TTF-VLA: Temporal Token Fusion via Pixel-Attention Integration for Vision-Language-Action Models

Chenghao Liu[1*], Jiachen Zhang[1*], Chengxuan Li[1*], Zhimu Zhou[1], Shixin Wu[1], Songfang Huang[1†], Huiling Duan[1†]

[1]School of Advanced Manufacturing and Robotics, Peking University
chliu@stu.pku.edu.cn, z89498323286@gmail.com, lichengxuan0904@gmail.com, z18082717992@gmail.com, wushixin@stu.pku.edu.cn, hsf@pku.edu.cn, hlduan@pku.edu.cn

Abstract

Vision-Language-Action (VLA) models process visual inputs independently at each timestep, discarding valuable temporal information inherent in robotic manipulation tasks. This frame-by-frame processing makes models vulnerable to visual noise while ignoring the substantial coherence between consecutive frames in manipulation sequences. We propose Temporal Token Fusion (TTF), a training-free approach that intelligently integrates historical and current visual representations to enhance VLA inference quality. Our method employs dual-dimension detection combining efficient grayscale pixel difference analysis with attention-based semantic relevance assessment, enabling selective temporal token fusion through hard fusion strategies and keyframe anchoring to prevent error accumulation. Comprehensive experiments across LIBERO, SimplerEnv, and real robot tasks demonstrate consistent improvements: 4.0 percentage points average on LIBERO (72.4% vs 68.4% baseline), cross-environment validation on SimplerEnv (4.8% relative improvement), and 8.7% relative improvement on real robot tasks. Our approach proves model-agnostic, working across OpenVLA and VLA-Cache architectures. Notably, TTF reveals that selective Query matrix reuse in attention mechanisms enhances rather than compromises performance, suggesting promising directions for direct KQV matrix reuse strategies that achieve computational acceleration while improving task success rates.

Code — https://github.com/PKU-XLab/TTF-VLA

1 Introduction

Vision-Language-Action (VLA) models have emerged as a transformative paradigm in robotic manipulation, seamlessly integrating visual perception, natural language understanding, and action generation within unified neural architectures. Building upon the success of large-scale vision-language transformers, recent VLA systems (Brohan et al. 2022, 2023; Octo Model Team et al. 2024; Kim et al. 2024; Kim, Finn, and Liang 2025; Black et al. 2024, 2025) have demonstrated unprecedented capabilities in executing complex manipulation instructions across diverse environments. These models fundamentally reshape robotic control by treating action prediction as a multimodal sequence generation task, where visual observations and natural language instructions are jointly processed to produce discrete action tokens that guide continuous robot behavior.

However, despite their remarkable achievements, current VLA models suffer from a critical limitation: they process visual inputs in temporal isolation, treating each frame independently without leveraging the substantial temporal coherence inherent in robotic manipulation sequences. This frame-by-frame processing systematically recomputes all visual tokens from scratch at each timestep, discarding valuable temporal information even when the majority of visual content remains consistent across adjacent frames. Moreover, this approach makes models vulnerable to visual noise including lighting fluctuations, motion blur, and sensor artifacts that are common in robotic manipulation environments.

This temporal myopia creates a fundamental challenge: while naive historical token integration risks overlooking critical changes in object poses or environmental conditions, completely ignoring temporal context misses opportunities to leverage the structured patterns inherent in robotic manipulation. Specifically, visual changes typically concentrate in localized, task-relevant regions while background areas remain static. This observation suggests that effective temporal integration requires distinguishing between *spatial dynamics* from physical movements and *semantic relevance* shifts reflecting task-specific importance.

Motivated by these insights, we propose a training-free temporal token fusion framework that intelligently integrates historical and current visual representations to enhance VLA inference quality. Our approach introduces a dual-dimension detection mechanism that combines computationally efficient grayscale pixel difference analysis with attention-guided semantic relevance assessment, enabling informed decisions about temporal token integration. Through hard fusion strategies coupled with adaptive keyframe mechanisms, our method effectively balances temporal coherence with responsiveness to task-critical changes while preventing long-term error accumulation.

Our key contributions are:

- A novel temporal token fusion framework featuring dual-

[*]Equal contribution.
[†]Corresponding authors.

dimension detection that combines efficient grayscale pixel difference analysis with attention-guided semantic relevance assessment for intelligent integration of historical and current visual representations.

- An adaptive fusion strategy employing hard token selection with keyframe anchoring mechanisms that balances temporal coherence with responsiveness to task-critical changes, applicable across diverse VLA architectures without requiring model retraining.
- Comprehensive experimental validation: 4.0 percentage points average improvement on LIBERO (Liu et al. 2023), cross-environment generalization on SimplerEnv (Li et al. 2024), and 8.7% relative improvement on real robot tasks, with model-agnostic applicability across OpenVLA (Kim et al. 2024) and VLA-Cache (Xu et al. 2025) architectures.
- Discovery of beneficial Query matrix reuse through temporal token fusion, revealing promising "free lunch" directions for direct KQV matrix reuse that achieve computational acceleration while improving task performance.

Our approach provides a principled solution to the fundamental tension between leveraging temporal coherence and maintaining sensitivity to dynamic changes in VLA models.

2 Related Work

Vision-Language-Action Models Vision-Language-Action (VLA) models have emerged as a unified framework for robotic manipulation, integrating visual perception, language understanding, and action prediction. Early works like RT-1 and RT-2 tokenized actions with visual and linguistic inputs for end-to-end policy learning (Brohan et al. 2022, 2023). Recent advancements include Octo and OpenVLA for open-source implementations (Octo Model Team et al. 2024; Kim et al. 2024; Kim, Finn, and Liang 2025), and Pi-0/Pi-0.5 for flow-based architectures (Black et al. 2024, 2025). However, current models exhibit limitations in handling temporal dynamics and visual artifacts (Chi et al. 2023; Wen et al. 2024; Collaboration et al. 2023; Huang et al. 2024), often processing frames independently while ignoring temporal coherence.

Token Processing Techniques Efficient token processing has become crucial for transformer deployment. In vision transformers, attention-guided methods such as DynamicViT, AdaViT, and EViT progressively prune redundant tokens (Rao et al. 2021; Meng et al. 2022; Liang et al. 2022), while ToMe optimizes efficiency through strategic token consolidation (Bolya et al. 2023). For vision-language models, recent approaches like FastV and SparseVLM focus on visual token sparsification using text-guided selection strategies (Chen et al. 2024; Zhang et al. 2024). VLA-Cache specifically targets robotic scenarios through KV-cache reuse mechanisms (Xu et al. 2025). However, these methods primarily address spatial redundancy within individual frames. Unlike these spatial compression approaches, our work targets temporal redundancy across sequential frames, proposing a dual-dimension strategy that leverages historical information to enhance VLA inference quality.

3 Methodology

In robotic manipulation tasks, consecutive frames often exhibit substantial visual redundancy, yet subtle but critical changes in object poses, lighting conditions, or environmental context can significantly impact action prediction quality. Our approach leverages temporal token fusion to enhance VLA inference quality through intelligent integration of historical and current visual representations. Figure 1 provides an overview of our complete framework.

3.1 Problem Formulation

Vision-Language-Action models process sequential inputs of the form $\{\mathbf{I}_t, \mathbf{L}_t\} \rightarrow \mathbf{A}_t$, where $\mathbf{I}_t \in \mathbb{R}^{H\times W\times C}$ represents the visual observation, $\mathbf{L}_t$ denotes the language task instruction, and $\mathbf{A}_t \in \mathbb{R}^7$ represents the predicted 7-DoF robotic action at timestep t.

Current VLA models typically employ vision-language transformer architectures that process visual observations through patch-based encoders. The vision encoder extracts patch tokens $\mathbf{T}_t = \{\mathbf{t}_t^{(i)}\}_{i=1}^N$ from input images, which are then projected to the language model's embedding space and integrated with tokenized task instructions before being processed by the transformer backbone.

Temporal Token Fusion Given a sequence of visual observations $\{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_t\}$ and corresponding patch tokens $\{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_t\}$, our goal is to learn a fusion function $\mathcal{F}$ that intelligently integrates temporal information:

$$\tilde{\mathbf{T}}_t = \mathcal{F}(\mathbf{T}_t, \mathbf{T}_{t-1}, \mathbf{I}_t, \mathbf{I}_{t-1}, \mathbf{L}_t) \tag{1}$$

where $\tilde{\mathbf{T}}_t$ represents the temporally-fused patch tokens that maintain critical current information while leveraging relevant historical context. The fusion function must balance temporal coherence with responsiveness to important changes in the visual scene.

3.2 Temporal Token Fusion Framework

Hard Fusion Strategy Our temporal fusion framework operates through a systematic process that evaluates each patch for intelligent temporal integration, as detailed in Figure 2. The framework employs a hard fusion strategy that makes binary selection decisions for each patch, choosing between current and historical tokens based on dual-dimension detection:

$$\tilde{\mathbf{t}}_t^{(i)} = \begin{cases} \mathbf{t}_t^{(i)} & \text{if } m_i^{\text{fusion}} = 1 \\ \mathbf{t}_{t-1}^{(i)} & \text{if } m_i^{\text{fusion}} = 0 \end{cases} \tag{2}$$

where $m_i^{\text{fusion}} \in \{0, 1\}$ is the binary fusion mask that determines temporal integration decisions for patch i, computed through our dual-dimension detection combining grayscale pixel difference detection and attention-based semantic relevance detection (detailed in Section 3.3). This strategy provides clear temporal context selection, aligning with the discrete nature of robotic manipulation tasks, where important patches (mask=1) use current frame tokens and others (mask=0) reuse previous frame tokens.

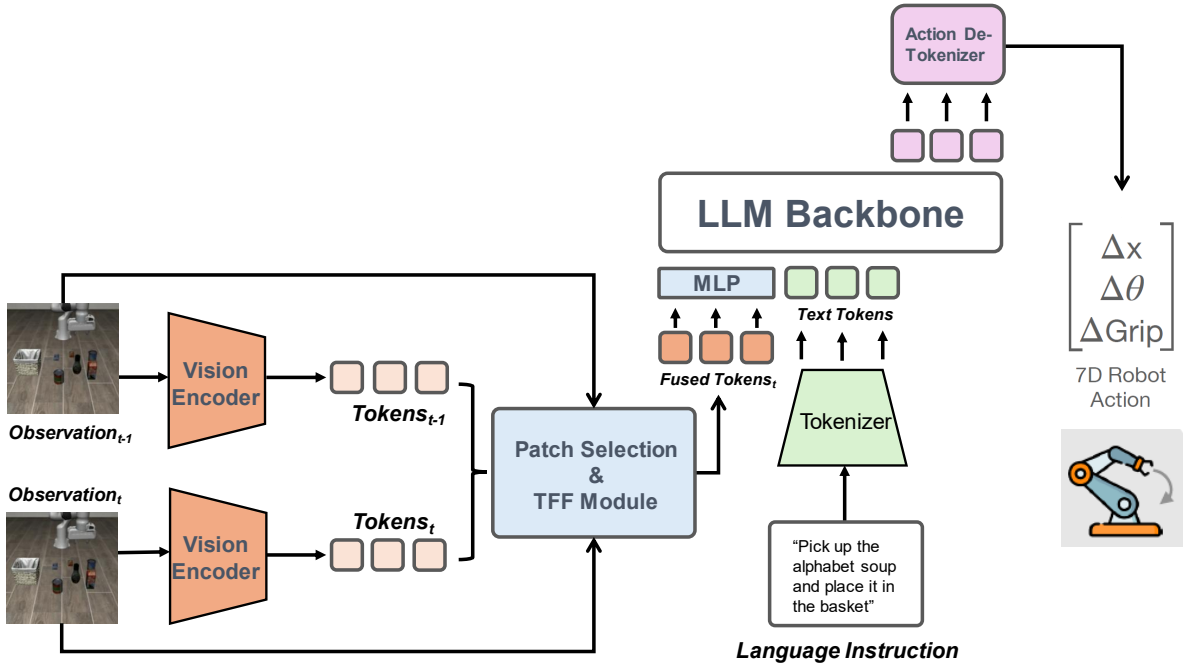

Figure 1: Overall Framework of Temporal Token Fusion for VLA Models. The framework illustrates the end-to-end process, where the Vision Encoder extracts tokens from current (Observation$_t$) and previous (Observation$_{t-1}$) frames. These are then processed by the Patch Selection module and TTF module for patch selection and token fusion. The fused tokens are subsequently fed into the LLM Backbone, combined with language instruction, to generate 7-DoF robotic actions via the Action Detokenizer.

Keyframe Mechanism To prevent long-term error accumulation, we introduce periodic keyframes where all patches are unconditionally recomputed:

$$\text{IsKeyframe}(t) = (t \bmod K = 0) \vee (\mathbf{T}_{t-1} = \emptyset) \tag{3}$$

The keyframe interval K balances temporal context with stability, preventing long-term error accumulation while preserving temporal coherence benefits.

The final fusion decision integrates both dimensions through carefully designed combination rules. The fusion mask $m_i^{\text{fusion}} \in \{0,1\}$ determines the final temporal integration decision for patch i, where 1 indicates using current tokens for important patches and 0 indicates reusing historical tokens for others, as guided by the dual-dimension detection in Figure 2(a). We use logical OR to ensure patches are updated when exhibiting either type of change:

$$m_i^{\text{fusion}} = m_i^{\text{pixel}} \vee m_i^{\text{attention}} \tag{4}$$

This OR operation ensures that patches use current frame tokens when either dimension indicates importance, providing conservative fusion that prioritizes inference quality through comprehensive temporal context integration.

3.3 Dual-Dimension Detection

Our fusion framework integrates two complementary analytical dimensions to identify important patches, as shown in Figure 2(a). This dual-constraint approach ensures comprehensive coverage of both low-level visual dynamics and high-level semantic relevance.

Grayscale Pixel Difference Detection The pixel-level dimension captures fine-grained spatial changes through efficient grayscale-based analysis (Fasola and Veloso 2006), contributing to the identification of important patches as part of the dual-dimension detection in Figure 2(a). Our approach offers several advantages over token-space similarity metrics commonly used in caching systems.

We convert RGB frames to grayscale using standard luminance weights to focus on brightness changes while maintaining computational simplicity:

$$\mathbf{G}_t = 0.299 \cdot \mathbf{I}_t^R + 0.587 \cdot \mathbf{I}_t^G + 0.114 \cdot \mathbf{I}_t^B \tag{5}$$

This conversion is motivated by robotic manipulation scenarios where meaningful changes (object movements, shadow variations, lighting shifts) are primarily reflected in luminance rather than chromatic information.(Yang, Tan, and Ahuja 2012; Smeulders et al. 2014; Secci and Ceccarelli 2023)

For each patch i spanning spatial region (u_i, v_i) to $(u_i + 13, v_i + 13)$ in the 14×14 pixel patches, we compute the average absolute difference:

$$d_i^{\text{pixel}} = \frac{1}{196} \sum_{(u,v) \in \text{patch}_i} |\mathbf{G}_t(u,v) - \mathbf{G}_{t-1}(u,v)| \tag{6}$$

Compared to cosine similarity on high-dimensional patch tokens, our pixel-based approach offers: (1) $O(1)$ complexity per patch vs. $O(d)$ for d-dimensional tokens, (2) direct interpretability of change magnitude, and (3) sensitivity to

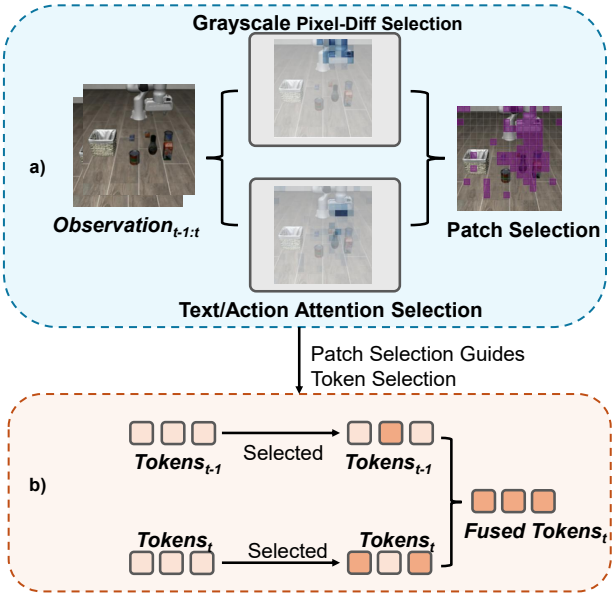

Figure 2: The Details of Patch Selection and Temporal Token Fusion. The process includes (a) Grayscale Pixel Difference Detection and Attention-Based Semantic Relevance Detection for identifying important patches, and (b) fusion of selected Tokens$_t$ with Tokens$_{t-1}$ into Fused Tokens$_t$, where important patches use current frame tokens and others use previous frame tokens.

subtle manipulator movements that may be lost in token compression.

The grayscale pixel difference mask $m_i^{\text{pixel}} \in \{0,1\}$ indicates whether patch i exhibits significant spatial changes, where 1 denotes using current frame tokens. The binary mask is computed as: $m_i^{\text{pixel}} = \mathbb{I}\left[d_i^{\text{pixel}} > \tau_{\text{pixel}}\right]$, where the pixel threshold τ_{pixel} is set based on scene statistics.

Attention-Based Semantic Relevance Detection The attention dimension identifies semantically important patches using transformer attention patterns. Building upon token selection methods in vision-language models (Chen et al. 2024; Zhang et al. 2024) and attention-guided approaches in vision transformers (Rao et al. 2021; Meng et al. 2022; Liang et al. 2022; Bolya et al. 2023; Xu et al. 2025), our approach introduces dual attention sources specifically adapted for temporal fusion in robotic manipulation tasks.

We extract attention weights $\mathbf{A}_{t-1}^{(l)}$ from a selected transformer layer and compute patch relevance through two complementary attention sources:

Text-to-Vision Attention: Captures semantic relevance based on task instruction by aggregating attention weights from text tokens to vision patches:

$$\mathbf{S}_{\text{text}}^{(l)} = \frac{1}{N_h}\sum_{h=1}^{N_h}\frac{1}{N_{\text{text}}}\sum_{j\in\text{text tokens}}\left(\mathbf{A}_{t-1}^{(l)}\right)_{h,j,\text{vision}} \tag{7}$$

Action-to-Vision Attention: Leverages attention from the first action token, which encodes high-level manipulation strategy and spatial relevance for end-effector positioning:

$$\mathbf{S}_{\text{action}}^{(l)} = \frac{1}{N_h}\sum_{h=1}^{N_h}\left(\mathbf{A}_{t-1}^{(l)}\right)_{h,\text{action}_1,\text{vision}} \tag{8}$$

For the selected layer l, we obtain the final task-relevance attention scores by selecting either text-to-vision or action-to-vision mode:

$$S_i^{\text{task}} = \mathbf{S}_{\text{mode}}^{(l)}[i] \tag{9}$$

To avoid computational overhead during inference, we utilize attention weights from the previous timestep $\mathbf{A}_{t-1}$, motivated by the temporal stability of task-relevant regions in robotic manipulation. The attention mask $m_i^{\text{attention}} \in \{0,1\}$ is determined via top-k selection: $m_i^{\text{attention}} = \mathbb{I}[i \in \text{TopK}(S^{\text{task}}, k)]$, where patches with high attention scores (mask=1) use current frame tokens while others (mask=0) reuse historical tokens.

Algorithm 1: Temporal Token Fusion

Input: Current frame $\mathbf{I}_t$, previous frame $\mathbf{I}_{t-1}$, previous tokens $\mathbf{T}_{t-1}$, previous attention $\mathbf{A}_{t-1}$
Output: Fused tokens $\tilde{\mathbf{T}}_t$
if IsKeyframe(t) **then**
  $\tilde{\mathbf{T}}_t \leftarrow \text{VisionEncoder}(\mathbf{I}_t)$
else
  $\mathbf{T}_t, \mathbf{A}_t \leftarrow \text{VisionEncoder}(\mathbf{I}_t)$ {Extract tokens and attention weights}
  for $i = 1$ to N **do**
    $m_i^{\text{pixel}} \leftarrow \text{GrayscalePixelDiff}(\mathbf{I}_t, \mathbf{I}_{t-1}, i)$
    $m_i^{\text{attention}} \leftarrow \text{AttentionRelevance}(\mathbf{A}_{t-1}, i)$
    $m_i^{\text{fusion}} \leftarrow m_i^{\text{pixel}} \vee m_i^{\text{attention}}$
    $\tilde{\mathbf{t}}_t^{(i)} \leftarrow \text{TemporalFuse}(\mathbf{t}_t^{(i)}, \mathbf{t}_{t-1}^{(i)}, m_i^{\text{fusion}})$
  end for
end if

4 Experiments

4.1 Experimental Setup

Baseline Models We evaluate our temporal token fusion approach across multiple VLA architectures to demonstrate its model-agnostic applicability:

- **OpenVLA:** For LIBERO experiments, we employ officially released task-specific fine-tuned checkpoints (openvla-7b-finetuned-libero-{object, spatial, goal, long}), each optimized for the corresponding task suite. For SimplerEnv, we use the base OpenVLA-7B model without task-specific fine-tuning to evaluate cross-environment generalization. For real robot experiments, we fine-tune OpenVLA-7B on task-specific demonstration data.
- **VLA-Cache:** We incorporate this recent architecture to demonstrate TTF's cross-model generalizability across different VLA paradigms. Notably, VLA-Cache's existing Key-Value matrix reuse mechanism provides an

ideal testbed for exploring TTF's impact on Query matrix reuse, revealing promising directions for direct KQV matrix reuse strategies that could achieve computational acceleration while improving task performance.

Evaluation Benchmarks Our comprehensive evaluation spans both simulation and real-world environments:

- **LIBERO:** Simulation evaluation across four task suites: (1) *Object* - manipulation of diverse objects (e.g., varying shapes, sizes) through single-object interactions; (2) *Spatial* - tasks requiring precise spatial reasoning and object placement relative to landmarks; (3) *Goal* - complex goal-conditioned tasks with multi-step reasoning for specific configurations; (4) *Long* - long-horizon tasks testing temporal consistency over sequential manipulations. Each suite contains 10 distinct tasks with 20 evaluation episodes per task (200 total episodes per suite).
- **SimplerEnv:** Simulation benchmark for evaluating real-world robot manipulation policies across diverse scenarios. We evaluate three representative tasks: *Move Near Object* (240 episodes), *Pick Coke Can* with multiple orientations (300 episodes), and *Drawer Operations* (216 episodes), providing comprehensive validation across varying manipulation complexities and interaction types.
- **Real Robot Tasks:** Physical validation using a Franka Research 3 robot across three manipulation tasks spanning different complexities: single-object pick-and-place (*put the garlic on the plate*), multi-object sequential manipulation (*put the pepper and corn on the plate*), and contact-rich manipulation (*close the drawer*). We collect 80 demonstration episodes per task using Gello teleoperation at 5 Hz, and fine-tune OpenVLA-7B for 20,000 steps with a batch size of 8. During evaluation, the fine-tuned models are deployed at 5 Hz and tested with 20 episodes for each task.

4.2 Main Results

LIBERO Experiments Table 1 presents our core experimental results across four task suites using two representative VLA architectures, demonstrating the effectiveness and model-agnostic applicability of our temporal token fusion approach. Our analysis reveals several key insights: *Model-agnostic effectiveness*: Both OpenVLA and VLA-Cache demonstrate consistent improvements (4.0 and 2.7 percentage points average respectively), validating the generalizability of our approach across different VLA architectures. *Task-specific patterns*: Long-horizon tasks benefit most significantly from temporal fusion (+11.5% relative improvement for OpenVLA), suggesting that extended manipulation sequences particularly benefit from temporal context integration, while object manipulation shows strong absolute gains (+6.0 percentage points for OpenVLA). *Fusion efficiency*: The fusion rates (42.8% average for OpenVLA) indicate substantial feature reuse while maintaining performance gains, demonstrating that our dual-dimension detection successfully identifies stable regions for temporal integration without compromising inference quality. *Temporal progression analysis*: Figure 3 illustrates representative failure-to-success cases where baseline methods fail but TTF succeeds, highlighting the critical role of temporal coherence in successful robotic manipulation task completion. TTF introduces less than 2% additional runtime overhead, confirming its efficiency and suitability for real-time robotic inference.

Implicit Query Reuse and Noise Robustness: A particularly revealing insight emerges from VLA-Cache+TTF results. VLA-Cache (Xu et al. 2025) accelerates inference by reusing Key-Value matrices ($\mathbf{K}_{t-1}^{(l)}, \mathbf{V}_{t-1}^{(l)}$) for static visual patches while always recomputing Query matrices ($\mathbf{Q}_t^{(l)}$) to maintain contextual sensitivity. However, when TTF is applied to VLA-Cache, an important mechanism emerges: TTF's token-level fusion ($\tilde{\mathbf{t}}_t^{(i)} = \mathbf{t}_{t-1}^{(i)}$ for selected patches) means that the corresponding Query matrix portions are approximately and implicitly reused since $\mathbf{Q}_t^{(l)} = \mathbf{W}_q^{(l)} \cdot \tilde{\mathbf{T}}_t$. Thus, VLA-Cache+TTF effectively *nearly reuses all three attention matrices* (K, V, and Q) for static patches, going beyond VLA-Cache's original KV-only reuse strategy. Contrary to conventional wisdom that Query reuse degrades performance due to contextual sensitivity, our results show significant improvements, particularly for long-horizon tasks (+3.0 points: 55.0%→58.0%). This performance improvement demonstrates that our dual-dimension detection provides stabilized contextual representations that enhance robustness against visual observation noise—including lighting fluctuations, motion blur, and sensor artifacts common in robotic manipulation. The selective Query reuse validates TTF's core hypothesis: temporal coherence in stable regions enhances rather than compromises inference quality. This finding reveals a promising "free lunch" future direction worth exploring: *direct KQV matrix reuse* for static patches could achieve computational acceleration while simultaneously improving task success rates.

Configuration Strategy: Parameter selection demonstrates TTF's robustness. OpenVLA employs consistent parameters: keyframe interval K=3, attention top-k=70, pixel threshold=0.03, and text-to-vision attention. VLA-Cache requires minimal adaptation with task-adaptive fusion rates (30% for Object/Spatial/Long, 50% for Goal tasks). Since VLA-Cache uses patch importance scoring, we directly utilize its scores for patch selection, avoiding redundant computation. Neither model requires task-specific hyperparameter tuning.

SimplerEnv Cross-Environment Validation To validate cross-environment generalization, we evaluate TTF on SimplerEnv using identical parameters as our LIBERO experiments. The experimental results in Table 2 demonstrate consistent improvements across all three tasks, with an average improvement of 1.6 percentage points (4.8% relative). TTF shows particularly strong gains in Pick Coke Can (+10.0% relative), indicating that complex grasping tasks with multiple orientations benefit from temporal coherence.

These results validate TTF's environment-agnostic effectiveness, confirming that temporal coherence benefits extend across different simulation platforms without requiring parameter re-tuning.

Base Model	Object	Spatial	Goal	Long	Average
OpenVLA	66.5	82.0	77.0	48.0	68.4
OpenVLA + TTF	**72.5**	**84.5**	**79.0**	**53.5**	**72.4**
Fusion Rates	44.6	41.2	43.5	42.0	42.8
Improvement	+6.0 (+9.0%)	+2.5 (+3.0%)	+2.0 (+2.6%)	+5.5 (+11.5%)	+4.0 (+5.8%)
VLA-Cache	69.0	84.0	77.0	55.0	71.3
VLA-Cache + TTF	**73.0**	**84.0**	**81.0**	**58.0**	**74.0**
Improvement	+4.0 (+5.8%)	+0.0 (+0.0%)	+4.0 (+5.2%)	+3.0 (+5.5%)	+2.7 (+3.8%)

Table 1: Task success rates (%) and temporal fusion rates (%) on the LIBERO benchmark using OpenVLA models fine-tuned for each task suite. Success rates denote the percentage of successful task completions across 200 episodes per task suite (10 tasks × 20 episodes each). The fusion rates represent the proportion of vision tokens reused from the previous frame in the fused representation.

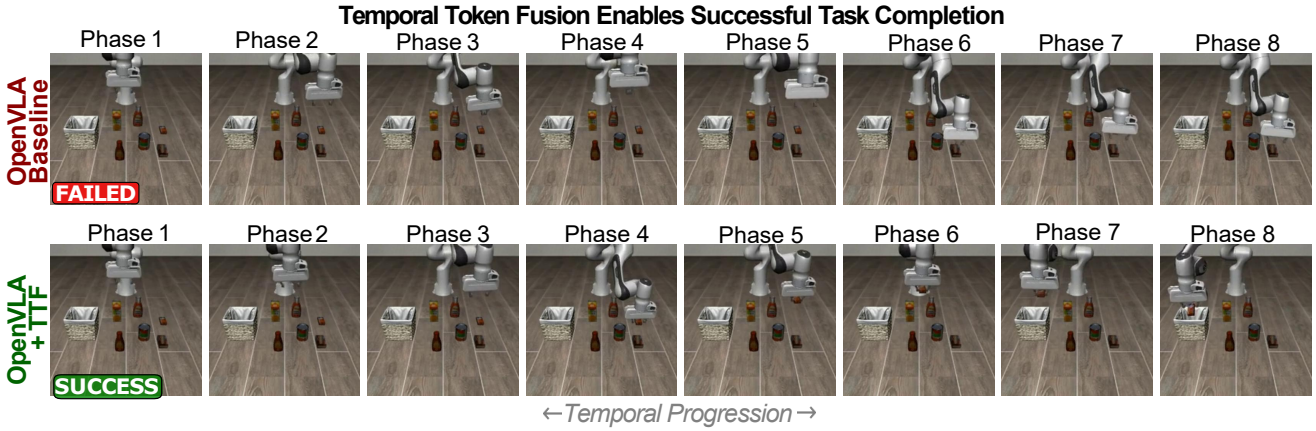

Figure 3: Temporal progression analysis illustrating a failure-to-success transition for the task instruction "pick up the butter and place it in the basket." Eight key phases showing OpenVLA baseline (failed) vs. OpenVLA + TTF (successful), demonstrating the critical role of temporal consistency in successful manipulation.

Task	Success Rate (%)		Improvement	
	OpenVLA	TTF	Absolute	Relative
Move Near	49.4	52.1	+2.7	+5.5%
Pick Coke	17.0	18.7	+1.7	+10.0%
Drawer	33.3	33.8	+0.5	+1.5%
Average	33.2	34.9	+1.6	+4.8%

Table 2: Cross-environment validation on SimplerEnv benchmark using the base OpenVLA-7B model (without task-specific fine-tuning) with identical TTF parameters as LIBERO experiments. Results demonstrate consistent generalization across simulation platforms and manipulation complexities.

Figure 4: Real robot manipulation tasks used for physical validation of TTF: (a) single-object pick-and-place, (b) multi-object sequential manipulation, and (c) contact-rich drawer closing.

Real Robot Experiments For real robot experiments, we employ more sensitive parameter settings to handle visual noise: pixel threshold=0.01 and attention top-k=100, while maintaining other configurations consistent with LIBERO experiments. Real-world environments provide an ideal testbed for TTF's noise robustness due to dynamic lighting, sensor noise, motion blur, and other visual artifacts absent in simulation.

The experimental results in Table 4 demonstrate TTF's effectiveness in real-world scenarios, achieving an average improvement of 3.3 percentage points (8.7% relative). Both pick-and-place tasks (Garlic, Pepper & Corn) show substantial gains from temporal coherence (+5.0 points each, 12.5% and 16.7% relative), demonstrating TTF's particular strength in visual tracking and object manipulation tasks. These results validate TTF's practical applicability in real-world deployment, especially for manipulation scenarios involving object interaction and spatial reasoning.

4.3 Ablation Studies

We conduct comprehensive ablation studies to validate the necessity of each component in our dual-dimension tempo-

Method	Object	Spatial	Goal	Long	Average	Fusion Rates
OpenVLA	66.5	82.0	77.0	48.0	68.4	–
Pixel-only TTF	72.0 +5.5 (+8.3%)	80.5 -1.5 (-1.8%)	76.5 -0.5 (-0.6%)	52.5 +4.5 (+9.4%)	70.4 +2.0 (+2.9%)	61.1/57.8/59.9/59.8
Attention-only TTF	68.0 +1.5 (+2.3%)	83.0 +1.0 (+1.2%)	77.5 +0.5 (+0.6%)	56.5 +8.5 (+17.7%)	71.3 +2.9 (+4.2%)	48.3/48.2/48.3/48.4
Pixel-Attention TTF	**72.5** +6.0 (+9.0%)	**84.5** +2.5 (+3.0%)	**79.0** +2.0 (+2.6%)	**53.5** +5.5 (+11.5%)	**72.4** +4.0 (+5.8%)	**44.6/41.2/43.5/42.0**

Table 3: Task success rates (%) and fusion rates (%) in analysis dimension ablation study across LIBERO task suites using OpenVLA task-specific fine-tuned checkpoints. Success rates denote the percentage of successful completions across 200 episodes per task suite (10 tasks × 20 episodes each). Fusion rates represent the proportion of vision tokens from the previous frame retained in the fused representation.

Task	Success Rate (%)		Improvement	
	OpenVLA	TTF	Absolute	Relative
Garlic	40.0	45.0	+5.0	+12.5%
Pepper & Corn	30.0	35.0	+5.0	+16.7%
Drawer	45.0	45.0	+0.0	+0.0%
Average	38.3	41.7	+3.3	+8.7%

Table 4: Real robot manipulation success rates (%) across 20 episodes per task using OpenVLA-7B fine-tuned on task-specific data. Results demonstrate TTF's effectiveness in noisy real-world environments with enhanced temporal stability benefits.

ral fusion approach.

Analysis Dimension Validation Table 3 validates the necessity of our dual-dimension detection approach. Pixel-based analysis excels in detecting spatial changes, while attention-based analysis better captures task-relevant semantics. Our hybrid method delivers the best average performance (72.4%) by adaptively integrating both dimensions. Notably, its conservative OR-based fusion logic results in the lowest fusion rates (42.8%), prioritizing inference quality and confirming the complementary nature of pixel and attention-based detection.

Keyframe Mechanism Analysis Figure 5 presents keyframe interval analysis across 14 configurations (K=2 to K=200), revealing three distinct regimes: *Stable Range* (K≤15) with optimal performance at K=3, *Degradation Onset* (K=20-30) where error accumulation begins, and *Error Accumulation* (K≥30) with performance plateau. Long-horizon tasks show higher sensitivity to temporal drift. Fusion rates increase monotonically with keyframe interval (from 35% to 70%), highlighting the efficiency-performance trade-off and validating our conservative dual-dimension approach.

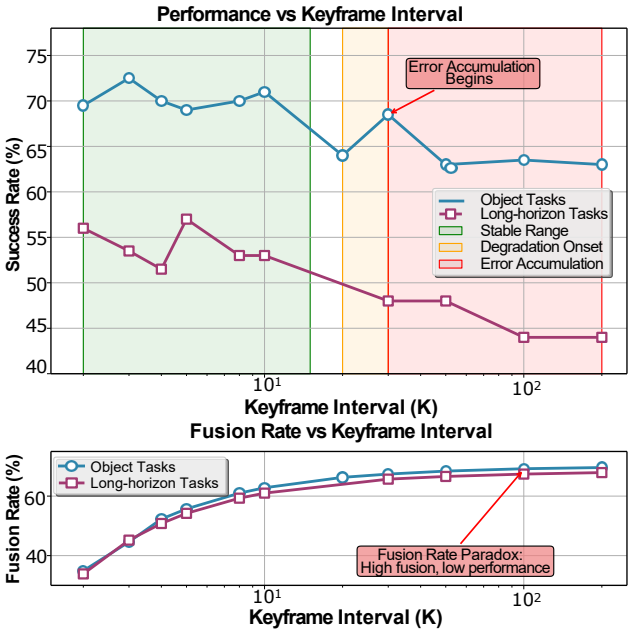

Figure 5: Keyframe interval analysis across Object and Long task suites. Top: Performance vs. Keyframe Interval showing error accumulation beyond K=30. Bottom: Fusion Rates vs. Keyframe Interval revealing the efficiency-performance trade-off. The analysis demonstrates three distinct regimes: stable performance (K≤15), degradation onset (K=20–30), and error accumulation (K≥30).

5 Conclusion

We present Temporal Token Fusion (TTF), a training-free approach that addresses the fundamental limitation of current VLA models in leveraging temporal coherence during robotic manipulation. Our dual-dimension detection framework combines pixel-level spatial dynamics with attention-based semantic relevance assessment to intelligently integrate historical and current visual representations. Comprehensive evaluation demonstrates consistent improvements across LIBERO (4.0 percentage points average), SimplerEnv (cross-environment generalization), and real robot tasks (8.7% relative improvement), with model-agnostic applicability across OpenVLA and VLA-Cache architectures. Beyond performance gains, our work reveals the unexpected benefits of selective Query matrix reuse in attention mechanisms, suggesting promising directions for direct KQV matrix strategies that achieve computational acceleration while improving task success rates.

References

Black, K.; Brown, N.; Darpinian, J.; Dhabalia, K.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Galliker, M. Y.; et al. 2025. $\pi_{0.5}$: A vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.

Bolya, D.; Fu, C.-Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; and Hoffman, J. 2023. Token Merging: Your ViT but Faster. In *International Conference on Learning Representations*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Chen, L.; Zhao, H.; Liu, T.; Bai, S.; Lin, J.; Zhou, C.; and Chang, B. 2024. An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models. In *Computer Vision – ECCV 2024*.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2023. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *The International Journal of Robotics Research*.

Collaboration, O. X.-E.; O'Neill, A.; Rehman, A.; Gupta, A.; Maddukuri, A.; et al. 2023. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv preprint arXiv:2310.08864*.

Fasola, J.; and Veloso, M. M. 2006. Real-time object detection using segmented and grayscale images. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 4088–4093. IEEE.

Huang, S.; Chang, H.; Liu, Y.; Zhu, Y.; Dong, H.; et al. 2024. A3VLM: Actionable Articulation-Aware Vision Language Model. *arXiv preprint arXiv:2406.07549*.

Kim, M. J.; Finn, C.; and Liang, P. 2025. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. *arXiv preprint arXiv:2502.19645*. Accepted to Robotics: Science and Systems (RSS) 2025.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Li, X.; Hsu, K.; Gu, J.; Pertsch, K.; Mees, O.; Walke, H. R.; Fu, C.; Lunawat, I.; Sieh, I.; Kirmani, S.; Levine, S.; Wu, J.; Finn, C.; Su, H.; Vuong, Q.; and Xiao, T. 2024. Evaluating Real-World Robot Manipulation Policies in Simulation. *arXiv preprint arXiv:2405.05941*.

Liang, Y.; Ge, C.; Tong, Z.; Song, Y.; Wang, J.; and Xie, P. 2022. EViT: Expediting Vision Transformers via Token Reorganizations. In *International Conference on Learning Representations*.

Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *arXiv preprint arXiv:2306.03310*.

Meng, L.; Li, H.; Chen, B.-C.; Lan, S.; Wu, Z.; Jiang, Y.-G.; and Lim, S.-N. 2022. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12309–12318.

Octo Model Team; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; et al. 2024. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems (RSS)*.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Advances in Neural Information Processing Systems*, 13937–13949.

Secci, F.; and Ceccarelli, A. 2023. RGB Cameras Failures and Their Effects in Autonomous Driving Applications. *IEEE Transactions on Dependable and Secure Computing*, 20(4): 2731–2745.

Smeulders, A. W. M.; Chu, D. M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; and Shah, M. 2014. Visual Tracking: An Experimental Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7): 1442–1468.

Wen, J.; Zhu, M.; Zhu, Y.; Tang, Z.; Li, J.; et al. 2024. Diffusion-VLA: Scaling Robot Foundation Models via Unified Diffusion and Autoregression. *arXiv preprint arXiv:2412.03293*. Initial version (v1).

Xu, S.; Wang, Y.; Xia, C.; Zhu, D.; Huang, T.; and Xu, C. 2025. VLA-Cache: Towards Efficient Vision-Language-Action Model via Adaptive Token Caching in Robotic Manipulation. *arXiv preprint arXiv:2502.02175*.

Yang, Q.; Tan, K.-H.; and Ahuja, N. 2012. Shadow Removal Using Bilateral Filtering. *IEEE Transactions on Image Processing*, 21(10): 4361–4368.

Zhang, Y.; Fan, C.-K.; Ma, J.; Zheng, W.; Huang, T.; Cheng, K.; Gudovskiy, D.; Okuno, T.; Nakata, Y.; Keutzer, K.; et al. 2024. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.