LEAP: MAKING PRIVILEGED GEOMETRY SUPERVISION EFFECTIVE FOR VISUOMOTOR LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Privileged 3D supervision uses additional geometric information during training to guide RGB-based visuomotor policy learning, without requiring geometric inputs at deployment. However, low reconstruction error does not ensure that visual representations capture geometry useful for control. We identify three limitations that weaken this supervision: proprioceptive shortcuts, dominant-view reliance, and reconstruction objectives dominated by task-irrelevant geometry. To address these limitations, we propose Latent Encoding with Aligned Privileged Geometry (LEAP). Our framework uses an auxiliary decoder to reconstruct point clouds within the manipulation workspace from visual features alone, while retaining proprioception for action prediction. Alongside full reconstruction, we introduce wrist-view dropout and partial reconstruction targets matched to the retained wrist, encouraging the encoder to capture complementary local geometry. The auxiliary decoder is removed at inference, leaving only RGB observations and robot state as policy inputs. Extensive experiments on RoboTwin, ManiSkill, and real-world tasks demonstrate consistent and substantial improvements over Diffusion Policy and ACT, with only a small increase in parameter count and no reduction in inference speed.

1 INTRODUCTION

Reliable robotic manipulation depends on the 3D relationships among objects and the robot. RGB-based imitation learning methods, such as Diffusion Policy (Chi et al., 2023) and ACT (Zhao et al., 2023), learn complex skills from demonstrations, but their training objectives do not explicitly require visual features to encode these relationships. Point-cloud policies such as DP3 (Ze et al., 2024) provide explicit geometric inputs, but require depth sensing and point-cloud processing at deployment, adding sensing requirements and computational cost. Privileged geometry supervision instead uses 3D information during training, through auxiliary geometric prediction (Li et al., 2025c; He et al., 2026a; Kim et al., 2026) or representation alignment (Li et al., 2025b), while retaining an RGB policy at deployment.

While RGB-based policies are easier to deploy, this line of work suffers from the following potential issue. Without explicit 3D observations at deployment, an RGB policy must infer task-relevant geometry from its images. A low geometric training loss does not establish that it has done so. In this paper, considering policies that take multi-view RGB images and robot state as input, we investigate the following two questions: (1) when does privileged supervision with depth-derived geometric information fail to make the visual representation capture task-relevant 3D structure? (2) how can privileged 3D supervision be formulated to circumvent this failure? To answer the first question, we identify three limitations of privileged geometry supervision: *proprioceptive shortcuts*, *view dominance*, and *task-irrelevant global prediction*. To address the second, we propose a novel training framework. We expand on those points next.

A *proprioceptive shortcut* occurs when an auxiliary geometry decoder uses robot state to reduce reconstruction error without extracting the corresponding geometry from images. Such shortcuts have been studied in action prediction (Zhao et al., 2025), but their effect on privileged geometry supervision remains underexplored. *View dominance* occurs when the broad-coverage camera explains much of a fused reconstruction target, leaving the contribution of other views unclear. Finally, *task-irrelevant global prediction* can devote much of the auxiliary reconstruction objective to back-

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

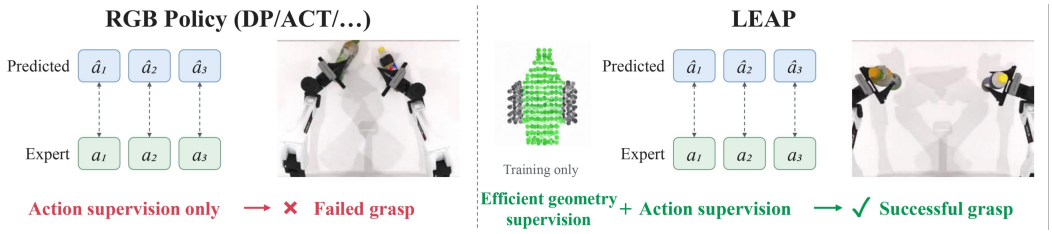


Figure 1: Illustration of LEAP's design goal. LEAP learns geometry-aware policies through training-only 3D supervision, without requiring 3D inputs at deployment.

ground and robot geometry while giving little weight to the local contact and occlusion geometry needed for control. Together, these limitations allow low geometric loss without ensuring that the visual features preserve the geometry relevant to manipulation.

To overcome these limitations, we propose **L**atent **E**ncoding with **A**ligned **P**rivileged Geometry (LEAP), a training framework that directs privileged geometry supervision toward task-relevant visual learning. Its geometry decoder receives only visual features, limiting proprioceptive shortcuts. During training, we randomly mask views and matches a partial geometry target to the remaining views, making wrist contributions easier to identify. At deployment, the policy uses only multi-view RGB images and robot state.

Our contributions can be summarized as follows:

- We identify proprioceptive shortcuts, view dominance, and task-irrelevant global prediction as limitations of privileged geometry supervision for multi-view visuomotor learning.
- We introduce LEAP, combining visual-only geometric reconstruction, view dropout with visibility-aligned partial reconstruction, and workspace-constrained supervision.
- We evaluate these limitations and the corresponding designs through abundant controlled experiments and policy evaluations across simulation and real-robot settings with different RGB-based policy architectures.

2 Related Work

Precise 3D manipulation can be achieved by directly using 3D observations or multiview RGB images as policy inputs, but also via geometric supervision. Next, we recall those three research directions and discuss related work on learning shortcuts.

Explicit 3D Inputs for Robotic Manipulation. Shridhar et al. (2023) voxelize RGB-D inputs, Goyal et al. (2023) render virtual views from point clouds, and Gervet et al. (2023) use sensed depth to lift visual features into a 3D field. GNFactor shares a voxel representation between reconstruction and control (Ze et al., 2023b). Other policies encode point clouds (Ze et al., 2024; Wang et al., 2024a), predict trajectories from 3D scenes (Ke et al., 2024), or combine point maps with RGB (Jia et al., 2025). These approaches retain geometric observations or representations at deployment. LEAP uses geometry only for training supervision and acts from RGB and proprioception.

Multi-View Learning and Camera Availability. Multi-view observations provide complementary evidence for manipulation. Hsu et al. (2022) show that hand-centric and global views differ in their coverage and generalization behavior. MV-MWM (Seo et al., 2023) reconstructs masked views, while 3D-MVP (Qian et al., 2024) and DiffuView (Zhang et al., 2026a) use multi-view pretraining to learn spatial representations. MAD (Almuzairee et al., 2025) learns fused and per-view features for missing-camera robustness, and Multi-Camera View Scaling (Xie et al., 2026) uses synchronized views to increase the effective training data. Akinola et al. (2020) use camera dropout to make RGB policies robust to missing views. In contrast, our concern is the geometric objective under such masking: a low loss on a fused scene may leave the contribution of a wrist view unclear.

Geometric Representation Learning for RGB Policies. Geometric supervision can improve RGB policies without requiring 3D observations at inference. DPR and SPA pretrain encoders

through depth prediction and multi-view rendering, while eVGGT and GP3 transfer pretrained geometric knowledge (Wang et al., 2024b; Zhu et al., 2024; Vuong et al., 2025; Qian et al., 2025b). Wang et al. (2025) distill depth-guided spatial, temporal, and action cues into an RGB policy. During policy learning, methods predict depth, future scenes, or 3D motion (Li et al., 2025c; He et al., 2026a; Qian et al., 2025a; Kim et al., 2026; He et al., 2026b), align features with geometric teachers (Li et al., 2025b; Wang et al., 2026b;a; Shi et al., 2026), or use training-only 3D Gaussian prediction heads (Jiang et al., 2026; Zhang et al., 2026c).

Other methods focus on interaction-relevant structure: object geometry (Ding et al., 2026; Liu et al., 2026), subtask-relevant regions (Gao et al., 2026), or affordances and goal regions (Zheng et al., 2026). Yang et al. (2026a) combine depth and cross-view correspondence supervision, while Xu et al. (2026) jointly predict actions and future point maps from features that include proprioception. LEAP instead reconstructs workspace and view-matched partial point clouds from visual features alone, encouraging geometric learning from complementary camera views.

Proprioceptive Shortcuts in Visuomotor Learning. Prior work identifies proprioceptive overreliance (Lin et al., 2024) and optimization-driven visual suppression (Lu et al., 2026). State-free Policy (Zhao et al., 2025) removes proprioception to improve spatial generalization, while ReViP (Li et al., 2026) rebalances vision and proprioception to reduce false task completion. We instead examine, both theoretically and empirically, how proprioceptive shortcuts can weaken privileged point-cloud supervision in multi-view imitation learning. To mitigate these shortcuts and focus supervision on task-relevant geometry, LEAP reconstructs workspace point clouds from visual features alone and supplements full reconstruction with visibility-aligned partial supervision under wrist-view dropout. This design encourages complementary local geometric representations while retaining proprioception for action prediction.

3 BACKGROUND

Visuomotor Imitation Learning. In this work, we mainly consider imitation learning for two-arm robots equipped with a head camera and two wrist cameras. A visuomotor policy comprises a visual encoder and an action predictor trained on expert demonstrations. Let $\mathcal{V} = \{h, l, r\}$ index the head, left-wrist, and right-wrist cameras, and let $I_{\mathcal{V}} = (I_c)_{c\in\mathcal{V}}$ denote their RGB observations. The proprioceptive state S includes measurements such as joint angles and gripper configurations. A visual encoder f_θ, parameterized by θ, extracts features from each view:

$$Z_c = f_\theta(I_c), \qquad c \in \mathcal{V}. \tag{1}$$

These features are aggregated into a visual representation Z. An action predictor π_η, parameterized by η, conditions on Z and S:

$$a \sim \pi_\eta(\cdot \mid Z, S). \tag{2}$$

Here, a may denote a single command, such as joint targets and gripper commands for the next control step, or a sequence of such commands. ACT predicts action sequences in chunks (Zhao et al., 2023), while Diffusion Policy generates them through iterative denoising (Chi et al., 2023).

Privileged Geometric Supervision. Privileged geometric supervision augments imitation learning with a geometric target G, such as a point cloud derived from depth measurements, that is available during training but not required at deployment. An auxiliary decoder g_ϕ, with parameters ϕ, reconstructs G from the same visual representation Z used by the action predictor, optionally conditioned on robot state. We distinguish two reconstruction settings:

$$\underbrace{g_\phi(Z, S)}_{\text{state-conditioned}}, \qquad \underbrace{g_\phi(Z)}_{\text{visual-only}}. \tag{3}$$

The reconstruction loss trains the auxiliary decoder and the shared visual encoder alongside the imitation-learning objective. At deployment, the auxiliary decoder is removed, and the policy predicts actions using only RGB observations and proprioceptive state.

Reconstruction Risk. We quantify the recoverability of target geometry G from decoder input X using the optimal population reconstruction risk $\mathcal{R}_d(G \mid X)$ defined as follows for $d(\widehat{G}, G) \geq 0$ a

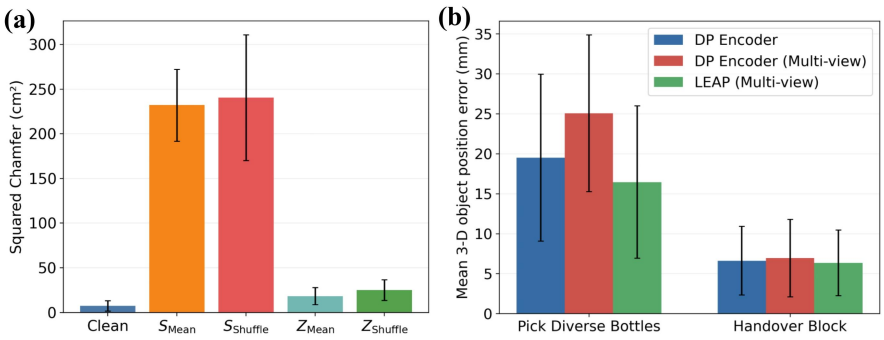


Figure 2: Preliminary diagnostics of geometry learning. (a) Reconstruction error under unperturbed inputs (Clean), mean replacement (Mean), or shuffling (Shuffle) of robot state S or visual features Z. (b) Mean 3D object-position error on two tasks, comparing DP Encoder, DP Encoder with Multi-view setting, and LEAP Encoder with Multi-view setting. Error bars denote standard deviations.

divergence that measures error between a prediction and its target:

$$\mathcal{R}_d(G \mid X) := \inf_g \mathbb{E}[d(g(X), G)]\,, \tag{4}$$

where the infimum ranges over all measurable decoders g, and the expectation is over the joint distribution of (G, X). The divergence takes the prediction first and the target second. We consider $X = Z$ for visual-only reconstruction and $X = (Z, S)$ for state-conditioned reconstruction. For vector-valued G, $\mathcal{R}_2$ denotes the risk under squared Euclidean error, $d_2(\widehat{G}, G) = \|\widehat{G} - G\|_2^2$.

4 Method

We first identify three limitations of privileged geometric supervision: proprioceptive shortcuts, dominant-view reliance, and reconstruction objectives dominated by task-irrelevant geometry. These issues can allow low reconstruction error without ensuring that visual features capture the geometry needed for control. Guided by this analysis, we introduce LEAP, which aims to make geometric supervision an observation-aligned and policy-relevant constraint on visual representations. As shown in Figure 3, its core principles are to separate geometric learning from proprioceptive shortcuts, encourage complementary information across views, and focus supervision on task-relevant geometry. The resulting policy requires only RGB observations and proprioceptive state at deployment.

4.1 When Does Privileged Geometry Benefit Policy Learning?

Low reconstruction error alone cannot indicate that a visual representation captures the geometry required for control. We first analyze proprioceptive and dominant-view shortcuts, accompanied by preliminary empirical diagnostics. We then discuss how global reconstruction objectives can underweight task-relevant local geometry.

Proprioceptive Shortcut. Recent methods (Xu et al., 2026; Yang et al., 2026b) predict privileged geometry from both visual features and robot state. Robot state determines the robot's configuration and may correlate with scene geometry, allowing the decoder to reduce reconstruction error without requiring visual features to encode the corresponding geometry.

We investigate this shortcut using a DP policy trained with state-conditioned geometry reconstruction on RoboTwin. With the encoder and decoder frozen, we replace either robot state or visual features with their training-set mean (S_{Mean}, Z_{Mean}) or shuffle them across evaluation samples (S_{Shuffle}, Z_{Shuffle}), keeping the other input and reconstruction target unchanged. We measure reconstruction quality using squared Chamfer error, which averages mean squared nearest-neighbor errors between predicted and target point clouds. As shown in Figure 2(a), both state interventions sharply increase reconstruction error, whereas the corresponding visual interventions produce much smaller increases. This difference suggests that the trained decoder relies strongly on proprioception, although it does not establish that visual features are completely invalid.

To explain why the reconstruction objective permits this behavior, the following proposition compares the optimal population risks with and without state conditioning (see Section B.1 for proof).

Proposition 4.1 (State-induced non-identifiability)**.** *Let $d : \mathcal{G} \times \mathcal{G} \to [0, \infty)$ be a measurable reconstruction cost; no metric properties are required. For the population risk in Equation* (4)*, with the infimum taken over all measurable decoders,*

$$\mathcal{R}_d(G \mid Z, S) \leq \mathcal{R}_d(G \mid Z). \tag{5}$$

For $\mathcal{G} = \mathbb{R}^m$, $\mathbb{E}[\|G\|_2^2] < \infty$, and squared Euclidean divergence $d_2(\widehat{G}, G) = \|\widehat{G} - G\|_2^2$,

$$\mathcal{R}_2(G \mid Z) - \mathcal{R}_2(G \mid Z, S) = \mathbb{E}\Big[\|\mathbb{E}[G \mid Z, S] - \mathbb{E}[G \mid Z]\|_2^2\Big]. \tag{6}$$

Here, $\mathbb{E}[G \mid Z, S]$ and $\mathbb{E}[G \mid Z]$ are conditional means under the joint data distribution: the population-optimal predictions under squared Euclidean loss, rather than outputs of two trained decoders. The general inequality applies to squared Chamfer error, while the risk-gap identity concerns vector-valued reconstruction under squared Euclidean error.

This limitation can occur even with perfect state-conditioned reconstruction. In the squared Euclidean setting, if $G = \psi(S)$ is not almost surely constant and Z is constant, then we can have $\mathcal{R}_2(G \mid Z, S) = 0$, whereas

$$\mathcal{R}_2(G \mid Z) = \mathbb{E}\big[\|G - \mathbb{E}[G]\|_2^2\big] > 0. \tag{7}$$

Thus, low state-conditioned reconstruction error alone cannot certify that geometry is recoverable from visual features.

Dominant-View Shortcut. Even when reconstruction uses only visual inputs, the representation may rely primarily on a subset of views. The head camera provides broad scene coverage, while wrist cameras reveal complementary local geometry. If the head view already explains most reconstruction target, low reconstruction error cannot indicate the contribution of wrist observations.

We examine whether wrist features improve object-position prediction using frozen visual representations. For the DP baseline trained with full point-cloud reconstruction, *DP Encoder* uses only head-view features Z_h, whereas *DP Encoder (Multi-view)* uses concatenated features $[Z_h, Z_l, Z_r]$ from the head and both wrist cameras. We additionally evaluate *LEAP (Multi-view)*, which uses the same three-view feature configuration but with representations learned through LEAP. In all settings, camera-specific features are extracted before fusion and kept frozen while a small MLP is trained to predict task-object 3D positions. We report mean Euclidean error on held-out episodes over three random seeds for MLP training. As shown in Figure 2(b), adding wrist features to the full-reconstruction DP baseline does not reduce mean prediction error on either task, whereas LEAP achieves lower mean error using the same three-view feature configuration. These results suggest that simply including wrist features does not guarantee better object-position prediction, motivating supervision that encourages complementary geometric information across views. The following proposition formalizes a related limitation of the reconstruction objective: when the head view alone permits accurate reconstruction, low error cannot establish that wrist observations contribute.

Proposition 4.2 (Non-identifiability of view-wise contributions)**.** *Let I_h, I_l, and I_r denote observations from the head, left-wrist, and right-wrist cameras, respectively, and let G denote the target geometry. Suppose a head-view encoder f_h and decoder g_h satisfy*

$$\mathbb{E}[d(g_h(f_h(I_h)), G)] \leq \varepsilon.$$

Consider full-view models with concatenated embeddings. If their hypothesis classes admit constant wrist encoders and a decoder that applies g_h to the head embedding alone, then there exists an admissible pair (F, g) satisfying

$$F(I_h, I_l, I_r) = [f_h(I_h), c_l, c_r], \quad \mathbb{E}[d(g(F(I_h, I_l, I_r)), G)] \leq \varepsilon,$$

The proof (see Section B.2) constructs a full-view model that reproduces the head-only predictor while ignoring both wrist observations. Consequently, reconstruction accuracy alone cannot certify wrist-view use. This matters for control because geometric coverage and control relevance need not coincide. A wrist view may reveal a gripper–bottle gap hidden from the head camera: errors in this local geometry may contribute little to reconstruction error while determining grasp success.

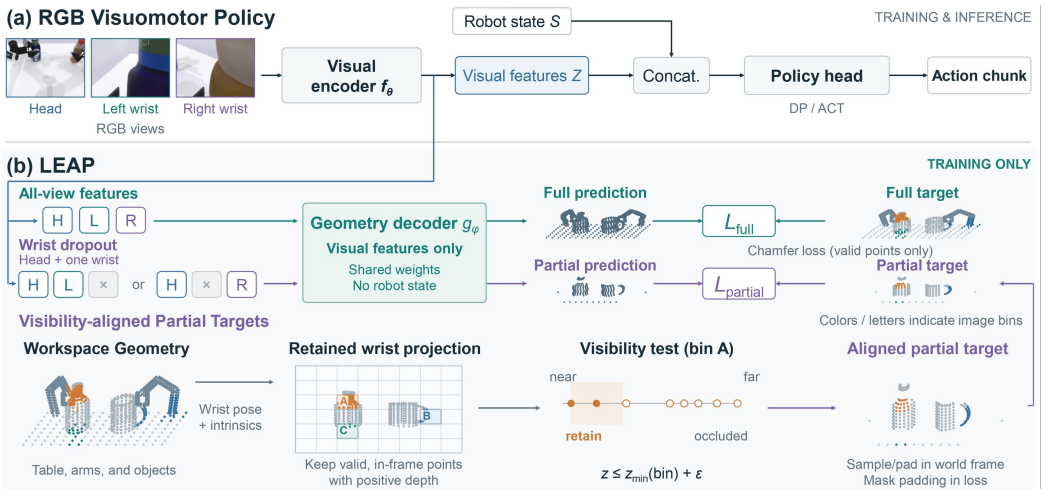


Figure 3: Overview of LEAP. (a) The workflow of an RGB-based policy. (b) During training, LEAP supervises the visual encoder through full and partial point-cloud reconstruction, using visual features alone. Partial reconstruction retains the head and one randomly selected wrist, with targets matched to that wrist's approximate visibility. The geometry decoder is removed at inference.

Task Relevance of Geometric Supervision. Uniform spatial averaging can dilute supervision of task-relevant geometry. Consider a depth target with N pixels and a task-relevant subset $\mathcal{R}$ occupying a fraction $\alpha = |\mathcal{R}|/N$. For per-pixel reconstruction errors ℓ_i, define the regional means as

$$\mathcal{L}_{\text{relevant}} = \frac{1}{|\mathcal{R}|} \sum_{i \in \mathcal{R}} \ell_i, \qquad \mathcal{L}_{\text{remaining}} = \frac{1}{|\mathcal{R}^c|} \sum_{i \in \mathcal{R}^c} \ell_i.$$

The global average therefore satisfies

$$\mathcal{L}_{\text{global}} = \alpha\, \mathcal{L}_{\text{relevant}} + (1 - \alpha)\, \mathcal{L}_{\text{remaining}}. \tag{8}$$

For small α, low global error can mask large task-relevant errors, motivating local supervision alongside global reconstruction.

The propositions identify possible, not inevitable, shortcuts. Together with our diagnostics and task-relevance analysis, they motivate supervision that promotes visual geometric encoding, complementary view contributions, and emphasis on control-relevant regions.

4.2 Latent Encoding with Aligned Privileged Geometry (LEAP)

We introduce LEAP to address the limitations identified in Section 4.1. The method combines grounding geometry in visual representations, learning complementary geometry with visibility-aligned partial reconstruction, and workspace-constrained geometric supervision.

Grounding Geometry in Visual Representations. State-conditioned reconstruction can exploit proprioceptive information without requiring the corresponding geometry in visual features (Proposition 4.1). We therefore allow the action predictor to use robot state but exclude it from the geometry decoder. The reconstruction pathway receives only visual features, and its losses update the encoder shared with the policy. This removes direct proprioceptive access from geometric reconstruction while preserving its use for action prediction.

Learning Complementary Geometry. The head camera provides broad scene coverage, whereas wrist cameras reveal local geometry that may be obscured from the head view. To encourage each wrist to contribute complementary geometric information, we apply view dropout during training, always retaining the head view and randomly retaining one wrist:

$$M \in \big\{\{h, l\}, \{h, r\}\big\}. \tag{9}$$

Each wrist is thus trained as the sole wrist-side observation accompanying the head camera, encouraging the encoder to exploit its local observations without relying on the other wrist.

Visibility-Aligned Partial Reconstruction. To couple view dropout with local geometric supervision, we supplement full-scene reconstruction with a partial objective targeting surfaces approximately visible from the retained wrist. Let G^{full} denote the fused point cloud within the manipulation workspace, whose construction is described below, and let $w(M) \in \{l, r\}$ denote the retained wrist. We construct its partial target as

$$G_M^{\text{partial}} = \widetilde{\text{Vis}}_{w(M)}\big(G^{\text{full}}\big)\,, \tag{10}$$

where $\widetilde{\text{Vis}}_{w(M)}$ selects approximately visible points through projection and depth filtering. Specifically, we transform the full-cloud points from world coordinates into the retained camera's coordinate frame using its position and orientation, then project them into the image using its calibrated projection model. After removing invalid points and those behind the camera or outside the image, we group nearby projections into image bins. Within each bin, we retain points close in depth to the nearest point and treat more distant points as occluded. Visibility is approximate because occlusion is inferred from the sampled cloud. The selected points retain their world coordinates and are sampled or padded to a fixed size, with validity masks excluding padded entries from the reconstruction loss. The partial target therefore follows the retained wrist, while the head view remains available as context. Together with full-scene reconstruction, this objective encourages the representation to capture local wrist-visible geometry.

Workspace-Constrained Geometric Supervision. We construct the full reconstruction target from depth-derived point clouds captured by all three cameras. Camera-space points are transformed into a common world coordinate frame, cropped to the configured manipulation workspace, and merged. Farthest-point sampling produces the full target G^{full} with up to a fixed number of valid points; If the number of points is insufficient, we pad the target to the fixed number. Geometric supervision uses the world-coordinate XYZ values.

The workspace crop provides a spatial relevance constraint by excluding geometry outside the manipulation region. Point-cloud targets make this selection explicit in 3D and place observations from different cameras in a common coordinate frame. Here, full reconstruction refers to the fused target within this workspace.

Training Objective. Let $\widehat{G}^{\text{full}}$ and $\widehat{G}_M^{\text{partial}}$ denote the full and partial predictions. We supervise both using Chamfer reconstruction loss:

$$\mathcal{L}_{\text{geo}} = \mathbb{E}_M\Big[\lambda_{\text{full}}\,\ell_{\text{CD}}\Big(\widehat{G}^{\text{full}}, G^{\text{full}}\Big) + \lambda_{\text{partial}}\,\ell_{\text{CD}}\Big(\widehat{G}_M^{\text{partial}}, G_M^{\text{partial}}\Big)\Big]. \tag{11}$$

Here, $\lambda_{\text{full}}, \lambda_{\text{partial}} \geq 0$ weight the two reconstruction terms, and $\mathbb{E}_M$ averages over the wrist-view sampling rule. The loss ℓ_{CD} is evaluated with the validity and frame masks. Padded points are excluded from distance computation. In practice, we specify a minimum number of valid points for full and partial targets separately, excluding targets below their respective thresholds from the corresponding reconstruction losses.

The complete training objective combines action prediction with both geometric terms:

$$\mathcal{L}_{\text{LEAP}} = \mathcal{L}_{\text{policy}} + \mathcal{L}_{\text{geo}}. \tag{12}$$

The policy loss and both reconstruction terms jointly update the shared visual encoder. At deployment, the policy uses RGB observations and robot state, without privileged point-cloud targets or the auxiliary reconstruction pathway.

5 Experimental Results

5.1 Experimental Setup

Datasets and Tasks. We evaluate LEAP on 50 RoboTwin tasks, six ManiSkill tasks, and three real-world manipulation tasks. To examine performance across different geometric requirements, we group the RoboTwin tasks into four categories: **Object Spatial** (14), **Multi-object** (14), **Interaction** (14), and **Less-geometry** (8). We report success rates for each category and the overall average across all 50 tasks. The ManiSkill and real-world experiments provide complementary evaluations across simulation environments and physical manipulation settings. More details in Section C.2.

Table 1: Overall performance on the 50 RoboTwin tasks across 3 random seeds. Tasks are grouped according to their geometric requirements. DP, RDT, π_0 use the official results with no SD.

Method	Obs.	Success Rate (%) by Task Category ↑					Params. ↓
		Object Spatial(14)	Multi-object(14)	Interaction(14)	Less-geometry(8)	Average	
DP3 (RSS-24)	P.C.	45.3±3.2	41.6±2.2	66.9±5.5	80.4±4.3	55.9±3.7	262.43M
DP (RSS-23)	RGB(1)	19.8	24.2	30.4	45.0	28.0	85.62M
DP-MultiView (RSS-23)	RGB	18.0±2.6	19.5±4.9	30.2±4.7	44.9±3.5	26.1±4.0	86.35M
DP-eVGGT (arxiv-25)	RGB	25.1±3.0	25.5±4.2	38.1±3.7	51.3±4.1	33.0±3.8	129.66M
RDT (ICLR-25)	RGB	26.4	21.0	46.4	51.5	34.5	1200M
DP-SF (ICLR-26)	RGB	35.6±2.0	32.5±2.8	50.0±4.7	60.8±5.1	42.8±3.5	129.66M
π_0 (RSS-25)	RGB	34.3	38.9	52.1	70.9	46.4	3300M
GAP (CVPR-26)	RGB	39.9±2.5	38.9±4.3	54.6±3.6	62.8±4.9	47.4±3.7	1200M
DP-LEAP (Ours)	RGB	**44.1±2.6**	**47.1±3.6**	**68.8±3.8**	**73.4±4.2**	**56.5±3.5**	87.78M
ACT-pointcloud	P.C.	32.9±2.4	29.3±2.9	58.6±3.3	79.1±3.6	46.5±3.0	262.84M
ACT (RSS-23)	RGB	16.6±3.0	18.8±3.4	38.9±4.2	50.4±6.9	28.9±4.1	83.90M
ACT-LEAP (Ours)	RGB	36.8±3.5	33.2±2.6	58.8±4.4	75.5±4.0	48.1±3.6	85.63M

Baseline Methods. We compare against Action Chunking with Transformers (ACT) (Zhao et al., 2023) and Diffusion Policy (DP) (Chi et al., 2023) as standard RGB-based imitation learning baselines. We also include eVGGT (Vuong et al., 2025), GAP (Xu et al., 2026), and Spatial Forcing (Li et al., 2025b) as geometry-aware approaches; DP3 (Ze et al., 2024), which directly consumes point-cloud, as a geometry-input upper bound; and π_0 (Black et al., 2025) and RDT (Liu et al., 2025) as robot foundation models. For the DP-based comparisons, we integrate the eVGGT visual encoder and adapt Spatial Forcing's representation-alignment strategy to DP, denoting these variants as eVGGT-DP and Spatial Forcing-DP. We apply LEAP to both policy architectures, yielding DP-LEAP and ACT-LEAP, to evaluate its effectiveness across different action prediction frameworks.

Training and Evaluation. Our RGB policies use observations from three cameras, including one head camera and two wrist cameras, together with robot proprioceptive state. We train DP-based policies for 600 epochs and ACT-based policies for 4,000 epochs, using three independent random seeds. For LEAP, privileged point-cloud supervision is used only during training; inference requires only RGB observations and proprioception. We summarize performance using task success rates, reporting the mean and standard deviation across seeds. More implementation details in Section C.

5.2 Performance Analysis

Quantitative Evaluations on RoboTwin. As shown in Table 1, LEAP improves both backbones across all four task categories, raising average success from 28.0% to 56.5% for DP and from 28.9% to 48.1% for ACT, with only 2.5% and 2.1% more parameters, respectively. Among RGB policies, DP-LEAP achieves the highest average success (56.5%), outperforming geometry-aware baselines DP-eVGGT (33.0%), DP-SF (42.8%), and GAP (47.4%), as well as RDT (34.5%) and π_0 (46.4%). Compared with GAP, the strongest competing RGB baseline, its advantage holds across all categories and is largest on Interaction tasks (68.8% vs. 54.6%).

We use DP3 as an empirical upper-bound reference because it directly receives point clouds at inference. Despite requiring no point-cloud input at deployment, DP-LEAP surpasses this reference in average success rate, achieving 56.5% versus 55.9%. ACT-LEAP likewise exceeds its point-cloud counterpart, achieving 48.1% compared with 46.5% for ACT-pointcloud. These results show that our training-only geometry supervision enables RGB policies to reach, and slightly surpass, the average performance of their counterparts with direct access to 3D observations.

Quantitative Evaluations on Maniskill. As shown in Table 2, LEAP improves both policy backbones across all six ManiSkill tasks. Average success increases from 26.4% to 42.5% for DP and from 20.0% to 33.2% for ACT. DP-LEAP achieves the highest average success among the evaluated RGB policies, outperforming DP-SF (35.9%) and RDT (36.8%) while matching or exceeding their success rates on every task. These improvements substantially narrow the gap to the point-cloud policies, which serve as empirical upper-bound references with direct access to 3D observations. DP-LEAP approaches DP3 in average success (42.5% versus 43.3%), while ACT-LEAP

Table 2: Success Rate(%) on the 6 Maniskill tasks across 3 random seeds.

Method	PickCube	StackCube	Peg Insertion	Plug Charger	PullCube Tool	PushT	Average
DP3	45 ± 2	71 ± 2	7 ± 3	13 ± 2	91 ± 4	34 ± 2	43.3 ± 1.5
DP	32 ± 4	41 ± 1	0 ± 1	0 ± 0	68 ± 3	18 ± 3	26.4 ± 1.7
DP-SF	43 ± 3	55 ± 5	$\mathbf{6} \pm 1$	6 ± 1	85 ± 2	21 ± 1	35.9 ± 1.2
RDT	40 ± 2	61 ± 4	4 ± 1	8 ± 4	88 ± 3	20 ± 4	36.8 ± 1.3
DP-LEAP	$\mathbf{49} \pm 2$	$\mathbf{66} \pm 4$	$\mathbf{6} \pm 2$	$\mathbf{9} \pm 2$	$\mathbf{96} \pm 2$	$\mathbf{31} \pm 2$	$\mathbf{42.5} \pm 1.1$
ACT-pointcloud	41 ± 2	57 ± 6	4 ± 1	5 ± 2	63 ± 5	34 ± 4	34.1 ± 1.4
ACT	19 ± 1	30 ± 2	0 ± 1	1 ± 1	45 ± 2	25 ± 2	20.0 ± 1.3
ACT-LEAP	$\mathbf{40} \pm 2$	$\mathbf{50} \pm 1$	$\mathbf{4} \pm 1$	$\mathbf{8} \pm 1$	$\mathbf{69} \pm 3$	$\mathbf{28} \pm 4$	$\mathbf{33.2} \pm 1.7$

similarly approaches ACT-pointcloud (33.2% versus 34.1%). Both variants also exceed their point-cloud counterparts on PullCubeTool, demonstrating competitive manipulation performance without requiring point clouds at deployment.

Table 3: Ablation: mean success (%) over five tasks. Task-level results appear in Table 6.

LEAP variant	DP	ACT
Full	**76.6**	**49.0**
w/o wrist dropout	55.2	38.6
w/ state in decoder	58.2	36.8
w/ depth reconstruction	71.0	45.4
w/o target alignment	62.0	44.4

Table 4: Real-world success rates (%). Bold marks the better RGB result.

Task	DP3	DP	DP-LEAP
Pick-Place	95	35	**90**
Assemble	70	15	**85**
Handover-Block	30	0	**30**
Average	65.0	16.7	**68.3**

Real-world Experiments. As shown in Table 4, LEAP improves DP across all three real-world tasks, raising average success from 16.7% to 68.3%. The largest gain occurs on Assemble, from 15% to 85%. Without point clouds at deployment, DP-LEAP also slightly surpasses DP3 on average (68.3% vs. 65.0%), achieving 90% vs. 95% on Pick-Place, 85% vs. 70% on Assemble, and matching DP3 at 30% on the challenging Handover-Block task. Real-world demonstrations are provided in the supplementary materials.

Ablation Study. Table 3 shows that each design contributes to average performance. Removing wrist dropout reduces average success from 76.6% to 55.2% for DP and from 49.0% to 38.6% for ACT. Retaining wrist dropout but removing target alignment yields 62.0% and 44.4%, respectively, demonstrating the additional benefit of matching geometric supervision to the available views. Adding robot state to the geometry decoder lowers success to 58.2% and 36.8%, supporting state-separated visual reconstruction. Replacing point-cloud reconstruction with depth reconstruction yields 71.0% for DP and 45.4% for ACT, indicating that point-cloud supervision provides a more effective training signal. See full results in Section D.1 and sensitivity analysis in Section D.2.

6 Conclusion and Limitation

We presented LEAP, which improves RGB visuomotor learning through visual-only reconstruction, workspace-constrained targets, and wrist-view dropout with aligned partial supervision alongside full reconstruction. Across abundant experiments, it improves Diffusion Policy and ACT in all four categories, achieving average performance comparable to their point-cloud counterparts while requiring only RGB and robot state at deployment. Ablations support each component, highlighting the value of observation-aligned, task-relevant geometric supervision. The current implementation uses predefined workspace bounds; automating their selection remains future work.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, improving the clarity and presentation of the manuscript, and supporting literature retrieval and discovery. All technical content, research ideas, experimental design, analysis, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide our code implementation in the supplementary materials. Detailed hyperparameters, training configurations, data collection procedures, and evaluation protocols are documented in Section C. The official codebase will be made publicly available upon acceptance.

REFERENCES

Iretiayo Akinola, Jacob Varley, and Dmitry Kalashnikov. Learning precise 3D manipulation from multiple uncalibrated cameras. In *IEEE International Conference on Robotics and Automation*, 2020. URL https://arxiv.org/abs/2002.09107.

Abdulaziz Almuzairee, Rohan Patil, Dwait Bhatt, and Henrik I. Christensen. Merging and disentangling views in visual reinforcement learning for robotic manipulation. *arXiv preprint arXiv:2505.04619*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2410.24164.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/947018640bf36a2bb609d3557a285329-Abstract.html.

Xingyu Ding, Yuzhong Zhao, Yang Wu, Chunhai Zhao, Chaoyang Zhao, Yifan Zhang, and Jian Cheng. Mind-VLA: Instruction-aware spatial representation alignment for vision-language-action models. *arXiv preprint arXiv:2608.04633*, 2026. URL https://arxiv.org/abs/2608.04633. arXiv:2608.04633v1, submitted 2026-08-05.

Mhairi Dunion and Stefano V. Albrecht. Multi-view disentanglement for reinforcement learning with multiple cameras. *Reinforcement Learning Journal*, 2:498–515, 2024. URL https://arxiv.org/abs/2404.14064.

Zhiyuan Gao, Di Wen, Yanxiang Zhan, Mohammad Khoshnazar, Jeroen Schäfer, Kunyu Peng, and Michael Beetz. FOCAL-VLA: Subtask-guided geometry distillation and implicit world modeling for vision-language-action models. *arXiv preprint arXiv:2609.21228*, 2026. URL https://arxiv.org/abs/2609.21228.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3D: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023. URL https://arxiv.org/abs/2306.17817.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. RVT: Robotic view transformer for 3d object manipulation. *arXiv preprint arXiv:2306.14896*, 2023. URL https://arxiv.org/abs/2306.14896.

Yuxin He, Ruihao Zhang, Xianzu Wu, Zhiyuan Zhang, Cheng Ding, and Qiang Nie. 3d dynamics-aware manipulation: Endowing manipulation policies with 3d foresight. In *IEEE International Conference on Robotics and Automation*, 2026a.

Zeyuan He, Bowen Yang, Zhirui Fang, Keru Zhou, Lei Jiang, Jingjing Qian, Fan Mo, Junchi Yan, Philip Torr, Xiu Li, Li Jiang, and Jialin Yu. ELAN4D: Embodiment-centric 4d supervision for vision-language-action models via plug-and-play adaptation. *arXiv preprint arXiv:2605.30484*, 2026b. URL https://arxiv.org/abs/2605.30484. arXiv:2605.30484v1.

Kyle Hsu, Moo Jin Kim, Rafael Rafailov, Jiajun Wu, and Chelsea Finn. Vision-based manipulators need to also see from their hands. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2203.12677.

Rishabh Jangir, Nicklas Hansen, Sambaran Ghosal, Mohit Jain, and Xiaolong Wang. Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation. *arXiv preprint arXiv:2201.07779*, 2022. URL https://arxiv.org/abs/2201.07779.

Xiaogang Jia, Qian Wang, Anrui Wang, Han A. Wang, Balázs Gyenes, Emiliyan Gospodinov, Xinkai Jiang, Ge Li, Hongyi Zhou, Weiran Liao, Xi Huang, Maximilian Beck, Moritz Reuss, Rudolf Lioutikov, and Gerhard Neumann. PointMapPolicy: Structured point cloud processing for multi-modal imitation learning. *arXiv preprint arXiv:2510.20406*, 2025. URL https://arxiv.org/abs/2510.20406.

Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, and Shanghang Zhang. Lift3D foundation policy: Lifting 2D large-scale pretrained models for robust 3D robotic manipulation. *arXiv preprint arXiv:2411.18623*, 2024. URL https://arxiv.org/abs/2411.18623.

Yuqing Jiang, Zijian Zhang, Weitao Zhou, Jiawei Wang, Junjie He, Lei Yang, Haifang Qing, Si Liu, Ding Zhao, Ping Luo, and Haibao Yu. GaussianDream++: Efficient 3D gaussian world modeling for robotic manipulation. *arXiv preprint arXiv:2608.25659*, 2026. URL https://arxiv.org/abs/2608.25659.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. URL https://arxiv.org/abs/2402.10885.

Jisoo Kim, Jungbin Cho, Sanghyeok Chu, Ananya Bal, Jinhyung Kim, Gunhee Lee, Sihaeng Lee, Seung Hwan Kim, Bohyung Han, Hyunmin Lee, Laszlo A. Jeni, and Seungryong Kim. Pri4R: Learning world dynamics for vision-language-action models with privileged 4d representation. *arXiv preprint arXiv:2603.01549*, 2026.

Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. PointVLA: Injecting the 3d world into vision-language-action models. *arXiv preprint arXiv:2503.07511*, 2025a. URL https://arxiv.org/abs/2503.07511.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025b.

Yixuan Li, Yuhui Chen, Mingcai Zhou, Haoran Li, Zhengtao Zhang, and Dongbin Zhao. QDepth-VLA: Quantized depth prediction as auxiliary supervision for vision-language-action models. *arXiv preprint arXiv:2510.14836*, 2025c.

Zhuohao Li, Yinghao Li, Jian-Jian Jiang, Lang Zhou, Tianyu Zhang, Jiadong Yin, Mu Lin, Yi-Lin Wei, and Wei-Shi Zheng. ReViP: Mitigating false completion in vision-language-action models with vision-proprioception rebalance. *arXiv preprint arXiv:2601.16667*, 2026. URL https://arxiv.org/abs/2601.16667.

Xingyu Lin, John So, Sashwat Mahalingam, Fangchen Liu, and Pieter Abbeel. SpawnNet: Learning generalizable visuomotor skills from pre-trained networks. In *IEEE International Conference on Robotics and Automation*, 2024. URL https://arxiv.org/abs/2307.03567.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yAzN4tz7oI.

Zonghe Liu, Shanyuan Jie, Xiaoquan Sun, Chen Cao, Zetian Xu, Zongsheng Liu, and Jiayu Chen. SAM3D-guided object-centric representation alignment for vision-language-action models. *arXiv preprint arXiv:2607.25912*, 2026. URL https://arxiv.org/abs/2607.25912.

Jingxian Lu, Wenke Xia, Yuxuan Wu, Zhiwu Lu, and Di Hu. When would vision-proprioception policies fail in robotic manipulation? *arXiv preprint arXiv:2602.12032*, 2026. URL https://arxiv.org/abs/2602.12032. arXiv:2602.12032v1; accepted by ICLR 2026.

Xincheng Pang, Wenke Xia, Zhigang Wang, Bin Zhao, Di Hu, Dong Wang, and Xuelong Li. Depth helps: Improving pre-trained rgb-based policy with depth information injection. *arXiv preprint arXiv:2408.05107*, 2024. URL https://arxiv.org/abs/2408.05107.

Jingjing Qian, Boyao Han, Chen Shi, Lei Xiao, Long Yang, Shaoshuai Shi, and Li Jiang. GeoPredict: Leveraging predictive kinematics and 3d gaussian geometry for precise vla manipulation. *arXiv preprint arXiv:2512.16811*, 2025a. URL https://arxiv.org/abs/2512.16811.

Quanhao Qian, Guoyang Zhao, Gongjie Zhang, Jiuniu Wang, Ran Xu, Junlong Gao, and Deli Zhao. GP3: A 3d geometry-aware policy with multi-view images for robotic manipulation. *arXiv preprint arXiv:2509.15733*, 2025b. URL https://arxiv.org/abs/2509.15733.

Shengyi Qian, Kaichun Mo, Valts Blukis, David F. Fouhey, Dieter Fox, and Ankit Goyal. 3D-MVP: 3d multiview pretraining for robotic manipulation. *arXiv preprint arXiv:2406.18158*, 2024. URL https://arxiv.org/abs/2406.18158.

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*, volume 202, pp. 30613–30632. PMLR, 2023.

Jiaxin Shi, Xidong Zhang, Fucai Zhu, Zhe Li, Siyu Zhu, and Weihao Yuan. 3DThinkVLA: Endowing vision-language-action models with latent 3d priors via 3d-thinking-guided co-training. *arXiv preprint arXiv:2606.04436*, 2026. URL https://arxiv.org/abs/2606.04436. arXiv:2606.04436v1.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 785–799. PMLR, 2023. URL https://proceedings.mlr.press/v205/shridhar23a.html.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. ReconVLA: Reconstructive vision-language-action model as effective robot perceiver. *arXiv preprint arXiv:2508.10333*, 2025. URL https://arxiv.org/abs/2508.10333.

An Dinh Vuong, Minh Nhat Vu, and Ian Reid. Improving robotic manipulation with efficient geometry-aware vision encoder. *arXiv preprint arXiv:2509.15880*, 2025.

Chenxi Wang, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. RISE: 3d perception makes real-world robot imitation simple and effective. *arXiv preprint arXiv:2404.12281*, 2024a. URL https://arxiv.org/abs/2404.12281.

Chenyi Wang, Xinkai Wang, Bokai Lin, Jialin Tian, Fucheng Zhang, Cewu Lu, and Lixin Yang. Track4Action: Distilling world-centric 3d tracker into vision-language-action policies. *arXiv preprint arXiv:2608.03727*, 2026a. URL https://arxiv.org/abs/2608.03727. arXiv:2608.03727v1, submitted 2026-08-04.

Hao Wang, Xiaobao Wei, Jingyang He, Chengyu Bai, Chun-Kai Fan, Jiajun Cao, Jintao Chen, Ying Li, Shanyu Rong, Ming Lu, Xiaozhu Ju, Jian Tang, and Shanghang Zhang. VEGA: Visual encoder grounding alignment for spatially-aware vision-language-action models. *arXiv preprint arXiv:2605.10485*, 2026b. URL https://arxiv.org/abs/2605.10485.

Wanying Wang, Jinming Li, Yichen Zhu, Zhiyuan Xu, Zhengping Che, Yaxin Peng, Chaomin Shen, Dong Liu, Feifei Feng, and Jian Tang. Visual robotic manipulation with depth-aware pretraining. *arXiv preprint arXiv:2401.09038*, 2024b. URL https://arxiv.org/abs/2401.09038.

Ziru Wang, Mengmeng Wang, Guang Dai, Yongliu Long, and Jingdong Wang. MonoLift: Learning 3D manipulation policies from monocular RGB via distillation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/file/07bdb30a6aa9792c404d5d62b626f9d7-Paper-Conference.pdf.

Yichen Xie, Yixiao Wang, Shuqi Zhao, Cheng-En Wu, Masayoshi Tomizuka, Jianwen Xie, and Hao-Shu Fang. Multi-camera view scaling for data-efficient robot imitation learning. *arXiv preprint arXiv:2604.00557*, 2026. URL https://arxiv.org/abs/2604.00557. arXiv:2604.00557v1.

Chongyang Xu, Haipeng Li, Shen Cheng, Jingyu Hu, Haoqiang Fan, Ziliang Feng, and Shuaicheng Liu. Action-geometry prediction with 3d geometric prior for bimanual manipulation. *arXiv preprint arXiv:2602.23814*, 2026. URL https://arxiv.org/abs/2602.23814.

Jiarui Yang, Yehao Lu, Yuning Su, Yufeng Xie, Yu Zhong, Haiyu Lan, Tianjing Hao, Kaixiang Lu, Peiwen Lin, Chuang Wang, Enyu Li, and Junwei Liang. Multi-view unified camera fields: Geometry-shaped action-facing representations for rgb-only multi-camera vla policies. *arXiv preprint arXiv:2608.01826*, 2026a. URL https://arxiv.org/abs/2608.01826. arXiv:2608.01826v1, submitted 2026-08-03.

Sizhe Yang, Linning Xu, Hao Li, Juncheng Mu, Jia Zeng, Dahua Lin, and Jiangmiao Pang. Robo3r: Enhancing robotic manipulation with accurate feed-forward 3d reconstruction, 2026b. URL https://arxiv.org/abs/2602.10101.

Yanjie Ze, Nicklas Hansen, Yinbo Chen, Mohit Jain, and Xiaolong Wang. Visual reinforcement learning with self-supervised 3D representations. *IEEE Robotics and Automation Letters*, 2023a. URL https://arxiv.org/abs/2210.07241.

Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. GNFactor: Multi-task real robot learning with generalizable neural feature fields. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 284–301. PMLR, 2023b. URL https://proceedings.mlr.press/v229/ze23a.html.

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*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.067.

Kaizhao Zhang, Tian Niu, Tianyu Liu, Chenen Guo, Zijun Xu, Qingda Hu, and Wenchao Ding. DiffuView: Multi-view diffusion pretraining for 3d aware robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_DiffuView_Multi-View_Diffusion_Pretraining_for_3D_Aware_Robotic_Manipulation_CVPR_2026_paper.html.

Xuening Zhang, Qi Lv, Xiang Deng, Miao Zhang, Xingbo Liu, and Liqiang Nie. Cortical policy: A dual-stream view transformer for robotic manipulation. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2603.21051.

Zijian Zhang, Yuqing Jiang, Weitao Zhou, Minglei Li, Jinhao Zhang, Yao Mu, Xiaofan Li, Hao Zhao, and Haibao Yu. GaussianWAM: Distilling geometry and semantics from 3D gaussian fields into world-action models. *arXiv preprint arXiv:2608.24714*, 2026c. URL https://arxiv.org/abs/2608.24714.

Juntu Zhao, Wenbo Lu, Di Zhang, Yufeng Liu, Yushen Liang, Tianluo Zhang, Yifeng Cao, Junyuan Xie, Yingdong Hu, Shengjie Wang, Junliang Guo, Dequan Wang, and Yang Gao. Do you need proprioceptive states in visuomotor policies? *arXiv preprint arXiv:2509.18644*, 2025.

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*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Yupeng Zheng, Xiang Li, Songen Gu, Yuhang Zheng, Shuai Tian, Weize Li, Linbo Wang, Chaoyue Li, Qichao Zhang, Haoran Li, Zhongpu Xia, Ya-Qin Zhang, Shuicheng Yan, and Dongbin Zhao. GIFT: Guided intermediate feature training via action-oriented structural supervision for robotic manipulation. *arXiv preprint arXiv:2609.04193*, 2026. URL https://arxiv.org/abs/2609.04193.

Allan Zhou, Moo Jin Kim, Lirui Wang, Pete Florence, and Chelsea Finn. NeRF in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Zhou_NeRF_in_the_Palm_of_Your_Hand_Corrective_Augmentation_for_CVPR_2023_paper.html.

Haoyi Zhu, Honghui Yang, Yating Wang, Jiange Yang, Limin Wang, and Tong He. SPA: 3d spatial-awareness enables effective embodied representation. *arXiv preprint arXiv:2410.08208*, 2024. URL https://arxiv.org/abs/2410.08208. arXiv:2410.08208v3.

Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3D representations. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3418–3433. PMLR, 2023. URL https://proceedings.mlr.press/v229/zhu23b.html.

A Additional Related work

Structured 3D Manipulation. Zhu et al. (2023) use object-centric 3D representations to improve robustness to background and camera changes. Lift3D reconstructs depth in task-relevant image regions before adapting a pretrained 2D model to encode point clouds (Jia et al., 2024). PointVLA injects point-cloud features into a pretrained vision-language-action model (Li et al., 2025a). These methods retain explicit 3D representations for policy inference, whereas LEAP uses geometry as training supervision for a policy that takes RGB images and proprioception as inputs.

Other Routes to Spatial Supervision. Ze et al. (2023a) pretrain voxel-based 3D representations before visual reinforcement learning, while Zhou et al. (2023) use NeRF-synthesized wrist views and corrective actions to augment demonstrations for an RGB policy. Depth Helps trains with depth and predicts virtual depth from RGB at inference (Pang et al., 2024). ReconVLA instead reconstructs image regions around manipulated objects to focus visual attention (Song et al., 2025). LEAP uses full and view-matched partial point-cloud reconstruction as auxiliary supervision during policy training.

Camera Fusion and View-Specific Features. Jangir et al. (2022) combine external and wrist images through cross-view attention. Multi-view Disentanglement learns both shared and camera-specific features to support deployment with fewer views (Dunion & Albrecht, 2024). Cortical Policy combines geometrically aligned static-view features with a dynamic egocentric stream (Zhang et al., 2026b). LEAP addresses a complementary question by matching its partial geometric target to the head–wrist views retained during masking.

State Conditioning and Geometric Prediction. de Haan et al. (2019) show how predictive observation cues can lead imitation policies to learn incorrect causal relationships. Robo3R uses RGB images and robot state to reconstruct metric scene geometry for downstream manipulation (Yang et al., 2026b). In LEAP, reconstruction serves as supervision for the visual encoder, so the auxiliary decoder excludes robot state while the action predictor retains it.

B Theoretical Analysis

B.1 Proof of Proposition 4.1

Proposition B.1 (State-induced non-identifiability)**.** *Let $d : \mathcal{G} \times \mathcal{G} \to [0, \infty)$ be a measurable reconstruction cost, with prediction first and target second; no metric properties are required. For the population risk in Equation* (4)*, with the infimum taken over all measurable decoders,*

$$\mathcal{R}_d(G \mid Z, S) \le \mathcal{R}_d(G \mid Z). \tag{13}$$

For $\mathcal{G} = \mathbb{R}^m$, $\mathbb{E}[\|G\|_2^2] < \infty$, and squared Euclidean distortion $d_2(\widehat{G}, G) = \|\widehat{G} - G\|_2^2$,

$$\mathcal{R}_2(G \mid Z) - \mathcal{R}_2(G \mid Z, S) = \mathbb{E}\Big[\|\mathbb{E}[G \mid Z, S] - \mathbb{E}[G \mid Z]\|_2^2\Big]. \tag{14}$$

Proof. Every measurable visual-only decoder $g(Z)$ can be represented by a state-conditioned decoder $\widetilde{g}(Z, S) = g(Z)$. Taking infima over the respective decoder classes yields Equation (5).

Under squared Euclidean distortion, the finite-second-moment assumption ensures that the optimal predictions are the conditional means $\mu_Z = \mathbb{E}[G \mid Z]$ and $\mu_{ZS} = \mathbb{E}[G \mid Z, S]$. Since $\mathbb{E}[G - \mu_{ZS} \mid Z, S] = 0$, the cross term in the squared-norm expansion of $G - \mu_Z = (G - \mu_{ZS}) + (\mu_{ZS} - \mu_Z)$ has zero expectation. Therefore,

$$\begin{aligned}\mathbb{E}\big[\|G - \mu_Z\|_2^2\big] = \mathbb{E}\big[\|G - \mu_{ZS}\|_2^2\big] \\ + \mathbb{E}\big[\|\mu_{ZS} - \mu_Z\|_2^2\big].\end{aligned} \tag{15}$$

Substituting the corresponding optimal risks proves Equation (6). □

B.2 Proof of Proposition 4.2

Proposition B.2 (Non-identifiability of view-wise contributions)**.** *Let I_h, I_l, and I_r denote observations from the head, left-wrist, and right-wrist cameras, respectively, and let G denote the target geometry. Suppose a head-view encoder f_h and decoder g_h satisfy*

$$\mathbb{E}[d(g_h(f_h(I_h)), G)] \le \varepsilon.$$

Consider full-view models with concatenated embeddings. If their hypothesis classes admit constant wrist encoders and a decoder that applies g_h to the head embedding alone, then there exists an admissible pair (F, g) satisfying

$$F(I_h, I_l, I_r) = [f_h(I_h), c_l, c_r], \quad \mathbb{E}[d(g(F(I_h, I_l, I_r)), G)] \le \varepsilon,$$

Proof. Let $\pi_h(i_h, i_l, i_r) = i_h$ be the projection onto the head observation, and define

$$f^\star := f_h \circ \pi_h, \qquad g^\star := g_h.$$

By the model-class assumption, $(f^\star, g^\star)$ is an admissible full-view encoder–decoder pair. Its reconstruction risk satisfies

$$\begin{aligned}
&\mathbb{E}[d(g^\star(f^\star(I_h, I_l, I_r)), G)] \\
&\quad = \mathbb{E}[d(g_h(f_h(\pi_h(I_h, I_l, I_r))), G)] \\
&\quad = \mathbb{E}[d(g_h(f_h(I_h)), G)] \le \varepsilon.
\end{aligned}$$

Moreover, for any fixed i_h and any wrist observations (i_l, i_r) and (i'_l, i'_r),

$$f^\star(i_h, i_l, i_r) = f_h(i_h) = f^\star(i_h, i'_l, i'_r).$$

Thus, $f^\star(I_h, I_l, I_r)$ is measurable with respect to $\sigma(I_h)$ and invariant to both wrist inputs, establishing the claimed solution. □

C Implementation Details

C.1 Task Category of RoboTwin

We divide the RoboTwin 50 tasks in the following manner:

- **Object-centric Spatial Manipulation (14):** `adjust_bottle`, `move_pillbottle_pad`, `move_stapler_pad`, `place_bread_basket`, `place_bread_skillet`, `place_container_plate`, `place_empty_cup`, `place_fan`, `place_mouse_pad`, `place_object_scale`, `place_object_stand`, `place_phone_stand`, `place_shoe`, `rotate_qrcode`.
- **Multi-object Spatial Reasoning (14):** `blocks_ranking_rgb`, `blocks_ranking_size`, `move_can_pot`, `place_a2b_left`, `place_a2b_right`, `place_burger_fries`, `place_can_basket`, `place_cans_plasticbox`, `place_dual_shoes`, `place_object_basket`, `stack_blocks_three`, `stack_blocks_two`, `stack_bowls_three`, `stack_bowls_two`.
- **Interaction / Articulation (14):** `beat_block_hammer`, `click_alarmclock`, `click_bell`, `dump_bin_bigbin`, `handover_block`, `handover_mic`, `hanging_mug`, `open_laptop`, `open_microwave`, `press_stapler`, `put_object_cabinet`, `scan_object`, `stamp_seal`, `turn_switch`.
- **Less-geometry-dependent (8):** `grab_roller`, `lift_pot`, `move_playingcard_away`, `pick_diverse_bottles`, `pick_dual_bottles`, `put_bottles_dustbin`, `shake_bottle`, `shake_bottle_horizontally`.

C.2 Experimental Details

Task Overview. Figure 4 illustrates the three evaluation settings. RoboTwin provides diverse bimanual manipulation tasks, while ManiSkill covers single-arm tasks including stacking, insertion, and tool use. Our real-world evaluation comprises three tasks—Pick Place, Assemble, and Handover Blocks—that test object placement, precise assembly, and coordination between arms, respectively.

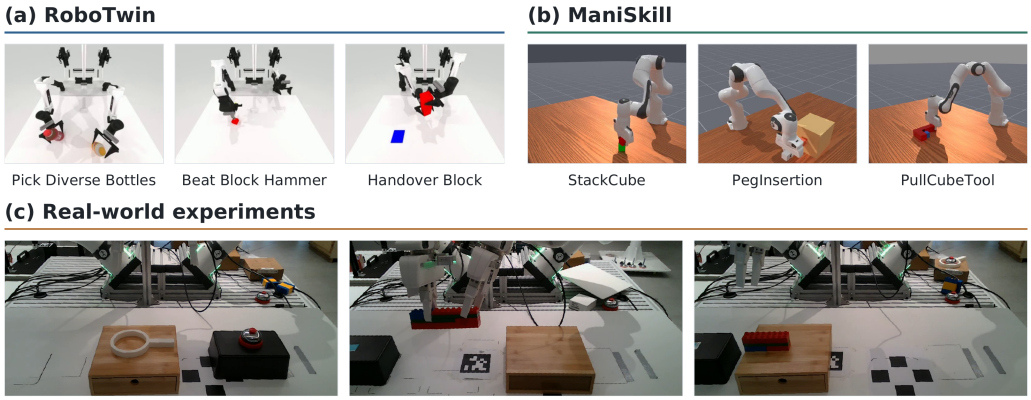


Figure 4: Evaluation environments. (a) Representative bimanual manipulation tasks from RoboTwin. (b) Representative tasks in ManiSkill. (c) Three real-world manipulation setups, including Pick Place, Assemble and Handover Blocks. Simulation panels show representative frames from benchmark demonstrations.

Table 5: Main hyperparameters for DP-LEAP and ACT-LEAP. Each RoboTwin task provides 50 demonstrations and each Maniskill task provides 50 demonstrations. Point clouds are used only for training supervision; both policies use RGB images and robot state at inference.

Hyperparameter	**DP-LEAP**	**ACT-LEAP**
Policy inputs	Head, left-wrist, and right-wrist RGB; 14-D robot state	
Visual encoder	Independent ResNet-18 per camera (GroupNorm; no pretraining)	
View fusion	2-layer, 8-head transformer (width 512)	
Geometry supervision	Full and partial 1024-point world-frame targets; visual-only decoder (width 512)	
View sampling	Head retained; one wrist sampled for partial supervision	
Geometry loss	$0.5\mathcal{L}_{\text{full}} + 0.5\mathcal{L}_{\text{partial}}$	$1.5\mathcal{L}_{\text{full}} + 1.5\mathcal{L}_{\text{partial}}$
Policy backbone	Conditional 1-D U-Net (widths 256/512/1024)	CVAE Transformer (hidden width 512; latent width 32; 4 encoder and 7 decoder layers)
Action prediction	8-step horizon; 3 observations; execute 6 steps	50-step chunk from 1 observation
Policy objective	Diffusion loss (DDPM; 100 training and inference steps)	$\mathcal{L}_1 + 10\mathcal{L}_{\text{KL}}$
Optimizer (LR; weight decay)	AdamW (10^{-4}; 10^{-6})	AdamW (10^{-5}; 10^{-4})
Batch size	32	8
Training epochs	600	4000

C.3 Hyperparameters

Hyperparameters. We train DP-LEAP and ACT-LEAP on 50 demonstrations per RoboTwin task. Both policies take RGB images from a head camera and two wrist cameras, together with a 14-D robot state. Each camera has an independently trained ResNet-18 encoder, and a two-layer transformer fuses the visual features. During training, we retain the head view and sample one wrist view for partial geometry supervision. A visual-only decoder reconstructs full and partial 1024-point clouds in the world frame; the geometry loss uses a 10-epoch linear warmup. DP-LEAP uses a conditional 1-D U-Net and is trained for 600 epochs, while ACT-LEAP uses a CVAE Transformer and is trained for 4000 epochs. Point clouds and the geometry decoder are used only during training.

Pre-Fusion View Masking. Each camera is processed by an independent ResNet-18 encoder, producing camera-specific features $Z_c = f_{\theta_c}(I_c)$ without cross-view interaction. For partial reconstruction, we apply view masking to these features *before* the fusion layer, retaining the head view

Table 6: Full ablation results on five manipulation tasks, reported as success rates (%).

Variant	BeatBlock Hammer	Handover Blocks	PickDiverse Bottles	PlaceCans Plasticbox	PlaceObject Basket	Average
DP-LEAP (full)	**85**	**89**	**48**	**87**	**74**	**76.6**
w/o wrist dropout	73	59	34	49	61	55.2
w/ state in decoder	68	70	36	64	53	58.2
w/ depth reconstruction	79	83	47	79	67	71.0
w/o target alignment	78	63	39	60	70	62.0
ACT-LEAP (full)	55	**79**	**49**	20	**42**	**49.0**
w/o wrist dropout	49	63	43	13	25	38.6
w/ state in decoder	47	58	28	15	36	36.8
w/ depth reconstruction	**59**	70	43	**24**	31	45.4
w/o target alignment	51	72	47	17	35	44.4

and one randomly selected wrist. The fusion layer then aggregates visual information only from the retained views. Because masking precedes all cross-view interaction, the dropped camera's features cannot reach the partial reconstruction decoder, either directly or indirectly through the retained features. This ensures that the partial prediction is conditioned only on the selected head–wrist observations, rather than on features that have already incorporated information from all cameras.

Data Collection and Evaluation Protocol. We collect 50 expert trajectories per RoboTwin task, 100 expert trajectories per ManiSkill task, and 50 expert trajectories per real-world task. For each simulation task, we evaluate each policy using three independent random seeds, with 100 rollouts per seed, yielding 300 evaluation rollouts per policy and task. We compute the success rate separately for each seed and report the mean and standard deviation across the three seeds. For real-world experiments, we use a single random seed and evaluate each policy over 20 rollouts per task, reporting the percentage of successful trials.

C.4 One-Arm Camera Configuration

For one-arm tasks, we use three RGB views: a head camera h, a wrist camera w, and an additional outside camera o with broad scene coverage. Because either h or o remains available when the other is dropped, we uniformly mask any one of the three cameras:

$$M \in \big\{\{h, w\}, \{o, w\}, \{h, o\}\big\}. \tag{16}$$

When the wrist is retained, we construct the partial target from wrist-visible points, as above. When the wrist is masked, we instead use points approximately visible from the retained outside camera. Thus, the partial target is always aligned with an available view, while the two configurations containing the wrist encourage the policy to use its local geometric observations.

D Additional Experimental Results

D.1 Full Results of Ablation Study

D.2 Sensitivity Analysis

Sensitivity to Geometry Reconstruction Weight. We vary the geometry reconstruction weights λ_{full} and λ_{partial} on five RoboTwin tasks. DP-LEAP achieves its highest average success rate at $\lambda_{\text{full}} = \lambda_{\text{partial}}$=0.5 (76.6%), compared with 74.6% and 73.4% at $\lambda_{\text{full}} = \lambda_{\text{partial}} = 0.4$ and 0.6, respectively. ACT-LEAP performs best on average at $\lambda_{\text{full}} = \lambda_{\text{partial}} = 1.5$ (49.0%), compared with 44.0% at $\lambda_{\text{full}} = \lambda_{\text{partial}} = 1.2$ and 45.6% at $\lambda_{\text{full}} = \lambda_{\text{partial}} = 1.8$. The best weight varies across individual tasks, but the average results support $\lambda_{\text{full}} = \lambda_{\text{partial}} = 0.5$ for DP and $\lambda_{\text{full}} = \lambda_{\text{partial}} = 1.5$ for ACT.

D.3 Full Results on RoboTwin

Table 7: Sensitivity to the geometry reconstruction weight λ. Entries are success rates; Average is the arithmetic mean over five tasks. The best result within each policy group is shown in bold. Default λ_{full} and λ_{partial} for DP-LEAP is 0.5 and for ACT-LEAP is 1.5.

Task	DP-LEAP			ACT-LEAP		
$\lambda_{\text{full}}/\lambda_{\text{partial}}$	0.4	**0.5** (Default)	0.6	1.2	**1.5** (Default)	1.8
BeatBlockHammer	0.79	**0.85**	0.82	0.51	**0.55**	0.52
HandoverBlock	0.83	**0.89**	0.79	0.72	**0.79**	0.69
PickDiverseBottles	0.45	**0.48**	0.41	0.42	0.49	**0.53**
PlaceCansPlasticbox	0.89	0.87	**0.93**	0.17	**0.20**	0.13
PlaceObjectBasket	**0.77**	0.74	0.72	0.38	**0.42**	0.41
Average	0.746	**0.766**	0.734	0.440	**0.490**	0.456

Table 8: Per-task success rates (%) on the 50 RoboTwin 2.0 tasks under clean evaluation. DP, RDT, and π_0 are from the official single-task leaderboard; the other results are from our evaluations.

Task	DP3	DP	DP-MultiView	DP-eVGGT	RDT	DP-SF	π_0	GAP	DP-LEAP	ACT-pointcloud	ACT	ACT-LEAP
adjust_bottle	99	97	95	96	81	99	90	98	99	93	90	92
beat_block_hammer	72	42	27	43	77	60	43	56.3	85	58	43	55
blocks_ranking_rgb	4	0	0	1	3	4	19	3	6	0	1	1
blocks_ranking_size	1	1	0	1	0	2	7	1	2	1	0	1
click_alarmclock	77	61	52	47	61	58	63	64	92	90	35	87
click_bell	97	54	50	56	80	73	44	80	97	88	63	84
dump_bin_bigbin	87	49	43	51	64	59	83	61	89	93	61	89
grab_roller	95	98	80	87	74	89	96	84	95	97	83	96
handover_block	87	10	21	34	45	47	45	68.7	89	89	47	79
handover_mic	99	53	46	59	90	69	98	71	100	99	88	99
hanging_mug	26	8	3	7	23	15	11	23	21	13	8	17
lift_pot	96	39	49	40	72	51	84	63	55	95	85	93
move_can_pot	72	39	27	33	25	53	58	55.3	67	70	30	65
move_pillbottle_pad	42	1	0	12	8	17	21	9	23	37	2	41
move_playingcard_away	69	47	47	38	43	50	53	41	60	74	37	66
move_stapler_pad	8	1	0	1	2	3	0	2	4	1	0	1
open_laptop	75	49	55	50	59	62	85	79	64	85	61	79
open_microwave	88	5	21	43	37	69	80	62	67	66	81	86
pick_diverse_bottles	46	6	12	22	2	37	27	24.3	48	56	3	49
pick_dual_bottles	81	24	25	37	42	56	57	69	74	67	31	63
place_a2b_left	39	2	0	15	3	22	31	26	34	21	1	33
place_a2b_right	37	13	10	14	1	16	27	24.7	21	16	0	23
place_bread_basket	23	14	6	9	10	27	17	20.7	36	20	4	32
place_bread_skillet	37	11	6	18	5	29	23	34	42	26	10	30
place_burger_fries	70	72	51	50	50	56	80	72	70	68	46	70
place_can_basket	69	18	23	32	19	40	41	27	59	15	10	30
place_cans_plasticbox	48	40	31	37	6	55	34	61.3	87	35	8	20
place_container_plate	86	41	41	53	78	71	88	83	74	86	75	88
place_dual_shoes	13	8	3	5	4	10	15	27	29	6	2	9
place_empty_cup	65	37	31	43	56	67	37	76.3	95	59	45	67
place_fan	36	3	3	8	12	12	20	26	27	19	3	27
place_mouse_pad	4	0	0	2	1	3	7	5	9	0	0	1
place_object_basket	65	15	15	23	33	49	16	60.3	74	37	17	42
place_object_scale	16	1	0	4	1	7	10	8	10	11	0	17
place_object_stand	54	22	21	29	15	38	36	42.7	43	29	2	25
place_phone_stand	44	13	13	20	15	40	35	46	49	35	0	38
place_shoe	49	23	23	31	35	37	28	49	45	19	2	28
press_stapler	66	6	13	44	41	75	62	78	93	57	32	65
put_bottles_dustbin	60	22	22	41	21	51	54	61	63	44	35	44
put_object_cabinet	72	42	42	37	33	44	68	32	63	23	10	37
rotate_qrcode	71	13	13	25	50	49	68	59	62	25	0	28
scan_object	27	9	18	23	4	27	18	30.7	37	10	4	7
shake_bottle	98	65	65	73	74	83	97	93.3	94	100	56	97
shake_bottle_horizontally	98	59	59	72	84	69	99	67	98	100	73	96
stack_blocks_three	1	0	0	1	2	3	17	2	4	1	0	1
stack_blocks_two	22	7	3	19	21	22	42	32	47	7	19	25
stack_bowls_three	57	63	49	50	51	51	66	57	63	49	50	57
stack_bowls_two	85	61	61	76	76	72	91	95.3	96	84	79	88
stamp_seal	18	2	2	5	1	9	3	7	10	27	4	19
turn_switch	46	36	30	35	35	33	27	51	56	23	8	20
Average	55.94	28.04	26.14	33.04	34.50	42.80	46.42	47.38	56.54	46.48	28.88	48.14