific Token Selection.** Action relevance and interaction cues guide VLA pruning. ADP adapts retention across manipulation phases (Pei et al., 2026); SpecPrune-VLA coordinates action- and layer-level pruning (Wang et al., 2026). BFA++ learns token and camera-view importance (Li et al., 2026a); IAprune uses semantic–motion agreement and boundary coverage (Cheng et al., 2026). VLA-Pruner combines semantic and temporally smoothed action attention with visual-feature diversity (Liu et al., 2025), and SAFE-Pruner forecasts deep-layer saliency from historical attention (Ma et al., 2026). Action-JND learns token-wise feature-perturbation tolerances under an action-deviation constraint for pruning and cache reuse (Li et al., 2026c). PFT-Prune instead sketches each token's local effect on an action-related representation, using fingerprint energy and alignment to estimate influence and pairwise functional redundancy within each policy query.

## 7 Conclusion

PFT-Prune combines PFF energy and alignment for visual-token pruning using sketches of local effects on an intermediate action-related representation. Deletion diagnostics support directional information beyond scalar importance. Budget analysis shows that Functional NMS is active under tight budgets, while larger budgets rely more on fill. Evaluations on two VLA families and physical-robot tasks show reduced inference cost while preserving most of the unpruned success rate.

### AI USE STATEMENT

In this work, we employed generative AI tools to improve the clarity, fluency, and grammatical quality of the English writing, including the abstract, introduction, methodology, and experimental sections. They were not involved in aspects such as literature search or data analysis.

### ETHICS STATEMENT

The evaluations use simulated and physical robot manipulation; they do not study human participants or collect personal data. Token pruning may remove safety-relevant visual evidence under distribution shift, so benchmark performance and the reported physical-robot trials do not by themselves establish deployment safety. Deployment requires target-domain validation, recovery procedures, collision protection, and emergency-stop mechanisms.

### REPRODUCIBILITY STATEMENT

The method and evaluation sections specify the pruning rule and experimental protocol. The appendix provides theoretical derivations, architecture-specific interfaces and pruning boundaries, exact token budgets and selection details, diagnostic sampling, selector-complexity and hyperparameter analyses, and the physical-robot setup and trial protocol.

## REFERENCES

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.

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 *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a.

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. $\pi_0$: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010.

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.

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Khalil Meftah, Maxime Ellerbach, Jess Moss, and Thomas Wolf. LeRobot: State-of-the-art machine learning for real-world robotics in PyTorch, 2024.

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, 2024.

Jintao Cheng, Weibin Li, Haozhe Wang, Gang Wang, Yipu Zhang, Xiaoyu Tang, Jin Wu, Xieyuanli Chen, Yunhui Liu, and Wei Zhang. Training-free interaction-aligned visual token pruning for efficient embodied manipulation. *arXiv preprint arXiv:2603.22991*, 2026.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. doi: 10.1214/aos/1176344552.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics—Simulation and Computation*, 18(3):1059–1076, 1989. doi: 10.1080/03610918908812806.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.017.

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 *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025b.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. ZOO-Prune: Training-free token pruning via zeroth-order gradient estimation in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39572–39582, 2026.

Haosheng Li, Weixin Mao, Zihan Lan, Hongwei Xiong, Hongan Wang, Chenyang Si, Ziwei Liu, Xiaoming Deng, and Hua Chen. BFA++: Hierarchical best-feature-aware token prune for multi-view vision language action model. *IEEE Robotics and Automation Letters*, 11(5):6002–6009, 2026a. doi: 10.1109/LRA.2026.3673974.

Yuquan Li, Lianjie Ma, Han Ding, and Lijun Zhu. DepthCache: Depth-guided training-free visual token merging for vision-language-action model inference. *arXiv preprint arXiv:2603.10469*, 2026b.

Zhuoyuan Li, Rui Zhao, Jin Wang, Hanwei Zhu, Cong Zhang, Giuseppe Valenzise, Weisi Lin, and Kin-Man Lam. Just noticeable difference modeling for token compression in vision-language-action models. *arXiv preprint arXiv:2608.21247*, 2026c.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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*, volume 36, 2023.

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. *arXiv preprint arXiv:2511.16449*, 2025. Revised May 2026.

Shilin Ma, Chubin Zhang, Changyuan Wang, Yuji Wang, Yue Wu, Zixuan Wang, Jingqi Tian, Zheng Zhu, and Yansong Tang. SAFE-Pruner: Semantic attention-guided future-aware token pruning for efficient vision-language-action manipulation. In *European Conference on Computer Vision*, 2026.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *Proceedings of the 18th International Conference on Pattern Recognition*, volume 3, pp. 850–855. IEEE, 2006. doi: 10.1109/ICPR.2006.479.

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.

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*, volume 34, 2021.

Charles Spearman. The proof and measurement of association between two things. In James J. Jenkins and Donald G. Paterson (eds.), *Studies in Individual Differences: The Search for Intelligence*, pp. 45–58. Appleton-Century-Crofts, New York, 1961. doi: 10.1037/11491-005.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

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 *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for important tokens in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9961–9980, 2025. doi: 10.18653/v1/2025.emnlp-main.505.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968.

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*, volume 38, 2025. doi: 10.52202/085713-5484.

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*, volume 38, 2025. doi: 10.52202/085713-1365.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023.

# A Theoretical Details

The method links local deletion effects to an online selection rule through key derivatives and their randomized sketches. We establish each link below, from the single-head deletion identity to the separation and exact-budget guarantees of Budgeted Functional Selection.

## A.1 Proof of the Local Deletion Geometry

We first prove Proposition 3.1. Consider one attention query $q \in \mathbb{R}^{d_k}$ with attention-coordinate source keys $k_j \in \mathbb{R}^{d_k}$ and values $v_j \in \mathbb{R}^{d_v}$ indexed by $j \in \mathcal{I}$, where $d_v$ is the value dimension. These keys enter the dot product after any positional rotation. Let

$$s_j = \frac{q^\top k_j}{\sqrt{d_k}}, \qquad \alpha_j = \frac{\exp(s_j)}{\sum_{t\in\mathcal{I}} \exp(s_t)}, \qquad o = \sum_{j\in\mathcal{I}} \alpha_j v_j. \tag{14}$$

**Exact Local Deletion.** Removing token $i$ renormalizes each remaining attention weight:

$$\alpha_j^{(-i)} = \frac{\exp(s_j)}{\sum_{t\in\mathcal{I}\setminus\{i\}} \exp(s_t)} = \frac{\alpha_j}{1-\alpha_i}, \qquad j \neq i. \tag{15}$$

Consequently,

$$o^{(-i)} = \sum_{j\neq i} \frac{\alpha_j}{1-\alpha_i} v_j = \frac{o - \alpha_i v_i}{1-\alpha_i}, \tag{16}$$

$$o^{(-i)} - o = \frac{\alpha_i}{1-\alpha_i}(o - v_i). \tag{17}$$

**Key-Side Derivative.** To connect this finite deletion effect to local sensitivity, we differentiate the attention weights with respect to token $i$'s key:

$$\nabla_{k_i} \alpha_j = \frac{\alpha_j}{\sqrt{d_k}} \left(\mathbf{1}\{j = i\} - \alpha_i\right) q. \tag{18}$$

Therefore,

$$\begin{aligned} \frac{\partial o}{\partial k_i} &= \sum_{j\in\mathcal{I}} v_j \left(\nabla_{k_i} \alpha_j\right)^\top \\ &= \frac{1}{\sqrt{d_k}} \left( \alpha_i v_i - \alpha_i \sum_{j\in\mathcal{I}} \alpha_j v_j \right) q^\top \\ &= \frac{\alpha_i}{\sqrt{d_k}} (v_i - o) q^\top. \end{aligned} \tag{19}$$

Multiplying Equation 19 by $q$ gives

$$\left(\frac{\partial o}{\partial k_i}\right) q = \frac{\alpha_i \|q\|_2^2}{\sqrt{d_k}} (v_i - o). \tag{20}$$

For $q \neq 0$, combining Equations 17 and 20 yields

$$o^{(-i)} - o = -\frac{\sqrt{d_k}}{(1-\alpha_i)\|q\|_2^2} \left(\frac{\partial o}{\partial k_i}\right) q, \tag{21}$$

which proves Proposition 3.1.

**Exact Directional Equivalence.** For nonzero deletion effects from the same query and head, assume $q \neq 0$ and $0 < \alpha_i, \alpha_j < 1$. Equation 19 gives

$$\left\langle \frac{\partial o}{\partial k_i}, \frac{\partial o}{\partial k_j} \right\rangle_F = \frac{\alpha_i \alpha_j \|q\|_2^2}{d_k} (v_i - o)^\top (v_j - o), \tag{22}$$

$$\left\| \frac{\partial o}{\partial k_i} \right\|_F = \frac{\alpha_i}{\sqrt{d_k}} \|v_i - o\|_2 \|q\|_2, \tag{23}$$

$$\|o^{(-i)} - o\|_2 = \frac{\alpha_i}{1-\alpha_i} \|v_i - o\|_2.$$

Define $\cos_F(A, B) = \langle A, B\rangle_F/(\|A\|_F\|B\|_F)$ and $\cos(a, b) = a^\top b/(\|a\|_2\|b\|_2)$ for nonzero arguments. Each deletion effect is a negative multiple of $v_i - o$: the two sign reversals cancel in their inner product, and the positive scale factors cancel on normalization. Hence

$$\cos_F\left(\frac{\partial o}{\partial k_i}, \frac{\partial o}{\partial k_j}\right) = \cos\Big(o^{(-i)} - o, o^{(-j)} - o\Big). \tag{24}$$

Thus the normalized key-Jacobian Gram matrix exactly represents finite-deletion directional geometry in this base case. The keys are in attention coordinates, after RoPE when present; Appendix B.4 explains why this equality does not directly apply to PFF alignment computed from native pre-RoPE keys. It concerns full local Jacobians, not finite-probe PFF cosines or final actions.

### A.2 Why Use Key-Side Derivatives?

The directional equivalence above motivates key-side differentiation. To compare it with value-side differentiation, consider the same bare attention output:

$$\frac{\partial o}{\partial v_i} = \alpha_i I_{d_v}. \tag{25}$$

Its Frobenius magnitude is

$$\left\|\frac{\partial o}{\partial v_i}\right\|_F = \alpha_i\sqrt{d_v}. \tag{26}$$

Moreover, for any two tokens with positive attention weights,

$$\frac{\left\langle \frac{\partial o}{\partial v_i}, \frac{\partial o}{\partial v_j}\right\rangle_F}{\left\|\frac{\partial o}{\partial v_i}\right\|_F\left\|\frac{\partial o}{\partial v_j}\right\|_F} = 1. \tag{27}$$

At this attention output, value-side derivatives scale with the attention weight and share the same normalized derivative direction across source tokens.

In contrast, the key-side derivative retains both attention strength $\alpha_i$ and residual direction $v_i - o$. Its norm measures sensitivity, and its Frobenius cosine exactly matches finite-deletion cosine in the base case, as shown in Equation 24. The attention weight alone does not specify this residual direction.

Equation 27 applies only to the bare single-head attention output. Output projections, multiple heads, gating, and other local transformations can make value-side derivatives more expressive for the full action-related representation $F$. The ablations in Tables 2 and 7 therefore compare key-side and value-side PFFs.

### A.3 Multiple Queries, Heads, and Local Transformations

The preceding derivation treats one query and head. For the action-related representation $F$ in Equation 1, we combine derivatives across action tokens, attention heads, and local transformations by the chain rule. Let $o_{a,h}$ denote the attention output for action token $a \in \mathcal{A}$ and head $h$, and let $d_h$ denote the query/key dimension of each head. We define

$$G_{a,h} := \frac{\partial F}{\partial o_{a,h}} \tag{28}$$

to be the Jacobian of $F$ with respect to this attention output. Then the derivative with respect to the attention-coordinate head key of visual token $i$ is

$$\frac{\partial F}{\partial k_{i,h}} = \sum_{a\in\mathcal{A}} G_{a,h}\left[\frac{\alpha_{a,h,i}}{\sqrt{d_h}}(v_{i,h} - o_{a,h})q_{a,h}^\top\right]. \tag{29}$$

The token-wise Jacobian thus combines query-conditioned residual directions across action-token queries and heads after local transformations. The architecture determines the matrices $G_{a,h}$; the randomized sketch applies to any differentiable choice of $F$. The single-query cosine equality does not generally extend through these sums and transformations.

### A.4 Randomized Functional Sketch

Token selection requires comparing these Jacobians across visual tokens. The shared-probe sketch estimates their inner products without constructing the full matrices. Let

$$J_i = \frac{\partial F(K;x)}{\partial K_i} \in \mathbb{R}^{d_F \times d_k} \tag{30}$$

be the visual-key Jacobian from Equation 7, and define the dimension-scaled functional Gram entry

$$\kappa_{ij} = \frac{1}{d_F}\langle J_i, J_j\rangle_F. \tag{31}$$

Following the Rademacher trace estimator of Hutchinson (1989), for $R$ independent probes let

$$z_r = \frac{1}{\sqrt{d_F}}\epsilon_r, \qquad \epsilon_r \in \{-1,+1\}^{d_F}, \qquad r = 1,\dots,R, \tag{32}$$

where all entries of $\epsilon_r$ are independent Rademacher variables. The same probes are shared across all visual tokens. Define

$$g_i^{(r)} = J_i^\top z_r, \qquad \tau_i = \frac{1}{\sqrt{R}}\,\mathrm{vec}\left(g_i^{(1)},\dots,g_i^{(R)}\right). \tag{33}$$

*Proof of Proposition 3.2.* By construction,

$$\begin{aligned}\tau_i^\top \tau_j &= \frac{1}{R}\sum_{r=1}^{R}\left(g_i^{(r)}\right)^\top g_j^{(r)} \\ &= \frac{1}{R}\sum_{r=1}^{R} z_r^\top J_i J_j^\top z_r.\end{aligned} \tag{34}$$

Since

$$\mathbb{E}[z_r z_r^\top] = \frac{1}{d_F} I_{d_F}, \tag{35}$$

we obtain

$$\begin{aligned}\mathbb{E}\left[z_r^\top J_i J_j^\top z_r\right] &= \mathrm{tr}\left(J_i J_j^\top \mathbb{E}[z_r z_r^\top]\right) \\ &= \frac{1}{d_F}\,\mathrm{tr}\left(J_i J_j^\top\right) \\ &= \frac{1}{d_F}\langle J_i, J_j\rangle_F.\end{aligned} \tag{36}$$

Averaging over the $R$ probes proves the first identity in Equation 11; setting $i = j$ gives its second identity. □

Unbiasedness describes the expectation over probes; each pruning decision uses only $R$ probes. To quantify the resulting variability, let

$$A_{ij} = J_i J_j^\top, \qquad H_{ij} = \frac{A_{ij} + A_{ij}^\top}{2}. \tag{37}$$

Because $z^\top A_{ij} z = z^\top H_{ij} z$, independence of the Rademacher entries gives

$$\begin{aligned}\mathrm{Var}\left[\tau_i^\top \tau_j\right] &= \frac{4}{R d_F^2}\sum_{p<q}(H_{ij})_{pq}^2 \\ &\le \frac{2}{R d_F^2}\|H_{ij}\|_F^2 \\ &\le \frac{2}{R d_F^2}\|J_i J_j^\top\|_F^2.\end{aligned} \tag{38}$$

The variance decreases as $1/R$; accuracy at a given $R$ depends on the local Jacobians.

**Corollary A.1** (Finite-probe deviation)**.** *For any $\delta > 0$,*

$$\Pr\left(\left|\tau_i^\top \tau_j - \kappa_{ij}\right| \geq \delta\right) \leq \frac{2\|J_i J_j^\top\|_F^2}{R d_F^2 \delta^2}. \tag{39}$$

*Proof.* The result follows from Equation 38 and Chebyshev's inequality. □

Proposition 3.2 establishes unbiasedness for the dimension-scaled Gram entries. The PFF alignment used by Functional NMS,

$$c_{ij} = \frac{\tau_i^\top \tau_j}{\|\tau_i\|_2 \|\tau_j\|_2 + \epsilon}, \tag{40}$$

is a nonlinear plug-in estimate of normalized Jacobian alignment and need not be unbiased. As $R$ increases, the numerator and squared norms converge to their functional Gram entries. For nonzero Jacobians and negligible $\epsilon$, $c_{ij}$ therefore approaches

$$\rho_{ij} = \frac{\langle J_i, J_j \rangle_F}{\|J_i\|_F \|J_j\|_F}. \tag{41}$$

### A.5 Functional NMS via Iterative Matrix Refinement

The fingerprints supply both the priority and alignment scores needed by Functional NMS. To implement its retained-anchor rule with matrix operations, consider one camera view with a stable order of decreasing PFF energy,

$$(i_1, \ldots, i_{N_v}). \tag{42}$$

Define the strictly upper-triangular conflict matrix $H_v \in \{0,1\}^{N_v \times N_v}$ by

$$(H_v)_{rs} = \mathbf{1}\{r < s\}\, \mathbf{1}\{c_{i_r i_s} > \gamma\}. \tag{43}$$

The Functional NMS update is

$$\Phi_v(k)_s = \mathbf{1}\left\{\sum_{r=1}^{s-1} (H_v)_{rs} k_r = 0\right\}. \tag{44}$$

The sequential retained-anchor rule defines $g_v$ from left to right by $g_{v,s} = \Phi_v(g_v)_s$. Starting from $k_v^{(0)} = \mathbf{1}$, the matrix implementation iterates $k_v^{(u+1)} = \Phi_v(k_v^{(u)})$ until consecutive masks agree.

The fixed point is unique. Position $s = 1$ has no predecessor and is fixed after the first pass. If the first $u$ positions agree with $g_v$, then the first $u + 1$ positions agree after the next pass because each position depends only on earlier entries. Each pass thus fixes at least one more leading position, so the recurrence reaches $g_v$ within $N_v$ passes.

The fixed point determines the anchors; we then enforce the budget from Section 3.3. With anchors $\mathcal{A}_v$ and suppressed tokens $\mathcal{B}_v$ each in decreasing-energy order, the final selection is

$$\mathcal{S}_v = \mathrm{First}_{m_v}\big(\mathcal{A}_v \,\|\, \mathcal{B}_v\big), \tag{45}$$

where $\|$ denotes ordered concatenation and $\mathrm{First}_{m_v}$ returns its first $m_v$ tokens. If there are fewer than $m_v$ anchors, the remaining slots are filled by suppressed tokens in energy order.

*Proof of Proposition 3.3.* At the fixed point, an anchor has no conflict with any earlier retained anchor. Every pair of distinct anchors therefore satisfies $c_{ij} \leq \gamma$. Because only entries with fixed-point mask value one participate in $\Phi_v$, suppressed candidates cannot suppress later candidates. Finally, $\mathcal{A}_v$ and $\mathcal{B}_v$ partition all $N_v$ candidates. Their ordered concatenation therefore contains $N_v$ entries, and Equation 45 returns exactly $m_v$ of them. □

The separation guarantee applies to the Functional NMS anchor set $\mathcal{A}_v$, including any anchors omitted when the budget is smaller than $|\mathcal{A}_v|$. The final set $\mathcal{S}_v$ may contain pairs with PFF alignment above $\gamma$ when exact-budget fill reintroduces suppressed tokens.

# B ARCHITECTURE-SPECIFIC REALIZATIONS

To apply the local sensitivity analysis to a policy, we must specify the action-related representation $F$ (Equation 1), its visual-key coordinates, and the physical pruning boundary. We use the earliest attention operation with an informative action-to-vision interaction, which may differ from the physical pruning boundary. In both integrations, physical pruning occurs after transformer layer $\ell_p = 0$. Each PFF is constructed from the Jacobian of the local action-related attention update with respect to the native visual keys before rotary position embedding (RoPE) (Su et al., 2024). All nonvisual states, value-side inputs, masks, positional information, and model parameters are fixed during this local differentiation. Retained visual states are restored to sequence order and physically gathered with all nonvisual tokens to resume execution. These protected tokens include language, robot-state, special, and action tokens when present; none is eligible for visual-token pruning.

## B.1 $\pi_{0.5}$

For the evaluated $\pi_{0.5}$ checkpoint (Black et al., 2025a), the first transformer layer already provides an informative action-to-vision interaction. We therefore use the layer-0 action-query attention update to construct $F$ and prune immediately after this layer. The local computation follows the native attention operation, including normalization, query/key/value projection, positional rotation, masked attention, output projection, and the architecture's adaptive attention gating. The action-related representation $F$ is the mean of the resulting action-token updates.

The native visual layout contains three groups of 256 visual slots together with 200 protected nonvisual tokens. The third visual group corresponds to the masked camera input and is assigned zero pruning budget, while the main and wrist views receive a fixed $4{:}6$ allocation. Thus, for nominal visual retention ratios of 50%, 40%, 30%, and 20%, the corresponding budgets are $(154, 230, 0)$, $(123, 184, 0)$, $(92, 138, 0)$, and $(62, 92, 0)$ across the main, wrist, and masked views, respectively. The nominal ratio is computed over all 768 native visual slots, including the zero-budget masked group. Each 512-dimensional PFF uses key dimension $d_k = 256$ and $R = 2$ probes.

Layer 0 is evaluated densely. PFT-PRUNE compacts its states and associated sequence-dependent tensors; transformer layers 1–17 then process only the retained sequence. The resulting visual–language conditioning representation is reused through its compact key–value cache throughout the ten denoising queries. Pruning leaves the action-token sequence unchanged.

## B.2 OPENVLA-OFT

OpenVLA-OFT (Kim et al., 2025a) requires a different realization because its input action placeholders are initialized to zero. Consequently, the corresponding layer-0 action queries do not provide an informative action-to-vision functional signal. We therefore complete layer 0 densely and use layer 1 as the earliest informative local computation of $F$.

The completed layer-0 action states pass through the frozen layer-1 attention transformation to construct the local action-to-vision update. This proxy follows the checkpoint's native attention semantics, including layer-1 normalization, query/key/value projections, positional rotation, attention operation, and output projection. Only this local attention computation is replayed for fingerprint construction; later decoder layers and the action head are not re-evaluated during selection. The action-related representation $F$ averages the projected action-token updates.

The physical pruning boundary remains after layer 0. Using fingerprints from the layer-1 proxy, Budgeted Functional Selection selects visual tokens from the completed layer-0 sequence, which is then compacted. Transformer layers 1–31 and the action head execute once on this shortened sequence. This preserves an early pruning boundary while measuring informative action-conditioned visual gradients at the earliest available layer. Reported PFT-PRUNE policy latency includes the layer-1 proxy forward pass, vector–Jacobian products, Functional NMS, exact-budget fill, gathering, and compact forward execution.

OpenVLA-OFT contains two 256-token visual views. We again use a fixed main:wrist allocation of $4{:}6$. At visual retention ratios of 50%, 40%, 30%, and 20% over these 512 tokens, this yields budgets of $(102, 154)$, $(82, 123)$, $(62, 92)$, and $(41, 61)$, respectively. With $R = 2$, the resulting

PFFs use the same definitions of PFF energy, PFF alignment, and Budgeted Functional Selection as in the main text.

### B.3 Exact-Deletion Diagnostic Sampling

With the local computation of $F$ fixed, exact token deletion provides a reference for evaluating PFF energy and alignment. The diagnostics use 100 fixed OpenVLA-OFT policy queries, with 25 from each LIBERO suite (Liu et al., 2023). Figure 1 uses the original sample stratified by PFF energy, described below; the main mechanism validation in Figure 6 uses independent uniform sampling as defined in Appendix B.3.1. Each query contains 512 visual tokens: 256 from the main camera and 256 from the wrist camera. Exhaustively deleting every token would require 51,200 separate local recomputations. We instead evaluate 64 distinct tokens per query, reducing this total to 6,400 recomputations. The queries are shared across the two sampling protocols, while sampled indices are fixed within each protocol and need not coincide between them.

The original 64 tokens are selected deterministically rather than randomly. Within each query, we stably sort all 512 tokens from low to high PFF energy $a_i$ and divide the ordered list into five consecutive, approximately equal-sized rank strata. We then take evenly spaced rank positions within each stratum, selecting $(13, 13, 13, 13, 12)$ tokens from the five strata in low-to-high order; the one-token difference occurs because 64 is not divisible by five. "Evenly spaced" refers to PFF-energy rank, not to patch location in either image. Because the energy ordering depends on the current query, the resulting spatial positions also differ across queries. The 64 stored indices are reused for all attention, deletion-magnitude, and deletion-direction comparisons within that query. This stratification covers low- to high-energy tokens, including the high-energy candidates most relevant to pruning, while deterministic spacing makes the sample reproducible. The resulting statistics describe this fixed stratified sample rather than an unbiased estimate over all 512 positions.

**Rank Construction.** Let $\bar{\alpha}_i$ denote token $i$'s scalar attention score, obtained by averaging its attention weights over action-token queries and heads, and let $m_i$ be its exact deletion magnitude. Within each query, we rank both quantities over the same 64-token subset, assigning average ranks to ties. Spearman's rank correlation is the Pearson correlation between these rank vectors (Spearman, 1961):

$$\rho_S(\bar{\alpha}, m) = \frac{\sum_{i=1}^{64}(r_i^\alpha - \bar{r}^\alpha)(r_i^m - \bar{r}^m)}{\sqrt{\sum_{i=1}^{64}(r_i^\alpha - \bar{r}^\alpha)^2}\sqrt{\sum_{i=1}^{64}(r_i^m - \bar{r}^m)^2}}, \tag{46}$$

where $r_i^\alpha$ and $r_i^m$ are the ranks of $\bar{\alpha}_i$ and $m_i$, and $\bar{r}^\alpha$ and $\bar{r}^m$ are their within-query mean ranks. Query correlations are averaged equally; $+1$ denotes identical rankings and $0$ no linear association between ranks. The same rank convention is used for PFF energy and for the pairwise directional comparisons. A percentile for average rank $r$ among $n$ values is $100(r-1)/(n-1)$.

Figure 1(a) reports the equal-query mean of these within-query attention–magnitude Spearman correlations, $\overline{\rho_S} = 0.932$.

To examine differences that remain despite this strong scalar correlation, Figure 8 shows same-query, same-view pairs whose attention percentiles among all 512 tokens differ by at most two points and whose deletion magnitudes both exceed the sampled subset's 10th percentile. From the eligible pairs, the figure shows one from each named suite with at least a 20-point deletion-percentile gap and patches distinguishable at print scale. A and B identify the larger and smaller deletion effects within each pair; they are not fixed token classes.

**Attention-Controlled Comparisons.** We also test whether PFF energy remains associated with deletion magnitude after accounting for scalar attention. Within each query's 64-token sample, we rank deletion magnitude, the scalar attention score, and PFF energy, assigning average ranks to ties. We regress deletion-magnitude ranks on attention ranks by ordinary least squares with an intercept, then compute Spearman correlation between the residuals and energy ranks. Only deletion magnitude is residualized, so this statistic is not a partial correlation. For the conditioned curves, we divide each query's sample into five attention-rank groups, then divide each group into five energy-rank groups. Each cell's median deletion magnitude is normalized by the query's median deletion magnitude; the plotted value is the median across queries. Figure 9 shows increasing aggregate deletion magnitude across PFF-energy quintiles within each attention group. These aggregate curves do not imply per-query monotonicity. The mean residual correlation is $0.297$ and is positive in all

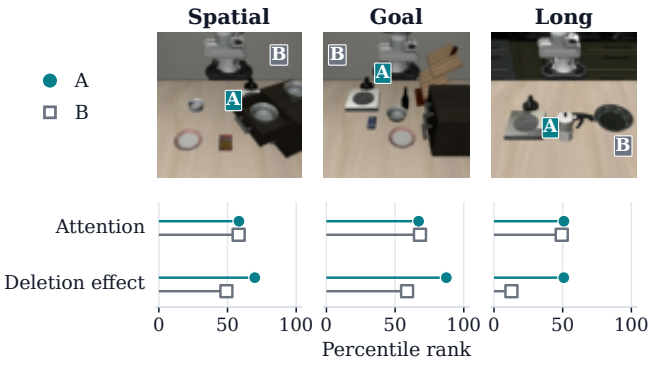



Figure 8: Attention-matched tokens can have different deletion magnitudes, consistent with a strong but incomplete scalar signal. Each image marks one audited pair at its actual patch locations; A has the larger deletion effect. Attention and deletion percentiles rank all 512 visual tokens and the 64 sampled tokens, respectively.

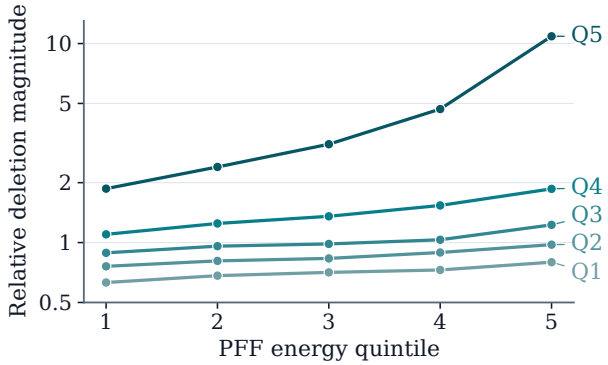



Figure 9: Deletion magnitude conditioned on attention and PFF energy in the original 100-query OpenVLA-OFT diagnostic. Q1–Q5 are increasing attention quintiles; PFF-energy quintiles are formed within each query's attention group. Points are across-query medians of cell medians, normalized by each query's median deletion magnitude. The vertical scale is logarithmic.

100 queries, indicating additional local sensitivity information beyond the linear attention-rank fit. This conditional association does not establish a scalar-ranking advantage over attention.

**Deletion-Direction Comparisons.** The scalar diagnostics do not reveal whether influential tokens have similar deletion directions. To examine this relationship within each view, we assign sampled tokens to their original cameras, yielding 200 query–view groups. For Figure 1(b), we take the ten sampled tokens with largest $m_i$ in each group; 176/200 groups contain at least one pair with deletion-direction similarity $d_{ij} \geq 0.8$.

### B.3.1 Sampling-Independence Checks

Because the preceding sample is stratified by PFF energy, it cannot by itself establish that the directional comparison persists under independent token sampling. The main mechanism validation in Figure 6 therefore uses uniform sampling. On the same 100 fixed OpenVLA-OFT queries, 25 per suite, we sample 64 unique positions uniformly without replacement from all 512 visual tokens, using a fixed random seed. There is no camera balancing, PFF-energy or attention stratification, or deletion-effect conditioning. A single pseudorandom stream processes the fixed query order; all index sets are frozen before model evaluation and reused across every metric, without metric-specific resampling.

**Local Computation and Statistical Comparison.** Each deletion masks one visual source and recomputes attention with fresh softmax renormalization, holding the remaining local inputs fixed. The intermediate action-related representation $F$ averages local attention outputs over action tokens (Equation 1). In the OpenVLA-OFT implementation, these outputs include the native output projection; $F$ is not the final action output. Attention-profile similarity is the Pearson correlation between two tokens' attention profiles across action-token queries and heads; averaging each profile gives the scalar attention score. We retain the original deletion implementation, PFF estimator, and residual-rank calculation. Correlations are computed within each query and averaged with equal query weights. Two-sided paired Wilcoxon signed-rank tests (Wilcoxon, 1945) compare the 100 query-level correlations, discarding exactly zero paired differences; individual token pairs are not treated as independent observations. Table 4 shows no statistically detected magnitude-fidelity dif-

Table 4: Exact-local-deletion diagnostics with 64 tokens per query on the same 100 OpenVLA-OFT queries. Uniform sampling is independent of PFF energy and underlies the main mechanism validation; stratification provides a supporting comparison. Correlations are mean within-query Spearman $\rho_S$. Directional advantage is the mean paired difference between the query-level correlations of PFF alignment and attention-profile similarity with $d_{ij}$. Aligned groups contain a pair with $d_{ij} \geq 0.8$ among their ten largest sampled effects.

| **Metric** | **Energy-stratified** | **Uniform** |
|---|---|---|
| *Deletion magnitude* | | |
| Scalar attention score vs. $m_i$ | 0.9321 | 0.9313 |
| PFF energy $a_i$ vs. $m_i$ | 0.9367 | 0.9332 |
| Magnitude paired-test $p$ | 0.2594 | 0.5520 |
| Residual PFF association | 0.2975 | 0.2987 |
| Positive residual queries | 100/100 | 100/100 |
| *Deletion direction* | | |
| Attention-profile similarity vs. $d_{ij}$ | 0.5162 | 0.5295 |
| PFF alignment $c_{ij}$ vs. $d_{ij}$ | 0.7197 | 0.7277 |
| Directional advantage | $+0.2035$ | $+0.1982$ |
| PFF directional wins | 100/100 | 99/100 |
| Direction paired-test $p$ | $3.90 \times 10^{-18}$ | $5.11 \times 10^{-18}$ |
| Groups with aligned strong effects | 176/200 | 176/200 |

ference under either sampling protocol, while PFF alignment has substantially stronger directional agreement. All 200 uniform-sampled query–view groups contain at least ten tokens and enter the strong-effect analysis; none is excluded.

**Sampling Robustness.** To check sensitivity to the sampled indices, we evaluate uniform sampling under three fixed random seeds in total. This gives a directional advantage of $0.1984 \pm 0.0041$, the mean and sample standard deviation across seed-level means. PFF alignment outperforms attention-profile similarity on 99/100 queries for every seed. Replications reuse 100 queries rather than provide 300 independent queries.

**Partial Exhaustive Evaluation.** To remove token sampling entirely on a smaller query set, we delete all 512 visual tokens on 20 fixed queries: the first five per suite ordered by task and initial-state index. Without token sampling, mean directional correlations are $0.7399$ for PFF alignment and $0.5102$ for attention-profile similarity, a gap of $+0.2297$, with PFF higher on all 20 queries. All 40 query–view groups contain an aligned pair among their ten largest effects. Here the ten effects are selected from all 256 tokens per view, so this frequency is not directly comparable to the sampled-group frequency. This supports the directional conclusion, but comparison with the 100-query aggregate confounds token-count and query-subset effects.

**Same-Forward Replay Control.** Fresh bf16 recomputation did not reproduce the historical saved deletion tensors bitwise. To separate token-sampling effects from these numerical replay differences, we repeat the original stratified analysis using quantities captured from the same forward pass as the uniform-sampling controls. The directional advantage remains $+0.2012$, compared with $+0.2035$ for the historical stratified result and $+0.1982$ for independent uniform sampling.

### B.4 Effect of the RoPE Coordinate Choice

The sampling checks hold the differentiation coordinates fixed. We next examine their effect: the single-head identity uses attention-coordinate keys, whereas production PFFs use native pre-RoPE keys. Let $\widetilde{K}_i = R_i K_i$ be the attention-coordinate key, where $K_i$ is the production native key and $R_i$ is the fixed orthogonal RoPE map on its flattened key block. With $\widetilde{J}_i = \partial F / \partial \widetilde{K}_i$, the chain rule gives

$$\begin{aligned} J_i &= \widetilde{J}_i R_i, \qquad \|J_i\|_F = \|\widetilde{J}_i\|_F, \\ \langle J_i, J_j \rangle_F &= \operatorname{tr}\Big( \widetilde{J}_i R_i R_j^\top \widetilde{J}_j^\top \Big). \end{aligned} \tag{47}$$

For $\widetilde{g}_i^{(r)} = \widetilde{J}_i^\top z_r$, identical probes give $g_i^{(r)} = R_i^\top \widetilde{g}_i^{(r)}$, hence identical PFF energies at every finite $R$ in exact arithmetic. Cross-token inner products and cosines need not agree because they contain the relative rotation $R_i R_j^\top$. PRE uses native coordinates; POST uses attention coordinates.

| **Alignment signal** | **Mean $\rho_S$** | **Median $\rho_S$** |
|---|---|---|
| Attention-profile similarity | 0.5295 | 0.5288 |
| PRE-RoPE PFF alignment | 0.7293 | 0.7463 |
| POST-RoPE PFF alignment | 0.7473 | 0.7619 |

Table 5: Exact deletion-direction fidelity under matched PRE/POST computation. Entries summarize query-level Spearman correlations.

Table 5 compares PRE and POST using the same 100 OpenVLA-OFT queries and the primary stored 64-token uniform samples from Appendix B.3.1, with identical deletion targets, local $F$, and $R = 2$ shared probes. PRE is recomputed here rather than taken from the historical float16-stored fingerprints in Table 4. POST improves directional correlation by $0.0179$, with 97 wins and 3 losses across queries, paired Wilcoxon $p = 1.50 \times 10^{-17}$, and a 95% query-bootstrap interval of $[0.0157, 0.0202]$. The mean gain across three uniform seeds is $0.0180$; the original stratified sample gives $0.0178$. PRE/POST energy Spearman correlation is $0.9999999$, with a mean relative difference of $0.049\%$.

Table 6: Closed-loop control at 20% nominal visual-token retention and $\gamma = 0.90$. Both conditions keep production PRE PFF-energy priority and change only the alignment coordinate. Success is in percent; $\Delta$ and paired-bootstrap intervals are POST$-$PRE in percentage points. Exact two-sided McNemar tests use 2,000 matched OpenVLA-OFT episodes and 800 matched $\pi_{0.5}$ episodes.

| **Policy** | **PRE (%)** | **POST (%)** | **$\Delta$ (pp)** | **McNemar $p$** | **95% paired CI (pp)** |
|---|---|---|---|---|---|
| OpenVLA-OFT | 93.20 | 93.15 | $-0.05$ | 1.000 | $[-1.05, +0.95]$ |
| $\pi_{0.5}$ | 94.13 | 92.88 | $-1.25$ | 0.320 | $[-3.50, +0.88]$ |

To test whether improved local fidelity also benefits control, we change only the alignment coordinate while keeping production PRE PFF-energy priority (Table 6). PRE and POST succeed in 1864 and 1863 of 2000 OpenVLA-OFT episodes, and 753 and 743 of 800 $\pi_{0.5}$ episodes, respectively. POST improves local directional fidelity on OpenVLA-OFT but does not increase observed closed-loop success on either policy; neither matched test detects a difference (Table 6). We therefore retain native pre-RoPE PFFs for production. This does not establish equivalence or make PRE a more exact local proxy: better single-token directional fidelity is not sufficient to improve multi-token selection under the fixed Functional NMS rule.

# C Selection Algorithm, Physical Execution, and Complexity

The architecture-specific computations of $F(K; x)$ in Appendix B feed into the shared inference procedure in Algorithm 1. Both policies construct PFFs through reverse-mode vector–Jacobian products (VJPs), apply Budgeted Functional Selection, and gather the tensors needed to resume execution.

## C.1 Generic PFT-Prune Algorithm

One reverse-mode differentiation obtains all token-wise derivatives for each probe. The $R$ directions can be evaluated sequentially or with batched gradient outputs; our implementations use batched reverse mode. Policy parameters remain frozen without gradient buffers.

The suppressed set $\mathcal{B}_v$ preserves decreasing-energy order because its mask is gathered according to $\pi_v$. Hence, $\mathrm{First}_k(\mathcal{B}_v)$ implements the deterministic fill rule in Equation 45 without an additional sort. Only retained anchors suppress later candidates.

## C.2 Physical Compact Execution

Selection provides the token indices; compact execution must apply them consistently to the policy state. Let $\mathcal{E}_{\ell_p}$ denote this state at the pruning boundary, after layer $\ell_p$. Depending on the architecture,

**Algorithm 1** Policy-Functional Token Pruning

**Require:** Visual-token sets $\{\mathcal{V}_v\}_{v=1}^{M}$ with $\mathcal{V} = \bigcup_{v=1}^{M} \mathcal{V}_v$; protected nonvisual tokens $\mathcal{P}$; per-view budgets $\{m_v\}_{v=1}^{M}$; pruning layer $\ell_p$; action-related representation $F(K; x)$; $R$ fixed shared Rademacher probes $z_r = \epsilon_r/\sqrt{d_F}$; PFF-alignment threshold $\gamma$; stabilizer $\epsilon$.

**Ensure:** Compact policy execution state and selected index set $\mathcal{S}$.

Run the policy with all visual tokens through layer $\ell_p$.
Construct the local computation of $F(K; x)$ with $K$ as the differentiable leaf.
**for** $r = 1, \ldots, R$ **do**
    $G^{(r)} \leftarrow \nabla_K \langle z_r, F(K; x) \rangle$ ▷ all visual-token VJPs jointly
**end for**
**for all** $i \in \mathcal{V}$ **do**
    For each $r = 1, \ldots, R$, extract $g_i^{(r)}$ from the block of $G^{(r)}$ corresponding to $K_i$.
    $\tau_i \leftarrow \dfrac{1}{\sqrt{R}} \operatorname{vec}\left(g_i^{(1)}, \ldots, g_i^{(R)}\right)$.
    $a_i \leftarrow \|\tau_i\|_2^2$.
**end for**
**for** $v = 1, \ldots, M$ **do**
    **if** $m_v = 0$ **then**
        $\mathcal{S}_v \leftarrow \varnothing$.
    **else**
        $\pi_v \leftarrow \mathrm{StableSort}(\mathcal{V}_v;\ \text{decreasing } a_i) = (i_1, \ldots, i_{N_v})$.
        Compute the within-view matrix

$$(C_v)_{ij} \leftarrow \frac{\tau_i^\top \tau_j}{\|\tau_i\|_2 \|\tau_j\|_2 + \epsilon}, \qquad i, j \in \mathcal{V}_v.$$

        Form $(H_v)_{rs} \leftarrow \mathbf{1}\{r < s\}\mathbf{1}\{(C_v)_{i_r, i_s} > \gamma\}$.
        $k_v^{(0)} \leftarrow \mathbf{1}$.
        **for** $u = 0, \ldots, N_v - 1$ **do**
            $k_{v,s}^{(u+1)} \leftarrow \mathbf{1}\left\{\sum_{r=1}^{s-1} (H_v)_{rs} k_{v,r}^{(u)} = 0\right\}$ for all $s$ in parallel.
            **if** $k_v^{(u+1)} = k_v^{(u)}$ **then**
                **break**
            **end if**
        **end for**
        $k_v^* \leftarrow k_v^{(u+1)}$.
        $\mathcal{A}_v \leftarrow [i_s : k_{v,s}^* = 1]$ in the order $\pi_v$.
        $\mathcal{B}_v \leftarrow [i_s : k_{v,s}^* = 0]$ in the order $\pi_v$.
        $\mathcal{S}_v \leftarrow \mathrm{First}_{m_v}(\mathcal{A}_v \,\|\, \mathcal{B}_v)$.
    **end if**
**end for**
$\mathcal{S} \leftarrow \mathcal{P} \cup \bigcup_{v=1}^{M} \mathcal{S}_v$.
$I_{\mathcal{S}} \leftarrow \mathrm{Sort}_{\mathrm{seq}}(\mathcal{S})$.
Physically gather the policy execution state using $I_{\mathcal{S}}$.
Discard the local derivative graph and resume policy execution from the compact state.
**return** compact execution state and $\mathcal{S}$.

this state can contain hidden representations, attention masks, position indices, key–value (KV) tensors from already evaluated layers, and other sequence-aligned quantities needed to resume policy execution. After selection, we first form the ordered index vector

$$I_{\mathcal{S}} = \mathrm{Sort}_{\mathrm{seq}}(\mathcal{S}), \tag{48}$$

which restores retained visual tokens and protected nonvisual tokens to their original sequence order.

The ordered vector $I_{\mathcal{S}}$ is then used to reindex all remaining sequence-aligned tensors. Hidden states are gathered along their token dimension, and required cached KV tensors along their sequence axis. Position indices and one-dimensional token masks follow the same ordering, while attention masks with both query and source sequence axes are reindexed along both axes when applicable. Using the same index vector throughout keeps hidden states, positions, masks, and cached attention states aligned.

Physical gathering differs from assigning zero attention weight to discarded tokens. Mask-only pruning preserves the original tensor dimensions and sequence length

$$L = |\mathcal{P}| + \sum_{v=1}^{M} N_v. \tag{49}$$

Physical gathering reduces the sequence length to

$$L_{\mathrm{c}} = |\mathcal{P}| + \sum_{v=1}^{M} m_v. \tag{50}$$

Layers through $\ell_p$ are not recomputed for policy execution. After fingerprint construction, the local derivative graph is released and execution resumes from the gathered state, without a second complete forward pass or dense-cache reconstruction.

### C.3 Selector Complexity

The savings from compact execution depend on the overhead of constructing fingerprints and selecting tokens. To quantify this overhead, let

$$N = \sum_{v=1}^{M} N_v, \qquad m = \sum_{v=1}^{M} m_v, \qquad D_\tau = Rd_k. \tag{51}$$

The complete token-wise Jacobians would contain

$$\mathcal{O}(Nd_F d_k) \tag{52}$$

entries. PFT-Prune instead stores $N$ fingerprints of dimension $D_\tau$, with memory cost

$$\mathcal{O}(NRd_k). \tag{53}$$

Let $\mathcal{C}_{\mathrm{proxy}}$ denote the forward cost of constructing the action-related representation $F$, including the layer-1 attention replay for OpenVLA-OFT. Let $\mathcal{C}_F$ denote the cost of one reverse-mode vector–Jacobian product through this graph, and $\mathcal{M}_F$ its retained activation memory. Constructing the graph once and computing the $R$ projected derivatives requires

$$\mathcal{O}(\mathcal{C}_{\mathrm{proxy}} + R\mathcal{C}_F) \tag{54}$$

arithmetic. The derivative cost is linear in probe count even with batched reverse mode.

For each view, the implementation forms a dense PFF-alignment matrix. Its arithmetic cost is

$$\mathcal{O}(N_v^2 D_\tau), \tag{55}$$

giving the total

$$\mathcal{O}\left(\sum_{v=1}^{M} N_v^2 Rd_k\right) \tag{56}$$

across views. The additional storage for these matrices is

$$\mathcal{O}\left(\sum_{v=1}^{M} N_v^2\right). \tag{57}$$

Stable priority ordering costs

$$\mathcal{O}\left(\sum_{v=1}^{M} N_v \log N_v\right). \tag{58}$$

Let $q_v \leq N_v$ be the number of matrix-refinement passes required for view $v$; Appendix A.5 establishes this bound. Each pass applies the strictly upper-triangular conflict matrix to the current anchor mask, requiring

$$\mathcal{O}\left(\sum_{v=1}^{M} q_v N_v^2\right) \tag{59}$$

boolean operations and at most $\mathcal{O}(\sum_v N_v^3)$ in the conservative worst case. The passes are whole-matrix operations rather than candidate-wise host iterations. Deterministic fill is linear in the number of candidates and requires no additional similarity computation.

Combining these terms, the selector has arithmetic cost

$$\mathcal{O}\left(\mathcal{C}_{\text{proxy}} + R\mathcal{C}_F + NRd_k + \sum_{v=1}^{M} N_v^2 R d_k + \sum_{v=1}^{M} N_v \log N_v + \sum_{v=1}^{M} q_v N_v^2\right), \tag{60}$$

and peak auxiliary storage, excluding the policy state itself, of

$$\mathcal{O}\left(\mathcal{M}_F + NRd_k + \sum_{v=1}^{M} N_v^2\right). \tag{61}$$

Physical gathering is linear in the retained element count of the sequence-aligned tensors.

The cost of computation through layer $\ell_p$ is unchanged. The benefit arises in subsequent layers that process the retained sequence, whose length is reduced from $L$ to $L_{\text{c}}$. For conventional transformer layers (Vaswani et al., 2017), this reduces both sequence-linear projection and feed-forward terms and the sequence-quadratic self-attention term. For $\pi_{0.5}$, the compact conditioning cache also reduces the context attended to during action denoising without changing the action-token count. The realized end-to-end benefit also depends on proxy construction, VJPs, gathering, hardware utilization, compilation, and the computation remaining after pruning; measured selector and policy latencies are therefore reported rather than inferred solely from nominal retention.

### C.4 Determinism and Edge Cases

The sketch analysis averages over random probes, but the production selector fixes its Rademacher probes, making selection deterministic for the same policy state and numerical backend. The stable sort preserves the original sequence order of equal-energy tokens, and the fixed-point recurrence converges to the unique retained-anchor mask. Only retained anchors suppress later candidates, while suppressed candidates retain their priority order for deterministic exact-budget fill.

Selection is performed independently within each view under its assigned budget, so a view with $m_v = 0$ contributes no selected visual token. Shared probes make fingerprints mathematically comparable across views, but the production rule does not allow one view to consume another view's assigned budget. This choice separates the functional selection criterion from the policy-specific view-allocation scheme. After visual selection, the final union with $\mathcal{P}$ preserves every nonvisual token.

The stabilizer $\epsilon$ keeps PFF alignment numerically defined when fingerprints have near-zero norm. Unless explicitly varied in Appendix F, all PFT-Prune configurations use $R = 2$ fixed probes, $\gamma = 0.90$, and $\epsilon = 10^{-12}$.

### C.5 Budget-Selection Diagnostic Protocol

Exact-budget fill can reintroduce suppressed tokens, so the final set need not preserve all separation imposed by Functional NMS. To measure how strongly alignment changes selection at each budget, we record the retained-anchor and fill shares and compare the selected set with PFF-Energy Top-K. This baseline retains the $m_v$ tokens with largest $a_i$ in each view without Functional NMS. Both selectors use the same fingerprints and energy order; the comparison isolates how pairwise alignment changes the retained set. Let the total retained budget be $m = \sum_v m_v > 0$. For each query, the retained-anchor count is $n_{\text{anchor}} = \sum_v \min(|\mathcal{A}_v|, m_v)$, and the fill count is $n_{\text{fill}} = m - n_{\text{anchor}}$. Their budget shares are $n_{\text{anchor}}/m$ and $n_{\text{fill}}/m$, respectively; anchors beyond the budget are not

counted as retained. We average these shares with equal query weights. Let $\mathcal{S}^{\text{vis}}_{\text{full}}$ and $\mathcal{S}^{\text{vis}}_{\text{energy}}$ denote the visual-only sets returned by full PFT-PRUNE and PFF-Energy Top-K, respectively. Each set combines retained visual indices from both camera views. We compute Jaccard overlap per query as $|\mathcal{S}^{\text{vis}}_{\text{full}} \cap \mathcal{S}^{\text{vis}}_{\text{energy}}|/|\mathcal{S}^{\text{vis}}_{\text{full}} \cup \mathcal{S}^{\text{vis}}_{\text{energy}}|$ and average these overlaps with equal query weights. We compute 95% percentile confidence intervals for aggregate means from 5,000 with-replacement bootstrap resamples (Efron, 1979) of 200 fixed $\pi_{0.5}$ queries, 50 per suite. This corpus is separate from the 100-query OpenVLA-OFT deletion diagnostic. Figure 5 in Section 5 reports the anchor/fill composition, suite-level variation, and selected-set overlap across the tested retention budgets.

# D ADDITIONAL ABLATION RESULTS

The budget diagnostic measures changes in selected sets; the ablations test how changes to the selector affect closed-loop success. Table 7 extends the OpenVLA-OFT component comparisons to $\pi_{0.5}$ at 20% nominal visual-token retention across the four LIBERO suites, using the checkpoint and protocol of Table 1. Unless varied, the selector uses $R = 2$ shared probes and PFF-alignment threshold $\gamma = 0.90$, with the same per-view budgets across variants.

Table 7: Component ablation of PFT-PRUNE on $\pi_{0.5}$ at 20% nominal visual-token retention across the four LIBERO suites. Entries are success rates (%) and policy latency (ms); bold marks the best average success. Latencies are query-weighted means, whereas Table 1 averages suite-level means.

| Variant | Spatial | Object | Goal | Long | Avg.↑ | Lat.↓ |
|---|---|---|---|---|---|---|
| Full PFT-PRUNE | 96.0 | 94.5 | 95.0 | 91.0 | **94.1** | 49.35 |
| Native-key norm + cosine | 93.5 | 94.5 | 95.0 | 88.5 | 92.9 | 47.18 |
| Attention priority | 97.0 | 97.0 | 91.5 | 89.5 | 93.8 | 49.74 |
| PFF-Energy Top-K | 92.0 | 88.0 | 82.0 | 88.5 | 87.6 | 48.27 |
| PFF energy + native-key cosine | 96.0 | 95.0 | 96.5 | 88.0 | 93.9 | 49.76 |
| Value-side PFF | 95.0 | 92.0 | 88.5 | 87.5 | 90.8 | 49.23 |

Native-key norm + cosine replaces PFFs for both priority and alignment, reducing average success from 94.1% to 92.9%. Attention priority replaces only PFF-energy priority with early-layer attention, retaining PFF alignment and yielding 93.8%. PFF-Energy Top-K selects by PFF energy under the same per-view budgets without Functional NMS, yielding 87.6%. Attention remains a useful priority signal, but energy-only selection loses 6.5 percentage points relative to the full method.

To test whether raw key features provide an effective alignment signal, PFF energy + native-key cosine replaces only the pairwise comparison, retaining PFF-energy priority, the threshold, retained-anchor rule, exact-budget fill, and view budgets. Here, native-key cosine compares raw native-key features, not native-coordinate PFFs derived from key Jacobians. Its 93.9% average success is close to the full method's 94.1% at closely matched latency. Long decreases by 3.0 percentage points, while Object and Goal improve by 0.5 and 1.5 points, respectively, and Spatial is unchanged. Compared with the larger loss on OpenVLA-OFT in Table 2, this indicates that the native-key cosine control has policy-dependent effects. Value-side PFFs yield 90.8%, supporting the key-side differentiation interface on $\pi_{0.5}$ as well.

**OpenVLA-OFT Native-Key Cosine Control.** For OpenVLA-OFT, this control lowers average success by 2.15 percentage points before rounding. Long has the largest drop while Object and Goal improve. These suite-level differences underlie the aggregate comparison in Table 2.

**OpenVLA-OFT Selection Examples.** To examine how Functional NMS changes the retained set, Figure 4 compares PFT-PRUNE with PFF-Energy Top-K at 20% nominal visual-token retention, holding the observation, PFF energies, checkpoint, and per-view budgets fixed. We use the fixed 100-query OpenVLA-OFT corpus, with 25 queries per suite, described in Appendix B.3. Selections span all 512 visual positions, not the 64-token deletion samples. Full-method indices are captured from the production selector; PFF-Energy Top-K takes the first 41 main-view and 61 wrist-view indices from the same saved PFF-energy ordering.

Table 8: Outcome-independent queries used in Figure 4. Jaccard values are rounded to four decimals for display; selection uses exact values.

| Suite | Query identifier | $J_{\text{main}}$ | $J_{\text{wrist}}$ | $\overline{J}$ |
|---|---|---|---|---|
| Spatial | `libero_spatial__task05__ep00__q000` | 0.4138 | 0.5443 | 0.4790 |
| Object | `libero_object__task02__ep01__q000` | 0.4643 | 0.6053 | 0.5348 |
| Goal | `libero_goal__task09__ep01__q000` | 0.7083 | 0.4353 | 0.5718 |
| Long | `libero_10__task05__ep00__q000` | 0.4643 | 0.4353 | 0.4498 |

For each query, let $\mathcal{V}_{\text{PFT},v}$ and $\mathcal{V}_{\text{Energy},v}$ be the two retained sets in view $v$. We compute

$$J_v = \frac{|\mathcal{V}_{\text{PFT},v} \cap \mathcal{V}_{\text{Energy},v}|}{|\mathcal{V}_{\text{PFT},v} \cup \mathcal{V}_{\text{Energy},v}|}, \qquad \overline{J} = \frac{J_{\text{main}} + J_{\text{wrist}}}{2}.$$

Within each suite, we choose the query whose $\overline{J}$ is closest to the suite median. Exact ties are broken by lexicographic query identifier, then task and initial-state index; comparisons use unrounded values. The four query identifiers are frozen before viewing or rendering images. Selection is independent of rollout success or failure, and both camera views come from the same query. Table 8 lists the selected cases. Each lies exactly at its suite median; Long has a tie between tasks 05 and 06 at initial state 00, resolved in favor of task 05.

Each panel overlays the shared tokens and the two selector-specific set differences on the unchanged model-input image. Boxes follow its $16 \times 16$ grid of $14 \times 14$-pixel patches; wrist indices are offset by 256. The three categories partition the union of the two selections. Patches outside this union are faded with a translucent white overlay; patches retained by either selector remain at full image opacity. The images illustrate changes in retained sets, not the causal contribution of individual patches to rollout outcomes.

## E Real-World Robot Evaluation

The physical-robot comparison in Table 3 evaluates pruning beyond the simulation setting used for the ablations. Every method uses the same task-specific policy, camera observations, robot interface, and rollout protocol; all pruning methods use 50% nominal visual-token retention.

**Robot Platform.** Experiments use an AgileX NERO 7-DoF follower arm with a controllable gripper. Figure 10 shows the leader arm and both cameras used for collection and evaluation.

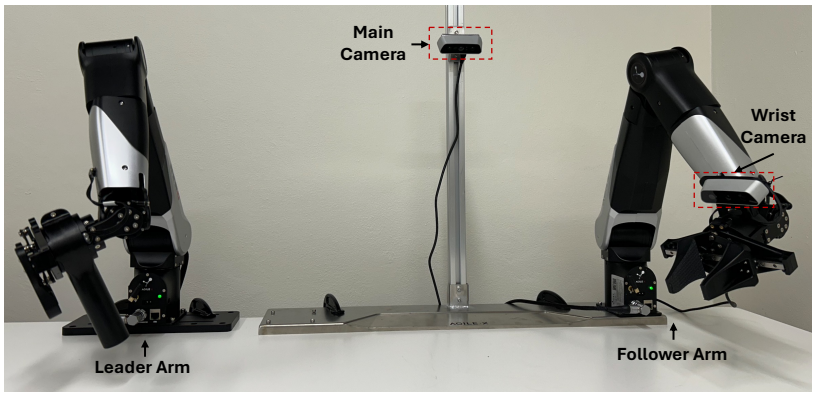



Figure 10: Robot setup: leader and NERO follower arms, fixed main camera, and wrist camera.

**Sensing and Execution.** The fixed camera observes the workspace and object arrangement, while the wrist-mounted camera follows the gripper and provides a close view during grasping and placement. The policy receives both images together with the language instruction and the robot's proprioceptive state. Policy inference runs remotely on an NVIDIA RTX 4090 workstation; a local computer collects observations, communicates with the policy server, and executes its commands.

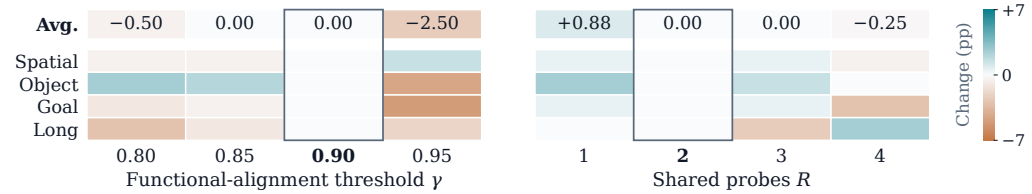



Figure 11: Hyperparameter sensitivity of $\pi_{0.5}$ at 20% nominal visual-token retention. The left map varies the PFF-alignment threshold $\gamma$ with $R = 2$; the right map varies the probe count $R$ with $\gamma = 0.90$. Cells show success-rate changes in percentage points (pp) relative to the default $\gamma = 0.90$ and $R = 2$, marked by outlined columns. Avg. is the unweighted four-suite mean change.

**Demonstrations and Policy Adaptation.** To adapt $\pi_{0.5}$ to this robot and its tasks, we collect 50 complete demonstrations per task through leader–follower teleoperation and store them in LeRobot format (Cadene et al., 2024). Each timestep contains the two camera images, an 8-dimensional robot state, and an 8-dimensional action target. In both vectors, the final coordinate represents the gripper and the other seven coordinates represent the arm joints. We fine-tune a separate OpenPI $\pi_{0.5}$ policy (Black et al., 2025a) for each task for 30,000 optimization steps. The policy predicts action chunks in $\mathbb{R}^{H\times 8}$, where $H$ is the action-chunk length. The client executes each chunk sequentially while requesting updated chunks from new observations.

**Tasks and Evaluation Protocol.** Before each trial, the arm returns to a fixed initial pose and objects are reset. Execution uses the demonstration control frequency. Task 1 places a green cucumber in a foil tray, and Task 2 places a pink block in a black target area. Task 3 first stacks two foil trays and then places the cucumber inside; Task 4 places the pink block in a soap dish and then closes the dish with its lid. Success requires completing the full instruction. We evaluate 100 trials per task–method pair and report the four-task mean success rate. Figure 3 shows representative long-horizon rollouts for Tasks 3 and 4 from both cameras.

## F Hyperparameter Sensitivity

The component ablations use a fixed probe count and alignment threshold. We assess sensitivity to these settings at 20% nominal visual-token retention. A candidate is suppressed when its PFF alignment with an already retained anchor exceeds the threshold $\gamma$; $R$ is the number of shared random probes in the output space of $F$. Each sweep changes one quantity and otherwise retains the full method's key-side fingerprints, per-view budgets, and iterative matrix refinement to convergence. Figure 11 summarizes the $\pi_{0.5}$ results.

**PFF-Alignment Threshold.** Table 9 varies $\gamma$ for $\pi_{0.5}$. Values between 0.80 and 0.90 yield 93.6%–94.1% average success; $\gamma = 0.85$ and the default $\gamma = 0.90$ both obtain 753 successes in 800 episodes. Relative to the default, $\gamma = 0.80$ differs by $-0.5$ percentage points, whereas $\gamma = 0.95$ decreases success by 2.5 points. Two-sided exact McNemar tests on paired binary episode outcomes give $p = 0.741$ and $p = 0.035$, respectively (McNemar, 1947). This exploratory four-value sweep is not corrected for multiple comparisons; the evidence against 0.95 is therefore limited to this setting. Increasing $\gamma$ reduces mean Functional NMS time from 0.954 to 0.700 ms because fewer pairs cross the threshold, while policy latency spans 49.35–50.22 ms. We retain $\gamma = 0.90$, which is tied for the highest average success and uses less NMS time than $\gamma = 0.85$.

**Number of Shared Probes.** With the alignment threshold fixed at $\gamma = 0.90$, Table 10 varies the shared-probe count $R$ to assess sensitivity to the fingerprint's sketch size. On $\pi_{0.5}$, average success spans 93.9%–95.0% for $R \in \{1, 2, 3, 4\}$. Compared with $R = 2$, the $R = 1$, $R = 3$, and $R = 4$ differences are $+0.88$, 0.00, and $-0.25$ percentage points; two-sided exact McNemar tests on paired binary episode outcomes give $p = 0.464$, $p = 1.000$, and $p = 0.904$, respectively, with no significant differences detected. At $R = 3$, suite-level differences relative to $R = 2$ are $+0.5$, $+1.5$, $+0.5$, and $-2.5$ points for Spatial, Object, Goal, and Long. Fixed-corpus policy-latency means span 1.37 ms across the four settings; because $R = 3$ was timed in a separate matched-protocol session, we do not interpret its lower absolute mean as an intrinsic speedup.

Table 9: PFF-alignment threshold sensitivity on $\pi_{0.5}$ at 20% nominal visual-token retention. Success is in percent; latencies are query-weighted synchronized means from closed-loop runs, whereas Table 1 averages suite-level means. Teal shading and bold parameter values mark the default.

| $\gamma$ | Spatial | Object | Goal | Long | Avg.↑ | NMS↓ (ms) | Lat.↓ (ms) |
|---|---|---|---|---|---|---|---|
| 0.80 | 95.5 | 97.0 | 94.0 | 88.0 | 93.6 | 0.954 | 50.22 |
| 0.85 | 95.5 | 96.5 | 94.5 | 90.0 | 94.1 | 0.895 | 49.99 |
| **0.90** | 96.0 | 94.5 | 95.0 | 91.0 | 94.1 | 0.820 | 49.35 |
| 0.95 | 97.5 | 90.0 | 90.0 | 89.0 | 91.6 | 0.700 | 49.45 |

On OpenVLA-OFT, aggregate success rises from 91.6% at $R = 1$ to 93.2% at $R = 2$; this difference is concentrated in Object, which rises from 86.0% to 92.8%. The intermediate $R = 3$ setting obtains 1870/2000 successes, or 93.50%, six more than $R = 2$, with suite-level differences of $+0.4$, $+0.8$, $+0.4$, and $-0.4$ points for Spatial, Object, Goal, and Long. Increasing to $R = 4$ yields 1863/2000 successes, or 93.15%, one fewer than $R = 2$. Mean policy latency is 59.22, 59.33, and 59.57 ms for $R = 2$, $R = 3$, and $R = 4$, respectively. Episode-level identities are unavailable in the existing production artifacts needed for matched comparisons, so these OpenVLA-OFT differences are descriptive rather than paired tests. We retain $R = 2$ as the common default across policies: $R = 1$ performs worse on OpenVLA-OFT without reducing measured latency, while the wider $R = 3$ and $R = 4$ fingerprints provide no established improvement over $R = 2$.

| $R$ | Spatial | Object | Goal | Long | Avg.↑ | Lat.↓ (ms) |
|---|---|---|---|---|---|---|
| | | | $\pi_{0.5}$ | | | |
| 1 | 96.5 | 97.0 | 95.5 | 91.0 | 95.0 | 48.33 |
| **2** | 96.0 | 94.5 | 95.0 | 91.0 | 94.1 | 48.33 |
| 3 | 96.5 | 96.0 | 95.5 | 88.5 | 94.1 | 46.96 |
| 4 | 95.5 | 94.5 | 92.0 | 93.5 | 93.9 | 47.59 |
| | | | **OpenVLA-OFT** | | | |
| 1 | 94.6 | 86.0 | 97.0 | 88.8 | 91.6 | 59.27 |
| **2** | 94.6 | 92.8 | 95.2 | 90.2 | 93.2 | 59.22 |
| 3 | 95.0 | 93.6 | 95.6 | 89.8 | 93.5 | 59.33 |
| 4 | 93.6 | 93.0 | 95.8 | 90.2 | 93.2 | 59.57 |

Table 10: Sensitivity to the number of shared probes at 20% nominal visual-token retention with $\gamma = 0.90$. Suite and average entries are success rates (%). Within-policy latency comparisons use matched protocols: a fixed post-warm-up query corpus for $\pi_{0.5}$ and the mean suite-level production latency for OpenVLA-OFT. Raw latency is not compared across policies. Teal shading and bold parameter values mark the default.

## G Scope of Comparison

The component and sensitivity studies hold the pruning schedule fixed. Comparisons with other methods also need to account for differences in when and how many tokens are removed, which affect the interpretation of nominal retention and measured latency.

### G.1 Compared Methods

The five baselines in Table 1 cover attention-based importance, text-conditioned sparsification, feature redundancy, action-adaptive pruning, and speculative layerwise pruning. Each method is integrated into the same $\pi_{0.5}$ or OpenVLA-OFT policy used by PFT-Prune and removes only visual tokens. All nonvisual tokens, including language, state, special, and action tokens, are retained.

**FastV.** FastV ranks image tokens by early-layer attention and retains the highest-scoring subset for subsequent layers (Chen et al., 2024). This forward-pass priority does not explicitly compare candidates' local effect directions.

**SparseVLM.** SparseVLM derives visual relevance from language tokens associated with the image, progressively sparsifies the visual sequence, and recycles information from discarded tokens into retained representations (Zhang et al., 2025). Its adaptive schedule can produce a realized token count that differs from a single fixed post-cut budget.

**DART.** DART targets representation-space redundancy (Wen et al., 2025), comparing visual tokens with compact visual and textual pivot sets and favoring less-duplicated content.

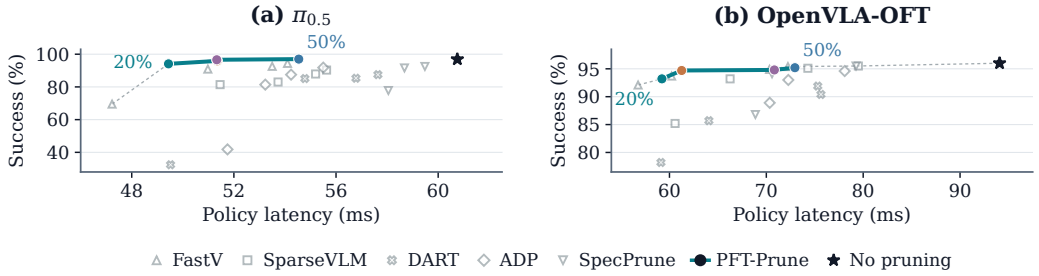



Figure 12: Measured success–latency trade-offs on LIBERO, using Table 1. Success is averaged over the four suites. The dashed line joins the empirical non-dominated configurations within each policy: no other measured configuration is at least as successful and at least as fast, with a strict improvement in either quantity. The teal line connects PFT-PRUNE settings in retention order. Filled-circle colors identify 20% (teal), 30% (orange), 40% (purple), and 50% (blue) retention; text labels mark 20% and 50%. Gray marker shapes identify the other pruning methods, and stars denote no pruning. Axis ranges differ between policies.

**ADP.** ADP combines instruction-conditioned visual relevance with an action-aware gate that adjusts pruning according to the policy trajectory (Pei et al., 2026). We report its aggressive-gate configuration, in which the amount of retained visual context varies across policy queries; consequently, its reported token usage is an execution average rather than a fixed retained-token count.

**SpecPrune-VLA.** SpecPrune-VLA couples decisions across policy queries with pruning across transformer depth (Wang et al., 2026). It uses prior action-level information together with current evidence to initialize selection, then refines token importance through a staged layer schedule controlled by robot motion. We abbreviate this VLA-specific method as "SpecPrune".

### G.2 PROTOCOL AND TOKEN ACCOUNTING

The simulation workstation has two NVIDIA GeForce RTX 4090 GPUs with 24 GB of memory each. Independent evaluation runs are assigned to separate GPUs; both policies use one GPU per run with batch size one.

Comparisons use the same nominal retention groups, tasks, and episode counts within each policy. Fixed-cut methods admit a direct retained-token count, whereas adaptive or layerwise methods may report average tokens, retention after dense warm-up, or layer-weighted token computation (Zhang et al., 2025; Pei et al., 2026; Wang et al., 2026). These quantities are not interchangeable with PFT-PRUNE's fixed physical post-cut count. We use nominal retention to organize results and evaluate efficiency through measured policy latency, with speedup relative to the corresponding unpruned policy. Raw latency is compared only within one architecture because $\pi_{0.5}$ and OpenVLA-OFT have different inference pipelines.

To compare methods across their differing token schedules, Figure 12 plots success against measured policy latency across all retention settings. PFT-PRUNE lies on the empirical frontier at three of four tested settings on $\pi_{0.5}$, namely 20%, 40%, and 50% retention, and at two of four on OpenVLA-OFT, namely 20% and 30%. The frontier covers all configurations per policy, not separate retention groups.

## H LIMITATIONS AND BROADER IMPACT

**Limitations.** PFT-Prune requires access to intermediate policy representations and their gradients, and therefore assumes a white-box inference setting rather than a black-box model interface. Fingerprint construction and token selection also introduce computation before compact execution, so the realized speedup depends on how much subsequent computation benefits from the shortened sequence. Finally, exact-budget fill can reintroduce previously suppressed candidates when the retained-anchor set is smaller than the requested budget; accordingly, the pairwise-separation

guarantee applies to retained anchors rather than every token in the final set. Our budget analysis shows that this effect becomes more pronounced at larger retention ratios.

**Broader Impact.** PFT-Prune reduces visual-token computation during VLA inference without additional training, potentially lowering the computation required for closed-loop robotic deployment. As with visual-token pruning more generally, aggressive retention settings trade computation for available visual context. Practical deployments should therefore choose the retention budget according to the required task-performance and latency trade-off.