SPATIAL GRAFTING: GROUNDING 3D FEATURES FOR FLOW-MATCHING ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained robot manipulator policies such as vision-language-action models (VLAs) or world-action models (WAMs) leave interaction-relevant metric geometry implicit. Recent breakthroughs in spatial reconstruction can supply the necessary geometry reliably, but their features only describe local shape without stating where it lies with respect to the robot and the most effective method to deliver this information is unsettled. We propose **SPATIAL GRAFTING**, a versatile, lightweight spatial module, which binds frozen reconstruction features to metric, robot-relative geometry. SPATIAL GRAFTING supplies the flow-matching-based action model with metric-grounded spatial tokens, and grafts the resulting spatial tokens by cross-attention into the flow-matching action expert, without modifying the host's perceptual pathway, so the host retains the full benefit of its pretraining. We evaluate it more broadly than any geometry-aware policy we compare against: one graft architecture, with no per-host redesign, on two VLAs and two WAMs, across four simulation benchmarks that span short-horizon manipulation, visual robustness, clutter and long-horizon mobile manipulation, and on three real-robot platforms with single- and dual-arm configurations. On RoboTwin 2.0, a dual-arm manipulation benchmark, the graft improves every host across VLAs and WAMs, gaining 11.3% and 15.6% on clean and randomized scenes. It reaches 94.0% and 92.4%, above the strongest published 3D-conditioned policy, WAM4D (93.8% and 89.9%). The margin widens as the horizon lengthens: on tasks from BEHAVIOR-1K, a dual-arm mobile manipulation challenge scored by average task progress, it surpasses the 2025 challenge winner on five of six tasks, by up to 0.47 Q-score, and exceeds a map-conditioned spatial policy on average across the three tasks both report.

1 INTRODUCTION

Coupling a pretrained backbone with a flow-matching action expert has become the dominant recipe for robot manipulation, with vision-language-action models (VLAs) building the expert on a vision-language model (VLM) and world-action models (WAMs) on a video world model (Black et al., 2025b; Yuan et al., 2026a). Neither backbone was designed for manipulation or for the 3D structure it depends on: both read a scene as pixels, with no notion of metric scale or of where a surface lies relative to the gripper, so the policy selects the right object and the right sequence of motions yet misplaces the grasp itself. What is missing is not coarse behaviour but contact-level precision. The problem this paper addresses is how to supply a flow-matching policy with manipulation relevant spatial information it can act on.

Spatial reconstruction foundation models such as Depth Anything 3 (DA3) (Lin et al., 2026) and VGGT-Ω (Wang et al., 2026) are the natural source. They recover dense metric structure from ordinary images, and because they are learned networks rather than geometric pipelines, a policy can read their latent features directly. A growing body of spatial action models already adds 3D information to policies, but three gaps remain. First, most are validated on one or two benchmarks and a single host, so it is difficult to attribute a gain to the geometry rather than to the surrounding design. Second, many depend on inputs a deployed robot does not have: privileged instance labels or object state from a simulator, dedicated depth or point-cloud sensors, or a scene map maintained across the episode. Third, each is built for one policy family, often as a new 3D architecture trained

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.

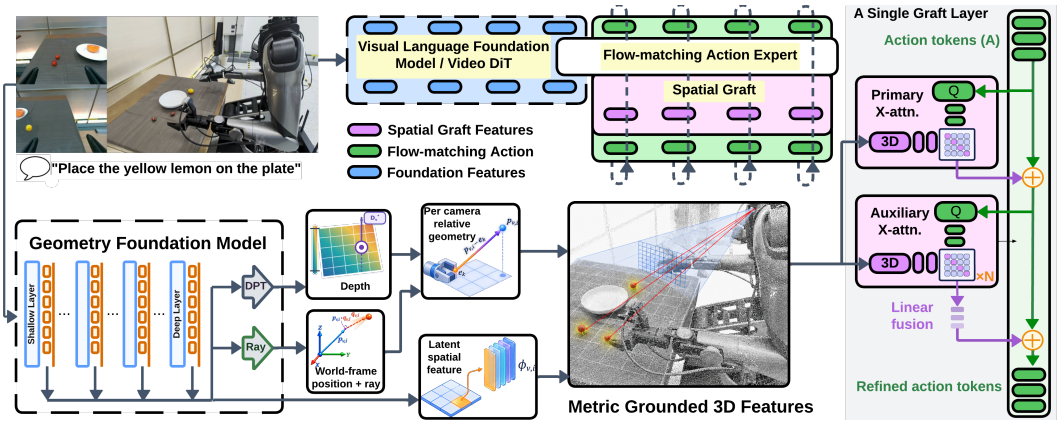

Figure 1: **Overview of SPATIAL GRAFTING.** A frozen geometry foundation model extracts multi-level features from each RGB view. Depth, calibrated rays, and end-effector anchors bind those features to metric, robot-relative 3D locations, forming a spatial bank per view. The magnified graft layer acts only on the flow-matching action expert: action tokens first cross-attend to the primary-view bank; the updated states then query the auxiliary-view banks. A single bias-free linear projection fuses the auxiliary attention outputs into the action update. This layer repeats in selected late action blocks, leaving the host's vision-language or video pathway ungrafted.

from scratch, and does not transfer to the strongest pretrained hosts. The open question is therefore not whether geometry helps, but how to effectively deliver it to flow-matching action policies.

SPATIAL GRAFTING is that interface (Figure 1). Rather than converting reconstruction output into an explicit representation, we ground the reconstruction model's latent space itself: each frozen latent is bound, at its own grid location, to its metric position in the workspace and to its offset from every end effector, and the bound tokens are grafted by cross-attention into the final blocks of the host's action expert. The graft needs RGB, camera calibration and a depth map, measured or predicted by the same frozen model; it uses no segmentation, no object state, and keeps no map or state between observations. The host's perception pathway is untouched, the added parameters are 4–5% of a 3–5B host, and the graft is routed so that it cannot help without reading the geometry. Because the interface asks only for a flow-matching action expert, one construction serves both VLAs and WAMs. We evaluate it more broadly than any spatial action model we compare against (Appendix Table 5): four hosts, two VLAs and two WAMs; four simulation benchmarks that each isolate a different demand—short-horizon manipulation on LIBERO (Liu et al., 2023), visual robustness on RoboTwin 2.0 (Chen et al., 2025), clutter and collision on RoboPRO (Li et al., 2026e), and long-horizon mobile manipulation on BEHAVIOR-1K (B1K) (Stanford Vision and Learning Lab, 2025); and three real robots, single-arm, bimanual and humanoid. Our main contributions are threefold:

- **Robot-grounded reconstruction latents.** Prior work has used reconstruction latents or metric geometric grounding separately. We bind each latent to metric and end-effector-relative coordinates, turning a frozen reconstruction space into a robot-grounded representation for direct action prediction. Bank-off and bank-shuffle ablations show that the gain comes from grounded content rather than added capacity (Section 4.3).
- **A grafting interface for any flow-matching action expert.** The graft uses bank-only cross-attention in the final action-expert blocks, which are revisited at every denoising step, and requires no other host modification. With bias-free projections, zero banks induce zero change (Proposition 1). The same interface improves two VLAs and two WAMs from public checkpoints and remains effective across reconstruction backbones (Sections 4.1 and 4.3).
- **Broad evaluation from unprivileged spatial inputs.** Across four hosts, four benchmarks, and three real robots, we use only RGB, calibration, and measured or predicted depth. Grafted $\pi_{0.5}$ outperforms the strongest published 3D-conditioned policy on RoboTwin, improves all eight RoboPRO host-condition cells, and raises the B1K challenge winner on five of six tasks (Sections 4.1 and 4.2, Tables 1–4).

2 Related Work

Action models and reconstruction models. Modern manipulation policies attach a continuous action expert, trained by diffusion or flow matching (Chi et al., 2023; Black et al., 2025b; Liu et al., 2025), to a pretrained backbone: a VLM in VLAs (Octo Model Team et al., 2024; Black et al., 2025a; Zheng et al., 2026) or a video world model in WAMs (Yuan et al., 2026a; Li et al., 2026c). Both backbones perceive in 2D, so the expert receives neither metric scale nor the position of a surface relative to the gripper, and the residual failures concentrate at contact. Reconstruction foundation models are the complementary resource: they regress dense metric depth and pointmaps from ordinary images (Ranftl et al., 2022; Yang et al., 2024a;b; Wang et al., 2024; Leroy et al., 2024; Wang et al., 2026; Lin et al., 2026), and their latent features encode structure learned far beyond any robot dataset.

Spatial action models. Prior work has supplied geometry to policies as explicit signals: point clouds through a branch into the action module (Li et al., 2026a; Sun et al., 2026), estimated-depth position encodings on the VLM's tokens (Qu et al., 2025), depth tokens (Yuan et al., 2026b; Lee et al., 2026), and persistent scene maps (Kim et al., 2026); earlier 3D policies attended to lifted features relative to the gripper but trained from scratch (Gervet et al., 2023; Ke et al., 2024). These designs pay for their geometry with an RGB-D or point-cloud sensor, with privileged simulator labels and a prebuilt map, or with a bespoke architecture, and each serves one policy family. A second line uses reconstruction *latents* instead, either as a training-time alignment target (Li et al., 2026b; Guo et al., 2025; Li et al., 2026d; Zhang et al., 2026), so that no geometry is present at test time and the gain is bounded by what the host internalizes. Alternatively, the latents are fused into the VLM's visual tokens (Yu et al., 2026; Yang et al., 2026), where the latent arrives in the camera's frame at an unknown scale and is compressed by the trunk before the action expert sees it. Both lines report at most three evaluation settings on a single host (Appendix Table 5), and the two studies that compare injection routes find that where geometry enters decides the outcome (Yu et al., 2026; Yang et al., 2026).

The missing interface. Taken together, the direction has treated *where geometry comes from* (a sensor or a reconstruction model) and *where it enters* (the input, the loss or the action module) as separate questions, and has not delivered reconstruction latents to a pretrained action expert in the frame a controller needs. A latent describes the scene relatively, in the camera's frame and up to scale; a policy must learn a correlation between what it observes and where, in absolute metric units and relative to its end effectors, it should act. Spatial Grafting supplies that interface. It binds each frozen latent, at its own grid location, to its absolute metric position and to its offset from every end effector, so no point cloud, map or privileged label is needed; it delivers the bound tokens through a bank-only route into the final blocks of the action expert, so added capacity cannot help without the geometry; and it asks the host for nothing but a flow-matching expert, so one construction both VLAs and WAMs. We compare against six of the methods above: SERF on B1K, and SpatialVLA, GeoVLA, Spatial Forcing, GLaD and WAM4D on LIBERO and RoboTwin.

3 Method

3.1 Problem formulation

A pretrained policy $\pi_\theta(A \mid I, \mathcal{T}, S)$ maps N RGB views $I = \{I_v\}_{v=1}^N$, an instruction $\mathcal{T}$ and proprioceptive state S to an action chunk A. We consider hosts whose action expert is a transformer trained by flow matching (Lipman et al., 2023; Black et al., 2025b; Yuan et al., 2026a) and re-run once per integration step at inference, whether it reads a VLM trunk or a video stream. Beyond the host's inputs, the graft assumes per-view calibration $\mathcal{C} = \{(K_v, T_v)\}_{v=1}^N$ (intrinsics and camera-to-world pose $T_v = (R_v, t_v)$ in one common frame), a metric depth map $D_v^\star$ measured by the sensor or predicted by the same frozen reconstruction model, and at least one wrist camera per arm; no other input is assumed: no segmentation, object state or map. From these Spatial Grafting builds one bank per view, $B_v = B_v(I_v, \mathcal{C}, D_v^\star)$, and adds graft parameters ψ, leaving the host intact: $A \sim \pi_{\theta,\psi}(A \mid I, \mathcal{T}, S, \{B_v\}_{v=1}^N)$.

3.2 GROUNDING THE LATENT SPACE: THE SPATIAL BANK

Binding happens at the grid: one bank token per feature-grid location, so the correspondence between a latent and its metric point is never broken. The result is a grounded latent space rather than a geometric representation: no point is ever materialized as a cloud or map, and the tokens are rebuilt from the current observation alone. For each view v we tap frozen backbone layers $\mathcal{L}_G$ on a common grid of P locations. Taps differ in scale by orders of magnitude, so each is normalized, projected, rescaled by a learned gain and tagged with a layer embedding before fusion to $\phi_{v,i} \in \mathbb{R}^d$, for d the action-expert width.

Metric and robot-relative geometry. Back-projecting grid location i through (K_v, T_v) and $D_v^\star$ yields a point $p_{v,i}$ and viewing direction $q_{v,i}$ in the common calibration frame. Coordinates are normalized as $\hat{p} = \mathrm{clip}((p-c)/\lambda, -1, 1)$ about a workspace centre c with a single isotropic scale λ per embodiment, so that a 10 cm displacement is encoded identically along every axis. With $e_1, \ldots, e_E$ the end-effector anchors, taken as the optical centres $e_k = t_k$ of the wrist cameras (a fixed rigid offset from the gripper, so no kinematic model is needed), each location carries

$$u_{v,i} = \left[\gamma_B(\hat{p}_{v,i}); \left\{ \gamma_B(\widehat{p_{v,i} - e_k}) \right\}_{k=1}^{E} ; q_{v,i}; m_{v,i} \right], \tag{1}$$

with γ_B a Fourier encoding (Tancik et al., 2020) and $m_{v,i}$ flagging valid depth. The absolute coordinate places the feature in the workspace; the relative ones expose reaching and contact, and change as the arms move even when the scene is static.

Binding. $\phi_{v,i}$ and $u_{v,i}$ describe the same location and become one token: $u_{v,i}$ modulates $\phi_{v,i}$ through FiLM (Perez et al., 2018), and the two are concatenated and projected into the bank token $z_{v,i} \in \mathbb{R}^d$ (Appendix B), so geometry enters both directly and by modulation.

3.3 A BANK-ONLY GRAFT FOR ANY FLOW-MATCHING ACTION EXPERT

Geometry enters only the final M blocks of the action expert, whose states already encode much of the base trajectory; the perceptual pathway receives no graft. Flow matching motivates this placement: the sampler re-enters these blocks at every integration step on a progressively cleaner chunk, so the banks are read over exactly the interval in which a coarse action becomes precise. M spans the last quarter to third of the stack; grafting every block is worse (Section 4.3). The interface touches the host at exactly two points, the action-token states $H^{(l)}$ and the residual update at block l; a new host changes only the expert width d, the number of grafted blocks M and the residual scale ρ, plus a linear bridge when its blocks are wider than the injection (Appendix Table 6).

Each view v has an independent multi-head cross-attention with queries Q from the action states and keys and values from its bank B:

$$\mathrm{CA}_v^{(l)}(Q, B) = W_O^{(l,v)} \,\mathrm{Attn}\Big(\nu\Big(W_Q^{(l,v)} \,\mathrm{LN}(Q)\Big),\ \nu\Big(W_K^{(l,v)} \mathcal{N}(B)\Big),\ W_V^{(l,v)} \mathcal{N}(B) \Big), \tag{2}$$

where $\mathcal{N}$ normalizes the key/value stream, ν normalizes queries and keys per head, and the logits carry a clamped learned gain. The *primary* view $v = 1$, typically the global scene camera, writes into the action stream first, with residual scale ρ: $\bar{H}^{(l)} = H^{(l)} + \rho\, \mathrm{CA}_1^{(l)}(H^{(l)}, B_1)$. Each *auxiliary* view then queries its own bank from this updated state through a learned projection, $A_v^{(l)} = \mathrm{CA}_v^{(l)}(P_v^{(l)} \bar{H}^{(l)}, B_v)$. A single bias-free linear projection $W_{\mathrm{merge}}^{(l)}$ fuses the concatenated auxiliary outputs $[A_2^{(l)}; \ldots; A_N^{(l)}]$ back into the stream, and the result $\tilde{H}^{(l)}$ replaces $H^{(l)}$ in each grafted block (Appendix B.1, Algorithm 1).

Bank-only routing. The projections $P_v^{(l)}$ build queries only, and the merge consumes the attention outputs $A_v^{(l)}$ rather than the branch activations, which would compose two linear layers into a bank-independent action-state map. Zero banks then imply a zero graft delta (Proposition 1); we verify it empirically with bank-off rather than assume it (Appendices B.2 and B.3). The graft adds 147M trainable parameters to $\pi_{0.5}$ and X-VLA, 214M to Fast-WAM and 277M to LingBot-VA, which is 4.4% of the 3.3B $\pi_{0.5}$ and 5.4% of the 5.1B LingBot-VA (Appendix Table 6).

Training objective. Each grafted host keeps its own flow-matching objective and finetuning recipe; the graft only adds the banks to the conditioning set of the velocity field. With noise $\epsilon \sim \mathcal{N}(0, I)$, level $\sigma \in [0, 1]$, interpolant $A_\sigma = (1-\sigma)A + \sigma\epsilon$ and the host's weighting $w(\sigma)$,

$$\mathcal{L}(\theta, \psi) = \mathbb{E}_{A,\epsilon,\sigma}\, w(\sigma) \big\| v_{\theta,\psi}\big(A_\sigma, \sigma \mid I, \mathcal{T}, S, \{B_v\}\big) - (\epsilon - A) \big\|_2^2, \tag{3}$$

where $v_{\theta,\psi}$ is the host's velocity head evaluated with $H^{(l)} \leftarrow \tilde{H}^{(l)}$ in the final M blocks; there is no auxiliary geometric supervision (Appendix C). Three scheduling choices are not neutral and are ablated in Section 4.3: the output projections start at small non-zero values, the host is held fixed for a brief warmup, and the graft residual is dropped jointly across blocks with probability p_{drop}.

4 EXPERIMENTS

Hosts and comparison protocol. We apply SPATIAL GRAFTING to four hosts chosen to span both families of flow-matching policy: two VLAs, $\pi_{0.5}$ and X-VLA, and two WAMs, Fast-WAM and LingBot-VA, and the graft is identical across every host. The reconstruction backbone is a frozen DA3, tapped at four layers on a common feature grid, and the spatial banks enter the final six action-expert blocks of the VLAs and the final ten of the WAMs; the only host-specific settings are the ones listed in Section 3.3.

Benchmarks and baselines. We evaluate on four simulation benchmarks and three real robots, each under its standard protocol and each chosen to stress a different demand. LIBERO (130 tasks, 50 seeds per task) is near saturated and tests whether the graft leaves a working policy intact. RoboTwin 2.0 (50 dual-arm tasks, 20 seeds per task in each of the Clean and Randomized configurations) tests SPATIAL GRAFTING's manipulation capability and robustness to appearance changes. RoboPRO (80 tasks, 20 seeds per task in clean and cluttered scenes, scored under both its Easy and Hard criteria) adds clutter objects in the robot's path and its success metric penalizes collisions. B1K (six tasks, 10 public evaluation instances each) is the most difficult manipulation benchmark we test: long-horizon mobile manipulation that chains sub-skills together within a large household environment, and shows the breadth of SPATIAL GRAFTING's application. The real-robot trials (10 per task) test whether the gains survive measured depth and real calibration error. Success rates (SR) are reported in %, and every reported change is an absolute difference. Beyond each host's own base, we compare against the published 3D-conditioned policies listed in Table 1 and, on B1K, against SERF and the two leading entries of the 2025 challenge. Host configurations, training data, depth sources and the provenance of every quoted number are documented in Appendices C and D, and per-task results in Appendix E.

We ask three questions, one per contribution: does one graft improve both VLAs and WAMs (Section 4.1); do the gains hold under clutter, long horizons and real sensing (Section 4.2); and do they come from the grounded geometry and its placement rather than added parameters (Section 4.3)?

4.1 ONE GRAFT IMPROVES EVERY HOST

RoboTwin: consistent gains, robust to visual change. RoboTwin's paired Clean and Randomized configurations use the same dual-arm tasks, separating task capability from robustness to scene and appearance changes. All four hosts improve, by $+7.5\%$ on Clean and $+8.2\%$ on Randomized on average. The VLAs gain most ($+9.8\%$ to $+15.6\%$), while the WAMs also benefit despite their stronger base scores. Every host also improves on Randomized scenes, by $+1.5\%$ to $+15.6\%$, although the reconstruction backbone is frozen and never sees RoboTwin's randomized clutter, lighting or backgrounds: geometry learned from diverse scenes carries across appearance changes the host has not seen. Qualitatively, the mug column of Figure 2 shows the mechanism: the grafted policy estimates the rack height well enough to clear the hook, where the base approaches at the wrong height. On the other hand, the largest per-task gain for $\pi_{0.5}$ is Move can pot, $51\% \rightarrow 100\%$ Clean and $55\% \rightarrow 95\%$ Randomized (base per-task rates from Li et al. (2026c)): the can must be set down at a position defined relative to the pot, which is exactly the robot-relative metric placement the bank encodes. Blocks ranking size, which needs the blocks' physical sizes, is next ($49\% \rightarrow 85\%$, $26\% \rightarrow 70\%$).

Grafted $\pi_{0.5}$ reaches $94.0\%/92.4\%$, above the best published 3D-conditioned policy, WAM4D ($93.8\%/89.9\%$), and our full-benchmark Spatial Forcing run ($66.2\%/58.9\%$) is below every grafted

Table 1: **LIBERO and RoboTwin SR (%).** Above the line: published 3D-conditioned policies, quoted from their own papers, except Spatial Forcing on RoboTwin, which we trained ourselves (Appendix D). Below the line: our four hosts. LIBERO bases are quoted from each host's published results; the $\pi_{0.5}$ entry is from the openpi release (Physical Intelligence, 2025), as its paper reports no LIBERO result. On RoboTwin, the $\pi_{0.5}$ base is cited from Li et al. (2026c) and the X-VLA base from Bi et al. (2026), both trained on the same RoboTwin demonstrations as our grafts, so the pairs are directly comparable; the Fast-WAM and LingBot-VA bases are our evaluations of their public checkpoints with our seeds and pipeline. Every grafted entry is our run. Green/red: absolute change from the host's own base. NR: not reported in the cited paper.

Model	LIBERO Avg.	RoboTwin Clean	RoboTwin Rand.
GLaD	94.1	NR	NR
GeoVLA	97.7	NR	NR
SpatialVLA	78.1	NR	NR
Spatial Forcing	98.5	66.2	58.9
WAM4D	NR	<u>93.8</u>	89.9
$\pi_{0.5} \rightarrow \pi_{0.5}^{\text{graft}}$	96.9 → **99.6**$^{+2.7}$	82.7 → **94.0**$^{+11.3}$	76.8 → **92.4**$^{+15.6}$
X-VLA → X-VLA^{graft}	98.1 → <u>98.7</u>$^{+0.6}$	72.8 → 83.7$^{+10.9}$	72.8 → 82.6$^{+9.8}$
Fast-WAM → Fast-WAM^{graft}	97.6 → 96.9$^{-0.7}$	82.6 → 86.0$^{+3.4}$	81.8 → 83.3$^{+1.5}$
LingBot-VA → LingBot-VA^{graft}	98.5 → 97.4$^{-1.1}$	88.9 → 93.3$^{+4.4}$	85.7 → <u>91.4</u>$^{+5.7}$

SPATIAL GRAFTING*: base → graft*

Table 2: **B1K Q-score on six tasks.** Q-score is average task progress. $\pi_{0.5\text{-RLC}}$ and $\pi_{0.5\text{-Comet}}$ are the #1 and #2 entries of the 2025 challenge (Stanford Vision and Learning Lab, 2025) and SERF (Kim et al., 2026) is quoted on the three tasks it reports; all three are taken from their published scores. The grafted policy is evaluated on the 10 public evaluation instances of each task. Green/red: change from the host's own base.

Task	**SERF**	$\boldsymbol{\pi_{0.5\text{-Comet}}}$	$\boldsymbol{\pi_{0.5\text{-RLC}} \rightarrow \pi_{0.5\text{-RLC}}^{\text{graft}}}$
Assembling gift baskets	<u>0.525</u>	0.312	0.281 → **0.632**$^{+0.351}$
Collecting children's toys	<u>0.635</u>	0.583	0.500 → **0.673**$^{+0.173}$
Putting shoes on rack	<u>0.601</u>	0.470	**0.720** → 0.590$^{-0.130}$
Putting up Christmas decorations	NR	0.356	<u>0.478</u> → **0.510**$^{+0.032}$
Clean boxing gloves	NR	0.000	<u>0.200</u> → **0.600**$^{+0.400}$
Cleaning up plates and food	NR	0.043	<u>0.243</u> → **0.714**$^{+0.471}$

SPATIAL GRAFTING*: base → graft*

host. These cross-paper rows are not host-matched; the controlled result is that all four base/graft pairs improve. Per-task results for every policy we evaluated ourselves are in Appendix E, Table 8.

LIBERO: maintained performance on a saturated benchmark. Every base already scores 96.9–98.5, so this benchmark can only show whether the graft integrates well into a working policy. From the experiments results, the two VLAs improve ($+2.7\%$ for $\pi_{0.5}$, $+0.6\%$ for X-VLA) and the two WAMs slip by about 1% (-0.7% and -1.1%), which all reside within the saturated band (Table 1). Grafted $\pi_{0.5}$ is the highest entry at 99.6%, but a 1.1% lead on a saturated benchmark is not evidence about geometry; we treat LIBERO as a compatibility check of one frozen reconstruction model across four action experts and draw no conclusion about geometry from it. Per-suite results are in Appendix E, Table 7.

4.2 GAINS HOLD UNDER CLUTTER, LONG HORIZONS AND REAL SENSING

B1K: largest gains where grasping is hardest. Of the spatial action models we compare, only SERF (Kim et al., 2026) reports B1K results, on three tasks. Grafting the challenge-winning $\pi_{0.5\text{-RLC}}$ (Larchenko et al., 2025) increases average task progress (Q-score; Table 2); we evaluate on the 10 public evaluation instances of each task. The graft improves five of the six tasks, by $+0.216$

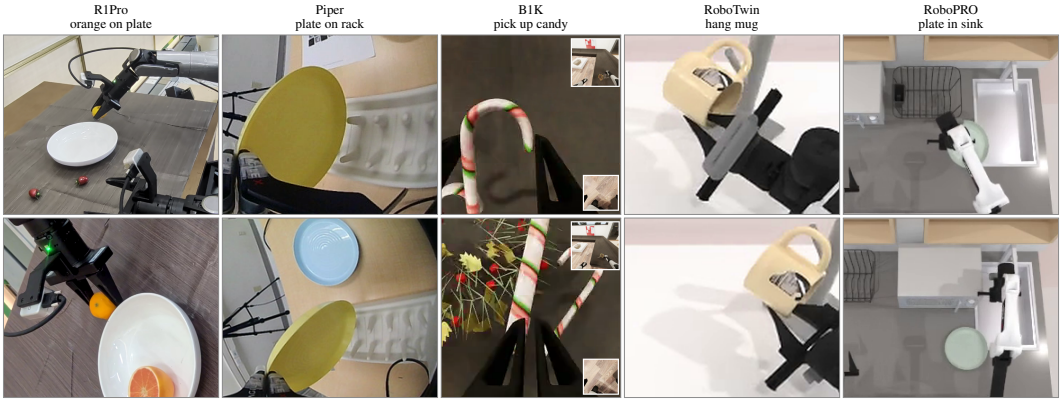

Figure 2: **Grafted success (top) against base failure (bottom), one column per benchmark.** Each column is the task discussed in the corresponding paragraph. R1Pro: the grafted policy estimates plate height and releases the orange with clearance, where the base drops it short. Piper: plate–rack alignment avoids the collision the base drives into. B1K: the base shuts its jaws beside the candy cane's shaft; the grafted policy closes on it. RoboTwin: rack height is estimated well enough to clear the hook. RoboPRO: a lower, better-placed grasp seats the plate in the sink instead of catching its rim.

Table 3: **RoboPRO success (%).** Each cell reads Easy/Hard. Easy is RoboPRO's success rate (SR); Hard is its stricter harsh success rate (HSR), scored on the same episodes. Bases are reported by Li et al. (2026e) and grafted scores are ours. Green: change from the host's own base.

Model	Clean (Easy/Hard)	Clutter (Easy/Hard)
$\pi_{0.5} \rightarrow \pi_{0.5}^{\text{graft}}$	70.3 / 65.7 $\rightarrow$ **78.8 / 73.7**$^{+8.5/+8.0}$	60.9 / 47.8 $\rightarrow$ **69.8 / 49.4**$^{+8.9/+1.6}$
X-VLA $\rightarrow$ X-VLA^{graft}	49.5 / 46.7 $\rightarrow$ 60.9 / 49.9$^{+11.4/+3.2}$	39.7 / 27.6 $\rightarrow$ 45.3 / 35.3$^{+5.6/+7.7}$

SPATIAL GRAFTING*: base $\rightarrow$ graft*

Q-score on average and up to $+0.471$; the exception is putting shoes on rack ($0.720 \rightarrow 0.590$). B1K chains minute-long mobile activities without resetting between contacts, so a small error in one action chunk leaves the robot in a state its demonstrations rarely cover, and the next chunk starts further out of distribution. We attribute the graft's gains at this horizon to the same precision it adds on the tabletop: each contact lands closer to where the demonstrations put it, so there is less error to compound.

SERF (Kim et al., 2026) builds on the same $\pi_{0.5\text{-RLC}}$ checkpoint but addresses a different failure: its persistent map gives a memoryless policy the objects that have left its view (Section 2), whereas the graft targets precision at contact and keeps no state across observations. On the three tasks SERF reports, the grafted policy averages 0.632 against SERF's reported 0.587; the margin is modest because those tasks are dominated by the failure SERF addresses.

The graft's largest gains come where the policy fails at the grasp itself: on the other three tasks it improves its base by $+0.301$ on average. Putting up Christmas decorations requires picking up a thin, smooth candy cane among the wire stems of a wreath; in Figure 2 the base closes beside the shaft while the grafted policy closes on it. Its Q-score gain is small ($0.478 \rightarrow 0.510$) because the base already completes much of the activity. Clean boxing gloves has rounded gloves that can be grasped only at the cuff ($0.200 \rightarrow 0.600$), and cleaning up plates and food requires lifting flat plates and bowls ($0.243 \rightarrow 0.714$). The graft helps most where the grasp is hardest, consistent with its design objective.

RoboPRO: improved collision avoidance in clutter. Both hosts improve in every condition on both Easy and Hard (Table 3). Cluttered scenes place distractors inside the path the demonstration sweeps, and the Hard rate additionally penalizes collisions with them. Hard rises together with Easy in every cell, so the grafted policies complete more tasks without colliding: under clutter, $\pi_{0.5}$ gains

Table 4: **Real-robot evaluation (SR, %).** Matched base/graft comparisons on three platforms, with 10 trials per task. A trial is considered successful only if all sub-tasks are completed. Green: change from the base.

Platform	Task	SR	Trials
UR5e	Pick up marker and place in pen cup (thin objects, constrained placement)	30 → **80** +50	10
Piper	Clean up table (transparent objects, long horizon)	20 → **70** +50	10
	Open tea can (bimanual dexterous manipulation)	10 → **40** +30	10
	Place plate on rack (thin objects, constrained placement)	70 → **90** +20	10
R1Pro	Pick up tangerine and place on plate (thin objects, constrained placement)	50 → **90** +40	10
	Clean up table (deformable object, long horizon)	50 → **80** +30	10

SPATIAL GRAFTING: *base* → *graft*

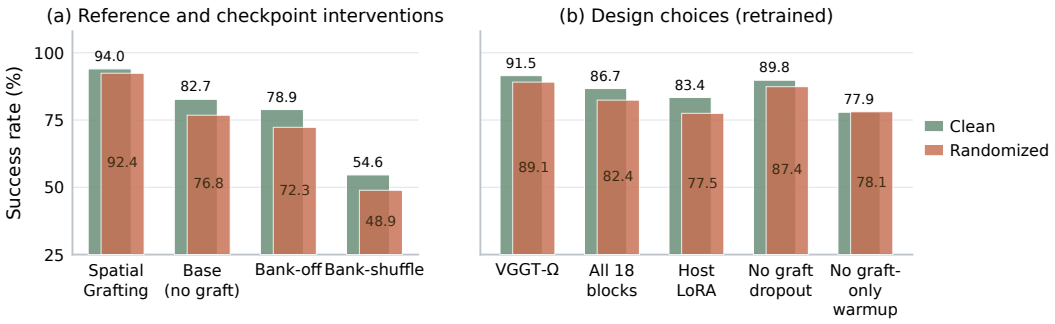

Figure 3: **Ablations on RoboTwin with $\pi_{0.5}$ (SR %).** (a) Reference, ungrafted base, and bank interventions on a fixed checkpoint. (b) One design choice retrained at a time. Equal-width Clean (green) and Randomized (terracotta) bars overlap at each configuration and share a 25% baseline.

+8.9% Easy and +1.6% Hard, and X-VLA gains +5.6% Easy and +7.7% Hard, more than its clean-scene Hard gain of +3.2%. Knowing where each distractor sits relative to the grippers lets the policy route around it; the sink column of Figure 2 shows a grasp low and square enough to seat the plate rather than catch its rim. Per-task results for all 80 tasks are in Appendix E, Table 9.

Real robots: gains transfer to physical sensing. The real-robot trials test whether the gains survive measured depth and real calibration error, on three platforms of different form: a single-arm UR5e, a bimanual pair of AgileX Pipers and the humanoid Galaxea R1Pro. Grafting improves success on all six tasks, by 20% to 50% (Table 4), and the largest gains come where the base fails on depth at contact: marker grasping (30% → 80%, approach at the wrong height), Piper table cleanup (20% → 70%, transparent objects missed and a white mug confused with the background) and R1Pro tangerine placement (50% → 90%, fruit dropped short of the plate). Plate-to-rack placement improves from 70% to 90% through better rack alignment (Figure 2; full-size rollouts in Appendix Figure 4) and deformable-object cleanup from 50% to 80%. Tea-can opening remains hardest (10% → 40%): lid-depth errors are reduced, but the task also demands bimanual coordination that geometry alone does not supply. That the graft carries to hardware without any adaptation of its geometry pathway is expected: DA3 is trained on large and diverse real-world imagery, so its reconstruction is, if anything, better matched to real scenes than to simulated ones, and the grounded latents the graft reads are the same on either side of the simulation boundary.

4.3 THE GAIN COMES FROM THE GROUNDED GEOMETRY AND ITS PLACEMENT

Bank content, not capacity, carries the gain. Figure 3(a) intervenes on a fixed $\pi_{0.5}$ checkpoint on RoboTwin. Zeroing the banks cuts Clean/Randomized by 15.1%/20.1% to 78.9%/72.3%, below the ungrafted base. Substituting another episode's banks cuts a further 24.3%/23.4% to 54.6%/48.9%. Weights and host inputs stay fixed: mismatched bank content hurts more than absent banks, and bank-only routing removes a direct action-state bypass.

The backbone is interchangeable; late injection is essential. The $\pi_{0.5}$ RoboTwin reference uses frozen DA3 features at four levels, the complete spatial grid, and injection into the final six action-expert blocks; host-specific configurations are in Appendix C. Panel (b) retrains one design change at a time from this reference, which also uses dropout and warmup. Replacing DA3 (Lin et al., 2026) with VGGT-Ω (Wang et al., 2026), everything else unchanged, reaches 91.5% clean and 89.1% randomized. This is 2.5% and 3.3% below the reference but 8.8% and 12.3% above the base: the interface transfers across reconstruction backbones, while its ceiling still depends on their quality. Conditioning all 18 blocks instead of the final six costs 7.3% and 10.0%, supporting late injection after the host has formed a task-conditioned action.

Stable adaptation needs warmup and dropout; LoRA is not enough. LoRA gives $83.4\%/77.5\%$, within 0.7% of the base on both configurations, whereas full finetuning delivers the graft. Training without whole-graft dropout ($p_{\text{drop}} = 0$) costs 4.2% and 5.0%, showing the value of periodically acting without the spatial residual. Skipping the graft-only warmup and adapting the host from the first update costs 16.1% and 14.3%, dropping Clean below the base. This is the largest single design effect and helps explain why schedules need to vary with the pretrained host.

Across all variants, the interface's main effect is to close the Clean-minus-Randomized gap (Appendix C).

5 Limitations

Three limitations remain. First, we evaluated only flow-matching action experts; other policies with accessible action tokens and residual updates could use the interface but remain untested. Second, the length of the graft-only warmup depended on how close each host already was to the target data; no single setting was best for all four hosts. Third, errors in the frozen backbone's depth or features limit the available geometry, particularly for high-precision contact.

6 Conclusion

This paper started from a specific failure of pretrained flow-matching policies: VLAs and WAMs select the right object and the right sequence of motions, yet misplace the grasp, because their 2D backbones encode neither metric scale nor where a surface lies relative to the gripper. We argued that the missing ingredient is not geometry as such, which sensors and reconstruction models already supply, but an interface. Prior spatial action models either consume reconstruction latents without metric grounding or supply explicit metric geometry without the latents; each is built for one policy family, validated on few settings, and often dependent on inputs a deployed robot lacks. Spatial Grafting is that interface. It binds each frozen reconstruction latent, at its own grid location, to its absolute metric position and its offset from every end effector, and grafts the bound tokens by bank-only cross-attention into the final blocks of the host's action expert, from RGB, calibration and depth alone.

The experiments answer the three questions this framing poses. One graft improves every host: the same construction, unchanged, raises two VLAs and two WAMs from their public checkpoints on RoboTwin by $+7.5\%$ Clean and $+8.2\%$ Randomized on average, with grafted $\pi_{0.5}$ reaching $94.0/92.4$, above the strongest published 3D-conditioned policy, while leaving saturated LIBERO intact. The gains hold where geometry matters most: all eight RoboPRO host–condition cells improve, including the stricter collision-penalizing criterion under clutter; on B1K the graft raises the challenge-winning policy on five of six tasks, by up to 0.47 Q-score and most on the tasks that fail at the grasp, and exceeds a map-conditioned policy in the mean of the tasks both report without keeping any scene state; and on three real robots every task improves by 20 to 50 points. The gain comes from the grounded geometry and its placement rather than from added parameters: zeroing the banks drops grafted $\pi_{0.5}$ below its own base, substituting another episode's banks drops it a further 24%, injecting into every block is worse than injecting late, and swapping DA3 for VGGT-Ω keeps most of the gain. Together these results support the claim of the introduction: a pretrained action expert can act on reconstruction latents once they are expressed in absolute, robot-relative units, and one such interface serves the current generation of flow-matching policies. Section 5 records what remains open.

AI USE STATEMENT

Large language models were used in preparing this manuscript as writing and editing aids: for language polishing, for revising and condensing prose under the authors' direction, for LaTeX formatting of tables and figures, and for checking citation details against the cited papers. The research questions, problem formulation, method, experimental design, data collection, training and evaluation code, and the analysis and interpretation of all results were conceived and carried out by the authors without LLM assistance. In Figure 4, the Piper tea-can and UR5e marker panels were cropped and enhanced with a generative deblurring tool for visual clarity; because such tools can reconstruct fine detail, these two panels are illustrative and no quantitative result depends on them. The authors reviewed and verified every sentence, number and reference in the paper and take full responsibility for its content. No AI system is an author or contributor.

ETHICS STATEMENT

This work involves no human subjects, personal data or crowd-sourced annotation. All simulation benchmarks, demonstration datasets and host checkpoints are publicly released and were used under their respective licenses. The method produces manipulation policies for physical robots; policies that misjudge depth or object appearance can collide with the environment, as our failure cases show, and the success rates reported here do not establish safety for unsupervised deployment. Deployed policies should be run under human supervision with the ability to stop the robot, and we release no policy intended for use outside controlled laboratory settings. We are not aware of any other ethical concerns raised by this work.

REPRODUCIBILITY STATEMENT

We have taken several measures to make this work reproducible. The main paper specifies the spatial-bank construction and the grafting architecture (Section 3), the training objective (Section 3.3) and the evaluation protocol of every benchmark (Section 4). Appendix B gives the complete grafting equations, the bank-only proposition with its proof and an algorithmic summary; Appendix C records the per-host graft configuration and trainable-parameter counts; Appendix D documents the training data, evaluation seeds and instances, and the provenance of every published number we compare against; and Appendix E reports per-task results for all RoboTwin and RoboPRO tasks. All hosts are grafted from publicly released checkpoints on publicly released datasets. The supplementary material contains our anonymized training code: the spatial-bank builder and graft modules for every host, the benchmark wrappers and training configurations used for RoboTwin and RoboPRO, and the calibration tools that reconstruct the camera extrinsics of the training data, so that every grafted model can be retrained from the public checkpoints and datasets. The BEHAVIOR-1K code is withheld until the ongoing challenge closes.

REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, 2026.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke,

Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025b.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. doi: 10.48550/arXiv.2506.18088.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3D: 3D feature field transformers for multi-task robotic manipulation. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3949–3965. PMLR, 2023.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4d world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*, 2026. doi: 10.48550/arXiv.2604.26694.

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. doi: 10.48550/arXiv.2512.09619.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D diffuser actor: Policy diffusion with 3D scene representations. In *Conference on Robot Learning (CoRL)*, 2024.

Sunghwan Kim, Byeonghyun Pak, Kehan Long, Yulun Tian, and Nikolay Atanasov. SERF: Spatiotemporal environment and robot feature map for long-horizon mobile manipulation. In *Conference on Robot Learning (CoRL)*, 2026. doi: 10.48550/arXiv.2606.12956.

I. Larchenko, G. Zarin, and A. Karnatak. Task adaptation of vision-language-action model: 1st place solution for the 2025 BEHAVIOR challenge. *arXiv preprint arXiv:2512.06951*, 2025. doi: 10.48550/arXiv.2512.06951.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. PointVLA: Injecting the 3D world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3):2506–2513, 2026a. doi: 10.1109/LRA.2026.3653303.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning (CoRL)*, 2023.

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 (ICLR)*, 2026b. doi: 10.48550/arXiv.2510.12276.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems (RSS)*, 2026c. doi: 10.15607/RSS.2026.XXII.016.

Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Peidong Jia, Jian Tang, Sirui Han, and Shanghang Zhang. WAM4D: Fast 4d world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026d. doi: 10.48550/arXiv.2606.14048.

Zhiyuan Li, Charles Eret, Xuan Zhao, Yangzheng Wu, and Amir Rasouli. Data and evaluation protocols for robust robotic manipulation. In *Data-Centric Robotics Workshop at Robotics: Science and Systems (RSS)*, 2026e. URL `https://robopro-bench.github.io/RoboPRO/`.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, 2026.

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 (ICLR)*, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations (ICLR)*, 2025.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. GWM: Towards scalable gaussian world models for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9263–9274, 2025.

Jiangran Lyu, Ziming Li, Xuesong Shi, Chaoyi Xu, Yizhou Wang, and He Wang. DyWA: Dynamics-adaptive world action model for generalizable non-prehensile manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11058–11068, 2025.

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. In *Robotics: Science and Systems (RSS)*, 2024.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Physical Intelligence. openpi: Open-source models and code for robot learning. `https://github.com/Physical-Intelligence/openpi`, 2025. LIBERO results for the `pi05_libero` checkpoint.

Jingjing Qian, Boyao Han, Chen Shi, Lei Xiao, Long Yang, Shaoshuai Shi, and Li Jiang. Geopredict: Leveraging predictive kinematics and 3d gaussian geometry for precise vla manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13529–13539, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems (RSS)*, 2025.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(3):1623–1637, 2022.

Stanford Vision and Learning Lab. BEHAVIOR challenge leaderboard: 2025 results. Official challenge leaderboard, 2025. URL https://behavior.stanford.edu/challenge/archive/2025/leaderboard.html. Accessed: 2026-08-28.

Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. GeoVLA: Empowering 3d representations in vision-language-action models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. doi: 10.48550/arXiv.2508.09071.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Yurou Yang, Muyuan Lin, Roberto Martin-Martin, Martin Labrie, Shreekant Gayaka, Cheng-Hao Kuo, and Luca Carlone. Understanding the impact of geometric foundation models on vision-language-action models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 21–38, 2026.

Bin Yu, Shijie Lian, Xiaopeng Lin, Zhaolong Shen, Yuliang Wei, Haishan Liu, Changti Wu, Hang Yuan, Bailing Wang, Cong Huang, and Kai Chen. 3d-mix for vla: A plug-and-play module for integrating vggt-based 3d information into vision-language-action models. *arXiv preprint arXiv:2603.24393*, 2026. doi: 10.48550/arXiv.2603.24393.

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*, 2026a. doi: 10.48550/arXiv.2603.16666.

Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. DepthVLA: Enhancing vision-language-action models with depth-aware spatial reasoning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2026b.

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4d geometric priors for inference-efficient world action models. *arXiv preprint arXiv:2607.05468*, 2026. doi: 10.48550/arXiv.2607.05468.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Yang Ren, Tao Kong, Junjie Yan, Xianyuan Zhang, Bin Zhao, and Xuelong Li. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026.

A Detailed related-work comparison

Geometry signals and host policies. Table 5 expands the comparison in Section 2 by separating the pretrained hosts from methods that explicitly supply geometric information. The host rows describe the policies to which we attach the graft—$\pi_{0.5}$ (Black et al., 2025a), X-VLA (Zheng et al., 2026), Fast-WAM (Yuan et al., 2026a), and LingBot-VA (Li et al., 2026c)—not competing 3D modules. Among the spatial action models, SpatialVLA (Qu et al., 2025) and GeoVLA (Sun et al., 2026) use depth-derived positions or point clouds, while Spatial Forcing (Li et al., 2026b) and GLaD (Guo et al., 2025) distill frozen reconstruction features during training. SERF (Kim et al., 2026) instead maintains a persistent map of the scene and robot. These are different sources of spatial information: explicit coordinates locate observations, distilled features provide a learned geometric prior, and persistent maps retain information beyond the current view. The "training only" annotation distinguishes supervision that shapes the deployed policy from an additional geometric input supplied at inference.

Where geometry enters. The route column distinguishes methods even when they use similar signals. SpatialVLA modifies visual-token positions, 3D-Mix (Yu et al., 2026) fuses reconstruction and semantic features, and Spatial Forcing and GLaD align intermediate representations. GeoVLA and PointVLA (Li et al., 2026a) deliver point-cloud information to an action expert. GeoPredict (Qian et al., 2026) uses auxiliary geometric prediction, while DyWA (Lyu et al., 2025), GWM (Lu et al., 2025), WAM4D (Li et al., 2026d), X-WAM (Guo et al., 2026), and MECo-WAM (Zhang et al., 2026) use geometric prediction or reconstruction to shape action-relevant representations or world-model rollouts. Our distinction is therefore not simply using 3D features or adding an action-side module. We bind frozen features to metric, end-effector-relative geometry and route the added residual through spatial banks in the late action layers. This separates the source of the features from their robot-relative grounding and from the mechanism by which they modify an action. The bank interventions in Section 4.3 test dependence on that information, rather than attributing every benefit of an added module to geometry.

Evaluation coverage and comparison limits. The final column records the settings reported for each method, not a common-protocol leaderboard. In particular, RoboTwin 1.0 is distinct from RoboTwin 2.0, and a real-robot evaluation can involve different embodiments, tasks, and trial counts. BEHAVIOR-1K coverage is also task-specific: SERF (Kim et al., 2026) reports three tasks, while our results report six individual tasks rather than a benchmark-wide aggregate. The table highlights that we examine the same grafting interface across tabletop simulation, distribution shift, long-horizon mobile manipulation, and physical robots, using both VLA and WAM hosts. It does not imply that every host was tested in every setting or that broader coverage establishes higher performance. Quantitative comparisons and their provenance are given in Section 4 and Appendix D; the controlled comparisons remain each host with and without its graft.

B Method details

B.1 Derivations and full grafting equations

Back-projection. Let $\tilde{x}_{v,i}$ be the homogeneous image coordinate of location i, with normalized camera ray $r_{v,i} = K_v^{-1}\tilde{x}_{v,i}/\|K_v^{-1}\tilde{x}_{v,i}\|_2$. Treating $D^\star_{v,i}$ as optical-axis depth, the world-frame point and viewing direction used in Section 3.2 are

$$p_{v,i} = R_v\left(\frac{D^\star_{v,i}}{r_{v,i,z}}\, r_{v,i}\right) + t_v, \qquad q_{v,i} = R_v r_{v,i}. \tag{4}$$

Injection. Writing $H^{(l)}$ for the action-token states at a grafted block l, $\mathrm{CA}_v^{(l)}$ for the per-view cross-attention of Eq. equation 2, $P_v^{(l)}$ for the view-specific query projections and $\rho > 0$ for the

Table 5: **Host policies and geometry-aware policies.** Rows follow Section 2. Status is the venue as of submission; "Real" denotes real-robot experiments; test-time inputs are what the deployed policy consumes beyond proprioception. The four hosts carry no geometry module. Every geometry-aware method reports at most three evaluation settings; SPATIAL GRAFTING reports five, on four hosts.

Method	Status	Geometry signal	Route into the model	Test-time inputs	Evaluation
Hosts: VLAs and WAMs					
$\pi_{0.5}$	CoRL'25	None	VLM trunk co-trained with a flow-matching action expert	RGB	Real
X-VLA	ICLR'26	None	Embodiment-specific soft prompts into a flow-matching transformer	RGB	LIBERO; SimplerEnv; CALVIN; VLABench; RoboTwin; Real
Fast-WAM	arXiv'26	None	Video and action experts in a shared mixture of transformers; no video generation at inference	RGB	LIBERO; RoboTwin; Real
LingBot-VA	RSS'26	None	Autoregressive video/action tokens in a mixture of transformers	RGB	LIBERO; RoboTwin; Real
Spatial action models compared in Section 4					
SpatialVLA	RSS'25	Egocentric 3D points from estimated depth	Position encoding added to the VLM's visual tokens	RGB (depth estimated)	LIBERO; SimplerEnv; Real
GeoVLA	IROS'26	Point cloud from depth	Learned point encoder; tokens fed with VLM embeddings to a 3D-enhanced action expert	RGB-D sensor	LIBERO; ManiSkill2; Real
Spatial Forcing	ICLR'26	Frozen VGGT features (training only)	Alignment of intermediate visual embeddings; no 3D input at inference	RGB	LIBERO; RoboTwin; Real
GLaD	arXiv'25	Frozen VGGT features (training only)	Distillation into the language model's hidden states over visual tokens	RGB	LIBERO; LIBERO-PRO
WAM4D	arXiv'26	Future depth (training only)	Spatial register tokens and a depth head, removed for deployment	RGB	RoboTwin; Real
SERF	CoRL'26	Persistent neural-point map of scene and robot	Map tokens prepended to $\pi_{0.5}$'s inputs	RGB-D, simulator instance labels, prior map	B1K (3 tasks)
Wider geometry-aware literature					
PointVLA	RA-L'26	Point-cloud features	Additive adapters in the action expert	RGB + point cloud	RoboTwin 1.0; Real
GeoPredict	CVPR'26	3D keypoint tracks and Gaussian geometry (training only)	Auxiliary predictive supervision	RGB	LIBERO; RoboCasa; Real
3D-Mix	arXiv'26	Frozen VGGT features	Gated fusion with the VLM's semantic features	RGB	LIBERO; SimplerEnv
DyWA	ICCV'25	Partial and goal point clouds	Joint action and next-state prediction	Point cloud	Custom simulation; Real
GWM	ICCV'25	Dynamic 3D Gaussians	Action-conditioned Gaussian rollouts feed a separate policy	RGB	Meta-World; RoboCasa; Real
X-WAM	arXiv'26	Future depth	Replicated late diffusion-transformer blocks predict depth	RGB	RoboCasa; RoboTwin; Real
MECo-WAM	arXiv'26	Frozen VGGT 4D targets (training only)	4D expert with decayed read-mask attention, removed for deployment	RGB	LIBERO; RoboTwin; Real
Ours	—	**Frozen multi-layer features + metric geometry**	**Bank-only cross-attention into the late action layers**	**RGB, calibration, depth (measured or predicted)**	**LIBERO; RoboTwin; RoboPRO; B1K; Real**

shared residual scale, the two stages inside the injection module are

$$\bar{H}^{(l)} = H^{(l)} + \rho\ \mathrm{CA}_1^{(l)}\Big(H^{(l)}, B_1\Big), \tag{5}$$

$$A_v^{(l)} = \mathrm{CA}_v^{(l)}\Big(P_v^{(l)}\bar{H}^{(l)},\ B_v\Big), \qquad v = 2, \ldots, N, \tag{6}$$

$$\tilde{H}^{(l)} = \bar{H}^{(l)} + \rho\, W_{\mathrm{merge}}^{(l)}\, \mathrm{Concat}_{v=2}^{N}\, A_v^{(l)}, \tag{7}$$

The merge is a single bias-free linear projection of the attention outputs $A_v^{(l)}$, not the branch activations $P_v^{(l)}\bar{H}^{(l)}$; with two auxiliary wrist views its input has width $2d$, and with one it has width d. The module returns $\Delta^{(l)} = \tilde{H}^{(l)} - H^{(l)}$, and the caller applies $H^{(l)\prime} = H^{(l)} + g_l\kappa\Delta^{(l)}$. Here g_l selects the final M blocks and whole-graft dropout draws one $\kappa \sim \mathrm{Bernoulli}(1 - p_{\mathrm{drop}})$ per sample per step, shared across those blocks; kept samples are not rescaled.

Table 6: **Trainable parameters added by the graft on RoboTwin.** Counts are measured by building the modules. The share is relative to the host's own parameter count.

Host	Grafted blocks	Bank builder	Injection	Bridges	Total (share)
$\pi_{0.5}$	6	46.0M	100.7M	–	146.8M (4.4% of 3.3B)
X-VLA	6	46.0M	100.7M	–	146.8M
$\pi_{0.5}$, VGGT-Ω	6	48.1M	100.7M	–	148.9M
Fast-WAM	10	46.0M	168.0M	–	214.0M
LingBot-VA	10	46.0M	168.0M	62.9M	276.9M (5.4% of 5.09B)

Algorithm 1 SPATIAL GRAFTING: bank construction and grafting at one denoising step. The exact-zero claim of Proposition 1 is conditional on bias-free bank-dependent paths.

Require: RGB images $\{I_v\}_{v=1}^{N}$, calibration $\{(K_v, T_v)\}$, metric depth $\{D_v^\star\}$, end-effector anchors $\{e_k\}_{k=1}^{E}$, host action states $H^{(l)}$, frozen spatial model G, grafted-block count M

Bank construction ▷ once per observation

for $v = 1, \dots, N$ **do**
  $\{F_v^{(l)}\}_{l \in \mathcal{L}_G} \leftarrow G(I_v)$ ▷ frozen multi-layer taps
  $\phi_v \leftarrow W_{\text{fuse}}\,\text{Concat}_l(\text{LN}(P_l(F_v^{(l)})) + \eta_l)$
  $p_v, q_v \leftarrow$ back-project $D_v^\star$ through (K_v, T_v) ▷ Eq. equation 4
  $u_v \leftarrow [\gamma_B(\hat{p}_v); \{\gamma_B(\widehat{p_v - e_k})\}_k; q_v; m_v]$ ▷ Eq. equation 1
  $s_v \leftarrow \text{MLP}(u_v); \quad f'_v \leftarrow (\mathbf{1} + \Gamma(s_v)) \odot \phi_v + \beta(s_v)$ ▷ FiLM binding
  $z_v \leftarrow \text{LN}(W_{\text{out}}[s_v; f'_v]) +$ grid and view embeddings
  $B_v \leftarrow z_v - \text{mean}_{\text{tokens}}(z_v)$ ▷ per-sample centering
end for

Injection into the final M blocks

draw $\kappa \sim \text{Bernoulli}(1 - p_{\text{drop}})$ ▷ one draw shared by every block
for $l = L - M + 1, \dots, L$ **do**
  $\bar{H} \leftarrow H^{(l)} + \rho\ \text{CA}_1^{(l)}(H^{(l)}, B_1)$ ▷ primary view
  $A_v \leftarrow \text{CA}_v^{(l)}(P_v^{(l)}\bar{H}, B_v)$ for $v = 2, \dots, N$ ▷ $P_v^{(l)}$ builds queries only
  $\tilde{H} \leftarrow \bar{H} + \rho\, W_{\text{merge}}^{(l)}\,\text{Concat}_{v \geq 2}\, A_v$ ▷ merges attention *outputs*
  $H^{(l)\prime} \leftarrow H^{(l)} + \kappa\,(\tilde{H} - H^{(l)})$
end for

Parameter cost. Each grafted block holds one primary cross-attention, $N-1$ query projections, $N-1$ auxiliary cross-attentions and one merge, and the bank builder is shared by all of them, so

$$|\psi| \;=\; \underbrace{M\,(6N-2)\,d^2}_{\text{grafted blocks}} \;+\; \underbrace{|\psi_{\text{bank}}|}_{\text{bank builder}} \;+\; |\psi_{\text{bridge}}|, \tag{8}$$

with d the injection width. With $N = 3$ views and $d = 1024$, each grafted block costs $16d^2 = 16.8$M, so the injection grows linearly in the number of grafted blocks M and quadratically in d. The bank builder is 46.0M for every host, and $|\psi_{\text{bridge}}|$ is non-zero only when the host's blocks are wider than the injection width. Table 6 gives the measured counts.

Frame parameters. The workspace centre c and isotropic scale λ define the frame the graft reasons in, so they are fitted once per embodiment on the training split and stored with the checkpoint. The hand-relative encodings use their own fitted centres, distinct from the workspace centre. Where the end-effector anchors are taken as wrist-camera optical centres, each differs from the true end-effector position by a fixed rigid offset, which the geometry encoder absorbs.

B.2 THE BANK-ONLY INVARIANT

Proposition 1 (Bank-only invariant)**.** *If every projection on a bank-dependent path (W_K, W_V, W_O, W_{merge}) is bias-free and the key/value normalization $\mathcal{N}$ is scale-only, then $B_1 = \cdots = B_N = 0$ implies $H^{(l)\prime} = H^{(l)}$ at every grafted block l.*

Proof. Scale-only normalization gives $\mathcal{N}(0) = 0$, so $B_v = 0$ yields zero keys and values. The attention output is a convex combination of zero value vectors, hence 0 for any query, and the bias-free output projection maps it to 0. Thus $\bar{H}^{(l)} = H^{(l)}$, and $A_v^{(l)} = 0$ for $v \geq 2$ regardless of $P_v^{(l)} \bar{H}^{(l)}$, so the bias-free merge projection maps the concatenation of zeros to zero and $H^{(l)\prime} = H^{(l)}$. □

Implementation status. The $\pi_{0.5}$ implementation uses the attention-output merge, eliminating the direct linear path from an auxiliary query projection to the action stream. Setting `spatial_no_bias=True` removes biases from the auxiliary attention output projections and the merge, but the primary attention output projection, key/value projections and key/value LayerNorm still have trainable biases. Their biases initialize at zero, so a zero bank gives a zero delta at initialization; Proposition 1 is not guaranteed for the trained checkpoint. A nonzero bias can emit a scene-independent constant even with a zero bank. We therefore interpret bank-off as a measurement of the trained model with its banks removed, not as an assumed-zero residual.

B.3 Architectural verification

Proposition 1 establishes an exact-zero invariant only under its stated bias and normalization conditions. Before training, we check numerically that a zero spatial bank gives zero graft residual and a nonzero bank gives a nonzero one. This verifies initialization, not that the invariant continues to hold after the remaining biases have trained. For the Fast-WAM instantiation, strict checkpoint loading verifies the frozen host snapshot and every added bank-builder and grafting parameter. The evaluation preflight additionally checks view ordering, RGB and depth preprocessing, calibrated intrinsics and extrinsics, the 32-action prediction and 24-action execution contract, and live construction of a nonzero bank from the current three-view observation. Measured depth is retained at millimeter precision and aligned to the 18×24 DA3 feature grid with nearest-neighbor resampling.

These checks separate initialization and wiring from downstream empirical performance. Bank-off measures the trained graft with its banks removed; bank-shuffle measures whether its gain depends on the banks' sample-specific geometry (Section 4.3).

C Reference graft configurations

This appendix records the graft configuration used for the reported RoboTwin results.

Objective. The loss of Eq. equation 3 is the host's own flow-matching objective with the banks added to its conditioning set; the noise distribution, weighting and action parameterization are the host's own, and there is no auxiliary geometric supervision, so a grafted run differs from its baseline only in the presence of $\{B_v\}$.

$\pi_{0.5}$–RoboTwin. Initialization is the public `pi05_base` checkpoint. The spatial backbone is DA3-GIANT-1.1 in backbone-only mode, frozen, with taps at layers $19, 27, 33, 39$ resampled to an 18×24 grid ($P = 432$) over three calibrated views, ordered as overhead, left wrist, right wrist. Metric depth is ground truth from the simulator, retained at millimeter precision and aligned to the feature grid by nearest-neighbor resampling. The two end-effector anchors are the optical centres of the left and right wrist views, recovered from the same extrinsics used for back-projection. The action-token transformer has $L = 18$ blocks and the graft attaches to the final six blocks 12–17.

Bank and grafting hyperparameters: residual scale $\rho = 1$; output projections and FiLM initialized at standard deviation 10^{-2}; Fourier bands $B = 10$; workspace centre $c = (-0.0222,\ 0.0967,\ 0.7403)$ and isotropic scale $\lambda = 0.5198$, fitted on the training split; grid and view embeddings scaled by 0.25; query/key normalization enabled; attention logit gain initialized at 3 and clamped to 8; whole-graft dropout $p_{\text{drop}} = 0.1$; bank-only routing and bias-free auxiliary attention output and merge projections enabled; per-tap, fused-latent and per-sample bank centering all enabled. Evaluation predicts action chunks under the host's own flow-matching sampler with 30 denoising steps.

Fourier band count. We set B against measurement precision rather than maximum precision. At the RoboTwin workspace scale $B = 10$ resolves roughly 5 mm in the highest band, while depth

is quantized at 1 mm and pooled over a patch covering one to five centimeters of surface, so further bands would encode noise rather than geometry.

X-VLA–RoboTwin. The X-VLA graft uses the same bank builder and the same grafting topology with identical settings—$\rho = 1$, initialization 10^{-2}, $B = 10$, query/key normalization, $p_{\text{drop}} = 0.1$, bank-only routing, bias-free auxiliary attention output and merge projections, and per-tap, fused and per-sample centering—grafted onto the final six of its 24 action blocks. The PyTorch implementation also omits the primary attention output bias; the JAX $\pi_{0.5}$ implementation retains it. The hosts, workspace constants, and camera layouts differ, while the bank construction and two-stage grafting scheme are shared.

Fast-WAM–RoboTwin. Fast-WAM (Yuan et al., 2026a) uses a 30-block ActionDiT with a pretrained Wan2.2 video expert. The graft starts from the released `robotwin_uncond_3cam_384` checkpoint, attaches to blocks 20–29, and uses three calibrated views, four DA3-NESTED taps (layers 19, 27, 33 and 39) on an 18×24 grid, and measured metric depth. Only the spatial bank builder and the ten grafting modules are trained, on the unchanged RoboTwin training split; DA3-NESTED, the video and action experts, the variational autoencoder and the proprio encoder remain frozen. Evaluation predicts 32 absolute-joint actions, executes 24 before replanning, and uses 10 flow-matching steps under the raw protocol.

LingBot-VA–RoboTwin. LingBot-VA (Li et al., 2026c) is an autoregressive video-action model with a dual-stream mixture of transformers. The graft starts from its released RoboTwin post-training checkpoint, retains its autoregressive video-action rollout, and attaches the same bank-only interface to the final ten blocks of its action stream through the linear bridges of Table 6, since its blocks are wider than the injection width; the video-generation pathway is left unchanged.

Clean-minus-Randomized gap under ablation. The ungrafted $\pi_{0.5}$ base has a 5.9% Clean-minus-Randomized gap on RoboTwin; the reference graft narrows it to 1.6%. Bank-off (6.6%) and bank-shuffle (5.7%) restore nearly the base gap, LoRA leaves it at 5.9%, and the retrained variants narrow it partially: 2.4% without dropout, 4.3% with all-block injection, and 2.4% with VGGT-Ω.

D Evaluation protocol and provenance

Matched comparisons. For matched in-house base/graft runs, we hold the host checkpoint, demonstrations, observations, action representation, training recipe, and rollout protocol fixed; the graft supplies the spatial banks. These paired results are comparable within a host, whereas scores from different hosts are not controlled comparisons. Other table entries are published baselines; their provenance is specified below.

Benchmark setup and training data. On LIBERO (Liu et al., 2023), the simulated embodiment is a single-arm Franka Panda. We meta-train on all 130 tasks of the original dataset, with 50 demonstrations per task (6,500 total), and evaluate over 50 seeds per task. On RoboTwin (Chen et al., 2025), the embodiment is a simulated dual-arm Aloha-AgileX. We use all 50 tasks, with 50 clean and 500 randomized trajectories per task (27,500 total), and evaluate our runs over exactly 20 seeds per task and configuration, the same seeds for every model (Table 8). Clean and Randomized share tasks but differ in clutter, lighting, background, table height and instruction phrasing. The VLA hosts start from broad multi-embodiment pretraining checkpoints; the WAM hosts start from their released RoboTwin checkpoints.

For B1K (Li et al., 2023; Stanford Vision and Learning Lab, 2025), we use the challenge's simulated Galaxea R1Pro, a wheeled dual-arm robot, and fine-tune a separate policy for each task from 100 demonstrations. Q-score is the fraction of an activity's goal conditions satisfied, averaged over episodes. We evaluate on the 10 public evaluation instances of each task.

On RoboPRO (Li et al., 2026e), we use the simulated dual-arm Aloha-AgileX and train on the full clean and cluttered demonstration set, approximately 16,000 trajectories across 80 tasks. Evaluation reports the benchmark's Easy and Hard success criteria. Cluttered scenes add distractors within the demonstrated motion path while holding the underlying skills fixed.

Table 7: **Detailed LIBERO SR (%).** Results across the four standard suites. Values for external methods are reported by their respective works; the $\pi_{0.5}$ base is from the openpi release (Physical Intelligence, 2025). Host averages match Table 1; dashes indicate unavailable per-suite results, not zero success. Blocks group other baselines, existing 3D methods, and our hosts. Green/red: change from the host's own base.

Method	Spat.	Obj.	Goal	Long	Avg.
π_0	96.8	98.8	95.8	85.2	94.1
OpenVLA-OFT	97.6	98.4	97.9	94.5	97.1
GLaD	95.0	97.4	94.4	89.4	94.1
GeoVLA	98.4	99.0	96.6	96.6	97.7
Spatial Forcing	99.4	99.6	98.8	96.0	98.5
SpatialVLA	88.2	89.9	78.6	55.5	78.1
$\pi_{0.5}$	98.8 → **99.8** +1.0	98.2 → 99.8 +1.6	98.0 → **99.8** +1.8	92.4 → **99.0** +6.6	96.9 → **99.6** +2.7
X-VLA	98.2 → 99.4 +1.2	98.6 → 99.6 +1.0	97.8 → 98.8 +1.0	97.6 → 97.0 −0.6	98.1 → 98.7 +0.6
Fast-WAM	98.2 → 98.0 −0.2	**100.0** → 98.4 −1.6	97.0 → 97.2 +0.2	95.2 → 94.0 −1.2	97.6 → 96.9 −0.7
LingBot-VA	98.5 → 98.6 +0.1	99.6 → 99.0 −0.6	97.2 → 98.2 +1.0	98.5 → 93.8 −4.7	98.5 → 97.4 −1.1

SPATIAL GRAFTING: *base* → *graft*

Real-robot evaluation uses UR5e, AgileX Piper, and Galaxea R1Pro, covering thin-object grasping, constrained placement, multi-step table cleanup, and bimanual manipulation. Each base/graft condition is evaluated over ten trials per task; success requires completing all subtasks. Table 4 specifies the tasks and platforms.

Result provenance. The main tables mix published numbers with our own runs; we record the source of each value here rather than marking it in place. Every SPATIAL GRAFTING entry is trained and evaluated in-house. *LIBERO*: every non-grafted entry is quoted from published work. The X-VLA, Fast-WAM and LingBot-VA bases are taken from their papers; the $\pi_{0.5}$ base is taken from the openpi release (Physical Intelligence, 2025), since the $\pi_{0.5}$ paper reports no LIBERO result. *RoboTwin*: the WAM4D, $\pi_{0.5}$ and X-VLA bases are published, the latter two taken from Li et al. (2026c) and Bi et al. (2026) respectively; we use these published values because those models were trained on the same 50-task RoboTwin demonstration set as our grafts, so base and graft are directly comparable. The Fast-WAM and LingBot-VA bases are our evaluations of their public checkpoints with our in-house evaluation seeds and pipeline, the same seeds used for every grafted run; and Spatial Forcing is the one reference we trained ourselves: its paper trains on a seven-task subset, and we used its public RoboTwin training code on the full 50-task dataset. *B1K*: the $\pi_{0.5\text{-Comet}}$ and $\pi_{0.5\text{-RLC}}$ scores are those published by the challenge organizers (Stanford Vision and Learning Lab, 2025) and the SERF column is quoted from Kim et al. (2026); our grafted scores are evaluated on the 20 public evaluation instances of each task. *RoboPRO*: every baseline is from Li et al. (2026e).

Published comparators. GLaD, GeoVLA and SpatialVLA do not report RoboTwin, and WAM4D does not report LIBERO; these NR entries reflect what the cited papers evaluate, not what the methods can run. WAM4D's $93.8/89.9$ covers all 50 RoboTwin tasks with 50 clean and 500 randomized demonstrations per task, but its host and training differ from ours, so it is a level rather than a matched comparison. On B1K, SERF reports three tasks, which we quote in Table 2.

E DETAILED BENCHMARK RESULTS

This appendix reports the per-suite and per-task results behind the aggregates of the main text. Every value in these tables that is not marked as quoted from a published work was produced by us with the evaluation code, seed lists and instance lists included in the supplementary material, under the protocols of Appendix D.

LIBERO by suite. Table 7 breaks the LIBERO averages of Table 1 into the four standard suites (Spatial, Object, Goal and Long) for every host, ungrafted and grafted, alongside the published reference policies. Each grafted entry is our evaluation over 50 seeds per task with the supplied evaluation script.



Figure 4: **Success and failure modes across platforms.** Left and right show successful and failed rollouts. For R1Pro fruit placement, improved plate-height estimation leaves clearance above the rim, while the failure approaches too low. For Piper plate placement, better plate–rack alignment avoids collision; the two panels use left- and right-wrist fisheye cameras, respectively. In B1K candy pickup and RoboPRO plate pickup, target depth supports a lower grasp pose. In RoboTwin mug hanging, estimating the rack height positions the mug at the support. The added Piper tea-can opening and UR5e marker-grasp panels are cropped and AI-enhanced for visual clarity; generative deblurring may reconstruct fine details, so these panels are illustrative rather than unaltered experimental frames.

RoboTwin by task. Table 8 lists Clean/Randomized success on each of the 50 RoboTwin tasks for every policy we evaluated ourselves: the four grafted hosts, the ungrafted Fast-WAM and LingBot-VA checkpoints, and the VGGT-Ω variant of grafted $\pi_{0.5}$. All runs use the same 20 seeds per task; the seed bank and the evaluation pipeline that produced these numbers are included in the supplementary material.

RoboPRO by task. Table 9 lists Easy/Hard success for the two grafted hosts on each of the 80 RoboPRO tasks, in clean and cluttered scenes, over 20 seeds per task. The evaluation configuration and seed lists are included in the supplementary material.

Table 8: **Per-task RoboTwin SR (%), Clean/Randomized.** Each cell reads Clean/Randomized over exactly 20 seeds per task; every model and evaluation uses the same seeds. DA3 is the spatial backbone unless marked VGGT-Ω; Fast-WAM and LingBot-VA without a superscript are the ungrafted public checkpoints. The average is the mean of the 50 per-task rates.

Task	$\pi_{0.5}^{\text{graft}}$	X-VLA^{graft}	Fast-WAM	Fast-WAM^{graft}	LingBot-VA	LingBot-VA^{graft}	$\pi_{0.5}^{\text{graft}}$ (VGGT-Ω)
Adjust bottle	100/100	100/95	100/95	100/100	100/100	95/95	100/100
Beat block hammer	100/100	90/85	100/100	100/95	100/100	100/100	100/95
Blocks ranking RGB	100/95	100/85	95/85	100/90	75/90	100/100	100/100
Blocks ranking size	85/70	75/85	60/65	65/85	85/50	95/90	85/75
Click alarmclock	95/100	10/30	100/100	100/100	100/100	100/100	95/100
Click bell	100/100	70/90	100/100	100/100	100/100	100/100	95/100
Dump bin bigbin	95/90	95/90	95/95	95/95	90/100	85/95	80/90
Grab roller	100/100	100/100	100/100	100/100	100/100	100/100	100/100
Handover block	90/95	100/85	60/95	95/85	80/80	100/95	95/95
Handover mic	100/100	95/90	95/85	100/100	65/100	95/95	100/95
Hanging mug	50/45	55/45	15/25	40/35	35/30	30/30	20/15
Lift pot	100/100	100/100	95/95	100/90	100/100	100/100	95/90
Move can pot	100/95	100/85	80/75	95/70	85/100	90/100	100/95
Move pillbottle pad	100/100	95/100	100/100	100/95	95/95	100/100	100/100
Move playingcard away	100/100	100/100	95/100	100/100	100/95	100/100	100/100
Move stapler pad	90/85	70/75	65/70	65/75	75/70	55/65	80/70
Open laptop	100/100	90/95	95/100	90/100	100/100	100/95	95/95
Open microwave	70/70	55/10	55/75	80/50	90/75	85/80	75/80
Pick diverse bottles	95/100	50/10	85/95	100/100	80/90	90/90	95/80
Pick dual bottles	100/100	100/100	95/95	100/95	100/100	100/100	100/100
Place a2b left	95/100	85/80	75/35	45/55	100/75	100/100	100/100
Place a2b right	95/100	70/75	70/30	40/55	95/60	100/95	100/100
Place bread basket	100/100	95/90	90/85	95/100	100/95	100/100	95/90
Place bread skillet	80/85	90/85	85/85	95/80	100/90	90/90	80/95
Place burger fries	95/95	100/95	95/100	90/95	90/95	100/85	100/95
Place can basket	85/75	80/75	70/85	80/70	80/90	85/80	90/75
Place cans plasticbox	100/100	100/100	100/100	100/100	100/100	100/100	100/100
Place container plate	100/100	100/100	95/90	100/100	100/100	100/95	100/100
Place dual shoes	100/80	80/80	90/95	85/80	65/90	100/90	90/80
Place empty cup	100/100	80/90	70/75	95/55	100/65	100/100	100/100
Place fan	100/95	75/90	90/100	80/100	90/95	100/95	100/95
Place mouse pad	80/95	70/75	75/80	75/80	75/75	100/85	80/70
Place object basket	85/95	90/95	55/40	50/55	85/50	90/90	85/70
Place object scale	100/95	85/90	90/40	60/75	90/90	95/95	90/100
Place object stand	100/100	90/100	90/75	90/90	100/85	100/95	100/100
Place phone stand	100/90	95/85	90/90	95/90	100/90	100/95	90/90
Place shoe	100/100	100/100	85/95	95/100	90/95	100/100	95/95
Press stapler	90/95	100/95	95/95	95/95	95/90	80/90	85/95
Put bottles dustbin	100/85	95/95	80/95	95/85	85/100	90/75	100/90
Put object cabinet	80/75	75/60	60/0	55/25	85/60	85/80	95/70
Rotate QR code	90/100	100/90	95/90	90/90	85/100	95/90	100/90
Scan object	80/85	90/80	5/10	40/0	70/5	100/95	90/80
Shake bottle	100/100	100/100	100/100	100/100	100/100	100/100	100/100
Shake bottle horizontally	100/100	100/100	100/100	100/100	100/100	100/100	100/100
Stack blocks three	95/90	25/35	100/100	80/100	90/100	100/80	55/75
Stack blocks two	100/85	95/85	100/100	100/95	100/100	100/100	95/95
Stack bowls three	95/90	65/75	70/95	85/85	60/85	90/90	80/70
Stack bowls two	100/100	85/95	95/100	95/95	100/95	100/100	100/100
Stamp seal	100/85	65/70	80/95	100/80	80/80	90/90	90/85
Turn switch	85/80	55/90	50/60	70/75	80/55	55/60	80/75
Average	94.0/92.4	83.7/82.6	82.6/81.8	86.0/83.3	88.9/85.7	93.3/91.4	91.5/89.1

Table 9: **Per-task RoboPRO success (%), Easy/Hard.** Each cell reads Easy (SR)/Hard (HSR) over exactly 20 seeds per task for all 80 tasks in each configuration, the same seeds for every model; averages are the mean of the 80 per-task rates and match Table 3.

	$\pi_{0.5}^{\text{graft}}$		X-VLA^{graft}	
Task	Clean	Clutter	Clean	Clutter
Chain apple bin bowl rack spoon sink	15/15	10/10	0/0	0/0
Chain apple sink plate bread board	95/85	100/90	0/0	0/0
Chain bowl rack apple sink	0/0	0/0	0/0	0/0
Chain heat hamburger	45/45	85/75	20/20	0/0
Chain serve hamburger	45/45	0/0	0/0	0/0
Close drawer	100/100	100/100	0/0	0/0
Close microwave	100/100	100/40	60/60	35/35
Drop apple in bin	100/95	80/80	80/20	55/0
Empty box	95/95	100/95	80/80	30/30
Move book onto table	95/85	100/70	100/100	95/95
Move bottle	45/45	65/55	40/40	75/70
Move bottle from fridge next to can	100/100	80/80	60/0	55/0
Move can from cabinet to basket	60/60	65/60	80/80	5/5
Move cup	70/70	35/25	20/20	40/40
Move cup next to book	100/100	80/80	100/100	45/15
Move cup onto table	95/95	60/45	60/60	75/75
Move cup put pen in cup	70/70	35/10	100/100	20/15
Move cups into box	95/95	70/40	60/60	20/20
Move hamburger onto plate	100/100	90/90	0/0	40/40
Move items around	20/20	0/0	0/0	0/0
Move milk close fridge	90/90	75/45	40/0	0/0
Move pen to box	95/95	60/0	80/80	90/0
Move seal cup next to box	90/90	100/90	80/80	10/10
Move seal next to box	100/100	100/100	100/100	95/95
Move seal next to pencup	90/90	90/90	100/100	10/10
Move seal onto book	95/95	100/75	100/100	15/15
Move seal onto table	100/100	100/40	100/100	90/90
Open drawer	100/100	100/95	60/60	60/60
Organize table	20/20	5/5	0/0	0/0
Pick apple from bowl	100/10	90/35	40/0	15/0
Pick apple from sink	95/95	100/80	0/0	0/0
Pick bottle from fridge	100/100	95/35	100/100	100/100
Pick boxdrink from basket	5/5	0/0	100/60	45/0
Pick can from basket	90/5	100/0	100/0	95/45
Pick can from cabinet	100/100	100/70	60/40	80/80
Pick fork from sink	100/100	100/95	0/0	0/0
Pick hamburger from microwave	50/50	45/15	0/0	25/25
Pick milk box from fridge	100/100	95/90	100/100	90/90
Pick sauce can from cabinet	100/100	100/5	100/100	100/100
Place bowl in dishrack	85/85	100/100	100/100	30/30
Put book in fileholder	50/50	5/5	100/60	30/15
Put book on book	85/85	95/95	40/20	60/60
Put bottle in basket	100/100	50/25	80/80	90/90
Put bottle in fridge	90/90	30/0	100/100	95/95
Put bowl in sink	0/0	0/0	60/60	10/10
Put bread on board	100/95	85/60	80/0	60/0
Put can close cabinet	95/90	100/20	80/80	0/0
Put can in cabinet	85/85	100/10	80/60	95/95
Put can infront of microwave	85/85	100/70	100/100	95/95
Put can next to basket	40/25	95/95	100/100	65/65
Put cup in box	100/100	90/90	100/100	95/95
Put cup on coaster	70/70	0/0	100/100	80/80
Put cup on table	80/80	65/60	80/80	45/45
Put glue in box	100/50	95/50	100/100	80/70
Put hamburger in microwave	100/95	70/55	0/0	0/0
Put milk box in fridge	90/90	90/40	100/100	90/90
Put milktea next to laptop	65/65	65/40	80/60	60/30
Put milktea on shelf	65/65	40/25	80/80	75/40
Put mouse next to stapler	90/90	95/80	100/0	70/0
Put mouse on pad	85/85	30/30	80/80	45/5
Put pen in box	100/100	95/50	100/100	35/35
Put pen in pencup	20/15	45/20	0/0	0/0
Put phone next to cube	90/90	100/50	60/0	45/30
Put phone on holder	70/70	70/65	60/60	55/55
Put plate in sink	70/70	50/50	60/50	45/35
Put rubikscube in drawer	95/95	90/85	80/80	50/45
Put rubikscube next to milktea	60/50	20/20	80/20	30/0
Put sauce can in basket	95/75	100/65	100/100	95/55
Put sauce can in cabinet	100/100	90/5	100/100	85/70
Put seal in box	100/100	100/100	100/100	100/100
Put spoon in dishrack	90/90	95/95	20/20	30/30
Put spoon in sink	100/100	100/100	0/0	35/35
Put spoon on plate	50/50	50/50	40/0	70/0
Put stapler in drawer	95/95	95/50	50/40	80/80
Put stapler next to mouse	90/85	20/0	0/0	10/0
Put stapler on book	100/95	100/100	100/100	80/80
Set up table	40/35	5/0	0/0	0/0
Store rubikscube on shelf	75/75	60/40	0/0	0/0
Store stapler in drawer	60/60	55/35	0/0	0/0
Switch can with bottle in basket	75/0	35/10	60/0	0/0
Average	78.8/73.7	69.8/49.4	60.9/49.9	45.3/35.3