WHEN SCENES SHIFT: TRANSITION-REFERENCED ADAPTATION FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies can degrade substantially under scene shifts even when task-relevant state evolution retains a similar structure. In few-shot adaptation, target demonstrations directly supervise *what to do* in the new scene, but provide limited guidance on how shifted observations should reconnect to the policy's existing control representations. Motivated by these observations, we introduce **SHIFTVLA**, a transition-referenced framework that treats the relation between internal control and task-relevant state evolution as a reference for few-shot scene adaptation. ***(i)*** To construct this reference, a transition learner captures demonstrated state progression through intermediate-state prediction and aligns transition relations with action-chunk similarity. ***(ii)*** A lightweight reader maps depth-aggregated Action-Expert control states into the transition space, while joint grounding shapes the underlying control representations with both transition and native action supervision. ***(iii)*** During target adaptation, source-calibrated control readouts drive depth-wise refinement, while gradient projection suppresses action-update components that conflict with the established correspondence. Under low-shot adaptation, **SHIFTVLA** improves average success over the strongest baseline by 1.8 and 3.5 percentage points on LIBERO-Plus and RoboTwin 2.0, respectively, and improves real-robot task success by 7.0 percentage points under controlled scene shifts.

1 INTRODUCTION

A robot that already knows how to grasp an object should not have to learn *how to grasp* again simply because it encounters a different scene. Scene shifts can substantially change the visual observations and language context presented to a vision-language-action (VLA) policy. Yet task execution may still follow a similar state progression under these shifts, as illustrated in Fig. 1 (left). We refer to the task-relevant state evolution induced by an action segment as an *execution transition*. Existing robustness evaluations nevertheless show that competence acquired in familiar environments does not reliably transfer to such deployment shifts Fei et al. (2025); Li et al. (2026b). This raises a natural question: *when adapting to a new scene, what should a VLA actually learn?*

In new-scene deployment, few target demonstrations may be available for adaptation (Li et al., 2026b; Kang et al., 2026). A natural approach is to continue optimizing the policy's original action-learning objective on these demonstrations (Kim et al., 2025). This directly supervises *what to do* at the observed target states, but provides limited guidance on how shifted observations should reconnect to the policy's existing control representations. Consequently, adaptation needs a reference for how this existing control representations relates to task execution across scenes.

This motivates a *transition-referenced* view of scene adaptation, illustrated in Fig. 1 (middle), ***control–transition correspondence*** between the policy's control representations and execution transitions serves as a reference for adapting to scene shifts. We instantiate this view in SHIFTVLA, which learns a transition reference, establishes its correspondence with Action-Expert control states, and organizes few-shot scene adaptation around the resulting control–transition correspondence.

Specifically, ***(i)*** SHIFTVLA first constructs a ***transition reference*** from complete demonstrations. Transition tokens inferred from segment endpoints are constrained by intermediate-state prediction, while their neighborhood geometry is aligned with relations induced by the corresponding action

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.

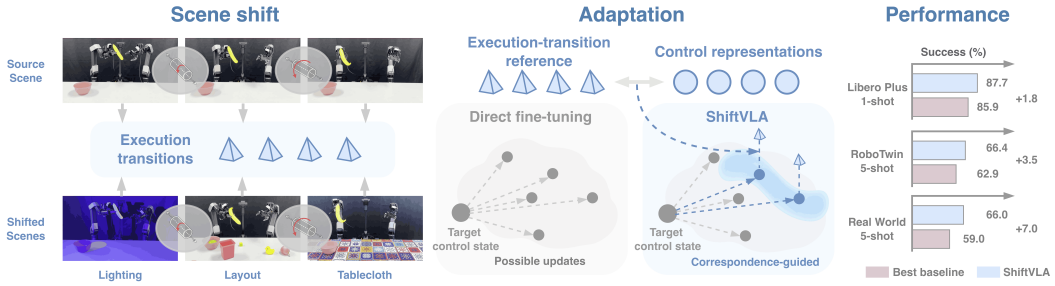


Figure 1: **Left:** Despite scene shifts, execution transitions can retain similar structure. **Middle:** SHIFTVLA uses the control–transition correspondence as a reference to guide few-shot adaptation. **Right:** Performance comparison on LIBERO-Plus, RoboTwin 2.0, and real-robot manipulation.

chunks. Together, these objectives make the reference capture both how state evolves within a segment and how different transitions relate in execution, rather than endpoint change alone. ***(ii)*** We next connect this transition structure to the policy's internal control process. A lightweight reader maps Action-Expert control states into the same transition space, establishing the ***control–transition correspondence*** while preserving their action-generation role through the native action objective. This grounding makes demonstrated execution structure accessible through the policy's current control states, enabling the transition reference to guide adaptation during control computation. ***(iii)*** During target adaptation, SHIFTVLA performs ***correspondence-guided adaptation***, with this correspondence providing a reference for control refinement and parameter updates. Source-calibrated control readouts modulate low-rank control refinement across Action-Expert depth, while gradient projection removes action-update components that would weaken the grounded correspondence. At inference, the transition teacher and future observations are no longer required; the fixed reader, source reference, and adapted low-rank modules remain active during policy denoising.

Our contributions are summarized as follows:

- We formulate few-shot scene adaptation from a *transition-referenced* perspective, where the correspondence between a VLA's internal control representations and execution transitions provides a reference for adapting to scene shifts.
- We establish this correspondence by learning a transition reference from complete demonstrations and mapping Action-Expert control states into the same transition space, grounding internal control in demonstrated execution structure.
- We develop correspondence-guided adaptation that retains the established control–transition correspondence as a reference for depth-wise control refinement and parameter updates throughout target-scene adaptation.
- We validate SHIFTVLA across LIBERO-Plus, RoboTwin 2.0, and real-robot manipulation under diverse scene shifts, showing improved average success over strong adaptation baselines in low-shot settings.

2 METHOD

As shown in Fig. 2, SHIFTVLA organizes few-shot scene adaptation around a transition reference and its correspondence with the VLA's internal control states. The transition reference is learned from complete demonstrations to capture state progression over an action segment together with execution-related relations across transitions. We then map Action-Expert control states into the same transition space, establishing the control–transition correspondence between demonstrated execution and internal control. During target adaptation, this correspondence provides the reference for depth-wise control refinement and parameter updates.

2.1 PROBLEM SETUP AND PRELIMINARIES

Few-shot scene adaptation. Let $\mathcal{D}_s$ denote the source demonstrations available before adaptation, and let $\mathcal{D}_t$ contain only a small number of demonstrations from the target scenes. Each demonstration contains RGB observations I_t, proprioceptive state p_t, task instruction g, and robot action a_t. We write the current policy context as $c_t = (I_t, p_t, g)$ and an H-step action chunk as

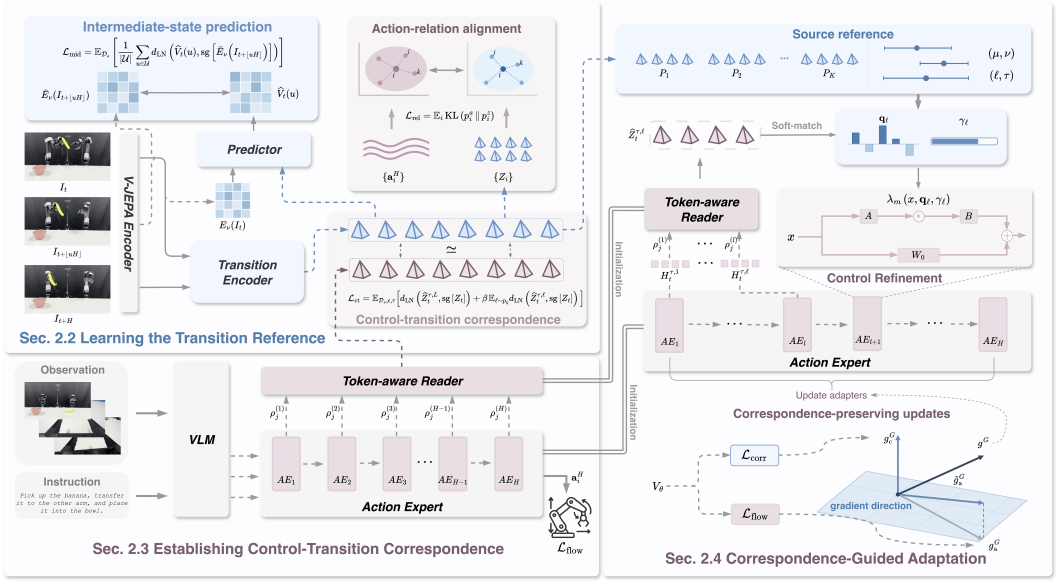


Figure 2: Complete demonstrations shape a transition reference through intermediate-state prediction and action-relation alignment. Action-Expert control states are mapped into the same transition space to establish the control–transition correspondence. During few-shot adaptation, source-calibrated control readouts guide depth-wise control refinement, while the grounded correspondence is retained as a reference for correspondence-preserving parameter updates.

$\mathbf{a}_t^H = (a_t, \ldots, a_{t+H-1})$. We consider unlabeled scene shifts that preserve manipulation semantics and the action interface while changing visual, linguistic, or environmental conditions. Future observations from complete demonstrations are used only as supervision during training and adaptation, and are never required when the deployed policy selects an action.

Flow-matching VLA. We build on a VLA with a pretrained vision-language backbone and an L-block flow-matching Action Expert, following $\pi_{0.5}$ (Intelligence et al., 2025). Given a demonstration action chunk and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, we use

$$\mathbf{a}_t^{H,\tau} = (1-\tau)\mathbf{a}_t^H + \tau\epsilon, \qquad \tau \in [0,1], \tag{1}$$

where $\tau = 0$ corresponds to the demonstrated action and $\tau = 1$ to Gaussian noise. The native policy objective is

$$\mathcal{L}_{\text{flow}}(\theta; \mathcal{D}) = \mathbb{E}_{\mathcal{D},\epsilon,\tau}\left[\left\| v_\theta(c_t, \mathbf{a}_t^{H,\tau}, \tau) - (\epsilon - \mathbf{a}_t^H)\right\|_2^2\right]. \tag{2}$$

Let $H_t^{\tau,\ell} \in \mathbb{R}^{N_a \times d_h}$ denote the sequence of Action-Expert tokens after block ℓ. Here, t denotes environment time, $u \in (0,1)$ transition progress, τ flow time, and ℓ Action-Expert depth.

2.2 Learning the Transition Reference

Endpoints summarize *what* changes over a demonstrated interaction, but do not fully characterize *how* the state evolves toward that outcome. Consequently, we infer a transition reference from the segment endpoints and constrain it with the complete demonstration: intermediate-state prediction captures within-segment state progression, while action-relation alignment organizes how different transitions relate in execution.

We initialize a visual encoder E_ν from V-JEPA 2 (Assran et al., 2025). For a valid H-step segment, a transition encoder F_ϕ maps the endpoint features to S transition tokens,

$$Z_t = F_\phi(E_\nu(I_t), E_\nu(I_{t+H})) \in \mathbb{R}^{S \times d_z}. \tag{3}$$

The transition encoder outputs S transition tokens with consistent indexing across demonstrations.

To make Z_t explain the demonstrated state progression, the intermediate-state predictor G_ω attends to the current visual features $E_\nu(I_t)$ and Z_t as shared memory and predicts the intermediate visual

representation at normalized progress $u \in \mathcal{U} \subset (0, 1)$:

$$\widehat{V}_t(u) = G_\omega(E_\nu(I_t), Z_t, u)\,. \tag{4}$$

The prediction $\widehat{V}_t(u)$ is matched to the stop-gradient EMA target $\bar{E}_\nu(I_{t+\lfloor uH \rfloor})$ in $\mathcal{L}_{\text{mid}}$. We denote by $d_{\text{LN}}(X, Y) = \text{mean}\,|\,\text{LN}(X) - \text{LN}(Y)|$ the distance between layer-normalized features.

Intermediate prediction constrains what evolution Z_t should explain, but does not determine how distinct transitions relate in terms of execution. We align transition-space neighborhoods with relations induced by the demonstrated action chunks.

For samples i and j in a batch,

$$p^a_{ij} \propto \exp\left(-\frac{d_a(\mathbf{a}^H_i, \mathbf{a}^H_j)}{T_{\text{rel}}}\right), \qquad p^z_{ij} \propto \exp\left(\frac{\langle \zeta_i, \zeta_j \rangle}{T_{\text{rel}}}\right), \tag{5}$$

where d_a is the standardized action-chunk distance and ζ_i is the normalized pooled representation of Z_i. Both distributions are normalized over $j \neq i$. We train the transition representation with

$$\begin{aligned} \mathcal{L}_{\text{mid}} &= \mathbb{E}_{\mathcal{D}_s}\left[\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} d_{\text{LN}}\left(\widehat{V}_t(u), \text{sg}[\bar{E}_\nu(I_{t+\lfloor uH \rfloor})]\right)\right], \\ \mathcal{L}_{\text{rel}} &= \mathbb{E}_i\,\text{KL}\left(p^a_i \,\|\, p^z_i\right), \qquad \mathcal{L}_{\text{tr}} = \mathcal{L}_{\text{mid}} + \lambda_{\text{rel}}\mathcal{L}_{\text{rel}}. \end{aligned} \tag{6}$$

The two objectives play complementary roles: $\mathcal{L}_{\text{mid}}$ constrains within-segment state progression, whereas $\mathcal{L}_{\text{rel}}$ organizes transition geometry according to execution-related relations. After training, E_ν and F_ϕ are frozen and define the transition teacher $T_{\text{tr}}(I_t, I_{t+H}) \equiv Z_t$.

2.3 Establishing Control–Transition Correspondence

An execution transition describes the realized state evolution of an action segment, whereas the Action Expert represents the internal control process that produces it. The transition reference is obtained from complete demonstrations, while target adaptation depends on the policy's current control states. Accordingly, We map Action-Expert control states into the transition space to establish the control–transition correspondence.

For each completed Action-Expert depth ℓ, we aggregate the available hidden states into a prefix control representation,

$$\rho^{(\ell)}_j = \frac{\exp(\kappa_j)}{\sum_{m=1}^{\ell}\exp(\kappa_m)}, \qquad \bar{H}^{\tau,\ell}_t = \sum_{j=1}^{\ell}\rho^{(\ell)}_j\,\text{LN}\left(H^{\tau,j}_t\right). \tag{7}$$

A shared token-aware reader R_ψ maps the aggregated action-token sequence into the transition space through a lightweight two-layer Transformer decoder with action-position embeddings and S learned queries,

$$\widehat{Z}^{\tau,\ell}_t = \text{TDec}_\psi\Big(Q_{\text{read}}; \Pi_{\text{r}}(\bar{H}^{\tau,\ell}_t) + E^a_{\text{pos}}\Big) \in \mathbb{R}^{S\times d_z}. \tag{8}$$

where Π_{r} is a linear projection to dimension d_z and E^a_{pos} denotes the action-position embeddings. At depth ℓ, the reader produces an indexed S-token transition readout from blocks $1{:}\ell$, available to subsequent Action-Expert blocks.

To establish the control–transition correspondence, we supervise Action-Expert readouts against the frozen transition teacher while retaining the native action objective. We define the control–transition grounding loss as

$$\mathcal{L}_{\text{ct}} = \mathbb{E}_{\mathcal{D}_s,\epsilon,\tau}\Big[d_{\text{LN}}\left(\widehat{Z}^{\tau,L}_t, \text{sg}\left[Z_t\right]\right) + \beta\,\mathbb{E}_{\ell\sim p_{\text{g}}} d_{\text{LN}}\left(\widehat{Z}^{\tau,\ell}_t, \text{sg}\left[Z_t\right]\right)\Big]. \tag{9}$$

where p_{g} samples intermediate Action-Expert depths and β weights the intermediate-depth term. The overall grounding objective is $\mathcal{L}_{\text{ground}} = \mathcal{L}_{\text{flow}}(\theta; \mathcal{D}_s) + \lambda_{\text{g}}\mathcal{L}_{\text{ct}}$.

We first optimize the reader and depth-aggregation weights with the policy frozen, and then jointly optimize the policy, reader, and depth weights on source demonstrations. The first stage fits the transition readout before updating the policy, whereas joint grounding allows the transition objective to shape the underlying control representations while $\mathcal{L}_{\text{flow}}$ preserves their action-generation role.

2.4 CORRESPONDENCE-GUIDED ADAPTATION

With the control–transition correspondence established, target control states can be read in the transition space. We calibrate these readouts against source transition structure to obtain a transition-reference coordinate for depth-wise control refinement, while the grounded correspondence remains the reference for parameter updates.

Transition-reference coordinates. We summarize source transition structure with K prototypes $\mathcal{P} = \{P_k\}_{k=1}^{K}$ obtained from source transition-teacher outputs. With fixed token indexing, we use slot-wise normalized cosine distance, $d_{\mathrm{tr}}(Z, P_k) = 1 - \frac{1}{S}\sum_{s=1}^{S}\langle \widetilde{z}_s, \widetilde{p}_{k,s}\rangle$, where $\widetilde{z}_s = \mathrm{LN}(z_s)/\|\,\mathrm{LN}(z_s)\|_2$.

For each Action-Expert depth ℓ, flow-time bin b, and prototype c, source trajectories define reader-to-prototype statistics $\boldsymbol{\mu}_{\ell bc}$ and $\boldsymbol{v}_{\ell bc}$. Given a current readout $\widehat{Z}_t^{\tau,\ell}$, let $\mathbf{d}_\ell = (d_{\ell 1}, \ldots, d_{\ell K})$ collect its distances to the prototype bank. We softly aggregate the source statistics with $p_c \propto \exp(-d_{\ell c}/T_{\ell b})$, obtaining matched statistics $\boldsymbol{\mu}_{\ell b}$ and $\boldsymbol{v}_{\ell b}$. The *transition-reference coordinate* is then

$$\mathbf{q}_\ell = \mathrm{clip}\left(\frac{\mathbf{d}_\ell - \boldsymbol{\mu}_{\ell b}}{\sqrt{\boldsymbol{v}_{\ell b} + \varepsilon}}, -q_{\max}, q_{\max}\right). \tag{10}$$

Thus, $\mathbf{q}_\ell$ measures the current control readout relative to source transition structure at the corresponding Action-Expert depth and flow time. Further details of prototype construction and source-statistics calibration are provided in Appendix A.2.

Correspondence-guided control refinement. During target adaptation, only low-rank adapter parameters are optimized. Within the Action Expert, the transition-reference coordinate further modulates their rank-wise activation along depth. For a correspondence-conditioned projection m with frozen base weight $W_{0,m}$,

$$y = W_{0,m}x + \frac{\alpha_{\mathrm{lr}}}{r} B_m\left[\boldsymbol{\lambda}_m(x, \mathbf{q}_\ell, \gamma_\ell) \odot A_m x\right], \tag{11}$$

where A_m and B_m define the low-rank basis. The rank activation is adjusted as $\boldsymbol{\lambda}_m = \boldsymbol{\lambda}_m^0(x) \odot \exp[\gamma_\ell \Delta_m(x, \mathbf{q}_\ell)]$, where $\boldsymbol{\lambda}_m^0(x)$ denotes the base activation and Δ_m is a lightweight bounded correction conditioned on $\mathbf{q}_\ell$. A source-derived weight $\gamma_\ell \in [0, 1]$ scales this correction. Further details are provided in Appendix A.3.

Refinement proceeds along Action-Expert depth. After block ℓ, source calibration maps the readout $\widehat{Z}_t^{\tau,\ell}$ to $(\mathbf{q}_\ell, \gamma_\ell)$. These quantities modulate the low-rank adapters in block $\ell{+}1$, written as $H_t^{\tau,\ell+1} = \mathcal{B}_{\ell+1}(H_t^{\tau,\ell}; \mathbf{q}_\ell, \gamma_\ell)$, where $\mathcal{B}_{\ell+1}$ denotes the $(\ell{+}1)$-th Action-Expert block. First-block initialization is detailed in Appendix A.3.

Correspondence-preserving parameter updates. Target action supervision directly supervises the demonstrated actions, but only indirectly constrains how the grounded control representations change. With only a few target demonstrations, action fitting can improve while weakening the grounded control–transition correspondence. We retain this correspondence as an explicit reference during parameter adaptation through $\mathcal{L}_{\mathrm{corr}} = \mathbb{E}_{\mathcal{D}_t,\epsilon,\tau} d_{\mathrm{LN}}\left(\widehat{Z}_t^{\tau,L}, \mathrm{sg}[T_{\mathrm{tr}}(I_t, I_{t+H})]\right)$.

For an adapted parameter group G, let $g_{\mathrm{a}}^G = \nabla_G \mathcal{L}_{\mathrm{flow}}$ and $g_{\mathrm{c}}^G = \nabla_G \mathcal{L}_{\mathrm{corr}}$ denote the action and correspondence gradients. Under an action-only step $G' = G - \eta g_{\mathrm{a}}^G$, the correspondence loss changes to first order as

$$\mathcal{L}_{\mathrm{corr}}(G') - \mathcal{L}_{\mathrm{corr}}(G) \approx -\eta\left\langle g_{\mathrm{c}}^G, g_{\mathrm{a}}^G\right\rangle. \tag{12}$$

Hence, $\langle g_{\mathrm{a}}^G, g_{\mathrm{c}}^G\rangle < 0$ indicates that an action-improving step would increase $\mathcal{L}_{\mathrm{corr}}$ to first order, thereby weakening the grounded correspondence.

Accordingly, we leave the action gradient unchanged when the objectives are compatible and remove only its correspondence-conflicting component otherwise:

$$\widetilde{g}_{\mathrm{a}}^G = g_{\mathrm{a}}^G - \frac{\min\left(\langle g_{\mathrm{a}}^G, g_{\mathrm{c}}^G\rangle, 0\right)}{\|g_{\mathrm{c}}^G\|_2^2 + \varepsilon} g_{\mathrm{c}}^G. \tag{13}$$

The final adaptation gradient is $g^G = \widetilde{g}_{\mathrm{a}}^G + \lambda_{\mathrm{corr}} g_{\mathrm{c}}^G$. The projection modifies only action-update directions that conflict with the correspondence objective, while the correspondence gradient supplies the explicit preservation term. At inference, the transition teacher is removed, while the reader, source transition structure, and adapted low-rank modules remain active during flow denoising.

Table 1: **Few-shot scene adaptation on LIBERO-Plus.** One- and five-shot adaptation use $\sim$1.39% and $\sim$6.97% of the available target training trajectories, respectively. Camera, Light, Background, and Noise are used for adaptation; Language, Robot, and Layout are held out. **Src. Avg.** reports the average over the four original LIBERO suites, and **Avg.** is rollout-weighted across the seven shifts.

Method	Src.	Cam.	Light	Bg.	Noise	Lang.	Robot	Layout	Avg.
One-shot ($\sim$ 1.39%)									
OpenVLA-OFT (Kim et al., 2025)	94.4	74.0	94.5	90.2	82.4	76.2	30.1	70.3	72.4
π_0 (Black et al., 2024)	93.2	30.1	88.5	87.8	81.2	59.0	8.1	69.2	58.1
$\pi_{0.5}$ (Intelligence et al., 2025)	96.1	80.9	95.0	95.8	89.4	83.9	73.9	85.2	85.5
VLA-JEPA (Sun et al., 2026)	96.4	76.8	96.1	94.9	78.5	82.1	65.9	85.3	81.6
FLA (Li et al., 2026b)	96.2	80.2	**96.6**	95.7	90.6	85.0	75.0	84.2	85.9
Robotic Steering (Mitra et al., 2025)	95.9	79.6	94.4	93.1	88.3	82.8	71.2	83.5	83.9
RETAIN (Yadav et al., 2026)	96.8	81.7	96.4	95.4	91.9	83.2	74.1	84.3	85.8
ShiftVLA	**97.0**	**84.3**	96.3	**95.9**	**94.1**	**86.6**	74.5	**86.6**	**87.7**
Five-shot ($\sim$ 6.97%)									
OpenVLA-OFT (Kim et al., 2025)	92.1	75.7	94.9	91.3	82.6	78.1	25.9	71.7	72.7
π_0 (Black et al., 2024)	92.6	34.6	89.1	90.3	85.6	58.2	8.2	68.9	59.7
$\pi_{0.5}$ (Intelligence et al., 2025)	95.7	82.1	96.3	95.5	90.5	83.1	73.3	84.2	85.6
VLA-JEPA (Sun et al., 2026)	96.1	77.9	95.8	94.8	80.2	82.9	64.5	84.3	81.8
FLA (Li et al., 2026b)	96.4	82.2	**96.4**	95.6	92.8	84.4	**73.8**	84.5	86.3
Robotic Steering (Mitra et al., 2025)	95.6	80.7	94.9	92.8	89.2	82.2	70.8	83.9	84.2
RETAIN (Yadav et al., 2026)	96.8	83.7	95.8	94.7	93.2	84.2	73.0	84.1	86.3
ShiftVLA	**96.9**	**89.9**	96.2	**95.8**	**95.4**	**84.6**	73.1	**86.0**	**88.1**

3 Experiments

3.1 Experimental Setup

Few-shot adaptation protocol. In the k-shot setting, each task receives k complete demonstrations per target condition used for adaptation. When multiple target conditions are available, demonstrations are pooled to adapt a single policy, without perturbation labels or condition-specific routing.

Simulation Benchmarks. We evaluate ShiftVLA on LIBERO-Plus (Fei et al., 2025) and RoboTwin 2.0 (Mu et al., 2025). For LIBERO-Plus, policies are trained on standard LIBERO (Liu et al., 2023). We jointly adapt a single policy using Camera, Light, Background, and Noise demonstrations available in the released training data, and evaluate it on all seven LIBERO-Plus shifts; Language, Robot, and Layout are held out from adaptation. For RoboTwin 2.0, we evaluate 20 bimanual manipulation tasks, adapting policies from the Easy setting to the randomized Hard setting. For each task, k demonstrations are sampled from the Hard training split for adaptation, and evaluation follows the standard Hard protocol.

Real-World Settings. We further evaluate five long-horizon manipulation tasks on the GALAXEA R1 Lite (Jiang et al., 2025) platform. For each task, 100 source demonstrations are used for policy training, with 5 target demonstrations for each of the four scene shift categories. We conduct 25 trials for each task–scene pair on this platform.

Implementation details. Training is conducted on 8$\times$ NVIDIA A800 GPUs, while real-world inference runs on a single NVIDIA RTX 5090 (32 GB). Exact support-set construction, benchmark splits, real-world task and scene configurations, and optimization details are provided in Appendix B.

3.2 Simulation Results

LIBERO-Plus. Table 1 reports few-shot adaptation across the LIBERO-Plus scene shifts. ShiftVLA achieves 87.7% and 88.1% average success under one- and five-shot adaptation, respectively, exceeding the best baseline by 1.8 points in both settings. Under five-shot adaptation, the largest margins occur on Camera and Noise, where ShiftVLA improves over the best baseline by 6.2 and 2.2 points, respectively. This pattern is consistent with recalibrating shifted control readouts against source transition structure. Performance on the original LIBERO suites remains high at 97.0% and 96.9% after one- and five-shot adaptation, respectively.

Table 2: **Few-shot adaptation on RoboTwin 2.0 Hard.** Five- and ten-shot results are shown for eight representative tasks; **Avg.** is computed over all 20 tasks. Full results are in Appendix C.2.

Method	Dump BB	Move PA	Place Cont. Plate	Place A2B-L	Shake Bottle	Place Obj. Basket	Open L	Stack Bowls 3
Five-shot ($\sim 1.0\%$)								
$\pi_{0.5}$ (Intelligence et al., 2025)	75%	65%	76%	40%	92%	44%	70%	56%
FLA (Li et al., 2026b)	79%	68%	78%	41%	93%	46%	73%	57%
Robotic Steering (Mitra et al., 2025)	72%	60%	72%	35%	92%	39%	73%	52%
RETAIN (Yadav et al., 2026)	78%	67%	77%	42%	**94%**	47%	72%	59%
ShiftVLA	**80%**	**69%**	**80%**	**49%**	92%	**53%**	**75%**	**67%**
Ten-shot ($\sim 2.0\%$)								
$\pi_{0.5}$ (Intelligence et al., 2025)	81%	69%	76%	40%	93%	48%	73%	61%
FLA (Li et al., 2026b)	**86%**	71%	78%	41%	**94%**	51%	75%	63%
Robotic Steering (Mitra et al., 2025)	78%	66%	73%	37%	93%	43%	77%	57%
RETAIN (Yadav et al., 2026)	83%	71%	79%	42%	93%	50%	75%	64%
ShiftVLA	85%	**73%**	**80%**	**48%**	92%	**57%**	**77%**	**72%**

RoboTwin 2.0. Table 2 reports few-shot adaptation on 20 bimanual manipulation tasks under the jointly randomized RoboTwin 2.0 Hard setting. With five-shot demonstrations per task, ShiftVLA achieves 66.4% average success, outperforming the strongest baseline by 3.5 points. With ten-shot demonstrations, it reaches 67.3%, exceeding the strongest baseline by 3.4 points. These results extend the gains observed under individual LIBERO-Plus shifts to jointly randomized scene conditions. Full task-wise results are provided in Appendix C.2.

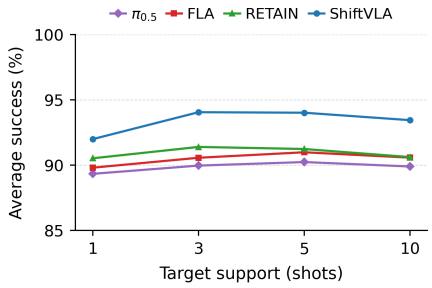


Figure 3: **Few-shot data efficiency on LIBERO-Plus.** Average success over Camera, Light, Background, and Noise with increasing target support.

Few-shot scaling. Figure 3 shows performance as target support increases from one to ten demonstrations per task and condition over Camera, Light, Background, and Noise. ShiftVLA improves from one to three shots and changes little at five shots. At ten shots, all methods exhibit a small drop, while the relative ordering remains similar. ShiftVLA stays above the competing methods across the evaluated support sizes, including the one-shot setting.

3.3 Real-World Results

Table 3 reports five-shot scene adaptation on the GALAXEA R1 Lite. Across all five tasks and four target-scene shifts, ShiftVLA achieves 66.0% TSR and 74.2% CSR, improving over $\pi_{0.5}$ by 7.0 and 5.7 points, respectively. Across the representative task–shift pairs, ShiftVLA also improves cumulative execution progress on all five tasks, with larger margins on Ring Stacking, Banana Handover, Towel Folding, and Bridge Building. The gains in both final completion and intermediate progress support effective few-shot adaptation across multi-stage real-world manipulation. Figure 4 illustrates representative scene shifts and execution stages.

3.4 Ablation Studies

Table 4 ablates source grounding and the correspondence-guided components used during target adaptation. Starting from standard low-rank adaptation with the native action objective, the Stage-2 checkpoint yields 89.6/90.7% one-/five-shot success, compared with 89.3/90.2% for the ungrounded source policy. Adding $\mathcal{L}_{\text{corr}}$ further raises performance to 90.6/91.5%. From this anchor, correspondence-guided control refinement adds 0.7/1.2 points, while gradient projection adds 0.4/0.8 points. Combining both reaches 92.0/94.0%. These complementary gains indicate that the grounded correspondence contributes through both control refinement and parameter updates.

3.5 Understanding the Control–Transition Correspondence

We further analyze the proposed control–transition correspondence from two perspectives: the execution structure captured by the learned transition reference, and its relation to the internal control representations of the Action Expert. Further details of these analyses are provided in Appendix D.

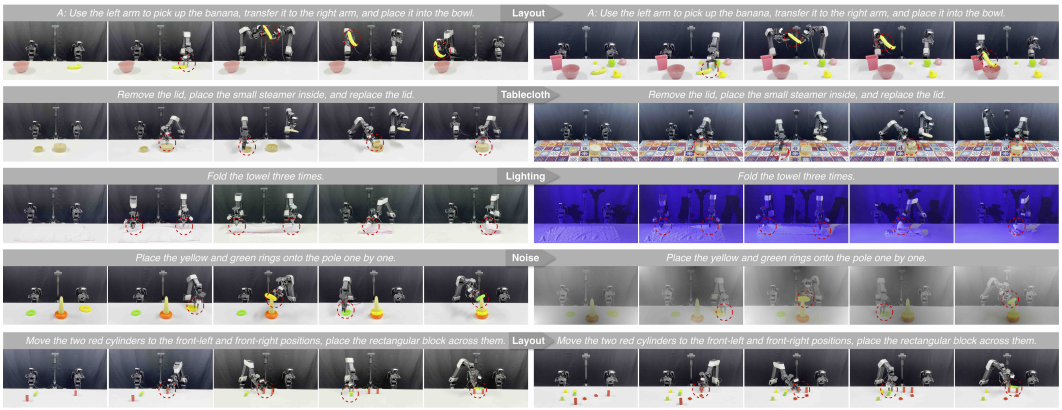


Figure 4: **Representative real-world scene adaptation.** Each row shows the source scene and representative execution stages for one target-scene shift. The noise effect is rendered for illustration. Full visualizations are provided in Appendix C.3.

Table 3: **Five-shot real-world scene adaptation on GALAXEA R1 Lite.** Representative task–shift pairs are shown, with checkpoint columns reporting cumulative successes out of 25 trials. **Avg.** is computed over all five tasks and four scene variations. Full results are in Appendix C.3.

	Banana Handover *(Layout)*				**Ring Stacking** *(Noise)*			**Towel Folding** *(Lighting)*			
Method	Left-Arm Grasp	+Hand-over	+Place in Bowl	CSR	Yellow Ring on Pole	+Green Ring on Pole	CSR	First Fold	+Second Fold	+Final Fold	CSR
OpenVLA-OFT (Kim et al., 2025)	15/25	10/25	7/25	42.7	16/25	10/25	52.0	14/25	11/25	8/25	44.0
$\pi_{0.5}$ (Intelligence et al., 2025)	19/25	15/25	13/25	62.7	20/25	15/25	70.0	18/25	16/25	15/25	65.3
ShiftVLA	20/25	17/25	15/25	69.3	21/25	18/25	78.0	20/25	18/25	16/25	72.0

	Steamer Assembly *(Tablecloth)*				**Bridge Building** *(Layout)*				**Average**	
Method	Remove Lid	+Insert Inner Steamer	+Replace Lid	CSR	Place Left Support	+Place Right Support	+Place Beam	CSR	TSR	CSR
OpenVLA-OFT (Kim et al., 2025)	15/25	12/25	9/25	48.0	15/25	11/25	7/25	44.0	36.4	49.0
$\pi_{0.5}$ (Intelligence et al., 2025)	20/25	18/25	16/25	72.0	20/25	17/25	14/25	68.0	59.0	68.5
ShiftVLA	20/25	18/25	17/25	73.3	21/25	19/25	16/25	74.7	66.0	74.2

Execution structure in the transition reference. We examine whether the action-related organization of the transition reference remains consistent across scene shifts. Figure 5 compares transition similarity as a function of action distance for pairs sampled within and across perturbations. The two pairing regimes follow similar trends, suggesting that the action–transition relation is retained across the analyzed shifts. Figure 6(a) visualizes the transition latents in a shared t-SNE embedding. Action-cluster coloring shows local groupings associated with similar control segments, whereas Camera, Light, Background, and Noise are interspersed across these regions. Together, these views indicate that the learned action-related organization remains similar across the scene shifts.

Control–transition structure in the Action Expert. We compare the geometric agreement of VLM and Action-Expert representations with the learned transition reference using task-wise Spearman correlations between pairwise distance matrices over the same execution windows. The comparison operates directly on native model representations, without the reader or any learned projection. As shown in Fig. 6(b), before grounding, Action-Expert representations exhibit stronger agreement with the transition reference than VLM representations, with the gap increasing toward later depths. The same relative pattern remains after grounding. This supports establishing the control–transition correspondence in the Action Expert.

4 Related Work

VLA adaptation under distribution shifts. Recent VLA policies build on large-scale pretraining over diverse robot and multimodal data (Brohan et al., 2023; O'Neill et al., 2024; Qu et al., 2025; Bjorck et al., 2025). Generalist robot-policy scaling has been explored by RT-1 (Brohan et al., 2022) and Octo (Team et al., 2024). Despite pretraining, policies such as OpenVLA, π_0, and $\pi_{0.5}$ (Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025) are adapted with additional robot demonstrations for deployment conditions beyond pretraining, through full or parameter-efficient

Table 4: **Ablation of transition-referenced adaptation.** Results are averaged over the target shifts used for adaptation; all variants share the same base adapter and target support.

Source	Target Adaptation			LIBERO-Plus	
Ground.	$\mathcal{L}_{\text{corr}}$	Refine.	Proj.	1-shot	5-shot
–	–	–	–	89.3	90.2
✓	–	–	–	89.6	90.7
✓	✓	–	–	90.6	91.5
✓	✓	✓	–	91.3	92.7
✓	✓	–	✓	91.0	92.3
✓	✓	✓	✓	**92.0**	**94.0**

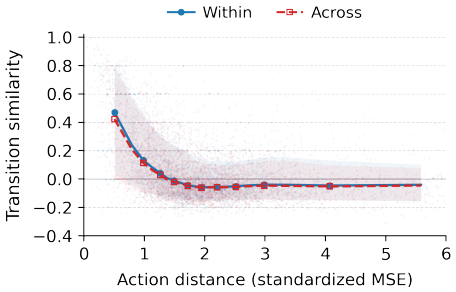
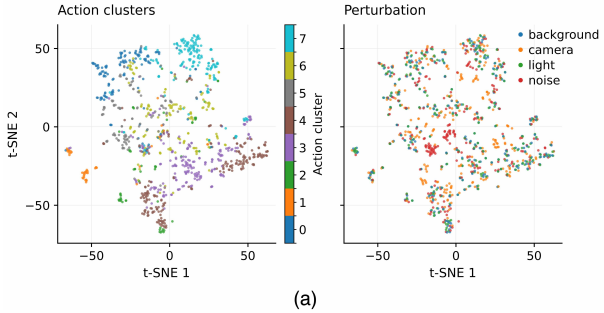
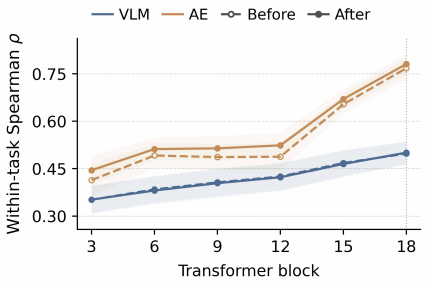


Figure 5: **Action–transition geometry within and across scene shifts.**



(a)



(b)

Figure 6: **Representation analysis of the control–transition correspondence.** (a) Joint t-SNE visualization of the same transition representations, colored by action clusters and scene perturbations, respectively. (b) Layer-wise geometric agreement of VLM and Action-Expert representations with transition-reference geometry, before and after grounding.

fine-tuning (Hu et al., 2021). OpenVLA-OFT (Kim et al., 2025) studies action representations, decoding, and objectives for VLA fine-tuning. Limited-data methods preserve pretrained capabilities through parameter merging (Yadav et al., 2026), recalibrate visual representations under viewpoint and appearance shifts (Li et al., 2026b), or selectively update task-relevant components (Mitra et al., 2025). Other deployment-time approaches use latent action-space guidance (Zhang et al., 2026c) or feedback-driven adaptation (Zang et al., 2026). ShiftVLA instead organizes few-shot scene adaptation around a transition reference grounded in the policy's internal control states.

Predictive and dynamics representations for VLA control. A complementary line of work models how robot interactions evolve through predictive representations. DreamVLA (Zhang et al., 2026a) and Cosmos Policy (Kim et al., 2026) incorporate future or world prediction into policy learning, while WorldVLA (Cen et al., 2025) jointly models actions and future visual states. At the representation level, UP-VLA (Zhang et al., 2025) combines multimodal understanding with future prediction, VLA JEPA (Sun et al., 2026) learns action-relevant latent dynamics, and FutureVLA (Xu et al., 2026) models joint visuomotor evolution. Fast-WAM (Yuan et al., 2026) further studies predictive world modeling without explicit future generation at inference. DeFI (Zhang et al., 2026b) separates forward- and inverse-dynamics pretraining, while LARA (Liu et al., 2026) and Motus (Bi et al., 2026) learn compact transition or motion representations for control. Structured priors have been grounded into VLA representations or action pathways (Li et al., 2026a). ShiftVLA instead connects demonstrated execution transitions to Action-Expert control states and retains the grounded correspondence as a reference for few-shot scene adaptation.

5 Conclusion

We presented ShiftVLA, a transition-referenced framework for few-shot VLA adaptation under scene shifts. ShiftVLA learns a transition reference from complete demonstrations and grounds it in Action-Expert control states, establishing a control–transition correspondence that guides depth-wise control refinement and is preserved during parameter updates. Across LIBERO-Plus, RoboTwin 2.0, and real-robot manipulation, ShiftVLA improves low-shot adaptation over strong baselines. More broadly, our results support viewing scene adaptation as recalibrating a pretrained policy's control representations against demonstrated execution transitions.

6 AI USE DISCLOSURE

Generative AI tools were utilized solely to assist with English language refinement and to ensure formatting and stylistic uniformity throughout the manuscript. These tools did not contribute to conceiving research concepts, constructing hypotheses or mathematical propositions, devising research methodologies and experimental protocols, implementing the proposed method, or analyzing and interpreting experimental outcomes. All AI-assisted revisions have been reviewed and verified by the authors, who bear full responsibility for the final content of this work.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. \pi_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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

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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. \pi_ {0.5}: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Yunfan Jiang, Ruohan Zhang, Josiah Wong, Chen Wang, Yanjie Ze, Hang Yin, Cem Gokmen, Shuran Song, Jiajun Wu, and Li Fei-Fei. Behavior robot suite: Streamlining real-world whole-body manipulation for everyday household activities. *arXiv preprint arXiv:2503.05652*, 2025.

Taewook Kang, Taeheon Kim, Donghyun Shin, and Jonghyun Choi. Domain arithmetic: One-shot vla adaptation under environmental shifts. *arXiv preprint arXiv:2607.00666*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 132324–132345, 2026a.

Weiqi Li, Quande Zhang, Ruifeng Zhai, Liang Lin, and Guangrun Wang. Vla models are more generalizable than you think: Revisiting physical and spatial modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35025–35035, 2026b.

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

Mengya Liu, Baoxiong Jia, Jiangyong Huang, Jingze Zhang, and Siyuan Huang. Lara: Latent action representation alignment for vision-language-action models. *arXiv preprint arXiv:2606.07100*, 2026.

Chancharik Mitra, Yusen Luo, Raj Saravanan, Dantong Niu, Anirudh Pai, Jesse Thomason, Trevor Darrell, Abrar Anwar, Deva Ramanan, and Roei Herzig. Mechanistic finetuning of vision-language-action models via few-shot demonstrations. *arXiv preprint arXiv:2511.22697*, 2025.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660. IEEE, 2025.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. In *European Conference on Computer Vision*, pp. 478–497. Springer, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Xiaoxu Xu, Hao Li, Jinhui Ye, Yilun Chen, Jia Zeng, Xinyi Chen, Linning Xu, Dahua Lin, Weixin Li, and Jiangmiao Pang. Futurevla: Joint visuomotor prediction for vision-language-action model. *arXiv preprint arXiv:2603.10712*, 2026.

Yajat Yadav, Zhiyuan Zhou, Andrew Wagenmaker, Karl Pertsch, and Sergey Levine. Robust fine-tuning of vision-language-action robot policies via parameter merging. In *International Conference on Learning Representations*, volume 2026, pp. 56385–56417, 2026.

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

Zehua Zang, Xi Wang, Fuchun Sun, Xiao Xu, Lixiang Lium, Jiahuan Zhou, and Jiangmeng Li. Test-time perturbation learning with delayed feedback for vision-language-action models. *arXiv preprint arXiv:2604.18107*, 2026.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38: 24195–24228, 2026a.

Wenyao Zhang, Bozhou Zhang, Zekun Qi, Wenjun Zeng, Xin Jin, and Li Zhang. Disentangled robot learning via separate forward and inverse dynamics pretraining. *arXiv preprint arXiv:2604.16391*, 2026b.

Yang Zhang, Chenwei Wang, Yuan Zhao, Yunfei Ge, Zhenglong Sun, Xiu Li, Chi Zhang, Chenjia Bai, Xuelong Li, et al. Align-then-steer: Adapting the vision-language action models through unified latent guidance. In *International Conference on Learning Representations*, volume 2026, pp. 14717–14749, 2026c.

The appendix is organized as follows:

- Sec. A: Method and Implementation Details.
- Sec. B: Experimental Protocols.
- Sec. C: Complete Experimental Results.
- Sec. D: Additional Analysis of the Control–Transition Correspondence.
- Sec. E: Limitations and Future Work.

A Method and Implementation Details

We provide additional details for the transition reference, source grounding, and correspondence-guided adaptation described in the main text. Experimental protocols and adaptation settings are reported separately in Appendix B.

A.1 Transition Reference Learning and Grounding

Transition-reference learning. We initialize the visual encoder from V-JEPA 2. For a demonstration segment, the transition representation and its intermediate prediction are

$$\begin{aligned} Z_t &= F_\phi(E_\nu(I_t), E_\nu(I_{t+H}))\,, \\ \widehat{V}_t(u) &= G_\omega(E_\nu(I_t), Z_t, u)\,, \qquad u \in \mathcal{U}. \end{aligned} \tag{14}$$

Intermediate predictions are matched to the corresponding stop-gradient EMA visual targets, while standardized action chunks provide the neighborhood relations in $\mathcal{L}_{\mathrm{rel}}$. The actions are used only for this relational supervision and are not inputs to the transition encoder. After training, the resulting encoder defines the frozen transition teacher.

Grounding in the Action Expert. At a completed Action-Expert depth ℓ, the shared reader operates on the prefix aggregation

$$\bar{H}_t^{\tau,\ell} = \sum_{j=1}^{\ell} \rho_j^{(\ell)}\, \mathrm{LN}\left(H_t^{\tau,j}\right), \qquad \widehat{Z}_t^{\tau,\ell} = R_\psi\left(\bar{H}_t^{\tau,\ell}\right). \tag{15}$$

The reader preserves the action-token sequence and maps it to the indexed transition-token interface. We first optimize the reader and depth aggregation with the source policy frozen, and then jointly optimize them with the source policy under the action and grounding objectives. The transition teacher remains fixed throughout grounding.

A.2 Source Reference Calibration

Source teacher transitions are summarized by a prototype bank $\mathcal{P} = \{P_k\}_{k=1}^{K}$. For each Action-Expert depth ℓ, flow-time bin b, and source transition mode c, source demonstrations provide reader-to-prototype moments $(\boldsymbol{\mu}_{\ell bc}, \boldsymbol{v}_{\ell bc})$. Given the current distance vector $\mathbf{d}_\ell$, the matched source statistics are

$$\begin{aligned} p_c &= \mathrm{softmax}_c\left(-d_{\ell c}/T_{\ell b}\right), \\ \boldsymbol{\mu}_{\ell b} &= \sum_c p_c \boldsymbol{\mu}_{\ell bc}, \\ \boldsymbol{v}_{\ell b} &= \sum_c p_c \left[\boldsymbol{v}_{\ell bc} + \left(\boldsymbol{\mu}_{\ell bc} - \boldsymbol{\mu}_{\ell b}\right)^{\odot 2}\right]. \end{aligned} \tag{16}$$

These moments instantiate the source calibration in Eq. 10, so that $\mathbf{q}_\ell$ is measured relative to source transition structure matched to the current readout rather than to a single global statistic.

Table 5: **Training configuration of ShiftVLA.** Settings are shared across benchmarks unless otherwise specified.

Setting	Transition Reference	Reader Grounding	Joint Grounding	Target Adaptation
Optimization steps	30k	2k	20k	10k
Global batch size	256 (LIBERO) 16 (RoboTwin/Real)	512	32	32
Optimizer	AdamW	AdamW	AdamW	AdamW
Adam $\boldsymbol{\beta}$	$(0.9, 0.95)$	$(0.9, 0.95)$	$(0.9, 0.95)$	$(0.9, 0.95)$
Weight decay	0.05	10^{-10}	10^{-10}	0.01
Learning rate	visual encoder: 10^{-5} transition modules: 10^{-4}	$10^{-4} \to 10^{-5}$	$10^{-5} \to 10^{-6}$	$10^{-5} \to 10^{-6}$
Warm-up steps	–	100	2k	1k
Loss weight	$\lambda_{\text{rel}} = 0.1$	$\lambda_g = 1.0$	$\lambda_g = 0.01$	$\lambda_{\text{corr}} = 0.01$

A.3 Correspondence-Conditioned Adapters

For each adapted projection, we use

$$\begin{aligned} \boldsymbol{\lambda}_m^0(x) &= \exp\big[\mathrm{RMSNorm}\left(W_{\lambda,m}^\top x\right)\big], \\ \Delta_m(x, \mathbf{q}_\ell) &= \tanh\big[g_m(\mathbf{q}_\ell) \odot \mathrm{RMSNorm}\left(W_{\lambda,m}^\top x\right) + b_m(\mathbf{q}_\ell)\big]. \end{aligned} \tag{17}$$

Here g_m and b_m are lightweight learned mappings. Combined with Eq. 11, this retains the base rank activation while adding a bounded residual conditioned on the transition-reference coordinate. The source-derived scalar γ_ℓ in the main text only scales this residual term.

The readout after block ℓ conditions the adapters in block $\ell+1$. For the first block, where no preceding control readout is available, we use a learned bootstrap conditioned on the current perception–language prefix and flow time. Adapter ranks and trainable scopes are reported with the experimental implementation settings in Appendix B.

A.4 Adaptation and Deployment

During target adaptation, the grounded backbone, transition teacher, reader, depth weights, and source reference remain fixed, while the adaptation modules are optimized. Target actions supervise the native flow-matching objective, and the frozen transition teacher provides the correspondence target on complete demonstration windows. Although the reader is frozen, the correspondence objective remains differentiable through its input and therefore constrains the adapted control representations.

The one-sided rule in Eq. 13 is applied to block-wise adaptation parameter groups, after which

$$g^G = \widetilde{g}_{\mathrm{a}}^G + \lambda_{\mathrm{corr}} g_{\mathrm{c}}^G. \tag{18}$$

At inference, the grounded readouts and source-relative coordinates are recomputed during flow denoising to condition subsequent Action-Expert blocks. The transition teacher and future observations are not required.

A.5 Training Configuration

Table 5 summarizes the optimization settings used in transition-reference learning, source grounding, and target adaptation.

B Experimental Protocols

B.1 Shared Few-Shot Protocol

We define k-shot adaptation at the trajectory level: each task receives k complete target demonstrations for each available target scene condition, irrespective of how many transition windows are

sampled from a trajectory. Demonstrations from different target conditions are pooled to adapt a single policy without perturbation labels or condition-specific routing.

For simulation experiments, all compared methods use the same sampled target support. We use three independently sampled support sets for each shot budget and average performance over the resulting adapted policies. The same fixed adaptation checkpoint is used across compared methods and shot settings. All source-side components of ShiftVLA are learned before target support is introduced and do not count toward the target-shot budget.

B.2 Simulation Benchmarks

LIBERO-Plus. We adapt policies trained on the four standard LIBERO suites using target demonstrations recovered from the released LIBERO-Plus training data for Camera, Light, Background, and Noise. A k-shot support set therefore contains $40 \times 4 \times k$ complete trajectories, jointly used to train a *single shared adapter* across all tasks and adaptation conditions. The same adapted policy is evaluated on the full seven-dimension LIBERO-Plus benchmark. Language, Robot, and Layout receive no adaptation demonstrations. The overall LIBERO-Plus score pools evaluation rollouts across all seven perturbation dimensions.

RoboTwin 2.0. We evaluate Clean (Easy)→Hard adaptation on 20 bimanual manipulation tasks. The source policy is trained jointly with 50 Clean demonstrations per task, while target adaptation uses five or ten complete Hard demonstrations per task. Demonstrations from all 20 tasks are pooled to adapt one shared policy, and target support is disjoint from evaluation. Each adapted policy is evaluated with 100 rollouts per task under the standard Hard setting. Complete task-wise results are provided in Appendix C.2.

B.3 Real-World Tasks and Evaluation

We evaluate real-world scene adaptation on the GALAXEA R1 Lite across five multi-stage manipulation tasks. For each task, we collect 100 demonstrations in the source scene and use five complete demonstrations for each of four target-scene shifts: Layout, Lighting, Tablecloth, and Noise. Each task–scene pair is evaluated over 25 trials under the same task-valid initialization protocol across methods. The task instruction is kept fixed across scene shifts, and no scene-shift identity is provided during adaptation or deployment.

Task descriptions. Each task is evaluated through ordered execution checkpoints that correspond to its major manipulation stages.

- **Banana Handover.** *Instruction: "Use the left arm to pick up the banana, transfer it to the right arm, and place it into the bowl."* The ordered checkpoints are: (1) *Left-Arm Grasp*—the banana is securely grasped by the left arm; (2) +*Handover*—the banana is successfully transferred to the right arm; and (3) +*Place in Bowl*—the banana is placed into the target bowl.
- **Ring Stacking.** *Instruction: "Place the yellow and green rings onto the pole one by one."* The ordered checkpoints are: (1) *Yellow Ring on Pole*—the yellow ring is placed onto the pole; and (2) +*Green Ring on Pole*—the green ring is subsequently placed onto the pole to complete the task.
- **Towel Folding.** *Instruction: "Fold the towel three times."* The ordered checkpoints correspond to the three prescribed folds: (1) *First Fold*, (2) +*Second Fold*, and (3) +*Final Fold*.
- **Steamer Assembly.** *Instruction: "Remove the lid, place the small steamer inside, and replace the lid."* The ordered checkpoints are: (1) *Remove Lid*—the outer lid is removed; (2) +*Insert Inner Steamer*—the small steamer is placed inside; and (3) +*Replace Lid*—the lid is returned to close the steamer.
- **Bridge Building.** *Instruction: "Move the two red cylinders to the front-left and front-right positions, then place the rectangular block across them."* The ordered checkpoints are: (1) *Place Left Support*—the first red cylinder is moved to the designated support position; (2)

Table 6: **Few-shot scaling on the four in-domain LIBERO-Plus shifts.** Success rates (%) under Camera, Light, Background, and Noise. Each shot count denotes the number of complete target demonstrations per task and scene condition. Avg. is rollout-weighted across the four shifts.

Method	Camera	Light	Background	Noise	Avg.
1-shot					
$\pi_{0.5}$	80.9	95.0	95.8	89.4	89.3
FLA (Li et al., 2026b)	80.2	96.6	95.7	90.6	89.8
RETAIN (Yadav et al., 2026)	81.7	96.4	95.4	91.9	90.5
ShiftVLA	84.3	96.3	95.9	94.1	92.0
3-shot					
$\pi_{0.5}$	81.7	96.4	95.7	89.8	90.0
FLA (Li et al., 2026b)	81.7	95.4	95.3	89.3	89.5
RETAIN (Yadav et al., 2026)	83.5	95.8	94.9	92.4	91.0
ShiftVLA	89.8	95.8	95.5	94.7	93.6
5-shot					
$\pi_{0.5}$	82.1	96.3	95.5	90.5	90.2
FLA (Li et al., 2026b)	82.2	96.4	95.6	92.8	91.0
RETAIN (Yadav et al., 2026)	83.7	95.8	94.7	93.2	91.2
ShiftVLA	89.9	96.2	95.8	95.4	94.0
10-shot					
$\pi_{0.5}$	81.5	96.2	95.4	90.1	89.9
FLA (Li et al., 2026b)	81.3	95.7	94.9	89.8	89.5
RETAIN (Yadav et al., 2026)	82.0	95.8	95.0	91.2	90.2
ShiftVLA	88.5	95.7	95.8	93.8	93.0

+*Place Right Support*—the second red cylinder is placed at the opposite support position; and (3) +*Place Beam*—the rectangular block is placed across the two supports to complete the bridge.

Evaluation metrics. For each task–scene pair, let $c_{i,j}^{(n)}$ indicate whether trial n reaches checkpoint j of task i. We report cumulative checkpoint success together with Task Success Rate (TSR) and Cumulative Success Rate (CSR):

$$\begin{aligned} \widetilde{c}_{i,j}^{(n)} &= \prod_{r=1}^{j} c_{i,r}^{(n)}, \quad \mathrm{SR}_{i,j} = \frac{1}{N}\sum_{n=1}^{N} \widetilde{c}_{i,j}^{(n)}, \\ \mathrm{TSR}_i &= \mathrm{SR}_{i,S_i}, \quad \mathrm{CSR}_i = \frac{1}{S_i}\sum_{j=1}^{S_i} \mathrm{SR}_{i,j}, \qquad N = 25. \end{aligned} \tag{19}$$

Overall TSR and CSR are obtained by averaging the corresponding task–scene metrics over all five tasks and four scene shifts.

C Complete Experimental Results

C.1 LIBERO-Plus Few-Shot Scaling

Table 6 reports one-, three-, five-, and ten-shot results on Camera, Light, Background, and Noise, the four LIBERO-Plus shifts used for target adaptation. The finer-grained scaling results characterize how adaptation changes as additional target demonstrations become available, both on average and across individual scene shifts.

Table 7: **Complete few-shot adaptation results on RoboTwin 2.0 Hard.** Task-wise success rates (%) after five- and ten-shot adaptation are reported for all 20 evaluated tasks. The tasks are split into two panels for readability.

Method	Adjust Bottle	Beat Block Hammer	Dump BB	Grab Roller	Move PA	Open L	Open Microwave	Place A2B-L	Place A2B-R	Place Cont. Plate
Five-shot										
$\pi_{0.5}$ (Intelligence et al., 2025)	83	69	75	94	65	70	56	40	43	76
FLA (Li et al., 2026b)	88	65	79	95	68	73	54	41	47	78
Robotic Steering (Mitra et al., 2025)	78	74	72	93	60	73	61	35	47	72
RETAIN (Yadav et al., 2026)	83	71	78	94	67	72	60	42	42	77
ShiftVLA	86	73	80	95	69	75	64	49	49	80
Ten-shot										
$\pi_{0.5}$ (Intelligence et al., 2025)	78	67	81	94	69	73	53	40	42	76
FLA (Li et al., 2026b)	82	68	86	93	71	75	55	41	45	78
Robotic Steering (Mitra et al., 2025)	75	72	78	95	66	77	57	37	45	73
RETAIN (Yadav et al., 2026)	80	65	83	95	71	75	58	42	43	79
ShiftVLA	83	72	85	95	73	77	62	48	49	80

Method	Place Dual Shoes	Place Obj. Basket	Place Phone Stand	Place Shoe	Put Bottles Dustbin	Shake Bottle	Shake Bottle Horiz.	Stack Bowls 3	Stack Bowls 2	Turn Switch
Five-shot										
$\pi_{0.5}$ (Intelligence et al., 2025)	29	44	51	51	42	92	94	56	79	14
FLA (Li et al., 2026b)	25	46	55	48	44	93	95	57	83	12
Robotic Steering (Mitra et al., 2025)	25	39	47	55	38	92	94	52	76	18
RETAIN (Yadav et al., 2026)	33	47	51	55	47	94	93	59	79	15
ShiftVLA	37	53	59	56	50	92	93	67	83	18
Ten-shot										
$\pi_{0.5}$ (Intelligence et al., 2025)	28	48	49	49	48	93	94	61	84	11
FLA (Li et al., 2026b)	30	51	53	52	51	94	94	63	87	10
Robotic Steering (Mitra et al., 2025)	24	43	53	52	44	93	94	57	86	15
RETAIN (Yadav et al., 2026)	28	50	56	49	49	93	95	64	83	13
ShiftVLA	36	57	58	55	57	92	94	72	87	15

C.2 Complete RoboTwin 2.0 Results

Tables 7 report five- and ten-shot success rates for all 20 evaluated tasks, complementing the representative results in Table 2. The complete results provide the corresponding task-level view of performance under the jointly randomized Hard setting.

C.3 Complete Real-World Results

Table 8 reports complete five-shot results on GALAXEA R1 Lite across all five tasks and four target-scene shifts. Each task–scene pair contains 25 trials, with cumulative checkpoint success, TSR, and CSR computed as in Eq. 19. The full table complements the representative task–shift pairs in the main paper with results over all evaluated scene shifts.

Additional real-world visualizations. Figure 7 presents the remaining task–scene shifts not shown in Fig. 4. Together, the main-paper and supplementary visualizations cover all five tasks under the four evaluated scene shifts.

D Additional Analysis of the Control–Transition Correspondence

D.1 Action–Transition Geometry across Scene Shifts

We further examine how the action-related organization learned by the transition reference behaves across scene changes. For Camera, Light, Background, and Noise, demonstrations sharing the same

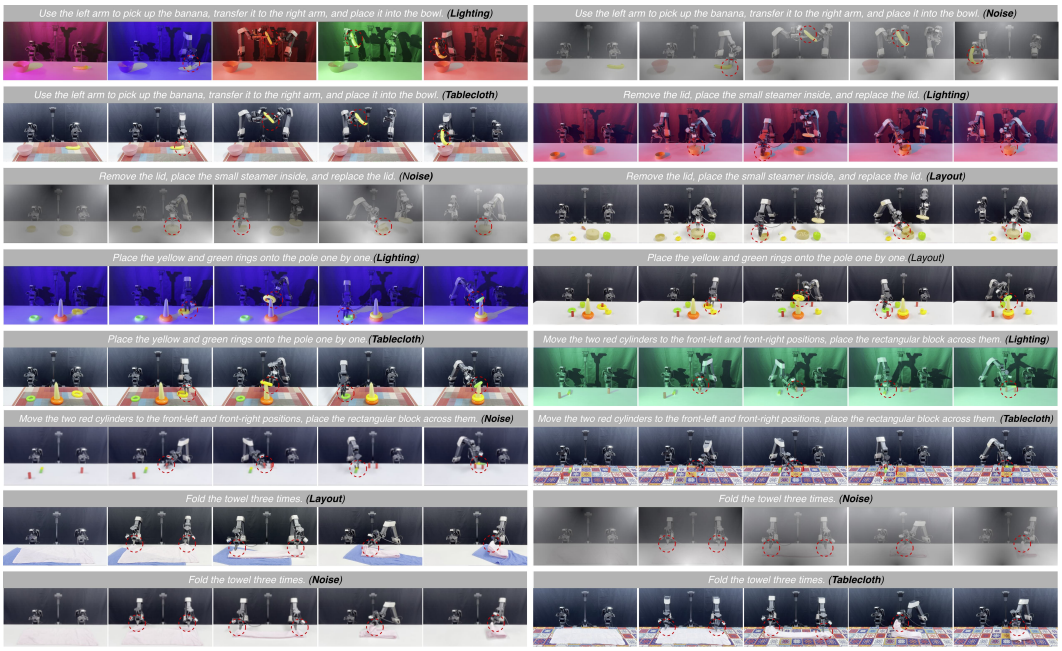


Figure 7: **Additional real-world scene adaptations on GALAXEA R1 Lite.** Target-scene shifts not shown in Fig. 4 are presented with representative execution stages. The noise effect is rendered for illustration.

underlying state–action trajectory are aligned by start index, and consecutive $H{=}10$ windows are extracted with stride one. Each window is represented by pooled tokens from the frozen transition teacher.

For window pairs (i, j), action distance d^a_{ij} is the MSE between standardized action chunks, while transition similarity s^z_{ij} is the cosine similarity between their teacher representations. We summarize their association using

$$\rho_{\text{A-T}} = \rho_{\text{S}} \left(\{-d^a_{ij}\}, \{s^z_{ij}\} \right), \tag{20}$$

where ρ_{S} denotes Spearman rank correlation. The statistic is computed separately for within-condition and cross-condition pairs. For the latter, paired copies of the same underlying execution window are excluded so that identical control segments do not directly contribute to the correlation. The resulting correlations are $\rho_{\text{within}} = 0.5939$ and $\rho_{\text{cross}} = 0.5860$.

Their comparable magnitudes indicate that the action-related organization learned by the transition reference remains similar across the evaluated scene shifts. Since action-neighborhood relations are used during transition-reference learning, the relevant observation is its consistency across scene conditions rather than the existence of action-related structure itself.

For Fig. 6(a), transition representations from all four conditions are embedded jointly once with t-SNE, and the same two-dimensional coordinates are colored by action cluster or scene perturbation. The visualization shows substantial mixing of perturbation labels within the action-related regions and is used only as a qualitative complement to the rank-correlation analysis.

D.2 Control–Transition Agreement across Depth

We compare VLM and Action-Expert representation geometry with action and transition geometry on the same source $H{=}10$ execution windows. Representations are extracted from complete Transformer blocks before and after source grounding. The policy is evaluated at the native first denoising step with pure Gaussian action inputs; ground-truth actions and future observations are used only to construct the comparison references, and no trained reader is applied.

Within each task, we construct pairwise distance matrices for policy representations, demonstrated action chunks, and frozen-teacher transitions. Geometric agreement is defined as

$$\text{Agr}(X, Y) = \rho_{\text{S}} \left(\text{ut}(D_X), \text{ut}(D_Y) \right), \tag{21}$$

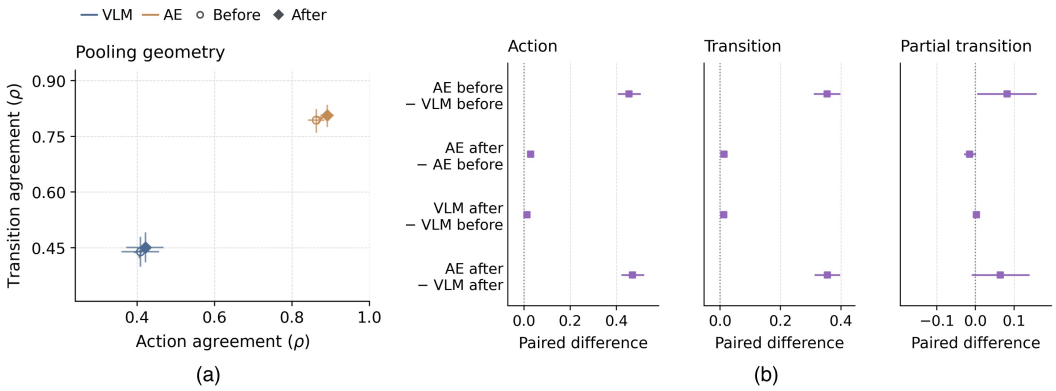


Figure 8: **Control–transition geometry in the source policy.** (a) Final-block agreement of VLM and Action-Expert representations with action and transition geometry. (b) Paired task-level AE–VLM differences, including transition agreement after controlling for action-distance ranks. Intervals denote task-level bootstrap intervals.

where $\mathrm{ut}(\cdot)$ vectorizes the upper-triangular entries. The primary analysis preserves corresponding token positions, matching the indexed transition-token interface used for grounding;

At the final block, Action-Expert representations exhibit higher transition agreement than VLM representations before grounding (0.768 versus 0.498), with the separation remaining after grounding (0.781 versus 0.501). The depth profiles show that the AE–VLM gap becomes more pronounced in later blocks, and action agreement follows the same ordering. Results are first averaged over Gaussian-noise seeds within each task and then equally across tasks; uncertainty in the paired AE–VLM difference is estimated by task-level bootstrap.

We additionally compute partial Spearman correlations after controlling for action-distance ranks. Under this conditioned comparison, we do not observe a consistent additional AE–VLM separation in transition agreement, consistent with the action-related organization explicitly introduced during transition-reference learning. Together, these results support our choice of the Action Expert, rather than the VLM, as the internal control interface for grounding.

E Limitations and Future Work

E.1 Limitations

Our study focuses on few-shot scene adaptation with shared manipulation semantics and action interfaces, while visual, linguistic, or environmental conditions may change. The transition reference provides an adaptation reference for task-relevant execution transitions within this setting. Our evaluation therefore centers on scene shifts that preserve the underlying manipulation semantics and action interfaces; broader transfer across changes in these factors remains outside the scope of the present study. The current adaptation setting also assumes access to a small number of complete target demonstrations, whose actions and future observations provide supervision during adaptation. Future observations, ground-truth actions, and the transition teacher are not required at inference.

E.2 Future Directions

Richer transition references could combine demonstrated state evolution with complementary geometric, contact, or proprioceptive cues and operate across multiple temporal scales. Such extensions would allow the control–transition correspondence to be studied under more complex interaction regimes while preserving the transition reference as a compact reference for adaptation. Reducing the reliance on complete target demonstrations through self-supervised transition consistency, unlabeled target interaction, or lightweight execution feedback could further broaden transition-referenced adaptation to settings with weaker target supervision.

Table 8: **Five-shot real-world scene adaptation on GALAXEA R1 Lite.** Representative task–shift pairs are shown in the main table; average TSR/CSR are computed over all five tasks and four target-scene variations, with 25 trials per pair. Full results are in Appendix C.3.

	Banana Handover *(Layout)*				**Ring Stacking** *(Layout)*			**Towel Folding** *(Layout)*			
Method	Left-Arm Grasp	+Hand-over	+Place in Bowl	CSR	Yellow Ring on Pole	+Green Ring on Pole	CSR	First Fold	+Second Fold	+Final Fold	CSR
OpenVLA-OFT (Kim et al., 2025)	15/25	10/25	7/25	42.7	17/25	12/25	58.0	13/25	10/25	6/25	38.7
$\pi_{0.5}$ (Intelligence et al., 2025)	19/25	15/25	13/25	62.7	21/25	17/25	76.0	18/25	16/25	12/25	61.3
ShiftVLA	20/25	17/25	15/25	69.3	22/25	19/25	82.0	20/25	17/25	14/25	68.0

	Steamer Assembly *(Layout)*				**Bridge Building** *(Layout)*				**Average**	
Method	Remove Lid	+Insert Inner Steamer	+Replace Lid	CSR	Place Left Support	+Place Right Support	+Place Beam	CSR	TSR	CSR
OpenVLA-OFT (Kim et al., 2025)	14/25	10/25	6/25	40.0	15/25	11/25	7/25	44.0	30.4	44.7
$\pi_{0.5}$ (Intelligence et al., 2025)	19/25	16/25	12/25	62.7	20/25	17/25	14/25	68.0	54.4	66.1
ShiftVLA	21/25	18/25	15/25	72.0	21/25	19/25	16/25	74.7	63.2	73.2

	Banana Handover *(Lighting)*				**Ring Stacking** *(Lighting)*			**Towel Folding** *(Lighting)*			
Method	Left-Arm Grasp	+Hand-over	+Place in Bowl	CSR	Yellow Ring on Pole	+Green Ring on Pole	CSR	First Fold	+Second Fold	+Final Fold	CSR
OpenVLA-OFT (Kim et al., 2025)	16/25	12/25	9/25	49.3	19/25	15/25	68.0	14/25	11/25	8/25	44.0
$\pi_{0.5}$ (Intelligence et al., 2025)	20/25	17/25	14/25	68.0	22/25	19/25	82.0	18/25	16/25	15/25	65.3
ShiftVLA	21/25	18/25	16/25	73.3	22/25	20/25	84.0	20/25	18/25	16/25	72.0

	Steamer Assembly *(Lighting)*				**Bridge Building** *(Lighting)*				**Average**	
Method	Remove Lid	+Insert Inner Steamer	+Replace Lid	CSR	Place Left Support	+Place Right Support	+Place Beam	CSR	TSR	CSR
OpenVLA-OFT (Kim et al., 2025)	16/25	13/25	10/25	52.0	17/25	14/25	11/25	56.0	42.4	53.9
$\pi_{0.5}$ (Intelligence et al., 2025)	20/25	18/25	16/25	72.0	21/25	18/25	16/25	73.3	64.0	72.1
ShiftVLA	21/25	19/25	17/25	76.0	22/25	20/25	18/25	80.0	69.6	77.1

	Banana Handover *(Tablecloth)*				**Ring Stacking** *(Tablecloth)*			**Towel Folding** *(Tablecloth)*			
Method	Left-Arm Grasp	+Hand-over	+Place in Bowl	CSR	Yellow Ring on Pole	+Green Ring on Pole	CSR	First Fold	+Second Fold	+Final Fold	CSR
OpenVLA-OFT (Kim et al., 2025)	16/25	12/25	9/25	49.3	19/25	15/25	68.0	15/25	12/25	9/25	48.0
$\pi_{0.5}$ (Intelligence et al., 2025)	20/25	17/25	15/25	69.3	22/25	19/25	82.0	19/25	17/25	15/25	68.0
ShiftVLA	21/25	18/25	16/25	73.3	22/25	20/25	84.0	21/25	18/25	16/25	73.3

	Steamer Assembly *(Tablecloth)*				**Bridge Building** *(Tablecloth)*				**Average**	
Method	Remove Lid	+Insert Inner Steamer	+Replace Lid	CSR	Place Left Support	+Place Right Support	+Place Beam	CSR	TSR	CSR
OpenVLA-OFT (Kim et al., 2025)	15/25	12/25	9/25	48.0	17/25	14/25	11/25	56.0	42.4	53.9
$\pi_{0.5}$ (Intelligence et al., 2025)	20/25	18/25	16/25	72.0	21/25	18/25	16/25	73.3	64.8	72.9
ShiftVLA	20/25	18/25	17/25	73.3	22/25	20/25	18/25	80.0	69.6	76.8

	Banana Handover *(Noise)*				**Ring Stacking** *(Noise)*			**Towel Folding** *(Noise)*			
Method	Left-Arm Grasp	+Hand-over	+Place in Bowl	CSR	Yellow Ring on Pole	+Green Ring on Pole	CSR	First Fold	+Second Fold	+Final Fold	CSR
OpenVLA-OFT (Kim et al., 2025)	14/25	9/25	6/25	38.7	16/25	10/25	52.0	12/25	9/25	6/25	36.0
$\pi_{0.5}$ (Intelligence et al., 2025)	18/25	14/25	12/25	58.7	20/25	15/25	70.0	17/25	14/25	12/25	57.3
ShiftVLA	19/25	16/25	14/25	65.3	21/25	18/25	78.0	18/25	16/25	14/25	64.0

	Steamer Assembly *(Noise)*				**Bridge Building** *(Noise)*				**Average**	
Method	Remove Lid	+Insert Inner Steamer	+Replace Lid	CSR	Place Left Support	+Place Right Support	+Place Beam	CSR	TSR	CSR
OpenVLA-OFT (Kim et al., 2025)	14/25	11/25	8/25	44.0	15/25	12/25	8/25	46.7	30.4	43.5
$\pi_{0.5}$ (Intelligence et al., 2025)	18/25	16/25	14/25	64.0	19/25	16/25	13/25	64.0	52.8	62.8
ShiftVLA	19/25	17/25	16/25	69.3	21/25	18/25	15/25	72.0	61.6	69.7