PRIMINGVLA: SELF-SUPERVISED ACTION PRIMING FOR PARAMETER-EFFICIENT VISION–LANGUAGE–ACTION ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting Vision–Language–Action (VLA) models requires learning task-specific control through an action expert while preserving the multimodal capabilities of a pretrained vision–language model (VLM). Existing methods primarily align multimodal representations with action generation, but overlook the need to establish structured action priors before task-conditioned learning. To address this issue, we introduce PrimingVLA, a parameter-efficient framework built on our proposed new principle of action priming before multimodal conditioning. Its Self-Supervised Action Priming (SAP) learns temporal action structure through masked trajectory reconstruction, after which multimodal inputs are introduced for joint adaptation. A cross-prompt calibration coordinates their adaptation by bounding relative action-prompt updates using VLM-prompt updates as a reference. Experiments on LIBERO, LIBERO-PLUS, and LIBERO-PRO demonstrate effective task adaptation and characterize robustness under distribution shifts, with ablations supporting the benefits of staged learning and coordinated updates. Real-world experiments on a Piper robotic arm further demonstrate its effectiveness in robotic manipulation. Neural Tangent Kernel monitoring across training stages suggests that the sensitivity gap between the two pathways progressively narrows during adaptation. Remarkably, PrimingVLA requires updating fewer than 0.05M parameters ($<0.001\%$ of the full model) while keeping both pretrained backbones frozen, enabling efficient adaptation and rapid task switching.

1 INTRODUCTION

Vision–Language–Action (VLA) models translate visual observations and language instructions into executable actions for robotic manipulation. The common architecture (Black et al., 2025) couples a pretrained Vision–Language Models (VLMs) for multimodal understanding with an action expert for continuous action generation. The key challenge of this adaptation is to meet the demands of action learning while preserving the pretrained multimodal capabilities of VLM.

Most existing VLA methods adapt pretrained VLMs to action generation by fine-tuning the VLM or learning alignment modules that transform multimodal representations into conditioning features for an action expert to generate continuous actions (Wang et al., 2026b; Dong et al., 2025; Black et al., 2025). However, this adaptation introduces a fundamental stability–plasticity dilemma: *the VLM must acquire task-specific control information while simultaneously preserving the cross-modal representations that underpin its pretrained capabilities.* Recent studies (Kumar et al., 2022) show that adapting VLMs for action generation can distort these representations and potentially compromise generalization in robotic manipulation. Preservation-based methods therefore constrain VLM updates to mitigate such degradation (Driess et al., 2026; Dey et al., 2025; Huang et al., 2026). Yet this line of work primarily asks how to prevent the VLM from changing too much, while largely overlooking a more fundamental question: *how should the action pathway itself be prepared to acquire task-specific control?* As a result, action learning is typically introduced only after multimodal conditioning, leaving the action expert to discover its temporal and structural priors concurrently with task adaptation. This can make adaptation unnecessarily difficult, especially when supervision is limited or the task distribution shifts.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

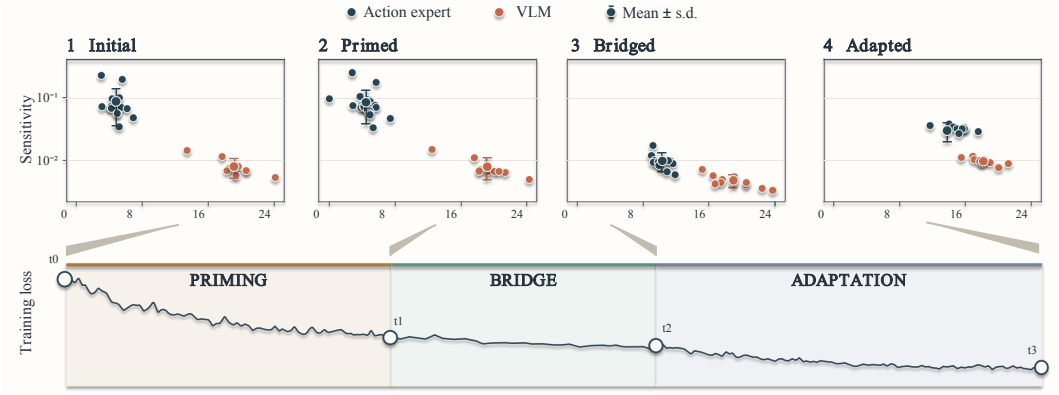


Figure 1: Illustrative task-conditioned adaptation geometry motivating the NTK diagnostics.

To address this imbalance between VLM preservation and action preparation, we advocate **learning action structure before introducing multimodal conditioning**. This new paradigm is motivated by the finding that training the prediction head before joint fine-tuning can reduce distortion of pretrained features (Kumar et al., 2022). To further examine how the two pathways respond to action learning, we conduct a Neural Tangent Kernel (NTK) spectral analysis of $\pi_{0.5}$ model. Figure 1 reveals that action predictions are largely more sensitive to parameter changes in action expert than in the VLM, proving the interpretability of explicit coordination of updates during adaptation.

Building on this perspective, we introduce PrimingVLA, a parameter-efficient framework centered on Self-Supervised Action Priming (SAP). We utilize dual prompts (VLM prompts and action prompts) to adapt the existing multimodal knowledge with downstream robotic tasks: VLM prompts learn task-relevant conditioning, while action prompts guide trajectory generation. The dual prompts are optimized in a three-stage training curriculum. First, action priming trains the action prompts to reconstruct masked action segments without multimodal inputs. Second, a conditioning bridge integrates visual, language, and robot state inputs and jointly trains dual prompts. Third, the main adaptation further refines both prompt banks by observation-conditioned flow matching. During the latter two stages, Cross-Prompt Calibration (CPC) regulates action-prompt updates using VLM-prompt updates as an adaptive reference. We train only dual prompts and keep all other model parameters frozen and therefore require fewer than 0.05M trainable parameters, making task-specific adaptation realizable under limited hardware resources. Our main contributions are threefold:

1. We propose action priming as a new principle for VLA adaptation, learning task-relevant action structure before multimodal conditioning through Self-Supervised Action Priming (SAP).
2. We develop PrimingVLA, a dual-prompt framework that bridges action priming and multimodal adaptation, with Cross-Prompt Calibration (CPC) coordinating their joint updates.
3. We achieve highly efficient adaptation with fewer than 0.05M trainable parameters ($<0.001\%$ of the full model) while keeping both pretrained backbones frozen, with experiments across LIBERO, LIBERO-PRO, LIBERO-PlPLUSus, and real-world manipulation validating its effectiveness.

2 Related Work

Recent VLA adaptation methods can be broadly grouped by their primary focus: improving action generation or preserving pretrained knowledge.

Action-oriented adaptation. Action-oriented approaches improve robot control by adapting multimodal representations to action generation. For example, OpenVLA-OFT (Kim et al., 2025a) improves continuous control through parallel decoding and action chunking. VLA-Adapter (Wang et al., 2026b) learns bridge attention between VLM features and an action policy, while VITA-VLA (Dong et al., 2025) transfers knowledge from a pretrained action model through alignment and selective fine-tuning. CofactVLA (Zhang et al., 2026) utilizes a language-masked counterfactual branch to mitigate visual shortcuts through feature-level deconfounding and action-level projection guidance. QuoVLA (Wang et al., 2026a) further compresses VLM representations through quotient-space

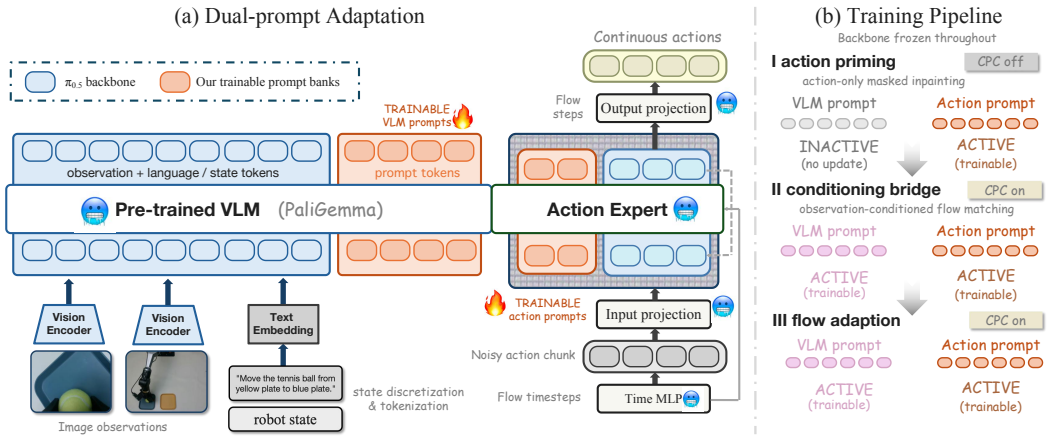


Figure 2: Architecture overview of our PrimingVLA.

quantization, removing action-irrelevant variations while retaining information needed for control. Recent Parameter-Efficient Fine-Tuning (PEFT) methods improve adaptation capacity through spectral expert decomposition in VLA-GSE (Jiang et al., 2026) and control-regime-conditioned low-rank updates in PhaseLoRA (Guo et al., 2026b). However, a compact adaptation module cannot ensure that pretrained representations remain useful under distribution shifts (Huang et al., 2026; Dalal et al., 2026). Therefore, our focus is the order of adaptation: Learn task-specific action structure first through masked action reconstruction, and then introduce multimodal conditioning. This self-supervised stage reuses the pretrained action expert and derives supervision directly from action sequences, requiring no separate action teacher or latent action model.

Preservation-oriented adaptation. These methods seek to retain pretrained knowledge during action adaptation. ReVLA (Dey et al., 2025) restores pretrained visual weights, while MAPS (Huang et al., 2026) constrains module-wise weight drift. Knowledge Insulation (Driess et al., 2026) blocks continuous-action gradients backpropagation to VLM. PriorVLA (Guo et al., 2026a) further reuses pretrained scene and motor priors to guide a separate adaptation expert. In these methods, the action pathway learns task-specific trajectories from conditioning representations that are still changing, which can complicate early optimization. PrimingVLA first prepares action prompts through SAP, learning action structure from masked action sequences. Building on this preparation, a conditioning bridge introduces multimodal inputs for joint prompt learning, with CPC coordinating updates while both backbones remain frozen.

3 Preliminaries

VLA Policies and Downstream Adaptation. Given demonstrations $\mathcal{D} = \{(o_i, \ell_i, a_i)\}_{i=1}^{N}$, a VLA policy learns $\pi_\Theta(a \mid o, \ell)$ from observations o (includes images and available robot states), language instructions ℓ, and action chunks $a \in \mathbb{R}^{H\times d}$. H and d denote the action horizon and dimension. Supervision uses token prediction or continuous generation (Zitkovich et al., 2023; Kim et al., 2025b; Black et al., 2024). We consider a robot-pretrained policy $\Theta = (\theta_v, \theta_a)$, where the VLM θ_v provides multimodal context and action pathway θ_a generates continuous actions.

Flow Matching for Continuous Actions. Flow matching learns to transform Gaussian noise into actions (Lipman et al., 2023). For $\epsilon \sim \mathcal{N}(0, I)$ and $t \in [0, 1]$, the input and target velocity are

$$x_t = t\epsilon + (1-t)a, \qquad u = \epsilon - a. \tag{1}$$

With b_{ih} indicating valid steps in minibatch $\mathcal{B}$, the predictor minimizes

$$\widehat{\mathcal{L}}_{\text{flow}}(\Theta; \mathcal{B}) = \frac{\sum_{i\in\mathcal{B}} \sum_{h=1}^{H} b_{ih} \left\| v_\Theta(x_{i,t_i}, t_i \mid o_i, \ell_i)_h - u_{ih} \right\|_2^2}{d \sum_{i\in\mathcal{B}} \sum_{h=1}^{H} b_{ih}}, \qquad \mathcal{L}_{\text{flow}} = \mathbb{E}\left[\widehat{\mathcal{L}}_{\text{flow}}\right]. \tag{2}$$

The expectation covers data, noise, and timestep sampling. Integrating the learned velocity from $t = 1$ to $t = 0$ produces an action chunk.

4 METHOD AND TRAINING RECIPE

PrimingVLA learns action structure before joint multimodal adaptation. Two prompt banks provide the trainable interface; SAP prepares the action prompts, a conditioning bridge introduces observations, and CPC coordinates subsequent updates (Figure 2).

4.1 DUAL-PROMPT POLICY PARAMETERIZATION

Dual prompts adapt the policy through a compact parameter space while preserving the pretrained weights. Let $P = (P_v, P_a)$, where $P_v \in \mathbb{R}^{n_v \times d_v}$ and $P_a \in \mathbb{R}^{n_a \times d_a}$ are prompt banks for the VLM and action expert, respectively. The resulting velocity predictor is $v_{\Theta,P}(x_t, t \mid o, \ell)$, with Θ fixed throughout training. Let O and X_t denote multimodal and noisy-action token blocks. The VLM processes $[O; P_v]$, and the expert processes $[P_a; X_t]$. For query and key blocks ordered as (O, P_v, P_a, X_t), attention follows

$$\mathcal{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \qquad N_P = n_v d_v + n_a d_a = 49{,}152, \tag{3}$$

where each entry specifies block visibility and padding is excluded. In our implementation, $n_v = n_a = 16$, $d_v = 2048$, and $d_a = 1024$, giving $49{,}152$ trainable prompt parameters. The VLM prompt bank is appended to the VLM token sequence and the action prompt bank is prepended to the action-token sequence at the first transformer block; both prompt states are then propagated through the 18-layer backbones. Each bank is initialized independently from $\mathcal{N}(0, 0.02^2)$. All backbone weights and action projections remain frozen. VLM prompts aggregate observation context. Action prompts interact bidirectionally with action tokens, which relay multimodal context across layers. The mask keeps observation states independent of the prompts, ensuring $\partial H_O^{(r)} / \partial P = 0$. Gradients still flow through the frozen backbones to update the prompts.

4.2 SELF-SUPERVISED ACTION PRIMING

Self-Supervised Action Priming prepares action prompts to exploit temporal structure before multimodal conditioning. In stage I, we remove O and P_v and optimize only P_a. For sample i, a binary mask $M_{ih} \le b_{ih}$ selects valid action positions for reconstruction. Using the flow path in Eq. (1), the expert receives

$$\widetilde{x}_{i,t_i,h} = M_{ih} x_{i,t_i,h} + (1 - M_{ih}) a_{ih}. \tag{4}$$

Masked actions follow noisy interpolation, while unmasked actions provide clean bidirectional context. For the expert-only predictor $v^{\mathrm{SAP}}_{\theta_a, P_a}$, the minibatch objective is

$$\widehat{\mathcal{L}}_{\mathrm{SAP}} = \frac{\displaystyle\sum_{i \in \mathcal{B}} \sum_{h=1}^{H} M_{ih} \left\| v^{\mathrm{SAP}}_{\theta_a, P_a}(\widetilde{x}_{i,t_i}, t_i)_h - u_{ih} \right\|_2^2}{\max\left(1, d \displaystyle\sum_{i \in \mathcal{B}} \sum_{h=1}^{H} M_{ih}\right)}. \tag{5}$$

We minimize $\mathcal{L}_{\mathrm{SAP}} = \mathbb{E}[\widehat{\mathcal{L}}_{\mathrm{SAP}}]$ over data, noise, timesteps, and masks. For $H = 50$, each mask selects $\min(40, \sum_h b_{ih})$ positions uniformly without replacement. This normalization weights valid masked elements equally across the minibatch and excludes padding. Supervision comes directly from demonstration actions, with no auxiliary labels or prediction head. Priming retains the action projections, time conditioning, and expert used in subsequent flow adaptation.

4.3 MULTIMODAL ADAPTATION AND INFERENCE

The conditioning bridge converts the primed action prompts into an observation-conditioned policy. Phase II restores O and P_v and optimizes both prompt banks using Eq. (2), with $v_{\Theta,P}$ as the predictor:

$$\min_{P_v, P_a} \mathcal{L}_{\mathrm{flow}}(\Theta, P_v, P_a), \qquad (P_v, P_a)_{\mathrm{init}} = (P_v^{(0)}, P_a^{\mathrm{I}}), \tag{6}$$

where $P_v^{(0)}$ is the initial VLM prompt bank and P_a^{I} is the priming output. The bridge continues the optimizer and scheduler from Phase I, retaining the action-prompt moments while introducing multimodal gradients. Phase III starts from the bridge checkpoint and resets the optimizer and scheduler. Both phases use the same objective, attention mask, and trainable parameters, with CPC active throughout. At inference, the observation and VLM-prompt states are independent of the noisy action block, so their keys and values are cached. For the final prompts $P^\star$, generation starts from $x_1 \sim \mathcal{N}(0, I)$ and follows

$$x_{t_{k+1}} = x_{t_k} - \frac{1}{K} v_{\Theta, P^\star}(x_{t_k}, t_k \mid o, \ell), \qquad t_k = 1 - \frac{k}{K}, \tag{7}$$

with $K = 10$ and $k = 0, \ldots, K-1$. The expert recomputes action and action-prompt states at each step, and x_0 gives the predicted action chunk.

4.4 Cross-Prompt Calibration

Cross-Prompt Calibration constrains action-prompt adaptation relative to the VLM prompts after optimizer preconditioning. For bank $j \in \{v, a\}$, let $\widehat{m}_j$ and $\widehat{v}_j$ be the bias-corrected AdamW moments. η_j is scheduled learning rate. The candidate learning update and its relative magnitude are

$$D_j = -\eta_j \frac{\widehat{m}_j}{\sqrt{\widehat{v}_j} + \varepsilon_{\mathrm{opt}}}, \qquad q_j = \frac{\|D_j\|_F}{\|P_j\|_F}. \tag{8}$$

These candidates exclude weight decay and are evaluated without advancing optimizer state. In exact arithmetic, CPC projects D_a onto a Frobenius-norm ball with radius $R_a = q_v \|P_a\|_F / \rho$

$$\begin{aligned} D_a^\star &= \operatorname*{arg\,min}_{\|D\|_F \le R_a} \frac{1}{2}\|D - D_a\|_F^2 = s_a D_a, \\ s_a &= \begin{cases} \min\left(1, \dfrac{q_v}{\rho q_a}\right), & q_a > 0, \\ 1, & q_a = 0, \end{cases} \qquad s_v = 1, \quad \rho \ge 1. \end{aligned} \tag{9}$$

The calibrated learning step preserves its original direction and satisfies $\|D_a^\star\|_F / \|P_a\|_F \le q_v / \rho$, with $\rho = 2$ fixed in all experiments. For each parameter group j, we temporarily scale the learning rate to $s_j \eta_j$. In exact arithmetic, this yields $P_j^+ = (1 - s_j \eta_j \lambda_j) P_j + s_j D_j$, where λ_j is the weight-decay coefficient. We verify the candidate update in the native parameter dtype and further reduce the action scale if rounding violates the bound. The scheduled learning rates are restored after the optimizer step. The bound applies only to the learning component and excludes the separately applied weight-decay term; when q_v is small, CPC can consequently clip the action-prompt update more strongly.

5 Simulation Experiments

5.1 Comparison with State-of-the-arts

We evaluate PrimingVLA on standard LIBERO (Liu et al., 2023) and transfer the same models to LIBERO-PLUS (Fei et al., 2025) and LIBERO-PRO (Zhou et al., 2025) without further adaptation. Simulation uses the image, language, and discretized state-token inputs expected by the pretrained policy; the qualitative real-robot action comparisons use RGB observations and language only. Ten rollouts are conducted in standard LIBERO per task, with overall success averaged across Spatial, Object, Goal, and Long (10). Published baselines retain their original protocols.

(1) LIBERO Evaluation. PrimingVLA achieves 89.00% success rate on LIBERO dataset, succeeding in 356 of 400 episodes. As shown in Table 1, success rates on Spatial, Object, Goal, and Long are 95.00%, 96.00%, 82.00%, and 83.00%. Performance is strongest on Object and Spatial, while Goal and Long remain more challenging. Our comparison includes general PEFT-VLA methods and six $\pi_{0.5}$-based configurations, which reports three-seed means after 30k training steps. PhaseLoRA (Guo et al., 2026b) achieves 68.9% with an estimated 167M trainable parameters, whereas our method achieves 89.00% using only 0.05M prompt parameters at the 3k flow checkpoint. The results demonstrate effective adaptation with a compact trainable interface.

Table 1: LIBERO performance and computational budgets.

Method	Trainable (M)	Success rate (%)				
		Spatial	Object	Goal	Long	Overall
OpenVLA (LoRA) (Kim et al., 2025b)	$\approx 111^{*}$	84.7	88.4	79.2	53.7	76.5
UniVLA (Bu et al., 2025)	≈ 123	96.5	96.8	95.6	92.0	95.2
VLA-Adapter (Wang et al., 2026b)	197.2	97.8	99.2	97.2	95.0	97.3
$\pi_{0.5}$ PEFT results reported by PhaseLoRA (Guo et al., 2026b)						
LoRA	$\approx 100^{\dagger}$	60.7	41.3	32.7	18.7	38.3
High-rank LoRA	$\approx 167^{\dagger}$	71.0	73.3	55.3	27.0	56.7
DoRA	$\approx 167^{\dagger}$	72.7	74.0	57.3	29.7	58.4
LoRA-MoE	$\approx 169^{\dagger}$	73.3	70.7	58.0	31.0	58.3
LoRA-SP	$\approx 167^{\dagger}$	76.3	72.0	56.0	32.3	59.2
PhaseLoRA	$\approx 167^{\dagger}$	85.3	88.0	64.0	38.3	68.9
PrimingVLA (Ours)	0.05	95.00	96.00	82.00	83.00	89.00

$*$: rank-32 LoRA reference (Kim et al., 2025a, Table IV). †: estimated via $pN_0/(1-p)$, using reported fractions p and the released-code base size $N_0 \approx 3.617$B.

Table 2: LIBERO-PLUS performance and computational budgets.

	Trainable		Success rate (%)							
Method	M	%	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
Qwen3-VL-4B + MLP action head: PEFT adaptation										
LoRA†	≈ 114	2.58	44.3	53.6	82.1	92.9	91.1	67.1	67.2	69.2
DoRA†	≈ 114	2.54	49.7	61.0	86.6	95.7	94.4	73.4	78.6	75.3
PiSSA†	≈ 114	2.54	48.3	59.4	86.0	95.4	94.0	72.8	78.1	74.5
GOAT†	≈ 114	2.54	54.5	61.8	87.7	96.2	95.2	74.6	79.3	76.8
VLA-GSE†	114.04	2.51	64.4	68.5	88.8	97.3	97.3	79.4	82.6	81.2
Qwen2.5-0.5B: suite-specific fine-tuning / model merging										
VLA-Adapter‡	197.2^{*}	–	36.4	37.4	73.8	71.0	76.6	57.2	70.2	59.0
MergeVLA (suite)‡	–	–	62.4	46.4	75.7	92.7	92.7	74.7	73.7	72.4
MergeVLA (TIES)‡	–	–	50.7	30.3	66.0	84.2	85.7	66.0	68.1	62.5
PrimingVLA (Ours)	0.05	≈ 0.001	40.46	56.19	68.51	75.13	70.54	62.27	64.98	61.58

†: budgets include the action head; $\approx$ 114M is a group estimate. $*$: original budget, unverified for this checkpoint (Wang et al., 2026b, App. F.2).

(2) LIBERO-PLUS Evaluation. LIBERO-PLUS evaluates robustness to seven perturbation categories without additional adaptation on the benchmark. With 0.05M trainable parameters, our transferred model succeeds in 6,176 of 10,030 variants, yielding an episode-weighted success rate of 61.58%. Table 2 compares PEFT configurations from VLA-GSE (†) (Jiang et al., 2026) and adaptation and merging methods from MergeVLA (‡) (Fu et al., 2025); their protocols and backbones are not identical to ours. Among our seven perturbation categories, lighting is the least challenging (75.13%), followed by background (70.54%) and language (68.51%), whereas camera changes are the most challenging (40.46%). Across base suites, success reaches 81.77% on Spatial and 68.78% on Object, compared with 50.83% on Goal and 46.17% on Long. These results indicate that the compact prompt interface transfers useful control beyond the training distribution, while camera variations and long-horizon tasks remain important challenges.

(3) LIBERO-PRO Evaluation. LIBERO-PRO evaluates five perturbation categories using 200 variants per checkpoint and one rollout per variant. Table 3 separates four-suite comparisons with OpenVLA (†) (Zhou et al., 2025), UniVLA (Bu et al., 2025), and GLaD (‡) (Guo et al., 2025) from Spatial-only comparisons with OpenVLA-OFT and VLA-Adapter (§) (Dalal et al., 2026). Overall (3) averages Object, Position, and Semantic, excluding Task and Environment because baseline coverage or configuration alignment is incomplete. Extending adaptation from 3k to 6k raises this shared score from 53.33% to 62.50%, while success across all five categories increases from 41.00% to 48.00%. At 6k steps, Position success reaches 37.50%, surpassing the reported four-suite reference scores. These

Table 3: LIBERO-PRO performance and computational budgets.

Method	Trainable (M)	Object	Position	Semantic	Task	Env∗	Overall (3)
(a) Four-suite evaluation							
OpenVLA (LoRA)[†]	$\approx 111^{*}$	95.78	0.00	97.50	0.00	–	64.43
UniVLA (LoRA)[‡]	$\approx 123^{*}$	72.25	2.25	95.50	–	–	56.67
GLaD (LoRA)[‡]	–	79.75	5.25	96.25	–	–	60.42
PrimingVLA (3k)	0.05	65.00	25.00	70.00	7.50	37.50	53.33
PrimingVLA (6k)	0.05	67.50	37.50	82.50	0.00	52.50	62.50
(b) Spatial-only evaluation: additional earlier-model references							
OpenVLA-OFT[§]	279^{*}	95.20	0.00	74.40	–	–	56.53
VLA-Adapter (frozen VLM)[§]	97.3^{*}	73.40	0.00	56.00	–	–	43.13
VLA-Adapter (LoRA)[§]	197.2^{*}	89.60	2.30	91.10	–	–	61.00
PrimingVLA (3k, Spatial)	0.05	90.00	40.00	90.00	10.00	50.00	73.33
PrimingVLA (6k, Spatial)	0.05	80.00	50.00	90.00	0.00	60.00	73.33

∗: Env. baseline configuration alignment is unverified.

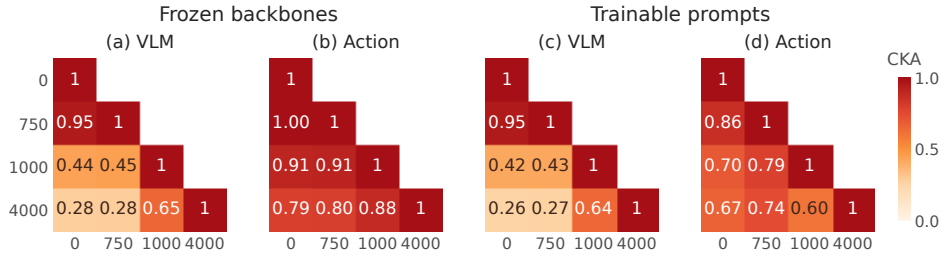


Figure 3: Cross-stage NTK similarity for frozen backbones (left) and trainable prompts (right). Entries are median CKA over 10 paired probe/flow seeds on the same 32 samples; darker cells indicate greater similarity. Axes show cumulative updates: initialization (0), SAP (750), bridge (1,000), and adaptation (4,000).

gains highlight the benefits of continued prompt adaptation for generalization under perturbations without increasing the trainable parameter budget. Appendix A provides detailed task-wise results.

5.2 Ablation Study

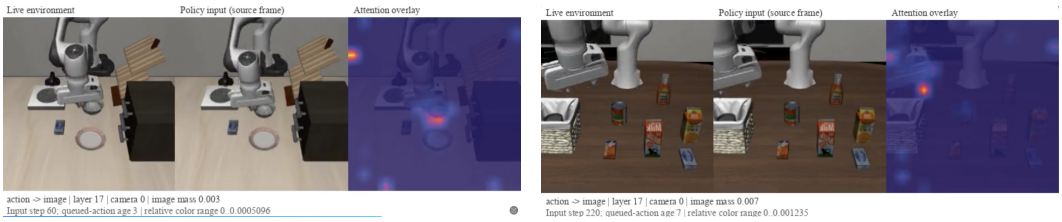
Table 4 highlights the complementary roles of dual prompts, staged training, and CPC. All variants freeze the backbones and action projections and use 3,000 flow updates. VLM prompts alone achieve 0% success, while action prompts alone reach 60.25%; combining both raises success to 78.75%. With a fixed budget of 1,000 preparation updates, 750 SAP followed by 250 bridge updates achieves 89.00%, compared with 84.75% for 1,000 SAP updates alone. This demonstrates the benefit of introducing multimodal conditioning after action priming rather than simply extending priming. Removing CPC under the same curriculum reduces success to 77.75%, supporting the value of coordinated prompt updates. Comparisons with variants without preparation involve 1,000 additional updates and therefore do not isolate the effect of SAP alone.

5.3 Cross-Stage Tangent Dynamics

We examine how each training stage reshapes NTK geometry using centered kernel alignment (CKA; Figure 3). All checkpoints, including SAP, are evaluated with the same fully observation-conditioned velocity predictor, providing a common functional reference despite their different training objectives. During SAP, the action-prompt kernel changes moderately (CKA 0.86), while the action-backbone kernel remains nearly aligned (0.998). This suggests that priming adjusts the prompt-side sensitivity structure while largely retaining the backbone's tangent geometry. The bridge produces the largest adjacent-checkpoint change on the VLM side, with CKA values of 0.45 for the backbone and 0.43

Table 4: Component ablations on LIBERO.

Variant	Components					Success rate (%)				
	P_v	P_a	SAP	Bridge	CPC	Spatial	Object	Goal	Long	Overall
VLM prompt only	✓	×	×	×	×	0.00	0.00	0.00	0.00	0.00
Action prompt only	×	✓	×	×	×	90.00	73.00	63.00	15.00	60.25
Dual prompts	✓	✓	×	×	×	91.00	89.00	69.00	66.00	78.75
Direct dual	✓	✓	×	×	✓	97.00	89.00	73.00	64.00	80.75
w/o Bridge	✓	✓	✓	×	✓	93.00	95.00	76.00	75.00	84.75
w/o CPC	✓	✓	✓	✓	×	87.00	86.00	74.00	64.00	77.75
PrimingVLA (ours)	✓	✓	✓	✓	✓	95.00	96.00	82.00	83.00	89.00



(a) LIBERO-Goal (b) LIBERO-Long

Figure 4: Attention visualizations on LIBERO-Goal and LIBERO-Long. The model attends to manipulated objects and nearby interaction regions, indicating a focus on task-relevant visual cues.

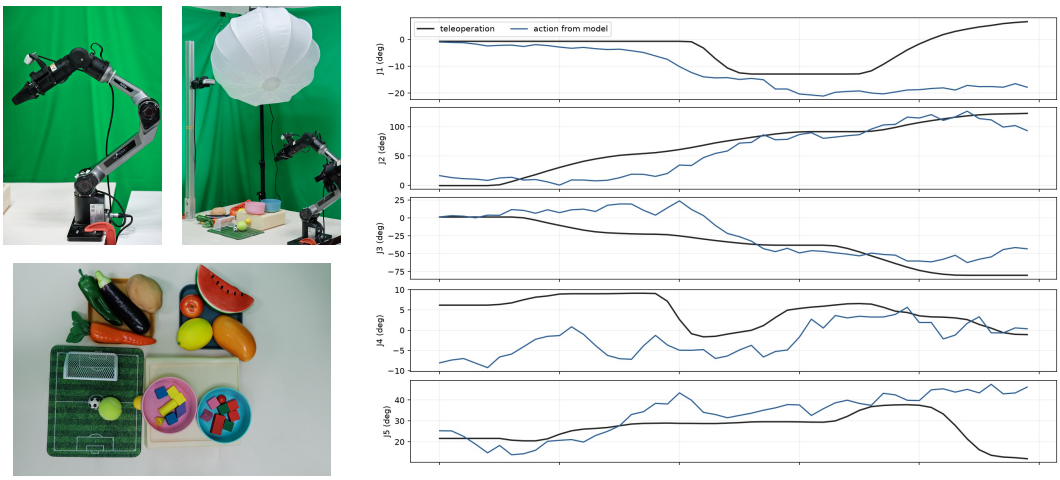
for its prompts, compared with 0.91 for the action backbone. This contrast is consistent with the bridge reorganizing the conditioning pathway around an already primed action pathway, giving this stage a distinct role beyond additional action-only training. During subsequent adaptation, action prompts continue to reorganize (0.60), while the action backbone retains higher alignment (0.88). Prompt refinement therefore continues after the conditioning transition, supporting further adaptation within the frozen expert. Together, these stage-specific patterns support the intended progression from action preparation to multimodal integration and joint refinement. Both backbones remain frozen throughout; changes in their kernels arise from prompt-dependent output sensitivity rather than backbone weight updates. CKA characterizes normalized tangent geometry, so these observations complement performance ablations without directly establishing changes in sensitivity magnitude or preservation of semantic knowledge.

Qualitative Results. Figure 4 visualizes PrimingVLA's attention in LIBERO-Goal and LIBERO-Long. For each example, we show the live scene, the corresponding policy input, and the attention overlay. The highlighted regions overlap the manipulated objects and nearby interaction areas, suggesting that the model focuses on visual cues relevant to the current action. Additional qualitative results can be found in Appendix B.

6 Real-Robot Experiments

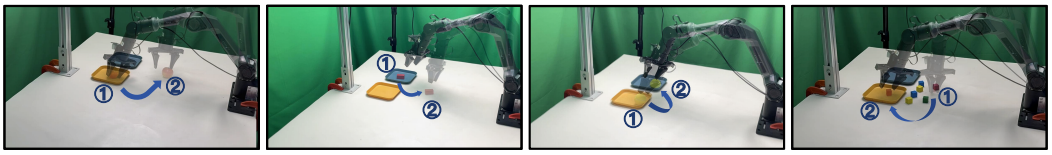
(1) Platform and Tasks. The real-robot platform comprises an AgileX Piper six-degree-of-freedom (6-DoF) robotic arm, a two-finger gripper, a wrist RGB camera, and a fixed external RGB camera (Figure 5). The four tasks in Figure 6 cover object placement, removal, and plate-to-plate transfer, requiring coordinated grasping, transport, and controlled release. Each task provides 100 demonstrations for prompt tuning, collected with the same camera setup and robot control interface used during policy evaluation. Blue arrows illustrate the pick-and-place paths for the following tasks:

Task 1: Apple placement: place the apple on the yellow plate.
Task 2: Cuboid removal: remove the cuboid from the blue plate.
Task 3: Tennis-ball transfer: move the tennis ball from the yellow plate to the blue plate.
Task 4: Cube placement: place the red cube on the yellow plate.



(a) Piper platform and objects (b) Predicted and demonstrated (groundtruth) joint actions

Figure 5: Piper platform and predicted joint actions (blue) versus teleoperation ground truth (black).



(a) Task 1 (b) Task 2 (c) Task 3 (d) Task 4

Figure 6: Four real-world pick-and-place tasks (T1–T4).

(3) Evaluation Protocol and Results. Real-robot experiments on the Piper platform demonstrate that PrimingVLA can successfully execute all four tabletop tasks. Figure 5(b) compares predicted joint actions with the corresponding ground-truth actions recorded during human teleoperation. The predictions capture several temporal trends in the demonstrations, showing meaningful correspondence despite joint-specific differences in timing and amplitude. Appendix C provides three complete comparisons, including camera observations, all six joint channels, and gripper actions.

7 Discussion

We presented PrimingVLA, a parameter-efficient VLA framework that learns action structure before introducing multimodal conditioning. Self-Supervised Action Priming gives action prompts a task-specific starting point, reducing the need to learn trajectory structure and multimodal conditioning simultaneously. The conditioning bridge builds on this preparation to enable observation-guided control, while CPC limits disproportionate action-prompt updates during joint adaptation. These mechanisms help a compact prompt interface make effective use of the multimodal and control capabilities in both frozen backbones. The benchmark results and ablations support this design, demonstrating that action preparation and update coordination enable effective VLA adaptation with fewer than 0.05M trainable parameters.

Limitations and scope. The tangent analysis characterizes local sensitivity on sampled demonstrations; its implications for semantic preservation and CPC require further validation. Evaluation with repeated seeds and matched protocols would strengthen comparisons, while broader visual shifts, task compositions, and diverse real-world tasks offer directions for extending generalization. The compact parameter budget reduces task-specific adaptation and storage costs, although inference still requires both pretrained backbones.

AI USE STATEMENT

AI-assisted tools were used to support manuscript preparation, related work discovery, and code development. The authors take full responsibility for the accuracy and integrity of the final manuscript.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning*, volume 305, pp. 17–40, 2025.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to act anywhere with task-centric latent actions. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.014.

Dwip Dalal, Shivansh Patel, Chahit Jain, Jeonghwan Kim, Utkarsh Mishra, Alex Baratian, Hyeonjeong Ha, Heng Ji, Svetlana Lazebnik, and Unnat Jain. Generalizable VLA finetuning via representation anchoring and language-action alignment. *arXiv preprint arXiv:2607.13429*, 2026. URL https://arxiv.org/abs/2607.13429v1. Version 1; Table 1 and Appendix C.2.

Sombit Dey, Jan-Nico Zaech, Nikolay Nikolov, Luc Van Gool, and Danda Pani Paudel. ReVLA: Reverting visual domain limitation of robotic foundation models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025. URL https://ieeexplore.ieee.org/document/11128823/.

Shaoqi Dong, Chaoyou Fu, Haihan Gao, Yi-Fan Zhang, Chi Yan, Chu Wu, Xiaoyu Liu, Yunhang Shen, Jing Huo, Deqiang Jiang, Haoyu Cao, Yang Gao, Xing Sun, Ran He, and Caifeng Shan. VITA-VLA: Efficiently teaching vision-language models to act via action expert distillation, 2025. URL https://arxiv.org/abs/2510.09607.

Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *Advances in Neural Information Processing Systems*, 38:102867–102888, 2026.

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

Yuxia Fu, Zhizhen Zhang, Yuqi Zhang, Zijian Wang, Zi Huang, and Yadan Luo. MergeVLA: Cross-skill model merging toward a generalist vision-language-action agent. *arXiv preprint arXiv:2511.18810*, 2025. URL https://arxiv.org/abs/2511.18810v2. Version 2, March 2026.

Minghao Guo, Meng Cao, Jiachen Tao, Rongtao Xu, Yan Yan, Xiaodan Liang, Ivan Laptev, and Xiaojun Chang. GLaD: Geometric latent distillation for vision-language-action models. *arXiv preprint arXiv:2512.09619*, 2025. URL https://arxiv.org/abs/2512.09619v1. Version 1; Table II.

Xinyu Guo, Bin Xie, Wei Chai, Xianchi Deng, Tiancai Wang, Zhengxing Wu, and Xingyu Chen. PriorVLA: Prior-preserving adaptation for vision-language-action models. *arXiv preprint arXiv:2605.10925*, 2026a. URL https://arxiv.org/abs/2605.10925.

Yufei Guo, Yinan Wu, Haoran Duan, Guiguang Ding, and Jungong Han. PhaseLoRA: Control-regime-conditioned low-rank adaptation for continuous-action vision-language-action policies. *arXiv preprint arXiv:2608.15285*, 2026b. URL https://arxiv.org/abs/2608.15285v1. Version 1.

Chengyue Huang, Mellon M. Zhang, Robert Azarcon, Glen Chou, and Zsolt Kira. MAPS: Preserving vision-language representations via module-wise proximity scheduling for better vision-language-action generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2511.19878.

Yuhua Jiang, Junjie Lu, Xinyao Qin, Xiaoyu Chen, Kaixin Wang, Feifei Gao, and Li Zhao. VLA-GSE: Boosting parameter-efficient fine-tuning in VLA with generalized and specialized experts. *arXiv preprint arXiv:2605.06175*, 2026. URL https://arxiv.org/abs/2605.06175v2. Version 2.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025a. URL https://arxiv.org/abs/2502.19645v1. Version 1; Table IV.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, volume 270, pp. 2679–2713, 2025b.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

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

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

Xuan Wang, Yinan Wu, Haoran Duan, and Jungong Han. Quovla: Quotient space for vision-language-action models. *arXiv preprint arXiv:2605.24890*, 2026a.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pp. 18638–18646, 2026b.

Yan Zhang, Yinan Wu, Haoran Duan, and Jungong Han. Cofactvla: Deconfounding vision-language-action models via counterfactual intervention. *arXiv preprint arXiv:2608.04396*, 2026.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025. URL https://arxiv.org/abs/2510.03827v1. Version 1; Tables 2–5.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, volume 229, pp. 2165–2183, 2023.