FAVP: FUTURE-AWARE VISION TOKEN PRUNING FOR UNIFIED WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision token pruning can accelerate inference in unified World Action Models (WAMs), yet existing approaches primarily use semantic, temporal, or action-related cues without explicitly exploiting the future-predictive knowledge learned during world modeling. To bridge this gap, we propose **FAVP**, a training-free **F**uture-**A**ware **V**ision Token **P**runing framework that reuses this capability to guide vision token selection during action inference. FAVP introduces three modules: (1) **Future-Prediction Query Augmentation** to efficiently extract future-aware visual information without full WM inference, (2) **Future-Aware Attention Masking** to preserve the original action-generation process and to block the queries from directly accessing the original non-visual tokens, and (3) **Action-Future Token Selection** to jointly exploit action-aware and future-aware importance to prune redundant and unimportant tokens. Extensive experiments across WAM backbones (i.e., RynnVLA-002 and UniVLA) in both simulation and real-world manipulation tasks demonstrate that FAVP achieves up to a 1.63$\times$ inference speedup on simulation benchmarks and a 1.67 $\times$ speedup in real-world manipulation, with relatively small reductions in success rate.

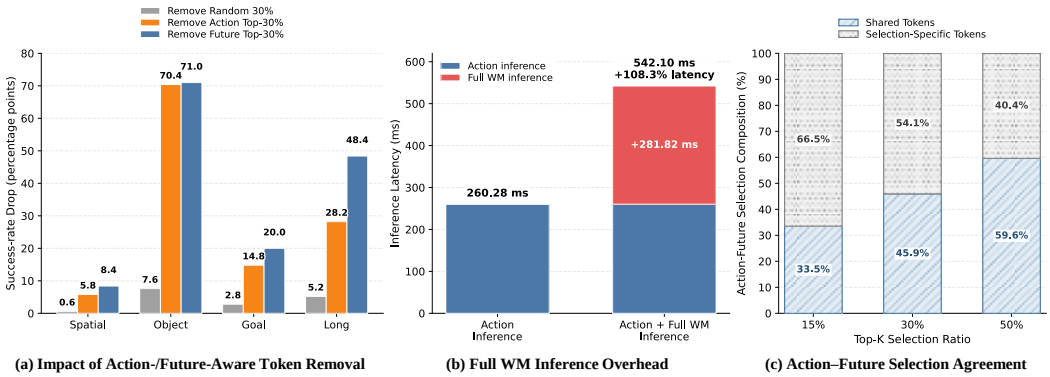
1 INTRODUCTION

Vision-Language-Action (VLA) models have emerged as a powerful paradigm for robotic manipulation by integrating visual perception, language understanding, and action generation within a unified policy (Zitkovich et al., 2023; Ghosh et al., 2024; Li et al., 2024a; Kim et al., 2025b;a; Black et al., 2025). Building on this progress, recent World Action Models (WAMs) further integrate predictive world modeling with action generation, jointly capturing future-state prediction and robot action generation within a unified framework (Cen et al., 2025b; Wang et al., 2026; Cen et al., 2025a; Yuan et al., 2026). Despite their richer predictive capabilities, WAMs suffer from substantial inference overhead in closed-loop control due to repeated processing of long multimodal sequences with hundreds or thousands of vision tokens (Li et al., 2026). Therefore, improving the inference efficiency of WAMs is critical for their practical deployment in real-time robotic systems.

Vision token pruning offers a promising way of reducing the inference cost by shortening the visual sequence processed by the Transformer backbone. Representative methods estimate token importance from attention scores (Chen et al., 2024), temporal redundancy (Xu et al., 2026), manipulation dynamics (Pei et al., 2026), semantic and action relevance (Liu et al., 2025), or semantic attention consistency (Ma et al., 2026). However, these approaches do not explicitly use signals from the WAM's future-prediction knowledge to guide vision token selection. Our preliminary analysis suggests that this knowledge provides useful pruning guidance. As shown in Fig. 1(a), removing the Top 30% of vision tokens ranked by future-aware attention causes substantially larger success-rate drops than random removal at the same ratio across all four LIBERO suites (Liu et al., 2023). This suggests that future-prediction knowledge can identify visual evidence relevant to action generation.

Exploiting future-aware information for vision token pruning presents two challenges. **First**, extracting future-aware information is constrained by the substantial computational cost of full WM inference. As shown in Fig. 1(b), performing the full WM pathway alongside action inference increases latency from 260.28 ms to 542.10 ms ($+108.3\%$) in the evaluated setting, potentially offsetting the computational savings from vision token pruning. This motivates a lightweight mechanism that extracts future-aware visual information with minimal additional computational overhead.

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.



(a) Impact of Action-/Future-Aware Token Removal **(b) Full WM Inference Overhead** **(c) Action–Future Selection Agreement**

Figure 1: Motivation and key challenges of future-aware vision token pruning in WAMs. (a) Removing action-aware or future-aware Top-30% tokens causes much larger success-rate degradation than random removal. (b) Full WM inference introduces substantial latency overhead. (c) Action-aware and future-aware Top-K selections contain both shared and selection-specific tokens across different retention ratios.

Second, incorporating future-predictive knowledge into vision token selection requires effectively coordinating it with the action-related information already available during policy inference. As shown in Fig. 1(c), the Top-K selections under action-aware and future-aware criteria contain both shared and selection-specific tokens across different retention ratios (i.e., shared tokens account for 33.5%, 45.9%, and 59.6% of the selections at selection ratios of 15%, 30%, and 50%, respectively). This motivates a selection strategy that exploits their agreement while preserving complementary information from their distinct preferences.

To address these challenges, we propose **FAVP**, a training-free and plug-and-play framework that exploits the future-predictive knowledge learned during WAM training for efficient vision token pruning. FAVP consists of three key modules. First, **Future-Prediction Query Augmentation** uses lightweight future-prediction queries during action inference to extract future-aware information instead of performing full WM inference. Second, **Future-Aware Attention Masking** decouples the action-generation and future-aware information streams, preserving the original action-generation process while preventing future-aware representations from being influenced by non-visual tokens in the original action-generation sequence. Third, **Action-Future Token Selection** jointly exploits action-aware and future-aware importance by retaining tokens emphasized under both estimates and completing the retained set with diversity-based selection. Overall, FAVP enables efficient WAM inference by effectively leveraging future-predictive knowledge to identify and preserve task-relevant visual information with reduced computational cost. We evaluate FAVP on RynnVLA-002 (Cen et al., 2025a) and UniVLA (Wang et al., 2026) using LIBERO (Liu et al., 2023), SimplerEnv (Li et al., 2024b), and real-world robotic manipulation tasks. On LIBERO, FAVP achieves up to a 1.63$\times$ inference speedup with average success-rate decreases of only 0.2 and 0.6 percentage points. In real-world manipulation, FAVP achieves 1.67$\times$ speedup with only a 1.1-percentage-point decrease in success rate.

Our contributions are summarized as follows:

- We propose **FAVP**, a training-free framework that explicitly reuses the future-prediction capability of unified WAMs to guide vision token pruning during action inference, without generating future observations.
- We introduce lightweight query augmentation and structured attention masking to obtain future-aware importance, together with an action–future selection strategy that combines shared high-priority tokens with diverse candidates under a fixed retention budget.
- We validate FAVP in both simulation and real-world robotic manipulation, demonstrating strong manipulation performance with small average success-rate reductions across two WAM backbones, simulation benchmarks, and real-world manipulation tasks.

2 Related Work

2.1 Vision–Language–Action Models

Vision–Language–Action models leverage pretrained vision–language models to map multimodal observations and language instructions to executable robot actions (Kawaharazuka et al., 2025). Existing methods explore different action representations and generation mechanisms. RT-2 (Zitkovich et al., 2023) discretizes continuous robot actions into tokens and generates them autoregressively, while OpenVLA (Kim et al., 2025b) adopts an open-source autoregressive VLA formulation based on a pretrained VLM. CogACT (Li et al., 2024a) employs a diffusion-based action transformer conditioned on VLM representations, whereas π_0 (Black et al., 2025) uses a flow-matching action expert for continuous control. OpenVLA-OFT (Kim et al., 2025a) further improves inference and control efficiency through action chunking and parallel action prediction. Despite their different action-generation mechanisms, these VLA models primarily predict actions from current observations and instructions without explicitly predicting future states. By contrast, more recent work integrates future-state prediction with action generation in unified World Action Models (WAMs).

2.2 World Models and Unified World Action Models

World Models (WMs) learn predictive representations of environment dynamics by modeling future states or visual observations conditioned on the current observation and robot actions (Ha & Schmidhuber, 2018; Hafner et al., 2020). Early approaches typically use world models as standalone predictive components for planning or policy learning. Visual foresight methods predict future observations to support action selection (Finn & Levine, 2017), while Dreamer (Hafner et al., 2020) learns latent dynamics and optimizes policies through imagined trajectories. More recent work increasingly integrates future prediction and action generation within unified model architectures. WorldVLA (Cen et al., 2025b) integrates autoregressive action generation with future visual prediction, while UniVLA (Wang et al., 2026) incorporates world-model post-training to improve downstream policy learning. RynnVLA-002 (Cen et al., 2025a) unifies VLA policy learning and future visual-state prediction within a shared transformer backbone. Despite their different formulations, these unified WAMs allow future-prediction learning to influence representations used for action generation. FAVP reuses this learned knowledge to guide vision token pruning during action inference.

2.3 Efficient VLA Inference

The high inference cost of VLA models has motivated growing efforts toward efficient deployment, including representation reuse, model compression, and vision token pruning (Yu et al., 2025). Training-free approaches aim to improve inference efficiency without requiring additional model training. FastV (Chen et al., 2024) removes redundant vision tokens according to attention-based importance, while VLA-Cache (Xu et al., 2026) exploits temporal redundancy by reusing visual representations across consecutive observations. VLA-ADP (Pei et al., 2026) dynamically adjusts pruning according to robot motion stages, and VLA-Pruner (Liu et al., 2025) incorporates both semantic and action-related information for token selection. SAFE-Pruner (Ma et al., 2026) leverages semantic attention consistency to anticipate token importance in subsequent Transformer layers. Despite their effectiveness, existing methods rely primarily on semantic, temporal, or action-related signals available during conventional VLA inference, risking the removal of vision tokens important for both future-state prediction and action generation. Instead, FAVP addresses this limitation by using the future-predictive knowledge learned by WAMs to guide token selection.

3 Preliminaries

VLA models and WMs capture two complementary aspects of robotic interaction. Given the current observation o and language instruction l, a VLA model predicts the robot action a as

$$p(a \mid o, l). \tag{1}$$

In contrast, a WM predicts the future observation or state o' conditioned on the current observation, action, and language instruction:

$$p(o' \mid o, a, l). \tag{2}$$

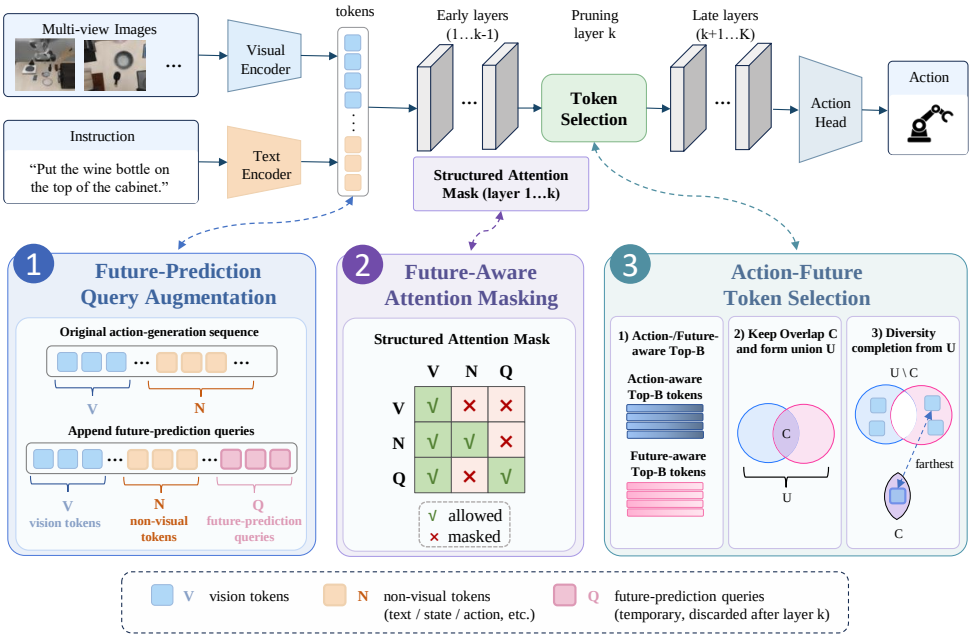


Figure 2: **Overview of FAVP and its three modules**. (1) *Future-Prediction Query Augmentation* appends lightweight future-prediction queries to the action-generation sequence. (2) *Future-Aware Attention Masking* preserves the original attention pattern while restricting these queries to vision and query tokens. (3) *Action–Future Token Selection* retains the overlap of action-aware and future-aware top-B sets, then fills the remaining budget with diverse tokens from their union. After pruning, the auxiliary queries are discarded, and the retained sequence proceeds to action generation.

Unified WAMs combine action generation and future prediction within a unified formulation:

$$p(o', a \mid o, l). \tag{3}$$

In this work, we refer to WAMs that integrate action generation and future prediction through a shared backbone as *unified WAMs*. These models share the Transformer backbone across the VLA and WM tasks while differing in their task-specific inputs and prediction targets. The VLA task focuses on action generation, whereas the WM task predicts future observations or states. For jointly trained WAMs, the training objective can generally be expressed as (Shen et al., 2026)

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{future}} + \lambda \mathcal{L}_{\text{action}}, \tag{4}$$

where $\mathcal{L}_{\text{future}}$ and $\mathcal{L}_{\text{action}}$ denote the future-prediction and action-generation objectives, respectively, and λ balances the two objectives. Such multi-task training enables the shared backbone to acquire knowledge relevant to both action generation and future-state prediction.

During inference, unified WAMs typically perform only action generation, while explicit future prediction is omitted. Although the shared backbone has acquired future-predictive knowledge during training, this knowledge is not explicitly exploited for vision token selection. These observations, together with the motivation presented in Fig. 1, inspire us to reuse the learned future-prediction capability for efficient vision token pruning.

4 FAVP

As illustrated in Fig. 2, FAVP leverages the future-predictive knowledge learned by the WM task to guide vision token pruning during action generation. Instead of performing full WM inference, FAVP appends a lightweight set of future-prediction queries to the original action-generation sequence and processes the augmented sequence through the first k Transformer layers. This allows

the model to obtain complementary action-aware and future-aware information for evaluating vision token importance with only limited additional computation. As further analyzed in Table 6, the additional components introduce only 2.87% latency overhead before token pruning.

FAVP consists of three main components. First, *Future-Prediction Query Augmentation* reuses queries from the WAM training to extract future-aware information during action generation. Second, *Future-Aware Attention Masking* is applied to the early Transformer layers to preserve the original action-generation process while preventing the future-prediction queries from accessing non-visual context from the original sequence. Third, *Action–Future Token Selection* first prioritizes vision tokens emphasized by both action-aware and future-aware importance, and then complements the retained set with diverse tokens selected from their union. After token selection, the unselected vision tokens and all future-prediction queries are discarded, and only the compressed action-generation sequence is forwarded through the remaining Transformer layers for action generation. The complete inference procedure of FAVP is provided in Algorithm 1 in Appendix A.1.

4.1 Future-Prediction Query Augmentation

A straightforward way to exploit future-predictive knowledge is to perform full WM inference, but this would introduce non-negligible additional computation. Instead, FAVP reuses the WAM's native future-prediction queries, which are used during WAM training to support future-state prediction. By appending these lightweight queries to the original action-generation sequence, FAVP leverages the future-predictive knowledge learned by the shared backbone to extract future-aware visual information without explicitly generating future observations.

We denote the future-prediction queries as $\mathcal{Q}_{t,\mathrm{fut}} = \{q_{1,\mathrm{fut}}, q_{2,\mathrm{fut}}, \ldots, q_{Q,\mathrm{fut}}\}$, where Q denotes the number of appended queries. These queries are appended to the original sequence and used to estimate future-aware vision token importance for subsequent token selection.

Let $X_{t,\mathrm{act}}$ denote the original action-generation input sequence, which contains the task-related non-visual tokens and the vision token set $\mathcal{V}_t$. We append the future-prediction queries to the original sequence to form the augmented sequence

$$\tilde{X}_t = [X_{t,\mathrm{act}}; \mathcal{Q}_{t,\mathrm{fut}}]. \tag{5}$$

After embedding, the augmented sequence is represented as

$$H_t^0 = \left[H_{t,\mathrm{act}}^0; H_{t,\mathrm{fut}}^0\right], \tag{6}$$

where $H_{t,\mathrm{act}}^0$ and $H_{t,\mathrm{fut}}^0$ correspond to the embeddings of original action-generation tokens and appended future-prediction queries, respectively. The augmented representation is processed by the first $k-1$ layers of the Transformer backbone:

$$H_t^\ell = F_\theta^\ell\left(H_t^{\ell-1}; \mathcal{M}\right), \qquad \ell = 1, \ldots, k-1, \tag{7}$$

where F_θ^ℓ denotes the ℓ-th Transformer layer and $\mathcal{M}$ is the structured attention mask detailed in the following Sec. 4.2.

At the pruning layer k, we obtain

$$H_t^k = F_\theta^k\left(H_t^{k-1}; \mathcal{M}\right) = \left[H_{t,\mathrm{act}}^k; H_{t,\mathrm{fut}}^k\right]. \tag{8}$$

A key advantage of the proposed query augmentation is that the original action-generation tokens and the appended future-prediction queries are processed within the same Transformer backbone. The two pathways therefore evaluate the same set of vision tokens at the pruning layer, providing a common basis for jointly considering action-aware and future-aware importance.

4.2 Future-Aware Attention Masking

After augmenting the input sequence with future-prediction queries, a key challenge is how to extract future-aware visual information without disturbing the original action-generation process. As shown in Fig. 3(a), a standard causal mask would allow the appended future-prediction queries to attend to all preceding action-generation tokens, including language, state, and action-related tokens. Consequently, the extracted future-aware information may incorporate non-visual context rather than

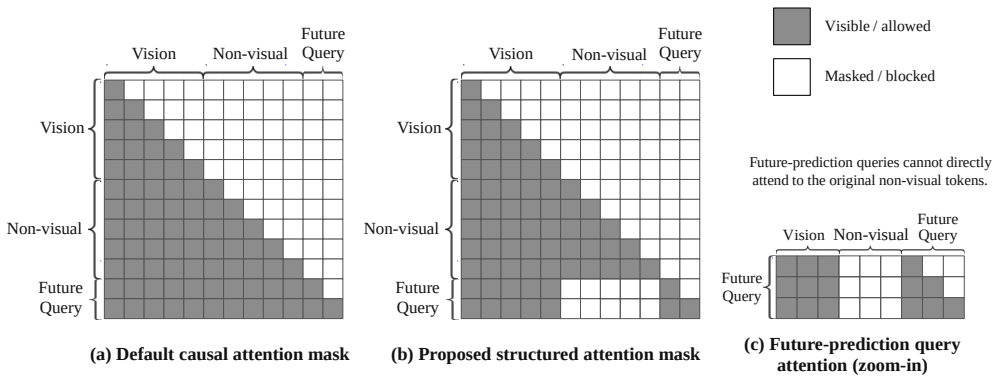


Figure 3: Illustration of the structured attention mask in FAVP. (a) Default causal attention after future-query augmentation. (b) The proposed structured mask preserves the original action-generation process while preventing future-prediction queries from directly attending to the original non-visual tokens. (c) Future-prediction queries attend to vision and query tokens, while original non-visual tokens are masked.

primarily reflecting the shared visual observation. To address this issue, we introduce a structured attention mask to control the information flow between the original action-generation pathway and the future-prediction queries, as illustrated in Fig. 3(b).

The masking mechanism is designed with two objectives. First, the original action-generation information flow should remain unaffected by the appended future-prediction queries, thereby preserving the original action generation process. Second, the future-prediction queries should focus on visual information relevant to future-state prediction, rather than directly attending to non-visual tokens in the original action-generation sequence. By preserving the original action-generation process and reusing the WAM's native future-prediction queries, FAVP can be integrated into unified WAMs without modifying model parameters or requiring additional training.

To realize these design principles, we construct a structured attention mask $\mathcal{M}$ over the augmented sequence. For each of the first k Transformer layers, the attention weights of the h-th attention head are computed as

$$A_h^\ell = \mathrm{Softmax}\left(\frac{Q_h^\ell (K_h^\ell)^\top}{\sqrt{d_h}} + \mathcal{M}\right), \qquad \ell = 1, \ldots, k, \tag{9}$$

where $Q_h^\ell, K_h^\ell \in \mathbb{R}^{L \times d_h}$ denote the query and key representations of the h-th attention head at layer ℓ, d_h is the head dimension, and L is the length of the augmented sequence. Collecting the attention weights from all H heads gives $A^\ell \in \mathbb{R}^{H \times L \times L}$. In addition, $\mathcal{M}_{ij} = 0$ indicates that query token i can attend to key/value token j, while $\mathcal{M}_{ij} = -\infty$ denotes a masked attention connection.

According to the original action-generation tokens and the appended future-prediction queries, the structured mask is defined as

$$\mathcal{M} = \begin{bmatrix} \mathcal{M}_{\mathrm{act}} & -\infty \\ \mathcal{M}_{\mathrm{fut}\to\mathrm{act}} & \mathcal{M}_{\mathrm{fut}} \end{bmatrix}, \tag{10}$$

where the row groups correspond to query tokens and the column groups correspond to key/value tokens. The upper-left block $\mathcal{M}_{\mathrm{act}}$ follows the original causal attention mask, while the upper-right block $-\infty$ is fully masked to prevent the original action-generation tokens from attending to the appended future-prediction queries. The lower-right block $\mathcal{M}_{\mathrm{fut}}$ preserves the native attention pattern among the future-prediction queries.

For the interaction between the future-prediction queries and the original action-generation sequence, we restrict their access to the shared vision tokens:

$$(\mathcal{M}_{\mathrm{fut}\to\mathrm{act}})_{ij} = \begin{cases} 0, & j \in \mathcal{V}_t, \\ -\infty, & j \in X_t^{\mathrm{act}} \setminus \mathcal{V}_t. \end{cases} \tag{11}$$

As illustrated in Fig. 3(c), the future-prediction queries can attend to vision tokens while language, state, and action-related tokens in the original sequence remain inaccessible.

With this structured attention mask, the original action-generation process remains unaffected by the auxiliary future-prediction queries, while the queries selectively extract future-aware information from the shared visual observation. This provides complementary action-aware and future-aware visual information for the token selection strategy introduced in the following subsection.

4.3 Action–Future Token Selection

With the structured attention mask, the original action-generation pathway and the future-prediction queries provide complementary action-aware and future-aware information over the shared vision tokens. Based on the attention weights defined in Eq. 9, let $\mathcal{Q}_{t,\mathrm{act}}$ denote the set of action-related token positions used to compute action-aware importance, while $\mathcal{Q}_{t,\mathrm{fut}}$ denotes the future-prediction queries introduced in Sec. 4.1. For the i-th vision token, the two importance scores are computed as

$$s_i^{\mathrm{act}} = \frac{1}{H|\mathcal{Q}_{t,\mathrm{act}}|}\sum_{h=1}^{H}\sum_{q\in\mathcal{Q}_{t,\mathrm{act}}} A_{h,q,i}^{k}, \qquad s_i^{\mathrm{fut}} = \frac{1}{H|\mathcal{Q}_{t,\mathrm{fut}}|}\sum_{h=1}^{H}\sum_{q\in\mathcal{Q}_{t,\mathrm{fut}}} A_{h,q,i}^{k}. \tag{12}$$

Based on the two scores, we independently select the top-B vision tokens:

$$\mathcal{T}^{\mathrm{act}} = \mathrm{TopK}\left(\{s_i^{\mathrm{act}}\}, B\right), \qquad \mathcal{T}^{\mathrm{fut}} = \mathrm{TopK}\left(\{s_i^{\mathrm{fut}}\}, B\right), \tag{13}$$

where B denotes the number of vision tokens to be retained. Rather than directly combining the two scores, which may require additional weighting hyperparameters (Ma et al., 2026), FAVP prioritizes vision tokens selected by both criteria and retains their intersection $\mathcal{C} = \mathcal{T}^{\mathrm{act}} \cap \mathcal{T}^{\mathrm{fut}}$.

As shown in Fig. 1(c), the two Top-B sets only partially overlap, so the intersection $\mathcal{C}$ generally contains fewer than B tokens. We therefore construct a candidate set from their union $\mathcal{U} = \mathcal{T}^{\mathrm{act}} \cup \mathcal{T}^{\mathrm{fut}}$. We initialize the retained set as $\mathcal{S} = \mathcal{C}$; if $\mathcal{C}$ is empty, $\mathcal{S}$ is initialized with the token in $\mathcal{U}$ having the largest summed action-aware and future-aware importance score. We then complete the retained set and enhance its diversity by selecting tokens from $\mathcal{U} \setminus \mathcal{S}$. Let $\mathbf{e}_i$ denote the representation of the i-th vision token at the pruning layer k. Following diversity-based vision token pruning (Alvar et al., 2025), we measure feature dissimilarity using cosine distance:

$$d(i,j) = 1 - \frac{\mathbf{e}_i^{\top}\mathbf{e}_j}{\|\mathbf{e}_i\|_2\|\mathbf{e}_j\|_2}. \tag{14}$$

To efficiently complete the retained set, we adopt a greedy max–min selection strategy over the candidate set $\mathcal{U}$. At each step, we select the candidate token whose minimum distance to the current retained set is the largest:

$$i^* = \arg\max_{i\in\mathcal{U}\setminus\mathcal{S}} \min_{j\in\mathcal{S}} d(i,j). \tag{15}$$

The retained set is then updated as

$$\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}, \tag{16}$$

and the process continues until $|\mathcal{S}| = B$. Finally, all unselected vision tokens and future-prediction queries are discarded, and the compressed action-generation sequence is forwarded through the remaining Transformer layers for action generation.

5 Experiments

5.1 Experimental Setup

Backbone Models. We implement FAVP on two unified WAM backbones. RynnVLA-002 (Cen et al., 2025a) unifies VLA policy learning and world modeling within a shared architecture, enabling joint modeling of action generation and future visual-state prediction. UniVLA (Wang et al., 2026) represents vision, language, and action as discrete tokens in a unified autoregressive Transformer and incorporates world-model post-training before downstream policy fine-tuning.

Baselines. We compare FAVP with representative training-free VLA acceleration methods, including FastV (Chen et al., 2024), VLA-Cache (Xu et al., 2026), VLA-ADP (Pei et al., 2026), and VLA-Pruner (Liu et al., 2025). These baselines cover different acceleration paradigms, including general vision token pruning, temporal token caching, action-aware dynamic pruning, and VLA-specific token pruning.

Table 1: Performance comparison on the LIBERO benchmark. **Bold** indicates the best result among acceleration methods, and underlined indicates the second-best result.

Model	Method	Success Rate (%) ↑					FLOPs (T)↓	Latency Speedup ↑
		Spatial	Object	Goal	Long	Average		
UniVLA	Vanilla	95.6	98.6	92.8	90.6	94.4	8.03	1.00×
	VLA-Cache	95.0	97.8	93.0	86.0	93.0	5.66	1.13×
	FastV	93.8	97.4	91.2	87.8	92.6	5.38	1.29×
	VLA-ADP	92.4	94.8	84.4	82.6	88.6	6.49	1.16×
	VLA-Pruner	94.6	98.0	92.6	**89.4**	93.7	4.75	1.46×
	FAVP	**95.8**	**98.2**	**93.4**	89.2	**94.2**	4.40	1.51×
RynnVLA-002	Vanilla	98.6	99.0	96.8	93.6	97.0	6.06	1.00×
	VLA-Cache	93.8	96.8	87.2	88.8	91.7	5.58	1.04×
	FastV	97.0	97.0	90.4	91.8	94.1	4.07	1.30×
	VLA-ADP	98.4	96.6	**94.4**	86.6	94.0	4.91	1.19×
	VLA-Pruner	97.4	97.0	93.6	93.0	95.3	3.58	1.61×
	FAVP	**99.0**	**99.4**	93.2	**93.8**	**96.4**	3.32	1.63×

Table 2: Performance comparison on the SimplerEnv-WidowX benchmark with UniVLA. **Bold** indicates the best result among acceleration methods, and underlined indicates the second-best result.

Method	Success Rate (%) ↑					FLOPs (T) ↓	Latency Speedup ↑
	Spoon	Carrot	Stack	Eggplant	Average		
Vanilla	75.0	66.7	62.5	100	76.0	13.66	1.00×
VLA-Cache	70.8	58.3	58.3	95.8	70.8	9.28	1.14×
FastV	66.7	58.3	54.2	91.7	67.7	9.71	1.21×
VLA-ADP	**79.2**	62.5	54.2	95.8	72.9	10.36	1.15×
VLA-Pruner	70.8	**66.7**	58.3	**100**	74.0	8.01	1.43×
FAVP	75.0	**66.7**	**62.5**	95.8	**75.0**	7.85	1.47×

Benchmarks and Evaluation Metrics. We evaluate FAVP on LIBERO (Liu et al., 2023) and SimplerEnv (Li et al., 2024b). LIBERO contains diverse language-conditioned manipulation tasks, and we report results on LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. SimplerEnv provides reproducible simulated evaluation environments for policies trained on real-world robot data, and we evaluate UniVLA on the WidowX manipulation tasks. We use task success rate to measure manipulation performance. Computational efficiency is evaluated using theoretical FLOPs and inference latency speedup relative to the corresponding full-token backbone. The detailed FLOPs formulation is provided in Appendix A.2. Additional implementation details, including model configurations, pruning settings, evaluation protocols, and latency measurement, are provided in Appendix A.3.

5.2 Main Results

Results on LIBERO. Table 1 reports the LIBERO results for UniVLA and RynnVLA-002. FAVP achieves success rates of 94.2% and 96.4% on average, respectively, only 0.2 and 0.6 percentage points below their full-token counterparts. FAVP reduces FLOPs from 8.03T to 4.40T on UniVLA and from 6.06T to 3.32T on RynnVLA-002, yielding $1.51\times$ and $1.63\times$ inference speedups, respectively. Compared with VLA-Pruner, FAVP improves average success rate by 0.5 and 1.1 percentage points on two backbones, respectively, while using fewer FLOPs and achieving higher speedups. These results show that FAVP improves inference efficiency with small success-rate reductions relative to the full-token models on both backbones.

Results on SimplerEnv. Table 2 evaluates FAVP on SimplerEnv benchmark with UniVLA. FAVP achieves 75.0% success rate on average, compared with 76.0% for the full-token model, while reducing FLOPs from 13.66T to 7.85T and providing a $1.47\times$ inference speedup. Compared with

Table 3: Real-world robotic manipulation results.

Method	Success Rate (%) ↑				Latency Speedup ↑
	Task 1	Task 2	Task 3	Average	
RynnVLA-002	93.3	90.0	83.3	88.9	1.00×
FAVP	96.7	86.7	80.0	87.8	1.67×

Table 4: Ablation study of FAVP on LIBERO with RynnVLA-002.

Variant	**Success Rate (%) ↑**					**Latency Speedup ↑**
	Spatial	**Object**	**Goal**	**Long**	**Average**	
FAVP	**99.0**	**99.4**	**93.2**	**93.8**	**96.4**	1.63×
w/o Future-Prediction Queries	97.4	97.2	90.4	91.0	94.0	1.67×
w/o Future-Aware Masking	98.8	98.4	92.6	92.0	95.5	1.64×
w/o Action–Future Selection	98.6	99.0	91.8	90.2	94.9	1.66×

VLA-Pruner, FAVP improves average success by 1.0 percentage point while using fewer FLOPs and achieving a higher speedup. FAVP matches the full-token model on Spoon, Carrot, and Stack, and the success rate slightly decreases from 100.0% to 95.8% on Eggplant.

5.3 Real-World Validation

We evaluate FAVP on a LeRobot SO-101 robotic arm (Cadene et al., 2026) equipped with wrist-view and third-person-view cameras. For each of three manipulation tasks, we collect 100 demonstration trajectories to train a task-specific policy. The experimental setup and evaluation protocol are detailed in Appendix A.5.

As shown in Table 3, FAVP achieves a success rate of 87.8% on average, compared with 88.9% for the full-token model, while providing a 1.67× inference speedup. Thus, FAVP reduces inference latency with only a 1.1-percentage-point decrease in average success rate in this real-world evaluation.

5.4 Ablation Study

Table 4 evaluates the three FAVP modules on LIBERO using RynnVLA-002. Removing Future-Prediction Query Augmentation, leaving only action-aware importance, reduces average success from 96.4% to 94.0%. Replacing Future-Aware Attention Masking with the default causal mask reduces it to 95.5%. Replacing Action–Future Token Selection with summed importance scores for Top-B selection reduces it to 94.9%. These ablations support the contributions of future-aware information, structured attention, and joint token selection. Meanwhile, all variants achieve similar latency speedups of 1.64×–1.67×, indicating that the three modules introduce only limited runtime overhead. Table 5 further shows that future-prediction queries outperform zero and random queries.

6 Conclusion

We proposed FAVP, a training-free vision token pruning framework that exploits the future-predictive knowledge learned by unified WAMs to improve action-inference efficiency. By combining lightweight future-prediction queries, structured attention masking, and action–future token selection, FAVP reduces inference cost without full WM inference. Experiments on LIBERO and SimplerEnv demonstrate up to a 1.63× inference speedup while preserving manipulation performance. We further evaluate FAVP on real-world robotic manipulation tasks, highlighting its potential for efficient WAM deployment.

AI USE STATEMENT

In this work, we used generative AI tools to assist with scientific writing and language editing, as well as literature retrieval and discovery. We did not use generative AI tools to generate synthetic datasets or prove mathematical claims. All AI-assisted writing was reviewed and revised by the authors, and relevant literature was checked against the original sources. We take full responsibility for the final content of this work, including its text, claims, and results.

REFERENCES

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010.

Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, et al. Lerobot: An open-source library for end-to-end robot learning. In *International Conference on Learning Representations*, volume 2026, pp. 122398–122417, 2026.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, et al. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025b.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 13:162467–162504, 2025. doi: 10.1109/ACCESS.2025.3609980.

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*, LosAngeles, 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*, pp. 2679–2713, 2025b.

Jiajun Li, Tiecheng Guo, Yifan Ye, Rongyu Zhang, Xiaowei Chi, Qianpu Sun, Ying Li, Yunfan Lou, Yan Huang, Zhihe Lu, et al. Efficient-wam: A 1b-parameter world-action model with low-cost future imagination. *arXiv preprint arXiv:2606.10040*, 2026.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Ho Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning*, 2024b. URL https://api.semanticscholar.org/CorpusID:269635843.

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

Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. Vla-pruner: Temporal-aware dual-level visual token pruning for efficient vision-language-action inference. *arXiv preprint arXiv:2511.16449*, 2025.

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*, pp. 163–180. Springer, 2026.

Xiaohuan Pei, Yuxing Chen, Siyu Xu, Yunke Wang, Yuheng Shi, and Chang Xu. Action-aware dynamic pruning for efficient vision-language-action manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Qiuhong Shen, Shihua Zhang, Yue Liao, Qi Li, Zhenxiong Tan, Shizun Wang, Shuicheng Yan, and Xinchao Wang. World action models: A survey. *arXiv preprint arXiv:2606.20781*, 2026.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

Zhaoshu Yu, Bo Wang, Pengpeng Zeng, Haonan Zhang, Ji Zhang, Zheng Wang, Lianli Gao, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. A survey on efficient vision-language-action models. *arXiv preprint arXiv:2510.24795*, 2025.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

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*, pp. 2165–2183. PMLR, 2023.

A APPENDIX

A.1 FAVP INFERENCE ALGORITHM

Algorithm 1 Overall Inference Pipeline of FAVP

Input: original action-generation input sequence $X_{t,\text{act}}$, vision token set $\mathcal{V}_t$, future-prediction queries $\mathcal{Q}_{t,\text{fut}}$, pruning layer k, total number of Transformer layers K, and retained vision token number B

Output: predicted robot action a_t

$\tilde{X}_t \leftarrow [X_{t,\text{act}}; \mathcal{Q}_{t,\text{fut}}]$ # Future-prediction query augmentation

Construct the structured attention mask $\mathcal{M}$ # Preserve action generation and constrain future-aware querying

$H_t^0 \leftarrow \text{Embed}(\tilde{X}_t)$

for $\ell = 1$ to $k-1$ **do**

  $H_t^\ell \leftarrow F_\theta^\ell(H_t^{\ell-1}; \mathcal{M})$

end for

$(H_t^k, A^k) \leftarrow F_\theta^k(H_t^{k-1}; \mathcal{M})$ # Pruning layer

Compute $\{s_i^{\text{act}}\}$ from A^k using $\mathcal{Q}_{t,\text{act}}$

Compute $\{s_i^{\text{fut}}\}$ from A^k using $\mathcal{Q}_{t,\text{fut}}$

$\mathcal{T}^{\text{act}} \leftarrow \text{TopK}(\{s_i^{\text{act}}\}, B)$

$\mathcal{T}^{\text{fut}} \leftarrow \text{TopK}(\{s_i^{\text{fut}}\}, B)$

$\mathcal{C} \leftarrow \mathcal{T}^{\text{act}} \cap \mathcal{T}^{\text{fut}}$ # Tokens important under both scores

$\mathcal{U} \leftarrow \mathcal{T}^{\text{act}} \cup \mathcal{T}^{\text{fut}}$ # Candidate set for diversity completion

$\mathcal{S} \leftarrow \mathcal{C}$

if $\mathcal{S} = \emptyset$ **then**

  $i^* \leftarrow \arg\max_{i \in \mathcal{U}} \left(s_i^{\text{act}} + s_i^{\text{fut}}\right)$ # Initialize from the strongest joint importance

  $\mathcal{S} \leftarrow \{i^*\}$

end if

while $|\mathcal{S}| < B$ **do**

  $i^* \leftarrow \arg\max_{i \in \mathcal{U} \setminus \mathcal{S}} \min_{j \in \mathcal{S}} d(i,j)$. # Greedy max–min diversity selection

  $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}$

end while

Discard all future-prediction query hidden states

$\hat{H}_t^k \leftarrow \text{Compress}(H_{t,\text{act}}^k, \mathcal{S})$ # Keep selected vision tokens and all non-vision tokens

for $\ell = k+1$ to K **do**

  $\hat{H}_t^\ell \leftarrow F_\theta^\ell(\hat{H}_t^{\ell-1})$

end for

$a_t \leftarrow \text{ActionHead}(\hat{H}_t^K)$

return a_t

A.2 COMPUTATIONAL COMPLEXITY ANALYSIS

We analyze the computational complexity of FAVP by considering the FLOPs of the shared Transformer backbone, where vision token pruning directly reduces the sequence length processed by subsequent layers. The computational costs of the vision encoder and action prediction head are omitted because they remain unchanged before and after pruning. We conservatively treat the masked attention computation as dense, since the structured attention mask does not reduce the tensor dimensions in standard Transformer implementations.

Let K denote the total number of Transformer layers and k denote the pruning layer. We use N to represent the number of text and other non-vision tokens, and M to represent the number of vision tokens, where $M \gg N$. The full sequence length before pruning is therefore

$$L_{\text{full}} = N + M.$$

For a Transformer layer with sequence length L, hidden dimension d, and FFN intermediate dimension d_{ff}, its FLOPs can be approximated as

$$\mathcal{C}(L) = 4Ld^2 + 2L^2d + 2Ldd_{\text{ff}},$$

where the three terms correspond to the linear projections in multi-head attention, the attention computation, and the feed-forward network, respectively.

Without token pruning, all K layers process the complete sequence, resulting in

$$\mathcal{F}_{\text{full}} = K\mathcal{C}(N + M).$$

FAVP additionally appends Q lightweight future-prediction queries up to and including the pruning layer k. Although $Q \ll M$ in practice, we explicitly account for the computation introduced by these queries. After obtaining the representations and attention weights at layer k, FAVP removes a fraction a of the vision tokens, where $a \in [0, 1]$ denotes the pruning ratio. The retained fraction of vision tokens is therefore $1 - a$, and the sequence length processed by the remaining layers becomes

$$L_{\text{pruned}} = N + (1 - a)M.$$

All future-prediction queries are discarded together with the unselected vision tokens after token selection. The overall Transformer FLOPs of FAVP can therefore be expressed as

$$\mathcal{F}_{\text{FAVP}} = k\mathcal{C}(N + M + Q) + (K - k)\mathcal{C}\big(N + (1 - a)M\big).$$

Accordingly, the theoretical FLOPs ratio relative to full-token inference is

$$R_{\text{FLOPs}} = \frac{k\mathcal{C}(N + M + Q) + (K - k)\mathcal{C}\big(N + (1 - a)M\big)}{K\mathcal{C}(N + M)},$$

and the corresponding theoretical FLOPs speedup is

$$S_{\text{FLOPs}} = \frac{\mathcal{F}_{\text{full}}}{\mathcal{F}_{\text{FAVP}}} = \frac{1}{R_{\text{FLOPs}}}.$$

Since all future-prediction queries are discarded after layer k and $M \gg N$, the relative sequence length in the post-pruning layers can be approximated as

$$\frac{L_{\text{pruned}}}{L_{\text{full}}} = \frac{N + (1 - a)M}{N + M} \approx 1 - a.$$

Therefore, in the layers after k, the quadratic attention term is approximately reduced by a factor of $(1 - a)^2$, while the linear projection and FFN terms are reduced by approximately $(1 - a)$. A larger pruning ratio and an earlier pruning layer generally provide greater computational savings.

A.3 Implementation Details

Backbone and Hardware. For RynnVLA-002, we use the officially released LIBERO checkpoints for the Spatial, Object, Goal, and Long suites. All LIBERO evaluations with RynnVLA-002 are conducted on a single NVIDIA RTX 3090 GPU. For UniVLA, we use the released pretrained model and evaluate it on LIBERO and SIMPLER-WidowX using a single NVIDIA RTX 5090 GPU with 32 GB memory. All methods are evaluated under the same precision and inference configuration for each backbone.

Method Configuration. For a fair comparison, FAVP and all baselines are evaluated under a consistent evaluation protocol. FAVP and all baselines except VLA-Cache perform token pruning at the third Transformer layer. For VLA-Cache, we follow the configuration recommended by its official repository and progressively prune tokens across multiple layers. We evaluate pruning ratios ranging from 0.3 to 0.5 for each method and select the final configuration by jointly considering task success rate and latency speedup. The final pruning ratios are 30% for VLA-Cache, 35% for FastV, 50% for VLA-ADP, 45% for VLA-Pruner, and 50% for FAVP. All other evaluation settings, including the pretrained backbone, input configuration, hardware platform, numerical precision, and evaluation protocol, are kept identical for methods compared on the same backbone.

Future-Prediction Query Configuration. FAVP does not introduce additional learnable query embeddings, but instead reuses the native future-prediction interface of each pretrained WAM to construct lightweight future-prediction queries. For RynnVLA-002, we reuse its native world-modeling interface and construct a compact future-prediction suffix from the model-specific context, resulting in $Q = 27$ additional tokens. This increases the sequence length from 1,144 to 1,171 tokens under the LIBERO-Spatial setting, accounting for only 2.31% of the complete sequence. For UniVLA, the future-prediction query encodes the future-frame target and visual-token grid information using only $Q = 9$ additional tokens, accounting for less than 0.43% of the approximately 2.1K-token sequence. In both backbones, these queries are used only up to the pruning layer to estimate future-aware visual token importance; no future observation is explicitly decoded, and all query hidden states are discarded immediately after token selection.

Evaluation Protocol. For LIBERO, we follow the official evaluation protocol and use the recommended number of evaluation episodes. Each task suite contains 10 manipulation tasks, and each task is evaluated over 50 episodes. Success rates are reported separately for LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long, with the overall performance obtained by averaging across the four suites. For SimplerEnv-WidowX, we also follow the official evaluation setup and use the recommended evaluation budget, where each of the four manipulation tasks is evaluated over 24 episodes. All methods are evaluated using identical task configurations and rollout settings within each benchmark.

Latency Measurement. We measure wall-clock inference latency with a batch size of 1. Timing is implemented using `time.perf_counter`, with `torch.cuda.synchronize()` called immediately before and after each timed inference step to ensure accurate CUDA timing. The reported latency includes the additional computation introduced by token scoring and selection, as well as the future-prediction queries in FAVP. We report the average latency over multiple inference steps and compute the speedup relative to the corresponding full-token backbone.

A.4 Benchmark Details

LIBERO. LIBERO (Liu et al., 2023) is a Robosuite-based benchmark for lifelong robot manipulation that evaluates language-conditioned policy learning across diverse manipulation scenarios. LIBERO comprises 130 language-conditioned manipulation tasks in total. We evaluate FAVP on four 10-task suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. LIBERO-Spatial focuses on spatial reasoning by varying the spatial configurations of objects while keeping the manipulation objective consistent. LIBERO-Object evaluates object generalization by changing the manipulated objects while preserving the scene layout and task structure. LIBERO-Goal studies goal-conditioned manipulation by varying task objectives under similar object configurations. LIBERO-Long contains long-horizon manipulation tasks involving multiple sequential operations and requires policies to execute complex instruction chains. Together, these suites evaluate different aspects of robotic manipulation, including spatial understanding, object interaction, goal reasoning, and long-horizon planning.

SimplerEnv-WidowX. SimplerEnv (Li et al., 2024b) is a simulation benchmark designed to evaluate robot manipulation policies in realistic environments with improved reproducibility and controllable variations. It provides simulation settings that closely follow real-world robot manipulation scenarios while enabling systematic evaluation under controlled conditions. In our experiments, we use the WidowX manipulation environment, which contains diverse tabletop manipulation tasks involving object grasping, placement, and rearrangement. Specifically, we evaluate four representative tasks: Put Spoon on Towel, Put Carrot on Plate, Stack Green on Yellow Block, and Put Eggplant in Yellow Basket. These tasks cover different manipulation skills, including object relocation, goal-directed placement, and multi-object interaction.

A.5 Real-World Experiment Details

Real-Robot Setup. We conduct real-world experiments using a 6-DoF SO-101 robotic arm equipped with two Intel RealSense RGB-D cameras, providing wrist-mounted and third-person views. The model takes 256×256 RGB images from both cameras as visual inputs.

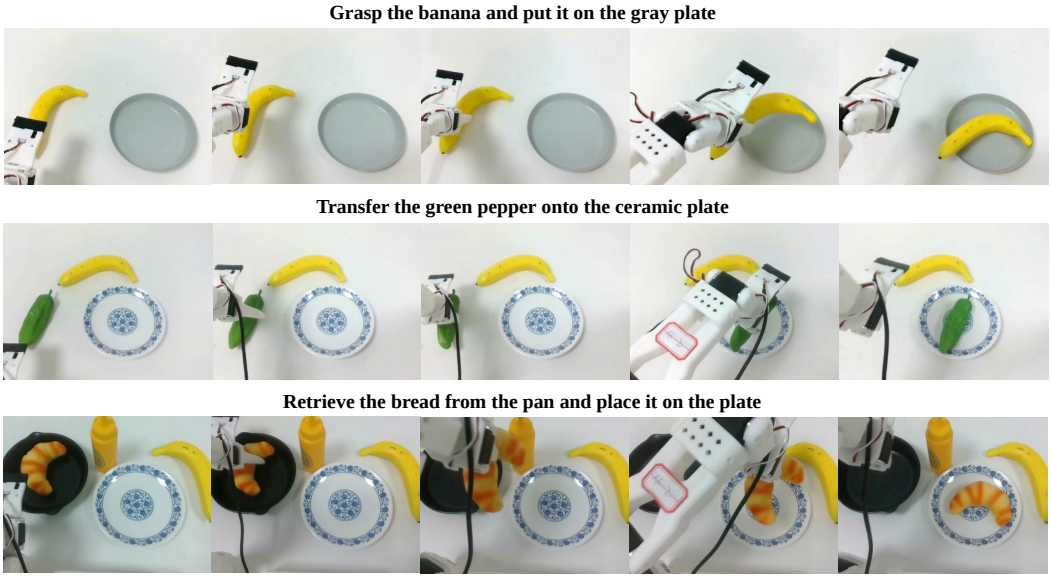


Figure 4: Real-world manipulation tasks performed with the SO-101 robotic arm.

Table 5: Effect of different future-prediction query configurations on LIBERO using RynnVLA-002.

Query Configuration	Success Rate (%) ↑				
	Spatial	Object	Goal	Long	Average
Future-Prediction Queries	**99.0**	**99.4**	**93.2**	**93.8**	**96.4**
Zero Queries	98.2	98.2	90.6	90.8	94.5
Random Queries	98.4	98.0	87.8	92.4	94.2

Task Design. We design three tabletop manipulation tasks with progressively increasing scene complexity by introducing additional distractor objects. The tasks are specified by three language instructions: Task 1, "Grasp the banana and put it on the gray plate"; Task 2, "Transfer the green pepper onto the ceramic plate"; and Task 3, "Retrieve the bread from the pan and place it on the plate." The corresponding task environments and execution sequences are shown in Fig. 4.

Training and Evaluation. We collect 100 demonstration trajectories for each task and independently fine-tune a separate RynnVLA-002 model. Fine-tuning is performed for 70,000 steps on four NVIDIA A800 80GB GPUs, using a learning rate of 5×10^{-6} and a weight decay of 0.15. During evaluation, both the original RynnVLA-002 model and FAVP are deployed on a single NVIDIA RTX 4090 GPU. We conduct 30 trials per task, with slight variations in the initial object positions across trials. A trial is considered successful only if the robot completes the entire task. FAVP is applied to the resulting task-specific policies without any additional training or parameter updates.

A.6 Additional Experiments and Analysis

Effect of Future-Prediction Queries To investigate the importance of the native future-prediction queries, we replace them with zero queries and random queries while keeping all other FAVP components unchanged. As shown in Table 5, the native future-prediction queries achieve an average success rate of 96.4%, outperforming zero and random queries by 1.9 and 2.2 percentage points, respectively. These results indicate that the native future-prediction queries provide more effective guidance for vision token selection than arbitrary auxiliary queries.

Inference Overhead Analysis. To quantify the additional computation introduced by FAVP, we analyze the inference overhead of its additional computation modules with RynnVLA-002. Specifi-

Table 6: Latency overhead analysis of FAVP components. Tech.1 denotes Future-Prediction Query Augmentation in Sec. 4.1, and Tech.2 denotes Action–Future Token Selection in Sec. 4.3.

Configuration	**Base Latency**	**Tech.1 Cost**	**Tech.2 Cost**	**Total Latency**	**Overhead**
Vanilla inference	260.28 ms	–	–	260.28 ms	–
+ Tech.1	260.28 ms	3.87 ms	–	264.15 ms	+1.49%
+ Tech.1 & Tech.2	260.28 ms	3.87 ms	3.59 ms	267.74 ms	+2.87%

Table 7: Effect of pruning layer on LIBERO using RynnVLA-002.

Pruning Layer	Success Rate (%) ↑					FLOPs (T) ↓
	Spatial	Object	Goal	Long	Average	
Layer 1	91.8	39.0	62.4	84.0	69.3	**3.13**
Layer 2	98.6	89.0	82.8	92.0	90.6	3.23
Layer 3	**99.0**	**99.4**	93.2	**93.8**	**96.4**	3.32
Layer 4	97.4	**99.4**	**94.0**	93.6	96.1	3.41

cally, Tech.1 denotes the Future-Prediction Query Augmentation module, while Tech.2 denotes the Action–Future Token Selection module. As shown in Table 6, enabling Tech.1 introduces only 3.87 ms additional latency, corresponding to a 1.49% overhead compared with vanilla action inference. Adding Tech.2 further introduces only 3.59 ms additional latency, resulting in a total overhead of 2.87% when both components are enabled without token pruning. These results demonstrate that FAVP can effectively extract future-aware information with negligible inference overhead.

Pruning Layer Sensitivity Analysis. We investigate the effect of the pruning layer on FAVP by varying its position from the first to the fourth Transformer layer while keeping the other settings unchanged. As shown in Table 7, pruning at the first layer results in a substantial performance drop, with an average success rate of 69.3%. Moving the pruning layer to the second and third layers improves the average success rate to 90.6% and 96.4%, respectively. Pruning at the fourth layer achieves a comparable success rate of 96.1%, but incurs higher computational cost than pruning at the third layer. These results suggest that pruning at excessively early layers may discard visual information before sufficient feature processing, whereas delaying pruning reduces computational savings. We therefore select the third Transformer layer as the default pruning layer for FAVP.