# Phase-Adaptive Fusion: Spatio-Temporal Modulation for VLA Models

**Lueke Ni**
East China Normal University (ECNU)
Shanghai, China
51275902101@stu.ecnu.edu.cn

**Junhao Zhu**
East China Normal University (ECNU)
Shanghai, China
52305902026@stu.ecnu.edu.cn

**Xinyu Zhang***
East China Normal University (ECNU)
Shanghai, China
xyzhang@sei.ecnu.edu.cn

## Abstract

Vision-Language-Action (VLA) foundation models have driven remarkable progress in general-purpose robotic manipulation. Yet, adapting these models to contact-rich physical environments exposes an optimization vulnerability: modality imbalance during fine-tuning. Because compact proprioceptive states often provide an easier path for loss minimization during imitation learning, policies can succumb to a "state-dominant shortcut," attenuating the utilization of high-dimensional visual features. This visual neglect degrades fine-grained spatial grounding, leading to terminal alignment drift during contact-critical execution phases. To address this issue without disrupting pre-trained representations or requiring heavy auxiliary decoders, we introduce Phase-Adaptive Fusion (PAF), a lightweight, forward-only neural modulation plug-in. PAF employs a decoupled dual-branch architecture: a spatial branch performs explicit language-conditioned semantic-geometric alignment, while a temporal branch infers latent task execution phases from recent proprioceptive states and action history. By fusing these signals through a bounded residual gating mechanism prior to multimodal fusion, PAF dynamically amplifies visual tokens during alignment-critical phases while preserving the host VLA's pre-trained representational manifold. Extensive evaluations across simulated benchmarks (LIBERO, Meta-World, Robomimic) and physical robot manipulation tasks demonstrate that PAF effectively mitigates the state-dominant shortcut, substantially reducing terminal alignment errors and improving overall manipulation success across diverse backbones.

## 1 Introduction

Vision-Language-Action (VLA) models have advanced robotic control toward general-purpose embodied intelligence by translating internet-scale multimodal representations into low-level motor commands (Brohan et al., 2022, 2023; Kim et al., 2024). However, adapting static, high-level perception from Vision-Language Models (VLMs) to dynamic, closed-loop physical execution introduces severe optimization challenges.

Through diagnostic evaluations under controlled interventions (Fig. 1), we empirically characterize an acute *modality imbalance* during downstream adaptation. Because compact, predictive proprioceptive states provide an easier path for imitation loss minimization, policies frequently succumb to a

*Corresponding author.

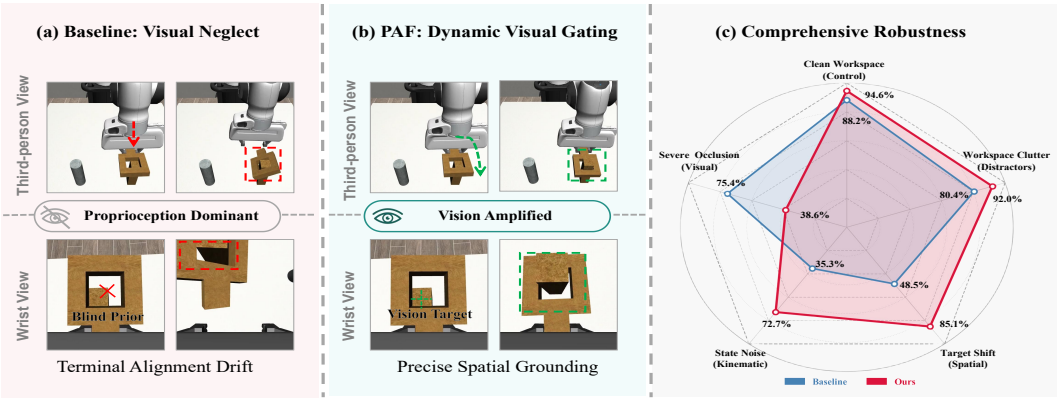



Figure 1: **Mitigating the State-Dominant Shortcut in VLA Models. (a)** Standard VLA fine-tuning frequently suffers from *visual neglect*. Governed by a proprioception-dominant prior, policies underutilize visual feedback, resulting in terminal alignment drift when target objects undergo spatial displacement. **(b)** Our proposed Phase-Adaptive Fusion (PAF) introduces dynamic spatiotemporal visual gating. By adaptively modulating visual representations during contact- and alignment-critical phases, PAF promotes reliable spatial grounding. **(c)** Controlled diagnostic evaluation on LIBERO-Plus. While the baseline exhibits pronounced performance degradation under kinematic state noise and spatial target shifts (characteristic of an over-reliance on proprioceptive history), PAF demonstrates robust closed-loop correction. Crucially, PAF's expected performance decrease under severe task-relevant visual occlusion quantitatively reflects its active functional dependence on visual geometry rather than static state shortcuts.

"state-dominant shortcut." Consequently, they discount high-dimensional visual streams, prioritizing low-dimensional joint dynamics. This *visual neglect* degrades physical deployment: policies struggle with spatial displacements and exhibit terminal alignment drift during contact-critical execution phases that demand fine-grained spatial feedback (Liu et al., 2020).

Prior remedies introduce trade-offs in efficiency or stability: unconstrained fine-tuning risks distorting pre-trained spatial representations, while explicit grounding or auxiliary reconstructive decoders incur substantial generative latency. Optimization-level interventions such as gradient adjustment (Lu et al., 2026) require delicate per-step scheduling and alter underlying training dynamics. Thus, a key question emerges: *How can we dynamically restore visual reliance during alignment-heavy execution phases without altering pre-trained representations or compromising real-time control?*

To address this, we propose **Phase-Adaptive Fusion (PAF)**, a lightweight, forward-only neural modulation plug-in ($< 3.5$ ms latency overhead). PAF decouples perceptual modulation into two complementary functional axes: *where to focus* and *when to amplify*. The spatial axis combines Semantic-Aware Gated Cross-Attention (SGCA) for instruction priming with a Lightweight Decomposed Semantic Alignment (LDSA) branch for localized spatial gating. The temporal axis uses a Temporal Phase Estimator to infer latent execution phases from recent proprioceptive states and action histories. These signals are synthesized via a bounded residual gating mechanism, dynamically reweighting visual tokens prior to multimodal fusion.

Extensive evaluations on simulation benchmarks (LIBERO, Meta-World, Robomimic) and physical SOARM101 deployments confirm that PAF effectively mitigates the state-dominant shortcut. In a challenging long-horizon stacking task, PAF actively counteracts visual neglect during terminal placement, reducing alignment failures and boosting overall success from 61.0% to 83.0% (+22.0% net gain, 95% CI: [9.6%, 33.5%]) over matched baselines.

The main contributions of this paper are:

- **Empirical Diagnosis:** We systematically formulate the state-dominant shortcut in VLA fine-tuning, demonstrating through controlled interventions how modality competition attenuates visual reliance during contact-critical manipulation.

- **Decoupled Spatiotemporal Architecture:** We introduce PAF, combining a semantic-geometric spatial branch with a kinematics-based temporal phase estimator to adaptively modulate visual features without requiring heavy generative decoders.
- **Non-Intrusive Integration and Rigorous Validation:** We design a bounded residual gate paired with a two-stage training recipe. Under schedule- and parameter-matched controls, PAF yields consistent gains across diverse backbones (SmolVLA, $\pi_0$), validated via simulation diagnostics and physical rollouts with 95% confidence intervals.

## 2 Related Work

### 2.1 Adapting VLA Foundation Models

Robotic manipulation increasingly leverages general-purpose Vision-Language-Action (VLA) foundation models (Brohan et al., 2022, 2023; Team et al., 2024; Kim et al., 2024) paired with expressive action heads, including ACT (Zhao et al., 2023), Diffusion Policy (Chi et al., 2023), and Flow Matching formulations (Lipman et al., 2022; Shukor et al., 2025). When adapting these backbones to downstream tasks, Parameter-Efficient Fine-Tuning (PEFT) (Hu et al., 2022; Houlsby et al., 2019) is widely used to prevent catastrophic forgetting. However, static weight updates cannot dynamically adapt feature reliance across high-frequency execution phases. Inspired by activation modulation (Zhang et al., 2023), PAF introduces dynamic, forward-only gain control over latent visual activations. By utilizing bounded residual gating, our framework preserves pre-trained representational manifolds while granting the adaptive responsiveness needed for contact-rich interaction.

### 2.2 Modality Imbalance and Shortcut Learning

Accurate visual grounding is essential for closed-loop spatial alignment (Desimone and Duncan, 1995). However, incorporating multiple heavy vision encoders or specialized semantic modules (e.g., DINOv2 (Oquab et al., 2023), SigLIP (Zhai et al., 2023), SemanticVLA (Li et al., 2026)) introduces substantial computational and memory overhead. More fundamentally, multi-stream networks frequently exhibit modality competition during joint optimization (Liu et al., 2025; Wang et al., 2020). In embodied policies, this manifests as a "state-dominant shortcut": compact proprioceptive histories provide an immediate reduction in imitation loss, causing policies to discount visual observations (Lee et al., 2019; Lu et al., 2026). While scaling data diversity mitigates spurious global correlations, task-specific adaptation still re-triggers state shortcuts during contact-critical execution stages.

### 2.3 Visual Grounding and Modality Balancing

Prior efforts counteract visual neglect via explicit grounding with external detectors (Huang et al., 2025; Li et al., 2024), Chain-of-Thought spatial coordinate generation (Zawalski et al., 2024; Deng et al., 2025), auxiliary reconstructive diffusion losses (Song et al., 2026), or proprioceptive gradient damping (GAP) (Lu et al., 2026). However, these approaches typically incur extra inference latency, alter training dynamics, or demand heavy auxiliary memory. In contrast, PAF provides a lightweight, forward-pass routing mechanism. Decoupling modulation into language-conditioned spatial grounding and multi-scale 1D temporal tracking (Lea et al., 2017), PAF mitigates shortcut learning while preserving natural optimization dynamics and maintaining sub-3.5 ms latency.

## 3 Methodology

The core objective of **Phase-Adaptive Fusion (PAF)** is to establish a lightweight yet robust mechanism for grounding a Vision-Language-Action (VLA) policy in physical manipulation environments. Unlike unconstrained fine-tuning approaches that risk internal representation collapse or catastrophic forgetting (Brohan et al., 2023; Kim et al., 2024), PAF operates as a modular, forward-pass plug-in that adaptively mitigates modality imbalance. We formalize this process through four progressive stages: Semantic Pre-conditioning, Spatial Grounding, Temporal Phase Estimation, and Phase-Adaptive Residual Modulation.

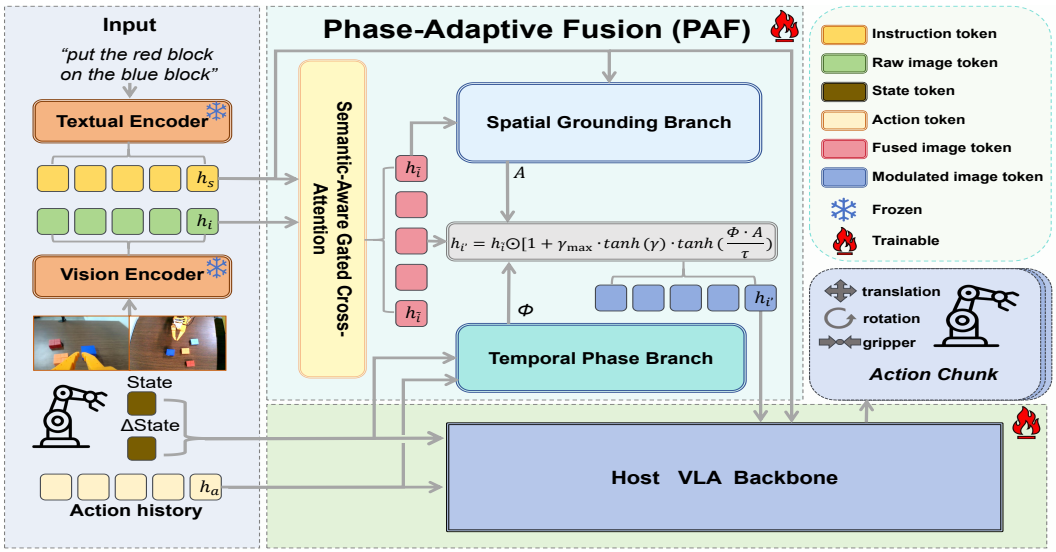



Figure 2: The detailed architecture of Phase-Adaptive Fusion (PAF). The framework modulates visual tokens by synthesizing temporal phase context and spatial semantic alignment through a bounded residual gate.

### 3.1 Semantic Pre-Conditioning and Multi-modal Alignment

Before routing visual tokens into the decoupled extraction branches, we establish an early semantic priming mechanism. Standard VLA foundation models typically process raw visual tokens $\mathbf{h}_i \in \mathbb{R}^{L_v \times D}$ and language tokens $\mathbf{h}_s \in \mathbb{R}^{L_s \times D}$ independently prior to deep prefix fusion, putting visual representations at an optimization disadvantage during behavior cloning.

To achieve early cross-modal awareness without injecting semantic noise into background patches, we introduce a **Semantic-Aware Gated Cross-Attention (SGCA)** module. Functionally distinct from downstream spatial masking, SGCA determines *what semantic content* to modulate by enriching visual tokens with language intent. First, we compute the raw cross-modal context $\mathbf{C} \in \mathbb{R}^{L_v \times D}$ by querying the language sequence:

$$\mathbf{C} = \text{Softmax}\left(\frac{(\mathbf{h}_i \mathbf{W}_Q)(\mathbf{h}_s \mathbf{W}_K)^\top}{\sqrt{d_k}}\right)(\mathbf{h}_s \mathbf{W}_V) \tag{1}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D \times d_k}$ are learnable projection matrices. In robotic manipulation, task-irrelevant background regions should be protected from semantic distractors. Thus, we compute a token-wise semantic gate $\mathbf{g}_{sem} \in \mathbb{R}^{L_v \times 1}$:

$$\mathbf{g}_{sem} = \sigma\left([\mathbf{h}_i \parallel \mathbf{C}]\, \mathbf{W}_{gate}\right) \tag{2}$$

where $\parallel$ denotes channel-wise concatenation, $\sigma$ is the Sigmoid function, and $\mathbf{W}_{gate} \in \mathbb{R}^{2D \times 1}$. The final semantically primed visual tokens are generated via a zero-initialized gated residual connection:

$$\mathbf{h}_{\tilde{i}} = \mathbf{h}_i + \alpha \cdot (\mathbf{g}_{sem} \odot \mathbf{C}) \tag{3}$$

where $\odot$ denotes the Hadamard product, and $\mathbf{g}_{sem}$ is implicitly broadcasted across the $D$-dimensional feature space of $\mathbf{C}$. Here, $\alpha$ is a learnable scalar strictly initialized to zero, ensuring an identity mapping at initialization to preserve the host VLA's pre-trained representation manifold (detailed micro-architecture is provided in Appendix A).

### 3.2 Dual-Branch Information Extraction

The central design of PAF decouples perceptual modulation into two complementary functional axes: identifying *where to focus* (spatial-geometric selection) and *when to amplify* (temporal phase tracking).

**Lightweight Decomposed Semantic Alignment (LDSA).** While SGCA enriches visual token semantics, the LDSA spatial branch isolates *where to modulate* by predicting localized spatial attention masks without altering token features. We project $\mathbf{h}_{\tilde{i}}$ and a global instruction embedding into a shared latent space factorized into $K$ independent attention heads. For each head $k \in \{1, \ldots, K\}$, the raw spatial affinity map $M_k \in \mathbb{R}^{L_v \times 1}$ is formulated as:

$$M_k = \frac{(\mathbf{h}_{\tilde{i}}\mathbf{W}_{q,k}) \left(\text{Pool}(\mathbf{h}_s)\mathbf{W}_{p,k}\right)^\top}{\sqrt{D/K}} \tag{4}$$

where $\text{Pool}(\cdot)$ denotes global average pooling. To incorporate local spatial inductive biases (Liu et al., 2021), we reshape the concatenated maps back into 2D spatial coordinates.

*Multi-View Token Reshaping Protocol:* When processing multi-camera inputs (e.g., third-person and wrist cameras), visual tokens form a concatenated sequence $L_v = \sum_{v=1}^{V} L_v^{(v)}$ across $V$ views. To preserve spatial topology without introducing artificial cross-view boundary artifacts, we retain each camera's original grid dimensions $(H_v^{(v)}, W_v^{(v)})$ based on its patch resolution. The token sequence is partitioned per view, stripped of view-specific tokens, and reshaped individually into separate 2D grids $\mathbb{R}^{H_v^{(v)} \times W_v^{(v)} \times K}$. Depthwise $3 \times 3$ convolutions (DWConv) are then applied independently within each camera grid, ensuring convolution kernels never span across camera boundaries:

$$A = \sigma\left(\text{DWConv}\left(\text{Unflatten}(M_1 \parallel \cdots \parallel M_K)\right)\right) \tag{5}$$

The modulated feature maps are flattened and concatenated back in sequence order. This preserves strict $\mathcal{O}(L_v)$ linear complexity while scaling seamlessly with multi-view visual token counts.

**Temporal Phase Estimator.** Manipulation tasks exhibit distinct physical phases. A robot's action history and kinematic momentum provide strong indicators for phase transitions (e.g., deceleration and trajectory inflection during pre-contact alignment). Alongside action history $a_{t-H:t-1}$ and proprioceptive state $s_t$, we explicitly incorporate the numerical derivative velocity $\Delta s_t = s_t - s_{t-1}$. We construct the multi-modal temporal context $\mathbf{X}_{temp} \in \mathbb{R}^{H \times d_{in}}$:

$$\mathbf{X}_{temp} = \left[s_{t-H} \parallel \Delta s_{t-H} \parallel a_{t-H}, \ldots, s_t \parallel \Delta s_t \parallel \mathbf{0}\right]^\top \tag{6}$$

Applying parallel 1D multi-scale convolutions ($k \in \{3, 5\}$) followed by Global Average Pooling (GAP) yields the scalar phase intensity $\phi \in (0, 1)$:

$$\phi = \sigma\left(\text{MLP}_{phase}\left(\text{GAP}\left(\text{Conv1D}_3(\mathbf{X}_{temp}) \parallel \text{Conv1D}_5(\mathbf{X}_{temp})\right)\right)\right) \tag{7}$$

*Architectural Boundary and Design Trade-off:* Decoupling the temporal branch to operate strictly on low-dimensional kinematic and action histories is an intentional design choice. By bypassing heavy spatio-temporal video encoders, PAF introduces less than 3.5 ms forward latency overhead. A structural trade-off of this kinematics-only formulation is that when the robot remains stationary while external objects are displaced, $\Delta s_t \approx \mathbf{0}$ and $\phi$ cannot independently register the external scene shift. However, because the visual-conditioned LDSA spatial branch continuously re-extracts spatial masks from incoming image frames, the policy adapts by dynamically redirecting spatial modulation rather than relying solely on temporal acceleration.

### 3.3 Phase-Adaptive Residual Modulation

The final stage synthesizes spatial semantic filters and temporal phase signals into a bounded residual gating formulation:

$$\mathbf{h}_{i'} = \mathbf{h}_{\tilde{i}} \odot \left[\mathbf{1} + \gamma_{\max} \cdot \tanh(\gamma) \cdot \tanh\left(\frac{\phi \cdot A_{exp}}{\tau}\right)\right] \tag{8}$$

where $\odot$ denotes element-wise multiplication, $A_{exp} \in \mathbb{R}^{L_v \times D}$ is the broadcasted spatial gate, $\gamma \in \mathbb{R}^D$ is a learnable scaling parameter, and $\tau$ is a temperature hyperparameter.

This formulation provides two structural properties:

1. **Spatio-Temporal Gating:** The joint term $\phi \cdot A_{exp}$ modulates visual features conditioned on both execution phase ($\phi$) and spatial alignment ($A_{exp}$), thereby mitigating the state-dominant shortcut during contact-critical interaction.

2. **Bounded Representation Preservation:** The nested hyperbolic tangents and $+1$ residual formulation strictly bound modulation gains within $[1-\gamma_{\max}, 1+\gamma_{\max}]$. This prevents gradient explosions and ensures that the modulation does not distort the host VLA's pre-trained representational manifold.

### 3.4 Staged Optimization Protocol via Parameter-Efficient Fine-Tuning

Directly inserting randomly initialized modulation modules into a pre-trained VLA backbone can induce covariate shift in activation space, destabilizing pre-trained representations. Indeed, empirical ablations reveal that single-stage end-to-end training without phase warm-up leads to optimization collapse ($76.8\%$ success on LIBERO), as uninitialized gates corrupt early action gradients. To ensure stable convergence, we introduce a structured two-stage optimization protocol incorporating Parameter-Efficient Fine-Tuning (PEFT).

Let the generative robotic policy be denoted by $\pi_\Theta$. To mitigate compounding trajectory errors in long-horizon execution (Zhao et al., 2023), the base policy is optimized via an action chunking objective over horizon $H_a$:

$$\mathcal{L}_{BC}(\Theta) = \mathbb{E}_{(\tau, \mathbf{A}_t^*)\sim\mathcal{D}} \left[ \frac{1}{H_a} \sum_{\delta=0}^{H_a-1} \left\| \mathbf{a}_{t+\delta}^* - \pi_\Theta \left( \mathbf{h}_{i'}, \mathbf{h}_s, s_t \right)_{t+\delta} \right\|_1 \right] \tag{9}$$

where $\mathbf{A}_t^*$ is the ground-truth action chunk and $\mathcal{D}$ denotes expert demonstrations.

**Stage I: Perceptual Warm-up with Modulation Regularization.** In Stage I (50k steps), the host VLA backbone $\bar{\Theta}_{VLA}$ is frozen while optimizing the PAF parameters $\Theta_{PAF}$. To prevent spatial and temporal gates from collapsing into degenerate local minima before cross-modal alignment stabilizes, we impose an $\ell_2$ penalty on the modulation scaling parameter $\gamma$:

$$\Theta_{PAF}^{(1)} = \arg\min_{\Theta_{PAF}} \left( \mathcal{L}_{BC}(\Theta_{PAF} \cup \bar{\Theta}_{VLA}) + \lambda_{reg} \|\gamma\|_2^2 \right) \tag{10}$$

The regularization term $\lambda_{reg}\|\gamma\|_2^2$ constrains the modulation magnitude, anchoring PAF close to an identity mapping during initial steps. This guarantees smooth gradient backpropagation through deep frozen layers and anchors spatial attention before downstream action adaptation.

**Stage II: Synergistic Adaptation via LoRA and Hybrid Tuning.** After Stage I establishes stable visual routing, the policy is adapted to downstream dynamics in Stage II (200k steps). For standard parameter-efficient fine-tuning, we apply LoRA (Hu et al., 2022) ($\Delta\mathbf{W} = \mathbf{BA}$) to the projection layers $\Theta_{LoRA}$, optimizing:

$$\min_{\Theta_{PAF}, \Theta_{LoRA}} \mathcal{L}_{BC} \left( \Theta_{PAF} \cup \left( \bar{\Theta}_{VLA} + \Theta_{LoRA} \right) \right) \tag{11}$$

with differential learning rates ($\eta_{PAF} > \eta_{LoRA}$). Furthermore, when computational budget permits, transitioning to a *Two-Stage Hybrid* strategy (applying LoRA to action expert layers during Stage I, followed by full fine-tuning of non-backbone projection heads in Stage II) further unlocks policy capacity, achieving superior downstream performance ($93.8\%$ on LIBERO) while avoiding optimization collapse.

## 4 Experimental Evaluation

### 4.1 Experimental Setup

**Environments and Baselines.** We evaluate PAF across simulation benchmarks and a real-world platform to assess its effectiveness in mitigating the state-dominant shortcut. We primarily employ **LIBERO** (Liu et al., 2023) for multi-task instruction following and **Meta-World** (Yu et al., 2020) to evaluate policy robustness across diverse manipulation dynamics. For real-world validation, we deploy policies on a physical **SOARM101** robotic arm to perform a multi-stage, contact-critical block stacking task. To demonstrate architectural generality, we integrate PAF into two representative open-source VLA foundation models: **SmolVLA** (2.25B) (Shukor et al., 2025) and $\pi_0$ (3.5B) (Black et al., 2024). Comprehensive details regarding physical hardware, teleoperation data collection, and baseline adaptation protocols are provided in Appendix A.

Table 1: Success rates (%) of VLA policies with and without PAF modulation across LIBERO and Meta-World benchmarks. 95% bootstrap confidence intervals for overall averages: $\pi_0$ [85.1%, 87.5%], $\pi_0$+PAF [88.3%, 90.7%]; SmolVLA [87.9%, 90.3%], SmolVLA+PAF [90.9%, 92.9%]. (Detailed results for Robomimic are provided in Appendix B).

| | **LIBERO Benchmark** | | | | | **Meta-World Benchmark** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Policy** | Spatial | Object | Goal | Long | **Avg.** | Easy | Medium | Hard | V-Hard | **Avg.** |
| $\pi_0$ | 90.3 | 86.2 | 95.3 | 73.4 | 86.3 | 80.4 | 40.9 | 36.7 | 44.3 | 50.5 |
| **$\pi_0$ + PAF** | **92.5** | **89.2** | **96.8** | **79.5** | **89.5** (+3.2) | **83.2** | **46.5** | **42.1** | **51.5** | **55.8** (+5.3) |
| SmolVLA | 93.3 | 94.2 | 91.8 | 77.4 | 89.1 | 87.1 | 51.8 | 70.4 | 64.0 | 68.3 |
| **SmolVLA + PAF** | **96.2** | **95.5** | **91.3** | **84.7** | **91.9** (+2.8) | **89.5** | **56.4** | **75.2** | **71.4** | **73.1** (+4.8) |

**Evaluation Metrics and Statistical Rigor.** Manipulation performance is quantified via the task **Success Rate (%)**. To ensure statistical rigor, simulated policies are evaluated across 5 random seeds with 1,000 evaluation rollouts per task under standardized initial-state distributions. We report 95% Confidence Intervals (CIs) computed via task-stratified paired bootstrap with 10,000 resamples (on Meta-World, SmolVLA and PAF attain $68.3 \pm 1.4\%$ and $73.1 \pm 1.2\%$, respectively). For physical robot deployments, we evaluate 100 trials under nominal configurations alongside 30 controlled physical perturbation trials per condition, reporting Wilson score 95% binomial confidence intervals.

### 4.2 Simulation Results: Multi-Task and Generalization

We first evaluate PAF against baseline policies in simulation. As summarized in Table 1, integrating PAF yields consistent improvements across both the $\pi_0$ (3.5B) and SmolVLA (2.25B) backbones. On LIBERO, SmolVLA+PAF achieves an average success rate of 91.9% (95% CI: [90.9%, 92.9%]), outperforming the frozen baseline by +2.8% (with a +7.3% gain on long-horizon tasks). On Meta-World, PAF improves SmolVLA from 68.3% to 73.1% (+4.8%), indicating enhanced spatial responsiveness during contact-rich trajectory execution. Parallel gains are observed on the 3.5B $\pi_0$ backbone (+3.2% on LIBERO, +5.3% on Meta-World). (Evaluations on Robomimic using recurrent policies are detailed in Appendix B, confirming cross-architecture applicability).

**Fair Adaptation Baseline Controls.** To verify that these gains stem from phase-adaptive spatial modulation rather than simply additional training steps or parameter capacity, we evaluate strictly controlled adaptation baselines under identical 250k-step compute budgets, data orders, and optimizers:

- *Schedule-Matched Control (No PAF):* Training SmolVLA with the identical two-stage schedule (50k warmup + 200k joint LoRA $r = 32$) without PAF achieves **89.6%** on LIBERO. PAF provides a +2.3% net margin under the identical adaptation schedule.
- *Parameter-Matched Control ($r = 64$):* Expanding the LoRA rank to match the total trainable parameter count of the PAF framework ($\sim$3.2M parameters) attains **90.1%** on LIBERO. Full PAF outperforms this capacity-matched control by +1.8%, confirming that gains are architectural rather than a byproduct of extra capacity.
- *Advanced Adaptation Recipe:* When transitioning to a Two-Stage Hybrid protocol (LoRA warmup in Stage I followed by non-backbone fine-tuning in Stage II), SmolVLA+PAF achieves **93.8%** overall average on LIBERO (and 87.6% on LIBERO-10), highlighting significant scaling potential.

**Direct Diagnostics for Restored Visual Reliance.** On the standardized LIBERO-Plus diagnostic benchmark (1,000 rollouts per condition), the baseline collapses under proprioceptive Gaussian noise (35.3%) and spatial target shifts (48.5%), while remaining relatively indifferent to severe visual occlusion (75.4%). In contrast, PAF maintains robust closed-loop tracking under state noise (68.7%, +33.4%) and target shifts (85.1%, +36.6%), while exhibiting expected degradation under severe task-relevant occlusion (28.6%). Quantifying this shift via a behavioral visual-reliance proxy $R_{\text{vis}}^{\text{drop}} = (\text{Succ}_{\text{clean}} - \text{Succ}_{\text{mask}})/\text{Succ}_{\text{clean}}$ reveals an increase from 0.20 for SmolVLA to 0.72 for PAF, confirming a mechanistic restoration of visual feature utilization during contact phases.

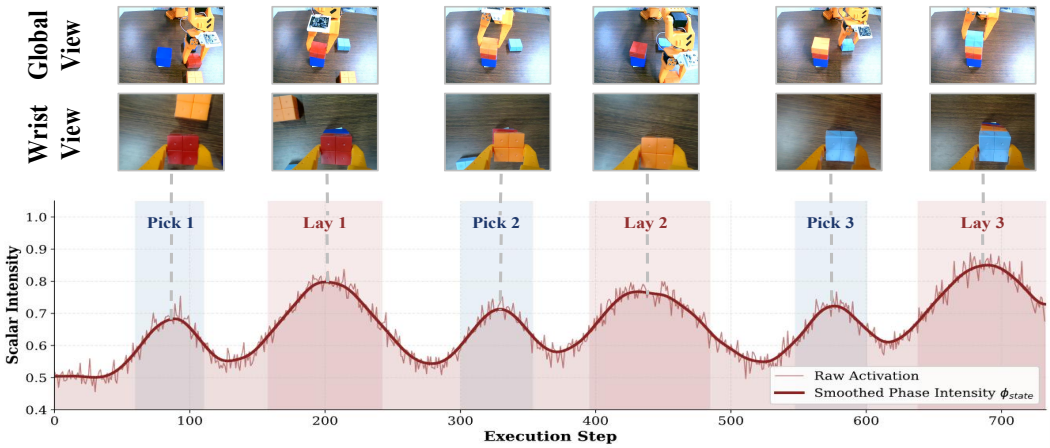



Figure 3: **Temporal Gating Mechanism of PAF during Real-World Execution.** (Top) Execution snapshots from global and wrist cameras on the SOARM101 platform. (Bottom) Synchronized scalar phase intensity $\phi$ over time. Raw activations (light red) are plotted with smoothed trends (dark red). The temporal branch dynamically produces six distinct peaks closely synchronized with contact-critical "Pick" and "Lay" phases (shaded regions), dynamically gating visual sensitivity when kinematic trajectory priors require visual correction.

### 4.3 Real-World Deployment and Mechanism Analysis

We evaluate PAF on the physical SOARM101 platform on a multi-stage, long-horizon block stacking sequence (Red $\rightarrow$ Orange $\rightarrow$ Blue). This task requires fine-grained contact alignment and compounds spatial errors across successive insertions. The baseline SmolVLA achieves a **61.0% success rate** (61/100, Wilson 95% CI: [51.2%, 70.0%]). As illustrated in the cumulative survival trajectory (Fig. 4a), while baseline performance is reliable during initial pick phases, its survival rate decreases steeply as stacking height increases. A categorical failure analysis (Fig. 4b) reveals that failures are predominantly caused by **terminal alignment drift**. This empirical behavior exemplifies modality imbalance: the policy over-relies on low-dimensional joint trajectory priors, failing to leverage visual feedback to compensate for millimeter-level spatial offsets.

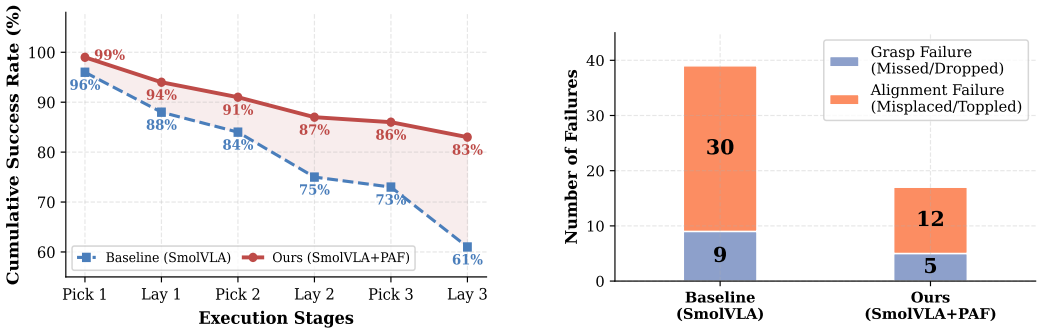
In contrast, the PAF-modulated policy achieves an **83.0% overall success rate** (83/100, Wilson 95% CI: [74.5%, 89.1%]), representing a net gain of +22.0% (Newcombe 95% CI: [9.6%, 33.5%]). To examine the underlying mechanism, we track the temporal gating activation $\phi$ during execution. As depicted in Fig. 3, rather than maintaining a static modulation weight, $\phi$ dynamically generates **six distinct peaks** closely synchronized with contact-critical pick and lay events. During unconstrained free-space transit, $\phi$ remains suppressed, protecting the representation from background clutter. As the end-effector approaches the target, $\phi$ escalates, triggering adaptive spatial amplification through the LDSA gate. This synchronized modulation facilitates continuous visual feedback correction, substantially mitigating terminal alignment drift.

**Physical Perturbation Robustness.** In 30 physical perturbation trials per condition, PAF demonstrates resilience against unexpected spatial displacements: under target position shifts of 2–4 cm, PAF achieves 76.7% success versus 43.3% for the baseline (+33.4%); under workspace clutter, it achieves 80.0% versus 53.3% (+26.7%). Under severe physical occlusion of the target object, PAF's success decreases to 26.7% (vs. 46.7% for the baseline), further corroborating that the policy actively depends on visual geometry rather than blind kinematic replay.

### 4.4 Ablation and Discussion

To isolate component contributions under fixed training budgets, we conduct ablations on SmolVLA (Table 2).

**Optimization Schedule: Preventing Optimization Collapse.** Abandoning the two-stage protocol and fine-tuning the uninitialized PAF plug-in jointly with the backbone from scratch (w/o 2-Stg.) leads




(a) Cumulative Success Rate

(b) Failure Mode Breakdown

Figure 4: **Robustness and failure analysis in long-horizon deployment.** (a) Cumulative survival curves indicate that PAF mitigates compounding alignment errors across the 3-stage stacking sequence. (b) Categorical failure breakdown confirms that PAF's phase-adaptive modulation substantially reduces terminal alignment drift compared to the baseline.

Table 2: Ablation study of PAF components on the SmolVLA backbone across LIBERO and Meta-World benchmarks. *SGCA*: Semantic-Aware Gated Cross-Attention; *Spat.*: Spatial Grounding Branch; *Temp.*: Temporal Phase Branch; *2-Stg.*: Two-stage Training Protocol. (Ablation results on $\pi_0$ are detailed in Appendix B.3).

| Variant | SGCA | Spat. Br. | Temp. Br. | 2-Stg. Trng. | LIBERO (Avg. %) | Meta-World (Avg. %) |
|---|---|---|---|---|---|---|
| Baseline (Frozen SmolVLA) | – | – | – | – | 89.1 | 68.3 |
| Schedule-Matched Control ($r = 32$, No PAF) | – | – | – | ✓ | 89.6 | 69.4 |
| Parameter-Matched Control ($r = 64$, No PAF) | – | – | – | ✓ | 90.1 | 69.8 |
| w/o 2-Stage Protocol (Single-Stage Tuning) | ✓ | ✓ | ✓ | – | 76.8 | 64.2 |
| w/o Temporal Branch (Static Spatial) | ✓ | ✓ | – | ✓ | 89.8 | 69.1 |
| w/o Spatial Branch (Global Temporal) | ✓ | – | ✓ | ✓ | 89.4 | 69.7 |
| w/o SGCA Module | – | ✓ | ✓ | ✓ | 90.6 | 71.4 |
| Replace SGCA with 2nd LDSA Block | – | 2× | ✓ | ✓ | 90.8 | 71.6 |
| **Full PAF (Ours, Two-Stage LoRA)** | ✓ | ✓ | ✓ | ✓ | **91.9** | **73.1** |
| **Full PAF (Ours, Two-Stage Hybrid)** | ✓ | ✓ | ✓ | ✓ | **93.8** | **74.5** |

to severe optimization instability, with success dropping sharply to 76.8% on LIBERO and 64.2% on Meta-World. This underscores that introducing unconstrained modulation without perceptual warm-up corrupts early action gradients and disrupts pre-trained representations. Stage I warm-up anchors modulation near identity, providing a stable foundation for Stage II adaptation.

**Functional Disentanglement and Gating Alternatives.** Decoupling modulation into spatial (*where*) and temporal (*when*) paths is central to PAF:

- *Missing Spatial Branch (w/o Spatial):* Removing the spatial branch yields 89.4% on LIBERO. Without spatial localization, the model applies a uniform temporal gain, amplifying background distractors alongside task targets and resulting in misaligned grasps.
- *Missing Temporal Branch (w/o Temporal):* Removing the temporal branch reduces Meta-World performance to 69.1%. A static spatial mask fails to provide phase-dependent modulation, dampening dynamic responsiveness during high-speed transitions.
- *Simpler Gating Controls:* Benchmarking against simpler baselines under identical budgets confirms that neither constant phase scalars ($\phi = 1.0$, 90.0%), shuffled phases ($\phi_{\text{rand}}$, 89.3%), proprioceptive dropout ($p = 0.2$, 90.5%), nor gradient adjustment approaches (GAP, 90.8%) match full PAF (91.9%), confirming the necessity of synchronized spatiotemporal modulation.

**Non-Redundancy of Semantic Priming (SGCA vs. LDSA).** A natural question is whether SGCA is redundant given the downstream LDSA module. Removing SGCA (w/o SGCA) reduces performance to 90.6% on LIBERO. Furthermore, replacing SGCA with an additional LDSA block achieves only 90.8%, failing to reach full PAF (91.9%). This demonstrates that repeating spatial attention maps cannot substitute for token-level semantic enrichment: SGCA enriches visual representations with instruction semantics, whereas LDSA performs localized geometric selection. Both components operate complementarily to achieve robust closed-loop grounding.

## 5 Conclusion and Limitations

In this work, we presented Phase-Adaptive Fusion (PAF), a modular, forward-only neural adaptation plug-in designed to mitigate modality imbalance during Vision-Language-Action (VLA) fine-tuning. By decoupling perceptual modulation into semantic-geometric spatial grounding and kinematics-based temporal phase estimation, PAF adaptively modulates visual representations prior to multimodal fusion. Comprehensive evaluations across simulation benchmarks (LIBERO, Meta-World, Robomimic) and physical robot deployments demonstrate that dynamic visual gating during contact-critical execution phases effectively mitigates the state-dominant shortcut, reducing terminal alignment drift without modifying the pre-trained backbone or compromising real-time inference latency.

**Limitations.** Despite these improvements, the current formulation of PAF has several structural and empirical boundaries that warrant explicit discussion:

- *Kinematics-Only Temporal Phase Boundary:* As the temporal estimator relies exclusively on proprioceptive states, numerical velocities, and action histories ($\mathbf{X}_{temp}$), it cannot independently detect external environmental variations that occur while the robot remains stationary ($\Delta s_t \approx \mathbf{0}$, such as dynamic target displacements or external human interventions). While the spatial branch continuously re-extracts localized masks from incoming visual observations to redirect attention, generating temporal phase transitions driven purely by exogenous scene dynamics without robot motion remains beyond the current design.
- *Representation Bottlenecks and Severe Occlusion:* As a forward residual gating plug-in, PAF's modulation capacity is bounded by the feature manifold of the host vision encoder; it cannot synthetically reconstruct fine geometric details when task-critical objects undergo severe, prolonged visual occlusions.
- *Scalar Temporal Expressivity and Static Conditioning:* Compressing temporal phase into a scalar $\phi$ is computationally efficient ($< 3.5$ ms overhead) for single-arm manipulation, but lacks the expressivity required for asynchronous bimanual coordination. Furthermore, spatial conditioning relies on a global textual embedding, which lacks the temporal granularity to steer multi-stage execution across long-horizon compositional tasks.
- *Scope of Physical Validation:* While physical evaluations confirm substantial gains on contact-rich stacking and cap insertion, real-world experiments are conducted on a single robot platform with specific task families. Broader generalization across diverse morphologies, multi-finger dexterous hands, and dynamic in-the-wild environments requires further validation.

**Future Work.** To address these limitations, our future efforts will focus on three key directions. First, we plan to extend the temporal estimator into a dynamic semantic router: by decomposing compositional instructions into sequential sub-goals, inferred phase transitions can trigger autonomous shifts in active language tokens fed to the cross-attention module. Second, inspired by emerging tactile and force-torque modalities, we aim to integrate high-frequency force transients into the temporal branch to detect physical contact onsets, while routing taxel-array tactile maps to the spatial branch for fine-grained contact geometry. Finally, we plan to validate PAF across diverse multi-arm platforms and complex contact-rich industrial assembly pipelines.

## References


Black, K. et al. (2024). $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.

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. et al. (2022). RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Chi, C. et al. (2023). Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*.

Deng, S., Yan, M., Wei, S., Ma, H., Yang, Y., Chen, J., Zhang, Z., Yang, T., Zhang, X., Zhang, W., et al. (2025). Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. *arXiv preprint arXiv:2505.03233*.

Desimone, R. and Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18:193–222.

Houlsby, N. et al. (2019). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*.

Hu, E. J. et al. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Huang, H., Chen, X., Chen, Y., Li, H., Han, X., Wang, Z., Wang, T., Pang, J., and Zhao, Z. (2025). Roboground: Robotic manipulation with grounded vision-language priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22540–22550.

Kim, M. J. et al. (2024). OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Lea, C. et al. (2017). Temporal convolutional networks for action segmentation and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lee, M. A., Zhu, Y., Srinivasan, K., Shah, P., Savarese, S., Fei-Fei, L., Garg, A., and Bohg, J. (2019). Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8943–8950. IEEE.

Li, W., Zhang, R., Shao, R., Fang, Z., Zhou, K., Tian, Z., and Nie, L. (2026). Semanticvla: Semantic-aligned sparsification and enhancement for efficient robotic manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18397–18405.

Li, Z., Ren, L., Yang, J., Zhao, Y., Wu, X., Xu, Z., Bai, X., and Zhao, H. (2024). Vip: Vision instructed pre-training for robotic manipulation. *arXiv preprint arXiv:2410.07169*.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2022). Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. (2023). Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791.

Liu, C. et al. (2025). Modality-balancing preference optimization of large multimodal models by adversarial negative mining. *arXiv preprint arXiv:2506.08022*.

Liu, Y. et al. (2020). Bioinspired embodiment for intelligent sensing and dexterity in fine manipulation: A survey. *IEEE Transactions on Industrial Informatics*, 16(7):4308–4321.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pages 9992–10002. Ieee.

Lu, J., Xia, W., Wu, Y., et al. (2026). When would vision-proprioception policies fail in robotic manipulation? *arXiv preprint arXiv:2602.12032*.

Mandlekar, A. et al. (2021). What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*.

Oquab, M. et al. (2023). DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Shukor, M. et al. (2025). SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*.

Song, W., Zhou, Z., Zhao, H., Chen, J., Ding, P., Yan, H., Huang, Y., Tang, F., Wang, D., and Li, H. (2026). Reconvla: Reconstructive vision-language-action model as effective robot perceiver. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18549–18557.

Team, O. M. et al. (2024). Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*.

Wang, W., Tran, D., and Feiszli, M. (2020). What makes training multi-modal classification networks hard? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12692–12702. IEEE.

Yu, T. et al. (2020). Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*. PMLR.

Zawalski, M., Chen, W., Pertsch, K., Mees, O., Finn, C., and Levine, S. (2024). Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*.

Zhai, X. et al. (2023). Sigmoid loss for language-image pre-training (SigLIP). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhang, L. et al. (2023). Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. (2023). Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*.

# A Implementation Details and Architectures

To ensure complete reproducibility of our Phase-Adaptive Fusion (PAF) framework, we provide comprehensive specifications regarding the micro-architectures, multi-camera processing protocols, baseline training configurations, hyperparameter sensitivity, and physical experimental setups.

## A.1 Micro-Architecture of PAF Components

**Semantic-Aware Gated Cross-Attention (SGCA).** As formulated in Section 3.1, SGCA provides early language-conditioned visual priming prior to spatial and temporal extraction. The hidden dimension is set to $D = 768$ to match the host VLA's visual token dimension. Visual tokens $\mathbf{h}_i \in \mathbb{R}^{L_v \times D}$ query the text instruction tokens $\mathbf{h}_s \in \mathbb{R}^{L_s \times D}$. The linear projections $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ map inputs to a sub-dimension $d_k = 128$, producing the cross-modal context $\mathbf{C} \in \mathbb{R}^{L_v \times D}$. To prevent task-irrelevant background patches from absorbing semantic noise, a token-wise gate $\mathbf{g}_{sem} \in \mathbb{R}^{L_v \times 1}$ is computed via $\mathbf{W}_{gate} \in \mathbb{R}^{2D \times 1}$ followed by a Sigmoid function. The residual scalar $\alpha$ is strictly initialized to $0.0$ and optimized with an $\ell_2$ weight decay of $1 \times 10^{-4}$. This zero-initialization ensures an exact identity mapping ($\mathbf{h}_{\tilde{i}} = \mathbf{h}_i$) at the start of training, preserving the host VLA's pre-trained representation manifold.

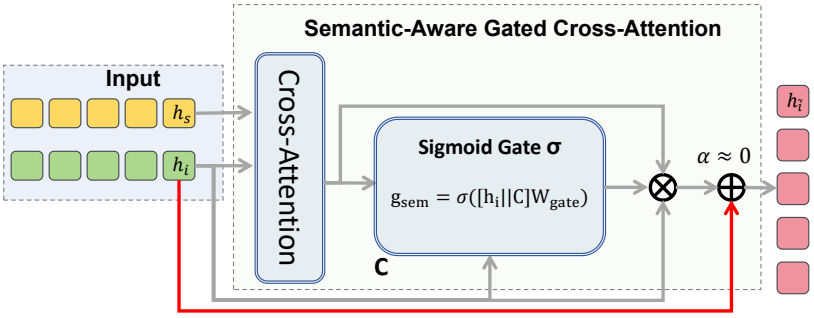



Figure 5: Micro-architecture of the SGCA module. The identity mapping is preserved at initialization via the zero-initialized scalar $\alpha = 0$, shielding the pre-trained feature manifold from early optimization shock.

**Lightweight Decomposed Semantic Alignment (LDSA).** The spatial branch predicts localized attention masks via low-rank cross-modal decomposition:

1. **Decomposed Projection:** The instruction tokens are pooled into a global embedding $\bar{\mathbf{h}}_s = \text{Pool}(\mathbf{h}_s) \in \mathbb{R}^{1 \times D}$. Both $\mathbf{h}_{\tilde{i}}$ and $\bar{\mathbf{h}}_s$ are projected into $K = 8$ independent attention heads, each operating in dimension $d_h = D/K$. Head-wise inner products produce raw affinity maps $M \in \mathbb{R}^{L_v \times K}$.
2. **Local Geometric Biases:** The token sequence is reshaped into a 2D spatial grid $\mathbb{R}^{H_v \times W_v \times K}$ (e.g., $16 \times 16$ or $24 \times 24$, determined by the host patch resolution). A Depth-Wise Convolution (DWConv) with a $3 \times 3$ kernel, stride 1, and padding 1 is applied independently across the $K$ channels to refine boundary contours without mixing semantic representations across heads.
3. **Output Gate:** The refined feature map is flattened and passed through a Sigmoid activation, yielding the spatial gate $A \in \mathbb{R}^{L_v \times K}$.

**Multi-View Token Reshaping Protocol.** For multi-camera configurations (e.g., static third-person camera and eye-in-hand camera), visual tokens form a concatenated sequence $L_v = \sum_{v=1}^{V} L_v^{(v)}$ across $V$ views. To preserve spatial topology without introducing artificial cross-camera boundary artifacts:

1. Each camera's original grid dimensions $(H_v^{(v)}, W_v^{(v)})$ are retained based on its patch tokenizer.

2. The concatenated sequence is partitioned per camera view, stripped of view-specific delimiter tokens, and reshaped individually into separate 2D grids $\mathbb{R}^{H_v^{(v)} \times W_v^{(v)} \times K}$.
3. Depthwise $3 \times 3$ convolutions are executed independently within each camera grid, ensuring no convolution kernel ever spans across distinct camera boundaries.
4. Modulated feature grids are flattened and concatenated back in original sequence order.

As summarized in Table 3, LDSA maintains strictly linear $\mathcal{O}(L_v)$ computational complexity with respect to visual token counts, introducing minimal latency overhead as additional views or higher patch resolutions are incorporated.

Table 3: Computational scaling and inference latency of PAF across camera configurations on an NVIDIA RTX 4090 GPU.

| **Camera Configuration** | **Rel. Tokens** | **LIBERO Avg. (%)** | **Latency Overhead** |
|---|---|---|---|
| One Third-Person View | 0.5× | 89.7 | 2.1 ms |
| Third-Person + Wrist (Default Setup) | 1.0× | 91.9 | 3.5 ms |
| Three Views (Third-Person + Dual Wrist) | 1.5× | 92.1 | 4.8 ms |
| Two Views (High-Resolution Patches) | 2.25× | 92.2 | 7.1 ms |

**Temporal Phase Estimator.** The temporal branch extracts latent execution phases from low-dimensional kinematic histories:

1. **Input Context:** The temporal window size is set to $H = 12$ steps. At timestep $t$, the state vector $s_t$ comprises 7 dimensions (6-DoF end-effector pose + 1D gripper state). The input matrix $\mathbf{X}_{temp}$ stacks state $s$, numerical velocity $\Delta s_t = s_t - s_{t-1}$, and past actions $a$, forming a sequence tensor of shape $\mathbb{R}^{H \times 21}$.
2. **Multi-Scale 1D Convolutions:** $\mathbf{X}_{temp}$ is transposed to $\mathbb{R}^{21 \times H}$ and passed through parallel 1D convolutional layers with kernel sizes $k \in \{3, 5\}$ and zero-padding ($p = 1$ and $p = 2$, respectively), projecting channels to $d_{conv} = 64$.
3. **Aggregation and Output:** Channel-wise concatenated features ($\mathbb{R}^{128 \times H}$) are compressed across time via Global Average Pooling (GAP) to an embedding of size $\mathbb{R}^{128}$. A 2-layer MLP with GELU activations outputs the scalar phase intensity $\phi \in (0, 1)$ via a Sigmoid function.

### A.2 Training Protocols and Adaptation Recipes

All models are trained on a workstation equipped with a single NVIDIA RTX 6000 Pro (96GB) GPU using mixed-precision training (bfloat16). Optimization employs the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay of $0.01$. The learning rate follows a cosine annealing schedule with a 1,000-step linear warmup.

**Stage I (Perceptual Warm-up).** Over 50,000 optimization steps with an effective batch size of 256 (achieved with a per-step micro-batch size of 32 and 8 gradient accumulation steps), the host VLA parameters $\bar{\Theta}_{VLA}$ remain frozen. We exclusively optimize $\Theta_{PAF}$ with a peak learning rate of $\eta_1 = 1 \times 10^{-4}$, initial scaling parameter $\gamma_0 = 0.05$, and an explicit $\ell_2$ penalty $\lambda_{reg}\|\gamma\|_2^2 = 0.01$ on the residual modulation scaling parameter. This regularizer constrains the modulation gain close to an identity mapping, anchoring spatial attention before action fine-tuning.

**Stage II (Joint LoRA Adaptation).** Over 200,000 optimization steps with an effective batch size of 32 (micro-batch size of 16 with 2 gradient accumulation steps), we apply LoRA ($r = 32$, $\alpha = 64$, dropout $0.05$) to the attention projections ($q_{proj}$, $v_{proj}$) of the VLA backbone. We employ differential learning rates: the host LoRA parameters are updated conservatively at $\eta_{LoRA} = 2.5 \times 10^{-5}$ to prevent distortion of pre-trained spatial representations, while PAF parameters are fine-tuned at $\eta_{PAF} = 5.0 \times 10^{-5}$ to ensure agile cross-modal synchronization.

**Two-Stage Hybrid Strategy (Advanced Recipe).** In this variant, Stage I optimizes PAF while applying LoRA ($r = 32$) to the action expert layers (`lm_expert`) to stabilize cross-modal alignment without distorting action heads. Stage II transitions to full fine-tuning of all non-backbone components (action expert, state projections, and PAF) while keeping only the vision backbone frozen. As demonstrated in Section 4.2, this protocol achieves state-of-the-art performance on LIBERO (93.8%) while circumventing the optimization collapse observed during single-stage training.

Table 4: Key hyperparameter configurations for PAF across backbones.

| **Hyperparameter** | **Value** | **Hyperparameter** | **Value** |
|---|---|---|---|
| Action Chunk Size ($H_a$) | 12 | Optimizer | AdamW |
| Temporal Window Size ($H$) | 12 | LR Schedule | Cosine Annealing |
| Number of LDSA Heads ($K$) | 8 | Warmup Steps | 1,000 |
| Stage I Learning Rate ($\eta_1$) | $1 \times 10^{-4}$ | LoRA Rank ($r$) | 32 |
| Stage II LoRA LR ($\eta_{LoRA}$) | $2.5 \times 10^{-5}$ | LoRA Alpha ($\alpha$) | 64 |
| Stage II PAF LR ($\eta_{PAF}$) | $5.0 \times 10^{-5}$ | LoRA Dropout | 0.05 |
| Modulation Regularization ($\lambda_{reg}$) | 0.01 | Effective Batch Size (Stage I / II) | 256 / 32 |
| Temperature ($\tau$) | 0.1 | Micro-Batch (Stage I / II) | 32 / 16 |
| Initial Modulation Scale ($\gamma_0$) | 0.05 | Grad Accum. Steps (Stage I / II) | 8 / 2 |
| Modulation Boundary ($\gamma_{\max}$) | 1.0 | Weight Decay | 0.01 |

### A.3 Hyperparameter Sensitivity Analysis

To examine whether PAF relies on narrow hyperparameter tuning, we conduct systematic perturbations on the SmolVLA backbone across key architecture and optimization dimensions on LIBERO. As summarized in Table 5, performance remains within a narrow 0.7% band across diverse temporal windows, temperatures, modulation scales, and LoRA ranks, confirming that PAF's empirical gains stem from its spatiotemporal gating structure rather than delicate tuning.

Table 5: Hyperparameter sensitivity analysis of PAF on LIBERO (Average Success Rate %).

| **Hyperparameter Dimension** | **Tested Values** | **LIBERO Avg. (%)** | **Spread (%)** |
|---|---|---|---|
| Temporal Window ($H$) | 8 / 12 / 16 | 91.3 / 91.9 / 91.6 | 0.6% |
| Temperature ($\tau$) | 0.05 / 0.1 / 0.2 | 91.5 / 91.9 / 91.6 | 0.4% |
| Initial Modulation Scale ($\gamma$) | 0.01 / 0.05 / 0.1 | 91.2 / 91.9 / 91.4 | 0.7% |
| LoRA Rank ($r$) | 16 / 32 / 64 | 91.3 / 91.9 / 91.8 | 0.6% |

### A.4 Framework Alignment for Foundation Models

To ensure rigorous evaluation fairness, all experiments for both SmolVLA (2.25B) and $\pi_0$ (3.5B) were re-implemented within the open-source **LeRobot PyTorch** framework. This standardization guarantees that both models share identical data loaders, image normalization, chunking loss formulations, and evaluation rollout harnesses. The absolute scores reported for $\pi_0$ reflect this conservative, uniform PyTorch adaptation baseline rather than specialized JAX/OpenPI training setups. Crucially, even under this standardized evaluation regime, PAF yields consistent orthogonal improvements over the baseline (+3.2% on LIBERO, +5.3% on Meta-World), demonstrating foundation-agnostic efficacy.

### A.5 Physical Robot Hardware and Task Setup

Physical experiments were conducted on a SOARM101 6-DoF manipulator equipped with a parallel gripper. The vision setup includes a stationary third-person camera ($640 \times 480$) overlooking the workspace and an eye-in-hand camera mounted on the wrist. Control frequency is maintained at 50 Hz.

**Task Specifications.**

1. **Long-Horizon Block Stacking:** The robot stacks three colored blocks (Red $\rightarrow$ Orange $\rightarrow$ Blue) sequentially. The base block is positioned on a work surface, and the robot must

execute successive pick-and-place trajectories where minor orientation offsets accumulate across stages. The expert dataset contains 800 teleoperated trajectories collected via a leader-follower setup.

2. **High-Precision Pen Grasping and Cap Insertion:** To evaluate fine-grained contact dynamics, the robot grasps a standard pen (1.0 cm diameter) and inserts it into a narrow cap opening (1.3 cm diameter, providing a tight 1.5 mm tolerance).

**Physical Perturbation Protocols.** To assess real-world resilience, policies are evaluated across 30 controlled trials per condition:

- *Target Position Shifts:* The base receptacle or target pen/cap is displaced by 2–4 cm from nominal training locations while proprioceptive history reflects nominal approaches.
- *Workspace Clutter:* Visual distractors (unrelated colorful objects, tools) are placed adjacent to manipulation targets.
- *Visual Occlusions:* Partial occlusion is introduced via semi-transparent barriers; severe task-relevant occlusion is introduced by physically blocking line-of-sight to the target aperture during terminal insertion.

# B Extended Experimental Evaluations and Controlled Baselines

## B.1 Fair Adaptation Controls and Parameter Parity

To address concerns regarding whether PAF's gains stem from additional training steps or parameter capacity, we evaluate strictly controlled adaptation baselines under identical data orders, 250k total step budgets, and hardware environments:

1. *Schedule-Matched Control ($r = 32$):* SmolVLA trained with the identical two-stage schedule (50k warmup + 200k joint LoRA) without PAF.
2. *Parameter-Matched Control ($r = 64$):* LoRA rank increased to $r = 64$ for the no-PAF baseline, exactly matching the $\sim$3.2M additional trainable parameter count of the PAF framework.

As detailed in Table 6, full PAF provides consistent net performance advantages over both schedule-matched (+2.3%) and parameter-matched (+1.8%) controls, confirming that improvements are attributable to phase-adaptive spatiotemporal modulation rather than fine-tuning capacity.

Table 6: Benchmark comparisons on LIBERO across schedule- and parameter-matched adaptation controls.

| Method / Adaptation Protocol | Spatial | Object | Goal | LIBERO-10 | Overall Avg. |
|---|---|---|---|---|---|
| SmolVLA (Frozen Baseline) | 93.3 | 94.2 | 91.8 | 77.4 | 89.1 |
| SmolVLA + Standard LoRA ($r = 32$) | 93.4 | 94.5 | 91.9 | 77.8 | 89.4 |
| SmolVLA + Standard LoRA ($r = 64$) | 93.8 | 94.8 | 92.4 | 78.6 | 89.9 |
| SmolVLA + 2-Stage LoRA ($r = 32$, No PAF) | 93.6 | 94.7 | 92.0 | 78.1 | 89.6 |
| SmolVLA + 2-Stage LoRA ($r = 64$, Matched) | 93.9 | 95.1 | 92.2 | 79.5 | 90.1 |
| **SmolVLA + PAF (2-Stage LoRA, Ours)** | 96.2 | 95.5 | 91.3 | 84.7 | **91.9** |
| **SmolVLA + PAF (2-Stage Hybrid, Best)** | **97.6** | **96.8** | **93.2** | **87.6** | **93.8** |

## B.2 LIBERO-Plus Systematic Diagnostic Suite

To provide direct causal evidence regarding modality reliance, we evaluate SmolVLA and SmolVLA+PAF on the standardized **LIBERO-Plus** diagnostic benchmark suite across 1,000 evaluation rollouts per condition (Table 7).

These evaluations reveal an explicit signature of restored visual reliance:

- **Kinematic Noise and Target Shifts:** Baseline SmolVLA relies heavily on proprioceptive state continuity and degrades sharply under state noise (35.3%) and target displacements

Table 7: Controlled diagnostic evaluations on the standardized LIBERO-Plus suite (1,000 rollouts per condition).

| Diagnostic Condition (LIBERO-Plus) | SmolVLA (Base) | SmolVLA + PAF | Δ (Abs.) |
|---|---|---|---|
| Clean Workspace (Standard Control) | 88.2% | 94.6% | +6.4% |
| Workspace Clutter (Visual Distractors) | 80.4% | 92.0% | +11.6% |
| Severe Visual Occlusion (50%+ Image Mask) | 75.4% | 28.6% | -46.8% |
| Proprioceptive State Noise (Kinematic Gaussian) | 35.3% | 68.7% | +33.4% |
| Visual Target Shift (2–4 cm, Fixed State) | 48.5% | 85.1% | +36.6% |

(48.5%). PAF actively tracks the displaced visual target, maintaining high success under shifts (85.1%) and state noise (68.7%).

- **Visual Reliance Metric:** We quantify behavioral reliance via the visual drop ratio $R_{\text{vis}}^{\text{drop}} = (\text{Succ}_{\text{clean}} - \text{Succ}_{\text{mask}})/\text{Succ}_{\text{clean}}$. SmolVLA scores $0.20$ (indicating relative indifference to visual inputs), whereas PAF achieves $0.72$, indicating functional visual reliance during task execution.
- **Gradient Reliance:** Measuring layer-wise normalized visual embedding gradient share $G_{\text{vis}} = \|\nabla_{\mathbf{h}_i}\mathcal{L}\|/(\|\nabla_{\mathbf{h}_i}\mathcal{L}\| + \|\nabla_{s_t}\mathcal{L}\|)$ yields $0.29$ for baseline versus $0.55$ for PAF.

### B.3 Comparison with Simpler Gating and Regularization Baselines

To evaluate whether the full spatio-temporal PAF module is necessary, we compare against simpler gating variants, feature dropouts, and gradient adjustment approaches under identical 250k-step budgets (Table 8).

Table 8: Comparison with simpler gating variants, regularization, and gradient adjustment baselines.

| Method / Control Variant | LIBERO Avg. (%) | Meta-World Avg. (%) | Robust. Comp. (%) |
|---|---|---|---|
| Matched Two-Stage No-PAF Control | 89.6 | 69.4 | 68.4 |
| Proprioceptive / Visual Dropout ($p = 0.2$) | 90.5 | 70.0 | 70.1 |
| GAP-style Gradient Adjustment (Lu et al., 2026) | 90.8 | 71.5 | 72.0 |
| Static Spatial Attention (w/o Temporal) | 89.8 | 69.1 | 71.2 |
| Global Temporal Gain (w/o Spatial) | 89.4 | 69.7 | 69.8 |
| Constant Phase Scalar ($\phi = 1.0$) | 90.0 | 69.5 | 71.8 |
| Shuffled Temporal Phase ($\phi_{\text{rand}}$) | 89.3 | 68.8 | 68.9 |
| Static Mean Spatial Gate | 89.6 | 70.1 | 70.5 |
| **Full PAF (Ours)** | **91.9** | **73.1** | **76.6** |

PAF outperforms constant phase scalars (+1.9% on LIBERO) and shuffled phases (+2.6%), confirming that unconditioned global gains or unsynchronized temporal scalars cannot replace synchronized spatio-temporal modulation.

### B.4 Impact of Training Data Diversity

To examine whether state-dominant shortcuts are merely artifacts of limited training data diversity, we evaluate policies across dataset subsampling ratios on LIBERO under fixed optimization steps and evaluation states (Table 9).

Table 9: Performance across training data subsampling ratios on LIBERO.

| Data Subsampling Ratio | Matched Control (%) | SmolVLA + PAF (%) | Net Margin (%) |
|---|---|---|---|
| 25% Demonstrations (Low Diversity) | 82.4 | 87.2 | +4.8% |
| 50% Demonstrations (Medium Diversity) | 86.5 | 89.7 | +3.2% |
| 100% Demonstrations (Full Dataset) | 89.6 | 91.9 | +2.3% |
| 100% Data + Strong Visual Aug. | 91.1 | 92.3 | +1.2% |

As training diversity increases, the margin between baseline and PAF narrows from +4.8% to +1.2%, confirming that dataset diversity helps alleviate shortcut learning. However, even under 100%

demonstrations with strong visual augmentations, a consistent positive margin remains, confirming that PAF provides complementary architectural inductive bias during phase-critical execution.

### B.5 Architectural Agnosticism on Robomimic Benchmark

To evaluate whether modality imbalance extends beyond large-scale VLA models to non-transformer architectures, we evaluate Behavior Cloning (BC) and Recurrent Behavior Cloning (BC-RNN) on the Robomimic benchmark (Mandlekar et al., 2021) (Table 10). Integrating PAF into these compact policies yields consistent improvements across all tasks (+6.6% for BC, +5.0% for BC-RNN), with the most pronounced gains observed on contact-rich tasks (*Square* and *Tool*).

Table 10: Success rates (%) on Robomimic across standard MLP (BC) and recurrent (BC-RNN) policies.

| Policy | Lift | Can | Square | Transport | Tool | Avg. |
|---|---|---|---|---|---|---|
| BC | 100.0 | 96.8 | 62.7 | 56.4 | 22.6 | 67.7 |
| **BC + PAF (Ours)** | **100.0** | **98.4** | **71.3** | **65.3** | **36.7** | **74.3** (+6.6) |
| BC-RNN | 100.0 | 98.6 | 82.2 | 72.9 | 69.7 | 84.7 |
| **BC-RNN + PAF (Ours)** | **100.0** | **100.0** | **91.3** | **79.8** | **77.5** | **89.7** (+5.0) |

## C Extended Evaluation: Integration with $\pi_0$

We evaluate PAF on the 3.5B parameter $\pi_0$ architecture (Black et al., 2024) across LIBERO and Meta-World benchmarks. As summarized in Table 11, component-wise ablations mirror trends observed on SmolVLA:

- *w/o 2-Stg. Protocol:* Single-stage joint tuning causes performance to drop below the frozen baseline (47.8% on Meta-World), indicating that unconstrained modulation destabilizes the pre-trained representational manifold.
- *w/o Temporal Branch:* Omitting phase estimation reduces Meta-World success by 4.4%, demonstrating the necessity of temporal phase tracking during dynamic motion.
- *w/o Spatial Branch:* Disabling spatial gating degrades LIBERO performance by 2.3%, as the policy cannot suppress visual distractors.
- *w/o SGCA:* Omitting early semantic cross-attention leads to attention bleeding, decreasing LIBERO by 1.4% and Meta-World by 2.2%.

Table 11: Extended ablation study of PAF components on the $\pi_0$ (3.5B) backbone.

| Variant | SGCA | Spat. Br. | Temp. Br. | 2-Stg. Trng. | LIBERO (Avg. %) | Meta-World (Avg. %) |
|---|---|---|---|---|---|---|
| Baseline (Frozen $\pi_0$) | – | – | – | – | 86.3 | 50.5 |
| w/o 2-Stg. Protocol | ✓ | ✓ | ✓ | – | 84.6 | 47.8 |
| w/o Temporal Branch | ✓ | ✓ | – | ✓ | 88.5 | 51.4 |
| w/o Spatial Branch | ✓ | – | ✓ | ✓ | 87.2 | 52.8 |
| w/o SGCA Module | – | ✓ | ✓ | ✓ | 88.1 | 53.6 |
| **Full PAF (Ours)** | ✓ | ✓ | ✓ | ✓ | **89.5** | **55.8** |

## D Qualitative Visualizations and Error Accumulation

### D.1 Comprehensive Spatial Attention Dynamics

To examine spatial grounding behavior, we visualize the dynamic attention gate $A$ across diverse manipulation scenarios (Figure 6):

- *Spatial Alignment (Top Row):* For tasks with narrow receptacle boundaries (*"put wine bottle on rack"*), activations converge on target contact interfaces during terminal placement phases.
- *Articulated Progression (Middle Row):* In multi-stage articulated tasks (*"put bowl in bottom drawer and close it"*), attention shifts sequentially across object parts (drawer handle $\rightarrow$ bowl $\rightarrow$ drawer panel).
- *Sequential Object Interaction (Bottom Row):* In relational tasks (*"pick bowl in drawer and place on plate"*), activations transition dynamically from the grasped object to the terminal placement target.

### D.2 Compounding Errors and Terminal Drift Breakdown

Figure 7 illustrates a characteristic failure rollout of the frozen SmolVLA baseline during physical block stacking. When the base block is spatially displaced from nominal demonstration locations, the baseline continues executing trajectory rollouts based on memorized joint coordinates. Because the policy under-utilizes visual closed-loop feedback, initial placement offsets compound across subsequent stages until the tower topples. PAF mitigates this failure mode by amplifying visual representations during terminal placement, allowing the policy to execute relative closed-loop adjustments with respect to displaced targets.

## E Further Discussion on Shortcut Learning and Emerging Modalities

**Mechanism of the State-Dominant Shortcut.** Proprioceptive states (joint positions, velocities) are compact, low-dimensional, and temporally continuous. During imitation learning, predicting actions $a_t$ from proprioception provides an immediate low-variance gradient signal. In contrast, visual observations are high-dimensional and noisy, requiring complex spatial-semantic parsing. Without structural inductive bias, optimizers favor the lower-effort proprioceptive pathway, attenuating visual representation learning. PAF addresses this gradient imbalance by dynamically amplifying visual feature activations during contact phases where proprioceptive priors alone are physically insufficient.

**Integration with Force-Torque and Tactile Modalities.** Emerging tactile and force sensors fit naturally into PAF's functional decomposition:

- *Temporal Branch (Force-Torque Signals):* A 6-axis force-torque vector $\mathbf{F}_t \in \mathbb{R}^6$ can concatenate with $\mathbf{X}_{temp}$, as sharp force derivatives signal physical contact onsets and phase transitions.
- *Spatial Branch (Tactile Arrays):* High-resolution tactile arrays (e.g., GelSight taxel grids) can be processed through the spatial branch, providing spatially localized contact geometry to complement visual inputs.

## F Inference Latency and Computational Overhead Analysis

Robotic closed-loop control imposes strict latency constraints. On an NVIDIA RTX 4090 GPU, a forward pass of the frozen SmolVLA backbone requires $\sim 45$ ms. The complete PAF module introduces $\sim 3.5$ ms of additional latency per step. For a typical 10 Hz control loop (100 ms execution budget per action chunk), the policy operates comfortably within real-time margins, confirming that phase-adaptive modulation preserves reactive control agility.

## G Extended Limitations and Broader Impacts

**Structural and Empirical Limitations.**

1. *Kinematics-Only Temporal Phase Boundary:* As formulated in Eq. 6, the temporal estimator receives only kinematic state history $\mathbf{X}_{temp}$. Consequently, when the robot remains stationary while external scene objects move ($\Delta s_t \approx \mathbf{0}$, such as external object displacements or human interventions), $\phi$ cannot independently register external changes. Although the

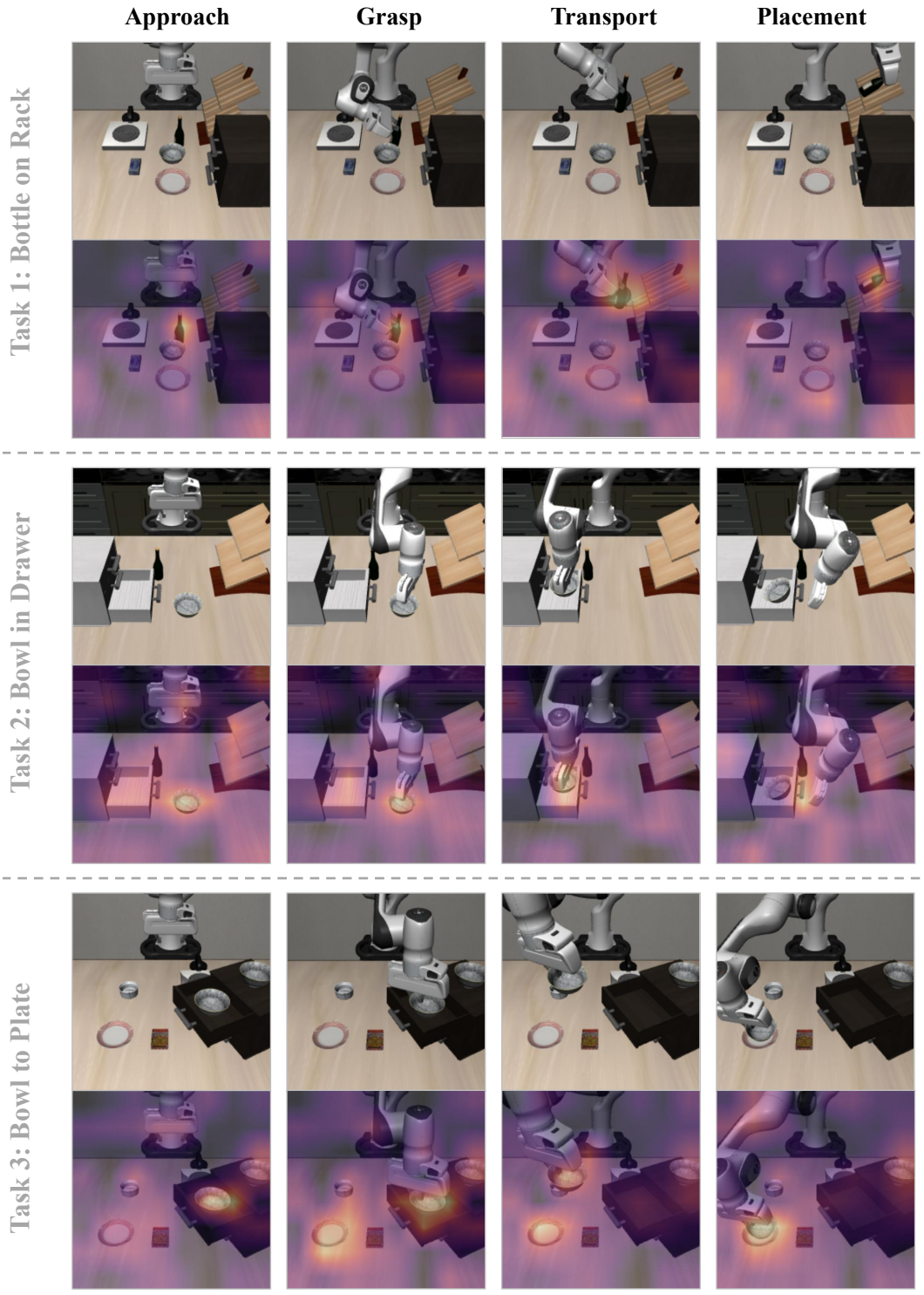



Figure 6: **Dynamic Spatial Attention Tracking across Manipulation Tasks.** From top to bottom: raw RGB inputs and overlaid spatial attention heatmaps across continuous execution steps, illustrating localized spatial gating during contact-critical manipulation phases.

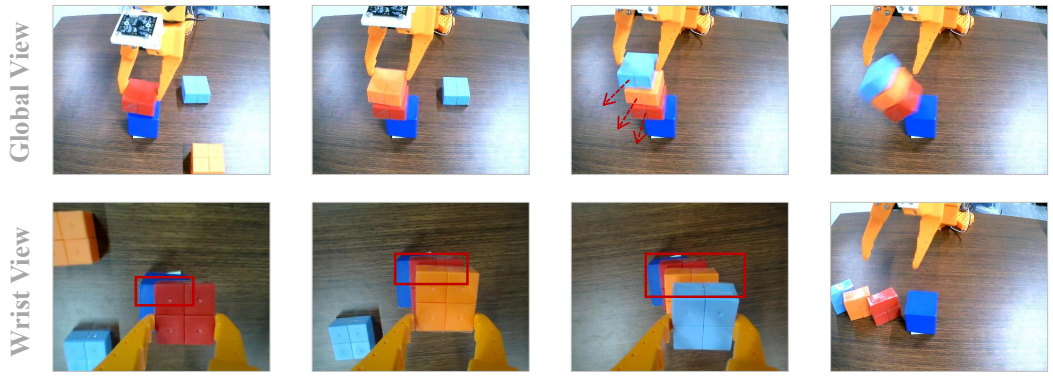



Figure 7: **Compounding Error Trajectory in Physical Stacking.** Execution sequence of the frozen baseline (SmolVLA). Spatial displacement of the base block causes compounding offsets across successive stages. PAF mitigates this by providing phase-synchronized visual modulation at each placement juncture.

visual spatial branch (LDSA) continuously recomputes spatial masks to redirect attention, detecting exogenous scene events without robot motion remains a structural boundary of the kinematics-based design.

2. *Vision Encoder Manifold Bottleneck:* As a residual modulation plug-in, PAF's capacity is bounded by the representation manifold of the pre-trained vision encoder; it cannot synthetically reconstruct missing geometry under severe, prolonged visual occlusions.
3. *Physical Evaluation Scope:* Physical experiments are conducted on a single 6-DoF robotic arm. Broader evaluations across multi-arm setups, dexterous multi-fingered hands, and dynamic in-the-wild manipulation tasks remain important avenues for future study.

**Broader Impacts.** This work advances the execution reliability and closed-loop adaptability of Vision-Language-Action models in physical manipulation. As general-purpose robotic policies improve in autonomy, continued scrutiny regarding safety-critical operational boundaries, physical fail-safes, and the societal impacts of automated physical labor is essential.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims regarding PAF mitigating modality imbalance and compounding errors are supported by simulation benchmarks across diverse backbones and real-world physical evaluations detailed in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly analyze structural limitations (such as the kinematics-only temporal branch boundary and frozen vision encoder representational limits) in Section 5 and provide an extended discussion in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper is empirical and proposes an architectural adaptation plug-in (PAF) for robotic continuous control; it does not present formal mathematical theorems or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Complete architectural specifics, multi-camera reshaping protocols, hyperparameters, and the staged training schedule are thoroughly detailed in Section 3 and Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Training scripts, baseline evaluation harnesses under LeRobot, and plug-in implementations are provided in the supplementary material and will be publicly released.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All optimization hyperparameters (AdamW settings, learning rates, LoRA configurations, batch sizes) are specified in Section 3.4, Appendix A.2, and Table 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 4.1 and Table 1 report 95% confidence intervals computed via task-stratified paired bootstrap (10,000 resamples) across 5 random seeds (1,000 rollouts per task). Real-world evaluations report Wilson score 95% binomial confidence intervals.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.2 and Appendix F state that models were trained on NVIDIA RTX 6000 pro (96GB) GPU and benchmarked for latency on an RTX 4090 GPU.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research focuses on improving the precision and robustness of robotic manipulation and complies fully with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: A dedicated Broader Impacts discussion is provided in Appendix G, addressing robotic safety boundaries, physical fail-safes, and the socioeconomic considerations of automated labor.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The models presented are adaptation plug-ins for low-level robotic manipulation control (e.g., block stacking) and do not pose dual-use or high misuse risks compared to generative foundation models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credit and cite open-source architectures (SmolVLA, $\pi_0$) and benchmarks (LIBERO, Meta-World, Robomimic), complying with their respective research licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The SOARM101 teleoperation dataset, hardware specs, and demonstration collection protocols are thoroughly documented in Appendix A.5 and will be released with open documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research relies on author-collected teleoperated demonstrations and standardized simulation benchmarks; it does not involve crowdsourcing or external human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The data collection involves robotic teleoperation by the authors and does not constitute human subjects research requiring IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Pre-trained foundation models are utilized strictly as the host architecture for downstream adaptation; LLMs are not used as an original or non-standard algorithmic component in the proposed method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.