GIVE THE POLICY ITS VIEW BACK: RECOVERING FROZEN VISUOMOTOR POLICIES FROM CAMERA DRIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Visuomotor policies can fail under small camera drift even when the task and action space remain unchanged. We study how to recover such policies without updating their weights. We introduce **ReView**, a test-time input calibration method that fits a six-parameter image warp from stored demonstration images and only four unlabeled reset images from the displaced camera. ReView matches early spatial features across the two unpaired image sets, requiring no camera poses, depth, target actions, or interaction rollouts. The fitted warp corrects incoming images while the policy remains frozen. Across six tasks in robomimic and ManiSkill, ReView improves mean success over no adaptation by 28, 22, and 24 percentage points for Diffusion Policy, ACT, and SmolVLA, respectively, and by 27 and 30 percentage points on two real-robot tasks. Ablations show that spatial correspondence is critical for recovery. Our analysis further shows that geometric alignment and policy recovery need not coincide: under parallax, an imperfect scene alignment can better preserve the visual relationships that matter for control. Recovering a policy therefore requires a view it can act on, not necessarily an exact reconstruction of its original view.

1 INTRODUCTION

Imitation learning has made substantial progress in robotic manipulation, from task-specific visuomotor policies (Chi et al., 2023; Zhao et al., 2023) to large vision–language–action models (Kim et al., 2025a; Black et al., 2025a). Yet these policies are often deployed under the assumption that an external camera remains at the viewpoint used during training or fine-tuning, which we call the *nominal view*. Even a small camera drift can substantially alter the visual observation while leaving the task and action space unchanged. A robot may still know how to grasp an object after its camera is bumped, yet fail to determine where to reach because the robot and object now appear at unfamiliar image locations. Such viewpoint changes can sharply degrade policies trained from demonstrations (Jiang et al., 2026; Fei et al., 2026).

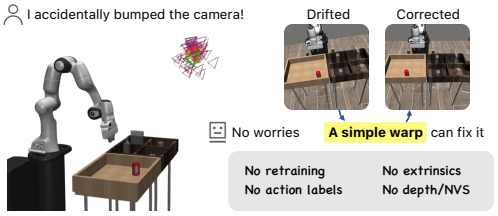

Figure 1: ReView corrects camera drift with an image warp, without policy retraining, depth, camera extrinsics, or view synthesis.

Existing approaches improve viewpoint robustness mainly by exposing the policy to more viewpoints during training or providing explicit geometric information. Viewpoint augmentation expands the training distribution (Chen et al., 2025); 3D observations and camera-conditioned representations make spatial information available to the policy (Ze et al., 2024; Jiang et al., 2026). Camera pose can also be inferred from RGB after pose-supervised training (Li et al., 2026b). For an existing policy, however, these approaches generally require revisiting the training procedure or observation interface, even when the task itself is unchanged.

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.

A complementary approach is to correct the observation at deployment while keeping the policy fixed. However, existing methods typically require additional geometric information or supervision from the displaced camera. Novel view synthesis reconstructs the nominal view using known camera parameters and a learned synthesis model, as in AnyCamVLA (Heo et al., 2026); GS-VLA additionally relies on aligned metric depth (Park & Kim, 2026). Other approaches require target-domain transitions or demonstrations for adaptation (Yang et al., 2023; Li et al., 2026a). These signals may not be available after a camera is bumped or remounted. In that setting, we may have only the frozen policy, stored RGB demonstrations from the nominal view, and a few unlabeled reset images from the displaced camera. Can the original skill be recovered from these existing assets alone? This is the first problem we address.

A second challenge is what to align. Activation matching provides a supervision signal without camera poses (Mirza et al., 2023), but source and target images are unpaired and may differ in object and robot configurations as well as viewpoint. A matching objective can therefore confuse scene variation with camera motion. Moreover, the best image alignment need not be best for the policy: under parallax, aligning the table can still leave the gripper displaced. A useful correction must preserve spatial structure while tolerating errors that do not matter to the policy.

We introduce **ReView** (*Restoring the policy's View*), which recovers a frozen policy by fitting a six-parameter image warp. Using stored demonstration frames and only four unlabeled reset images, ReView requires no depth, camera extrinsics, target actions, interaction rollouts, or view-synthesis model (Figure 1). It matches mean spatial features across unpaired images, then reuses the fitted warp during deployment (Figure 2).

Our contributions are threefold:

- **Deployment-Time Camera Calibration:** We introduce **ReView**, which recovers frozen visuomotor policies from camera drift using stored demonstrations and only four unlabeled reset images, without camera poses, depth, target actions, or interaction rollouts.
- **Analysis of Alignment and Policy Sensitivity:** We show that spatial correspondence is critical for identifying useful corrections, and explain why under parallax, better image alignment does not always lead to better policy performance.
- **Broad Empirical Validation:** We evaluate ReView across Diffusion Policy, ACT, and SmolVLA on six simulation tasks and two real-robot tasks, demonstrating consistent recovery across policy families and platforms.

2 Related work

Visuomotor policies Learning from demonstrations has enabled effective visuomotor control (Mandlekar et al., 2022). Task-specific policies such as Diffusion Policy and ACT model action sequences with diffusion and transformer-based architectures, respectively (Chi et al., 2023; Zhao et al., 2023). More recent generalist and vision–language–action (VLA) policies leverage large-scale robot data and pretrained vision–language representations (Ghosh et al., 2024; Zitkovich et al., 2023; Kim et al., 2025b; Black et al., 2025b;a; Kim et al., 2025a; Shukor et al., 2025). These advances focus primarily on policy learning and action prediction while robustness to changes in camera viewpoint poses a distinct challenge.

Viewpoint robustness Domain randomization, viewpoint augmentation, and multi-view learning improve robustness by broadening the visual distribution seen during training (Tobin et al., 2017; Chen et al., 2025; Seo et al., 2023). A complementary line of work makes geometry explicit through 3D representations, camera conditioning, or joint pose and action prediction (Ze et al., 2024; Zhang et al., 2026; Jiang et al., 2026; Li et al., 2026b). Related approaches use action consistency across paired views (Huang et al., 2026) or robot-centric pointmaps constructed from depth and calibrated cameras (Lee et al., 2026; Yang et al., 2026). At deployment, DREAM and EasyHeC estimate camera-to-robot geometry (Lee et al., 2020; Chen et al., 2023), while AnyCamVLA and GS-VLA reconstruct a canonical view for a frozen policy through novel view synthesis (Heo et al., 2026; Park & Kim, 2026). Review complements these approaches by correcting the input of an already-trained policy without retraining, camera extrinsics, depth, or a view-synthesis model.

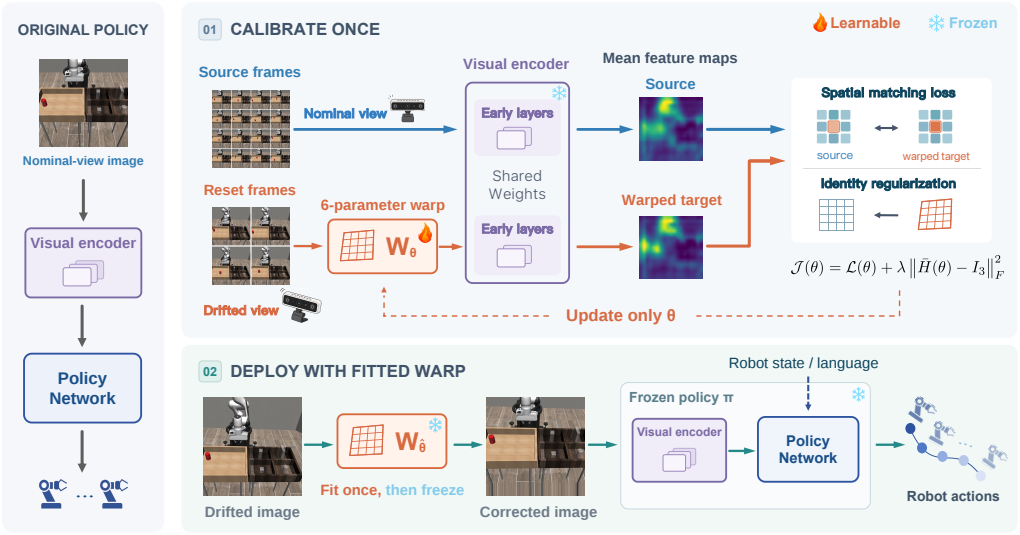

Figure 2: ReView overview. **Left: original policy.** The visual encoder and policy network map nominal-view images to actions. **Top: calibration.** Nominal demonstration images and warped reset images pass through shared, frozen early encoder layers. Their feature maps are averaged over images and matched at corresponding spatial locations. Only the six warp parameters θ are optimized, with identity regularization; the image sets need not be paired. **Bottom: deployment.** The fixed warp corrects images for the frozen policy; state and language inputs are unchanged.

Test-time adaptation Test-time adaptation methods update models at deployment using objectives such as self-supervision, entropy minimization, or activation matching (Sun et al., 2020; Wang et al., 2021; Zhang et al., 2022; Mirza et al., 2023). Other approaches adapt normalization statistics or visual prompts (Li et al., 2018; Sun et al., 2023). In control, PAD and ILA use target interactions to adapt representations (Hansen et al., 2021; Yoneda et al., 2022), while MoVie adapts spatial transformers and the visual encoder using target transitions and a frozen dynamics model (Jaderberg et al., 2015; Yang et al., 2023). Target demonstrations can also support in-context adaptation (Wang & Hager, 2024) or one-shot VLA fine-tuning (Li et al., 2026a). Our method also adapts at deployment, but updates only a geometric input transform before execution and keeps the policy fixed thereafter.

Image alignment Lucas–Kanade (Baker & Matthews, 2004) and enhanced correlation coefficient (ECC) maximization (Evangelidis & Psarakis, 2008) align image intensities. The scale-invariant feature transform (SIFT) (Lowe, 2004) instead supplies local matches for robust transform estimation with random sample consensus (RANSAC) (Fischler & Bolles, 1981). ReView fits the warp from unpaired spatial features of the frozen policy encoder, with alignment serving policy recovery rather than image reconstruction.

3 ReView: Reference-Based Input Alignment

3.1 Policy and Calibration Images

At step t, a visuomotor policy π_ψ maps visual input $\mathbf{I}_t$ (a single RGB image or a short history) and any nonvisual inputs p_t, such as robot state or a task instruction, to an action or action sequence $\mathbf{a}_t$. Its parameters ψ cover the visual encoder and policy network. ReView corrects its images:

$$\mathbf{a}_t = \pi_\psi\big(W_{\hat{\theta}}(\mathbf{I}_t^{\mathrm{d}}), p_t\big),$$

Here d denotes the displaced view. Only the six warp parameters θ are fitted; $W_{\hat{\theta}}$ is shared across frames, with policy parameters and nonvisual inputs fixed.

The source images $\mathcal{S} = \{S_i\}_{i=1}^{N_s}$ are initial frames from $N_s = 16$ training demonstrations at the nominal camera. The target images $\mathcal{T} = \{T_j\}_{j=1}^{N_t}$ are frames from $N_t = 4$ task resets after the

camera moves. These initial-task images need not share object or robot configurations. Calibration uses RGB alone, without target actions, rewards, or depth.

3.2 A SHARED GEOMETRIC CORRECTION

All images from the displaced camera share a warp parameterized by $\theta = (\omega, u) \in \mathbb{R}^6$. The rotation vector $\omega \in \mathbb{R}^3$ gives $R(\omega) = \exp([\omega]_\times)$, where $[\omega]_\times$ is the skew-symmetric matrix satisfying $[\omega]_\times x = \omega \times x$. The vector $u \in \mathbb{R}^3$ represents translation normalized by the reference plane's distance. With fixed plane normal $n = (0, 0, 1)^\top$ and normalized intrinsics $K = \mathrm{diag}(f, f, 1)$, the homography and image warp are

$$H(\theta) = K\big(R(\omega) + un^\top\big)K^{-1}, \tag{1}$$

$$W_\theta(I)(q) = I\big(\mathrm{proj}(H(\theta)q)\big). \tag{2}$$

Here $q = (x, y, 1)^\top$ is a normalized homogeneous output coordinate and $\mathrm{proj}(v) = (v_1/v_3, v_2/v_3)$ converts to image coordinates; $I(\cdot)$ uses bilinear interpolation. The warp captures camera rotation and the translation-induced motion of a plane. The shared correction balances image regions when depth variation prevents exact alignment (Section 4).

The focal length f comes from the nominal camera's field of view, adjusted for the policy's crop. We sample from the full sensor frame to retain surrounding pixels. No camera extrinsics are estimated. Appendices A and C.4 give sampling details and the local effect of focal-length error.

3.3 SPATIAL FEATURE MATCHING

Because source and target images are unpaired, we match features averaged within each set rather than individual image pairs. We extract an early feature map $F(I) \in \mathbb{R}^{C\times V}$ from the policy's visual encoder: convolutional features or transformer patch embeddings. Indices c and v select one of C channels and V spatial locations. From source images, we compute a channel standard deviation s_c and a normalized mean feature map M:

$$s_c = \mathrm{Std}_{i,v}\big[F(S_i)_{c,v}\big], \qquad M_{c,v} = \frac{1}{N_s s_c}\sum_i F(S_i)_{c,v}.$$

The standard deviation spans images and locations within each channel and is floored at 10^{-3}. This normalization balances channels; averaging reduces scene variation while retaining recurring structure such as the table and robot base. We match mean features of the warped target images to M at corresponding spatial locations:

$$\mathcal{L}(\theta) = \frac{\sum_v w_v \frac{1}{C}\sum_c \ell_{0.5}\Big(\frac{1}{N_t s_c}\sum_j F(W_\theta(T_j))_{c,v} - M_{c,v}\Big)}{\sum_v w_v}, \tag{3}$$

where ℓ_β is the smooth L1 loss with transition $\beta = 0.5$. The weight w_v is one inside the feature map and zero on its outer border. Encoder weights and normalization statistics stay frozen; gradients pass through the encoder and sampler to the warp parameters alone.

We normalize the homography as $\bar{H}(\theta) = H(\theta)/[H(\theta)]_{33}$ and regularize it toward identity:

$$\mathcal{J}(\theta) = \mathcal{L}(\theta) + \lambda\|\bar{H}(\theta) - I_3\|_F^2, \tag{4}$$

where $[H(\theta)]_{33}$ denotes the bottom-right entry of $H(\theta)$, I_3 is the 3×3 identity matrix, and $\lambda = 0.05$. We run Adam from three initializations and select the final iterate with the lowest $\mathcal{J}$ as $\hat{\theta}$. The fitted warp remains fixed during deployment and adds one resampling per frame. If the geometry checks fail, we fall back to the unadapted policy and count the fit as a calibration failure.

To test whether recovery requires learned features, we also evaluate RGB matching at the same spatial resolution and a pixel-correlation variant, ReView-pix, which matches mean grayscale images (Appendix A.2). Both alternatives use the same warp family.

4 FROM IMAGE ALIGNMENT TO POLICY RECOVERY

A useful correction must be identifiable from the reference images and preserve the information used for control. We analyze these requirements locally, writing π for the frozen policy π_ψ.

4.1 WHAT CAN A FEW IMAGES IDENTIFY?

Around a fixed warp θ^*, write the matching residual as $r(\theta^* + \delta) \simeq b_0 + \varepsilon + J\delta$. Here δ is a parameter perturbation, b_0 is systematic scene or warp-model mismatch, ε is zero-mean sampling noise, and J is the residual Jacobian. For independent images with covariances Σ_s and Σ_t, holding normalization and residual weights fixed, $\mathrm{Cov}(\varepsilon) = \Sigma_s/N_s + \Sigma_t/N_t$.

Proposition 1 (local reference error). *For fixed residual and regularizer Jacobians, let J have full column rank with smallest singular value $s > 0$. The minimizer of the local quadratic model of Eq.* (4) *satisfies*

$$\mathbb{E}\|\hat{\delta}\|^2 \leq \|d_0\|^2 + \frac{1}{s^2}\left(\frac{\operatorname{tr}\Sigma_s}{N_s} + \frac{\operatorname{tr}\Sigma_t}{N_t}\right), \tag{5}$$

where d_0 is the displacement selected by the noiseless, regularized local objective. The derivation and the expression for d_0 are in Appendix C.

More images reduce sampling noise; comparable scene states reduce systematic mismatch. Otherwise, an object shift can be mistaken for camera motion. Initial frames avoid state changes caused by executing the displaced policy.

Spatial gradients make image motion observable. For a translation-equivariant feature map on a periodic domain, translation preserves the spatial mean but changes individual locations. Spatial matching retains these changes, allowing many locations to constrain a few warp parameters. Section 5.3 tests spatial pooling, reference counts, and scene correspondence.

4.2 WHICH PARTS OF THE SCENE DOES ONE WARP ALIGN?

Pure rotation induces a depth-independent homography; translation creates parallax. Correcting a lateral camera displacement t_x for a plane at depth Z_0 leaves a point at depth Z misaligned by $f_{\mathrm{px}}|t_x||Z^{-1} - Z_0^{-1}|$ pixels, with focal length f_{px}.

Let each region k favor a local correction θ_k^*, with matching residual $r_k(\theta) \simeq J_k(\theta - \theta_k^*)$ and $C_k = J_k^\top J_k$. Minimizing the summed squared residuals gives

$$\hat{\theta} = \Big(\sum_k C_k\Big)^{-1} \sum_k C_k \theta_k^*, \tag{6}$$

for a full-column-rank stacked Jacobian. The regularized form is given in Corollary 1 (Appendix C).

The matrices C_k determine each region's influence. A large textured table can dominate pixel matching, while features sensitive to the robot can shift the fit toward the gripper. One warp thus balances alignment across depths (Section 5.4).

4.3 WHY CAN AN IMPERFECT IMAGE SUPPORT THE RIGHT ACTION?

Let z collect the policy's nominal image inputs at a fixed scene history, and let e_θ be the difference after warping the displaced observations. Keep the nonvisual inputs and policy sampling noise fixed. The policy responds to e_θ through its image Jacobian $A = D_z\pi(z, p)$; let $Q = A^\top A$.

Proposition 2 (local action error). *If the policy Jacobian is L-Lipschitz on the segment from z to $z + e_\theta$, then*

$$\|\pi(z + e_\theta, p) - \pi(z, p)\| \leq \sqrt{e_\theta^\top Q e_\theta} + \tfrac{L}{2}\|e_\theta\|^2. \tag{7}$$

The proof follows from a Taylor expansion (Appendix C.3). The leading term weights image errors by their effect on actions. Visible distortions can matter little in directions to which the policy is insensitive: recovering robot–object geometry can improve actions while leaving the background imperfect. Exact table alignment need not minimize action error.

Matching weights C_k select the warp; policy sensitivity Q determines its effect. They need not agree, so different policies can prefer different corrections. When the closed loop tolerates small action errors, recovery can extend over a rollout; Appendix C.3 gives the finite-horizon bound. Section 5.4 tests the image–action distinction.

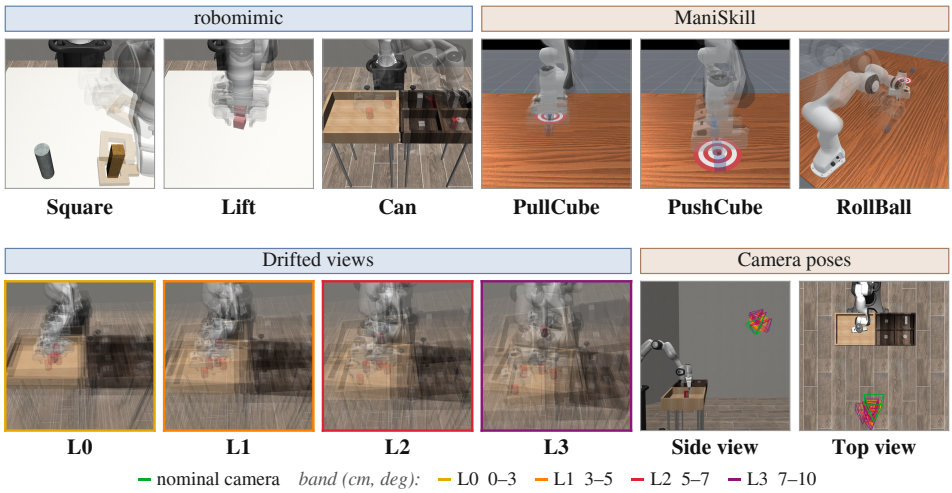

Figure 3: Six tasks and camera drift. Top: overlaid demonstration frames. Bottom: displaced views and camera poses on Can across drift levels. Green marks the nominal camera.

5 Experiments

5.1 Setup

Tasks and policies We train and evaluate Diffusion Policy (DP), ACT, and SmolVLA (Chi et al., 2023; Zhao et al., 2023; Shukor et al., 2025) on robomimic Square, Lift, and Can and ManiSkill3 PullCube, PushCube, and RollBall (Mandlekar et al., 2022; Tao et al., 2025) (Figure 3, top). On ManiSkill, DP and ACT use RGB alone; SmolVLA also receives end-effector and gripper state.

Camera drift and calibration The task and action coordinates remain unchanged while we translate and rotate the camera at four increasing drift levels, L0–L3 (Figure 3, bottom). Each task and level uses five camera poses, with 25 trials per pose on robomimic and 50 on ManiSkill. Methods evaluated with the same policy share camera poses, initial states, and calibration images. ReView fits one fixed warp per pose from 16 demonstration frames and four unpaired reset images, without target actions, depth, extrinsics, or task rollouts. We report success rates and variation across cameras. Training, drift ranges, and paired uncertainty are detailed in Appendices A and B.4.

Comparisons The main comparison includes no adaptation, ActMAD network adaptation (Mirza et al., 2023), and ReView. Nominal-view evaluation measures each policy's performance before camera drift. The table-plane oracle uses true camera geometry to align the table and provides a privileged geometric reference. We study matching objectives and classical registration separately in Section 5.3; AdaBN and implementation details are in Appendix A.6.

5.2 Recovering frozen policies

ReView improves all three policy families on both simulators (Table 1). Averaged over six tasks, success rises from 34.2% to 62.3% for DP, from 50.9% to 73.1% for ACT, and from 20.2% to 44.4% for SmolVLA. The same input correction recovers both task-specific policies and a fine-tuned VLA. With ReView, ACT nearly solves robomimic Lift and Can, while Square remains difficult, consistent with the precise nut-on-peg alignment required by the task.

The adaptation baselines are less consistent across policies. ActMAD changes DP performance little and reduces ACT performance, although it helps SmolVLA on ManiSkill. The table-plane oracle also behaves differently across policy families: it trails ReView for DP and ACT but performs better for SmolVLA. Section 5.4 examines how the same geometric correction can have different effects on different policies.

Table 1: Task success (%). Nominal view uses 100 trials at the original camera without adaptation; other rows average L0–L3. Subscripts give SDs across five groups of camera poses, each pooling L0–L3 (Appendix B.4). Avg. averages six tasks; colored values give gains over no adaptation in percentage points. Table 2 compares input-alignment methods.

	robomimic			ManiSkill			
	Square	Lift	Can	Pull	Push	Roll	Avg.
Diffusion Policy							
Nominal view	71.0	99.0	89.0	100.0	91.0	66.0	86.0
No adapt.	$14.2_{\pm 11.8}$	$45.6_{\pm 28.6}$	$35.0_{\pm 14.6}$	$57.9_{\pm 19.5}$	$29.1_{\pm 14.5}$	$23.3_{\pm 4.2}$	34.2
Oracle	$24.4_{\pm 11.1}$	$70.2_{\pm 23.6}$	$66.4_{\pm 3.2}$	$74.5_{\pm 13.1}$	$30.0_{\pm 20.5}$	$39.3_{\pm 12.9}$	50.8 +16.6
ActMAD	$14.4_{\pm 14.0}$	$46.8_{\pm 29.3}$	$35.2_{\pm 15.5}$	$58.4_{\pm 21.0}$	$28.9_{\pm 14.6}$	$24.4_{\pm 4.5}$	34.7 +0.5
ReView (ours)	$26.2_{\pm 11.8}$	$76.0_{\pm 24.7}$	$76.4_{\pm 4.3}$	$91.7_{\pm 15.8}$	$54.9_{\pm 14.1}$	$48.6_{\pm 8.0}$	62.3 +28.1
ACT							
Nominal view	53.0	100.0	96.0	99.0	97.0	62.0	84.5
No adapt.	$31.2_{\pm 6.1}$	$75.2_{\pm 13.2}$	$65.0_{\pm 17.1}$	$63.8_{\pm 14.0}$	$43.9_{\pm 21.5}$	$26.5_{\pm 6.4}$	50.9
Oracle	$42.4_{\pm 6.1}$	$98.6_{\pm 1.5}$	$92.6_{\pm 5.3}$	$73.1_{\pm 12.4}$	$42.7_{\pm 25.2}$	$40.0_{\pm 13.0}$	64.9 +14.0
ActMAD	$29.4_{\pm 7.7}$	$51.6_{\pm 8.4}$	$46.8_{\pm 9.0}$	$47.0_{\pm 8.9}$	$44.2_{\pm 21.0}$	$27.3_{\pm 7.3}$	41.1 −9.9
ReView (ours)	$44.2_{\pm 5.1}$	$99.6_{\pm 0.9}$	$98.2_{\pm 1.9}$	$80.3_{\pm 24.4}$	$72.8_{\pm 9.0}$	$43.4_{\pm 7.7}$	73.1 +22.1
SmolVLA							
Nominal view	42.0	95.0	76.0	95.0	74.0	43.0	70.8
No adapt.	$5.4_{\pm 3.5}$	$44.8_{\pm 9.4}$	$4.4_{\pm 4.5}$	$39.2_{\pm 15.5}$	$18.8_{\pm 9.9}$	$8.4_{\pm 5.0}$	20.2
Oracle	$11.6_{\pm 3.2}$	$92.6_{\pm 2.5}$	$52.6_{\pm 14.0}$	$82.8_{\pm 6.5}$	$46.1_{\pm 3.4}$	$11.4_{\pm 5.2}$	49.5 +29.3
ActMAD	$5.0_{\pm 3.2}$	$45.8_{\pm 12.9}$	$5.4_{\pm 3.8}$	$49.8_{\pm 19.8}$	$25.3_{\pm 14.8}$	$7.5_{\pm 4.6}$	23.1 +3.0
ReView (ours)	$12.8_{\pm 4.8}$	$88.6_{\pm 5.2}$	$53.2_{\pm 13.4}$	$64.6_{\pm 15.8}$	$31.7_{\pm 10.4}$	$15.8_{\pm 8.2}$	44.4 +24.3

Table 2: DP success (%) over three tasks and L0–L3 per simulator. ✓/× indicate spatial/pooled matching. ECC fits eight parameters; other fitted methods use six.

Method	Spatially resolved matching	robomimic	ManiSkill
No adapt.	–	31.6	36.8
Pooled features	×	32.5	35.3
RGB matching	✓	57.6	61.8
ReView-pix	✓	58.7	57.3
ECC	✓	58.5	56.6
ReView (ours)	✓	**59.5**	**65.1**

5.3 Spatial information and image alignment

We use DP to compare matching objectives and registration on both simulators (Table 2). Pooled features average the early feature map over space; RGB matches downsampled images on the same grid; ReView-pix uses pixel correlation. These variants share ReView's six-parameter warp and optimizer. ECC (Evangelidis & Psarakis, 2008) instead fits an eight-parameter homography with its own solver. All methods use the same images; failed calibration uses no adaptation.

Pooling the early features leaves success near the unadapted level on both simulators: 32.5% versus 31.6% on robomimic and 35.3% versus 36.8% on ManiSkill. Keeping the same features at their spatial locations raises success to 59.5% and 65.1%. Because the encoder, reference images, and warp are unchanged, this contrast isolates the information lost by spatial averaging. An activation can move across the image while barely changing its channel average; matching corresponding locations makes that motion visible to calibration, as analyzed in Section 4.1.

Pixels already supply much of this information. Spatial RGB matching reaches 57.6% on robomimic and 61.8% on ManiSkill, well above pooling and close to ReView. Thus the recovery does not require a learned representation in every scene. Feature choice still affects the resulting correction: on ManiSkill, ReView reaches 65.1%, compared with 57.3% for pixel correlation and 56.6% for ECC. Its gains over these methods come from PullCube and PushCube, while ECC does better on RollBall. The question is therefore which spatial relationships each objective preserves. We examine this directly below; comparisons for ACT and SmolVLA are in Appendix B.1.

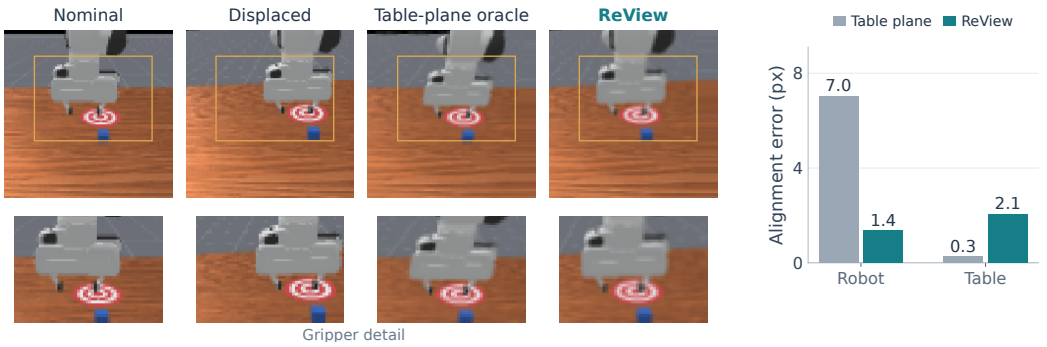

Figure 4: Aligning the table does not align the robot. In PullCube, the table-plane oracle leaves the gripper displaced, while ReView better aligns it with the nominal view. Enlarged views highlight the gripper; the plot compares alignment errors on the robot and table (lower is better).

Reference quality also matters at a fixed image count. With four source and four target images, replacing unpaired references by nominal-view renders of the same scenes reduces median warp error from 7.2 to 3.1 px for matching final-feature means and standard deviations (late moments). Can success rises from 3/30 to 25/30, while Square changes from 5/30 to 7/30, over three cameras per task. Separately, increasing source images from one to sixteen leaves aggregate success unchanged in the reference-count ablation (Appendices B.6 and B.7). These results are consistent with Proposition 1: increasing sample counts addresses noise, while pairing reduces systematic mismatch.

5.4 Which view does the policy need?

The oracle in Table 1 aligns the table using true camera geometry, yet does not consistently recover the best policy performance. We use regional image errors and action predictions to understand why.

PullCube makes the geometric compromise visible (Figure 4): the cube sits about 2 cm above the table and the gripper about 16 cm above it. Camera translation therefore moves their projections by different amounts. The table-plane oracle accurately aligns the table and nearby cube, but leaves the gripper displaced. ReView distributes the correction across these depths: at L2, robot error falls from 7.0 to 1.4 px while table error rises from 0.3 to 2.1 px. Moving the reference plane all the way to gripper height misaligns the cube and performs worse (Appendix D.5): recovery requires balancing the alignment of robot and object.

The regional weights in Eq. (6) help explain the fit. At L2 on PullCube and PushCube, the robot contributes 32–35% of $\mathrm{tr}(J^\top J)$ with features, versus 21–24% with pixel correlation. The regional model predicts ReView's warp within 1.4–1.9 px, compared with 4.3–4.6 px for the table plane (Appendix D.5). Its geometric prediction and the policy's preference remain distinct: DP and ACT benefit from this compromise, while SmolVLA prefers the table-plane oracle on the cube tasks.

The same distinction appears in action predictions. At matched scene histories and diffusion noise, ReView reduces DP action root-mean-square error (RMSE) by about 70% on Can and 36–48% on Square (L2; Appendix D.4). Pixel correlation performs similarly. On Square's unadapted trajectories, late moments give larger RGB error than ReView (0.251 versus 0.165) but smaller action error (0.246 versus 0.299). This reversal of the image and action rankings illustrates Proposition 2: policy sensitivity, rather than image error alone, determines how residual distortion affects actions.

5.5 Recovery across drift levels

Figure 5 separates task recovery from calibration failure on robomimic DP. We also include SIFT with RANSAC (Lowe, 2004; Fischler & Bolles, 1981) as a local-feature registration baseline. L0–L3 use the main evaluation cameras, while L4–L5 extend the drift to 10–18 cm and degrees (Appendix B.3).

At L3, ReView retains 45.1% success versus 5.9% without adaptation. ReView and ReView-pix remain close, with valid fits throughout L0–L3; SIFT and ECC fail in nine and five of 15 L3 conditions. ECC is competitive when calibration succeeds: over its 53 valid L0–L3 conditions, it reaches

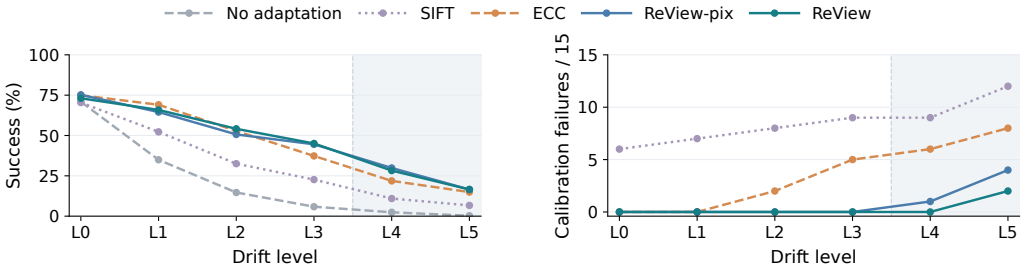

Figure 5: Recovery as camera drift increases. Left: DP success on robomimic (375 trials per method and level). Right: calibration failures among 15 task–camera conditions. Shading marks the severe-drift extension; failed calibration uses the unadapted policy.

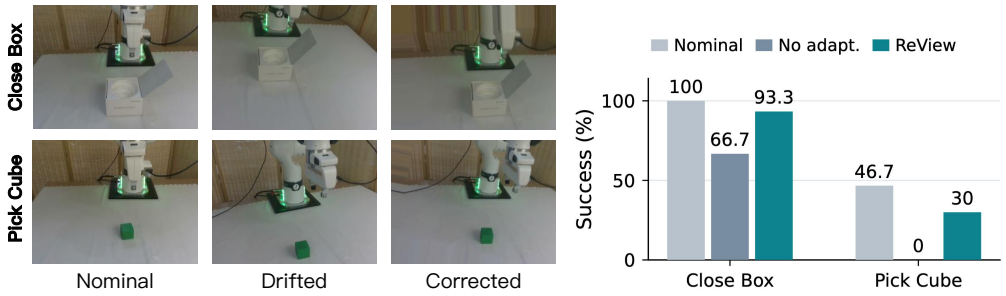

Figure 6: Example real-world views (left) and success rates (right). Per task, displaced-view results pool three camera poses with ten trials each; the nominal view uses 30 trials.

66.3% success versus 64.5% for ReView. Its lower overall performance therefore partly reflects calibration failures.

Under more severe drift, valid calibration is no longer sufficient to maintain performance. ReView has no calibration failures at L4, yet success falls to 28.3%. At L5, success drops further to 16.5%, with only two calibration failures.

5.6 Real-World Experiments

We evaluate Close Box and Pick Cube on a Franka Research 3 with Diffusion Policies trained on 100 demonstrations per task. Each task uses three displaced camera poses; one fitted warp is reused across ten trials with different object placements at each pose (Appendix A.5). On a single RTX 4090, reset-image collection, warp fitting, and resumed policy execution took at most five minutes after camera displacement, without retraining the policy.

ReView raises success from 66.7% to 93.3% on Close Box (nominal: 100%) and from zero to 30% on Pick Cube (nominal: 46.7%; Figure 6). Close Box tolerates larger alignment errors: closing the lid suffices. Pick Cube requires precise gripper–object alignment; in failed trials, the gripper often struck the cube instead of grasping it.

6 Conclusion

We presented ReView, which recovered frozen visuomotor policies from camera drift by fitting a six-parameter input warp from demonstration images and four unpaired reset images. Across three policy families in simulation and Diffusion Policy on a real robot, ReView substantially improves performance without updating the policy. Our analysis and ablations show that spatial correspondence is critical for calibration. Under parallax, useful corrections may trade off alignment across scene depths. Effective calibration therefore need not reconstruct the nominal view exactly; it must preserve the visual relationships that matter for the policy's actions.

AI USE STATEMENT

Generative AI tools, including OpenAI ChatGPT and Anthropic Claude, were used to assist with and polish manuscript writing, and for limited code review. They were not used for literature retrieval or discovery, research ideation or implementation, synthetic data generation, or mathematical proofs. All AI-assisted writing and code suggestions were reviewed by the authors, who take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work evaluates test-time input calibration for robot manipulation policies in simulation and on a real robot. Task success does not establish safe deployment: an incorrectly fitted warp could produce unsafe actions. Physical use should retain the deployment system's safeguards and supervision. No human-subject data were collected.

REPRODUCIBILITY STATEMENT

Section 3 specifies the ReView calibration procedure, including the warp parameterization and optimization objective. Appendix A provides policy training, checkpoint selection, camera-drift sampling, baseline implementations, and evaluation protocols, including the real-robot setup. Appendix C states the assumptions and provides complete proofs for the theoretical results, while the remaining appendices report additional ablations and geometric analyses. Simulation results are backed by stored per-episode records containing the evaluated checkpoints, reference images, fitted warps, and initial states. The supplementary material includes code for reproducing the robomimic experiments; the full codebase will be released with the camera-ready version.

REFERENCES

Simon Baker and Iain Matthews. Lucas–Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

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*, pp. 17–40, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b.

Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. RoVi-Aug: Robot and viewpoint augmentation for cross-embodiment robot learning. In *Proceedings of the 8th Conference on Robot Learning*, pp. 209–233, 2025.

Linghao Chen, Yuzhe Qin, Xiaowei Zhou, and Hao Su. EasyHeC: Accurate and automatic hand-eye calibration via differentiable rendering and space exploration. *IEEE Robotics and Automation Letters*, 8(11):7234–7241, 2023.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023.

Georgios D. Evangelidis and Emmanouil Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024.

Songen Gu, Yuhang Zheng, Weize Li, Yupeng Zheng, Yating Feng, Xiang Li, Yilun Chen, Pengfei Li, and Wenchao Ding. VistaBot: View-robust robot manipulation via spatiotemporal-aware view synthesis. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2026.

Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A. Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *Proceedings of the International Conference on Learning Representations*, 2021.

Hyeongjun Heo, Seungyeon Woo, Sang Min Kim, Junho Kim, Junho Lee, Yonghyeon Lee, and Young Min Kim. AnyCamVLA: Zero-shot camera adaptation for viewpoint robust vision-language-action models. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2026. To appear.

Bingqi Huang, Bingchuan Wei, Xuan Wang, Yingkai Cai, and Zhaokui Wang. Cross-view action consistency for camera-robust vision-language-action policies. *arXiv preprint arXiv:2608.06965*, 2026.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2017–2025, 2015.

Tianchong Jiang, Jingtian Ji, Xiangshan Tan, Jiading Fang, Anand Bhattad, Vitor Guizilini, and Matthew R. Walter. Do you know where your camera is? view-invariant policy learning with camera conditioning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2026.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, 2025a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, pp. 2679–2713, 2025b.

Byungkun Lee, Dongyoon Hwang, Dongjin Kim, Hojoon Lee, Hyunseung Kim, Jaegul Choo, and Minho Park. See like a robot: Robot-centric pointmaps for VLA models. *arXiv preprint arXiv:2607.11498*, 2026.

Timothy E. Lee, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Oliver Kroemer, Dieter Fox, and Stan Birchfield. Camera-to-robot pose estimation from a single image. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 9426–9432, 2020.

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, 2026a.

Wenhao Li, Xueying Jiang, Quanhao Qian, Deli Zhao, Shijian Lu, Gongjie Zhang, and Ran Xu. From fixed to free cameras: Calibration-free view-robust vision-language-action model. *arXiv preprint arXiv:2607.05396*, 2026b.

Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning*, pp. 1678–1690, 2022.

Muhammad Jehanzeb Mirza, Pol Jané Soneira, Wei Lin, Mateusz Kozinski, Horst Possegger, and Horst Bischof. ActMAD: Activation matching to align distributions for test-time-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24152–24161, 2023.

Yechan Park and HyunJin Kim. GS-VLA: Plug-and-play viewpoint canonicalization for frozen VLA policies via gaussian splatting. *arXiv preprint arXiv:2608.19066*, 2026.

Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 30613–30632, 2023.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Jiachen Sun, Mark Ibrahim, Melissa Hall, Ivan Evtimov, Z. Morley Mao, Cristian Canton Ferrer, and Caner Hazirbas. VPA: Fully test-time visual prompt adaptation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 5796–5806, 2023.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9229–9248, 2020.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating GPU parallelized robot simulation and rendering for generalizable embodied AI with ManiSkill3. In *Proceedings of Robotics: Science and Systems*, 2025.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23–30, 2017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *Proceedings of the International Conference on Learning Representations*, 2021.

Weiyao Wang and Gregory D. Hager. Domain adaptation of visual policies with a single demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 17208–17215, 2024.

Sizhe Yang, Yanjie Ze, and Huazhe Xu. MoVie: Visual model-based policy adaptation for view generalization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21507–21523, 2023.

Xinzhang Yang, Renjun Wu, Jinyan Liu, and Xuesong Li. ReMAP-DP: Reprojected multi-view aligned pointmaps for diffusion policy. *arXiv preprint arXiv:2603.14977*, 2026.

Takuma Yoneda, Ge Yang, Matthew R. Walter, and Bradly C. Stadie. Invariance through latent alignment. In *Proceedings of Robotics: Science and Systems*, 2022.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Proceedings of Robotics: Science and Systems*, 2024.

Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38629–38642, 2022.

Zhiyuan Zhang, Zhengtong Xu, Jai Nanda Lakamsani, and Yu She. Canonical policy: Learning canonical 3-D representation for SE(3)-equivariant policy. *IEEE Transactions on Robotics*, 42: 2325–2344, 2026.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, 2023.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, pp. 2165–2183, 2023.

A IMPLEMENTATION DETAILS

A.1 SAMPLING AND OPTIMIZATION

For robomimic DP and ACT, the policy input is a 76×76 crop from an 84×84 sensor frame. The normalized focal length is adjusted as $f_{\text{crop}} = f_{\text{sensor}}(84/76)$; the output crop is sampled from the full sensor frame. Bilinear sampling uses border padding and `align_corners=False`. Source references are saved initial RGB observations from training demonstrations.

These two encoders expose a $64\times19\times19$ early feature map; the SmolVLA and ManiSkill representations are specified below. The channel scale s_c uses the population standard deviation and is floored at 10^{-3}, the smooth-L1 transition is $\beta = 0.5$, w_v masks the feature-map boundary, and the regularization weight is $\lambda = 0.05$. The translation grid search uses offsets $\{-24, -16, -8, 0, 8, 16, 24\}$, measured in pixels of the 76×76 crop. The two lowest-objective nonidentity starts and identity each receive 200 Adam steps at learning rate 0.005. The final calibration objective selects the restart on the same target samples. For ReView and ReView-pix, geometric validity requires finite sampling coordinates, at least 50% of output sampling locations inside the full sensor image, $\det \bar{H} > 0$, and a projective denominator $(\bar{H}q)_3 \geq 0.05$ at each corner q of the normalized output crop. Failure of these checks, registration nonconvergence, or insufficient correspondences counts as a calibration failure. All trials for that task–camera condition then use the unadapted policy and remain in the reported success rate. No ReView calibration fails in the main evaluation.

A.2 PIXEL-CORRELATION VARIANT

Define $g(I) = 0.299I_R + 0.587I_G + 0.114I_B$, $x = 16^{-1} \sum_i g(S_i)$, and $y_\theta = 4^{-1} \sum_j g(W_\theta(T_j))$. Flattening the images and centering them as $\tilde{x} = x - \bar{x}\mathbf{1}$, we use

$$\mathcal{L}_{\text{corr}}(\theta) = 1 - \frac{\tilde{x}^\top \tilde{y}_\theta}{\sqrt{\|\tilde{x}\|^2 \|\tilde{y}_\theta\|^2 + 10^{-8}}}. \tag{8}$$

ReView-pix uses the classical correlation criterion (Evangelidis & Psarakis, 2008) on mean images, but retains ReView's six-parameter warp and Adam optimizer, rather than the ECC solver. It uses the same references and fitting budget as ReView. Unlike Eq. (3), it weights all image locations equally and uses no encoder features. The RGB-grid control below matches ReView's spatial resolution and border mask more closely.

A.3 ROBOMIMIC POLICIES AND EVALUATION

In the main evaluation, methods for a given policy share camera poses, initial states, random seeds, and the 16 source and four target reference images. The policy families use their own evaluation initializations. The smaller warp and reference ablations specify their sampling conditions alongside the corresponding results.

Rotation and translation magnitudes are drawn independently and uniformly from L0 $[0, 3)$, L1 $[3, 5)$, L2 $[5, 7)$, or L3 $[7, 10)$, in degrees and centimeters respectively, each about an independent random 3D direction; the translation radius is therefore not uniform in volume. Each policy's main robomimic evaluation contains 1,500 trials per method (three tasks, four drift levels, five cameras, and 25 initializations).

Diffusion Policy We train one DP from scratch on 200 demonstrations per task. Its ResNet–18 encoder uses GroupNorm and spatial-softmax pooling; visual features and the nine-dimensional end-effector and gripper state condition a 1D U-Net. The policy observes two frames, predicts 16 actions, and executes eight before replanning. Training uses random 76×76 crops from the 84×84 images; evaluation uses center crops. We minimize the noise-prediction mean squared error with a 100-step DDPM and a cosine noise schedule, using 100 denoising steps at inference. AdamW uses batch size 64, learning rate 10^{-4}, weight decay 10^{-6}, and cosine learning-rate decay after 500 warmup updates. We maintain an exponential moving average (EMA) of the weights and select models by success on 50 nominal-camera rollouts. The selected models have 15,000–25,000 training updates; no drift evaluation enters selection.

ACT We train ACT on 180 demonstrations and validate on 20, with all parameters, including its ResNet–18 backbone, randomly initialized. Its input is RGB and a nine-dimensional end-effector/gripper state. It predicts 16 seven-dimensional actions and executes eight before replanning. Training uses random 76×76 crops; evaluation uses center crops. Train-only statistics normalize state and action, and RGB uses ImageNet constants. Batch normalization is frozen at evaluation.

The ACT CVAE uses width 512, four encoder and seven decoder layers, eight heads, feed-forward width 3200, latent width 32, dropout 0.1, and KL weight 10. Training runs for 30,000 updates with batch size 128, AdamW learning rate and weight decay 10^{-4}, and gradient clipping at 1. Training uses bfloat16 autocast with the Gaussian KL computed in FP32; evaluation is FP32 with zero latent. We select checkpoints by success on ten nominal-camera development episodes, using validation action L1 loss to break ties. Drift outcomes are not used for selection.

SmolVLA We fine-tune SmolVLA (Shukor et al., 2025) separately for each task using the official LeRobot recipe. Starting from the pretrained base model, we freeze the vision-language backbone and train the action expert for 100,000 steps at batch size 64 with autocast disabled, on the same 200 demonstrations as DP. The frozen vision-language backbone uses bfloat16 at deployment. Its inputs are the 84×84 agent-view image and nine joint coordinates: seven arm joints and two gripper joints. It predicts chunks of 50 actions and executes each chunk before replanning. We use the 100,000-step checkpoint for each task. ReView matches the $768 \times 32 \times 32$ patch embedding of the frozen visual encoder and warps the full sensor frame, since this policy has no crop; the layer was fixed before evaluation.

A.4 MANISKILL

We use the same calibration recipe as on robomimic. DP and ACT receive RGB only, selected through nominal-camera development evaluations; SmolVLA also receives end-effector and gripper state and is co-trained across tasks. Thus cross-simulator comparisons change policy inputs and training as well as scene geometry.

PullCube, PushCube, and RollBall are the ManiSkill3 (Tao et al., 2025) tasks of the same name with a Franka arm under end-effector delta-pose control and a fixed third-person camera with a one-radian field of view. The distance to its look-at point is 0.67 m for PullCube and PushCube and 1.24 m for RollBall. Demonstrations are replayed from stored states and rendered at 84×84 with $3\times$ supersampling and a box filter. Anti-aliasing avoids sensitivity to fractional-pixel resampling in the native renderer: on PushCube at the nominal view, half-pixel shifts reduced success to 32–35/50 without anti-aliasing, versus 44–45/50 with it.

Diffusion Policy We train DP with the same crop augmentation, diffusion objective, and observation and action horizons as on robomimic. A convolutional encoder supplies flattened spatial features to the 1D U-Net. Training uses AdamW with learning rate 10^{-4}, weight decay 10^{-6}, batch size 256, and EMA weights. We select RGB-only input on nominal-view development rollouts. Training uses 30,000 updates for PullCube and PushCube and 200,000 for RollBall, whose longer demonstrations require a 400-step evaluation horizon. Model choices do not use drift outcomes. F is the encoder's first stage. Each camera evaluates the same 50 initial states across methods, in parallel environments, using 16 demonstration references and four reset images. This encoder has no normalization layers, so ActMAD's objective and optimizer constants are applied to the four max-pooling outputs.

ACT We train ACT with the robomimic architecture and optimizer (Appendix A.3) for 30,000 steps on the same rendered demonstrations. Actions are normalized as for DP, because z-scoring the nearly constant gripper channel produced non-finite gradients. As for DP, the proprioceptive input is zeroed based on nominal-view development performance. Checkpoints are selected using 50 nominal-camera development episodes. F is the output of the first residual stage ($64 \times 19 \times 19$), as on robomimic, and ActMAD acts on the 20 BatchNorm layers of the ResNet-18.

SmolVLA We co-train one SmolVLA on the three tasks using the same rendered frames with per-task language instructions and the robomimic optimization recipe (frozen vision-language backbone, action expert trained for 100,000 steps). Its nine-dimensional state combines end-effector position

and quaternion with two gripper joint coordinates. ReView matches the patch embedding of the frozen visual encoder and warps the full frame, as on robomimic. Its preference for the table-plane oracle on the cube tasks differs from the RGB-only policies (Table 5); architecture, training, and proprioception vary together in this comparison.

A.5 Real-robot platform

Two external Intel RealSense D435i cameras observe a Franka Research 3 (Figure 7). The fixed camera provides the training view; the second is displaced and reoriented to provide the deployment view. The policies are trained on 100 teleoperated demonstrations per task. For each task, we use three distinct displaced camera poses. At each pose, ReView fits one warp from 16 initial demonstration frames and four reset images from that view, without depth or measured extrinsics. Only the warp parameters are optimized; policy and encoder weights remain frozen. The fitted warp is reused for ten trials with different object placements at that pose. Displaced-view success pools these $3 \times 10 = 30$ trials per task and method; nominal-view success uses 30 trials at the fixed camera.



Figure 7: Real-robot platform and the nominal and displaced cameras.

A.6 Adaptation baselines

On robomimic, ActMAD (Mirza et al., 2023) observes all 20 normalization layers in the DP and ACT visual encoders. On ManiSkill, ACT uses the same layers, while DP uses four max-pooling outputs because its encoder has no normalization layers. For SmolVLA, ActMAD observes all 25 LayerNorm outputs in the visual encoder, with statistics computed per token. At every channel and spatial position, it matches batch means and population variances from the 16 source and four target images with an L1 objective, then takes one SGD step on every encoder parameter reached by this loss. We use its ImageNet optimizer settings (learning rate 2.5×10^{-4}, momentum 0.9, weight decay 5×10^{-4}, and Nesterov momentum). Normalization layers remain in evaluation mode during this update. For robomimic SmolVLA, the step uses an FP32 copy of the visual encoder, then casts the weights back to bfloat16 for deployment. The ManiSkill implementation updates the bfloat16 visual encoder directly, computing matching statistics in FP32. AdaBN (Li et al., 2018) instead replaces the running statistics of ACT's 20 BatchNorm layers using activation statistics from the four target images over three passes; DP's GroupNorm layers and SmolVLA's LayerNorm layers have no running statistics. Neither method uses target actions, rewards, adaptation rollouts, or evaluation success during calibration.

On robomimic, AdaBN on ACT succeeds in 4/1,500 trials (0.3%); ActMAD results are reported in Table 1.

These baselines use the same 16 source and four target images as ReView, rather than ActMAD's original online classification stream.

A.7 Computational cost

On a shared RTX 4090, three full DP Can fits take a median 24.21 seconds (range 19.66–25.01), excluding model loading, source-reference computation, and reset collection. The source feature

map, channel scales, and mask occupy 94,116 bytes. The full crop-and-warp input stage takes a median 14.13 ms for batch size two (P10/P90: 1.37/28.38 ms), excluding the encoder and policy. These measurements were collected under GPU contention. Four simulator resets take a median 46.72 seconds including environment setup; this is not a physical-robot collection measurement.

B Additional results

B.1 Input-alignment comparisons

Table 3 complements the DP comparison in Table 2 with task-wise results for all three policies, including camera variation. ReView-pix retains the six-parameter warp and optimizer but replaces feature matching with pixel correlation. ECC uses its own solver and an eight-parameter homography. On ManiSkill, ReView improves ACT over pixel correlation (65.5% versus 57.5%), while SmolVLA favors pixels (43.3% versus 37.4%).

Table 3: Input-alignment success (%) by task, averaged over L0–L3. Subscripts give SDs across five groups of camera poses, each pooling L0–L3 (Appendix B.4); colored values give gains over no adaptation. Bold marks the best result in each policy group. The evaluation conditions are those of Table 1.

	robomimic			ManiSkill			
	Square	Lift	Can	Pull	Push	Roll	Avg.
DP							
ECC	$25.4_{\pm 11.8}$	$\mathbf{76.2}_{\pm 18.5}$	$74.0_{\pm 5.6}$	$81.4_{\pm 11.1}$	$34.8_{\pm 21.7}$	$\mathbf{53.7}_{\pm 11.9}$	57.6 +23.4
ReView-pix	$\mathbf{27.0}_{\pm 12.8}$	$75.2_{\pm 21.3}$	$74.0_{\pm 4.8}$	$87.2_{\pm 15.8}$	$36.7_{\pm 22.2}$	$48.1_{\pm 9.5}$	58.0 +23.9
ReView (ours)	$26.2_{\pm 11.8}$	$76.0_{\pm 24.7}$	$\mathbf{76.4}_{\pm 4.3}$	$\mathbf{91.7}_{\pm 15.8}$	$\mathbf{54.9}_{\pm 14.1}$	$48.6_{\pm 8.0}$	**62.3** +28.1
ACT							
ReView-pix	$43.6_{\pm 2.7}$	$99.4_{\pm 1.3}$	$97.4_{\pm 1.5}$	$75.9_{\pm 21.5}$	$51.1_{\pm 25.7}$	$\mathbf{45.4}_{\pm 5.7}$	68.8 +17.9
ReView (ours)	$\mathbf{44.2}_{\pm 5.1}$	$\mathbf{99.6}_{\pm 0.9}$	$\mathbf{98.2}_{\pm 1.9}$	$\mathbf{80.3}_{\pm 24.4}$	$\mathbf{72.8}_{\pm 9.0}$	$43.4_{\pm 7.7}$	**73.1** +22.1
SmolVLA							
ReView-pix	$12.4_{\pm 5.1}$	$86.4_{\pm 7.8}$	$\mathbf{56.2}_{\pm 10.8}$	$\mathbf{76.8}_{\pm 8.0}$	$\mathbf{36.3}_{\pm 10.1}$	$\mathbf{16.8}_{\pm 8.8}$	**47.5** +27.3
ReView (ours)	$\mathbf{12.8}_{\pm 4.8}$	$\mathbf{88.6}_{\pm 5.2}$	$53.2_{\pm 13.4}$	$64.6_{\pm 15.8}$	$31.7_{\pm 10.4}$	$15.8_{\pm 8.2}$	44.4 +24.3

B.2 Per-level results

Tables 4 and 5 give the per-level robomimic and ManiSkill results underlying Tables 1 and 2. Each robomimic entry uses 125 trials (five cameras with 25 initializations each). Methods within a policy family share cameras, initial states, and the same 16 source and four target frames; ActMAD updates the network from those frames. Nominal-view results are in Table 1.

Each ManiSkill entry uses 250 trials (five cameras with 50 initializations each), on the evaluation conditions of Table 1. Oracle uses the true table-plane homography; bold marks the best non-oracle result per column. The nominal-view rows in Table 1 use 100 independent trials per policy and task, with the same checkpoints as the drift evaluation. These trials are separate from the development rollouts used to select checkpoints and do not affect model selection.

Table 4: Robomimic success rates (%) by drift level. Oracle uses the true table-plane homography. Bold marks the best non-oracle result for each policy and column.

	Square				Lift				Can				
	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3	Avg.
Diffusion Policy													
No adapt.	44.8	9.6	2.4	0.0	92.0	52.8	21.6	16.0	76.0	42.4	20.0	1.6	31.6
ActMAD	43.2	9.6	4.0	0.8	96.0	52.8	20.8	17.6	**79.2**	40.0	20.8	0.8	32.1
ECC	50.4	**34.4**	11.2	5.6	96.8	**96.0**	**71.2**	40.8	77.6	76.8	76.0	65.6	58.5
Oracle (table plane)	44.8	30.4	19.2	3.2	92.0	81.6	60.8	46.4	76.0	76.8	63.2	49.6	53.7
ReView-pix	**51.2**	31.2	12.8	**12.8**	97.6	80.8	67.2	55.2	76.8	81.6	72.0	65.6	58.7
ReView	48.0	33.6	**16.0**	7.2	**98.4**	79.2	67.2	**59.2**	72.8	**84.8**	**79.2**	**68.8**	**59.5**
ACT													
No adapt.	64.8	40.0	14.4	5.6	**100.0**	97.6	65.6	37.6	**99.2**	84.8	58.4	17.6	57.1
ActMAD	57.6	38.4	16.0	5.6	67.2	60.8	44.8	33.6	64.8	58.4	46.4	17.6	42.6
Oracle (table plane)	70.4	56.0	30.4	12.8	100.0	100.0	99.2	95.2	97.6	95.2	90.4	87.2	77.9
ReView-pix	63.2	**59.2**	**38.4**	13.6	**100.0**	**100.0**	99.2	**98.4**	98.4	**99.2**	97.6	94.4	80.1
ReView	**68.0**	54.4	36.8	**17.6**	**100.0**	**100.0**	**100.0**	**98.4**	**99.2**	**99.2**	**99.2**	**95.2**	**80.7**
SmolVLA													
No adapt.	16.8	1.6	2.4	0.8	66.4	40.8	35.2	36.8	13.6	1.6	1.6	0.8	18.2
ActMAD	11.2	4.0	1.6	3.2	59.2	49.6	38.4	36.0	16.0	2.4	1.6	1.6	18.7
Oracle (table plane)	19.2	9.6	11.2	6.4	91.2	92.8	95.2	91.2	67.2	53.6	54.4	35.2	52.3
ReView-pix	**23.2**	**14.4**	**11.2**	0.8	**88.8**	**91.2**	**88.8**	76.8	**70.4**	**57.6**	**52.8**	**44.0**	**51.7**
ReView	21.6	12.8	10.4	**6.4**	88.0	**91.2**	**88.8**	**86.4**	68.0	56.0	52.0	36.8	51.5

Table 5: ManiSkill success rates (%) by policy and drift level. DP and ACT use RGB only; SmolVLA also receives robot state. Bold marks the best non-oracle result per policy and column.

	PullCube				PushCube				RollBall				
	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3	Avg.
Diffusion Policy													
No adapt.	98.4	75.2	38.8	19.2	65.2	29.2	17.2	4.8	56.0	24.0	10.0	3.2	36.8
ActMAD	**98.8**	76.4	38.4	20.0	65.6	29.2	16.0	4.8	55.6	26.4	11.2	4.4	37.2
ECC	98.4	92.8	86.0	48.4	47.6	40.8	26.8	24.0	64.0	**60.0**	**49.6**	**41.2**	56.6
Oracle (table plane)	98.4	90.8	68.4	40.4	52.8	34.4	21.2	11.6	58.4	45.2	31.6	22.0	47.9
ReView-pix	96.8	92.8	86.8	72.4	55.2	42.8	30.4	18.4	61.2	54.0	46.4	30.8	57.3
ReView	98.0	**93.6**	**90.0**	**85.2**	**66.4**	**60.4**	**53.6**	**39.2**	**66.0**	52.8	47.6	28.0	**65.1**
ACT													
No adapt.	96.0	**85.6**	56.8	16.8	86.0	47.6	27.2	14.8	53.2	30.8	16.4	5.6	44.7
ActMAD	80.4	64.0	32.8	10.8	87.6	47.2	26.8	15.2	54.0	32.0	17.6	5.6	39.5
Oracle (table plane)	97.2	90.4	66.8	38.0	92.8	45.6	20.0	12.4	60.8	45.2	31.6	22.4	51.9
ReView-pix	96.4	**85.6**	71.2	50.4	**93.2**	57.2	34.8	19.2	**60.4**	**54.4**	**40.8**	26.0	57.5
ReView	**98.0**	82.8	**76.0**	**64.4**	92.0	**77.2**	**74.8**	**47.2**	59.6	52.0	35.2	**26.8**	**65.5**
SmolVLA													
No adapt.	80.4	42.8	20.8	12.8	46.0	14.0	9.6	5.6	21.6	6.4	3.2	2.4	22.1
ActMAD	79.2	56.4	37.6	26.0	48.4	28.4	18.4	6.0	16.4	7.6	3.6	2.4	27.5
Oracle (table plane)	92.8	83.2	83.2	72.0	51.2	48.0	46.8	38.4	19.6	12.8	7.2	6.0	46.8
ReView-pix	**91.6**	**83.2**	**65.2**	**67.2**	**49.6**	**38.8**	**34.4**	**22.4**	22.4	**18.0**	**12.0**	**14.8**	**43.3**
ReView	90.4	69.6	57.2	41.2	46.0	34.4	28.8	17.6	**25.6**	17.2	7.6	12.8	37.4

B.3 Registration across drift levels

On robomimic, registration uses all 60 task–level–camera combinations of the DP main evaluation (L0–L3), with the same initial states, 16 source images, and four target images. Both classical baselines fit eight-parameter homographies to mean grayscale images using OpenCV. ECC starts from the identity and two offsets selected by a 49-translation grid search; SIFT matches local descriptors and RANSAC fits the homography. Failed fits use the unadapted policy. Figure 5 shows both task recovery and calibration failures. ECC is competitive when fitting succeeds: over its 53 valid conditions, success is 66.3% versus 64.5% for ReView. Its warps lie 1.1–2.4 px from the table-plane

homography. Across L0–L3, ECC fails in 0, 0, 2, and 5 of 15 conditions; SIFT with RANSAC has too few matches in 6, 7, 8, and 9. SIFT success falls from 70.4% at L0 to 22.7% at L3 (52.3% and 32.5% at L1/L2).

On ManiSkill, ECC uses the same fitting procedure and all 60 DP task–level–camera conditions, with 50 trials each. It reaches 56.6% against 65.1% for ReView, a paired difference of 8.4 points with a descriptive 95% camera-bootstrap interval of [2.8, 14.3]. Calibration fails in five conditions (none at L0 or L1, one at L2, and four at L3). On the 55 valid conditions, ECC reaches 60.2% and ReView 65.7%. The three tasks differ: ReView leads on PullCube and PushCube, while ECC recovers RollBall best (Table 3).

Severe drift We extend the robomimic comparison on its main cameras to two severe levels, L4 $[10, 14)$ and L5 $[14, 18)$ centimeters and degrees (375 trials per method and level; Table 6). All methods deteriorate as the displaced camera loses the training view. ReView has no successes on Square here; at L5, two of its 15 warps fail the geometry checks and the valid warps lie a median 12.5 px from the table-plane homography.

Table 6: Severe drift on the main-evaluation cameras: success (%, 375 trials) and calibration failures out of 15 task–camera conditions per level. Failed calibration uses the unadapted policy.

	Success (%)		Calibration failures	
Method	L4	L5	L4	L5
No adapt.	2.4	0.3	–	–
SIFT+RANSAC	10.9	6.7	9	12
ECC	21.9	14.9	6	8
ReView-pix	29.9	16.5	1	4
ReView	28.3	16.5	0	2

B.4 Uncertainty over cameras

For each task and method, let $r_{\ell k}$ be the success rate for camera draw k at drift level ℓ. Tables 1 and 3 report the sample standard deviation of the five rates $\bar{r}_k = \frac{1}{4}\sum_{\ell=0}^{3} r_{\ell k}$, $k = 1, \ldots, 5$. Each rate pools four distinct camera poses with the same draw index, one per level (100 trials on robomimic, 200 on ManiSkill). The reported SD therefore measures variation between these level-averaged rates, not between the 20 individual camera poses.

For paired comparisons, we resample draw indices jointly across methods, retaining the tasks and levels included in each comparison. At a fixed level, each index selects one camera pose per task. We take the 2.5/97.5 percentiles over all 5^5 resamples. Figure 8 shows per-task and per-level intervals for DP and ACT; these describe camera-sampling variation with checkpoints and initial states fixed. Aggregated over the three robomimic tasks and four levels, ReView's gain over no adaptation is 27.9 [19.0, 36.5] points for DP, 23.5 [16.5, 32.9] for ACT, and 33.3 [28.1, 37.4] for SmolVLA, and the ReView−ReView-pix difference is 0.8 [−0.7, 2.3], 0.5 [−0.5, 1.7], and −0.1 [−0.8, 0.4]; on ManiSkill DP (three tasks, four levels, five cameras) the gain is 28.3 [23.9, 32.7] for ReView and 20.6 [14.2, 26.9] for ReView-pix, and their difference 7.7 [4.2, 11.0].

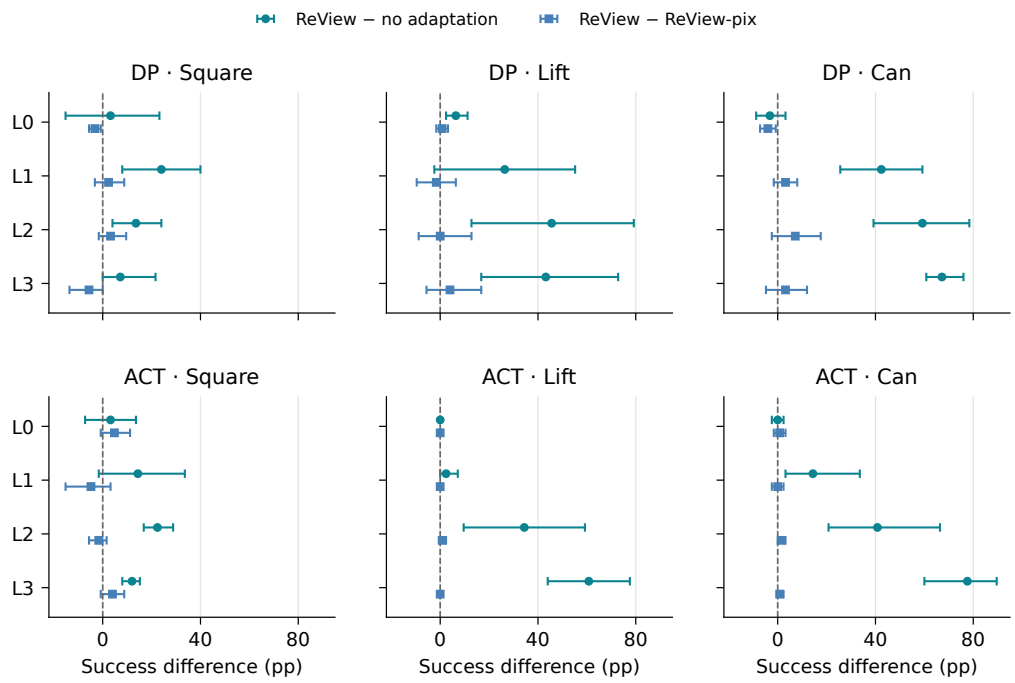

Figure 8: Paired success differences for DP and ACT on robomimic. Points show ReView minus the comparison method; bars are descriptive 95% camera-bootstrap intervals. Gains over no adaptation are generally larger than differences from pixel correlation.

B.5 Matching signals and warp capacity

Table 7 extends the matching-objective comparison to ACT and SmolVLA on robomimic. Spatial RGB matching and pixel correlation remain close to feature matching, while pooling loses most of the gain.

Table 7: Effect of the matching signal on success (%), with the warp and references fixed. Pooled averages early features over spatial locations; RGB matches downsampled images. The first three rows use robomimic.

Spatial grid	No adapt. –	Pooled no	RGB yes	ReView-pix yes	ReView yes
DP	31.6	32.5	57.6	58.7	**59.5**
ACT	57.1	57.1	**80.9**	80.1	80.7
SmolVLA	18.2	23.3	51.1	**51.7**	51.5
DP, ManiSkill	36.8	35.3	61.8	57.3	**65.1**

The signal ablations reuse the main cameras, initial states, references, warp, and optimizer. Pooled features average each channel over interior locations before matching. RGB matching downsamples images to the same 19×19 grid and uses the same border mask for DP and ACT. Figure 9 shows why spatial information matters: pooling can lose recovery even when the unpooled features work well. The late-moment objective matches means and population standard deviations at the final 64-dimensional representation. This is a learned linear projection of 32 spatial-softmax keypoints, each with two coordinates, so it retains location information without an explicit spatial grid. It recovers Lift better than ReView (88.8% versus 76.0%), but performs worse on Can. The loss uses a scalar root-mean-square (RMS) normalization computed from source features and squared error. Its reference images, cameras, initial states, and six-parameter warp match the DP main evaluation.

Warp parameterization The homography variant replaces the six-parameter warp with an unconstrained eight-parameter homography, retaining the spatial objective, sampling region, initialization

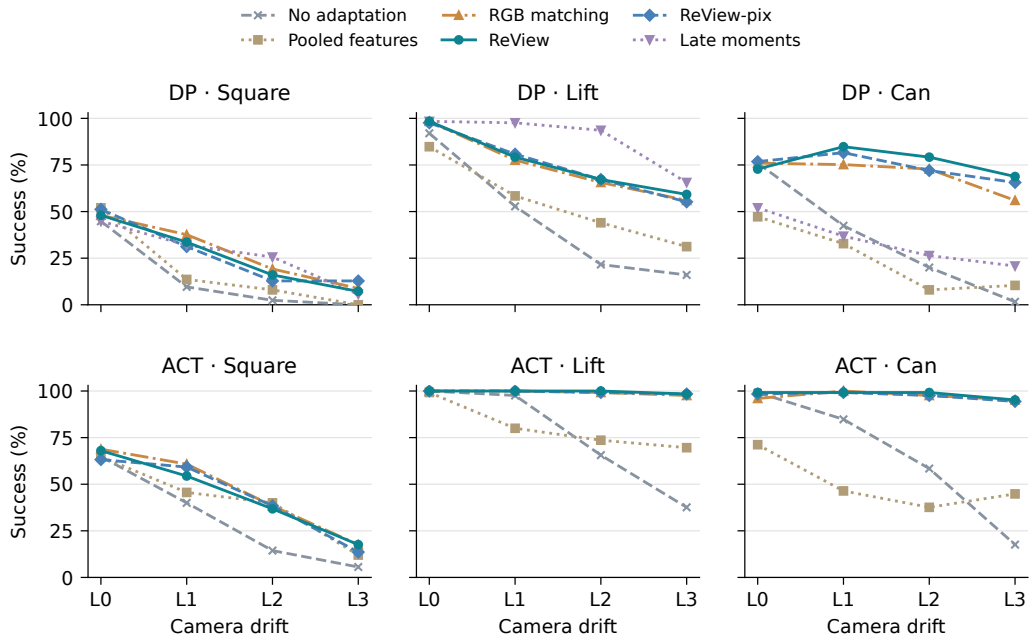

Figure 9: Matching signals under increasing camera drift on robomimic. Spatial matching retains much of the recovery lost by pooling; late-feature moments work well on Lift but less well on Can. Each point uses 125 trials; late moments are evaluated on DP only.

grid, restarts, optimization budget, and regularizer. On the full robomimic DP evaluation it reaches 63.3% against 59.5% (Figure 10), while its fitted warps lie farther from the true table-plane homography (2.3 against 1.8 px, Appendix D.3). The paired gain is 3.7 points, with a descriptive 95% interval of $[-0.6, 8.1]$ from resampling the five camera-draw blocks. In the smaller same-state Can diagnostic, it reaches 24/30 against 27/30 for the six-parameter warp (Table 9, paired references). Both variants recover substantial performance; the extra freedom changes the trade-off between alignment and control.

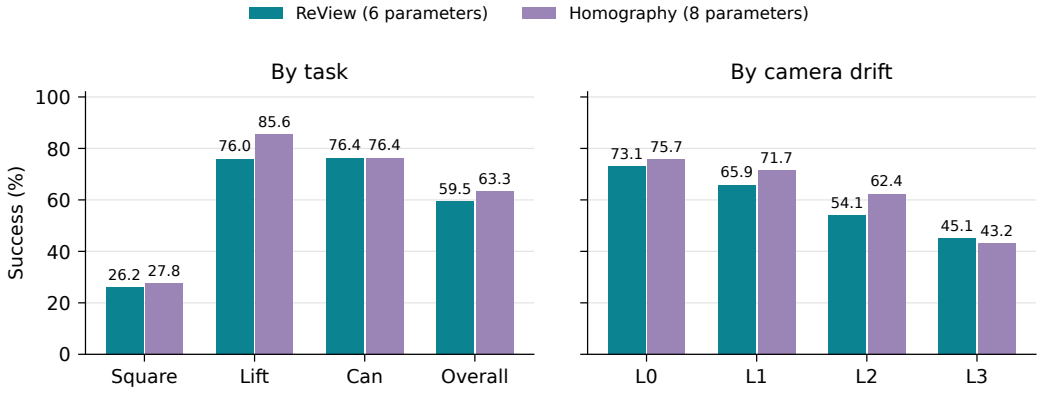

Figure 10: Warp capacity on robomimic DP with the same spatial matching objective. The eight-parameter homography improves Lift and the overall mean, but not the most severe main-evaluation level, L3. Task bars use 500 trials, level bars 375, and Overall 1,500.

Restricted warps For the DP Square/Can L2 warp ablations (three cameras, ten trials each), we compare rotation-only, translation-only, full six-parameter, affine, and unconstrained homography warps with the spatial objective. All use 16 source and four target images and the same optimization

budget. The rotation-only variant initializes pitch and yaw from the angles subtended by the translation grid; the translation-only variant fixes rotation to zero. All fitted warps are valid. These 30-trial comparisons are descriptive: rotation-only works best on Can in this subset, and affine works best on Square; no warp family dominates both tasks (Table 8).

Table 8: Warp ablations on DP Square/Can at L2, with 30 trials per task and warp family (three cameras $\times$ ten trials). Homography is the fitted eight-parameter variant.

	Success (%)				
Task	Rotation	Translation	ReView	Affine	Homography
Square	16.7	13.3	13.3	23.3	13.3
Can	90.0	66.7	73.3	73.3	86.7
Mean	53.3	40.0	43.3	48.3	50.0

B.6 Same-state references

The Square/Can L2 diagnostic uses three cameras and ten initial states per camera. Both late-moment variants use four source images: the first four stored source frames, or nominal-camera renders at the four target reset states. Other fitting and evaluation settings are fixed. Spatial paired fits use the same exact-state renders with six or eight warp parameters; the projected oracle uses the true table-plane homography projected onto the six-parameter family. Pose+Depth reprojects each control frame.

Table 9: Effect of privileged references and geometry on DP recovery: successes out of 30 on Square and Can. Paired references show the same scene states from the nominal camera. Paired 6/8 denote fitted six-/eight-parameter warps; the oracle is projected onto the six-parameter family.

		Late moments		Spatial			Projected	
	No adapt.	unpaired	paired	ReView	paired 6	paired 8	oracle	Pose+Depth
Square	0	5	7	4	5	5	3	5
Can	5	3	25	22	27	24	23	24

Pairing lowers late-moment matching loss in every fit, to 13–29% of the unpaired value on Square and 5–15% on Can. It adds 22 successes on Can but only two on Square. Scene mismatch therefore explains much of Can's failure, while Square remains difficult even with matched references. All fits pass the geometry checks. Median warp error falls from 7.2 to 3.1 px with pairing (Table 11). These results describe the three tested camera poses per task.

B.7 Number and selection of reference images

Ablations of the number of source and target images reuse the cameras and initializations of the DP Square/Can L2 warp ablations, with 30 trials per task and setting. Target-count experiments fix 16 source images; source-count experiments fix four target resets. The single-source variant takes the first stored image; the 32-image variant adds 16 distinct training initial frames. Results for 16 source and four target images, and for no adaptation, are reused across comparisons. Results appear in Figure 11; no-adaptation success is 0.0% on Square and 16.7% on Can.

One and sixteen source images give identical aggregate scores in this ablation. Each image already supplies many spatial constraints; additional references refine the mean feature map rather than add warp parameters. More target resets improve Can, whereas Square retains a substantial gap.

An equal-budget reset-versus-rollout comparison on DP Square/Lift L2 (three additional cameras, 30 trials per task) gives spatial success of 40.0/73.3% from resets and 36.7/83.3% from rollout frames; correlation gives 36.7/90.0% and 36.7/83.3%, respectively. Each episode contributes one reset or predetermined trajectory-quantile frame, with up to 300 control steps for rollout collection. The reset protocol avoids those rollouts; it is not uniformly better in success rate.

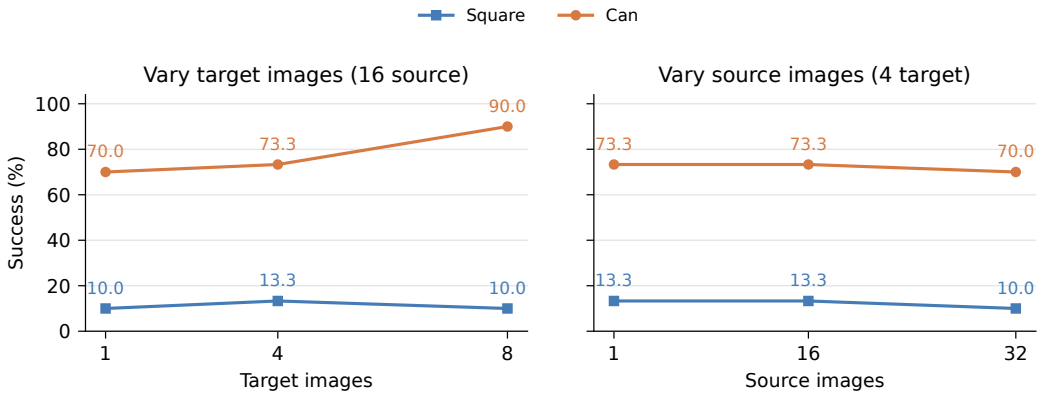

Figure 11: Reference image counts on DP at L2. More target resets help Can in this small diagnostic, whereas additional source images give no consistent gain. Each point uses 30 trials over three cameras; lines connect the tested settings.

Across four alternative source sets on DP Square/Lift/Can at L2, ReView ranges from 61.1% to 67.8% and ReView-pix from 61.1% to 65.6% (90 trials per set on the three additional cameras; no adaptation 8.9%). Four alternative target sets on the Square/Can image-count conditions give 36.7–45.0% for ReView and 40.0–46.7% for ReView-pix (60 trials per set; no adaptation 8.3%). Source sets may overlap; evaluation cameras and states are fixed within each comparison.

B.8 Nominal-camera adaptation

Nominal-camera fits are evaluated in a paired diagnostic with 25 trials per task. Fitting can harm Square even without camera displacement (Table 10).

Table 10: Paired changes in nominal-view DP success relative to no adaptation (percentage points, 25 trials per task).

Task	ReView-pix	ReView
Square	−16.0	−16.0
Lift	0.0	+4.0
Can	−4.0	+4.0
Mean	−6.7	−2.7

C Analysis and proofs

C.1 Systematic mismatch and finite-reference error

A reference-mismatch example Consider a scalar statistic $F(s, c) = s + c$ of scene state s and camera displacement c. If the camera moves by Δ and a correction subtracts u, matching source and target means minimizes $L(u) = (\mu_t + \Delta - u - \mu_s)^2$, giving

$$u^* = \Delta + (\mu_t - \mu_s). \tag{9}$$

Differentiating gives $L'(u) = 2(u - \Delta - \mu_t + \mu_s)$ and $L''(u) = 2$, hence the unique minimizer is $u^* = \Delta + \mu_t - \mu_s$. If the two state distributions are shifted copies, their standard deviations are equal and independent of u. Adding standard-deviation matching does not remove this bias.

Local quadratic model Around a fixed correction θ^*, write the weighted residual as $r(\theta^* + \delta) \simeq b + J\delta$, where b collects state mismatch and image-model error, and let $q(\theta) = \mathrm{vec}(\bar{H}(\theta) - I_3)$ with $q_0 = q(\theta^*)$ and $G = Dq(\theta^*)$. The local quadratic surrogate of Eq. (4) is

$$\min_{\delta} \ \tfrac{1}{2}\|b + J\delta\|^2 + \lambda\|q_0 + G\delta\|^2. \tag{10}$$

Proposition 1: explicit solution Assume a differentiable residual with fixed J of full column rank and $\sigma_{\min}(J) = s > 0$. Let $b = b_0 + \varepsilon$, where independent source and target samples have covariances Σ_s and Σ_t in the residual coordinates, so that $\mathbb{E}\varepsilon = 0$ and $C_\varepsilon = \mathrm{Cov}(\varepsilon) = \Sigma_s/N_s + \Sigma_t/N_t$. With $A = J^\top J + 2\lambda G^\top G$ and $d_0 = -A^{-1}(J^\top b_0 + 2\lambda G^\top q_0)$, the minimizer satisfies

$$\hat{\delta} = -A^{-1}(J^\top b + 2\lambda G^\top q_0), \tag{11}$$

$$\mathbb{E}\|\hat{\delta}\|^2 \le \|d_0\|^2 + \frac{1}{s^2}\left(\frac{\operatorname{tr}\Sigma_s}{N_s} + \frac{\operatorname{tr}\Sigma_t}{N_t}\right), \qquad \|d_0\| \le \frac{\|b_0\|}{s} + \frac{2\lambda\|G^\top q_0\|}{s^2}. \tag{12}$$

Proof of Proposition 1 Work in fixed weighted residual coordinates, absorbing the interior mask, channel scaling, and quadratic loss weights into the features. Independence within and between reference sets gives the stated C_ε for the difference of their sample means. Differentiating Eq. (10) yields $A\hat{\delta} = -J^\top b - 2\lambda G^\top q_0$. Since $J^\top J \succeq s^2 I$ and $\lambda \ge 0$, A is positive definite. Put $B = A^{-1}J^\top$. Then $\hat{\delta} = d_0 - B\varepsilon$, and the zero-mean cross term vanishes:

$$\mathbb{E}\|\hat{\delta}\|^2 = \|d_0\|^2 + \operatorname{tr}(BC_\varepsilon B^\top). \tag{13}$$

Moreover, $BB^\top = A^{-1}J^\top JA^{-1} \preceq A^{-1} \preceq s^{-2}I$, so $\|B\|_2 \le s^{-1}$ and $\operatorname{tr}(BC_\varepsilon B^\top) \le s^{-2}\operatorname{tr} C_\varepsilon$. Substituting gives Eq. (12). The deterministic term also satisfies

$$\|d_0\| \le \frac{\|b_0\|}{s} + \frac{2\lambda\|G^\top q_0\|}{s^2}. \tag{14}$$

This separates reference/model bias from regularization bias.

For a continuously differentiable residual with full-column-rank Jacobian, a nonsingular six-row minor and the inverse function theorem imply local identifiability of the image-transform parameters. This is a property of the residual map, not of physical camera pose.

Depth-dependent residual For two pinhole cameras with the same intrinsics and orientation, separated laterally by t_x, a point (X, Y, Z) has horizontal coordinates $x_0 = f_{\mathrm{px}}X/Z$ and $x_c = f_{\mathrm{px}}(X - t_x)/Z$. The sampling displacement needed to recover the nominal coordinate is $-f_{\mathrm{px}}t_x/Z$. Using the displacement for a frontoparallel plane at Z_0 leaves magnitude $f_{\mathrm{px}}|t_x||Z^{-1} - Z_0^{-1}|$. This holds on jointly visible points. If $|Z - Z_0| \le h < Z_0$, the residual is at most $f_{\mathrm{px}}|t_x|h/[Z_0(Z_0 - h)]$. Pure rotation induces a depth-independent homography. A fixed camera transform provides a shared correction across scene states, while changing depths and visibility govern the remaining error.

Corollary 1 (region-weighted alignment) For $r_k(\theta) \simeq J_k(\theta - \theta_k^*)$, $\lambda = 0$, and a full-rank stacked Jacobian, the minimizer of the local squared residual is the matrix-weighted average in Eq. (6).

Proof of Corollary 1 With $\lambda = 0$ the surrogate is $\sum_k \|J_k(\theta - \theta_k^*)\|^2$, whose gradient vanishes at $\sum_k J_k^\top J_k(\theta - \theta_k^*) = 0$; solving for θ gives Eq. (6) whenever $\sum_k J_k^\top J_k$ is invertible, which holds under the full-rank assumption of Proposition 1. With $\lambda > 0$, the regularization term adds $2\lambda G^\top G$ to the weight sum and pulls the average toward the identity, as in Eq. (11).

Scope of the local model Proposition 1 fixes normalization, J, G, and q_0 within a local approximation. In the full estimator these quantities also depend on the reference samples. Stable visibility, full rank, and independent sampling are assumptions of the analysis; systematic scene mismatch remains in b_0.

C.2 Spatial pooling and geometric sensitivity

Let $f : \Omega \to \mathbb{R}^C$ be a differentiable feature field on a periodic image domain, and let $T_u f(v) = f(v - u)$ be a translation. Change of variables gives

$$\frac{1}{|\Omega|}\int_\Omega T_u f(v)\, dv = \frac{1}{|\Omega|}\int_\Omega f(v)\, dv. \tag{15}$$

Its derivative with respect to u is zero, whereas the unpooled field has derivative $-\nabla f(v)$. For translation matching, the spatial normal matrix is consequently $\int_\Omega \nabla f(v)^\top \nabla f(v)\, dv$; it is positive

definite when the gradients span both translation directions. For a general warp, the image gradients compose with the warp Jacobian, and identifiability requires that the resulting measurements span the parameter directions. Repeating a spatially uniform patch provides no such constraint. A finite crop introduces boundary terms and real encoders are only approximately equivariant, so pooled features can still respond to motion. The example identifies the spatial information that pooling can remove; the pooled-feature controls test its importance in the actual policies.

C.3 From residual images to actions and trajectories

All vector norms below are Euclidean and matrix norms are induced operator norms. Fix the scene history, nonvisual inputs, policy memory, and random draws used by a stochastic policy. Stack its nominal visual inputs in z; the warped inputs are $z + e$. For policies using an observation history, z includes every input frame. The output may be an action or a predicted action chunk, with the same output convention in both calls.

Proof of Proposition 2 Assume that $\pi(\cdot, p)$ is differentiable on a neighborhood of the segment from z to $z + e$, with an L-Lipschitz Jacobian there. Writing $A = D_z\pi(z, p)$, the fundamental theorem of calculus yields

$$\pi(z + e, p) - \pi(z, p) = Ae + R(e), \tag{16}$$

$$R(e) = \int_0^1 \big[D_z\pi(z + \tau e, p) - A\big]e \, d\tau, \tag{17}$$

$$\|R(e)\| \le \int_0^1 L\tau\|e\|^2 \, d\tau = \tfrac{L}{2}\|e\|^2. \tag{18}$$

Since $\|Ae\| = \sqrt{e^\top A^\top Ae}$, the triangle inequality proves Eq. (7). The reverse triangle inequality also gives

$$\Big|\|\pi(z + e, p) - \pi(z, p)\| - \sqrt{e^\top Qe}\Big| \le \tfrac{L}{2}\|e\|^2, \qquad Q = A^\top A. \tag{19}$$

Thus two image corrections can be ranked by their action error whenever the gap between their leading terms exceeds their two remainder bounds. For e in the null space of A, action error is second order even though pixel error is first order. This conclusion applies within the region of the Lipschitz assumption. ReView fits feature agreement; Q explains how the resulting image error affects actions.

A two-region example. Consider scalar image displacements d_r and d_b on robot and background regions, and a shared correction $u_\alpha = \alpha d_r + (1 - \alpha)d_b$, $0 \le \alpha \le 1$. For a locally linear policy whose action depends on robot position as $a = kx_r$ with other inputs fixed, the action discrepancy is

$$|\Delta a_\alpha| = |k|(1 - \alpha)|d_r - d_b|, \qquad |e_b| = \alpha|d_r - d_b|. \tag{20}$$

At $\alpha = 0$ the background is exact. Increasing α worsens its alignment while improving the action, and $\alpha = 1$ recovers the nominal action despite a displaced background. This example makes the distinction between pixel and action fidelity explicit. Which image directions matter for a real policy is determined by its full Q, including cross-region terms and dependence on nonvisual inputs.

Effect of calibration error on actions The two local models can also be composed. Use $B = A_{\mathrm{cal}}^{-1}J^\top$ for the calibration solution in Proposition 1, where $A_{\mathrm{cal}} = J^\top J + 2\lambda G^\top G$, so that $\hat\delta = d_0 - B\varepsilon$. At a fixed evaluation scene history, write the local action discrepancy as $a(\theta^* + \delta) - a^0 = \eta^* + P\delta$. Here η^* is the action error at the reference correction and P is the derivative of the composed warp and policy. For these fixed affine models, with $C_\varepsilon = \Sigma_s/N_s + \Sigma_t/N_t$,

$$\mathbb{E}\|a(\theta^* + \hat\delta) - a^0\|^2 = \|\eta^* + Pd_0\|^2 + \mathrm{tr}(PBC_\varepsilon B^\top P^\top). \tag{21}$$

To prove this, substitute $\hat\delta$ and use $\mathbb{E}\varepsilon = 0$ to eliminate the cross term. The expression shows why warp error alone cannot determine action error: calibration uncertainty is filtered by P, and the reference correction itself may leave a task-relevant residual η^*. The nonlinear image sampler and policy add Taylor remainders outside this affine model.

Propagation through the closed loop Let x be an augmented state containing the physical history and policy memory, including any action queue, needed to make the closed-loop system Markov. Couple policy and environment randomness between the two systems. Let $f_t(x, a)$ denote the augmented transition and $F_t^0(x) = f_t(x, \pi_0(x))$ the nominal-camera closed-loop map. Suppose, on a tube around a nominal rollout, that F_t^0 is ρ-Lipschitz, f_t is L_a-Lipschitz in action, and $\|\pi_\theta(x) - \pi_0(x)\| \le \epsilon$ uniformly. The two policies here are evaluated at the same augmented state with displaced-and-warped or nominal visual histories. Starting from the same x_0, their trajectories satisfy

$$\|x_t^\theta - x_t^0\| \le L_a \epsilon \sum_{j=0}^{t-1} \rho^j. \tag{22}$$

Indeed, adding and subtracting $F_t^0(x_t^\theta)$ bounds the next deviation by $L_a\epsilon + \rho\|x_t^\theta - x_t^0\|$; induction gives the result. If the bound stays within the tube, it also ensures that the assumptions continue to apply. For $\rho < 1$ the deviation is at most $L_a\epsilon/(1-\rho)$; for $\rho \ge 1$ it may grow with the horizon. A sufficient condition for preserving success is that this bound remain below a trajectory margin that preserves task constraints and terminal success. Contact transitions and narrow insertion tolerances can make that margin small. We do not estimate these constants or margins in the experiments; the relation specifies how action recovery can carry through to a rollout and why calibration quality alone is insufficient.

C.4 Local focal-length relation

For $K = \mathrm{diag}(f, f, 1)$ and the fixed normal $n = e_3$, expand Eq. (1) near identity and divide by H_{33}. To first order,

$$\bar{H} \simeq \begin{pmatrix} 1 - u_z & -\omega_z & f(\omega_y + u_x) \\ \omega_z & 1 - u_z & f(u_y - \omega_x) \\ -\omega_y/f & \omega_x/f & 1 \end{pmatrix}. \tag{23}$$

For any other positive focal f', set

$$\omega_x' = (f'/f)\omega_x, \qquad \omega_y' = (f'/f)\omega_y, \qquad \omega_z' = \omega_z, \tag{24}$$

$$u_x' = (f/f')(\omega_y + u_x) - \omega_y', \tag{25}$$

$$u_y' = (f/f')(u_y - \omega_x) + \omega_x', \qquad u_z' = u_z. \tag{26}$$

Substitution preserves each entry to first order: the families share a local tangent space, though finite motions and optimizer conditioning can differ. This compensation concerns focal scale with a fixed principal point, square pixels, and known crop coordinates; it does not recover the physical camera motion.

D Geometric alignment

D.1 Ground-truth table-plane homography

The simulator exposes the nominal and displaced camera poses and intrinsics. For each task and camera pose, we form the homography induced by the task's support plane between the two cameras, in the same direction as Eq. (2): a pixel of the nominal crop is mapped to its sampling location in the displaced sensor frame. Each H_{GT} is checked by projecting 49 table-plane points through both cameras, with agreement within 10^{-8} px; the identity pose gives the identity homography. H_{GT} aligns the table exactly but only approximates other depths.

Warp errors are measured as the RMS pixel displacement between two sampling grids over the 76×76 policy crop, $e(H_1, H_2) = \mathrm{RMS}_q \, \|\,\mathrm{proj}(H_1 q) - \mathrm{proj}(H_2 q)\|_2$. The no-adaptation error is $e(I_3, H_{\mathrm{GT}})$. To separate the expressiveness of the six-parameter family from fitting error, H_{GT} is projected onto the family by minimizing $e(H(\theta), H_{\mathrm{GT}})$ over θ from five starts; the projection error is the approximation error of the six-parameter family and $e(H_{\mathrm{fit}}, H(\theta_{\mathrm{proj}}))$ the fitting error.

D.2 Residual Jacobians and the local error scale

For the spatial and late-moment objectives we compute the residual b and the Jacobian J of the weighted residual with respect to θ at the fitted warp by central finite differences (steps 10^{-3} and

5×10^{-4}; both give the same spectrum to within 10%). To make the smallest singular value comparable across objectives and parameters, $J^\top J$ is whitened by the Gram matrix of the sampling-grid Jacobian, so that singular values are residual change per unit RMS pixel displacement and $\|b\|/s$ is in pixels. Table 11 lists the six task–camera combinations for Square/Can at L2 in Appendix B.6. The ratio exceeds the measured error in 21 of the 24 entries and falls short by at most 0.6 px in the others, and it preserves both orderings, across signals and across the same-state intervention. Pairing lowers $\|b\|$ for late moments by roughly two to five times while s changes by less than a factor of two; the spatial objective has a much larger s. The six-parameter approximation error relative to

Table 11: Local error scale $\|b\|/s$, measured error against H_{GT}, and geometric sensitivity s at the fitted warp (px), for Square and Can at L2 with three cameras each. Paired variants use nominal renders at the four target reset states as source frames.

		Spatial (ReView)			Spatial, paired			Late, unpaired			Late, paired		
Task	Cam.	$\|b\|/s$	err.	s	$\|b\|/s$	err.	s	$\|b\|/s$	err.	s	$\|b\|/s$	err.	s
Square	15	4.1	1.41	16.0	4.1	1.20	14.6	10.0	4.72	0.22	4.8	2.99	0.17
	16	5.4	3.03	17.9	5.0	2.28	18.8	17.8	7.68	0.13	9.2	7.27	0.13
	17	6.1	2.05	12.9	6.9	2.44	11.0	16.6	6.05	0.15	8.9	2.62	0.14
Can	15	1.9	0.79	28.7	1.2	0.78	28.4	9.8	9.33	0.14	1.6	2.19	0.18
	16	2.3	1.54	26.9	1.7	1.30	27.7	12.6	11.65	0.09	2.7	3.29	0.17
	17	2.1	1.52	29.2	1.6	1.01	27.8	11.0	6.73	0.11	3.2	3.38	0.12
Median		3.2	1.53		2.9	1.25		11.8	7.21		4.0	3.14	

the table-plane homography is 0.55–1.31 px on these conditions (no-adaptation error 8.8–16.8 px), while ReView's total error is 0.8–3.0 px. The former measures the restriction of the warp family even for table-plane alignment; it is distinct from the depth-dependent residual elsewhere in the scene. All geometry uses the camera poses recorded during the evaluated rollouts.

D.3 Main evaluation

This analysis reuses the fitted main-evaluation warps and recorded outcomes; it does not add policy rollouts. In Figure 12a, boxes show the median and interquartile range over the 60 DP task–level–camera conditions, whiskers span the 5th–95th percentiles, and points show individual conditions. Panel b reports median pairwise warp distances over the same conditions. Errors and distances are root-mean-square sampling displacements on the policy crop, in pixels. Spatial pixel and feature matching yield similar warps, with a median distance of 0.56 px; their errors against the table-plane homography are 1.5–1.8 px, versus 7.3 px with pooling.

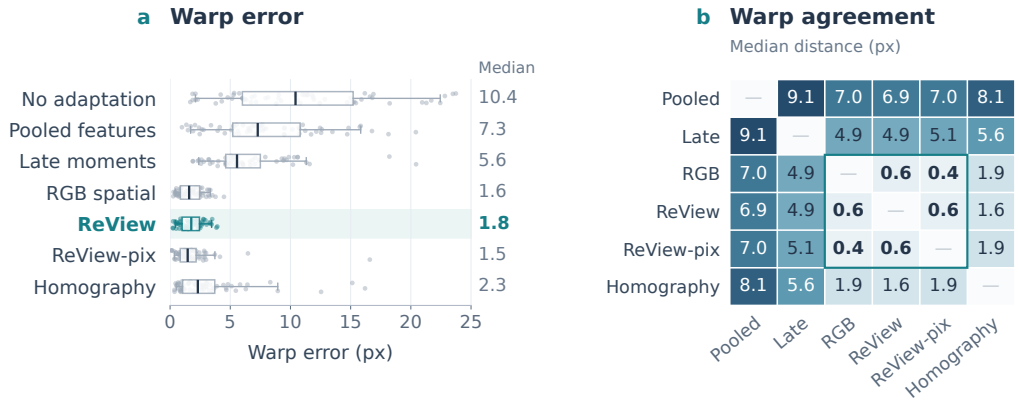

Figure 12: Geometry of the fitted DP warps across 60 task–level–camera conditions. (a) Error against the true table-plane homography. (b) Median pairwise warp distances; the outlined block groups spatial RGB, ReView, and ReView-pix. All distances are in pixels. Homography uses ReView's objective with eight free parameters.

For all 60 task–level–camera combinations in the main robomimic DP evaluation, the homography passes the direct-projection check on table-plane points, using the camera poses recorded during rollouts. Table 12 gives the statistics underlying Figure 12a,b. The six-parameter warp approximates the true table-plane homography to a median 0.73 px (IQR 0.47–1.01); fitting errors against this projection are 1.49, 1.66, and 1.26 px for RGB, ReView, and ReView-pix, versus 5.46 for late moments and 7.13 for pooled features. On ACT, ReView is 1.25 px from the true homography; the three spatial warps agree to 0.48–0.49 px pairwise, while pooled features depart by 4.9–5.0 px. Across seven input variants and 60 DP evaluation conditions (420 comparisons), warp error and success have Spearman $\rho = -0.50$ after removing task and level means. Within ReView, the task-wise correlations are -0.81 on Square, -0.48 on Lift, and -0.30 on Can. These associations complement the same-state intervention rather than establish a control guarantee.

Oracle projection. The true table-plane homographies use camera poses verified against the recorded evaluations. All are valid. Their complete per-level success rates are in Appendix B.2. Projecting these homographies onto the six-parameter warp family gives 56.2% on DP and 80.4% on ACT, compared with 53.7/77.9% for the exact table-plane oracle and 59.5/80.7% for ReView. The table-plane reference therefore does not supply a control ceiling. The Square/Can diagnostic in Table 9 reports the projected oracle alongside paired references and depth reprojection.

Table 12: Fitted warps against the ground-truth table-plane homography on the 60 task–level–camera combinations of the robomimic DP evaluation. Errors are RMS sampling displacements over the policy crop. Displacement measures the change from identity; the fraction removed is $1 - e/e_{\text{no adapt.}}$. Both are reported as medians over conditions. Success rates use 1,500 trials. The eight-parameter homography is fitted with ReView's spatial objective.

Signal	Error, median [IQR] (px)	Displacement (px)	Fraction removed	Success
No adaptation	10.42 [6.01, 15.21]	0.00	0.00	31.6%
Pooled features	7.28 [5.20, 10.80]	10.95	0.27	32.5%
Late moments	5.56 [4.62, 7.50]	10.55	0.37	49.9%
RGB spatial	1.58 [0.85, 2.45]	10.01	0.85	57.6%
ReView	1.76 [1.00, 2.44]	9.54	0.85	59.5%
ReView-pix	1.46 [0.84, 2.16]	9.93	0.87	58.7%
8-param. homography	2.31 [1.04, 3.71]	11.02	0.80	63.3%

D.4 Action agreement under paired observations

Table 13 compares nominal-view DP actions with predictions from corrected images on Square and Can at L2, using the fitted ReView, ReView-pix, late-moment, and table-plane warps. For three cameras and five initial states, we collect trajectories with either displaced-view or nominal-view images. At each comparison, all methods share the two-frame scene history, nonvisual inputs, and four diffusion-noise draws. Alternative predictions are evaluated without executing them or refitting the warp.

We compare predictions at eight replanning steps from 0 through 224. Action error is the root-mean-square error (RMSE) over the eight-action, seven-coordinate output chunk in the policy's normalized action space, relative to its nominal-view output under the same noise. RGB error compares the corresponding policy-input crops on the $[0, 1]$ intensity scale. We use the 405 of 480 sampled time points preceding the first task success, averaging within episodes, then equally over initializations and cameras. Shared initializations and noise draws are repeated measures.

Table 13: Agreement with nominal-view DP outputs on matched histories. The top block reports normalized action RMSE; the bottom block reports RGB RMSE in the policy crop. Lower values indicate closer agreement.

Task	Rollout input	No adapt.	ReView	ReView-pix	Late	Oracle
Action RMSE						
Square	Displaced view	0.463	0.299	0.295	0.246	0.284
	Nominal view	0.356	0.183	0.183	0.214	0.158
Can	Displaced view	0.396	0.118	0.115	0.213	0.140
	Nominal view	0.295	0.088	0.088	0.249	0.124
RGB RMSE						
Square	Displaced view	0.316	0.165	0.165	0.251	0.151
	Nominal view	0.321	0.172	0.171	0.259	0.160
Can	Displaced view	0.234	0.084	0.077	0.199	0.089
	Nominal view	0.224	0.079	0.072	0.189	0.084

ReView reduces action error in all 12 task–camera–rollout comparisons: by 70.0–70.1% on Can and 35.5–48.5% on Square in aggregate. ReView-pix is comparably effective. The table-plane warp gives closer actions on Square, whereas ReView is closer on Can. On Square's displaced-view trajectories, late moments give larger RGB error than ReView (0.251 versus 0.165) but smaller action error (0.246 versus 0.299). Thus image and action errors can rank corrections differently.

Including all 480 sampled time points preserves ReView's improvement over no adaptation in every condition and the Square late-moment counterexample. The corresponding reductions are 62.0–71.0% on Can and 35.5–53.2% on Square. The reference is the frozen policy at the nominal view, rather than expert actions; this experiment measures action agreement, not task success.

D.5 Regional alignment on ManiSkill

We measure alignment against dense correspondences obtained from simulator depth: each nominal-camera pixel is back-projected into 3D and projected into the displaced camera. Segmentation separates robot links, the table including its goal marker, and the cube. Table 15 reports median errors over three cameras.

The cube is only 2 cm above the table, but the gripper is about 16 cm above it. Table-plane alignment therefore preserves the cube while displacing the robot. ReView's best-aligned plane lies 3–8 cm above the table, trading table accuracy for robot alignment; RGB-only DP and ACT recover more success with this compromise. Moving the plane to gripper height instead misaligns the cube and lowers DP success to 52.5% on PullCube and 19.9% on PushCube, averaged over L0–L3 with 1,000 trials per task. RollBall shows the same pattern with less parallax: at L2, robot error is 3.6 px for the table-plane oracle and 1.1 px for ReView; overall success is 39.3% and 48.6%, respectively.

Regional weighting On PullCube and PushCube (L1–L3, three cameras per level), we accumulate regional $J_k^\top J_k$ from the feature or pixel residual and fit each region's warp to dense correspondences. Their weighted prediction, including the regularization term, lies 1.2–2.6 px from ReView, versus 2.8–6.2 px for the table plane and 3.3–7.6 px for the gripper plane (Table 14). Features give the robot more weight than pixels do, consistent with Corollary 1. Pixel-correlation predictions are less accurate (2.5–5.0 px). The quadratic model approximates smooth-L1 or correlation fitting: about 9% of curvature falls outside the measured regions, and automatic and finite-difference gradients differ by 10–46%.

In Table 14, curvature share is each region's fraction of $\mathrm{tr}(J^\top J)$. Distances compare the fitted warp with the local prediction, table plane, or gripper plane. Robot error uses the depth-based correspondences. Entries summarize medians over cameras and ranges over the two tasks. These warps are fitted separately from those in Table 15.

Table-plane error alone misses this trade-off. At the more distant robomimic camera, parallax is smaller and fitted warps lie within about two pixels of the table-plane correction, which still need not give the best control (Section 5.4). Figure 4 illustrates the PullCube L2 comparison; Table 15 covers L1–L3.

Table 14: Regional contributions to the fitted warp on ManiSkill PullCube and PushCube. Features give the robot more weight than pixel correlation; the local model predicts a compromise between robot and table alignment.

		Curvature share		Distance to fit (px)			Robot error (px)	
Signal	Level	Robot	Table	Pred.	Table	Robot	Fit	Pred.
ReView	L1	0.27–0.32	0.57–0.60	1.2–1.3	2.8–3.1	3.3–3.5	0.9–1.0	0.9
	L2	0.32–0.35	0.55–0.59	1.4–1.9	4.3–4.6	4.9–5.1	1.4	1.3–1.5
	L3	0.33–0.36	0.52–0.57	2.5–2.6	5.6–6.2	7.1–7.6	2.0–2.2	1.7–1.8
ReView-pix	L1	0.20	0.44–0.45	2.5–2.6	3.6–3.7	3.1–3.2	2.3–2.5	0.8–0.9
	L2	0.21–0.24	0.41–0.47	3.0	4.0–4.1	4.8–4.9	3.1–3.2	1.3
	L3	0.19–0.21	0.46–0.48	4.2–5.0	5.2–5.9	6.8	4.3–4.4	1.8

Table 15: Alignment error by region on ManiSkill PullCube: median RMS pixel distance from depth-based correspondences. A warp that aligns the table can leave substantial error on the robot.

Level	Warp	Robot	Table	Cube	Scene
L1	No adapt.	9.2	8.9	8.1	9.0
	Oracle (table plane)	4.5	0.3	0.5	2.0
	ReView-pix	2.5	2.3	2.1	2.5
	ReView	0.8	1.3	0.4	1.3
L2	No adapt.	13.5	12.9	11.4	13.0
	Oracle (table plane)	7.0	0.3	0.7	3.1
	Gripper-plane oracle	5.9	4.1	3.2	4.5
	ReView-pix	3.2	2.1	1.3	2.3
	ReView	1.4	2.1	0.7	2.3
L3	No adapt.	18.7	18.3	16.1	18.3
	Oracle (table plane)	9.9	0.3	0.9	4.4
	ReView-pix	4.4	3.0	1.7	3.3
	ReView	2.0	3.3	0.5	3.1

E GEOMETRY AND RECOVERY LIMITS

E.1 PHYSICAL DRIFT COMPONENTS AND ORBIT

We test DP on additional camera poses under isolated physical drift components, severe mixed drift, and camera orbit. Rotation-only L2 sets translation to zero; translation-only L2 sets rotation to zero. Mixed L4/L5 use independently sampled $[10, 14)/[14, 18)$ cm and degrees. These interventions change the camera, unlike the warp parameter restrictions above. Each entry uses five cameras with 25 paired initializations (125 trials per method), distinct from the main evaluation. One Lift L5 calibration fails; all 25 trials use the unadapted policy. Pose+Depth is a privileged reprojection diagnostic unavailable to ReView. True orbit rotates the entire camera pose around the world vertical axis through $(0, 0, 0.85)$ m. Each angle uses two prescribed signed directions and 25 initializations per direction; the camera center moves. The 50 trials per method therefore cover two viewpoints, not 50 independent viewpoints.

On these additional camera poses, mean ReView success is 42.4% at L4 and 28.8% at L5, compared with 28.3% and 16.5% in the registration comparison (Figure 5). Both evaluations show limited recovery under severe drift, with Square below 7% at both levels.

The no-adaptation and ReView comparisons across these conditions contain 4,200 evaluation outcomes, including 25 no-adaptation outcomes reused after one failed Lift L5 calibration. Rotation-only recovery is positive on all three task aggregates; translation-only harms Can. Orbit retains negative transfer on Can at $5°$ and Lift at $5°/10°$.

Pose+Depth uses known poses and per-step source-view depth to sample target RGB, with border padding and no occlusion completion. Its measured Square/Can subset contains 1,150 trials; source RGB is excluded from its policy input. On this subset, no adaptation, ReView, and Pose+Depth

Table 16: DP success rates (%) on additional camera poses under physical displacement and orbit. Pose+Depth is a privileged reprojection control. Dashes indicate unmeasured conditions.

Task	Condition	No adapt.	ReView	Pose+Depth
Camera displacement				
Square	Rotation-only L2	14.4	56.8	–
	Translation-only L2	18.4	43.2	53.6
	Mixed L4	0.0	6.4	19.2
	Mixed L5	0.0	2.4	12.0
Lift	Rotation-only L2	49.6	98.4	–
	Translation-only L2	80.8	93.6	–
	Mixed L4	1.6	74.4	–
	Mixed L5	0.8	57.6	–
Can	Rotation-only L2	26.4	83.2	–
	Translation-only L2	86.4	80.8	88.0
	Mixed L4	0.8	46.4	60.0
	Mixed L5	0.0	26.4	32.0
Camera orbit (angle in degrees)				
Square	5	30.0	54.0	60.0
	10	6.0	20.0	18.0
	15	0.0	2.0	8.0
	20	0.0	0.0	12.0
Lift	5	98.0	96.0	–
	10	98.0	94.0	–
	15	88.0	92.0	–
	20	74.0	82.0	–
Can	5	76.0	68.0	74.0
	10	12.0	40.0	16.0
	15	8.0	12.0	8.0
	20	4.0	4.0	14.0

succeed in 200, 357, and 436 trials, respectively. Pose and depth alone therefore do not restore reliable control under these camera changes.

F REQUIREMENTS OF RELATED APPROACHES

Table 17 lists what each approach needs in order to handle a new camera. The columns refer to deployment under the displaced camera unless stated otherwise. Classical registration, including SIFT with RANSAC (Fischler & Bolles, 1981), has the same requirements as ReView; Sections 5.3 and 5.5 compare them on the same images.

Table 17: Requirements under camera drift. Retrain: the policy is trained or modified for the method before deployment. Extr.: camera extrinsics of the displaced camera. 3D: depth, point clouds, or a learned 3D or view-synthesis model at deployment. Actions: actions or transitions recorded under the displaced camera. Update: network weights or normalization statistics change at test time.

Approach		Retrain	Extr.	3D	Actions	Update
Training-time robustness						
Domain randomization	(Tobin et al., 2017)	✓	–	–	–	–
RoVi-Aug	(Chen et al., 2025)	✓	–	–	–	–
Camera-conditioned BC	(Jiang et al., 2026)	✓	✓	–	–	–
CamVLA	(Li et al., 2026b)	✓†	–	–	–	–
VistaBot	(Gu et al., 2026)	✓	–	✓	–	–
Robot-centric pointmaps	(Lee et al., 2026)	✓	✓	✓	–	–
ReMAP-DP	(Yang et al., 2026)	✓	✓	✓	–	–
View correction before a frozen policy						
AnyCamVLA	(Heo et al., 2026)	–	✓	✓	–	–
GS-VLA	(Park & Kim, 2026)	–	✓	✓	–	–
Test-time adaptation						
Tent	(Wang et al., 2021)	–	–	–	–	✓
ActMAD	(Mirza et al., 2023)	–	–	–	–	✓
AdaBN	(Li et al., 2018)	–	–	–	–	✓
PAD	(Hansen et al., 2021)	✓	–	–	✓	✓
MoVie	(Yang et al., 2023)	–	–	–	✓	✓
One-shot VLA adaptation	(Li et al., 2026a)	–	–	–	✓	✓
Image registration						
ECC	(Evangelidis & Psarakis, 2008)	–	–	–	–	–
SIFT with RANSAC	(Lowe, 2004)	–	–	–	–	–
ReView (ours)		–	–	–	–	–

†Ground-truth camera poses are used as supervision during training.