BEYOND JOINT ANGLES: LEARNING SPATIAL ROBOT ARM STRUCTURE FOR COLLISION-FREE POLICY

Anonymous authors
Paper under double-blind review

ABSTRACT

Collision avoidance is a key challenge in bimanual manipulation. We argue that the essence of this problem is that proprioceptive representations such as joint angles lack information about the manipulator's spatial structure. Based on this insight, we propose RoboCloud, a structural-prior descriptor that models both the geometry of entire manipulator and the topology of link connection, providing both spatial structure and kinematics for predicting potential surface contact either between the manipulators or between a manipulator and an observed object. Building on this representation, we introduce SPACE Policy, an imitation learning framework that uses Gated Bimanual Fusion to perceive spatial relationships between manipulators for self-collision avoidance and Gated Robot-Prior Fusion to perceive object–robot spatial relationships for environmental collision avoidance. SPACE Policy achieves an average success rate of 69.1% across 31 tasks on the RoboTwin 2.0 benchmark with a 11.9% improvement, and an average success rate of 83% across 7 tasks on the RLBench2 benchmark with a 2.2% improvement, and it demonstrates strong collision avoidance capability on real-world tasks. Our code will be released publicly to facilitate reproducibility and further research.

1 INTRODUCTION

Collision avoidance is a key challenge in bimanual manipulation. Collisions can be divided into two categories, environmental collision and self collision (Deng et al., 2025; Zhang et al., 2025). Environmental collision arises primarily from more obstacles inherent to bimanual tasks (Chen et al., 2026; Zhang et al., 2025), as well as the more difficult obstacle avoidance because of the restricted motion for tightly coupled coordination (Zhang et al., 2025; Mishra et al., 2024; Özcan & Oguz, 2025; Deng et al., 2025; Chen et al., 2026). Self Collision arises mainly from overlap between two manipulators' workspace (Lv et al., 2025; Zhao et al., 2023; Shaw et al., 2024; Zhai et al., 2026), and flawed trajectories due to optimization difficulty of high-dimensional action space (Song et al., 2026; Li et al., 2025a; Lu et al., 2025; Liu et al., 2024).

To address collision problem, existing imitation learning methods focus on generating collision-free demonstrations (Li et al., 2026a; Liu et al., 2025a; Mu et al., 2025; Zhou et al., 2025), enhancing scene perception (Chen et al., 2025; Xu et al., 2026), modeling robot kinematic (Lv et al., 2025; Li et al., 2025b), designing physical constraints (Ding et al., 2025; Deng et al., 2025), and motion planning with collision avoidance (Liu et al., 2024). However, the proprioceptive representations in existing methods are based on kinematic information including joint angles (Jiang et al., 2025), end-effector poses (Yang et al., 2025), and kinematic graphs (Lv et al., 2025), but overlook spatial structure of robot. Kinematic information such as joint angle only determines the pose of each robot link, but whether a collision occurs also depends on the shape and size of one link. Kinematic information alone is insufficient to accurately determine whether a collision occurs (Lv et al., 2025; Das & Yip, 2020). Therefore, existing imitation learning policies struggle to avoid cases shown in Fig. 1, in which the manipulator's kinematic axes don't intersect with an entity, but its surface still collides with the entity. Recognizing this limitation, SPACE Policy explicitly models both the kinematics and the spatial structure of robot to address the collision problem.

Motivated by the limitations of existing work, we address the collision problem by proposing RoboCloud as a descriptor of manipulators. RoboCloud models both the geometric surface of entire manipulator and the topology of link connection, which provides both spatial structure and kine-

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.

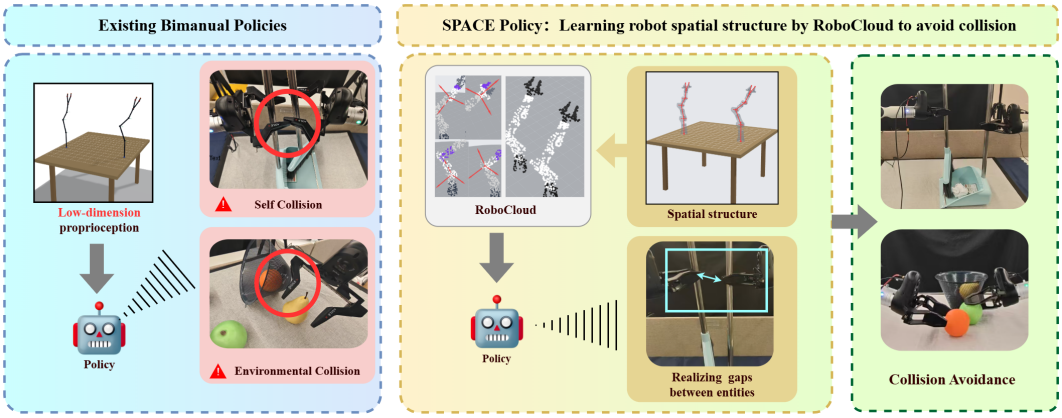


Figure 1: Existing bimanual policies rely on low-dimensional kinematic proprioception and fail to prevent self-collision and environmental collision when link surfaces intersect even if kinematic axes do not. Right: SPACE Policy instantiates RoboCloud to encode the robot's spatial structure and topology, enabling the policy to realize gaps between entities and to generate collision-free actions.

matic of robot to policy. Serving as a structural prior, RoboCloud enables the policy to predict potential surface contact occurring either between the manipulators or between a manipulator and an observed object, thereby avoiding collision.

Building on this representation, we introduce SPatial Arm Collision-free Embedding Policy (SPACE Policy), an imitation learning framework. SPACE Policy employs Gated Bimanual Fusion to fuse features extracted from the RoboCloud of each manipulator into Bimanual Feature that allows the policy to perceive the shapes and relative positions of two manipulators, and thereby help avoid self collision. Gated Robot-Prior fusion further injects Bimanual Feature into visual features and robot state features, enabling the policy to perceive the spatial relationships between scene objects and each robot component during action generation, thereby preventing generating actions that would lead to environment collision.

We evaluate SPACE Policy on 38 simulation tasks on RoboTwin 2.0 and RLBench2, as well as 5 real-world tasks. SPACE Policy achieves an average success rate of 69.1% across 31 tasks on the RoboTwin 2.0 benchmark with a 11.9% improvement, and an average success rate of 83% across 7 tasks on the RLBench2 benchmark with a 2.2% improvement. In particular, for real world tasks prone to collisions, we have significantly reduced the occurrence of collisions and substantially improved the success rates.

Our contributions can be summarized as follows: (1) We tackle the collision problem in bimanual manipulation by constructing a novel robot representation that describes both spatial structure and kinematics of the robot, enabling the policy to perceive the spatial relationships that may lead to collision, thereby helping collision avoidance. (2) Building on this representation, we propose SPACE Policy, an imitation learning framework. It incorporates Gated Bimanual Fusion building spatial relationships between two manipulators to help avoid self collision, and Gated Robot-Prior Fusion connecting the manipulator with scene objects to help avoid environmental collision. (3) We conduct comprehensive evaluations across 38 simulation tasks in bimanual manipulation benchmarks, together with 5 real-world tasks, demonstrating notable increases in average success rates. The occurrence of collisions is significantly reduced during evaluation.

2 Related Work

2.1 Imitation Learning for Bimanual Manipulation

Bimanual manipulation is more challenging than unimanual manipulation: its higher-dimensional action space and mutual coupling between the two manipulators require the policy to maintain bimanual coordination and collision avoidance while generating actions. Early work directly fits high-dimensional bimanual actions through deep behavior cloning and sequence modeling (Xie et al., 2020; Zhao et al., 2023; Park et al., 2024; Zhao et al., 2024). Given the high-dimensional action space and strong inter-arm coupling, some subsequent policies choose joint modeling, treating the

two arms as a joint system and letting the policy simultaneously reason about the actions of both arms in a unified representation (Lv et al., 2025; Mishra et al., 2024; Liu et al., 2025b; Xu et al., 2025), and others choose decoupled modeling, giving each arm a relatively independent policy that is then recoupled through composition, transfer, or role assignment (Jiang et al., 2025; Li et al., 2026b; Im et al., 2026; Lu et al., 2025; Liu et al., 2024). Recent bimanual policies place increasing emphasis on injecting priors such as 3D geometry, multimodal conditioning, and affordance into dual-arm control. GAP (Xu et al., 2026) leverages a pretrained 3D geometric foundation model to jointly predict actions and future scene geometry. DECO (Li et al., 2026b) disentangles vision, proprioception, and tactile inputs through specialized conditioning pathways with a plug-in tactile adapter. A3D (Liang et al., 2026) learns adaptive affordances to identify support and stabilization locations for long-horizon dual-arm assembly.

2.2 Collision Avoidance in Imitation Learning

Collision avoidance is a key challenge in bimanual manipulation. Existing imitation learning methods address it at the data, perception, decision, and control levels. At the data level, methods embed collision constraints into demonstration generation: MoMaGen formulates generation as constrained optimization with hard collision-avoidance constraints (Li et al., 2026a), while RoboTwin synthesizes dual-arm behaviors under spatial constraints and validates collision-free execution (Mu et al., 2025). At the perception level, methods improve perception of objects and bilateral manipulators: BiPreManip predicts anticipatory object poses for collision-free access (Shen et al., 2026), and a decoupled framework shares the other arm's state to avoid inter-arm collisions (Jiang et al., 2025). At the decision level, methods add safety costs or kinematic constraints during action generation: SafeBimanual imposes test-time safety costs on diffusion policies to avoid arm–object collisions and unsafe gripper behavior (Deng et al., 2025), while KStar Diffuser conditions generation on a bimanual spatial–temporal graph and uses differentiable kinematics to resolve self-collision (Lv et al., 2025). At the control level, policies interface with motion planning to produce executable collision-free trajectories: VoxAct-B predicts a collision-avoidance flag for the planner from a voxel-based acting–stabilizing policy (Liu et al., 2024), and AnyBimanual predicts when the planner should invoke collision avoidance for transferred unimanual policies (Lu et al., 2025). Despite their effectiveness, these methods primarily rely on kinematic proprioception and cannot handle the case where the manipulator's kinematic axes do not intersect the entity, but its actual spatial occupancy overlaps with the entity. In contrast, our approach equips the policy with the ability to learn the fine-grained spatial structure of robot, thereby filling the gap that existing methods have not addressed.

3 Method

3.1 Method Overview

SPACE Policy is a structured imitation learning framework that incorporates explicit RoboCloud. At each time step, the system first constructs two sets of robot point clouds separately from the joint angles of left manipulator and right manipulator, then encodes each manipulator's point clouds into an unimanual-spatial feature, then fuse two manipulators' unimanual-spatial features into a bimanual-spatial feature, and finally fuses the bimanual-spatial feature with visual evidence and robot state for action prediction. In compact form, $\mathcal{C}^{L/R} = F_{\text{cloud}}(q^{\text{L/R}})$, $\mathbf{z}_{uni}^{L/R} = F_{\text{enc}}(\mathcal{C}^{L/R})$, $\mathbf{z}_{bi} = F_{\text{bi}}(\mathbf{z}_{uni}^{L}, \mathbf{z}_{uni}^{R})$, $\hat{a}_t = \pi_\theta(\mathbf{z}_{bi}, \mathbf{z}_{vis}, \mathbf{z}_{state})$, where $q^{\text{L/R}}$ denotes the raw joint angles of left or right manipulators, $\mathcal{C}^{L/R}$ denotes the RoboCloud of left or right manipulators, $\mathbf{z}_{uni}^{L/R}$ is the unimanual-spatial feature of left or right manipulators, $\mathbf{z}_{bi}$ is the bimanual-spatial feature of whole robot, $\mathbf{z}_{vis} = F_{\text{vis}}(P_t)$ is the visual feature extracted from the scene point cloud P_t, and $\mathbf{z}_{state} = F_{\text{state}}(s_t)$ is the robot state feature.

Fig. 2 summarizes the full pipeline. RoboCloud Constructor takes joint angles and offline-prepared URDF as input to instantiate each arm's RoboCloud, which consists of geometric point cloud and topological point cloud; RoboCloud Encoder turns each robot arm's RoboCloud into unimanual-spatial feature; Gated Bimanual Fusion module fuses the unimanual-spatial features of two robot arms into a single bimanual-spatial feature; and the Policy Backbone combines bimanual-spatial feature with visual and proprioceptive features to predict actions. RoboCloud is not an external

annotation used only during training, but an intermediate representation through which the policy reasons about the fine-grained spatial structure of the robot when generating actions.

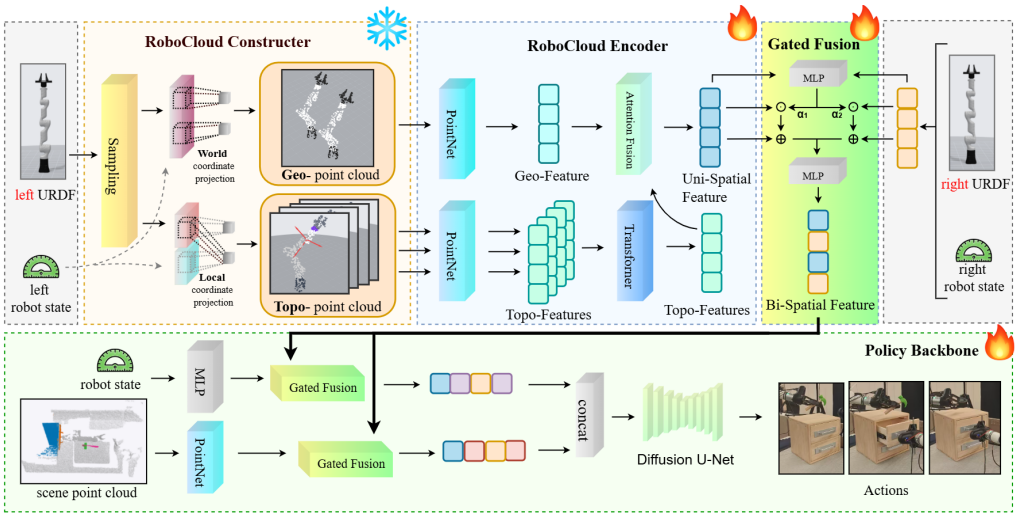


Figure 2: Overview of SPACE Policy. RoboCloud Constructor maps each arm's URDF and joint state into a geometric point cloud (world-frame surface occupancy) and topological point clouds (adjacent-link local frames). A trainable RoboCloud Encoder extracts unimanual spatial features, Gated Bimanual Fusion combines the two arms into a bimanual spatial feature. The Policy Backbone then gated-fuses this feature with scene point clouds and robot state, and a Diffusion U-Net predicts collision-aware actions.

3.2 RoboCloud Definition and Generation

As shown in Fig. 3, RoboCloud is a representation of one manipulator's spatial structure, instantiated from the current joint angles and offline-prepared URDF. Formally, RoboCloud is the union of a singleton set containing the single geometric point cloud and a set of multiple topological point clouds, $\mathcal{C} = \{P^{geo}\} \cup \mathcal{T}$ with $\mathcal{T} = \{P^{topo}_{i\to j}\}_{(i,j)\in\mathcal{E}}$, where geometric point cloud P^{geo} describes the envelope surface of the manipulator's spatial occupancy, and each topological point cloud $P^{topo}_{i\to j}$ captures the kinematics between two adjacent links.

The geometric point cloud describes the envelope surface of the manipulator's spatial occupancy. Geometric point cloud is generated through this pipeline: First, a fixed number of points are sampled from each link's collision mesh surface. Second, the 6D-pose of each link in the world frame is computed from the joint angles. Third, the surface-sampled points of each link are transformed into world coordinates. In compact form, we write the geometric point cloud of one arm as $P^{geo} = \{\mathbf{p}_i\}_{i=1}^{N_{geo}}$, where $N_{geo} = l \cdot M$, l is the number of links on whole manipulator, M is the number of surface samples per link, and each point $\mathbf{p}_i \in \mathbb{R}^3$ lies on the surface of one link.

One topological point cloud describes how two adjacent links are connected. Topological point cloud is generated through this pipeline: For an adjacent link pair $i \to j$, we sample M points on the collision mesh of link i and express them in the local frame of link j, yielding a point set $P^{topo}_{i\to j} = \{\mathbf{p}_k, a_k, b_k\}_{k=1}^{M}$, where $\mathbf{p}_k \in \mathbb{R}^3$ is the point coordinate, and a_k and b_k are the indices of link i and link j, respectively. One manipulator contains multiple such topological point clouds, one for each directed link pair. We combine them into set $\mathcal{T}$, $\mathcal{T} = \{P^{topo}_{i\to j}\}_{(i,j)\in\mathcal{E}}$, where $\mathcal{E}$ denotes the set of all adjacent link pairs in the kinematic chain, including the additional pairs between the gripper components.

The RoboCloud Constructor is a deterministic computational procedure with no learnable parameters. It takes a single manipulator's qpos $q^{\mathrm{L/R}} \in \mathbb{R}^n$ and an offline-prepared URDF as input, and outputs the instantiated RoboCloud $\mathcal{C}_t = \{P^{geo}\} \cup \mathcal{T}$, where $\mathcal{T} = \{P^{topo}_{i\to j}\}_{(i,j)\in\mathcal{E}}$. The geometric point cloud satisfies $P^{geo} \in \mathbb{R}^{N_{geo}\times 3}$, and each topological point cloud satisfies $P^{topo}_{i\to j} \in \mathbb{R}^{M\times 5}$, where the five channels encode the point coordinate (x, y, z) together with the indices of the two adjacent links.

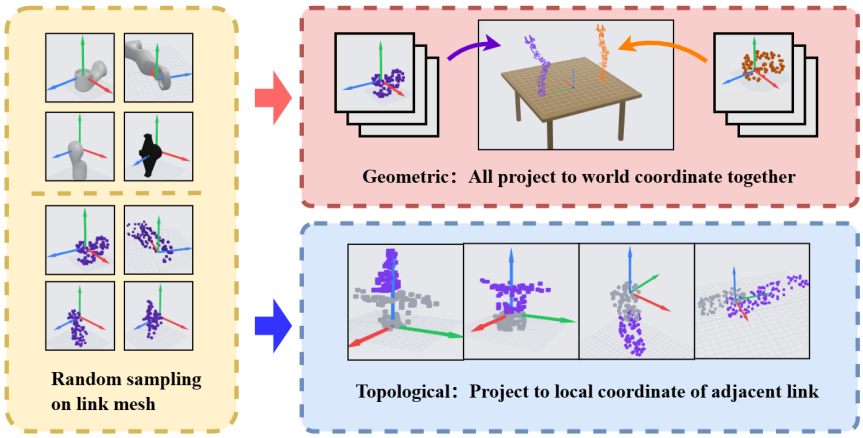


Figure 3: RoboCloud construction. Left: points are randomly sampled on each link's collision mesh in the link-local frame. Top right (geometric): all link samples are transformed into a shared world frame to form a surface occupancy cloud of the full manipulator. Bottom right (topological): for each adjacent link pair, samples of one link are expressed in the local frame of its neighbor, encoding relative kinematics between connected links.

3.3 RoboCloud Encoder

As shown in Fig. 2, RoboCloud Encoder encodes one manipulator's RoboCloud into a unimanual spatial feature: It extracts features from the geometric point cloud and the topological point clouds separately, and fuses these features into one single feature. The workflow of RoboCloud Encoder is as follows: The geometric point cloud is encoded by a PointNet (Qi et al., 2017a) into a geometric feature. Each topological point cloud is encoded by a parameter-shared PointNet (Qi et al., 2017a) into a topological feature. All topological features are then fused into one single global topological feature by a Transformer (Vaswani et al., 2017) encoder followed by mean pooling. Finally, the geometric feature and the global-topological feature are fused by attention fusion to produce the unimanual spatial feature.

3.4 Gated Bimanual Fusion

To obtain a unified bimanual representation, we fuse unimanual-spatial specific features of two manipulators with a learnable fusion function $G : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^D$, the definition of G is:

$$G(\boldsymbol{u}, \boldsymbol{v}) = W[\boldsymbol{u} + \boldsymbol{u} \odot \sigma(\mathrm{MLP}([\boldsymbol{u}; \boldsymbol{v}])) ; \boldsymbol{v} + \boldsymbol{v} \odot \sigma(\mathrm{MLP}([\boldsymbol{v}; \boldsymbol{u}]))] + \mathbf{b} \quad , \tag{1}$$

where $[\cdot;\cdot]$ denotes vector concatenation, $\odot$ denotes element-wise multiplication, σ is the sigmoid function, and MLP is a two-layer perceptron. W and $\mathbf{b}$ are the weights and bias of the output linear projection.

In our implementation, denoting the unimanual-spatial feature of left or right manipulators as $\mathbf{z}^L_{uni}, \mathbf{z}^R_{uni} \in \mathbb{R}^D$, we get bimanual-spatial feature $\mathbf{z}_{bi} \in \mathbb{R}^D$ by

$$\mathbf{z}_{bi} = G(\mathbf{z}^L_{uni}, \mathbf{z}^R_{uni}). \tag{2}$$

3.5 Policy Backbone

As shown in Fig. 2, the Policy Backbone predicts actions from the fused multimodal representation. In parallel with RoboCloud encoding, a PointNet (Qi et al., 2017a) extracts $\mathbf{z}_{vis}$ from the current scene point cloud observation, and a state encoder extracts $\mathbf{z}_{state}$ from the proprioceptive input. The bimanual-spatial feature $\mathbf{z}_{bi}$ is fused with the visual feature and the robot state feature through Gated Robot-Prior Fusion:$\tilde{\mathbf{z}}_{vis} = G(\mathbf{z}_{vis}, \mathbf{z}_{bi}), \tilde{\mathbf{z}}_{state} = G(\mathbf{z}_{state}, \mathbf{z}_{bi}), \mathbf{z}_{global} = [\tilde{\mathbf{z}}_{vis}; \tilde{\mathbf{z}}_{state}]$, where G is the learnable fusion function defined in Eq. 1, and $\mathbf{z}_{global}$ is the global condition that is subsequently fed into diffusion U-Net (Ho et al., 2020) to generate action $\hat{a}_t$.

4 EXPERIMENT

This section evaluates whether RoboCloud improves downstream manipulation policy learning beyond raw visual or proprioceptive input. We first report main results on simulation benchmarks in Sec. 4.1, then evaluate real-world transfer in Sec. 4.2. We further analyze the key design of RoboCloud and model component in Sec. 4.3.

4.1 MAIN EXPERIMENTS IN SIMULATION BENCHMARKS

We evaluate SPACE Policy on two simulation benchmarks: RoboTwin 2.0 (Chen et al., 2026) and RLBench2 (Grotz et al., 2024). Together, they cover tasks demanding spatio-temporal coordination, role assignment, and long-horizon planning.

Experiment Settings. We benchmark SPACE Policy on two suites. RoboTwin 2.0 (Chen et al., 2026) contains 31 representative tasks in our evaluation and focuses on testing whether RoboCloud helps avoid failures due to self collision originating from the gripper and environmental collision in scenarios with rich interaction and cluttered environments. RLBench2 (Grotz et al., 2024) contains 7 representative tasks in our evaluation and focuses on testing whether RoboCloud can similarly help reduce failures due to self collision originating from the arm body under settings with high workspace overlap and large-range motions. Across these two benchmarks, the main metric is task success rate.

RoboTwin 2.0 setup. We train all methods on 100 expert demonstrations per task. The demonstrations are synthesized with 3D generative models and LLMs starting from seed $= 0$, with diverse episode lengths. Baselines are reproduced with the default configurations from their original papers or the RoboTwin 2.0 benchmark. Consistent with GAP (Xu et al., 2026)'s protocol for 3D point-cloud policies, SPACE Policy is trained for 3000 epochs with batch size 256. For evaluation, we run 100 rollouts per checkpoint and report the mean and standard deviation of success rates over three seeds. Training is conducted on H100 GPUs, and evaluation on RTX 4090 GPUs.

RLBench2 setup. Following PPI (Yang et al., 2025), we use seven representative bimanual tasks on RLBench2 and adopt the regenerated demonstration set that corrects train–evaluation misalignment in the official data. Each task uses 100 demonstration episodes. SPACE Policy is trained for 1000 epochs, and we report the average success rate of three checkpoints, each evaluated over 100 rollouts with varied initial states. More detailed training and evaluation protocols are provided in Appendix A.

Baselines. We compare against strong baselines spanning diffusion-based policies, 2D visuomotor policies, and 3D manipulation policies. On RoboTwin 2.0 (Chen et al., 2026), we compare against ACT (Zhao et al., 2023), DP (Chi et al., 2023), DP3 (Ze et al., 2024), G3Flow (Chen et al., 2025), Xu et al. (Xu et al., 2025) and GAP (Xu et al., 2026). On RLBench2, we compare against PerAct2 (Grotz et al., 2024), DP3 (Ze et al., 2024), 3DDA (Ke et al., 2024), KStarDiffuser (Lv et al., 2025), AnyBimanual (Lu et al., 2025) and PPI (Yang et al., 2025).

Results. RoboTwin 2.0 tests bimanual manipulation skills in contact-rich and cluttered scenarios. As shown in Tab. 1, SPACE Policy achieves 69.1% average success rate, outperforming the strongest prior baseline GAP at 57.2%. The gains are particularly meaningful on reducing gripper-induced self-collisions and environmental collisions with cluttered objects. **RLBench2** tests bimanual manipulation skills in overlapping workspaces requiring large-range motions of two manipulators. As shown in Tab. 2, SPACE Policy achieves 83.0% average success rate, outperforming the strongest prior baseline PPI at 80.8%, indicating that RoboCloud benefits the policy by avoiding self-collision occurring on the arm body.

4.2 REAL-WORLD EXPERIMENTS

We evaluate whether the RoboCloud interface transfers from simulation to real manipulation on two Realman 75-BF robot arms, each with 7 DoF and a pika gripper, running at 30 Hz. RGB-D observations are captured by an NVIDIA RealSense D415, and point clouds are constructed with Open3D (Zhou et al., 2018). We reuse four representative baselines from our simulation study for the real-world setting, their proprioceptive representations are diverse: DP3 (Ze et al., 2024) and GAP (Xu et al., 2026), which are joint-angle-based; PPI (Yang et al., 2025), which is TCP-based;

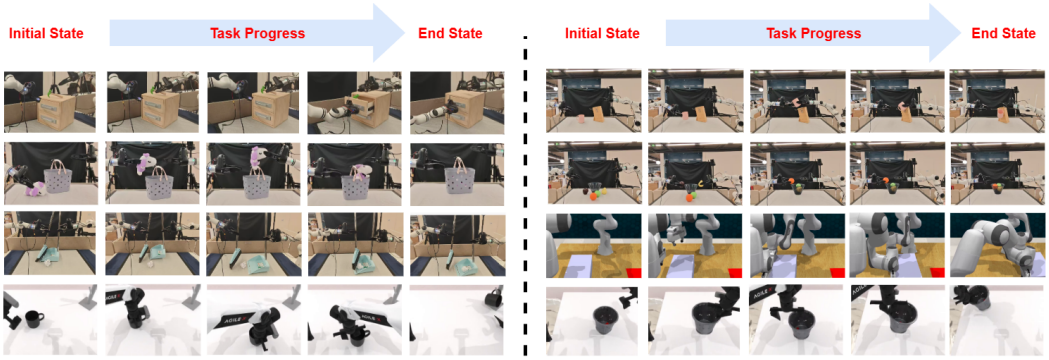


Figure 4: Real-world and simulation experimental rollouts. For all five real-world bimanual tasks (Put Item in Drawer, Handover Toy, Sweep Ball to Dustbin, Hang Mug, Put Fruits in Plate) and selected example tasks in simulation (RLBench2, RoboTwin), we show the initial state, intermediate progress, and successful end state.

and KStarDiffuser (Lv et al., 2025), which is graph-based. We execute each method for 100 trials and report task success rate as the main metric. Fig. 4 illustrates the real-robot platform and the five evaluation tasks.

Task Suite. We design five real-world bimanual tasks that cover synchronized cooperation, sequential coordination, and role assignment, all of which carry a high risk of collision during execution. (1) Handover Toy: one manipulator hands a toy to the other, and the other places it on a basket; prone to self collision during gripper handover and environmental collision with basket handle. (2) Put Fruits in Plate: two manipulators place three fruits into a small plate; prone to self collision when grasping two fruits simultaneously, and environmental collision with neighboring non-target fruit. (3) Put Item in Drawer: one manipulator opens the drawer by its handle while the other places an object into the drawer; prone to self collision when grasping item and handle simultaneously and environmental collision with drawer's non-handle region. (4) Hang Mug: one manipulator hands a mug to the other, and the other hangs it on a thin rod; prone to self-collision during handover and environmental collision with rod when hanging. (5) Sweep Ball to Dustbin: one manipulator holds a dustpan and the other a broom, sweeping a ball into the dustpan together; prone to self-collision during cooperative sweeping and environmental collision with the stand that broom leans against.

Results and Collision avoidance analysis. Tab. 3 reports the success rate over 100 real-robot trials for each task and each method. SPACE Policy transfers robustly to the real world and significantly outperforms all baselines across all five tasks on collision-prone settings. (1) Compared with DP3 (Ze et al., 2024) and GAP (Xu et al., 2026), whose proprioception is joint-angle-based, SPACE Policy performs well on Handover Toy, Hang Mug and Put Fruits in Plate. Joint angles cannot represent the gripper's spatial position and therefore cannot avoid collision when grippers are close to each other or task-irrelevant regions, whereas RoboCloud represents the gripper's occupied entity to solve this issue. (2) Compared with PPI (Yang et al., 2025), whose proprioception is TCP-based, SPACE Policy performs well on Sweep Ball to Dustbin and Hang Mug. TCP cannot represent the spatial structure of the arm body and therefore cannot avoid collisions between the arm body and a thin rod, broom stand, or other robot links, whereas RoboCloud captures the arm body as an occupied entity. (3) Compared with KStarDiffuser (Lv et al., 2025), whose proprioception is graph-based, SPACE Policy performs well on Put Item in Drawer and Put Fruits in Plate, because a graph represents only the kinematic skeleton and cannot explicitly represent the fine-grained spatial geometry of each link, and therefore cannot avoid collisions between link-surface protrusions and the drawer's non-handle region, non-target fruits or other robot links, whereas RoboCloud captures the shape and size of each link to solve this issue. These results are consistent with our central claim: by explicitly modeling both the robot's spatial structure and kinematics through RoboCloud, the policy can avoid collisions in synchronized, sequence-coupled, and role-differentiated bimanual tasks.

4.3 Ablation Studies

We design ablations to answer four questions: (1) whether both the geometric and topological point clouds in RoboCloud are necessary; (2) how the robot's spatial features should be fused across the two arms and fused with the visual and robot state observations; (3) how to choose the sampling

Table 1: Comparison on RoboTwin 2.0. We report the mean ± std of success rates (%) over three seeds. Best results are in **bold**, and second best are underlined.

Method	**Avg. Success↑**	Beat Block Hammer	Handover Block	Hang Mug	Move Can Pot	Open Laptop	Open Microwave	Pick Diverse Bottles
ACT	33.3	59.3±6.4	45.3±2.5	4.3±0.6	48.0±6.0	59.7±2.1	83.7±1.5	11.7±4.0
DP	39.6	62.0±4.0	32.0±1.0	15.3±3.2	51.0±4.0	61.3±3.5	53.7±6.5	18.3±3.8
DP3	50.8	59.3±2.3	29.7±3.1	23.3±7.8	63.7±1.5	77.7±2.1	70.3±2.5	27.7±8.6
G3Flow	52.4	–	–	26.7±3.1	–	–	–	23.0±3.6
Xu et al.	52.5	50.0±5.6	64.3±5.8	25.0±4.6	58.7±2.1	67.7±3.2	–	28.0±1.0
GAP	57.2	56.3±1.2	68.7±0.6	**40.0**±5.3	55.3±2.1	**88.7**±2.5	86.0±1.7	24.3±1.2
SPACE (Ours)	**69.1**	**92.7**±2.5	**95.3**±0.9	38.0±1.4	**93.7**±1.9	87.7±4.8	**93.0**±0.8	**71.7**±2.4

Method	Pick Dual Bottles	Place A2B Left	Place A2B Right	Place Bread Basket	Place Bread Skillet	Place Burger Fries	Place Cans Plasticbox	Place Cont. Plate
ACT	52.7±3.1	1.3±0.6	0.7±0.6	15.7±3.1	12.3±2.5	74.3±2.5	29.3±3.5	50.7±1.5
DP	52.3±0.6	10.0±5.3	11.7±5.0	16.7±3.5	25.0±4.6	76.7±4.2	62.7±3.1	53.0±6.9
DP3	59.0±3.0	29.3±2.5	23.0±2.6	23.7±6.8	36.3±6.7	60.3±3.2	66.0±3.6	92.3±1.5
G3Flow	58.7±1.2	25.7±1.5	25.7±3.2	19.3±4.2	31.0±1.0	73.7±5.0	67.3±2.1	92.3±1.5
Xu et al.	39.7±2.5	24.3±2.1	23.0±2.6	15.0±5.3	37.7±3.1	81.7±4.0	64.7±1.2	93.0±2.0
GAP	69.0±2.0	26.0±4.4	24.7±2.1	20.7±1.2	34.0±6.9	**82.3**±4.7	61.3±6.1	93.3±0.6
SPACE (Ours)	**90.7**±1.9	**53.3**±6.1	**49.0**±2.9	**35.0**±6.5	**54.0**±1.4	81.0±2.4	**90.7**±6.2	**96.3**±1.2

Method	Place Dual Shoes	Place Empty Cup	Place Fan	Place Object Basket	Place Object Stand	Place Phone Stand	Place Shoe	Press Stapler
ACT	3.0±1.0	60.3±2.5	2.3±1.2	6.3±0.6	1.3±1.2	43.3±5.8	14.3±4.0	27.7±1.5
DP	7.0±4.4	62.3±6.7	8.0±3.6	30.0±3.0	42.0±7.5	37.7±7.6	43.3±6.7	28.7±2.5
DP3	17.7±4.2	71.3±2.1	46.3±6.0	32.3±2.5	34.0±0.0	52.0±3.6	57.7±2.5	77.0±2.6
G3Flow	20.7±4.2	73.0±7.2	51.7±5.7	32.3±2.5	–	45.3±4.0	58.3±1.5	74.3±2.1
Xu et al.	40.3±4.2	64.3±8.6	60.0±2.6	52.7±5.0	39.3±4.2	52.7±2.9	58.7±9.1	72.3±3.8
GAP	**43.3**±1.2	76.3±6.0	52.0±2.0	60.3±3.5	42.7±4.2	52.3±1.5	59.7±1.5	**78.0**±3.0
SPACE (Ours)	20.3±3.1	**85.7**±3.7	**65.3**±3.1	**67.3**±3.3	**74.7**±0.9	**62.0**±7.0	**62.7**±3.4	73.0±4.3

Method	Put Bottles Dustbin	Rotate QRcode	Scan Object	Shake Bottle	Stack Blocks Three	Stack Blocks Two	Stack Bowls Three	Stack Bowls Two
ACT	60.0±4.6	3.0±1.0	5.7±3.8	81.7±7.5	0.7±1.2	23.3±3.5	64.3±3.2	85.3±3.2
DP	38.3±5.1	41.7±3.5	10.3±2.5	94.3±1.2	1.3±1.5	17.3±3.8	73.3±3.1	89.3±1.5
DP3	70.3±2.3	76.0±2.0	23.3±1.5	95.0±1.0	3.7±1.5	31.3±3.1	58.0±6.9	86.7±3.8
G3Flow	73.0±4.6	74.3±1.5	22.0±4.6	86.7±2.1	–	–	61.0±2.6	89.7±3.1
Xu et al.	74.0±2.0	68.3±7.2	28.3±3.1	96.3±3.5	0.0±0.0	30.7±3.1	73.3±1.5	90.0±1.0
GAP	75.7±3.2	75.0±4.4	30.7±2.1	93.3±2.9	2.0±1.0	32.0±6.2	74.0±3.6	**95.3**±1.5
SPACE (Ours)	**76.3**±2.1	**77.3**±2.1	**43.3**±2.6	**97.7**±0.5	**4.7**±0.9	**39.0**±2.2	**79.7**±1.7	92.0±1.4

Table 2: Comparison on RLBench2. We report the mean ± std of success rates (%) over three checkpoints. Best results are in **bold**, and second best are underlined.

Method	**Avg. Success↑**	Push Box	Lift Ball	Put item into Drawer	Pick up Laptop	Sweep Dust	Lift Tray	Handover item (easy)
PerAct^2	40.0	62.0±26.2	50.0±8.7	49.7±16.6	36.7±5.7	2.0±3.5	60.0±6.2	19.7±6.0
DP3	26.0	39.3±3.1	27.0±6.6	0.0±0.0	6.0±2.6	98.7±0.6	6.3±0.6	4.7±1.5
3DDA	64.7	54.7±3.8	87.3±1.9	52.7±14.1	40.7±6.1	96.7±2.6	76.0±4.3	44.7±3.3
KStar Diffuser	54.9	83.0±1.7	**98.7**±1.5	15.0±1.3	43.7±4.5	89.0±5.2	28.0±2.7	27.0±1.7
AnyBimanual	35.1	46.0±2.6	36.0±2.0	32.0±2.5	7.0±2.1	67.0±2.0	14.0±2.2	44.0±2.5
PPI	80.8	**96.7**±1.5	89.3±1.5	79.7±3.8	46.3±1.2	98.7±1.5	92.0±1.0	62.7±2.5
SPACE (Ours)	**83.0**	94.0±2.0	78.7±1.5	**85.7**±7.4	**58.3**±1.2	**100.0**±0.0	**96.3**±0.6	**68.0**±0.0

region and the number of points per link for RoboCloud; and (4) how to choose the backbone of the RoboCloud encoder. All ablations are conducted on 31 tasks from RoboTwin 2.0, with 100 demonstrations per task for training and 100 trials per task for evaluation, reporting the average success rate over the 31 tasks under a single seed. Additional protocol details and per-task ablation results are provided in Appendix A and Appendix E.

(1) In this ablation, except for the robot representation, all other model structures and training configurations are kept identical. We separately use only the geometric point cloud, only the topological point cloud, and both, with neither used as the control. As shown in Tab. 4, compared with the setting where neither is used, using only the geometric point cloud or only the topological point cloud both brings significant improvements, indicating that the geometric and topological point clouds each encode the robot's spatial structure but with different emphases: the geometric point cloud tends to characterize the robot's overall spatial configuration, i.e., the surface occupancy of each link in the world frame, while the topological point cloud tends to characterize the local spatial configuration between links, i.e., the relative positions and connections between adjacent links. Using both further brings additional gains, helping the policy understand both the global overall spatial configuration and the fine-grained local spatial configuration, thereby judging potential collisions more accurately. Therefore, both the geometric and topological point clouds are necessary.

(2) In this ablation, except for the fusion methods, all other settings are kept identical. We compare two fusion stages: the first is cross-arm fusion, which fuses the per-arm robot features of the two arms into a single bimanual robot feature; the second is robot feature injection, which injects the bimanual robot feature into the visual observation and the robot state observation. For both stages, we compare MLP and gated fusion. As shown in Tab. 6, gated fusion achieves the best overall performance when used in both stages; replacing either stage with MLP or concat+MLP leads to a drop in performance. This indicates that gated fusion is a more effective unified mechanism for the

Table 3: Real-world results. We report success rate (%) over 100 trials for each real-world manipulation task. Best results are in **bold**, and second best are underlined.

Method	**Avg. Success↑**	Handover Toy	Put Fruits in Plate	Hang Mug	Put Item in Drawer	Sweep Ball to Dustbin
DP3	12.8	29	12	0	15	8
KStar Diffuser	29.6	44	26	19	24	35
PPI	37.4	51	36	21	37	42
GAP	48.0	65	47	32	44	52
SPACE (Ours)	**63.2**	**84**	**68**	**41**	**59**	**64**

Table 4: Ablation study on RoboCloud components.

Exp.	Geometric	Topological	Result
EX1	–	–	50.8%
EX2	✓	–	68.5%
EX3	–	✓	67.6%
EX4	✓	✓	**69.7%**

Table 5: Ablation study on RoboCloud Encoder (PointNet, PointNet++, Point Transformer).

Exp.	PointNet	PointNet++	PointTransformer	Result
EX1	✓	–	–	**69.7%**
EX2	–	✓	–	68.2%
EX3	–	–	✓	69.5%

Table 6: Ablation study on fusion methods of spatial features.

Exp.	Cross-Arm		Injection		Result
	MLP	Gated	MLP	Gated	
EX1	✓	–	✓	–	65.4%
EX2	✓	–	–	✓	68.8%
EX3	–	✓	✓	–	63.3%
EX4	–	✓	–	✓	**69.7%**

Table 7: Ablation study on RoboCloud sampling config.

Exp.	Region		Num. Points			Result
	surface	inner	8	16	32	
EX1	✓	–	✓	–	–	67.8%
EX2	✓	–	–	✓	–	**69.7%**
EX3	✓	–	–	–	✓	68.0%
EX4	–	✓	✓	–	–	69.0%
EX5	–	✓	–	✓	–	69.1%
EX6	–	✓	–	–	✓	67.3%

robot's spatial features in both cross-arm fusion and observation injection, and we therefore adopt gated fusion in both stages.

(3) In this ablation, except for the sampling configuration of RoboCloud, all other settings are kept identical. We compare sampling on the link envelope surface and sampling inside the link envelope surface, and vary the number of points per link over 8, 16, and 32. As shown in Tab. 7, comparing different numbers of points within each sampling method shows that 16 points are already sufficient to represent the spatial structure of each link, and further increasing the number of points does not bring stable improvements. At the best number of points, 16, surface sampling brings higher gains than inner sampling.

(4) In this ablation, except for the RoboCloud Encoder backbone of RoboCloud, all other settings are kept identical. We compare PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), and Point Transformer (Zhao et al., 2021). As shown in Tab. 5, stronger point cloud encoder backbones do not bring stable improvements, indicating that the performance gain mainly comes from the structured representation of RoboCloud itself rather than the capacity of the point cloud encoder. We therefore adopt PointNet (Qi et al., 2017a) by default to control computational cost.

5 Conclusion

We addressed collision avoidance in bimanual manipulation, identifying that standard proprioceptive representations like joint angles reflect link poses but fail to capture the robot's physical spatial occupancy. To resolve this, we introduced SPACE Policy, driven by a structural-prior descriptor, RoboCloud, which models both link geometry and joint topology while using gated fusion mechanisms to prevent self- and environmental collisions. Evaluations across RoboTwin 2.0, RLBench2, and real dual-arm tasks demonstrate substantial improvements in success rates and collision avoidance, with extensive ablations confirming the complementarity of geometric-topological priors and the efficiency of link-surface sampling. These findings highlight explicit spatial modeling as crucial for safe manipulation, opening avenues to extend RoboCloud to diverse embodiments and open-world environments.

AI USE STATEMENT

We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing ingredients for proofs, assisting in writing proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation, cleaning or reformatting datasets, supporting qualitative or thematic data analysis, or interpreting results; proving mathematical claims and qualitative/thematic analysis are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code and artifacts (visualization scripts and LaTeX table-generation helpers); drafting and editing parts of the manuscript to improve readability (Abstract, Introduction, Related Work, Method, Experiment, Conclusion, and the reproducibility statement); summarizing and identifying related literature; formatting tables and references; and polishing figure captions. By section: Abstract/Introduction/Conclusion and Method used AI only for language polishing of author-written drafts; Related Work used AI to suggest candidate papers that we verified against original sources before citation; Experiment used AI for wording polish and table typesetting, while task selection, running experiments, and result interpretation were performed by the authors. We have reviewed all AI-assisted work: AI-suggested papers were checked against the original sources; and all AI-assisted text, tables, and captions were edited and approved by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We are committed to releasing the full source code of SPACE Policy, including RoboCloud construction, training and evaluation scripts, and configuration files used in our experiments, together with pretrained model checkpoints. Anonymous code and realworld experiment vedio will be provided as supplementary material, and public repository with pretrain checkpoint will be released upon acceptance. Implementation details and training/evaluation protocols are described in Sec. 4.1–4.3 and Appendix A.

REFERENCES

Tianxing Chen, Yao Mu, Zhixuan Liang, Zanxin Chen, Shijia Peng, Qiangyu Chen, Mingkun Xu, Ruizhen Hu, Hongyuan Zhang, Xuelong Li, and Ping Luo. G3flow: Generative 3d semantic flow for pose-aware and generalizable object manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 1735–1744, June 2025.

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.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

Nikhil Das and Michael C. Yip. Forward kinematics kernel for improved proxy collision checking. *IEEE Robotics and Automation Letters*, 5(2):2349–2356, 2020.

Haoyuan Deng, Wenkai Guo, Qianzhun Wang, Zhenyu Wu, and Ziwei Wang. Safebimanual: Diffusion-based trajectory optimization for safe bimanual manipulation. In *9th Annual Conference on Robot Learning*, 2025.

Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12248–12255, 2025.

Markus Grotz, Mohit Shridhar, Yu-Wei Chao, Tamim Asfour, and Dieter Fox. Peract2: Benchmarking and learning for robotic bimanual manipulation tasks. In *CoRL 2024 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

HoKyun Im, Euijin Jeong, Andrey Kolobov, Jianlong Fu, and Youngwoon Lee. Twin-vla: Data-efficient bimanual manipulation with twin single-arm vision-language-action models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 61969–61995, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/65475a59e2a2cb707c9462e8a308e980-Paper-Conference.pdf.

Jian-Jian Jiang, Xiao-Ming Wu, Yi-Xiang He, Ling-An Zeng, Yi-Lin Wei, Dandan Zhang, and Wei-Shi Zheng. Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12427–12437, October 2025.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *8th Annual Conference on Robot Learning*, 2024.

Chengshu Li, Mengdi Xu, Arpit Bahety, Hang Yin, Yunfan Jiang, Huang Huang, Josiah Wong, Sujay Garlanka, Cem Gokmen, Ruohan Zhang, Weiyu Liu, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Momagen: Generating demonstrations under soft and hard constraints for multi-step bimanual mobile manipulation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 112425–112446, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/b6b4906c1334656e97cc9968ccfca073-Paper-Conference.pdf.

Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 6991–7003, June 2025a.

Xukun Li, Yu Sun, Lei Zhang, Bo-Sheng Huang, Yibo Peng, Yuan Meng, Haojun Jiang, Shaoxuan Xie, Guocai Yao, Alois Knoll, Zhenshan Bing, Xinlong Wang, and Zhenguo Sun. DECO: Decoupled multimodal diffusion transformer for bimanual dexterous manipulation with a plugin tactile adapter. In *Forty-third International Conference on Machine Learning*, 2026b.

Zhuo Li, Junjia Liu, Dianxi Li, Tao Teng, Miao Li, Sylvain Calinon, Darwin Caldwell, and Fei Chen. Manidp: Manipulability-aware diffusion policy for posture-dependent bimanual manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9956–9962, 2025b.

Jiaqi Liang, Yue Chen, Qize Yu, Yan Shen, Haipeng Zhang, Hao Dong, and Ruihai Wu. A3d: adaptive affordance assembly with dual-arm manipulation. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. URL https://doi.org/10.1609/aaai.v40i22.38907.

I-Chun Arthur Liu, Sicheng He, Daniel Seita, and Gaurav S. Sukhatme. Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation. In *8th Annual Conference on Robot Learning*, 2024.

I-Chun Arthur Liu, Jason Chen, Gaurav S. Sukhatme, and Daniel Seita. D-CODA: Diffusion for coordinated dual-arm data augmentation. In *9th Annual Conference on Robot Learning*, 2025a.

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 Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/49f80e4d2471ad4f2edf4f5f1ab62339-Paper-Conference.pdf.

Guanxing Lu, Tengbo Yu, Haoyuan Deng, Season Si Chen, Yansong Tang, and Ziwei Wang. Anybimanual: Transferring unimanual policy for general bimanual manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13662–13672, October 2025.

Qi Lv, Hao Li, Xiang Deng, Rui Shao, Yinchuan Li, Jianye Hao, Longxiang Gao, Michael Yu Wang, and Liqiang Nie. Spatial-temporal graph diffusion policy with kinematic modeling for bimanual robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17394–17404, June 2025.

Utkarsh Aashu Mishra, Yongxin Chen, and Danfei Xu. Generative factor chaining: Coordinated manipulation with diffusion-based factor graph. In *8th Annual Conference on Robot Learning*, 2024.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 27649–27660, June 2025.

J. Hyeon Park, Wonhyuk Choi, Sunpyo Hong, Hoseong Seo, Joonmo Ahn, Changsu Ha, Heungwoo Han, and Junghyun Kwon. Hierarchical action chunking transformer: Learning temporal multimodality from demonstrations with fast imitation behavior. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12648–12654, 2024.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017b. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.

Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. In *8th Annual Conference on Robot Learning*, 2024.

Yan Shen, Feng Jiang, Zichen He, Xiaoqi Li, Yuchen Liu, Zhiyu Li, Ruihai Wu, and Hao Dong. Bipremanip: Learning affordance-based bimanual preparatory manipulation through anticipatory collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42430–42440, June 2026.

Mingchen Song, Xiang Deng, Jie Wei, Dongmei Jiang, Liqiang Nie, and Weili Guan. Energyaction: Unimanual to bimanual composition with energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20845–20855, June 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Fan Xie, Alexander Chowdhury, M. Clara De Paolis Kaluza, Linfeng Zhao, Lawson Wong, and Rose Yu. Deep imitation learning for bimanual robotic manipulation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2327–2337. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/18a010d2a9813e91907ce88cd9143fdf-Paper.pdf.

Chongyang Xu, Haipeng Li, Shen Cheng, Haoqiang Fan, Ziliang Feng, and Shuaicheng Liu. Action-geometry prediction with 3d geometric prior for bimanual manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35036–35046, June 2026.

Huilin Xu, Jian Ding, Jiakun Xu, Ruixiang Wang, Jun Chen, Jinjie Mai, Yanwei Fu, Bernard Ghanem, Feng Xu, and Mohamed Elhoseiny. Diffusion-based imaginative coordination for bimanual manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11469–11479, October 2025.

Yuyin Yang, Zetao Cai, Yang Tian, Jia Zeng, and Jiangmiao Pang. Gripper Pose and Object Pointflow as Interfaces for Robotic Bimanual Manipulation. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

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.

Xuanran Zhai, Zekai Huang, Longyan Wu, Qianyou Zhao, Qiaojun Yu, Jieji Ren, Ce Hao, and Harold Soh. SkillVLA: Tackling Combinatorial Diversity in Dual-Arm Manipulation via Skill Reuse. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026.

Silong Zhang, Quecheng Qiu, Yingtai Ni, Yuecheng Shao, Ziyang Feng, and Jianmin Ji. Hierarchical framework for constrained dual-arm cooperative manipulation with whole-body collision avoidance. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5131–5138, 2025.

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, October 2021.

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.

Tony Z. Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Seyed Kamyar Seyed Ghasemipour, Chelsea Finn, and Ayzaan Wahid. ALOHA unleashed: A simple recipe for robot dexterity. In *8th Annual Conference on Robot Learning*, 2024.

Huayi Zhou, Ruixiang Wang, Yunxin Tai, Yueci Deng, Guiliang Liu, and Kui Jia. You Only Teach Once: Learn One-Shot Bimanual Robotic Manipulation from Video Demonstrations. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

Melih Özcan and Ozgur S. Oguz. Opac: An optimization-augmented control framework for single and coordinated multi-arm robotic manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12240–12247, 2025.

A Training and evaluation protocols

This appendix complements Sec. 4.1–4.3 with additional implementation-level details of SPACE Policy. We follow the same experimental reporting conventions as in the main paper, and expand on observation formats, RoboCloud sampling, optimization settings, and evaluation protocols that are only briefly summarized in the main text.

Table 8: Hyperparameter summary for SPACE Policy.

Item	RoboTwin 2.0	RLBench2
#tasks	31	7
Demonstrations / task	100	100
Eval episodes / checkpoint	100	100
Report statistic	mean±std over 3 seeds	mean±std over top-3 checkpoints
Ablations	31-task mean, single seed	—
Training epochs	3000	1000
Batch size	256	128
Optimizer	AdamW	AdamW
Learning rate / weight decay	1×10^{-4} / 1×10^{-6}	1×10^{-4} / 1×10^{-6}
LR schedule	cosine, warmup 500	cosine, warmup 500
EMA	yes	yes
Observation horizon / action steps	3 / 6	keyframe + continuous (PPI-style)
Diffusion train / infer steps	100 / 10 (DDIM)	1000 / 1000
RoboCloud points / link	16 (surface)	16 (surface)
Robot encoder	PointNet (Qi et al., 2017a)	PointNet (Qi et al., 2017a)
Train / eval GPUs	H100 / RTX 4090	multi-GPU / multi-GPU

A.1 RoboTwin 2.0

Data and task protocol. We evaluate on the 31 bimanual tasks used in Tab. 1. Each method is trained with 100 expert demonstrations per task under the clean demonstration setting without domain randomization. Demonstrations are collected once and shared across SPACE Policy and all ablations, so differences in success rate are attributable to representation and fusion design rather than to changes in the demonstration distribution. The embodiment is a dual-arm platform with 14-dimensional proprioception and joint-space actions (7 dimensions per arm, including the gripper). Scene observations are provided as point clouds from head and wrist depth cameras; the policy consumes XYZ coordinates only and does not use RGB color channels.

Training configuration. Consistent with the GAP (Xu et al., 2026) protocol for 3D point-cloud policies, SPACE Policy is trained for 3000 epochs with batch size 256. We use AdamW with learning rate 1×10^{-4}, weight decay 1×10^{-6}, and cosine learning-rate decay with 500 warmup steps. Exponential moving average (EMA) of the policy parameters is enabled throughout training. The diffusion backbone follows a standard 1D temporal U-Net conditioned by FiLM features: the observation horizon is 3, the predicted action horizon is 6, training uses 100 denoising steps, and inference uses 10 DDIM steps. We save the final epoch-3000 checkpoint for evaluation. Training is conducted on H100 GPUs, and closed-loop evaluation is conducted on RTX 4090 GPUs.

Evaluation and reporting. For each checkpoint we run 100 rollouts with unseen language instructions and varied initial states. The main RoboTwin table (Tab. 1) reports the mean±std of success rates over three independent training seeds. Ablations in Sec. 4.3 use the same data budget, network capacity (unless the ablated component itself changes), and evaluation protocol, but report the 31-task average under a single seed for computational efficiency. Under this single-seed protocol, the full SPACE configuration reaches 69.7%, while the three-seed average reported in the main table is 69.1%. The gap between the two numbers is therefore a difference in aggregation, not a change in the method.

A.2 RLBench2

Data and task protocol. Following PPI, we use seven representative bimanual tasks and the regenerated demonstration set that corrects train–evaluation misalignment in the official data. Each task uses 100 demonstration episodes. The embodiment is DualPanda; RoboCloud is instantiated from the full link mesh of both arms, so that self-collision along the arm body—the failure mode emphasized on this benchmark—is visible in the geometric and topological clouds.

Training and reporting. SPACE Policy is trained for 1000 epochs with batch size 128 under multi-GPU data parallelism. The optimizer, learning-rate schedule, and EMA settings match the RoboTwin recipe. Unlike RoboTwin, where we report a fixed final checkpoint over multiple seeds, RLBench2 follows the PPI-style protocol: for each task we evaluate multiple checkpoints with 100 episodes each, keep the top-three checkpoints ranked by success rate, and report their mean±std. Averaging the seven task means yields the 83.0% result in Tab. 2.

B RoboCloud sampling and construction details

This section expands the RoboCloud construction in Sec. 3 with practical details of how point clouds are sampled and updated at runtime. These choices are shared by the main experiments and by the sampling ablations in Tab. 7.

B.1 Link meshes and offline templates

RoboCloud is constructed from an offline robot description (URDF) together with the current joint configuration. For each rigid link we use the collision mesh associated with that link, rather than a simplified stick model or end-effector frame alone. Before training, we draw a fixed set of candidate points on every link mesh in the link-local coordinate frame. At every control step, these local samples are rigidly transformed by forward kinematics; no additional mesh raycasting is required online. This design makes RoboCloud deterministic given the joint angles, and keeps the computational cost of construction comparable to a standard FK query.

B.2 Surface sampling versus inner sampling

By default we sample on the *surface* of each link's collision mesh. Surface samples approximate the outer envelope of the manipulator and are therefore directly informative about whether two bodies are about to touch. In the sampling ablation we also consider *inner* sampling, where points are drawn in the interior of the link volume rather than on the boundary. Inner samples still convey occupancy, but they under-represent the precise contact surface; Tab. 7 shows that, at the same number of points per link, surface sampling is consistently preferable.

In both cases, the number of points per link is denoted by M. We study $M \in \{8, 16, 32\}$ and adopt $M{=}16$ as the default: increasing from 8 to 16 improves the 31-task average, whereas further increasing to 32 does not yield a stable gain. Intuitively, 16 surface points already outline the major geometric extents of each link (length, thickness, and gripper opening), while denser sampling mostly adds redundant points that the subsequent PointNet (Qi et al., 2017a) aggregates away.

B.3 Geometric cloud

For the geometric branch, the M local samples of every link are transformed into a shared world frame and concatenated into a single cloud $P^{\text{geo}} \in \mathbb{R}^{(l \cdot M) \times 3}$, where l is the number of links on one arm. On the RoboTwin dual-arm platform, each arm contributes on the order of nine links (including gripper components), so the default geometric cloud contains 9×16 points per arm before batching over time. This cloud describes the global spatial occupancy of the manipulator and is the primary cue for predicting surface contact with objects or with the other arm.

B.4 Topological clouds

For the topological branch, we consider every directed adjacent link pair $(i \to j)$ on the kinematic chain, including gripper sub-links. Given such a pair, the M surface samples of link i are expressed in the local frame of link j, and each point is annotated with the discrete indices of the two links. The resulting cloud therefore encodes not only relative geometry, but also which kinematic connection the geometry belongs to. Repeating this construction for all adjacent pairs yields the set of topological clouds used by the RoboCloud Encoder. Because each topological cloud lives in a neighbor-centric frame, it is particularly suited to capturing local articulation and near-self-contact along the chain, complementary to the world-frame geometric occupancy.

B.5 Runtime update rule

At time t, only the joint angles change; the local sampling templates remain fixed. Forward kinematics maps each template into the current world or neighbor frame, producing the RoboCloud instance $\mathcal{C}_t$ consumed by the encoder. Importantly, we always apply FK to *raw* joint angles in radians. Normalized proprioceptive features used by the state MLP are never fed into FK, because normalization would distort link poses and corrupt the geometric prior. The same construction is used at train-

ing time and at closed-loop evaluation time, so there is no train–test mismatch in the robot spatial representation.

C Architecture and optimization details

Scene and state encoders. The scene point cloud is encoded by a lightweight PointNet (Qi et al., 2017a) into a 128-dimensional visual feature. Proprioception is encoded by an MLP into a 64-dimensional state feature. After gated injection of the bimanual RoboCloud feature, the concatenated global condition has dimensionality proportional to the observation horizon (on RoboTwin, horizon 3 yields a 576-dimensional FiLM condition for the diffusion U-Net).

RoboCloud Encoder capacity. Unless ablated, both the geometric cloud and each topological cloud are encoded by PointNet (Qi et al., 2017a). Topological features are aggregated by a small Transformer (Vaswani et al., 2017) encoder (4 layers, 8 heads, hidden size 128) followed by mean pooling, and then combined with the geometric feature by attention fusion into a unimanual spatial feature of dimension 128. Gated Bimanual Fusion and Gated Robot-Prior Fusion share the gated residual form in Eq. 1, each implemented with a two-layer MLP and a sigmoid gate. In the fusion ablation, replacing either gate with an MLP or concat-MLP pathway reduces the 31-task average, which supports using the same gated mechanism at both fusion stages.

Encoder-backbone ablation. Replacing PointNet (Qi et al., 2017a) with PointNet++ (Qi et al., 2017b) or Point Transformer (Zhao et al., 2021) for the robot clouds does not improve the overall average (Tab. 5). We therefore keep PointNet (Qi et al., 2017a) as the default robot encoder: it is sufficient to read out the structured geometry already present in RoboCloud, and it avoids unnecessary compute on the dual-arm point sets.

D Real-world deployment notes

The real-world setup follows Sec. 4.2. Two Realman 75-BF arms with pika grippers run at 30 Hz. RGB-D observations come from a RealSense D415; point clouds are reconstructed with Open3D (Zhou et al., 2018) and downsampled to the same cardinality used in simulation. RoboCloud is built from the Realman URDF and the measured joint angles with the same surface-$M{=}16$ recipe as in simulation, so the policy sees a consistent robot spatial prior across domains. Each method is evaluated for 100 trials per task; we report binary task success and qualitatively inspect collision events during execution.

E Per-task ablation results on RoboTwin 2.0

Tables 9–10 list per-task success rates (%) for the ablations in Sec. 4.3. Each entry is averaged over 100 evaluation trials under a single seed. Column headers follow the experiment indices in Tabs. 4–5: **F**-* denotes fusion variants, **C**-* denotes RoboCloud component variants (geometry-only / topology-only), **S**-* denotes sampling region and M variants, **E**-* denotes robot-encoder backbone variants, and **Ours** is the full SPACE configuration (69.7% mean under this single-seed protocol). Configurations that are identical to Ours by construction are omitted.

Table 9: Per-task ablation success rates (%) on RoboTwin 2.0 (part 1). Single seed, 100 trials per cell.

Task	F-EX1	F-EX2	F-EX3	C-EX2	C-EX3	S-EX1	S-EX3	S-EX4	S-EX5	S-EX6	E-EX2	E-EX3	Ours
Beat Block Hammer	95	92	95	92	91	90	97	96	93	92	94	95	96
Handover Block	98	95	95	97	93	97	92	97	97	95	93	96	96
Hang Mug	26	40	25	38	36	36	38	36	34	31	33	33	39
Move Can Pot	96	95	92	94	94	94	96	95	97	96	96	96	91
Open Laptop	86	92	89	94	87	90	92	89	83	91	88	95	92
Open Microwave	92	87	95	90	91	84	91	95	93	85	91	93	92
Pick Diverse Bottles	60	71	58	73	74	67	67	66	68	70	75	74	70
Pick Dual Bottles	72	87	71	85	87	91	86	92	90	90	91	90	92
Place A2B Left	52	52	44	51	51	55	49	52	54	50	57	53	49
Place A2B Right	53	45	54	59	47	48	49	51	55	54	53	55	48
Place Bread Basket	38	38	34	41	35	35	34	33	35	32	33	45	43
Place Bread Skillet	34	39	36	52	50	51	50	50	49	49	43	38	53
Place Burger Fries	82	84	68	80	83	83	82	88	84	81	80	84	84
Place Cans Plasticbox	79	90	62	93	93	91	93	93	95	94	91	88	96
Place Container Plate	99	97	97	93	96	92	96	95	97	94	96	98	96
Place Dual Shoes	23	19	14	14	20	26	23	18	22	17	18	17	22

Table 10: Per-task ablation success rates (%) on RoboTwin 2.0 (part 2) and 31-task means. Single seed, 100 trials per cell. Ours mean $= \mathbf{69.7}\%$.

Task	F-EX1	F-EX2	F-EX3	C-EX2	C-EX3	S-EX1	S-EX3	S-EX4	S-EX5	S-EX6	E-EX2	E-EX3	Ours
Place Empty Cup	92	87	90	86	87	86	88	86	87	88	87	90	86
Place Fan	63	60	55	59	56	65	60	59	68	61	61	63	67
Place Object Basket	62	71	63	65	70	71	69	65	74	71	67	68	68
Place Object Stand	73	75	68	74	77	75	80	79	77	76	79	81	76
Place Phone Stand	51	66	60	58	50	52	54	64	52	54	54	64	54
Place Shoe	58	69	62	63	61	66	67	67	60	63	65	71	66
Press Stapler	67	78	74	75	76	73	70	76	80	73	78	71	79
Put Bottles Dustbin	62	71	68	76	75	69	76	66	73	76	71	78	74
Rotate QR Code	80	84	74	77	75	68	75	79	71	77	78	76	80
Scan Object	28	38	17	36	28	36	31	41	36	29	34	33	42
Shake Bottle	98	96	99	98	99	98	94	97	99	98	98	97	98
Stack Blocks Three	7	6	7	8	4	7	5	7	7	8	5	10	6
Stack Blocks Two	32	41	38	41	45	36	36	38	40	28	38	32	37
Stack Bowls Three	76	81	71	71	77	79	74	80	83	73	78	77	78
Stack Bowls Two	93	88	87	89	88	92	93	89	88	91	90	95	90
Mean	**65.4**	**68.8**	**63.3**	**68.5**	**67.6**	**67.8**	**68.0**	**69.0**	**69.1**	**67.3**	**68.2**	**69.5**	**69.7**