DUAL-ROLE GEOMETRY, CONTEXTUAL RGB: REPRESENTING POINT CLOUDS FOR 3D ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Colored point clouds provide robot policies with spatial geometry and appearance cues, whose relative importance varies across manipulation tasks. Jointly encoding XYZ and RGB at each point can reduce performance on tasks requiring precise spatial execution, while separate pointwise encoding still lacks explicit interactions among neighboring appearance features. We propose GeoRGB, a 3D robot policy that preserves pointwise spatial references while organizing appearance features through geometry-guided interactions. Its asymmetric dual-tower encoding (ADTE) pairs a dedicated pointwise XYZ pathway with a multiscale RGB hierarchy, separating spatial reference encoding from appearance contextualization before fusion. Within the RGB hierarchy, geometry-guided RGB contextualization (GRC) aggregates neighboring appearance features within and across scales, using XYZ-defined neighborhoods and directional weights derived from relative positions. Across 50 RoboTwin 2.0 tasks, GeoRGB achieves 77.5% average success, improving over the strongest evaluated baseline by 14.3 percentage points. It further achieves 93.8% success across four real-world tasks. On MetaWorld, a deeper variant reaches 75.0% success, matching the best-performing baseline with fewer than half the parameters and less than one-quarter the training time.

1 INTRODUCTION

Recent advances in 3D robot policies have enabled increasingly capable manipulation through spatially grounded visual representations (Gervet et al., 2023; Ke et al., 2025; Goyal et al., 2023; 2024; Yang et al., 2025; Tie et al., 2025; Yan et al., 2025; Zhang et al., 2026). Colored point clouds provide a direct representation of both metric 3D geometry and visual appearance: XYZ coordinates specify spatial locations (Ze et al., 2024; Liu et al., 2023), while RGB cues help identify task-relevant objects and regions (Chen et al., 2023; Shridhar et al., 2022). Manipulation tasks place different demands on these cues, from precise spatial localization to appearance-based target selection, often requiring policies to relate appearance to spatial structure. This motivates studying how geometry and appearance should be organized into point-cloud representations that policies can effectively exploit.

Additional appearance information does not necessarily improve spatial execution. On the simulation task *Stamp Seal*, jointly encoding XYZ and RGB at each point (Joint) yields a lower completion rate than XYZ-only encoding, with the larger gap occurring after grasping (Figure 1a). The two encoders achieve grasp rates of 72.9% and 79.2%, respectively, but their completion rates conditioned on successful grasping are 24.3% and 80.3%. Encoding XYZ and RGB independently before fusion (Separate) raises conditional completion to 51.8% without changing the input modalities. This partial recovery highlights the importance of geometry–appearance organization, motivating a dedicated pointwise XYZ pathway that retains a direct spatial reference alongside appearance features.

Appearance cues are necessary when color determines the required object roles or manipulation order. In Stack Blocks Two, the green block must be placed on the red one. XYZ-only achieves a red-first grasp rate of 46.0% and an overall success rate of 40.0%, whereas Separate raises the red-first grasp rate to 96.0% (Figure 1b). Retaining appearance information thus helps establish the required object order. Separate pointwise encoding, however, does not explicitly relate neighboring appearance features within the encoder. We therefore examine whether organizing these features through geometric neighborhoods can improve subsequent manipulation. GeoRGB retains the pointwise XYZ pathway and introduces geometry-guided context within the RGB branch. It achieves

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.

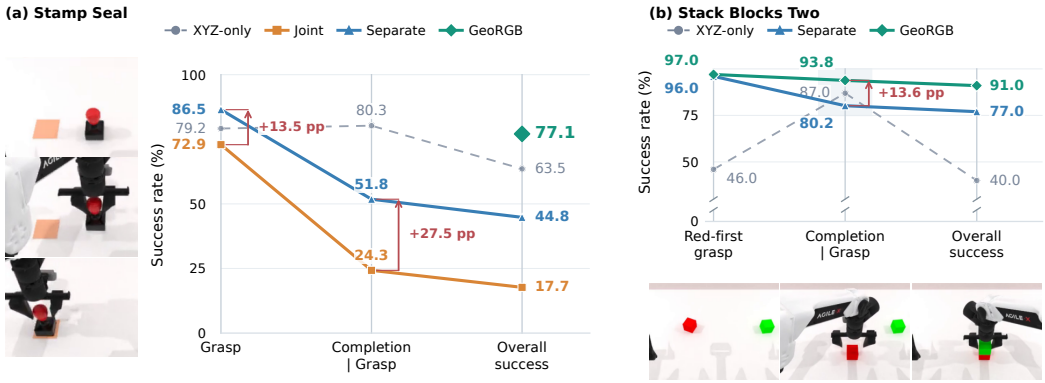

Figure 1: **Stage-wise effects of geometry–appearance encoding.** Joint and Separate denote joint and separate pointwise XYZ–RGB encoding. (a) On *Stamp Seal*, Joint has lower completion after grasping than XYZ-only; Separate partially recovers this performance. (b) On *Stack Blocks Two*, XYZ-only has a lower red-first grasp rate, while GeoRGB improves completion over Separate with a similar red-first grasp rate. Grasp denotes seal grasping in (a) and red-first grasping in (b). Completion | Grasp is final task success conditioned on grasp; Overall success covers all scenes.

a similar red-first grasp rate of 97.0%, while increasing completion conditioned on this grasp from 80.2% to 93.8%. Together, these observations motivate preserving pointwise spatial references while building geometry-guided appearance context to support task-dependent information demands.

Guided by these observations, we propose **GeoRGB**, a 3D robot policy that assigns distinct encoding roles to spatial coordinates and appearance attributes. Its **asymmetric dual-tower encoding (ADTE)** retains a pointwise XYZ pathway to provide direct spatial references, while a multiscale RGB pathway builds context for appearance cues. This asymmetry allows appearance features to incorporate neighboring information while retaining an independently encoded spatial reference at each point. Within the RGB pathway, **geometry-guided RGB contextualization (GRC)** uses spatial neighborhoods and relative positions to organize appearance interactions within and across scales. Geometry thus serves both as an explicit policy input and as the structure through which appearance context is formed. The two pathways are fused at corresponding points to condition a flow-matching action generator, providing a common encoding design for tasks with different geometric and appearance requirements. Our contributions are threefold:

- Through controlled encoding comparisons and stage-wise behavioral analysis, we characterize how encoding choices shape the benefits and costs of incorporating appearance across manipulation tasks with different information requirements.
- We propose GeoRGB, which separates pointwise spatial reference encoding from geometry-guided appearance contextualization. ADTE and GRC implement this division of roles in a common encoder architecture for tasks with differing information demands.
- We validate GeoRGB across RoboTwin 2.0, MetaWorld, and real-world tasks, demonstrating strong overall performance and a favorable performance–cost balance in multi-task learning. Ablations further reveal the complementary roles of pointwise spatial references and geometry-guided context, alongside task-dependent benefits of appearance cues.

2 Related Work

3D Visuomotor Policies. Explicit 3D representations connect visual observations with spatial action prediction through voxelized workspaces (Shridhar et al., 2023; James et al., 2022), point-based feature fields (Gervet et al., 2023), and rendered multi-view representations (Goyal et al., 2023; 2024). For diffusion-based and flow-based action generation, 3D scene information can be summarized into compact global conditions (Ze et al., 2024; Chisari et al., 2025) or retained as spatially grounded features that support interaction with action queries (Ke et al., 2025; Yan et al., 2025). GeoRGB

studies how geometry and appearance should be organized in point-level representations to support manipulation tasks with differing information demands.

Geometry-Guided Appearance Encoding. Geometric guidance can shape appearance encoding through aggregation weights, spatial neighborhoods, and feature transformations. Depth-dependent weighting modulates local convolutional aggregation (Wang & Neumann, 2018; Chen et al., 2021), while depth and spatial priors reweight visual self-attention (Yin et al., 2025). Geometry-aware receptive fields and sampling schemes organize visual context according to spatial structure (Chen et al., 2019; Xing et al., 2020; Wu et al., 2020). Geometry-conditioned feature transformations use depth-derived features to generate dynamic filters (Ding et al., 2020) or multiscale local linear transformations (Wang et al., 2022). GeoRGB combines geometry-guided appearance encoding with an independent pointwise XYZ pathway, allowing geometry to organize RGB feature interactions while remaining directly available as a spatial reference for action prediction.

Point-Cloud Encoding. Point-cloud encoders build contextual representations through local feature interactions, often organized hierarchically across scales. Shared MLPs followed by symmetric pooling summarize geometric neighborhoods into local descriptors (Qi et al., 2017; Qian et al., 2022). Kernel-based convolutions structure feature transformations according to relative point locations, with depthwise variants reducing the cost of local encoding (Thomas et al., 2019; 2024). Attention-based encoders learn content-dependent interactions within local neighborhoods or point groups organized by serialization (Zhao et al., 2021; Wu et al., 2024). Hierarchical point-cloud representations have also been used for language-guided manipulation (Chen et al., 2023). GRC adapts depthwise kernel point aggregation to encode RGB context within and across scales.

3 GeoRGB Policy

As illustrated in Figure 2, GeoRGB encodes colored point clouds into pointwise representations combining independently encoded spatial references with contextual appearance features. Together with robot states, these representations condition a flow-matching action generator.

3.1 Problem Formulation

We consider learning manipulation policies from demonstrations, with colored point clouds and robot states as observations. At time t, let $\mathcal{P}_t = \{(\mathbf{x}_{t,i}, \mathbf{c}_{t,i})\}_{i=1}^{N}$ denote the sampled point cloud supplied to the policy, where $\mathbf{x}_{t,i} \in \mathbb{R}^3$ and $\mathbf{c}_{t,i} \in \mathbb{R}^3$ are the metric coordinates and RGB color of point i, respectively. The policy also observes the robot state $\mathbf{q}_t$. Conditioned on the latest T_o observations, the policy predicts an H-step action chunk $\mathbf{A}_t = (\mathbf{a}_{t-T_o+1}, \ldots, \mathbf{a}_{t-T_o+H})$. The prediction window starts at the earliest timestep in this observation history; execution begins with $\mathbf{a}_t$ at the current timestep. Here, H is the prediction horizon and $\mathbf{a}_t$ is a robot control vector. The conditional action distribution is modeled as:

$$\mathbf{Z}_\tau = E(\mathcal{P}_\tau), \qquad \mathbf{A}_t \sim p_\theta(\cdot \mid \mathbf{Z}_{t-T_o+1:t}, \mathbf{q}_{t-T_o+1:t}). \tag{1}$$

Here, E is the point cloud encoder, $\mathbf{Z}_\tau$ contains its pointwise output features at time τ, and p_θ is the conditional action distribution parameterized by θ.

3.2 Asymmetric Colored Point Cloud Encoder

Asymmetric dual-tower encoding (ADTE). Figure 2a outlines ADTE. It first encodes normalized XYZ coordinates and RGB colors using separate pointwise feature extractors, $\mathbf{g}_i = f_{\text{geo}}(\mathbf{x}_i)$ and $\mathbf{h}_i^{(0)} = f_{\text{rgb}}(\mathbf{c}_i)$. The XYZ features $\mathbf{g}_i$ are retained as direct spatial references, while the RGB features are further processed by a multiscale hierarchy. Within this hierarchy, GRC integrates appearance features under geometric guidance to produce a contextual RGB feature $\mathbf{h}_i$ at each input point. Geometry thus provides both an independently encoded spatial reference and a structural guide for appearance contextualization. Pointwise concatenation ($\|$) of the two streams forms the final point-cloud representation:

$$\mathbf{Z}_t = E(\mathcal{P}_t) = \left[\mathbf{g}_i \parallel \mathbf{h}_i\right]_{i=1}^{N}, \tag{2}$$

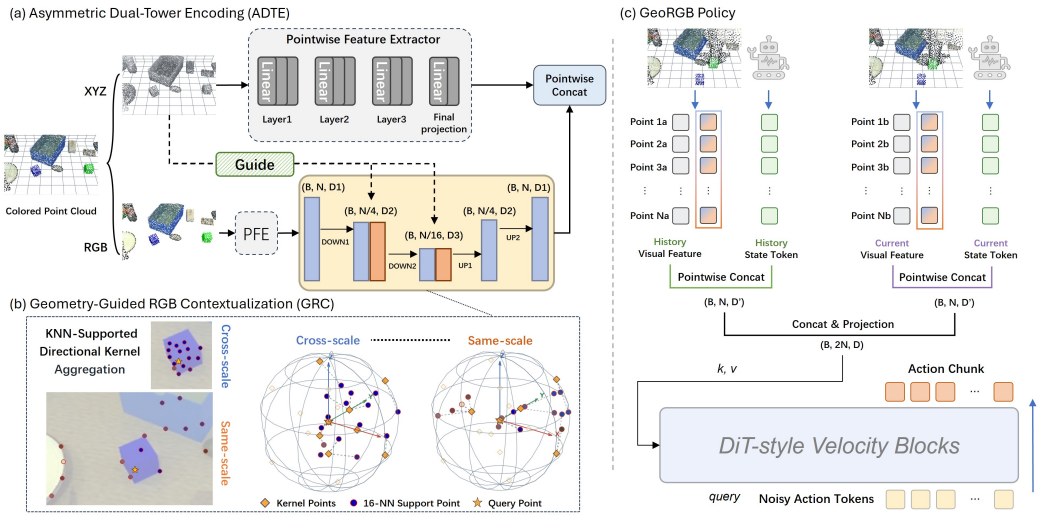

Figure 2: **Overview of GeoRGB.** (a) ADTE pairs an independent pointwise XYZ pathway with a multiscale RGB pathway, fusing their features at corresponding input points. (b) GRC aggregates RGB features within and across scales using XYZ-defined neighborhoods and relative-position-dependent weights. (c) For each frame, point features are concatenated with its robot-state embedding, and the resulting tokens across observation frames condition a flow-matching action generator.

Geometry-guided RGB contextualization (GRC). Figure 2b illustrates how GRC uses geometry to determine which appearance features interact and how their contributions are weighted. For each query point i, its context neighborhood $\mathcal{N}_{\mathbf{X}}(i)$ consists of the K nearest support points in XYZ space. In *cross-scale aggregation*, farthest-point sampling (FPS) selects query points at a coarser resolution, while support points come from the preceding resolution. In *same-scale aggregation*, query and support points both come from the current resolution. Repeating these operations across scales expands the spatial context of the RGB features.

Within each neighborhood, relative positions guide the weighting of appearance features. Both cross-scale and same-scale aggregation use a depthwise kernel point operator adapted from KPConv (Thomas et al., 2019; 2024). The aggregated feature for query i is

$$\mathbf{u}_i = \sum_{j \in \mathcal{N}_{\mathbf{X}}(i)} \alpha_{ij} \big(P(\mathbf{h}_{\mathrm{sup},j}) \odot \mathbf{w}_{\pi_{ij}} \big), \tag{3}$$

where $\mathbf{h}_{\mathrm{sup},j}$ is the RGB feature of support point j at the current aggregation stage, P is a learned linear projection, and $\odot$ denotes elementwise multiplication. The index π_{ij} identifies the fixed kernel point nearest to the normalized relative offset, selecting its learned channel-wise weight $\mathbf{w}_{\pi_{ij}}$; α_{ij} is a distance-based influence. A projected query feature is added to the aggregate, followed by normalization and a residual pointwise MLP.

To align the multiscale appearance context with the pointwise XYZ features, we restore the RGB features to the input resolution through three-neighbor distance-weighted interpolation and skip fusion (Qi et al., 2017). At each finer level, a pointwise MLP combines the interpolated features with the RGB features retained at that resolution.

3.3 Action Flow Matching with Dense Point Context

Figure 2c depicts how pointwise scene information and robot state condition the action generator. We construct one conditioning token per sampled point in each observation frame. Each point feature is concatenated with an MLP embedding of the corresponding robot state, projected to dimension D, and augmented with learned context embeddings. Tokens from the T_o observation frames are concatenated into the conditioning sequence $\mathbf{Z}_{\mathrm{ctx}} \in \mathbb{R}^{T_o N \times D}$.

Given an action chunk $\mathbf{A}^s$ at flow time s, a DiT-style transformer (Peebles & Xie, 2023) predicts the conditional velocity field $v_\theta(\mathbf{A}^s, s; \mathbf{Z}_{\text{ctx}})$. The action vectors in $\mathbf{A}^s$ are embedded into tokens, using self-attention for action interaction and cross-attention to the pointwise observation context.

During training, we pair a normalized demonstration action chunk $\mathbf{A}^1$ with independent Gaussian noise $\mathbf{A}^0 \sim \mathcal{N}(0, \mathbf{I})$. We sample $s \sim p(s)$, where $p(s)$ places greater probability near the noise endpoint $s = 0$. The interpolated action chunk $\mathbf{A}^s = (1 - s)\mathbf{A}^0 + s\mathbf{A}^1$ serves as the network input, with target velocity $\mathbf{A}^1 - \mathbf{A}^0$ along this linear path. We train the velocity field jointly with the encoder using conditional flow matching (Lipman et al., 2023):

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}\left[\left\| v_\theta(\mathbf{A}^s, s; \mathbf{Z}_{\text{ctx}}) - (\mathbf{A}^1 - \mathbf{A}^0) \right\|_{\text{F}}^2\right]. \tag{4}$$

At inference, we generate an action chunk by integrating the learned velocity field from Gaussian noise $\mathbf{A}^0$ at $s = 0$ to $s = 1$. We use Euler integration while keeping the observation context fixed, then unnormalize the resulting actions and execute them from the current timestep onward.

4 Experiments

Implementation details. GeoRGB uses two observation frames with 512 sampled points per frame. The default action generator has four DiT-style blocks with hidden dimension 512 and eight attention heads, with deeper variants obtained by increasing the number of blocks. We train separate policies for individual tasks on RoboTwin 2.0 and the real robot, and a single multi-task policy conditioned on task IDs on MetaWorld. Prediction horizons are 16, 4, and 32 for RoboTwin 2.0, MetaWorld, and the real-world experiments, respectively. For both simulation and real-world experiments, we train for 600 epochs using AdamW and BF16 mixed precision, with point-cloud color jitter during training. At inference, we use exponential moving average (EMA) weights and ten Euler integration steps to generate action chunks. Before replanning, we execute 15 actions on RoboTwin 2.0, 3 on MetaWorld, and 31 on the real robot.

Table 1: RoboTwin 2.0 C2C success rates (%) on seven tasks. Subscripts indicate standard deviations where available. The Δ row reports GeoRGB's difference from the strongest baseline in each column, in percentage points.

Method	Lift Pot	Open Laptop	Pick Dual Bottles	Stamp Seal	Put Object Cabinet	Stack Blocks Two	Blocks Ranking RGB	**Avg. Success**
DP	39.0	49.0	24.0	2.0	42.0	7.0	0.0	23.3
DP3	97.0	82.0	60.0	18.0	**72.0**	24.0	3.0	50.9
Hyper-DP3	95.0	85.0	90.0	43.0	63.0	41.0	3.0	60.0
π_0	84.0	85.0	57.0	3.0	68.0	42.0	19.0	51.1
ManiFlow	$97.3_{\pm 1.5}$	$86.7_{\pm 2.1}$	$99.0_{\pm 1.0}$	$28.7_{\pm 3.8}$	$49.3_{\pm 5.1}$	$77.7_{\pm 3.2}$	$42.3_{\pm 7.2}$	68.7
GeoRGB	$\mathbf{99.0}_{\pm 1.0}$	$\mathbf{88.0}_{\pm 1.0}$	$\mathbf{99.7}_{\pm 0.6}$	$\mathbf{80.7}_{\pm 4.0}$	$55.0_{\pm 3.5}$	$\mathbf{92.3}_{\pm 1.2}$	$\mathbf{83.7}_{\pm 3.8}$	**85.5**
Δ	**+1.7**	**+1.3**	**+0.7**	**+37.7**	**−17.0**	**+14.7**	**+41.3**	**+16.8**

4.1 Benchmark Evaluations

Evaluation on RoboTwin 2.0. RoboTwin 2.0 (Chen et al., 2026) provides diverse manipulation tasks with configurable scene randomization. We evaluate GeoRGB on 50 tasks under the clean-to-clean (C2C) setting (Table 5) and examine seven of these tasks under both C2C (Table 1) and randomized-to-randomized (R2R) settings (Figure 3). For each task and setting, we train GeoRGB with **three independent training seeds** using 50 demonstrations from the corresponding distribution. Each trained policy is evaluated for 100 episodes using one fixed evaluation seed; we report the mean and standard deviation across training seeds. The 50-task C2C comparison includes DP (Chi et al., 2025), DP3 (Ze et al., 2024), Hyper-DP3 (Zhang et al., 2026), and π_0 (Black et al., 2025). ManiFlow (Yan et al., 2025) is additionally evaluated on the seven-task C2C subset and serves as the baseline for R2R. DP3, Hyper-DP3, ManiFlow, and GeoRGB use point clouds as visual input, whereas DP and π_0 use RGB images.

The seven-task subset retains four tasks from ManiFlow's R2R evaluation: *Lift Pot*, *Open Laptop*, *Pick Dual Bottles*, and *Put Object Cabinet*. We add *Stamp Seal*, *Stack Blocks Two*, and *Blocks*

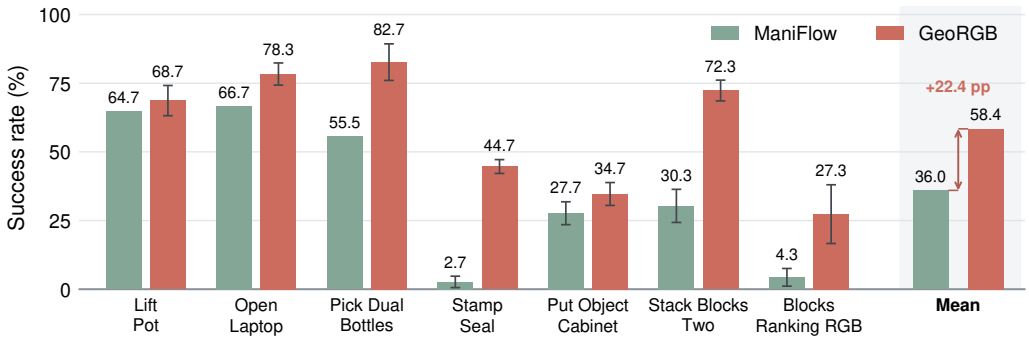

Figure 3: Per-task and average success rates (%) on seven RoboTwin 2.0 tasks under the R2R setting. Error bars show ± 1 standard deviation across three independent training seeds, each evaluated using one fixed evaluation seed.

Ranking RGB to cover precise localization, alignment, and color-dependent manipulation. *Stamp Seal* requires grasping a seal and aligning it with a target region; the two block tasks require distinguishing geometrically similar objects by color to satisfy a prescribed stacking or left-to-right order. Together, these tasks support examining policy performance under different geometric and appearance requirements.

Across all 50 C2C tasks, GeoRGB achieves **77.5%** average success, exceeding the strongest evaluated baseline by 14.3 percentage points. It also achieves the highest average success on the seven-task subset, with **85.5%** in C2C and **58.4%** in R2R. Under C2C, GeoRGB achieves 80.7% success on *Stamp Seal* and 83.7% on *Blocks Ranking RGB*, supporting both precise local manipulation and color-dependent ordering with the same encoder design. Both methods achieve near-perfect C2C success on *Pick Dual Bottles*; GeoRGB outperforms ManiFlow by 27.2 percentage points under R2R.

Evaluation on MetaWorld. MetaWorld (Yu et al., 2019) extends the evaluation to learning multiple tasks within a single policy. We use the same 48-task subset as ManiFlow and train a multi-task GeoRGB policy with three independent training seeds, using ten expert demonstrations per task and task IDs as conditioning signals. For each training seed, a single checkpoint is evaluated across all 48 tasks, with 20 episodes per task using one fixed evaluation seed. In our comparison, DP3 receives XYZ-only point clouds, while ManiFlow and GeoRGB receive XYZRGB point clouds.

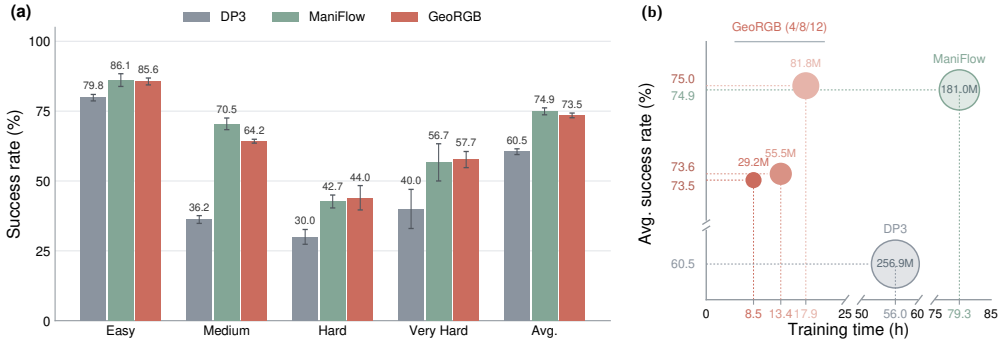

Figure 4: Multi-task performance and computational cost on MetaWorld. (a) Success rates by task difficulty and overall for the default four-block GeoRGB model and baselines; error bars indicate one standard deviation. (b) Average success versus training time for GeoRGB variants with 4, 8, and 12 DiT blocks and baselines. Bubble area represents parameter count, annotated in millions.

GeoRGB extends to multi-task learning with a favorable balance between success rate and computational cost. The default four-block model achieves 73.5% average success, compared with 60.5% for DP3 and 74.9% for ManiFlow (Figure 4a). The default model uses 29.2M parameters and trains in 8.5 hours, compared with 181.0M parameters and 79.3 hours for ManiFlow (Figure 4b). The twelve-block

variant achieves 75.0% average success, matching ManiFlow's 74.9%, while retaining fewer than half its parameters (81.8M) and less than one quarter the total training time (17.9 hours). All methods are trained on the same two GPUs using BF16 precision and a batch size of 384. GeoRGB and its deeper variants are trained for 600 epochs, while ManiFlow is trained for 2,000 epochs. Reported training times cover each method's full training schedule. Together, these results demonstrate GeoRGB's applicability to multi-task manipulation and a favorable balance between success rate, model size, and total training cost.

4.2 Ablation Studies

Figure 5 shows the two RoboTwin 2.0 C2C tasks used to study how spatial references and appearance context contribute to manipulation. *Stamp Seal* emphasizes precise localization and alignment, whereas *Stack Blocks Two* additionally requires color-based selection of geometrically identical blocks for green-over-red stacking. We hold the action-generator architecture and training budget fixed across variants.



Figure 5: Ablation tasks.

We examine ADTE through three sets of comparisons reported in Table 2. First, *Joint pointwise encoding* and *Separate pointwise encoding* are compared with GeoRGB to examine modality separation and geometry-guided contextual interaction. Second, *w/o pointwise XYZ branch* and *Constant RGB input* probe the roles of direct spatial references and color distinctions, with *XYZ-only encoding* providing a geometry-only reference. Finally, *XYZ + contextual XYZRGB* supplies coordinates alongside colors to the contextual branch, while *w/o color jitter* removes color augmentation during training. These variants examine contextual input choice and color augmentation, respectively.

Table 2: Mean success rates (%) for representation and color-augmentation ablations in RoboTwin 2.0 C2C. Avg. is the mean over both tasks.

Variant	Stamp Seal	Stack Blocks Two	**Avg.**
Joint pointwise encoding	$22.0_{\pm 3.5}$	$75.3_{\pm 3.5}$	48.7
Separate pointwise encoding	$50.7_{\pm 4.9}$	$78.0_{\pm 3.6}$	64.3
XYZ-only encoding	$67.7_{\pm 4.9}$	$41.0_{\pm 1.0}$	54.3
w/o pointwise XYZ branch	$17.7_{\pm 4.7}$	$19.0_{\pm 1.7}$	18.3
Constant RGB input	$\mathbf{89.0}_{\pm 1.7}$	$46.3_{\pm 2.5}$	67.7
XYZ + contextual XYZRGB	$67.3_{\pm 2.1}$	$\mathbf{93.0}_{\pm 3.0}$	80.2
w/o color jitter	$68.0_{\pm 2.6}$	$92.7_{\pm 1.2}$	80.3
GeoRGB	$80.7_{\pm 4.0}$	$92.3_{\pm 1.2}$	**86.5**

Modality separation and contextual interaction. Separate pointwise encoding improves average success from 48.7% to 64.3% over joint encoding. However, its success on *Stamp Seal* remains below XYZ-only encoding (50.7% versus 67.7%), indicating that separating modalities alone does not resolve the performance loss associated with appearance input on this task. With geometry-guided RGB context, GeoRGB raises success to 80.7% on *Stamp Seal* and 92.3% on *Stack Blocks Two*, compared with 50.7% and 78.0% for separate pointwise encoding. These results support organizing interactions among appearance features alongside independently encoded spatial references.

Spatial references and task-dependent appearance. Removing the pointwise XYZ branch leaves geometry-guided RGB aggregation intact but reduces average success from 86.5% to 18.3%. Thus, using geometry to guide contextual interactions does not substitute for directly encoding pointwise spatial references in this architecture. Fixing RGB inputs to zero further separates the contribution of geometric context from that of color distinctions: the contextual branch remains active, with geometry determining its neighborhoods and aggregation weights. On *Stamp Seal*, this variant achieves 89.0% success versus 67.7% for XYZ-only encoding, supporting the value of geometric context even without informative colors. Its higher success than GeoRGB (80.7%) also indicates that retaining color still incurs a performance cost on this task. On *Stack Blocks Two*, however, fixing RGB reduces success from 92.3% to 46.3%, reflecting the importance of color distinctions for the required object roles.

These contrasting outcomes distinguish the value of geometric context from the task-dependent benefit of color information. By retaining direct spatial references and contextualizing appearance, GeoRGB accommodates both requirements and achieves the highest two-task average among the evaluated variants.

Contextual input choice. We examine whether supplying coordinates as input features provides additional benefits beyond their use in guiding RGB contextualization. With the pointwise XYZ branch retained, contextual XYZRGB achieves 67.3±2.1% success on *Stamp Seal*, compared with GeoRGB's 80.7±4.0%, while performance on *Stack Blocks Two* remains close (93.0±3.0% versus 92.3±1.2%). Adding coordinate attributes therefore offers no consistent advantage across these tasks. GeoRGB is designed to preserve direct spatial references through independent pointwise XYZ encoding, while using geometry to determine how neighboring appearance features interact. This organization gives appearance features access to spatial context without requiring the contextual branch to jointly encode XYZ and RGB attributes.

Table 3: Ablations of directional aggregation and same-scale interaction in GRC on RoboTwin 2.0 C2C. Avg. is the mean over both tasks.

Variant	Stamp Seal	Stack Blocks Two	**Avg.**
Single central-kernel aggregation	$61.7_{\pm 7.5}$	$85.7_{\pm 0.6}$	73.7
w/o same-scale interaction	$79.0_{\pm 3.0}$	$\mathbf{92.7}_{\pm 2.5}$	85.8
GeoRGB	$\mathbf{80.7}_{\pm 4.0}$	$92.3_{\pm 1.2}$	**86.5**

Aggregation within GRC. We examine the roles of directional weighting and same-scale interaction in constructing RGB context using the comparisons in Table 3. Replacing directional kernels with a single kernel at the query location preserves neighborhood aggregation, but weights neighboring features by distance without distinguishing their directions. This lowers average success from 86.5% to 73.7%, with decreases on both tasks. Disabling same-scale neighborhood interaction while retaining pointwise updates yields 85.8% average success, compared with 86.5% for GeoRGB.

4.3 Real-World Evaluation

We conduct real-world experiments on an XLeRobot dual-arm platform equipped with a head-mounted Intel RealSense D435i RGB-D camera. Aligned RGB and depth images are converted into colored point clouds in the camera frame using the camera intrinsics. After image-region cropping, depth filtering, and table-plane removal, we construct a set of 4096 candidate points per frame. Within GeoRGB, these candidates are normalized and downsampled to 512 points using farthest point sampling before encoding. Figure 6 visualizes these preprocessing stages.



Figure 6: Point-cloud preprocessing for *Pick Carrot Red Bowl*. Left: RGB observation. Middle: pixels retained after cropping, depth filtering, and table removal. Right: the 512 points selected by GeoRGB, projected into the camera view.

Figure 7 presents demonstration keyframes for four tasks spanning spatial manipulation, color-dependent object selection, and extended action sequences. *Pick Carrot Single Bowl* requires placing a carrot into the sole bowl, while *Transfer Carrot to Bowl* requires switching the carrot to the other arm before placing it into the bowl. Neither task specifies a color-based target. In contrast, *Stack*

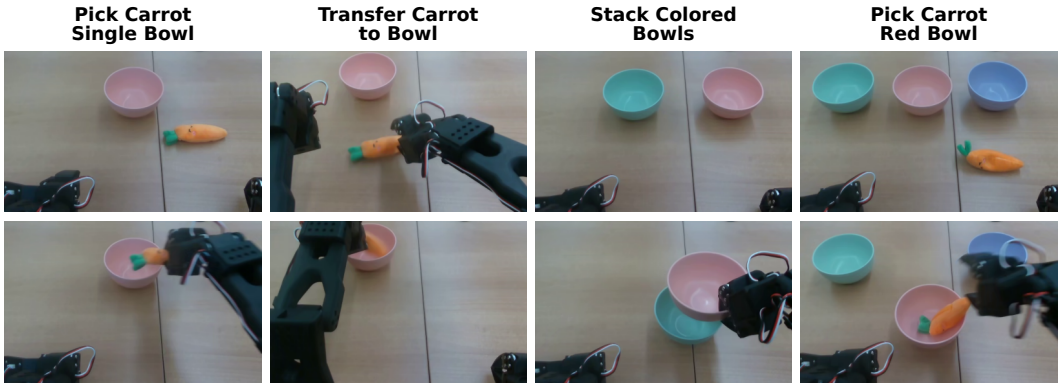

Figure 7: Demonstration keyframes for the four real-world tasks.

Colored Bowls requires placing the red bowl on top of the green bowl, while the long-horizon *Pick Carrot Red Bowl* task requires placing the carrot into the red bowl, potentially involving carrot transfer between arms and retrieval of the target bowl from multiple colored bowls. Each task provides 30 demonstrations, with 29 used for training and one held out for validation. We train a separate policy for each task and evaluate it over 20 trials, with randomized initial object placements for all methods.

We compare GeoRGB with DP3 (Ze et al., 2024), DP (Chi et al., 2025), and ManiFlow (Yan et al., 2025), where DP uses head-camera RGB images and an additional wrist-camera view. DP3 receives XYZ-only inputs for the two tasks without color-defined targets and XYZRGB inputs for the two color-dependent tasks; GeoRGB and ManiFlow use XYZRGB throughout. The 3D policies largely follow their RoboTwin 2.0 training configurations: GeoRGB and ManiFlow use 512 points per frame and train for 600 and 2,000 epochs, respectively, while DP3 uses 1,024 points and trains for 3,000 epochs. All three use a prediction horizon of 32, yielding 31 executable actions for GeoRGB and ManiFlow with two-frame observation histories, and 30 for DP3 with a three-frame history. DP uses the default training and action-chunk configurations provided by LeRobot (Cadene et al., 2024). The results in Table 4 show that GeoRGB achieves 93.8% average success, compared with 78.8% for ManiFlow, 58.8% for DP3, and 38.8% for DP. On *Pick Carrot Red Bowl*, which combines color-dependent target selection with multi-stage bimanual manipulation, GeoRGB achieves the highest success rate among the evaluated policies.

Table 4: Real-world task success. Entries report successful trials out of 20 per task; the final row gives the average success rate (%) across the four tasks. Bold indicates the highest value in each row.

Task	DP (LeRobot)	DP3	ManiFlow	**GeoRGB**
Pick Carrot Single Bowl	12/20	16/20	18/20	**19/20**
Transfer Carrot to Bowl	7/20	15/20	15/20	**17/20**
Stack Colored Bowls	10/20	12/20	17/20	**20/20**
Pick Carrot Red Bowl	2/20	4/20	13/20	**19/20**
Avg. success (%)	38.8	58.8	78.8	**93.8**

5 Conclusion

We presented GeoRGB, a 3D robot policy that addresses the differing geometric and appearance requirements of manipulation through asymmetric encoding of colored point clouds. Its design gives geometry two complementary roles: preserving direct spatial references through pointwise XYZ encoding and structuring contextual interactions among appearance features, supporting precise spatial execution and color-dependent manipulation within a common architecture. Experiments on RoboTwin 2.0, MetaWorld, and real robots demonstrate strong overall manipulation performance, with a favorable performance–cost balance in multi-task learning.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and revision, literature search, feedback on experimental analysis, interpretation of results, and code development for data processing and figure preparation. The authors reviewed the AI-assisted text and analyses, checked references and reported values against their sources, and inspected and tested the generated code. The authors take responsibility for the final content of this work, including its claims, results, and artifacts.

REPRODUCIBILITY STATEMENT

Our work is designed to facilitate reproducibility. The GeoRGB policy and its implementation details, including the training procedure and key training settings, are described in the main paper and appendix. Detailed descriptions of the benchmark datasets, data preprocessing, and experimental protocols are also provided in the main paper and appendix to facilitate the reproduction and further evaluation of the reported results.

REFERENCES

Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1534–1543, 2016.

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

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Khalil Meftah, Maxime Ellerbach, Jess Moss, and Thomas Wolf. LeRobot: State-of-the-art machine learning for real-world robotics in Pytorch. `https://github.com/huggingface/lerobot`, 2024.

Lin-Zhuo Chen, Zheng Lin, Ziqin Wang, Yong-Liang Yang, and Ming-Ming Cheng. Spatial information guided convolution for real-time RGBD semantic segmentation. *IEEE Transactions on Image Processing*, 30:2313–2324, 2021.

Shizhe Chen, Ricardo Garcia Pinel, Cordelia Schmid, and Ivan Laptev. PolarNet: 3D point clouds for language-guided robotic manipulation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1761–1781. PMLR, 2023.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Forty-third International Conference on Machine Learning*, 2026.

Yunlu Chen, Thomas Mensink, and Efstratios Gavves. 3d neighborhood convolution: Learning depth-aware features for rgb-d and rgb semantic segmentation. In *2019 International Conference on 3D Vision (3DV)*, pp. 173–182. IEEE, 2019.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Int. J. Robotics Res.*, 44(10-11):1684–1704, 2025.

Eugenio Chisari, Nick Heppert, Max Argus, Tim Welschehold, Thomas Brox, and Abhinav Valada. Learning robotic manipulation policies from point clouds with conditional flow matching. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 982–993. PMLR, 2025.

Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11672–11681. IEEE, 2020.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3D feature field transformers for multi-task robotic manipulation. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3949–3965. PMLR, 2023.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. RVT: Robotic view transformer for 3D object manipulation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 694–710. PMLR, 2023.

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. RVT-2: Learning precise manipulation from few demonstrations. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13739–13748, June 2022.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D Diffuser Actor: Policy diffusion with 3D scene representations. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1949–1974. PMLR, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Minghua Liu, Xuanlin Li, Zhan Ling, Yangyan Li, and Hao Su. Frame mining: a free lunch for learning robotic manipulation from 3D point clouds. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 527–538. PMLR, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23192–23204, 2022.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 894–906. PMLR, 2022.

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.

Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

Hugues Thomas, Yao-Hung Hubert Tsai, Timothy D. Barfoot, and Jian Zhang. KPConvX: Modernizing kernel point convolution with kernel attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5525–5535, 2024.

Chenrui Tie, Yue Chen, Ruihai Wu, Boxuan Dong, Zeyi Li, Chongkai Gao, and Hao Dong. ET-SEED: Efficient trajectory-level SE(3) equivariant diffusion policy. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 60114–60132, 2025.

Weiyue Wang and Ulrich Neumann. Depth-aware CNN for RGB-D segmentation. In *Proceedings of the European Conference on Computer Vision*, pp. 144–161, 2018.

Ziming Wang, Xiaoliang Huo, Zhenghao Chen, Jing Zhang, Lu Sheng, and Dong Xu. Improving RGB-D point cloud registration by learning multi-scale local linear transformation. In *European Conference on Computer Vision*, pp. 175–191, 2022.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851. IEEE, 2024.

Zongwei Wu, Guillaume Allibert, Christophe Stolz, and Cédric Demonceaux. Depth-adapted CNN for RGB-D cameras. In *Asian Conference on Computer Vision*, pp. 388–404, 2020.

Yajie Xing, Jingbo Wang, and Gang Zeng. Malleable 2.5 D convolution: Learning receptive fields along the depth-axis for RGB-D scene parsing. In *European conference on computer vision*, pp. 555–571, 2020.

Ge Yan, Jiyue Zhu, Yuquan Deng, Shiqi Yang, Ri-Zhao Qiu, Xuxin Cheng, Marius Memmel, Ranjay Krishna, Ankit Goyal, Xiaolong Wang, and Dieter Fox. ManiFlow: A general robot manipulation policy via consistency flow training. In *Conference on Robot Learning (CoRL)*, 2025.

Jingyun Yang, Ziang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. EquiBot: SIM(3)-equivariant diffusion policy for generalizable and data efficient learning. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1048–1068. PMLR, 2025.

Bo-Wen Yin, Jiao-Long Cao, Ming-Ming Cheng, and Qibin Hou. DFormerv2: Geometry self-attention for RGBD semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19345–19355, 2025.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2019.

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 *Robotics: Science and Systems*, 2024.

Jinhao Zhang, Zhexuan Zhou, Huizhe Li, Yichen Lai, Wenlong Xia, Haoming Song, Youmin Gong, and Jie Mei. Hyper-dp3: Frequency-aware right-sizing of 3d diffusion policies for visuomotor control. *arXiv preprint arXiv:2605.01581*, 2026.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16259–16268, 2021.

A Complete Benchmark Results on Simulation

Table 5: RoboTwin 2.0 C2C success rates (%) across 50 tasks, with GeoRGB seed columns show fixed evaluation seed 0; the final GeoRGB column reports the mean ± sample standard deviation across the three training seeds. Δ denotes the change from the strongest baseline in each row (percentage points). Bold indicates the best score.

Task	DP	DP3	π_0	Hyper-DP3	**GeoRGB**				**Δ**
					seed 0	seed 1	seed 2	mean ± std	
Adjust Bottle	97.0	99.0	90.0	92.0	99.0	100.0	100.0	**99.7**$_{\pm 0.6}$	**+0.7**
Beat Block Hammer	42.0	72.0	43.0	88.0	93.0	94.0	91.0	**92.7**$_{\pm 1.5}$	**+4.7**
Blocks Ranking RGB	0.0	3.0	19.0	3.0	88.0	81.0	82.0	**83.7**$_{\pm 3.8}$	**+64.7**
Blocks Ranking Size	1.0	2.0	7.0	8.0	32.0	31.0	38.0	**33.7**$_{\pm 3.8}$	**+25.7**
Click Alarmclock	61.0	77.0	63.0	**96.0**	78.0	89.0	83.0	83.3$_{\pm 5.5}$	**−12.7**
Click Bell	54.0	90.0	44.0	**100.0**	98.0	99.0	96.0	97.7$_{\pm 1.5}$	**−2.3**
Dump Bin Bigbin	49.0	85.0	83.0	84.0	85.0	87.0	89.0	**87.0**$_{\pm 2.0}$	**+2.0**
Grab Roller	98.0	98.0	96.0	**100.0**	100.0	100.0	100.0	**100.0**$_{\pm 0.0}$	**0.0**
Handover Block	10.0	70.0	45.0	70.0	100.0	98.0	99.0	**99.0**$_{\pm 1.0}$	**+29.0**
Handover Mic	53.0	**100.0**	98.0	**100.0**	93.0	95.0	97.0	95.0$_{\pm 2.0}$	**−5.0**
Hanging Mug	8.0	17.0	11.0	35.0	36.0	32.0	48.0	**38.7**$_{\pm 8.3}$	**+3.7**
Lift Pot	39.0	97.0	84.0	95.0	99.0	100.0	98.0	**99.0**$_{\pm 1.0}$	**+2.0**
Move Can Pot	39.0	70.0	58.0	**95.0**	96.0	94.0	89.0	93.0$_{\pm 3.6}$	**−2.0**
Move Pillbottle Pad	1.0	41.0	21.0	56.0	86.0	92.0	83.0	**87.0**$_{\pm 4.6}$	**+31.0**
Move Playingcard Away	47.0	68.0	53.0	73.0	88.0	85.0	82.0	**85.0**$_{\pm 3.0}$	**+12.0**
Move Stapler Pad	1.0	**12.0**	0.0	10.0	13.0	9.0	14.0	**12.0**$_{\pm 2.6}$	**0.0**
Open Laptop	49.0	82.0	85.0	85.0	89.0	87.0	88.0	**88.0**$_{\pm 1.0}$	**+3.0**
Open Microwave	5.0	61.0	80.0	92.0	97.0	92.0	94.0	**94.3**$_{\pm 2.5}$	**+2.3**
Pick Diverse Bottles	6.0	52.0	27.0	71.0	87.0	89.0	90.0	**88.7**$_{\pm 1.5}$	**+17.7**
Pick Dual Bottles	24.0	60.0	57.0	90.0	99.0	100.0	100.0	**99.7**$_{\pm 0.6}$	**+9.7**
Place A2B Left	2.0	46.0	31.0	50.0	61.0	60.0	54.0	**58.3**$_{\pm 3.8}$	**+8.3**
Place A2B Right	13.0	**49.0**	27.0	27.0	40.0	44.0	43.0	42.3$_{\pm 2.1}$	**−6.7**
Place Bread Basket	14.0	26.0	17.0	39.0	78.0	86.0	87.0	**83.7**$_{\pm 4.9}$	**+44.7**
Place Bread Skillet	11.0	19.0	23.0	40.0	68.0	76.0	64.0	**69.3**$_{\pm 6.1}$	**+29.3**
Place Burger Fries	72.0	72.0	80.0	90.0	100.0	97.0	96.0	**97.7**$_{\pm 2.1}$	**+7.7**
Place Can Basket	18.0	67.0	41.0	**78.0**	69.0	74.0	58.0	67.0$_{\pm 8.2}$	**−11.0**
Place Cans Plasticbox	40.0	48.0	34.0	96.0	100.0	99.0	100.0	**99.7**$_{\pm 0.6}$	**+3.7**
Place Container Plate	41.0	86.0	88.0	87.0	95.0	95.0	96.0	**95.3**$_{\pm 0.6}$	**+7.3**
Place Dual Shoes	8.0	13.0	15.0	8.0	23.0	20.0	30.0	**24.3**$_{\pm 5.1}$	**+9.3**
Place Empty Cup	37.0	65.0	37.0	89.0	89.0	92.0	92.0	**91.0**$_{\pm 1.7}$	**+2.0**
Place Fan	3.0	36.0	20.0	**39.0**	28.0	24.0	27.0	26.3$_{\pm 2.1}$	**−12.7**
Place Mouse Pad	0.0	4.0	7.0	7.0	9.0	13.0	14.0	**12.0**$_{\pm 2.6}$	**+5.0**
Place Object Basket	15.0	65.0	16.0	65.0	75.0	80.0	79.0	**78.0**$_{\pm 2.6}$	**+13.0**
Place Object Scale	1.0	15.0	10.0	9.0	44.0	59.0	55.0	**52.7**$_{\pm 7.8}$	**+37.7**
Place Object Stand	22.0	60.0	36.0	65.0	82.0	85.0	85.0	**84.0**$_{\pm 1.7}$	**+19.0**
Place Phone Stand	13.0	44.0	35.0	65.0	81.0	84.0	86.0	**83.7**$_{\pm 2.5}$	**+18.7**
Place Shoe	23.0	58.0	28.0	57.0	64.0	70.0	74.0	**69.3**$_{\pm 5.0}$	**+11.3**
Press Stapler	6.0	69.0	62.0	73.0	88.0	82.0	91.0	**87.0**$_{\pm 4.6}$	**+14.0**
Put Bottles Dustbin	22.0	60.0	54.0	64.0	76.0	82.0	77.0	**78.3**$_{\pm 3.2}$	**+14.3**
Put Object Cabinet	42.0	**72.0**	68.0	63.0	57.0	57.0	51.0	55.0$_{\pm 3.5}$	**−17.0**
Rotate QRcode	13.0	74.0	68.0	55.0	75.0	82.0	80.0	**79.0**$_{\pm 3.6}$	**+5.0**
Scan Object	9.0	31.0	18.0	45.0	79.0	80.0	77.0	**78.7**$_{\pm 1.5}$	**+33.7**
Shake Bottle	65.0	98.0	97.0	**100.0**	100.0	100.0	100.0	**100.0**$_{\pm 0.0}$	**0.0**
Shake Bottle Horizontally	59.0	**100.0**	99.0	**100.0**	100.0	100.0	100.0	**100.0**$_{\pm 0.0}$	**0.0**
Stack Blocks Three	0.0	1.0	17.0	2.0	78.0	81.0	91.0	**83.3**$_{\pm 6.8}$	**+66.3**
Stack Blocks Two	7.0	24.0	42.0	41.0	91.0	93.0	93.0	**92.3**$_{\pm 1.2}$	**+50.3**
Stack Bowls Three	63.0	57.0	66.0	74.0	83.0	83.0	77.0	**81.0**$_{\pm 3.5}$	**+7.0**
Stack Bowls Two	61.0	83.0	91.0	89.0	96.0	98.0	97.0	**97.0**$_{\pm 1.0}$	**+6.0**
Stamp Seal	2.0	18.0	3.0	43.0	77.0	80.0	85.0	**80.7**$_{\pm 4.0}$	**+37.7**
Turn Switch	36.0	46.0	27.0	55.0	65.0	69.0	73.0	**69.0**$_{\pm 4.0}$	**+14.0**
Avg. Success	28.0	55.2	46.4	63.2	76.5	78.0	77.9	**77.5**	**+14.3**

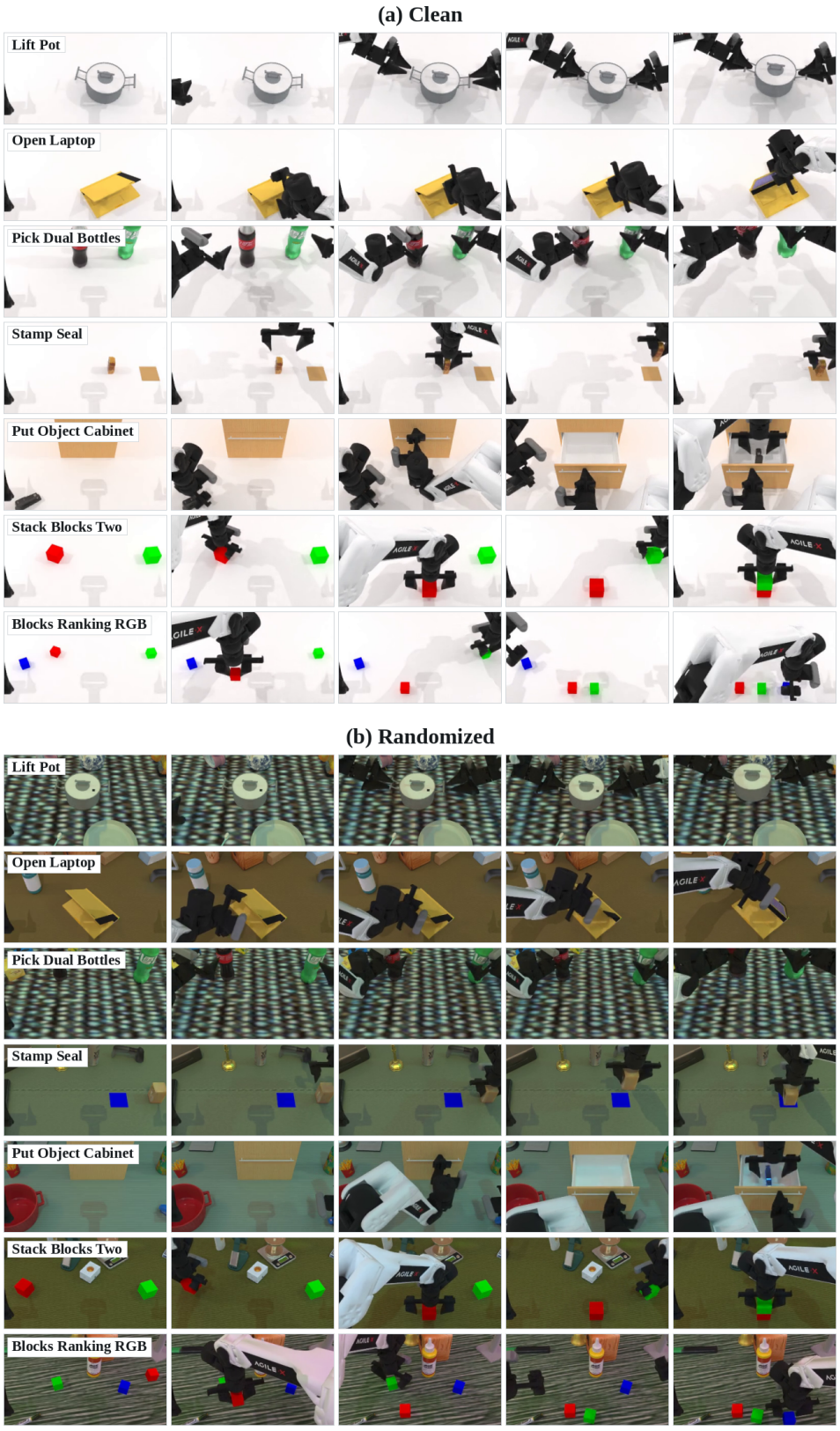

Figure 8: Representative RoboTwin 2.0 task executions under (a) clean and (b) randomized scenes. Each row shows five uniformly spaced frames from one evaluation rollout.

Table 6: MetaWorld success rates (%), reported as mean with sample standard deviation across three training seeds. For GeoRGB, 4/8/12 blocks denote the number of DiT blocks in the action generator.

Task	DP3	ManiFlow	**GeoRGB**		
			4 blocks	8 blocks	12 blocks
Easy					
Button Press	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Button Press Top-Down	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Button Press Top-Down Wall	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Button Press Wall	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Coffee Button	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Dial Turn	$26.7_{\pm 11.5}$	$85.0_{\pm 15.0}$	$63.3_{\pm 10.4}$	$58.3_{\pm 18.9}$	$50.0_{\pm 21.8}$
Door Close	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Door Lock	$78.3_{\pm 15.3}$	$100.0_{\pm 0.0}$	$98.3_{\pm 2.9}$	$96.7_{\pm 2.9}$	$98.3_{\pm 2.9}$
Door Open	$86.7_{\pm 7.6}$	$100.0_{\pm 0.0}$	$98.3_{\pm 2.9}$	$100.0_{\pm 0.0}$	$96.7_{\pm 2.9}$
Door Unlock	$96.7_{\pm 2.9}$	$91.7_{\pm 7.6}$	$95.0_{\pm 5.0}$	$80.0_{\pm 13.2}$	$91.7_{\pm 7.6}$
Drawer Close	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Drawer Open	$85.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Faucet Close	$98.3_{\pm 2.9}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$95.0_{\pm 0.0}$	$96.7_{\pm 2.9}$
Faucet Open	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Handle Press	$100.0_{\pm 0.0}$	$98.3_{\pm 2.9}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Handle Pull	$18.3_{\pm 10.4}$	$75.0_{\pm 5.0}$	$81.7_{\pm 10.4}$	$51.7_{\pm 7.6}$	$68.3_{\pm 7.6}$
Handle Pull Side	$6.7_{\pm 2.9}$	$23.3_{\pm 5.8}$	$25.0_{\pm 10.0}$	$28.3_{\pm 5.8}$	$30.0_{\pm 5.0}$
Lever Pull	$51.7_{\pm 7.6}$	$55.0_{\pm 10.0}$	$45.0_{\pm 10.0}$	$60.0_{\pm 5.0}$	$46.7_{\pm 22.5}$
Plate Slide	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$98.3_{\pm 2.9}$
Plate Slide Back	$98.3_{\pm 2.9}$	$71.7_{\pm 2.9}$	$45.0_{\pm 13.2}$	$55.0_{\pm 8.7}$	$50.0_{\pm 13.2}$
Plate Slide Back Side	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Plate Slide Side	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Reach	$38.3_{\pm 10.4}$	$38.3_{\pm 10.4}$	$50.0_{\pm 5.0}$	$48.3_{\pm 2.9}$	$55.0_{\pm 5.0}$
Reach Wall	$45.0_{\pm 13.2}$	$35.0_{\pm 8.7}$	$40.0_{\pm 8.7}$	$40.0_{\pm 8.7}$	$46.7_{\pm 17.6}$
Window Close	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$
Window Open	$83.3_{\pm 14.4}$	$85.0_{\pm 13.2}$	$91.7_{\pm 7.6}$	$68.3_{\pm 10.4}$	$100.0_{\pm 0.0}$
Peg Unplug Side	$41.7_{\pm 16.1}$	$66.7_{\pm 15.3}$	$78.3_{\pm 2.9}$	$70.0_{\pm 0.0}$	$66.7_{\pm 2.9}$
Easy Avg.	$79.8_{\pm 1.2}$	$86.1_{\pm 2.3}$	$85.6_{\pm 1.2}$	$83.4_{\pm 1.2}$	$85.0_{\pm 0.5}$
Medium					
Basketball	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$96.7_{\pm 5.8}$	$93.3_{\pm 2.9}$	$90.0_{\pm 5.0}$
Bin Picking	$25.0_{\pm 10.0}$	$70.0_{\pm 13.2}$	$56.7_{\pm 14.4}$	$75.0_{\pm 10.0}$	$78.3_{\pm 2.9}$
Box Close	$36.7_{\pm 5.8}$	$25.0_{\pm 0.0}$	$28.3_{\pm 15.3}$	$46.7_{\pm 10.4}$	$23.3_{\pm 2.9}$
Coffee Pull	$53.3_{\pm 12.6}$	$66.7_{\pm 2.9}$	$66.7_{\pm 5.8}$	$58.3_{\pm 22.5}$	$63.3_{\pm 12.6}$
Coffee Push	$20.0_{\pm 5.0}$	$95.0_{\pm 5.0}$	$78.3_{\pm 10.4}$	$91.7_{\pm 2.9}$	$95.0_{\pm 0.0}$
Hammer	$21.7_{\pm 7.6}$	$65.0_{\pm 8.7}$	$68.3_{\pm 2.9}$	$65.0_{\pm 8.7}$	$71.7_{\pm 12.6}$
Peg Insert Side	$18.3_{\pm 12.6}$	$88.3_{\pm 5.8}$	$83.3_{\pm 12.6}$	$88.3_{\pm 2.9}$	$85.0_{\pm 8.7}$
Push Wall	$21.7_{\pm 7.6}$	$93.3_{\pm 2.9}$	$93.3_{\pm 2.9}$	$95.0_{\pm 5.0}$	$93.3_{\pm 5.8}$
Soccer	$20.0_{\pm 13.2}$	$28.3_{\pm 2.9}$	$15.0_{\pm 8.7}$	$25.0_{\pm 10.0}$	$20.0_{\pm 13.2}$
Sweep	$38.3_{\pm 10.4}$	$81.7_{\pm 5.8}$	$53.3_{\pm 5.8}$	$68.3_{\pm 5.8}$	$70.0_{\pm 15.0}$
Sweep Into	$43.3_{\pm 2.9}$	$61.7_{\pm 2.9}$	$66.7_{\pm 12.6}$	$75.0_{\pm 0.0}$	$71.7_{\pm 10.4}$
Medium Avg.	$36.2_{\pm 1.4}$	$70.5_{\pm 2.1}$	$64.2_{\pm 0.7}$	$71.1_{\pm 2.3}$	$69.2_{\pm 3.4}$
Hard					
Assembly	$73.3_{\pm 5.8}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$93.3_{\pm 7.6}$	$95.0_{\pm 5.0}$
Hand Insert	$35.0_{\pm 5.0}$	$50.0_{\pm 8.7}$	$55.0_{\pm 5.0}$	$35.0_{\pm 8.7}$	$58.3_{\pm 2.9}$
Pick Out Of Hole	$20.0_{\pm 5.0}$	$10.0_{\pm 13.2}$	$15.0_{\pm 8.7}$	$10.0_{\pm 13.2}$	$8.3_{\pm 7.6}$
Pick Place	$10.0_{\pm 5.0}$	$28.3_{\pm 5.8}$	$20.0_{\pm 8.7}$	$18.3_{\pm 2.9}$	$21.7_{\pm 17.6}$
Push	$11.7_{\pm 7.6}$	$25.0_{\pm 5.0}$	$30.0_{\pm 5.0}$	$30.0_{\pm 5.0}$	$35.0_{\pm 15.0}$
Hard Avg.	$30.0_{\pm 2.6}$	$42.7_{\pm 2.3}$	$44.0_{\pm 4.4}$	$37.3_{\pm 4.2}$	$43.7_{\pm 1.5}$
Very Hard					
Shelf Place	$8.3_{\pm 5.8}$	$20.0_{\pm 13.2}$	$13.3_{\pm 10.4}$	$30.0_{\pm 5.0}$	$38.3_{\pm 7.6}$
Disassemble	$45.0_{\pm 15.0}$	$63.3_{\pm 2.9}$	$78.3_{\pm 16.1}$	$86.7_{\pm 7.6}$	$90.0_{\pm 5.0}$
Stick Pull	$40.0_{\pm 13.2}$	$50.0_{\pm 15.0}$	$48.3_{\pm 17.6}$	$51.7_{\pm 7.6}$	$51.7_{\pm 7.6}$
Stick Push	$83.3_{\pm 12.6}$	$91.7_{\pm 2.9}$	$100.0_{\pm 0.0}$	$96.7_{\pm 2.9}$	$91.7_{\pm 2.9}$
Pick Place Wall	$23.3_{\pm 10.4}$	$58.3_{\pm 15.3}$	$48.3_{\pm 15.3}$	$46.7_{\pm 7.6}$	$53.3_{\pm 10.4}$

Task	DP3	ManiFlow	**GeoRGB**		
			4 blocks	8 blocks	12 blocks
Very Hard Avg.	$40.0_{\pm 7.0}$	$56.7_{\pm 6.7}$	$57.7_{\pm 2.9}$	$62.3_{\pm 1.2}$	$65.0_{\pm 3.6}$
Overall Avg.	$60.5_{\pm 1.0}$	$74.9_{\pm 1.2}$	$73.5_{\pm 0.8}$	$73.6_{\pm 1.0}$	$75.0_{\pm 0.6}$

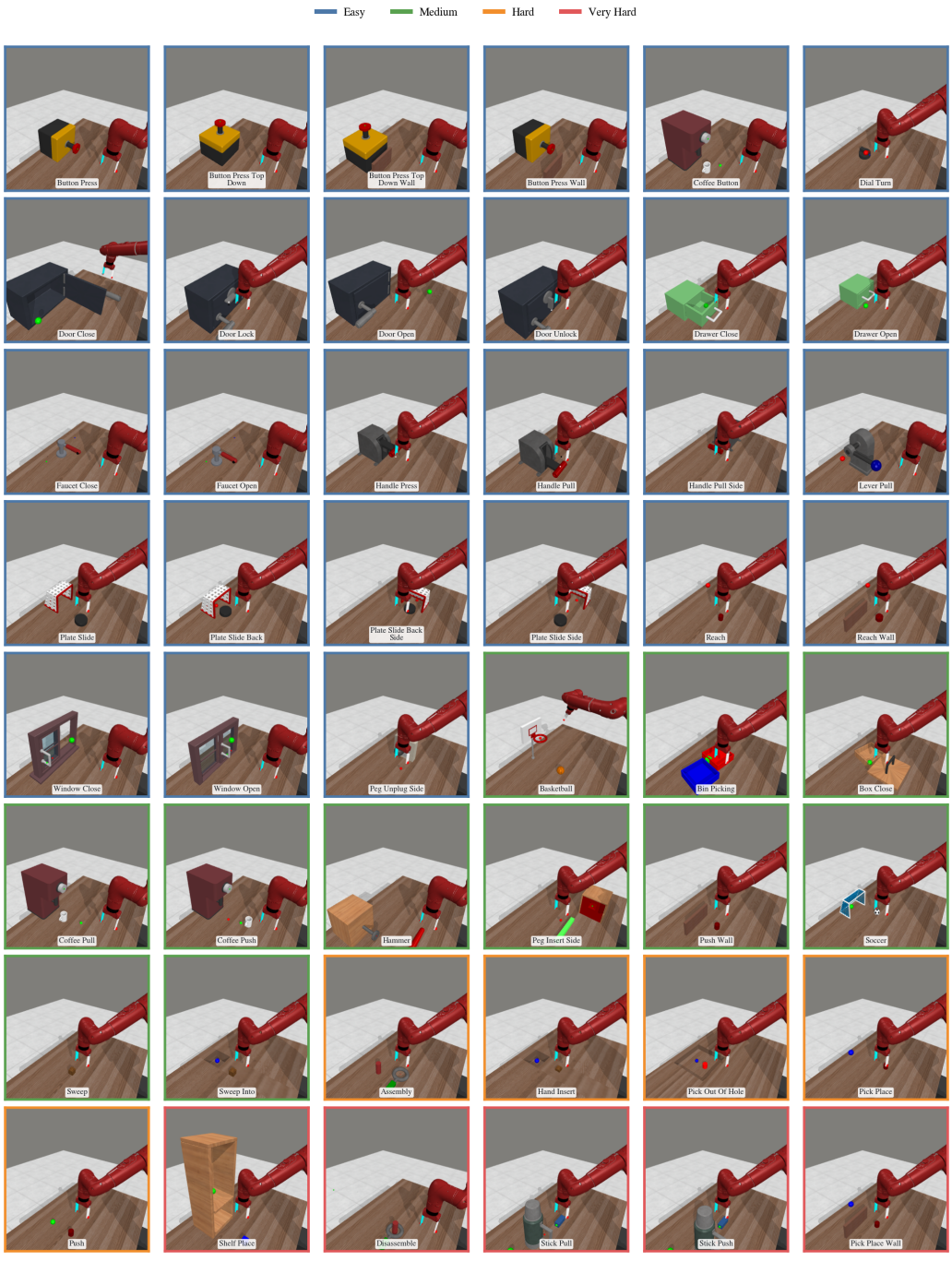

Figure 9: MetaWorld 48-task suite. Panels show representative initial observations rendered with a fixed seed and follow the task order in Table 6. Border colors indicate the easy (blue), medium (green), hard (orange), and very hard (red) groups.

B Implementation Details

B.1 Single-Frame Colored Point-Cloud Encoding

Algorithm 1 expands the complete data flow for one normalized colored point-cloud frame. It separates the multiscale schedule from the two reusable point operations: geometry-guided RGB contextualization and feature propagation.

Algorithm 1: Single-frame GeoRGB encoding

Require: normalized coordinates $\mathbf{X} \in \mathbb{R}^{N\times 3}$ and colors $\mathbf{C} \in \mathbb{R}^{N\times 3}$, where $N = 512$; neighborhood size $K = 16$; stage sizes $(N_0, N_1, N_2) = (512, 128, 32)$ and widths $(D_0, D_1, D_2) = (128, 192, 256)$.
Ensure: point tokens $\mathbf{Z} \in \mathbb{R}^{N\times 256}$.

$\mathbf{X}^0 \leftarrow \mathbf{X}$, $\mathbf{G} \leftarrow f_{\mathrm{geo}}(\mathbf{X}^0)$, $\mathbf{H}^0 \leftarrow f_{\mathrm{rgb}}(\mathbf{C})$
for $\ell = 1, 2$ **do**
  $\mathbf{I}^\ell \leftarrow \mathrm{FPS}(\mathbf{X}^{\ell-1}, N_\ell)$; $\mathbf{X}^\ell \leftarrow \mathbf{X}^{\ell-1}[\mathbf{I}^\ell]$
  $\mathbf{Q}^\ell \leftarrow \mathbf{H}^{\ell-1}[\mathbf{I}^\ell]$ ▷ query shortcuts
  $\mathbf{H}^\ell \leftarrow \mathrm{GRC}(\mathbf{X}^\ell, \mathbf{X}^{\ell-1}, \mathbf{Q}^\ell, \mathbf{H}^{\ell-1}, D_\ell, K)$ ▷ cross-scale
  $\mathbf{H}^\ell \leftarrow \mathrm{GRC}(\mathbf{X}^\ell, \mathbf{X}^\ell, \mathbf{H}^\ell, \mathbf{H}^\ell, D_\ell, K)$ ▷ same-scale
end for
for $\ell = 1$ **down to** 0 **do**
  $\mathbf{H}^\ell \leftarrow \textsc{FeaturePropagate}(\mathbf{X}^\ell, \mathbf{X}^{\ell+1}, \mathbf{H}^\ell, \mathbf{H}^{\ell+1}, D_\ell)$
end for
for $i = 1, \ldots, N$ **do** $\mathbf{z}_i \leftarrow \mathbf{g}_i \,\|\, \mathbf{h}_i^0$ **end for**
return $\mathbf{Z} = [\mathbf{z}_i]_{i=1}^N$
procedure $\mathrm{GRC}(\mathbf{X}_q, \mathbf{X}_s, \mathbf{H}_q, \mathbf{H}_s, D, K)$
  for each query $\mathbf{x}_{q,i}$ **do**
    $\mathcal{N}_i \leftarrow \mathrm{KNN}(\mathbf{x}_{q,i}, \mathbf{X}_s, K)$; $r_i \leftarrow \max_{j\in\mathcal{N}_i} \|\mathbf{x}_{s,j} - \mathbf{x}_{q,i}\|_2$
    for each $j \in \mathcal{N}_i$ **do**
      $\boldsymbol{\delta}_{ij} \leftarrow (\mathbf{x}_{s,j} - \mathbf{x}_{q,i}) / \max(r_i, 10^{-6})$
      $\pi_{ij} \leftarrow \arg\min_m \|\boldsymbol{\delta}_{ij} - \boldsymbol{\kappa}_m\|_2$; $\alpha_{ij} \leftarrow [1 - \|\boldsymbol{\delta}_{ij} - \boldsymbol{\kappa}_{\pi_{ij}}\|_2]_+$
    end for
    $\mathbf{u}_i \leftarrow \sum_{j\in\mathcal{N}_i} \alpha_{ij}\big(P(\mathbf{h}_{s,j}) \odot \mathbf{w}_{\pi_{ij}}\big)$
    $\mathbf{y}_i \leftarrow \textsc{ResMLP}(\mathrm{LN}(S(\mathbf{h}_{q,i}) + \mathbf{u}_i))$
  end for; **return** $[\mathbf{y}_i]_{i=1}^{|\mathbf{X}_q|}$
procedure $\textsc{FeaturePropagate}(\mathbf{X}_f, \mathbf{X}_c, \mathbf{H}_f, \mathbf{H}_c, D)$
  for each fine point $\mathbf{x}_{f,i}$ **do**
    $\mathcal{J}_i \leftarrow \mathrm{3NN}(\mathbf{x}_{f,i}, \mathbf{X}_c)$; $a_{ij} \leftarrow \max(\|\mathbf{x}_{f,i} - \mathbf{x}_{c,j}\|_2^2, 10^{-8})^{-1}$
    $\bar{\mathbf{h}}_i \leftarrow \sum_{j\in\mathcal{J}_i} \frac{a_{ij}}{\sum_{k\in\mathcal{J}_i} a_{ik}} \mathbf{h}_{c,j}$
    $\mathbf{y}_i \leftarrow \textsc{ResMLP}\big(\mathrm{GELU}\big(\mathrm{LN}\big(F(\mathbf{h}_{f,i} \,\|\, \bar{\mathbf{h}}_i)\big)\big)\big)$
  end for; **return** $[\mathbf{y}_i]_{i=1}^{|\mathbf{X}_f|}$

Notation. P, S, and F denote the learned input, shortcut, and fusion projections of the current module. Each GRC invocation has its own projections and kernel weights.

B.2 Benchmark and Optimization Configuration

Table 7: GeoRGB benchmark settings. Demonstrations and evaluation episodes are counted per task.

Setting	RoboTwin 2.0	MetaWorld	Real world
Tasks	50 C2C; 7 R2R	48	4
Policy training	Per task/setting	Multi-task	Per task
Demonstrations (train / validation)	49 / 1	9 / 1	29 / 1
Observation frames / points per frame	2 / 512	2 / 512	2 / 512
Prediction horizon	16	4	32
Actions executed before replanning	15	3	31
Batch size	128	384	128
Evaluation episodes per policy	100	20	20
Task-ID conditioning	No	Yes	No

Table 8: GeoRGB policy and optimization settings for simulation.

Setting	Value
Action-generator blocks / width / heads	4 / 512 / 8
MLP expansion / dropout	4 / 0
Flow-time sinusoidal embedding width	128
Optimizer	AdamW
Learning rate / weight decay	10^{-4} / 10^{-3}
AdamW betas / epsilon	$(0.9, 0.95)$ / 10^{-8}
Schedule / warm-up	Cosine / 500 optimizer steps
Numerical precision / training epochs	BF16 / 600
EMA inverse gamma / power / maximum decay	1 / 0.75 / 0.9999
Training flow time	$s = 0.999u$, $u \sim \mathrm{Beta}(1, 1.5)$
Inference	EMA weights; 10 uniform Euler steps

C Semantic Segmentation on S3DIS

S3DIS (Armeni et al., 2016) provides a complementary setting for examining whether the asymmetric encoding principle also benefits semantic segmentation of real indoor colored point clouds. Rather than transferring representations learned by the manipulation policy, we train each encoder from scratch with a segmentation head, allowing us to assess the encoding design independently of transferred policy representations.

We first evaluate the two-level GeoRGB encoder used in the policy against the Pointwise XYZRGB baseline. Unlike the 512-point manipulation inputs, however, each S3DIS training sample uses 24,000 points and retains 1,500 points after two downsampling stages. This substantially higher spatial resolution leaves room for additional contextual aggregation, motivating a deeper four-level RGB hierarchy, $24{,}000 \to 6{,}000 \to 1{,}500 \to 375 \to 93$, while keeping the geometry branch pointwise. All variants use the PointNeXt-S segmentation-head design and are trained for 100 epochs with batch size 32. Following the standard Area 5 protocol, we train on the remaining five areas and evaluate on all 68 rooms in Area 5. Model selection is based on Area 5 validation mIoU. We report mIoU and mean class accuracy (mAcc) over the 13 semantic classes.

The GeoRGB encoders substantially outperform the pointwise baseline. In Table 9, the two-level GeoRGB encoder achieves 52.12% mIoU, compared with 23.39% for Pointwise XYZRGB. Increasing the RGB hierarchy to four levels further improves performance to 64.61% mIoU and 70.91% mAcc, bringing it into the range of the PointNet++ and PointNeXt-S references. These results support the potential of the asymmetric encoding principle beyond robot manipulation.

Table 9: Semantic segmentation on S3DIS (Area 5). Reported PointNet++ and PointNeXt-S results, including the OpenPoints released checkpoint, are from (Qian et al., 2022). The other rows are our experiments.

Encoder	mIoU (%) ↑	mAcc (%) ↑
Pointwise XYZRGB	23.39	31.07
PointNet++ (original)	53.5	–
PointNet++ (OpenPoints released)	63.6	70.2
PointNeXt-S	$63.4_{\pm 0.8}$	$70.0_{\pm 0.7}$
GeoRGB (2-level hierarchy)	52.12	59.99
GeoRGB (4-level hierarchy)	**64.61**	**70.91**

D Controlled Representation Comparisons

D.1 Local Backbones with and without Independent XYZ

Motivation and design. We test whether performance depends on ADTE rather than the choice of local backbone by adapting four-level PointNet++ and PointNeXt-S as two controls. Their original quarter-per-stage sampling would reduce 512 input points to two, so we retain half per stage, matching GeoRGB's 32 points at the deepest level. The *joint XYZRGB* control replaces the full visual encoder; the *RGB context + pointwise XYZ* control applies the same hierarchy only to RGB and retains the independent XYZ pathway. We remove segmentation heads while keeping the policy interface, action generator, and training protocol fixed.

Table 10: RoboTwin 2.0 C2C success rates (%) under different local-backbone roles. All rows use training seed 0, the epoch-600 checkpoint, and 100 evaluation episodes with evaluation seed 0. Bold indicates the highest success rate in each column.

Encoder	Organization	Stamp Seal	Stack Blocks Two	Avg.
GeoRGB	Pointwise XYZ + GRC RGB	**77**	91	**84.0**
PointNet++	Joint XYZRGB	6	51	28.5
	RGB context + pointwise XYZ	69	**94**	81.5
PointNeXt-S	Joint XYZRGB	8	43	25.5
	RGB context + pointwise XYZ	60	93	76.5

Results. Switching from joint encoding to the ADTE-style organization raises the two-task average by 53.0 points with PointNet++ and 51.0 points with PointNeXt-S. This repeated reversal under the same backbone and point schedule shows that representation organization, rather than deeper local processing alone, drives the recovery. The contextual replacements reach 93–94% on *Stack Blocks Two* but 60–69% on *Stamp Seal*, compared with GeoRGB's 91% and 77%. This pattern supports ADTE across different contextual backbones, while the relative benefit of GRC remains task dependent. Across the two tasks, GeoRGB achieves the highest average success and provides the best aggregate balance.

D.2 Contextual RGB versus Contextual XYZRGB

We extend the two-task C2C comparison in the main text to all seven tasks under both C2C and R2R. Both variants retain the independent pointwise XYZ pathway; contextual XYZRGB changes only the context-stem input from RGB to XYZRGB.

Table 11: Success rates (%) for contextual attribute encoding. Entries show mean and standard deviation across training seeds 0, 1, and 2 using epoch-600 checkpoints and 100 evaluation episodes per seed. The final row averages the seven task means. Bold indicates the higher mean within each setting; equal means are both bold.

	C2C		R2R	
Task	GeoRGB	Contextual XYZRGB	GeoRGB	Contextual XYZRGB
Lift Pot	$\mathbf{99.0}_{\pm 1.0}$	$98.7_{\pm 0.6}$	$\mathbf{68.7}_{\pm 5.5}$	$64.3_{\pm 2.1}$
Open Laptop	$\mathbf{88.0}_{\pm 1.0}$	$84.7_{\pm 2.1}$	$78.3_{\pm 4.0}$	$\mathbf{80.7}_{\pm 3.2}$
Pick Dual Bottles	$\mathbf{99.7}_{\pm 0.6}$	$\mathbf{99.7}_{\pm 0.6}$	$\mathbf{82.7}_{\pm 6.7}$	$78.0_{\pm 6.1}$
Stamp Seal	$\mathbf{80.7}_{\pm 4.0}$	$67.3_{\pm 2.1}$	$\mathbf{44.7}_{\pm 2.5}$	$41.3_{\pm 12.5}$
Put Object Cabinet	$\mathbf{55.0}_{\pm 3.5}$	$52.0_{\pm 8.9}$	$34.7_{\pm 4.2}$	$\mathbf{35.7}_{\pm 5.5}$
Stack Blocks Two	$92.3_{\pm 1.2}$	$\mathbf{93.0}_{\pm 3.0}$	$72.3_{\pm 3.8}$	$\mathbf{79.7}_{\pm 4.6}$
Blocks Ranking RGB	$83.7_{\pm 3.8}$	$\mathbf{85.3}_{\pm 1.5}$	$\mathbf{27.3}_{\pm 10.7}$	$17.7_{\pm 9.8}$
Avg. success	**85.5**	83.0	**58.4**	56.8

Across seven tasks, RGB-only context averages 85.5% versus 83.0% in C2C and 58.4% versus 56.8% in R2R. Adding XYZ to the context stem yields task-dependent gains and regressions rather than a consistent benefit; its gains on *Open Laptop* and *Stack Blocks Two* do not offset the regressions, especially on *Stamp Seal* and *Blocks Ranking RGB*.

E Behavioral Analysis of Geometry and Color Roles

Table 12 provides the counts underlying Figure 1. We use the 600-epoch checkpoints trained with seed 0 and evaluated with seed 0 under RoboTwin C2C, aligning rollouts by scene seed across 96 recorded scenes for Stamp Seal and 100 for Stack Blocks Two. These single-checkpoint counts need not equal the three-seed means in the main ablation table.

Stage annotation. We count a grasp when the gripper lifts or transports the target object. The event is seal grasping for Stamp Seal and grasping the red block first for Stack Blocks Two. Grasp labels are annotated from rollout videos, while final success follows the environment logs.

Table 12: Stage analysis counts. N: evaluated scenes; G: successful seal grasp or red-first grasp; S: final task successes. "Completion given grasp" is $100S/G$ for these rollouts, in which all task successes satisfy the grasp event.

Task	Encoder	N	G	S	Completion given grasp (%)
Stamp Seal	XYZ-only	96	76	61	80.3
	Pointwise XYZRGB	96	70	17	24.3
	Separate pointwise	96	83	43	51.8
	GeoRGB	96	—	74	—
Stack Blocks Two	XYZ-only	100	46	40	87.0
	Pointwise XYZRGB	100	98	79	80.6
	Separate pointwise	100	96	77	80.2
	GeoRGB	100	97	91	93.8

F Real-World RGB-D Processing and Task Demonstrations

The head-mounted RealSense D435i provides an RGB image I and a depth map D aligned to the RGB frame. Its calibrated color-camera intrinsics are

$$\mathbf{K} = \begin{bmatrix} 606.34 & 0 & 330.34 \\ 0 & 606.14 & 246.20 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

For a pixel (u, v), we convert its depth to metres as $Z = sD(u, v)$ and back-project it into the camera frame:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z\mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \qquad \mathbf{c} = I(u, v). \tag{6}$$

We retain pixels inside the image crop with depths between 0.1 and 0.9 m, remove points within 8 mm of the estimated table plane, and deterministically thin the valid pixels to 4096 XYZRGB candidates. GeoRGB then applies the checkpoint normalization and farthest point sampling on normalized XYZ to select 512 encoder inputs.

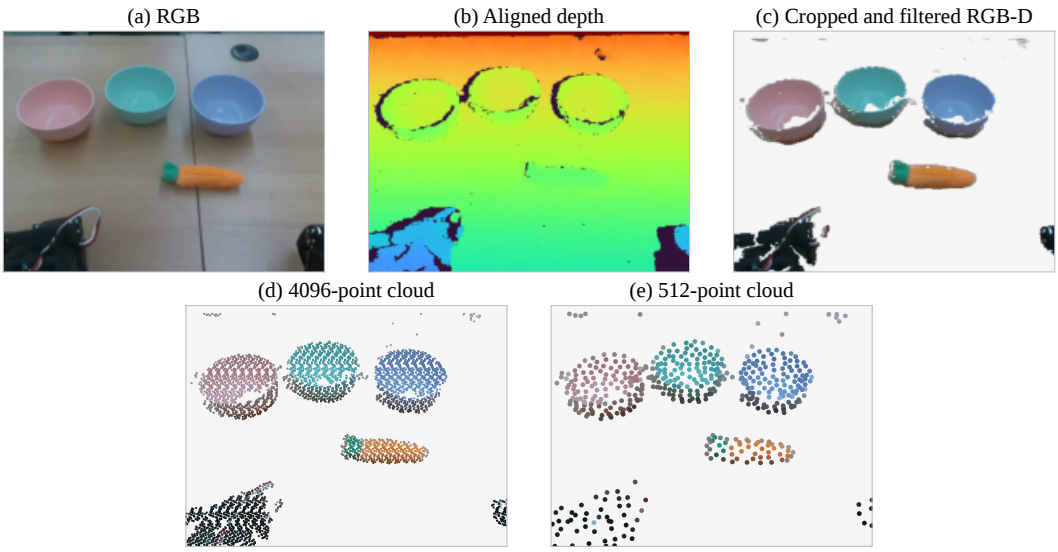

Figure 10: RGB-D point-cloud construction for *Pick Carrot Red Bowl*. The aligned RGB and depth observations are cropped and filtered before forming 4096 XYZRGB candidates; farthest point sampling selects the 512 points. The point clouds are projected into the camera view for visualization.



Figure 11: Demonstration sequences for the four real-world tasks. Each row follows one task from its initial scene to task completion.