

Phase-Conditioned Affordance Fields for $SE(3)$ -Equivariant Visuomotor Policies

Xinzhe Chen, Liqi Huang, Sihua Ren, Haowen Sun
Mingyang Li, Zeyang Liu, Xingyu Chen*, Xuguang Lan*

National Key Laboratory of Human-Machine Hybrid Augmented Intelligence
Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University, Xi'an, China
Project page: <https://paf-policy.github.io/>

Abstract: $SE(3)$ -equivariance provides a useful geometric inductive bias for visuomotor policies, encoding that actions covary with rigid transformations of the observation. However, existing $SE(3)$ -equivariant visuomotor policies build this bias on a global equivariance assumption that treats the entire observation as governed by a single rigid transformation, which breaks down in real-world manipulation tasks where the task-relevant region changes across different objects. We address this by learning a phase-conditioned affordance field (PAF) over the observed point cloud, a per-point score that the $SE(3)$ -equivariant policy uses in two complementary ways. Its score-weighted centroid serves as a learned, phase-conditioned anchor that canonicalizes the observation and the action around the active region, replacing the fixed gripper-frame reference, and the score itself is appended as a per-point feature. This introduces a local geometric inductive bias, while preserving the policy’s global $SE(3)$ equivariance. We validate on multiple MimicGen tasks under varying spatial randomization, and PAF consistently improves the success rate of the underlying policy. On multi-object real-robot tasks, PAF outperforms baselines and keeps its lead under lighting variations.

Keywords: Robot Manipulation, $SE(3)$ Equivariance, Visuomotor Policy

1 Introduction

Real-world robotic manipulation requires policies that generalize across varying object poses and scene layouts. Visuomotor policies [1, 2] such as Diffusion Policy [3] capture diverse action distributions from expert demonstrations, but often degrade under such variations when demonstrations are scarce. Equivariant visuomotor policies introduce a geometric bias for this setting [4, 5, 6, 7, 8]. By encoding observations and decoding actions in a spherical Fourier space, Spherical Diffusion Policy (SDP) [9] makes the predicted action covary with rigid transformations of the scene, improving robustness to object pose variation.

However, SDP relies on a global equivariance assumption that treats the entire observation as governed by a single rigid transformation. This assumption breaks down in multi-object tasks where objects move independently. For example, Figure 1 shows that hanging a cup on a rack requires the policy to focus first on the cup for grasping, then on the rack for transport, and finally on the cup handle and rack pole for alignment. A single scene-level transform cannot select which geometry governs each phase. Object-level $SE(3)$ -equivariant formulations are also difficult to apply broadly, since per-object poses are often unavailable or unreliable when estimated from raw observations, and many phases involve inter-object interactions rather than independent motion.

Our key insight is to address this problem along two complementary axes, temporal and spatial. Along the temporal axis, the robot’s recent motion signals the current phase, allowing us to decom-

*Corresponding authors: xglan@mail.xjtu.edu.cn and chenxingyu1990@xjtu.edu.cn

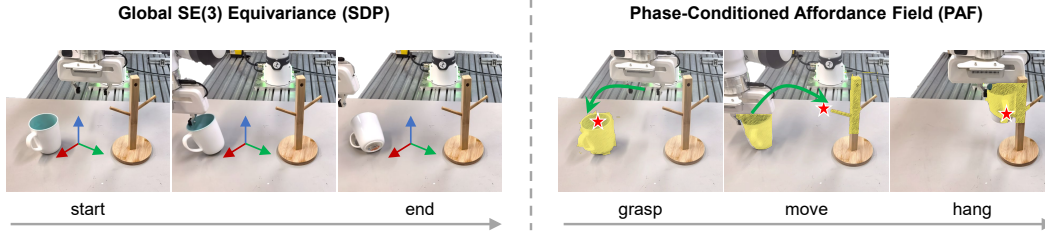


Figure 1: SDP versus PAF on the hang-cup task. Left, SDP uses a single global $SE(3)$ frame for the entire task. Right, PAF predicts affordance scores (yellow) and their centroid anchor (red star) across the grasp, move, and hang phases.

pose a multi-object task into a sequence of phases. Along the spatial axis, each phase has its own active region whose geometry should govern the action. We represent this region as an affordance field [10, 11, 12, 13], a per-point score over the observed point cloud that highlights the task-relevant geometry. Conditioning the equivariant policy on this field provides a local geometric inductive bias while retaining the policy’s global $SE(3)$ equivariance.

We instantiate this insight as Phase-Conditioned Affordance Fields (PAF). PAF encodes the recent end-effector and gripper history into a phase context, then conditions an invariant message-passing network [14, 15] on this context to score every observed point. The score-weighted centroid acts as a soft anchor that recenters the observation on the active region, as illustrated on the right of Figure 1. The score itself is also appended to the point cloud as a per-point feature. We then feed this recentered, score-augmented observation to the SDP backbone, whose $SO(3)$ -equivariant diffusion network produces actions in the active-region frame. Moreover, PAF requires no segmentation labels and is trained with self-supervised pseudo-labels derived from expert demonstrations.

We evaluate PAF on MimicGen simulation tasks [16] and on six multi-object real-robot tasks. Across twelve MimicGen settings, PAF raises the average success rate from the published SDP baseline’s 48.8% to 56.3%, with the largest gain at the d3 randomization level, where the average improves from 48.5% to 60.3%. On the real robot, PAF averages 67.0% success under normal lighting, 9.0 points above SDP, and retains a lead of 10.6 points under out-of-distribution lighting.

2 Related Work

Diffusion-based Visuomotor Policies. Visuomotor policies are commonly learned from expert demonstrations [1, 2], mapping observations to robot actions. Diffusion Policy [3] adopts a diffusion model that captures multimodal action distributions. Subsequent variants extend this paradigm with richer scene representations [17, 18, 19, 20, 21]. DP3 [22] uses point-cloud observations with a 3D encoder, exposing scene geometry directly to the diffusion policy and improving spatial reasoning. Other variants further inject affordance priors over task-relevant geometry. AnchorDP3 [23] and AffordDP [24], for instance, draw these priors from external segmentation or foundation models. These methods leave the observation-to-action transformation largely implicit, so geometric generalization depends on the pose coverage of the demonstrations.

Equivariant Visuomotor Policies. Equivariance [25, 26, 27] provides a geometric inductive bias for visuomotor policies by encoding the rigid-transformation relationship between observation and action. Early work improves sample efficiency for grasping and pick-and-place [6, 28, 29]. Recent methods combine equivariant networks with diffusion or trajectory-level policy learning [5, 8]. EquiDiff [4] applies discrete $SO(2)$ -equivariance to a voxel diffusion policy end-to-end. Spherical Diffusion Policy [9] attains continuous $SO(3)$ -equivariance through spherical Fourier features [30] and translation invariance through gripper-frame canonicalization of observations and actions, together yielding an $SE(3)$ -equivariant closed-loop policy. These methods improve manipulation generalization but tie action prediction to a fixed or global reference frame, whereas the task-relevant geometry shifts across manipulation phases.

Local Frames for Visuomotor Policies. Other work attaches the policy to a local frame. EDF [31] and Diffusion-EDFs [7] sample a target end-effector pose from an $SE(3)$ -equivariant energy or score model and hand it to a planner, so they predict goal poses rather than closed-loop actions. HEP [32] predicts a keypose translation with a high-level agent and runs a low-level trajectory policy in the resulting frame, and Oriented Affordance Frames [33] obtain the frame from external detection, segmentation, and pose estimation models and compose independently trained sub-policies in it. In both cases the frame is one predicted or detected pose, and the policy is split into levels or sub-policies. PAF builds on the diffusion backbone of SDP but derives the frame origin from a phase-conditioned affordance field, whose score-weighted centroid is the anchor and whose values condition one end-to-end closed-loop policy, so the frame origin tracks each phase’s active region without hierarchy or external perception while the policy retains global $SE(3)$ equivariance.

3 Global $SE(3)$ Equivariance

At time t , the policy observes a state $S_t = (O_t, H_t)$ consisting of a point cloud O_t and a robot history H_t . The point cloud $O_t = \{(x_i, f_i)\}_{i=1}^N$ contains N points, each with coordinate $x_i \in \mathbb{R}^3$ and non-geometric features f_i such as color. The robot history $H_t = \{(e_\tau, r_\tau, q_\tau)\}_{\tau=t-L_h+1}^t$ collects end-effector position e_τ , rotation r_τ , and gripper state q_τ over the last L_h steps. The policy outputs an action trajectory $A_t = \{a_\tau\}_{\tau=t}^{t+T_a-1}$, where each action $a_\tau = (e_\tau^a, r_\tau^a, q_\tau^a)$ specifies a translation, a rotation, and a gripper command.

A visuomotor policy π_θ is $SE(3)$ -equivariant if, for any rigid transformation $g = (R, u) \in SE(3)$, applying g to the observed state transforms the predicted action trajectory by the same g ,

$$\pi_\theta(\cdot \mid g \cdot S_t) = g \cdot \pi_\theta(\cdot \mid S_t), \quad \forall g \in SE(3). \quad (1)$$

SDP realizes this property in two steps. Positions are first canonicalized around the current end-effector, $\bar{x}_i = x_i - e_t$ and $\bar{e}_\tau^a = e_\tau^a - e_t$, which removes the translation part of g , and an $SO(3)$ -equivariant backbone then handles the remaining rotation, since a global $g = (R, u)$ maps $\bar{x}_i$ to $R\bar{x}_i$. This construction assumes that a single global g acts on the entire observed state. However, the global assumption fails when a scene contains multiple objects that move independently, since no single rigid transformation captures their combined motion. A canonicalization reference such as the gripper can absorb at most one object’s translation, since a point on object j translated by u_j becomes $\bar{x}_i + u_j - u_e$ when the reference moves by u_e , leaving residual offsets for the remaining objects, and a global $SO(3)$ -equivariant architecture can only enforce equivariance with respect to one shared scene rotation. A formal derivation is given in Appendix A.1.

4 Method

4.1 Design Principle of PAF

Localizing equivariance to the task-relevant object would, in the strictest form, require per-object $SE(3)$ equivariance, which is difficult to realize because objects interact and their relative poses couple in the observation. We instead model manipulation as a sequence of phases, each focusing on an active region that may be a single object or a multi-object interaction area. Intuitively, the predicted action should be governed by the geometry of the active region, while changes to phase-irrelevant geometry should have limited effect.

Let ζ_t denote the phase context at time t inferred from the robot history H_t , and let $\mathcal{R}_t \subseteq \{1, \dots, N\}$ index the observed points belonging to the active region in phase ζ_t . For a rigid transformation $g_{\mathcal{R}_t} \in SE(3)$ that acts only on the points indexed by $\mathcal{R}_t$, we would like

$$\pi_\theta(\cdot \mid g_{\mathcal{R}_t} \cdot S_t, \zeta_t) = g_{\mathcal{R}_t} \cdot \pi_\theta(\cdot \mid S_t, \zeta_t), \quad \forall g_{\mathcal{R}_t} \in SE(3). \quad (2)$$

Eq. (2) localizes Eq. (1) by replacing the global $g \cdot S_t$ with the region-restricted $g_{\mathcal{R}_t} \cdot S_t$. A region-restricted $g_{\mathcal{R}_t}$ alters the geometric relationships between the active region and the rest of the scene,

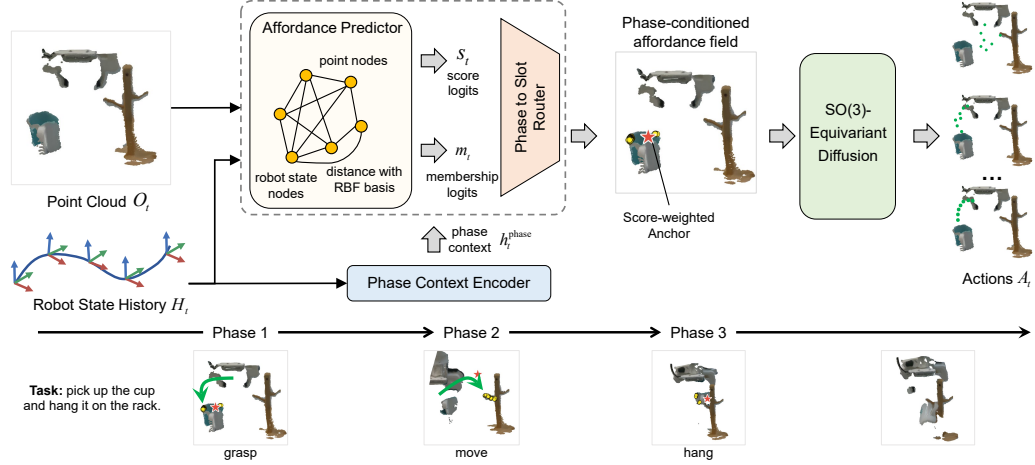


Figure 2: Overview of the PAF framework.

so PAF’s prediction does not transform cleanly under $g_{\mathcal{R}_t}$. We therefore treat Eq. (2) as a target that PAF approaches empirically through a learned, phase-conditioned geometric signal, rather than as a property it provably satisfies.

PAF approaches this goal along two complementary axes, temporal and spatial. The phase context h_t^{phase} instantiates ζ_t , summarizing the current manipulation phase from the robot history. The affordance field $\hat{s}_t$ identifies $\mathcal{R}_t$, highlighting the active-region geometry and defining a local frame origin at the score-weighted anchor c_t . Figure 2 illustrates the resulting framework. These two signals combine to give the PAF policy

$$\pi_{\theta}^{\text{PAF}}(\cdot \mid S_t, h_t^{\text{phase}}) = T_{c_t} \cdot \pi_{\phi}(\cdot \mid \tilde{O}_t, \tilde{H}_t), \quad (3)$$

where π_{ϕ} is the $SO(3)$ -equivariant SDP backbone, $\tilde{O}_t$ and $\tilde{H}_t$ are the point cloud and history canonicalized by c_t , with $\hat{s}_t$ appended to $\tilde{O}_t$ only, and T_{c_t} shifts action positions back by c_t .

4.2 Phase Context Encoding

Phase context vector. We form a phase context vector $\xi_t \in \mathbb{R}^{d_{\xi}}$ from six rigid-motion invariant statistics over the recent robot history H_t . From consecutive end-effector poses we compute the speed $v_{\tau} = \|e_{\tau} - e_{\tau-1}\|$, the gripper change $\Delta q_{\tau} = q_{\tau} - q_{\tau-1}$, and the angular displacement ω_{τ} between consecutive orientations. From triples of poses we compute the speed change $\Delta v_{\tau} = v_{\tau} - v_{\tau-1}$ and the turning angle γ_{τ} between consecutive displacement vectors. Together with the gripper state q_{τ} , these statistics form ξ_t . Full formulas are given in Appendix A.2.

Phase context encoder. A learnable phase context encoder $\Phi_{\text{phase}} : \mathbb{R}^{d_{\xi}} \rightarrow \mathbb{R}^{d_{\text{phase}}}$ maps ξ_t to the phase context $h_t^{\text{phase}} = \Phi_{\text{phase}}(\xi_t)$, which encodes phase-dependent motion patterns such as gripper interactions and transitions between manipulation phases. Because ξ_t is invariant, h_t^{phase} is also $SE(3)$ invariant, and it conditions the affordance field predictor and the phase-to-slot router.

4.3 Phase-Conditioned Affordance Field

Affordance field predictor. To determine which points belong to the active region, each point must encode its local geometry and its relationship to the end-effector under the current phase context. We build a k -nearest-neighbor (KNN) graph over point and recent end-effector nodes using Euclidean distance $d_{ij} = \|p_i - p_j\|$, where p_i is the coordinate of node i , and encode each distance with a radial basis function (RBF) basis ρ_{ij} . The phase context is injected into each node feature by a broadcast

linear projection, and node features are updated over L_g layers of invariant message passing [14, 15],

$$h_i^{\ell+1} = h_i^\ell + \psi_h \left(h_i^\ell, \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \psi_e(h_i^\ell, h_j^\ell, \rho_{ij}) \right), \quad (4)$$

where h_i^ℓ is the feature of node i at layer ℓ , $\mathcal{N}(i)$ is its KNN neighbor set, and ψ_e, ψ_h are MLP edge and node update functions. After the final layer, two MLP heads decode each point-node feature $h_i^{L_g}$ into a base score logit $s_i \in \mathbb{R}$ and slot membership logits $m_i \in \mathbb{R}^K$. Graph construction details are given in Appendix A.3.

Phase-to-slot router. Because the active region changes with the manipulation phase, we introduce K learned slots to model distinct regions and combine the per-point logits s_i, m_i with the phase context h_t^{phase} into a phase-conditioned affordance field. The slot membership logits m_i assign each point to slots via $\beta_{i,k} = \text{softmax}_k(m_{i,k})$, and a phase-to-slot router R_θ selects the active slot from the phase context, $\alpha_t = \text{softmax}(R_\theta(h_t^{\text{phase}}))$. The affordance score $\hat{s}_t(i)$ combines the base score $b_i = \sigma(s_i)$, where σ is the sigmoid function, with the slot weighting, $\hat{s}_t(i) = b_i \sum_{k=1}^K \alpha_{t,k} \beta_{i,k}$. The anchor c_t is the score-weighted centroid of the point cloud,

$$w_i = \text{softmax}_i(\log(\hat{s}_t(i) + \epsilon)), \quad c_t = \sum_{i=1}^N w_i x_i, \quad (5)$$

where x_i is the coordinate of point i , ϵ is a small constant for numerical stability, and w_i is the normalized anchor weight.

Affordance-enhanced action generation. The score $\hat{s}_t$ is appended as a per-point feature, and the anchor c_t recenters all positional inputs, $\tilde{x}_i = x_i - c_t$, $\tilde{e}_\tau = e_\tau - c_t$, $\tilde{e}_\tau^a = e_\tau^a - c_t$, while orientations, gripper states, and point features are unchanged. The augmented cloud $\tilde{O}_t = \{(\tilde{x}_i, f_i, 2\hat{s}_t(i) - 1)\}_{i=1}^N$ and canonicalized history $\tilde{H}_t = \{(\tilde{e}_\tau, r_\tau, q_\tau)\}_{\tau=t-T_o+1}^t$ over the last T_o observation steps are passed to the SDP backbone [9], whose $SO(3)$ -equivariant denoising network samples $\tilde{A}_t \sim \pi_\phi(\cdot \mid \tilde{O}_t, \tilde{H}_t)$. At inference, predicted action positions are shifted back by c_t while orientation and gripper commands are kept as predicted.

4.4 Training Objective

Affordance score. We supervise the base score logit s_i with pseudo-labels derived from the future action position y_t in the current phase, rather than object annotations. The pseudo-label assigns high values to points near the target action position and is damped near the current end-effector. The score loss $\mathcal{L}_{\text{score}}$ is binary cross entropy between s_i and the pseudo-label. The full pseudo-label formulation is given in Appendix A.5, with its scale parameters listed in Table 6.

Phase and slot supervision. A separate phase classifier head maps the phase context h_t^{phase} to sub-phase and cycle predictions, supervised by cross-entropy loss $\mathcal{L}_{\text{phase}}$ against the sub-phase label φ_t and cycle index κ_t , both derived automatically from the gripper commands and observed gripper states. The phase-to-slot router output α_t is supervised by a slot cross-entropy loss $\mathcal{L}_{\text{slot}}$ against a slot label that combines φ_t with κ_t . The per-point slot membership logits m_i receive gradient only through the DDPM [34] denoising loss $\mathcal{L}_{\text{ddpm}}$. Label definitions are given in Appendix A.5.

Overall objective. The full objective combines $\mathcal{L}_{\text{ddpm}}$ with $\mathcal{L}_{\text{score}}$, $\mathcal{L}_{\text{phase}}$, and $\mathcal{L}_{\text{slot}}$,

$$\mathcal{L} = \mathcal{L}_{\text{ddpm}} + \lambda_s \mathcal{L}_{\text{score}} + \lambda_p \mathcal{L}_{\text{phase}} + \lambda_z \mathcal{L}_{\text{slot}}, \quad (6)$$

where $\lambda_s, \lambda_p, \lambda_z$ are scalar loss weights.

4.5 Equivariance Analysis of PAF

Consider a global rigid transformation $g = (R, u) \in SE(3)$ applied to all observed point and end-effector poses. The affordance scores are invariant under g , so the score-weighted anchor c_t transforms with the scene. The following proposition states this formally.



Figure 3: The four MimicGen tasks at the d2, d3, and d4 spatial randomization levels.

Table 1: Simulation success rate in percent on four MimicGen tasks. *Equ.* denotes the equivariance group of each policy, and DiffPo denotes Diffusion Policy. Bold marks the best entry in each column. Published SDP results are shown here. See Table 5 for reproduction and seed variability.

Method	Equ.	Coffee			Three Pc.			Square			Threading			Average			All
		d2	d3	d4	d2	d3	d4	d2	d3	d4	d2	d3	d4	d2	d3	d4	
SDP	$SE(3)$	63	54	33	67	49	37	62	38	31	60	53	39	63.0	48.5	35.0	48.8
EquiDiff	$SO(2)$	65	43	29	37	15	8	39	3	3	39	20	10	45.0	20.3	12.5	25.9
DiffPo	N/A	44	23	16	4	2	2	8	0	1	17	10	8	18.3	8.8	6.8	11.3
DP3	N/A	34	20	0	0	0	0	7	0	13	12	4	1	13.3	6.0	3.5	7.6
PAF (Ours)	$SE(3)$	68	62	34	68	56	40	73	69	44	65	54	42	68.5	60.3	40.0	56.3

Proposition 1. Let $g = (R, u) \in SE(3)$ act on all observed point and end-effector poses, and let $\hat{s}'_t(i)$ and c'_t denote the affordance score and anchor computed from the transformed observation $g \cdot S_t$. Then $\hat{s}'_t(i) = \hat{s}_t(i)$ (score invariance) and $c'_t = Rc_t + u$ (anchor equivariance).

The score invariance follows because the phase features, the KNN graph, and the RBF edge encoding are invariant under a shared global rigid transformation. The anchor equivariance follows from $\sum_{i=1}^N w_i = 1$. A full proof is given in Appendix A.4.

For transformations that act only on the active region, cross-region KNN edges change the graph structure, so the anchor does not track them exactly. PAF still provides two practical benefits. First, the learned score concentrates on the active region, so the anchor is a stable reference point robust to changes in phase-irrelevant geometry. Second, the score feature provides selective attention to task-relevant geometry, making the policy more sensitive to the active region than to the rest of the scene. We verify both properties through visualization and quantitative analysis in Section 5.2.

5 Experiments

5.1 Simulation Experiments

Simulation tasks. We evaluate on four multi-object manipulation tasks from MimicGen [16, 35], built atop the Robosuite [36] simulator: Square, Coffee, Threading, and Three-Piece Assembly, each instantiated at the d2, d3, and d4 spatial randomization levels, yielding twelve task-difficulty settings. The d2 level applies $SE(2)$ randomization to the initial object poses, and d3 and d4 apply $SE(3)$ randomization, with the randomization range increasing from d2 to d4. Figure 3 shows the simulator setup for each task and randomization level. For each setting we use 100 demonstrations for training and report the mean success rate over 50 trials and 3 seeds.

Baselines and implementation. We compare against SDP [9], EquiDiff [4], DP3 [22], and Diffusion Policy [3], spanning $SE(3)$, $SO(2)$, and non-equivariant architectures across point cloud, voxel, and RGB modalities. Table 1 uses published baseline results [9] except DP3 at d4, which is our implementation. PAF uses the SDP authors’ released demonstrations for d2 and demonstrations regenerated with their official pipeline for d3 and d4. We also reproduce SDP on PAF’s exact demonstration sets under the same evaluation protocol, sharing its $SO(3)$ -equivariant diffusion backbone to isolate the affordance-field contribution. Appendix B.3 reports the reproduction and standard deviations across seeds. Baseline and hyperparameter details are in Appendices B.2 and B.4.

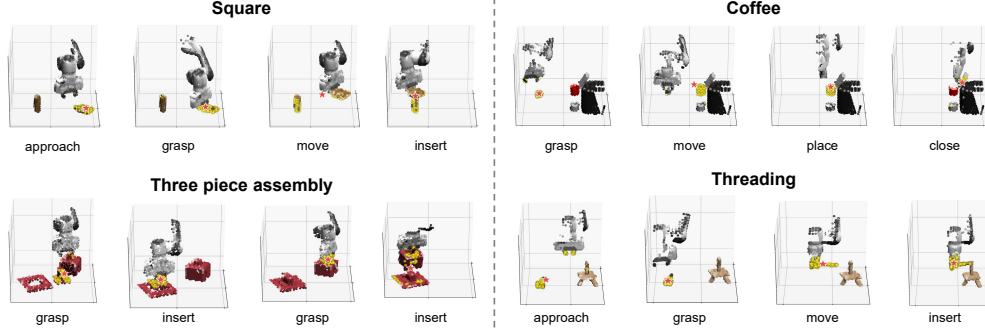


Figure 4: Affordance visualization on four MimicGen tasks.

Table 2: Component ablation on four task- Table 3: Policy selectivity, anchor stability, and difficulty settings. score change under local perturbations.

Method	Square d3	Coffee d3	Three Pc. d3	Square d4	Avg
PAF (full)	69	62	56	44	57.8
w/o anchor	43	46	40	33	40.5
w/ keypose anchor	35	51	48	32	41.5
w/ hard scope	67	51	36	40	48.5
w/o phase	60	55	52	38	51.3
w/o routing	64	54	49	40	51.8
w/o score	68	58	51	35	53.0

Perturb.	Task	Selectivity ratio $\uparrow$		Anchor	Score
		PAF	w/o anchor	disp. (mm)	MAE
Trans.	Square d3	1.41	1.29	0.05	–
	Coffee d3	1.57	1.31	0.00	–
Rot.	Square d3	1.22	1.02	0.03	2×10^{-4}
	Coffee d3	1.35	1.00	0.00	2×10^{-4}

Main results analysis. On the twelve task-difficulty settings reported in Table 1, PAF raises the average success rate of SDP from 48.8% to 56.3%, with the largest improvement at the intermediate d3 level, where the average rises from 48.5% to 60.3%. This suggests that PAF is most useful when object poses vary in full $SE(3)$ randomization but the task remains within a regime where the underlying equivariant visuomotor policy can still exploit the recovered geometric structure. The other baselines drop substantially below SDP under d3 and d4 randomization, confirming that SDP is the strongest existing comparison and the gain we report is against a strong baseline.

5.2 In-Depth Analysis

Affordance visualization. Figure 4 highlights the top-256 scored points in yellow and the anchor c_t as a red star. The field concentrates on the object that governs the current sub-phase rather than spreading over the whole scene. On Square the score stays on the nut through approach and grasp, then shifts to the peg for the move and insert sub-phases. On Coffee it stays on the capsule at grasp, hands off to the machine slot at move and place, and to the lid at close, tracking three distinct objects across four sub-phases. Across all tasks the affordance field correctly identifies the current manipulation phase from the phase context and predicts the active region accordingly, providing a local geometric inductive bias for the downstream policy.

Ablation studies. We ablate PAF on Square d3, Coffee d3, Three-Piece Assembly d3, and Square d4. Each row modifies PAF as described below. *w/o anchor* replaces c_t with the gripper position e_t . *w/ keypose anchor* regresses c_t directly to the next keypose y_t and removes the score channel, slot routing, and phase supervision. *w/ hard scope* keeps only the top-256 points by $\hat{s}_t$ instead of appending $\hat{s}_t$ as a feature. *w/o phase* removes the phase context encoder Φ_{phase} and with it the phase-to-slot router that takes the phase context as input. *w/o routing* sets the slot router α_t to uniform. *w/o score* drops the score channel from $\tilde{O}_t$.

Table 2 shows that the learned anchor is the dominant component. *w/o anchor* loses 17.3 points on average and falls below SDP on Coffee d3 and Three-Piece d3, and *w/ keypose anchor* loses 16.3 points, so a directly regressed keypose anchor, which also drops the score channel, routing,

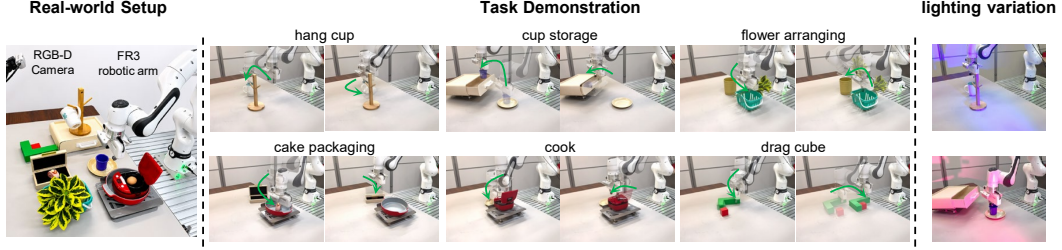


Figure 5: Real-robot setup and tasks. From left to right, a Franka Research 3 arm with an RGB-D camera, keyframes of the six tasks, and the lighting variation used to probe robustness.

Table 4: Real-robot success rate on six long-horizon tasks. N denotes normal lighting and L the out-of-distribution lighting variation. Bold marks the best entry in each column.

Method	hang cup		cup storage		flower arr.		cake pkg.		cook		drag cube		Avg	
	N	L	N	L	N	L	N	L	N	L	N	L	N	L
SDP	65	53	45	38	60	48	63	53	40	28	75	75	58.0	49.2
DP3	63	53	40	28	53	40	50	33	43	28	65	60	52.3	40.3
PAF (Ours)	78	68	55	43	68	65	70	65	48	38	83	80	67.0	59.8

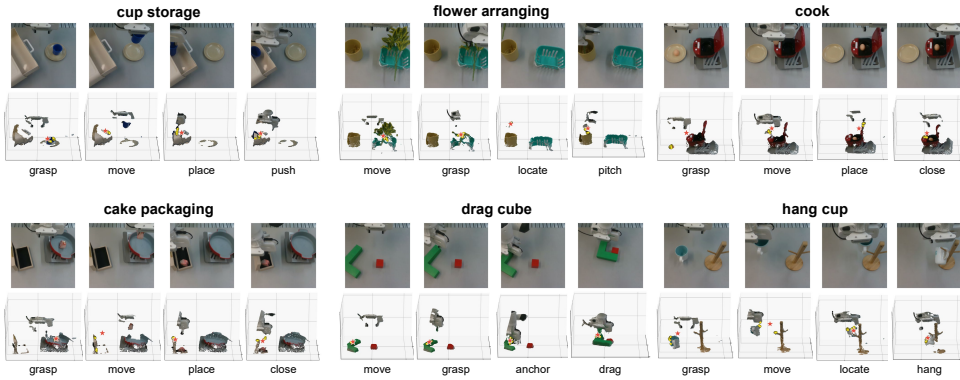


Figure 6: Phase-conditioned affordance field on the six real-robot tasks. For each task the top row shows the RGB keyframe per sub-phase and the bottom row shows the point cloud with the affordance field, where the highlighted region marks the top-256 points by predicted score.

and phase supervision, recovers none of the gain. *w/ hard scope* loses 9.3 points, concentrated on Coffee d3 and Three-Piece d3, so the policy needs the full point cloud to track several objects across phases. *w/o phase*, *w/o routing*, and *w/o score* lose 6.5, 6.0, and 4.8 points, with routing mattering more on these two settings and the phase context more on the Square settings.

Policy selectivity and anchor stability. To verify the two properties of Section 4.5, we measure how the policy responds to localized translations and rotations of the point cloud. The active subset is the top-256 points by anchor weight, and the inactive subset is low-score points at least 10 cm from the anchor. Each subset is translated by 1/3/5 cm along $\pm x/\pm y/\pm z$ or rotated about the anchor by 5/10/15 degrees about $\pm x/\pm y/\pm z$, giving 18 perturbations of each kind. We average the action change over them and the absolute change of $\hat{s}_t$ on the rotated active points.

Table 3 shows that PAF responds more selectively to the active region than the *w/o-anchor* variant. The selectivity ratio, the action change under active perturbations divided by that under inactive ones, is 1.41 vs 1.29 on Square and 1.57 vs 1.31 on Coffee under translation, and 1.22 vs 1.02 and 1.35 vs 1.00 under rotation, where the *w/o-anchor* variant no longer separates the two regions. Under inactive perturbations the anchor moves by at most 0.05 mm against translations of 30 mm

on average and by at most 0.03 mm under rotation, and under active rotations the affordance score changes by about 2×10^{-4} , so the anchor ignores phase-irrelevant geometry and the score is stable under rotation of the region it attends to. These measurements characterize selectivity and anchor stability rather than exact local equivariance.

5.3 Real-Robot Experiments

Real-world settings. We design six multi-object, long-horizon manipulation tasks, with planar or $SE(3)$ object pose variation applied to four of them across episodes. Figure 5 shows the robot platform and task examples. Each task requires the active object to switch across two or three phases of the manipulation. Point clouds are obtained by back-projection of a single RGB-D image without multi-view fusion. Per task, we train a single policy on 50 demonstrations collected under normal lighting only, and evaluate under both normal lighting (N) and an out-of-distribution (OOD) lighting variation (L) that applies colored illumination, disturbing the RGB channel without changing the geometry. For each condition we run two rollouts of 20 trials and report their mean success rate. Per-task descriptions, pose ranges, slot settings, and rollout keyframes are in Appendix C.1.

Real-robot results. Table 4 compares PAF against SDP [9] and DP3 [22], both of which share the point cloud input modality with PAF. PAF wins every task under both lighting conditions. Under normal lighting (N), PAF averages 67.0% success vs SDP 58.0% and DP3 52.3%. Under the out-of-distribution lighting variation (L), the averages become 59.8%, 49.2%, and 40.3%. PAF thus remains above both baselines under OOD lighting, and its lead over SDP on the 240 OOD-lighting trials is significant at $p = 0.02$ under a two-sided Fisher exact test. Compared to SDP, PAF rarely fails in early approach and grasp phases, and its remaining failures concentrate on later contact-rich interactions such as insertion and tool use. A side-by-side failure comparison between PAF and SDP is provided in Appendix C.2, and inference overhead is reported in Appendix C.3.

Affordance field visualization. Figure 6 shows that on the real robot the field concentrates on the object that governs the current sub-phase rather than spreading over the whole scene, mirroring the behavior observed in simulation in Figure 4. On cup storage the score focuses on the cup for *grasp*, shifts to the drawer for *move*, returns to the cup for *place*, and moves to the drawer front for *push*. On cake packaging the score tracks the cake during *grasp*, shifts to the box for *move*, and switches to the lid after *place*. On hang cup the score concentrates on the cup during *grasp* and shifts to both the cup and the rack during *move* and *locate*. The phase names are author annotations, and Appendix C.1 gives their mapping to the gripper-derived sub-phase labels.

6 Conclusion

We presented PAF, a phase-conditioned affordance field that augments an $SE(3)$ -equivariant visuomotor policy with a local geometric inductive bias. PAF scores every observed point by its relevance to the current manipulation phase, recenters the policy on the score-weighted centroid, and feeds the scores back as a feature, so the frame origin follows the active region while global $SE(3)$ equivariance is preserved. PAF raises the average success rate of SDP from 48.8% to 56.3% on twelve MimicGen settings and leads SDP and DP3 on six real-robot tasks under both lighting conditions, and the simulation ablations attribute most of the gain to the learned anchor.

7 Limitations

PAF has three limitations that point to future work. First, it provides a learned inductive bias rather than strict local $SE(3)$ -equivariance, since score invariance and anchor equivariance hold provably only under a shared global transformation. Second, the affordance field is supervised by pseudo-labels derived from future end-effector positions, which may not capture the fine geometry of contact-rich steps. Third, the slot count is currently tied to the grasp-release structure of a task. We plan to pursue tighter guarantees on local $SE(3)$ -equivariance, richer supervision signals for the affordance field, and more flexible slot formulations that adapt to diverse task structures.

Acknowledgments

This work was supported in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant No. JYB2025XDXM210, the NSFC under Grant Nos. 62125305, U23A20339, 62573339, and 62503380, the Natural Science Foundation of Shaanxi Province under Grant No. 2025SYSSYSZD-083, the State Grid Corporation of China Science and Technology Projects 52060025005B-439-ZN, the Shaanxi Provincial Key R&D Program under Grant No. 2026JC-ZDGG-17, and the CIE-Tencent Robotics X Rhino-Bird Focused Research Program.

References

- [1] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pages 158–168. PMLR, 2022.
- [2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
- [3] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.
- [4] D. Wang, S. Hart, D. Surovik, T. Kelestemur, H. Huang, H. Zhao, M. Yeatman, J. Wang, R. Walters, and R. Platt. Equivariant diffusion policy. In *Conference on Robot Learning*, pages 48–69. PMLR, 2025.
- [5] C. Tie, Y. Chen, R. Wu, B. Dong, Z. Li, C. Gao, and H. Dong. ET-SEED: EFFICIENT TRAJECTORY-LEVEL SE(3) EQUIVARIANT DIFFUSION POLICY. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [6] H. Huang, D. Wang, R. Walters, and R. Platt. Equivariant Transporter Network. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi:10.15607/RSS.2022.XVIII.007.
- [7] H. Ryu, J. Kim, H. An, J. Chang, J. Seo, T. Kim, Y. Kim, C. Hwang, J. Choi, and R. Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on se(3) for visual robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18007–18018, June 2024.
- [8] J. Yang, Z. Cao, C. Deng, R. Antonova, S. Song, and J. Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. In *Conference on Robot Learning*, pages 1048–1068. PMLR, 2025.
- [9] X. Zhu, F. Wang, R. Walters, and J. Shi. Se (3)-equivariant diffusion policy in spherical fourier space. In *International Conference on Machine Learning*, pages 80187–80206. PMLR, 2025.
- [10] R. Wu, Y. Zhao, K. Mo, Z. Guo, Y. Wang, T. Wu, Q. Fan, X. Chen, L. Guibas, and H. Dong. Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects. *arXiv preprint arXiv:2106.14440*, 2021.
- [11] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. In *European Conference on Computer Vision*, pages 222–239. Springer, 2024.
- [12] S. Nasiriany, S. Kirmani, T. Ding, L. Smith, Y. Zhu, D. Driess, D. Sadigh, and T. Xiao. Rt-affordance: Affordances are versatile intermediate representations for robot manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8249–8257. IEEE, 2025.

- [13] Y. Wang, R. Wu, K. Mo, J. Ke, Q. Fan, L. J. Guibas, and H. Dong. Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions. In *European conference on computer vision*, pages 90–107. Springer, 2022.
- [14] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. Pmlr, 2017.
- [15] V. G. Satorras, E. Hoogeboom, and M. Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.
- [16] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, pages 1820–1864. PMLR, 2023.
- [17] M. Reuss, M. Li, X. Jia, and R. Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.
- [18] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *Conference on Robot Learning*, pages 1949–1974. PMLR, 2025.
- [19] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu. Generalizable humanoid manipulation with 3d diffusion policies. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2873–2880. IEEE, 2025.
- [20] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pages 29982–30009, 2025.
- [21] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In *Conference on Robot Learning*, pages 3949–3965. PMLR, 2023.
- [22] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.
- [23] Z. Zhao, K. Fan, H.-Y. Xu, N. Qiao, B. Peng, W. Gao, D. Li, and H. Shen. Anchor3d: 3d affordance guided sparse diffusion policy for robotic manipulation. *arXiv preprint arXiv:2506.19269*, 2025.
- [24] S. Wu, Y. Zhu, Y. Huang, K. Zhu, J. Gu, J. Yu, Y. Shi, and J. Wang. Afforddp: Generalizable diffusion policy with transferable affordance. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6971–6980, 2025. doi:10.1109/CVPR52734.2025.00654.
- [25] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400, 2022. doi:10.1109/ICRA46639.2022.9812146.
- [26] F. Fuchs, D. Worrall, V. Fischer, and M. Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.
- [27] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

- [28] X. Zhu, D. Wang, O. Biza, G. Su, R. Walters, and R. Platt. Sample efficient grasp learning using equivariant models. *arXiv preprint arXiv:2202.09468*, 2022.
- [29] H. Huang, D. Wang, X. Zhu, R. Walters, and R. Platt. Edge grasp network: A graph-based $se(3)$ -invariant approach to grasp detection. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3882–3888, 2023. doi:10.1109/ICRA48891.2023.10160728.
- [30] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.
- [31] H. Ryu, H.-i. Lee, J.-H. Lee, and J. Choi. Equivariant descriptor fields: SE(3)-equivariant energy-based models for end-to-end visual robotic manipulation learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=dnjZSPGmY50>.
- [32] H. Zhao, D. Wang, Y. Zhu, X. Zhu, O. L. Howell, L. Zhao, Y. Qian, R. Walters, and R. Platt. Hierarchical equivariant policy via frame transfer. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=nAv5ketrHq>.
- [33] K. Rana, J. Abou-Chakra, S. Garg, R. Lee, I. Reid, and N. Suenderhauf. Learning from 10 demos: Generalisable and sample-efficient policy learning with oriented affordance frames. In *9th Annual Conference on Robot Learning*, 2025. URL <https://openreview.net/forum?id=1K3kj091Q1>.
- [34] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.
- [35] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *5th Annual Conference on Robot Learning*, 2021.
- [36] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, A. Maddukuri, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.
- [37] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [38] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

A Additional Method Details

A.1 Multi-Object Equivariance Derivations

Translation invariance under global canonicalization. Let $\bar{x}_i = x_i - e_t$ and $\bar{e}_\tau^a = e_\tau^a - e_t$ denote canonicalized point and action positions around the global reference e_t , typically the current end-effector position. Under a global rigid transformation $g = (R, u) \in SE(3)$, both the points and the reference translate together, so the canonicalized coordinates transform purely by rotation,

$$\bar{x}'_i = (Rx_i + u) - (Re_t + u) = R\bar{x}_i, \quad \bar{e}'_\tau^a = R\bar{e}_\tau^a, \quad (7)$$

which is then handled by $SO(2)$ - or $SO(3)$ -equivariant networks.

Independent object translation residual under global canonicalization. The same canonicalization reference e_t tracks the end-effector, which need not move with all task objects. With object-wise translations u_j on the points of object j and reference translation u_e on e_t , the canonicalized coordinate of point i on object j becomes

$$\bar{x}'_i = (x_i^j + u_j) - (e_t + u_e) = \bar{x}_i^j + (u_j - u_e). \quad (8)$$

A single global reference therefore leaves residual offsets $u_j - u_e$ whenever objects translate independently of the reference.

Rotation equivariance mismatch. After translation is factored out, a whole-scene equivariant network expects the canonicalized scene to rotate by one shared rotation, whereas object-wise scene variation gives independent rotations,

$$\bar{O}_t'^{\text{global}} = \bigcup_{j=1}^J R\bar{O}_t^j = R\bar{O}_t, \quad \bar{O}_t'^{\text{obj}} = \bigcup_{j=1}^J R_j\bar{O}_t^j \not\equiv R\bar{O}_t \quad \text{in general.} \quad (9)$$

The first case admits one well-defined group action on the whole scene. The second is not generated by any single group element, so a global rotation-equivariance law has no scene-level target transformation.

Eq. (8) shows that the fixed gripper reference leaves residual offsets $u_j - u_e$ under independent object motion. PAF instead learns a phase-conditioned reference, the anchor c_t , that recenters on whichever region governs the current phase.

A.2 Definitions of Phase Context Vector

The phase context vector ξ_t consists of six rigid-motion invariant statistics computed from the recent robot state history H_t , namely end-effector speed v_τ , speed change Δv_τ , turning angle γ_τ , relative orientation angular displacement ω_τ , gripper state q_τ , and gripper change Δq_τ . Their definitions follow. For consecutive history steps $\tau = t - L_h + 2, \dots, t$, we compute the end-effector displacement, speed magnitude, and gripper change as

$$\Delta e_\tau = e_\tau - e_{\tau-1}, \quad v_\tau = \|\Delta e_\tau\|, \quad \Delta q_\tau = q_\tau - q_{\tau-1}. \quad (10)$$

For triples of positions $\tau = t - L_h + 3, \dots, t$, we compute the speed change $\Delta v_\tau = v_\tau - v_{\tau-1}$ and the turning angle

$$\gamma_\tau = \arccos \left(\text{clip} \left(\frac{\Delta e_{\tau-1}^\top \Delta e_\tau}{\|\Delta e_{\tau-1}\| \|\Delta e_\tau\| + \epsilon}, -1 + \epsilon, 1 - \epsilon \right) \right), \quad (11)$$

where ϵ is a small numerical constant and clip guards the arccos domain against numerical overshoot and degenerate zero-displacement steps. For orientation over $\tau = t - L_h + 2, \dots, t$, in our implementation, the orientation representation r_τ is converted to a unit quaternion, and we compute the relative quaternion $\delta r_\tau = r_\tau^{-1} \otimes r_{\tau-1}$, where $\otimes$ denotes quaternion multiplication. Using r_τ here to denote the converted quaternion, let $\delta r_{\tau,w}$ and $\delta r_{\tau,xyz}$ denote the scalar and vector parts of δr_τ . The angular displacement is

$$\omega_\tau = 2 \operatorname{atan2} (\|\delta r_{\tau,xyz}\|, |\delta r_{\tau,w}|), \quad (12)$$

where $\text{atan2}(y, x)$ is the two-argument arctangent and the factor 2 converts the half-angle represented by a unit quaternion into a rotation angle. Each statistic is invariant under shared global $SE(3)$ transforms because it depends only on displacement norms, angular differences, or scalar gripper values.

A.3 Graph Construction Details

KNN graph and RBF edges. Let p_i denote the coordinate of a graph node, either a point in O_t or a recent end-effector in H_t . The neighbor set $\mathcal{N}(i)$ is obtained by k -nearest neighbors under $d_{ij} = \|p_i - p_j\|$. Each distance is encoded by an RBF basis,

$$\rho_{ij,m} = \exp\left(-\left(\frac{d_{ij} - \mu_m}{\sigma_\rho}\right)^2\right), \quad m = 1, \dots, M, \quad (13)$$

where M is the number of RBF kernels, $d_{\max}$ is the maximum encoding distance, the centers $\{\mu_m\}_{m=1}^M$ are uniformly spaced on $[0, d_{\max}]$, $\sigma_\rho = d_{\max}/(M-1)$, and $\rho_{ij} = (\rho_{ij,1}, \dots, \rho_{ij,M}) \in \mathbb{R}^M$ is the RBF edge feature.

Node initialization. Let $\tilde{h}_i^0 \in \mathbb{R}^{d_h}$ be the initial feature of node i , obtained from its non-geometric attributes (e.g., color for point nodes, gripper state for end-effector nodes). We inject the phase context by broadcast addition followed by layer normalization,

$$h_i^0 = \text{LN}\left(\tilde{h}_i^0 + P_{\text{phase}} h_t^{\text{phase}}\right), \quad (14)$$

where LN denotes layer normalization, $P_{\text{phase}} \in \mathbb{R}^{d_h \times d_{\text{phase}}}$ projects the phase context to the node hidden dimension d_h , and the same projected vector is added to all graph nodes.

Score and slot heads. After the final message-passing layer, two independent MLP heads decode each point-node feature $h_i^{L_g}$,

$$s_i = \phi_{\text{score}}(h_i^{L_g}) \in \mathbb{R}, \quad m_i = \phi_{\text{slot}}(h_i^{L_g}) \in \mathbb{R}^K, \quad (15)$$

where L_g is the number of graph layers, ϕ_{score} predicts the base score logit, and ϕ_{slot} predicts the K slot membership logits.

A.4 Proof of Score Invariance and Anchor Equivariance

We restate and prove Proposition 1 from Section 4.5 under a shared global $SE(3)$ transformation. It does not claim invariance to arbitrary object-wise transformations, which can change cross-object distances and the whole-scene graph.

Proposition 1, restated. *Let $g = (R, u) \in SE(3)$ act on all observed point and end-effector poses, and let $\hat{s}'_t(i)$ and c'_t denote the affordance score and anchor computed from the transformed observation $g \cdot S_t$. Then $\hat{s}'_t(i) = \hat{s}_t(i)$ (score invariance) and $c'_t = Rc_t + u$ (anchor equivariance).*

Proof. For any pair of graph nodes,

$$d'_{ij} = \|(Rp_i + u) - (Rp_j + u)\| = \|R(p_i - p_j)\| = \|p_i - p_j\| = d_{ij}. \quad (16)$$

Therefore the KNN graph and the RBF edge encoding are unchanged. Since each message-passing layer uses only node features and the RBF-encoded distances, the point-node features after every layer are unchanged by the global transformation.

The phase context is also invariant. End-effector speed magnitudes, speed changes, turning angles, relative quaternion angular displacements, gripper states, and gripper deltas are unchanged by applying the same rigid transform to the whole scene and robot trajectory. The phase context vector is therefore invariant,

$$\xi_t(g \cdot H_t) = \xi_t(H_t), \quad (17)$$

and the phase context $h_t^{\text{phase}} = \Phi_{\text{phase}}(\xi_t)$ is invariant as well. The score head, slot head, and phase-to-slot router are functions of invariant node features and invariant phase context. Hence

$$b'_i = b_i, \quad \beta'_{i,k} = \beta_{i,k}, \quad \alpha'_{t,k} = \alpha_{t,k}, \quad \hat{s}'_t(i) = \hat{s}_t(i). \quad (18)$$

The anchor weights are functions only of the invariant scores, so $w'_i = w_i$. The transformed anchor is therefore

$$c'_t = \sum_i w'_i (Rx_i + u) = R \sum_i w_i x_i + u \sum_i w_i = Rc_t + u, \quad (19)$$

because $\sum_i w_i = 1$. This proves score invariance and anchor equivariance under a shared global $SE(3)$ transformation. Hence the PAF policy of Eq. (3) is globally $SE(3)$ -equivariant. Under g , $\tilde{x}'_i = R\tilde{x}_i$, $\tilde{c}'_\tau = R\tilde{c}_\tau$, and $r'_\tau = Rr_\tau$, while the score channel, point features, and gripper states are unchanged, so the $SO(3)$ -equivariant backbone rotates its predicted actions by R , and shifting positions back by $c'_t = Rc_t + u$ gives Eq. (1).

A.5 Phase Label and Slot Routing

Pseudo-label formulation. The target action position y_t is the expert end-effector position at the first gripper open-close transition within the future action horizon. When no transition occurs within the horizon, y_t falls back to the end-effector position at the horizon end. The pseudo-label and end-effector repulsion are

$$\ell_i = \exp(-\|x_i - y_t\|/\tau_y), \quad \tilde{\ell}_i = \ell_i[1 - (1 - \eta) \exp(-\|x_i - e_t\|/\tau_{ee})], \quad (20)$$

where τ_y and τ_{ee} are spatial scales and $\eta \in [0, 1)$ is a minimum-scale floor, so the pseudo-label is damped near the current end-effector but never below a fraction η of ℓ_i . The score loss is binary cross entropy on logits,

$$\mathcal{L}_{\text{score}} = \frac{1}{N} \sum_{i=1}^N \text{BCE}(s_i, \tilde{\ell}_i). \quad (21)$$

Phase and slot labels. The sub-phase label $\varphi_t \in \{0, 1, 2, 3\}$ is derived from the gripper state at the start and end of the future action horizon. $\varphi_t = 0$ when the gripper remains open across the horizon, $\varphi_t = 1$ when it transitions from open to closed, $\varphi_t = 2$ when it remains closed, and $\varphi_t = 3$ when it transitions from closed to open. For multi-cycle tasks, $\kappa_t \in \{0, \dots, \text{max_cycles} - 1\}$ is the number of gripper closures observed in the episode before the current step, clipped at $\text{max_cycles} - 1$. A separate phase classifier head maps h_t^{phase} to sub-phase and cycle predictions. The phase loss $\mathcal{L}_{\text{phase}}$ is the sum of cross-entropy on the sub-phase prediction against φ_t and cross-entropy on the cycle prediction against κ_t . The slot label combines the cycle index with the open/closed gripper state,

$$z_t = 2\kappa_t + \mathbf{1}[\varphi_t \geq 2], \quad z_t \in \{0, \dots, K - 1\}, \quad (22)$$

yielding an open-gripper and a closed-gripper slot per value of κ_t and $K = 2 \times \text{max_cycles}$ slot classes. The slot loss $\mathcal{L}_{\text{slot}}$ is cross-entropy on the phase-to-slot router output α_t against z_t .

Gradient routing of slot membership. The per-point slot membership logits m_i are not directly supervised. They receive gradient only through the anchor c_t and the score-as-feature input to the diffusion policy under $\mathcal{L}_{\text{ddpm}}$, so m_i is shaped by what the policy needs from $\hat{s}_t$ and c_t to predict the action, not by a hand-specified per-point segmentation target.

B Simulation Experiment Details

B.1 MimicGen Tasks

We evaluate on four multi-object MimicGen tasks. *Square* inserts a square nut onto a peg. *Coffee* places a capsule into a coffee machine and closes the lid. *Threading* threads a needle through a target hole. *Three-Piece Assembly* sequentially assembles three parts, involving two separate grasp-and-place cycles. Each task is instantiated at three spatial randomization levels, d2 with $SE(2)$ initial pose randomization, and d3 and d4 with $SE(3)$ randomization at increasing range. Figure 3 shows the simulator setup for these tasks and randomization levels.

B.2 Baselines

We compare against four visuomotor policy baselines. SDP [9] is an $SE(3)$ -equivariant point cloud diffusion policy that achieves translational invariance via gripper-frame canonicalization and rotational equivariance via spherical Fourier features. EquiDiff [4] applies discrete $SO(2)$ -equivariance to a voxel diffusion policy in an end-to-end manner. Diffusion Policy [3] is a 2D RGB diffusion policy without equivariance. DP3 [22] is a point cloud diffusion policy without equivariance. We omit three additional baselines that are not directly comparable under our protocol. EquiBot [8] requires per-task object segmentation, which does not generalize to multi-object tasks where the active object changes across phases. AnchorDP3 [23] and AffordDP [24] are evaluated under different settings (dual-arm sparse-keypose prediction and cross-category zero-shot transfer respectively) and depend on segmentation masks or foundation-model-estimated affordances, which are unavailable in our self-supervised, single-arm protocol. As an in-setting substitute for these affordance-augmented diffusion baselines, the *w/o phase* ablation in Table 2 removes phase conditioning and reduces the affordance predictor to a single phase-agnostic score, approximating the phase-agnostic affordance prior of those methods while keeping the data, backbone, and evaluation protocol identical to PAF.

B.3 SDP Reproduction and Seed Variability

To check the effect of demonstration provenance, we reproduce SDP on the exact demonstration sets used to train PAF. The d2 sets are the SDP authors’ released files, whereas the unreleased d3 and d4 sets are regenerated using their official pipeline. Table 5 reports the mean success rate and standard deviation across three seeds for PAF and the reproduced SDP, with 50 evaluation trials per seed and setting. The published SDP results are included separately for reference.

PAF outperforms reproduced SDP across all twelve settings in Table 5. Averaging the displayed per-setting means gives approximately 56.3% for PAF and 47.6% for reproduced SDP. The published SDP average is 48.8%. All twelve reproduced SDP results differ from the published values by at most 3 percentage points. For Square d3, reproduced SDP achieves 39%, compared with the published result of 38%. These results support the performance advantage of PAF over SDP when the training demonstrations are held fixed.

Table 5: Success rate in percent on identical training demonstrations for PAF and reproduced SDP. Higher is better. Entries are mean $\pm$ standard deviation across three seeds. Published SDP entries are the means retained in Table 1.

Setting	PAF	SDP (reproduced)	SDP (published)
Coffee d2	68 $\pm$ 0.0	62 $\pm$ 2.0	63
Coffee d3	62 $\pm$ 2.0	53 $\pm$ 2.3	54
Coffee d4	34 $\pm$ 0.0	31 $\pm$ 2.3	33
Three-Piece d2	68 $\pm$ 4.0	64 $\pm$ 3.5	67
Three-Piece d3	56 $\pm$ 0.0	48 $\pm$ 2.0	49
Three-Piece d4	40 $\pm$ 2.0	37 $\pm$ 2.3	37
Square d2	73 $\pm$ 2.3	61 $\pm$ 3.1	62
Square d3	69 $\pm$ 2.3	39 $\pm$ 3.1	38
Square d4	44 $\pm$ 5.3	29 $\pm$ 1.2	31
Threading d2	65 $\pm$ 1.2	58 $\pm$ 2.0	60
Threading d3	54 $\pm$ 0.0	51 $\pm$ 1.2	53
Threading d4	42 $\pm$ 2.0	38 $\pm$ 0.0	39

B.4 Implementation Details

Table 6 lists the hyperparameters used in the main simulation results. The slot count K is task-dependent and equals $2 \times \text{max_cycles}$, where $\text{max_cycles} = 1$ for the single-cycle tasks Square, Coffee, and Threading, and $\text{max_cycles} = 2$ for the two-cycle task Three-Piece Assembly. The

diffusion policy backbone matches Spherical Diffusion Policy [9], which is also the backbone of the reproduced SDP baseline in Table 5, so this same-data comparison isolates the contribution of the phase-conditioned affordance field. We optimize with AdamW [37] and apply EMA weight averaging following the diffusion-policy backbone. The diffusion policy backbone parameters listed below are inherited from SDP and included for reproducibility.

Table 6: Hyperparameters used in the main simulation results. Values correspond to the PAF configuration used in Table 1.

Parameter	Value
<i>Loss weights</i>	
Score loss weight λ_s	1.0
Phase loss weight λ_p	0.2
Slot loss weight λ_z	0.1
<i>Pseudo-label</i>	
Pseudo-label scale τ_y	0.03
End-effector repulsion scale τ_{ee}	0.12
Repulsion floor η	0.5
<i>Slot routing</i>	
Total slot count K	2 for single-cycle tasks, 4 for Three-Piece Assembly
<i>Affordance graph</i>	
KNN neighbors k_{nn}	24
Graph layers L_g	2
Hidden dim d_h	64
RBF basis count M	16
RBF range d_{max} on normalized coordinates	0.125
Motion history L_h	16
Phase context dim d_{phase}	128
<i>Diffusion policy backbone</i>	
Encoder	5-layer $SO(3)$ -equivariant EquiformerV2
Sphere channels	{16, 32, 64, 128, 256}
Maximum degree $\ell_{max} = m_{max}$	2
Action horizon T_a	16
Observation steps T_o	2
Action steps	8
Noise schedule	DDPM, 100 timesteps
Inference steps	100 (simulation, DDPM [34]), 20 (real robot, DDIM [38])
Score-as-feature channel	enabled
<i>Optimization</i>	
Optimizer	AdamW
Learning rate	1×10^{-4}
Betas	(0.95, 0.999)
Weight decay	1×10^{-6}
LR schedule	cosine, 500 warmup steps
Batch size	32
Epochs at $n_{demo} = 100$	500
EMA power, max	0.75, 0.9999
Hardware	8× RTX 3090, one training run per GPU

C Real-Robot Experiment Details

C.1 Setup and Rollouts

Platform. The robot is a Franka Research 3 arm with a parallel-jaw gripper. Scene observation comes from a single RealSense D435 RGB-D camera mounted in a third-person view of the workspace, and point clouds are obtained by direct back-projection of the depth image without multi-view fusion.

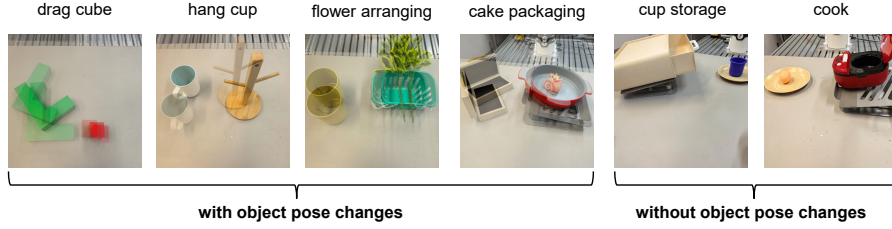


Figure 7: Real-robot task settings. The first four tasks involve object pose changes across episodes, and the last two have fixed object layouts.

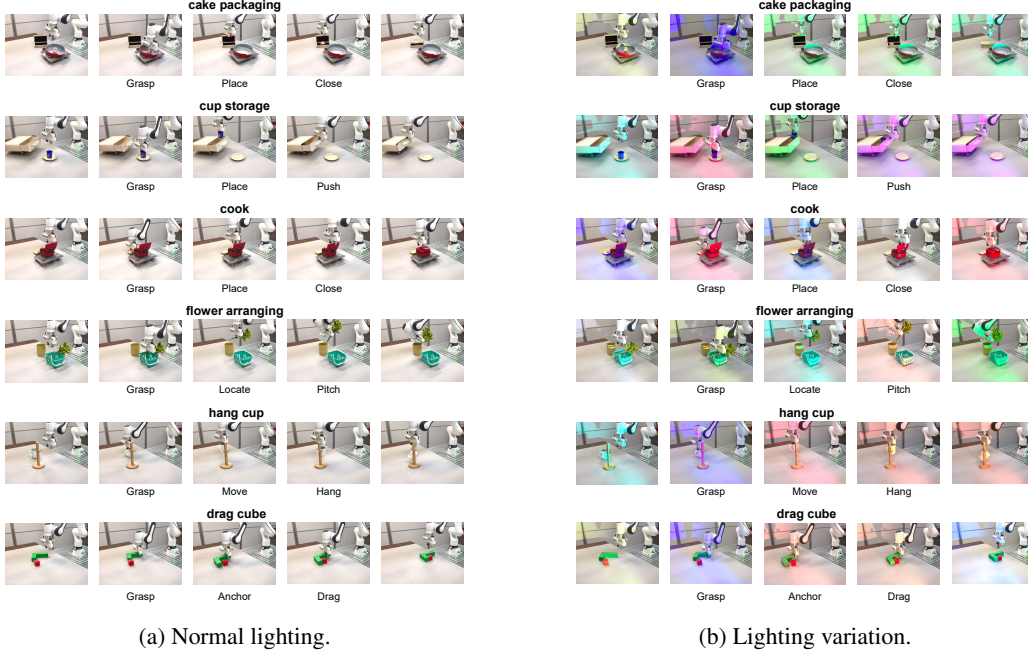


Figure 8: Real-robot rollouts on the six tasks under normal lighting and under varied lighting conditions. Each row visualizes the keyframes of one task labeled with the active sub-phase.

Task descriptions. We design six long-horizon tasks, shown in Figure 7. The first four tasks involve object pose changes across episodes, and the last two have fixed object layouts. *drag cube* uses a tool to drag a block to a target region. *hang cup* hangs a cup on a rack. *flower arranging* inserts a flower into a vase. *cake packaging* places a cake into a box and closes the lid. *cup storage* places a cup into a drawer and closes the drawer. *cook* places an egg into a pot and closes the lid.

Pose variation. Four tasks vary object poses across episodes. In *drag cube* the tool is displaced by ± 5 cm along x and rotated by ± 45 degrees, and the cube is displaced by ± 3 cm along y . In *hang cup* either the cup is displaced by about 12 cm along x or the rack is displaced by 5 cm along the diagonal between $-x$ and $-y$. In *flower arranging* either the vase is displaced by ± 8 cm along x or the flower and its basket are displaced together by ± 5 cm along y . In *cake packaging* the box is displaced by 15 cm along x and rotated by about 90 degrees, and the plate holding the cake is rotated by about 90 degrees on a stand inclined by 30 degrees. Cup storage and cook use fixed layouts, with the drawer and the cookware placed on stands inclined by 30 degrees, so three tasks involve non-upright object orientations.

Slots and phase names. All six tasks contain one grasp-release cycle, and every demonstration has exactly one open-to-closed gripper transition, so the slot count is $K = 2$ for every real-robot policy by the rule $K = 2 \times \max_cycles$ of Appendix A.5. The supervision consists only of the four sub-phase classes φ_t derived from the gripper command and the cycle index κ_t . The phase names in

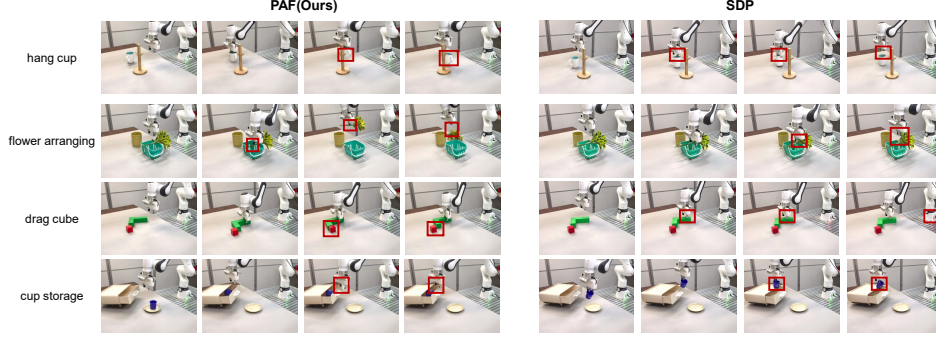


Figure 9: Representative failure cases of PAF and SDP on four real-robot tasks. Red boxes mark the failure moments.

Figures 6 and 8 are author annotations that describe the end-effector state change of the upcoming segment, and each name maps to the sub-phase whose gripper transition it describes. Move before the grasp, push, and close map to $\varphi_t = 0$ with the gripper staying open, grasp maps to $\varphi_t = 1$, move after the grasp, locate, anchor, and drag map to $\varphi_t = 2$ with the gripper staying closed, and place, hang, and pitch map to $\varphi_t = 3$. The names are therefore neither additional supervision classes nor fine-grained phase predictions of the model.

Rollouts. Figure 8 visualizes per-task keyframes for PAF under normal lighting and under the colored-lighting variation used to probe robustness.

C.2 Failure Case Analysis

Figure 9 compares representative failure modes of PAF and SDP on four tasks. SDP fails in earlier phases, missing the cup in *hang cup*, failing to grasp the flower stem in *flower arranging*, colliding with the tool in *drag cube*, and placing the cup inaccurately in *cup storage*. PAF rarely fails in these earlier phases. Its failures concentrate in later, contact-rich interactions, such as the cup loop slipping off the peg in *hang cup*, the stem slipping during rotation in *flower arranging*, the tool failing to hook the block in *drag cube*, and the drawer closing push being misaligned in *cup storage*. This indicates that the affordance field provides an effective geometric inductive bias for spatial localization, and the remaining headroom lies in fine-grained control during contact-rich interactions.

C.3 Inference Overhead

Table 7 reports the per-step inference time measured on the real-robot deployment configuration. The affordance field prediction module adds approximately 100 ms over SDP, a 13% overhead relative to the SDP backbone. The inference bottleneck remains the $SO(3)$ -equivariant diffusion backbone shared by both PAF and SDP, which accounts for the majority of the 750 ms SDP baseline. DP3 is faster because it uses a simpler non-equivariant backbone, at the cost of substantially lower success rates (Table 4).

Table 7: Single-step inference time on a single NVIDIA RTX 4090 with DDIM 20-step denoising.

Method	Inference time (ms)
DP3	~450
SDP	~750
PAF (Ours)	~850