Gripper Keypose and Object Pointflow as Interfaces for Bimanual Robotic Manipulation

Yuyin Yang[*,1,2] Zetao Cai[*,1,3] Yang Tian[1,4] Jia Zeng[1] Jiangmiao Pang[1†]
[1]Shanghai AI Laboratory [2]Fudan University [3]Zhejiang University [4]Peking University
[*]Equal contributions [†]Corresponding author
https://yuyinyang3y.github.io/PPI/

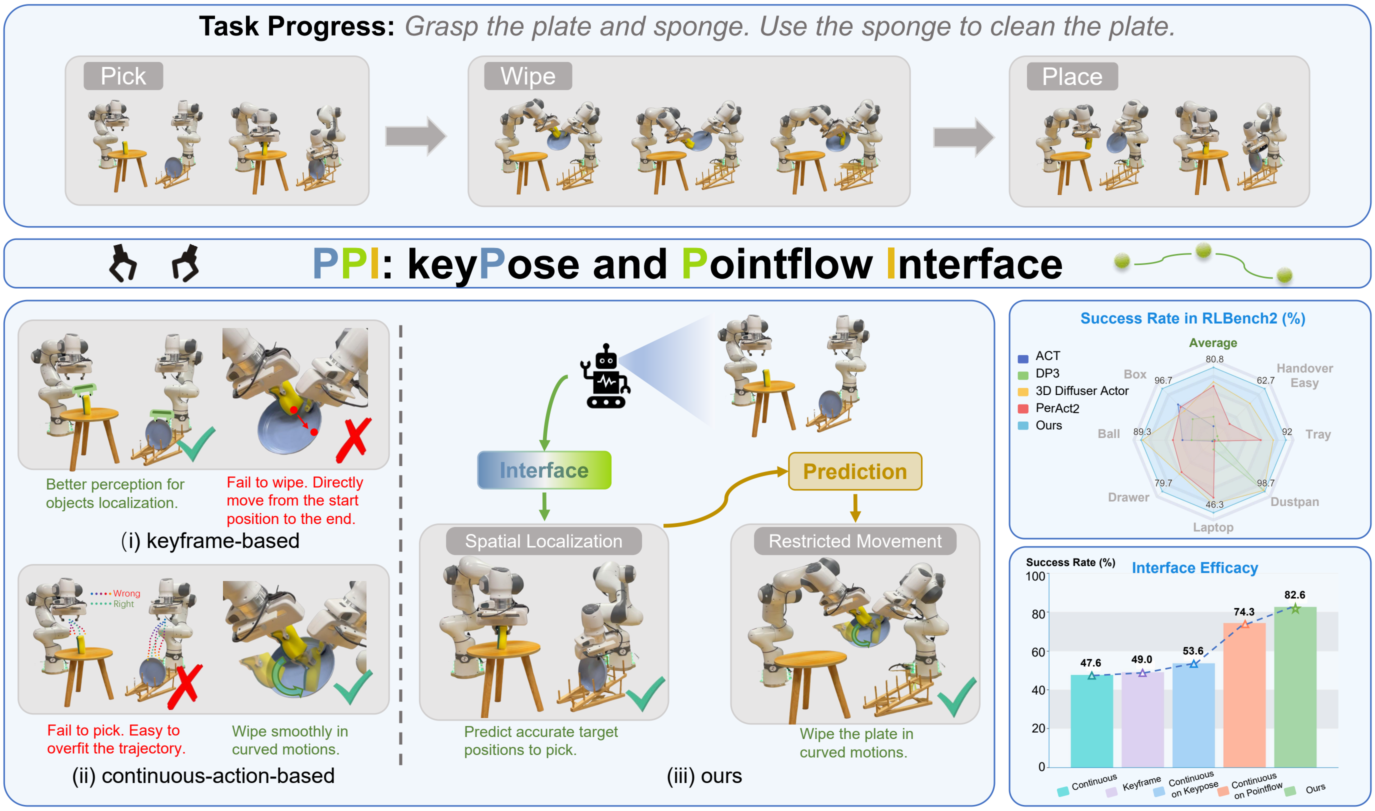
Task Progress: Grasp the plate and sponge. Use the sponge to clean the plate.
Pick
Wipe
Place
PPI: keyPose and Pointflow Interface
Better perception for objects localization.
Fail to wipe. Directly move from the start position to the end.
(i) keyframe-based
Wrong
Right
Fail to pick. Easy to overfit the trajectory.
Wipe smoothly in curved motions.
(ii) continuous-action-based
Interface
Prediction
Spatial Localization
Restricted Movement
Predict accurate target positions to pick.
Wipe the plate in curved motions.
(iii) ours
Success Rate in RLBench2 (%)
ACT
DP3
3D Diffuser Actor
PerAct2
Ours
Average
80.8
Handover Easy
62.7
Tray
92
Dustpan
98.7
Laptop
46.3
Drawer
79.7
Ball
89.3
Box
96.7
Success Rate (%)
Interface Efficacy
100
80
60
40
20
0
47.6
49.0
53.6
74.3
82.6
Continuous
Keyframe
Continuous on Keypose
Continuous on Pointflow
Ours

Fig. 1: In contrast to (i) keyframe-based policies, which excel in spatial localization but struggle with movement restrictions (e.g., curved motion and collision-free actions), and (ii) continuous-action-based policies, which accommodate diverse trajectories but lack strong perception, we introduce a continuous action policy that incorporates two interfaces: target gripper poses and object pointflow, balancing task diversity with spatial awareness. Our model, PPI, surpasses previous states of the art and consistently outperforms its ablated variants.

***Abstract*—Bimanual manipulation is a challenging yet crucial robotic capability, demanding precise spatial localization and versatile motion trajectories, which pose significant challenges to existing approaches. Existing approaches fall into two categories: keyframe-based strategies, which predict gripper poses in keyframes and execute them via motion planners, and continuous control methods, which estimate actions sequentially at each timestep. The keyframe-based method lacks inter-frame supervision, struggling to perform consistently or execute curved motions, while the continuous method suffers from weaker spatial perception. To address these issues, this paper introduces an end-to-end framework PPI (keyPose and Pointflow Interface), which integrates the prediction of target gripper poses and object pointflow with the continuous actions estimation. These interfaces enable the model to effectively attend to the target manipulation area, while the overall framework guides diverse and collision-free trajectories. By combining interface predictions with continuous actions estimation, PPI demonstrates superior performance in diverse bimanual manipulation tasks, providing enhanced spatial localization and satisfying flexibility in handling movement restrictions. In extensive evaluations, PPI significantly outperforms prior methods in both simulated and real-world experiments, achieving state-of-the-art performance with a +16.1% improvement on the RLBench2 simulation benchmark and an average of +27.5% gain across four challenging real-world tasks. Notably, PPI exhibits strong stability, high precision, and remarkable generalization capabilities in real-world scenarios.**

I. Introduction

Endowing robots with dexterous bimanual skills similar to humans has become a main focus in robotic manipulation [3, 6, 14, 34, 36]. Recent efforts primarily fall into two categories: one focuses on "keyframe", like [9, 13], which predict actions in the reference frames and execute predictions via Inverse Kinematics (IK) solvers and motion planners [31]. The other emphasizes "continuous" and perform naive behavior cloning at each time step. For example, ACT [48] and RDT [18] learn RGB-based manipulation policies from multiple cameras, while DP3 [46] integrate 3D scene-level representations into a diffusion policy.

However, due to the temporal granularity of action prediction, the keyframe-based methods predicts actions only at few keyframes. This sparse supervision encourages the model to focus more on local features, enhancing spatial perception. Nevertheless, for tasks involving restricted movements (e.g., wiping a plate, which requires curved trajectories), keyframes are difficult to define and the motion planner [31, 30] tends to output near straight-line paths, making such tasks challenging to accomplish. For continuous-based methods, they are generally applicable to a wide range of tasks. Whereas, since they rely on naive behavior cloning with dense supervision on actions, the model tends to "take shortcuts" by overfitting to seen trajectories (e.g., proprioception). This results in weaker spatial perception capabilities. Therefore, implementing diverse general bimanual tasks while preserving strong perceptual capabilities remains a critical challenge.

To this end, this paper presents a simple yet effective end-to-end interface-based continuous policy that integrates the strengths of previous approaches. As illustrated in Figure 1, our model predicts continuous actions conditioned on two key interfaces: the target gripper keypose and object pointflow. These interfaces enable the model to capture fine-grained spatial features and comprehensively model the interaction between the robot and the object. We implement a diffusion transformer [13, 46] to process both interfaces, naming our approach PPI. By distilling spatial knowledge from these interfaces, PPI strikes a balance between handling diverse tasks and maintaining strong perception capabilities. Leveraging a unidirectional attention within the transformer, PPI progressively infers actions and is trained in an end-to-end manner.

We conduct extensive experiments on both simulation and real-world benchmarks. On the bimanual manipulation benchmark RLBench2 [9], our method achieves a 16.1% higher success rate across seven representative tasks compared to state-of-the-art baselines. We further provide comprehensive visualizations to validate the effectiveness of the two interfaces. Additionally, we evaluate our approach on four challenging real-world tasks [42], demonstrating superior performance in long-horizon task execution, generalization to unseen objects, robustness to lighting variations, and resilience against visual distractions.

Our contributions are summarized as follows:

- We present a novel framework that utilizes keyframe information to guide continuous action generation, improving flexibility in addressing movement constraints.
- We propose two effective interfaces—target gripper poses and object pointflow to boost spatial localization and generalization.
- We provide comprehensive analyses to validate the power of two interfaces. We achieve the state-of-the-art performance on a bimanual simulation benchmark and demonstrate strong robustness, effectiveness, and generalization in real-world long-horizon tasks.

II. Related Works

A. Behavior Cloning in General Bimanual Manipulation Tasks

Current behavior cloning methods for general bimanual manipulation tasks can be broadly classified into two categories. The first category involves keyframe-based strategies [9, 16, 8], where keyframe representations [2, 49, 33] are learned and executed through motion planners. Approaches such as PerAct2 [9] and VoxAct-B [16] predict target gripper poses in a reference frame using voxel-based representations. Additionally, DualAfford [49] learns collaborative object-centric affordances and applies heuristic policies for execution. However, these methods rely on rule-based keyframe split and motion planners, which limits their ability to handle tasks that require irregular motion trajectories (e.g., dishwashing) or strict temporal coordination (e.g., tray lifting). The second category involves continuous control, where actions are estimated sequentially at each time step. For instance, ACT [48] uses an action-chunking transformer to predict actions in an end-to-end manner, while RDT [18] employs a diffusion-based transformer, pre-trained on large robot datasets and fine-tuned on self-collected bimanual data. BiKC [44], is an RGB-based hierarchical framework consisting of a high-level keypose predictor and a low-level trajectory generator. Some works also extend single-arm manipulation policies [4, 46] to the bimanual setting. In contrast to these continuous control approaches, our method integrates a 3D semantic neural field [37, 38] and predicts pointflow as an additional interface, thereby enhancing spatial localization capabilities.

B. Flow-based Methods in Robotic Manipulation

Robot manipulation policies have utilized either 2D pixel-level motion [12, 32, 25, 41] or 3D point-level flow [47, 20, 5] for object interaction. In 2D flow-based approaches, recent pixel-tracking algorithms [12] estimate motion flows in robotic video data. Track2Act [1] integrates a residual strategy atop heuristic and flow-based policies, while ATM [40] learns a flow-conditioned behavior cloning policy trained on self-collected, in-domain data. Im2Flow2Act [43] further introduces a data-efficient, fully autonomous flow-conditioned policy, leveraging task-agnostic datasets for one-shot real-world transfer. Unlike these 2D methods, PPI leverages 3D point-level flow, enhancing spatial localization and enables more accurate manipulation. This approach builds upon prior work in 3D flow-based policies, which have shown promising results in articulated object manipulation [7], tool use [26, 22], and general skill learning [45]. However, these methods typically rely on manually designed or heuristic policies during execution after estimating 3D flow. In contrast, PPI introduces an end-to-end manipulation policy, eliminating the need for heuristic post-processing.

III. Method

In this section, we describe PPI in detail. We begin with a brief problem formulation (Section III-A). Next, we discuss the perception module (Section III-B) in PPI, involving the construction of 3D semantic neural field and initial query points. Subsequently, we elaborate on the key interface designs—Pointflow and Keypose (Section III-C), enhancing PPI spatial localization capabilities. Then, we illustrate the action prediction module (Section III-D), which is a diffusion-based transformer with unidirectional attention. Finally, we provide a detailed implementation details during training and inference phases (Section III-E).

A. Problem Formulation

At time step t, PPI takes inputs as the language instruction l and RGBD images $\mathcal{I}$ from K cameras, and outputs a sequence of h^c continuous actions $a_t^c = \{a_{t:t+h^c}\}$, where each action a_t represents the target gripper poses and openness for both the left and right grippers. Crucially, PPI incorporates two intermediate ***interfaces*** at ***keyframe*** timesteps as additional conditions for action prediction. The ***keyframes*** t^k are defined as turning points in the trajectory where there are significant changes in the grippers' openness and the arms' joint states [11, 27]. For the ***interfaces***, the first specifies the target gripper poses at the subsequent h^k keyframes: $a_t^k = \{a_{t_i^k}\}_{i=1}^{h^k}$. The second interface defines the positions of N_q spatial query points at the next h^k keyframes: $F \in \mathbb{R}^{h^k \times N_q \times 3}$. At each keyframe timestep t_i^k, the positions of the N_q points are denoted as $F_{t_i^k} \in \mathbb{R}^{N \times 3}$, with initial positions at the first frame given by $F_0 \in \mathbb{R}^{N \times 3}$.

B. Perception

3D Scene Representation. As is seen in Figure 2(a), we represent the scene using 3D semantic fields, focusing on both geometric and semantically meaningful regions. We begin by preprocessing the raw point clouds through cropping and downsampling. For each sampled 3D point, we project it onto 2D RGB images from multiple camera viewpoints to extract pixel-wise semantic features using the DINOv2 [19] model. We fuse these features through a weighted sum, where the weights are determined by the point's distance from the projected surface.

To mitigate the computational burden of numerous scene tokens in the transformer backbone, we downsample scene points while preserving their geometric and semantic information. We use a PointNet++ dense encoder [23] to obtain a compact scene representation $S_t \in \mathbb{R}^{N_s \times (3+D)}$, where each of the N_s points encodes spatial coordinates and a D-dimensional fused semantic feature. This compressed representation retains key geometric and semantic details while enhancing local point relationships through the set abstraction of PointNet++.

Initial Query Points Sampling. Instead of directly learning the pointflow distribution $p(F)$, we choose to approximate the conditional distribution $p(F|F_0)$, where F_0 represents the initial query points sampled from the manipulated object at the first frame. This approach shifts the model's focus from inferring global absolute coordinates to capturing overall object motion. Consequently, even when the object position is out of distribution, PPI can estimate pointflow accurately and robustly, demonstrating improved generalization.

To get the initial query points F_0, we randomly sample N_q query points from the object to be manipulated at the beginning of the task. We use the Grounding DINO model [17] to obtain a bounding box from the language prompt and image, then input the bounding box into the SAM model [15] to generate the object mask. We obtain the 3D coordinates $F_0 \in \mathbb{R}^{N_q \times 3}$ of the N_q pixels sampled from the mask. In practice, we find that $N_q = 200$ points are sufficient for all tasks, both in simulation and the real world. It is worth noting that in each episode, we will only perform this operation once.

C. Interface

Target Gripper Poses. We predict target gripper poses at keyframe timesteps as explicit action goals to better guide continuous action generation. To supervise the target gripper poses, we first use a heuristic algorithm to identify the keyframe timesteps in the trajectory. Once a keyframe t_i^k is established, its corresponding action label $a_{t_i^k}$ can be directly retrieved. If there are fewer keyframes remaining after the current timestep than h^k, we pad the sequence by repeating the action of the last keyframe.

Object Pointflow. A key challenge in obtaining ground truth labels for pointflow is the inevitable occlusion that occurs when the object moves or is manipulated. We address this challenge by leveraging the object's 6D pose to track the real-time points' positions. In simulation, we obtain the ground truth 6D pose label of rigid objects from the RLBench2 [9] dataset, while in the real world, we estimate it using BundleSDF [39] and Foundation Pose [2]. Given the object's 6D pose at the first and each keyframe timestep, we transform the query points from the first frame F_0 into the object's coordinate frame and then back into world coordinates, yielding their keyframe positions $F_{t_i^k} \in \mathbb{R}^{N \times 3}$, which serve as the ground truth for pointflow supervision. Here real-time object 6D pose estimation is not required during inference. Overall, object pointflow along with target gripper poses effectively model the interaction between the object and the robot.

D. Prediction

Observation Encoder. PPI processes four types of inputs: the 3D semantic neural field S_t, the language instruction ℓ, the robot states c_t and the initial positions of point queries F_0. The language is encoded using a CLIP text tokenizer [24] and projected into a latent space via a three-layer MLP. Similarly,

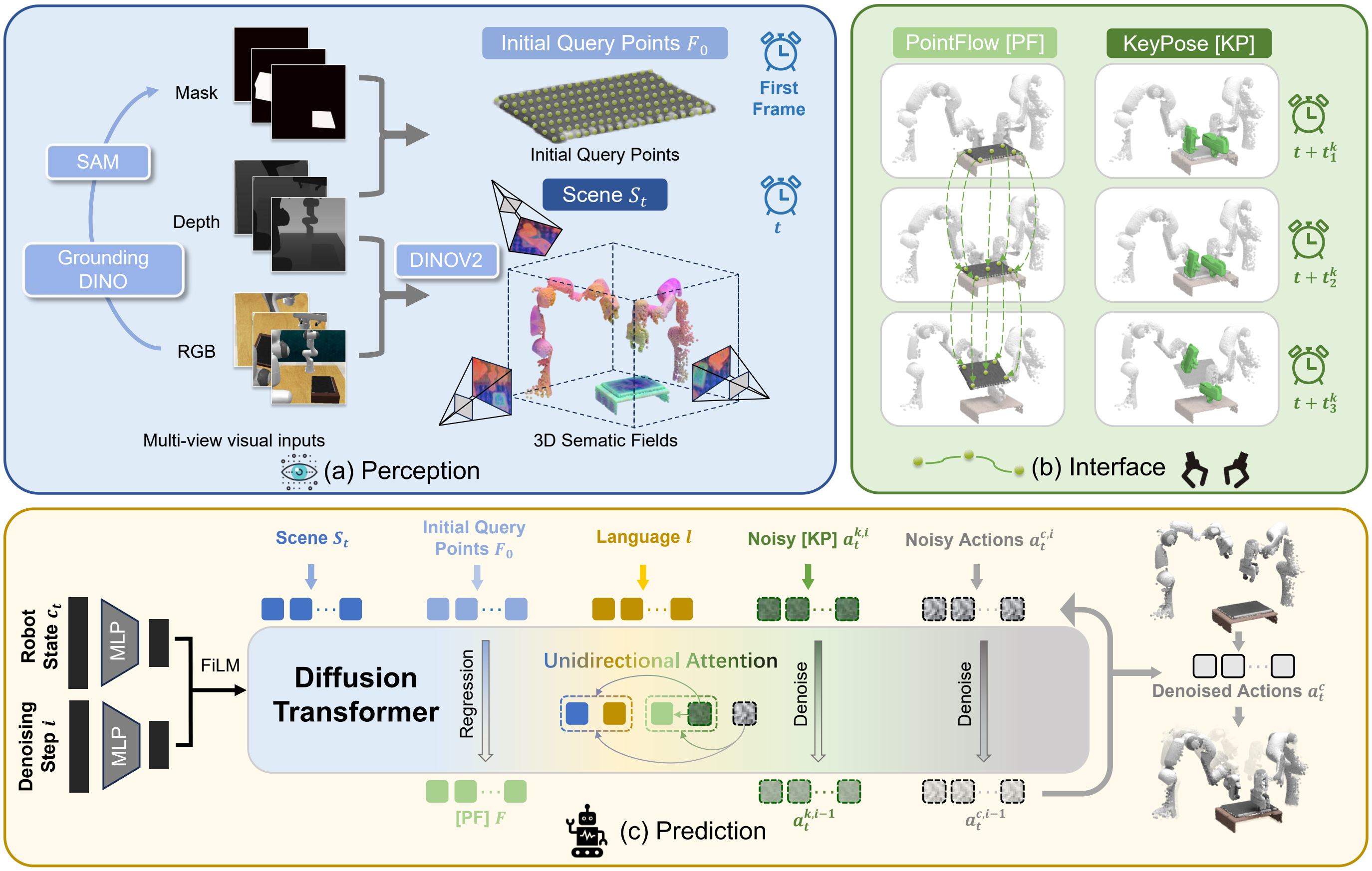

Fig. 2: **Overview of PPI.** (a) **Perception.** We first construct a 3D semantic neural field S_t and sample initial query points F_0 for pointflow prediction. (b) **Interface.** Next, we define two intermediate interfaces: target gripper poses a_t^k and object pointflow F. (c) **Prediction.** Finally, a diffusion transformer incorporates robot proprio tokens c_t, scene tokens S_t, language tokens l, pointflow query tokens F_0 and action tokens a_t^k and a_t^c with gaussian noise. Using a carefully designed unidirectional attention, the model progressively denoises action predictions conditioned on the interfaces.

the robot states and point queries are projected into the latent space through a three-layer MLP.

Diffusion Transformer. The backbone of the prediction module builds upon a diffusion transformer. At the time step t and denoising step i, let $a_t^{k,i}$ and $a_t^{c,i}$ be the keyframe action a_t^k and continuous action a_t^c with noise. The transformer incoporates the scene tokens S_t, language tokens l, query points tokens F_0 and noised action tokens $a_t^{k,i}$ and $a_t^{c,i}$. Outputs are supervised by gaussian noise ϵ_k^i, ϵ_c^i via DDPM [10] training and ground truth pointflow F via direct regression.

Notably, we design a unidirectional attention that leverages the interfaces to bridge the gap between input and output modalities. As shown in Figure 2(c), all pointflow and action tokens attend to the scene and language tokens, integrating spatial and semantic knowledge. Moreover, the noised keyframe action token $a_t^{k,i}$ attends to the pointflow token, aiming to extract additional object-level features. The final continuous action token $a_t^{c,i}$ attends to all the previous tokens, not only distilling regular scene-level features but also fully utilizing the local and detailed features contained in the interfaces.

We apply relative attention, as introduced in previous work [13], between point flow tokens F, keyframe action tokens $a_t^{k,i}$, continuous action tokens $a_t^{c,i}$, and scene point cloud tokens S_t, enabling the encoding of relative 3D positional information in the attention layers. This relative attention relies on the relative 3D positions of features and is implemented using rotary positional embeddings [29]. For language instructions l, we use regular cross and self-attention. The robot's proprioception c_t and denoising timesteps i affect the attention through Feature-wise Linear Modulation(FiLM) [21].

E. Implementation Details

Training. For the PPI network ϵ_θ, the continuous action loss $\mathcal{L}_c$, the keyframe action loss $\mathcal{L}_k$, and the pointflow prediction loss $\mathcal{L}_F$ are computed via L_1 loss.

$$\mathcal{L}_c = ||\epsilon_\theta(S_t, F_0, \ell, c_t, a_t^{c,i}, i) - \epsilon_c^i||$$

$$\mathcal{L}_k = ||\epsilon_\theta(S_t, F_0, \ell, c_t, a_t^{k,i}, i) - \epsilon_k^i||$$

$$\mathcal{L}_F = ||\epsilon_\theta(S_t, F_0, \ell, c_t, i) - F||$$

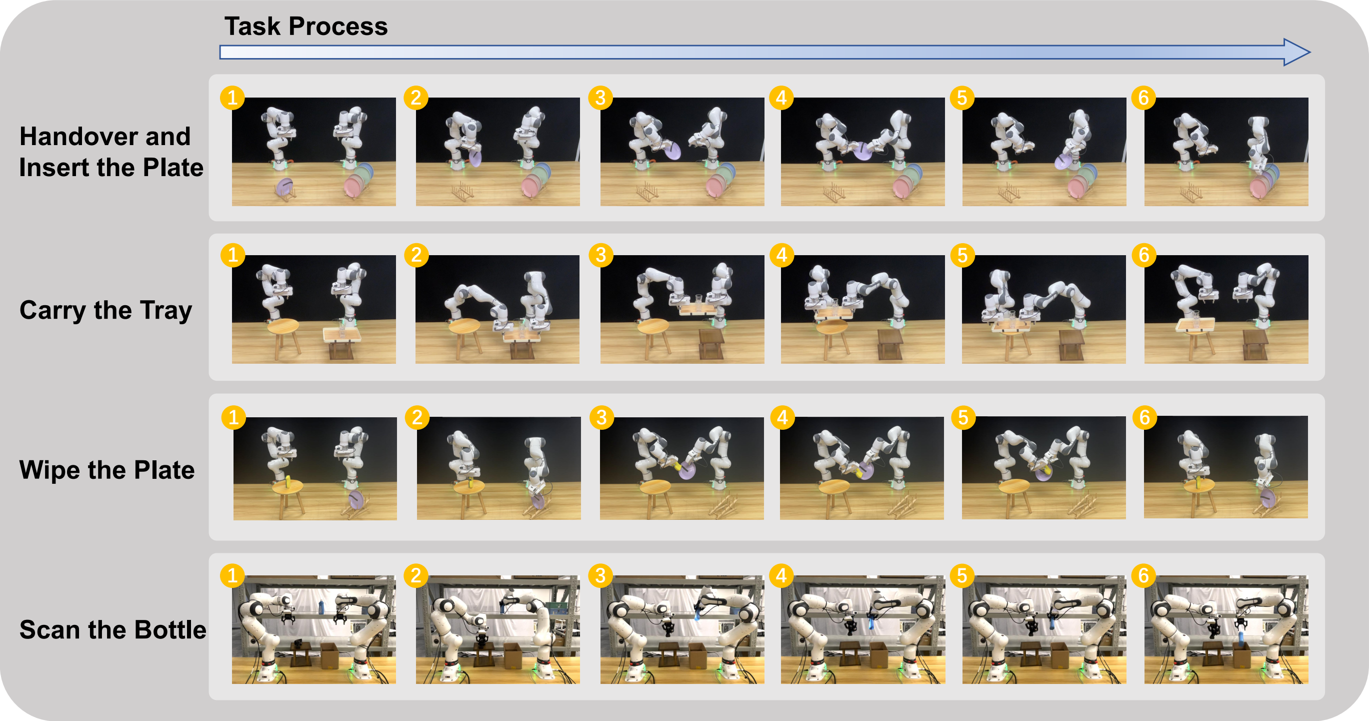

Fig. 3: Task process visualizations in four real-world tasks.

The overall training loss is:

$$\mathcal{L}_\theta = w_1\mathcal{L}_c + w_2\mathcal{L}_k + w_3\mathcal{L}_F$$

where w_1, w_2, and w_3 are hyperparameters set to 0.05, 0.05, and 1, respectively. We use DDPM with 1000 training timesteps for noise scheduling in all experiments. We train 500 epochs for tasks from RLBench2 benchmark and 5000 epochs in real-world tasks, with a batch size of 128 and learning rate of 1e-4 with a cosine decay learning rate scheduler. For tasks involving 100 episodes, each with 150-250 timesteps, we train the model using eight A100 GPUs for approximately 20 hours, and use the checkpoint with the lowest average validation loss for evaluation.

Inference. We begin by sampling 200 points on objects as query points at the first timestep. Then at each timestep t, we draw random initial continuous action samples a_t^c and keyframe action samples a_t^k from a Gaussian distribution and denoise 1000 steps with DDPM in simulation tasks and 20 steps with DDIM[28] for real-world tasks. In practice, we predict 50 continuous actions and 4 keyframe actions and pointflow.

IV. Real-world Experiments

We carefully design four real-world tasks with high localization demands and motion constraints to evaluate PPI 's capabilities in: (1) Effectiveness in long-horizon tasks. (2) Robustness under high-intensity disturbances in objects and environments.

A. *Real-world Experiments Setup*

Benchmark. We evaluate our method on two Franka Research 3 robots across four tasks in two distinct scenarios (Figure 4). Each scene is equipped with two Eye-on-Hand and one Eye-on-Base RealSense D435i cameras. In the first desktop environment, we test three long-horizon tasks requiring high localization accuracy and curved motion execution. In the second shelf scenario, we employ Robotiq-2f-85 grippers as end-effectors for the final task. Below, we briefly outline the process for each of the four tasks (Figure 3), with further details provided in the Appendix.

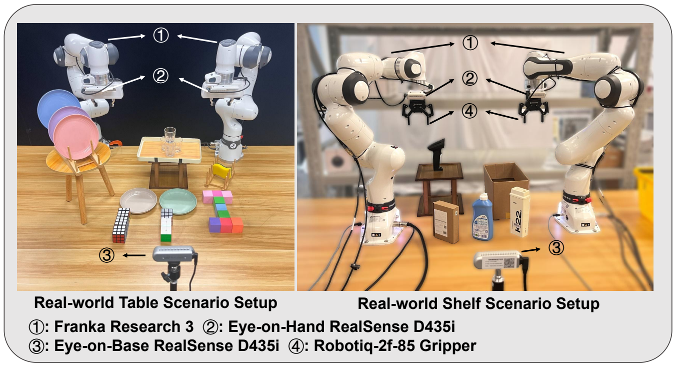
①
②
③
①
②
④
③
Real-world Table Scenario Setup
Real-world Shelf Scenario Setup
①: Franka Research 3 ②: Eye-on-Hand RealSense D435i
③: Eye-on-Base RealSense D435i ④: Robotiq-2f-85 Gripper

Fig. 4: Two real-world setups.

1) **Carry the Tray.** Both arms must collaboratively lift a tray stacked with cups or other objects from a lower platform and steadily transfer it to a small table. This task evaluates the ability to precisely locate the tray's

TABLE I: **Real-world main results.** We evaluate all the methods with 10 (settings) × 3 (repeated trials) rollouts per task. Our method achieves better performances among all tasks than baselines. The best results are bolded.

Method	SR (%) ↑ /Loc-SR (%) ↑ / Normalized Score ↑				
	Avg. Success ↑	Carry the Tray	Handover and Insert the Plate	Wipe the Plate	Scan the Bottle
ACT	15.0 / 50.0 / 4.7	40.0 / 40.0 / 5.5	10.0 / 70.0 / 5.0	0.00 / 40.0 / 3.8	10.0 / 50.0 / 4.3
DP3	35.0 / 57.5 / 5.5	50.0 / 50.0 / 6.0	20.0 / 80.0 / 4.7	40.0 / 60.0 / 6.5	30.0 / 40.0 / 4.8
3D Diffuser Actor	5.00 / 65.0 / 4.6	0.00 / 70.0 / 4.5	0.00 / 100 / 6.0	0.00 / 50.0 / 3.5	20.0 / 40.0 / 4.8
Ours	**62.5 / 92.5 / 8.2**	**50.0 / 100 / 7.8**	**40.0 / 100 / 7.7**	**70.0 / 80.0 / 8.3**	**90.0 / 90.0 / 9.3**

centerline and maintain stable, coordinated movements throughout the process.

2) **Handover and Insert the Plate.** The right arm picks up the plate and hands it over to the left arm, which then inserts it into an available slot in the rack. This task tests temporal coordination during handover and precise spatial perception.
3) **Wipe the Plate.** Each arm picks up the sponge and the plate, respectively, and uses the sponge to wipe the plate. After wiping, both objects are returned to their original positions. This task evaluates the ability to perform curved motions in interactive tasks.
4) **Scan the Bottle.** The right arm retrieves the bottle from the shelf, while the left arm picks up the scanner from the table to scan the bottle's barcode. After scanning, the bottle is placed into a box. This task assesses 6-DoF picking in a more constrained spatial environment and the coordination between both arms.

Expert demonstrations. We constructed two isomorphic teleoperation devices GELLO [42] for the Franka Research 3 to collect expert demonstrations. We collected 50 demonstrations for the task "Carry the Tray" and 20 for other tasks. The limited number of training samples is intended to evaluate whether the policies achieve excellent spatial localization and perception of objects with minimal data.

Baselines. We implement an Action Chunking Transformer (ACT) [48] that predicts target joint positions from 2D RGB inputs. We also adapt DP3 [46] into a bimanual framework, which is a 3D point-cloud-based continuous control policy. Additionally, we reproduce the 3D Diffuser Actor [13], a keyframe-based diffusion policy utilizing 3D semantic fields.

Metrics. Each method is evaluated across 10 settings, with 3 trials per setting. Due to the long-horizon and complex nature of the tasks, we have established three key metrics: Success Rate (SR), Localization Success Rate (Loc-SR), and Normalized Score. The Success Rate (SR) is only assigned a value of 100% upon the successful completion of the entire task. Loc-SR will be recorded 100% if the robots perform well to find the object contact positions. Additionally, we divide each tasks into 3 or 4 intermediate stages, and the Score is progressively accumulated through the completion of individual intermediate stages. Given that the number of steps varies across these tasks, the score will be normalized to a scale of 10. The scoring criteria and other details are available in the Appendix.

B. Real-world Main Results

As shown in Table I, PPI outperforms all baselines across tasks. Compared to state-of-the-art methods, it achieves a 27.5% increase in average success rate (SR), a 27.5% improvement in localization success rate (Loc-SR), and a 2.7 point gain in Normalized Score. Compared to ACT, which relies on RGB inputs, PPI demonstrates superior Loc-SR and overall scores. As a single-frame observation algorithm, ACT struggles with tasks requiring repetitive trajectories (e.g.,"Wipe the Plate"), whereas PPI integrates multistep proprioception, effectively leveraging historical information. Compared to DP3, a point-cloud-based method prone to overfitting seen trajectories and susceptible to noise [35], PPI exhibits stronger localization ability and robustness to noised pointcloud. While 3D Diffuser Actor, a keyframe-based policy using semantic neural fields and heuristic action execution, performs well in Loc-SR, it fails to account for collision constraints, leading to a lower SR. In contrast, PPI not only inherits the perception advantages of keyframe prediction but also effectively respects path constraints inherent in training demonstrations.

C. Real-world Generalization Tests

As shown in Table II-V and Figure 5-8, We introduce three types of generalizations to evaluate the generalizability and robustness: unseen objects to manipulated, different lighting backgrounds and interference from other objects. In each task, we select the setting where each method performed the best for generalization testing, and then we record the success rate, localization success rate, and normalized score. Each different generalization scenario has 3 trials. Details results are available in the Appendix.

TABLE II: Evaluation under object interference in Carry the Tray.

Method	Success / Localization / Normalized Score		
	Normal Setting	Rubik's Cubes	Colorful Cubes
ACT	✓/ ✓/ 10	✓/ ✓/ 10	✓/ ✓/ 10
DP3	✓/ ✓/ 10	✗/ ✗/ 2.5	✗/ ✓/ 5.0
3D Diffuser Actor	✗/ ✓/ 7.5	✗/ ✓/ 5.0	✗/ ✗/ 2.5
Ours	✓/ ✓/ 10	✓/ ✓/ 10	✓/ ✓/ 10

In **Carry the Tray**, we replace the cups with Robik's Cubes and colorful cubes to test robustness against **object interference**. These objects differ in size and color, which not only affect RGB inputs but also alter depth, influencing 3D-based policies. Thanks to the two interfaces, particularly

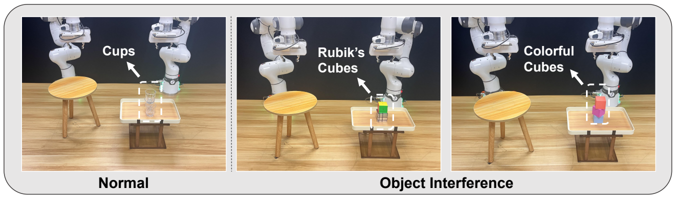

Fig. 5: Visualization under object interference in Carry the Tray.

the pointflow, PPI focuses on key object-related regions, such as the tray, without being distracted by other objects that could affect localization. Interestingly, two 3D-based baselines perform poorly, while the 2D-based ACT achieves better generalization. This suggests that although objects in this task have minimal impact at the pixel level in 2D images, they have a greater effect on the 3D scene representation.

TABLE III: Evaluation under different lighting backgrounds in Handover and Insert the Plate.

Method	Success / Localization / Normalized Score		
	Normal Setting	Dark Environment	Flickering Lighting Environment
ACT	✓/ ✓/ 10	✗/ ✗/ 0.0	✗/ ✗/ 0.0
DP3	✓/ ✓/ 10	✗/ ✓/ 6.7	✓/ ✓/ 10
3D Diffuser Actor	✗/ ✓/ 6.7	✗/ ✗/ 0.0	✗/ ✗/ 0.0
Ours	✓/ ✓/ 10	✓/ ✓/ 10	✓/ ✓/ 10

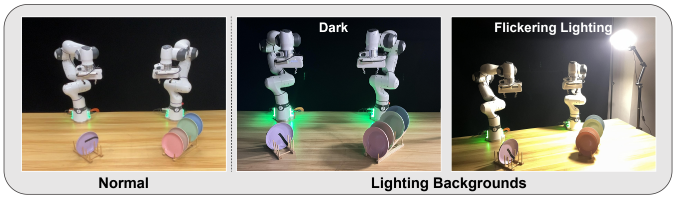

Fig. 6: Visualization under different lighting backgrounds in Handover and Insert the Plate.

In **Handover and Insert the Plate**, we evaluate PPI 's adaptability to varying **lighting conditions**. In dark and flickering lighting environments, RGB-based information is severely affected. However, despite using color information as inputs, PPI 's performance remains stable. This robustness probably stems from its reliance on pointflow and keypose, both derived mostly from the integration of semantic and positional features rather than pure color information. Meanwhile, DP3 maintains some success, likely due to its use of colorless point clouds.

TABLE IV: Evaluation under object interference in Wipe the Plate.

Method	Success / Localization / Normalized Score		
	Normal Setting	Colorful Cubes	Multi-plate
ACT	✗/ ✓/ 7.5	✗/ ✗/ 2.5	✗/ ✗/ 2.5
DP3	✓/ ✓/ 10	✗/ ✗/ 0.0	✗/ ✗/ 0.0
3D Diffuser Actor	✗/ ✓/ 5.0	✗/ ✓/ 5.0	✗/ ✓/ 5.0
Ours	✓/ ✓/ 10	✓/ ✓/ 10	✗/ ✓/ 7.5

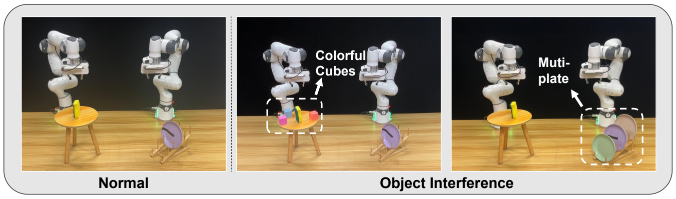

Fig. 7: Visualization under object interference in Wipe the Plate.

In **Wipe the Plate**, we introduce **object interference** separately for the objects manipulated by each arm, leading to partial occlusions and various visual distractions. Despite this, only PPI maintains robust localization.

In **Scan the Bottle**, we evaluate our method's generalization to **unseen objects**. Notably, PPI enables zero-shot manipulation of new objects by adjusting GroundingDino's [17] prompt to obtain novel initial query points F_0 for pointflow prediction. This is driven by PPI 's learned conditional distribution $p(F|F_0)$, which prioritizes object motion changes over absolute global coordinates $p(F)$. As a result, even when encountering novel rigid objects, PPI effectively estimates a rough object's relative transformations, enhancing task completion.

TABLE V: Evaluation under unseen objects in Scan the Bottle.

Method	Success / Localization / Normalized Score		
	Normal Setting	Box	Yogurt bottle
ACT	✓/ ✓/ 10	✗/ ✗/ 2.5	✗/ ✗/ 2.5
DP3	✓/ ✓/ 10	✗/ ✗/ 2.5	✗/ ✗/ 2.5
3D Diffuser Actor	✓/ ✓/ 10	✗/ ✗/ 2.5	✗/ ✗/ 2.5
Ours	✓/ ✓/ 10	✗/ ✓/ 7.5	✗/ ✓/ 7.5

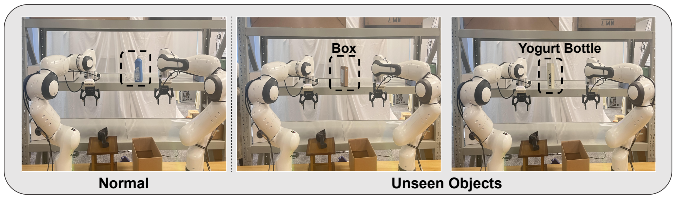

Fig. 8: Visualization under unseen objects in Scan the Bottle.

V. Simulation Experiments

We evaluate our approach on seven tasks using the bimanual simulation benchmark RLBench2 [9], aiming to address the following questions: (1) How well does PPI perform on complex bimanual tasks? (2) Are the proposed target gripper poses and object pointflow interfaces effective? (3) How do these interfaces learn scene features to enhance guidance for continuous action prediction?

A. Simulation Experiments Setup

Benchmark. RLBench2 [9] is a bimanual manipulation benchmark built on CoppeliaSim, encompassing tasks with different levels of coupling, coordination, language instructions, and manipulation skills. We select seven representative

TABLE VI: **Quantitative results on RLBench2.** For each task, we present the average performance of three checkpoints averaged over 100 rollouts. The metric "Avg. Success" measures the average success rate across seven tasks. PPI outperforms baselines with higher Avg. Success and better results on most tasks. The best results are bolded.

Method	Avg. Success ↑	Box	Ball	Drawer	Laptop	Dustpan	Tray	Handover Easy
ACT	15.4	67.0 ±7.0	38.3 ±10.0	1.7 ±2.1	0.0±0.0	0.0 ±0.0	1.3 ±1.5	0.0 ±0.0
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
3D Diffuser Actor	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
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
Ours	**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

TABLE VII: **Ablation studies.** For all ablated models, we report best performance, while for PPI, we additionally present the average performance across three checkpoints. Overall, integrating both keypose and pointflow achieves the highest performance.

Method	Avg. Success ↑	Box	Ball	Drawer	Laptop	Dustpan	Tray	Handover Easy
Continuous	47.6	84	24	41	0	98	82	4
Keyframe	49.0	84	**94**	36	0	92	12	25
Continuous on Keypose	53.6	71	81	29	1	99	86	8
Continuous on Pointflow	74.3	92	77	**84**	29	99	89	50
Ours (Best Ckpt)	**82.6**	**98**	91	**84**	**47**	**100**	**93**	**65**
Ours (Averaged)	80.8	96.7	89.3	79.7	46.3	98.7	92.0	62.7

and challenging tasks and regenerate the training data. The official dataset suffers from significant misalignment between training and evaluation, such as robot shadows present in the training set but absent during evaluation. Additionally, it lacks meta information for acquiring object pointflow.

Baselines. We use the same three baselines as in the real-world experiments: DP3, ACT, and 3D Diffuser Actor. Additionally, we include PerAct^2 [9], a state-of-the-art method previously reported on RLBench2. It employs a Perceiver architecture to voxelize 3D spaces and predict keyframe actions.

Metrics. Each method is evaluated across 100 rollouts per task with varying initial states for each tasks. We report both per-task and average success rates, with all performances computed from three different checkpoints.

B. Simulation Main Results

As shown in Table VI, PPI achieves an average success rate of 80.8%, significantly outperforming the baselines. Compared to ACT (2D-based algorithm), PPI leverages 3D semantic neural fields, and provides superior spatial perception. While both methods utilize 3D point clouds, PPI outperforms DP3 by 54% in success rate, demonstrating that PPI's spatial information processing is more useful than DP3's approach of encoding the entire scene into a single token.

Moreover, ACT and DP3, as continuous-action-based policies, underperforms keyframe-based policies like 3D Diffuser Actor and PerAct^2 by a margin of at least 10%. This disparity arises from keyframe-based methods offering more effective perception capabilities rather than overfitting to trajectories. By using keypose as an interface, PPI integrates the spatial awareness of keyframe methods with semantic neural fields, achieving substantial improvements over continuous-action-based approaches.

Beyond that, keyframe-based policies like 3D Diffuser Actor and PerAct^2 fall behind our model in tasks with movement constraints, such as the lift tray task (which requires both hands to remain level) and the push box task (which involves continuously pushing). This is because our policy provides greater flexibility in managing such constraints, enabled by PPI 's supervision of continuous actions between keyframes. Additionally, in tasks demanding high spatial precision, such as object-picking tasks like Drawer and Handover Easy, our model significantly outperforms baselines. This advantage stems primarily from the use of object pointflow as an interface, which enhances localization accuracy.

C. Ablation Study

As shown in Table VII, we analyze the contributions of target gripper poses and object pointflow interfaces on RLBench2 by comparing different ablated models at their best performance.

We begin by evaluating the vanilla keyframe and continuous baselines, which predict only keypose or continuous actions, respectively. In tasks requiring precise positioning, such as Handover Easy and Lift Ball, the keyframe-based policy demonstrates superior localization. However, in tasks involving horizontal lifting (Lift Tray) or curved motion trajectories (Sweep to Dustpan), the continuous-based approach significantly outperforms the keyframe method. These results indicate that relying solely on either keyframe or continuous actions is insufficient for general manipulation tasks.

Next, we modify the continuous-based policy by incorporating target gripper keypose and object pointflow predictions

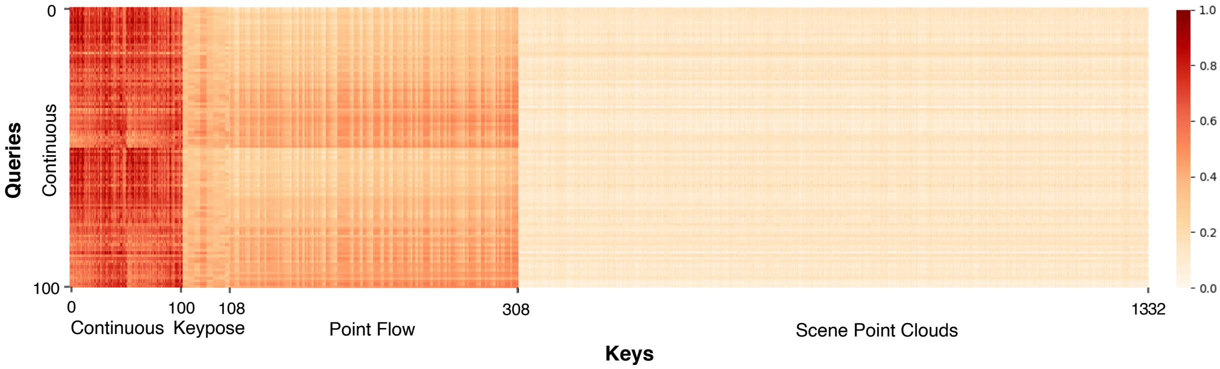

Fig. 9: **Heatmap of the attention weights of continuous action tokens.** The y-axis represents continuous action tokens as queries. The x-axis sequentially displays continuous action, keypose, pointflow, and scene tokens as keys.

(line 3 and 4 in Table VII). Conditioning on separate interfaces improves performance, likely due to the local spatial features they provide. Moreover, combining both interfaces yields further gains, highlighting the synergy between keypose and pointflow in enhancing performance on downstream tasks.

D. Visualization Analysis

In this section, we intuitively analyze how the proposed interfaces enhance localization accuracy and generalization to distractors and task-irrelevant backgrounds.

As shown in Figure 10, we visualize attention weights for our policy and the vanilla continuous-based policy in the ”Lift Tray” task on RLBench2. The left image reveals that in the continuous-based policy, action tokens fail to consistently focus on the tray—the key task-relevant object and disperse attention across both the tray and the robotic arm. This suggests why continuous-based policies often struggle with precise localization, as observed in previous experiments.

In contrast, as shown in the middle and right images, PPI's pointflow and keypose tokens strongly attend to the tray, with pointflow tokens specifically concentrating on its edges—where the gripper is poised to grasp. Since the initial query points are sampled from the tray rather than the cube atop it, the model learns to deprioritize the cube. This explains why in the ”Carry the Tray” task, replacing in-distribution cups with out-of-distribution Rubik's Cubes and colorful cubes does not impair PPI 's localization performance. By strategically selecting initial query points during training, the model learns to track the overall object's motion, improving generalization under disturbances.

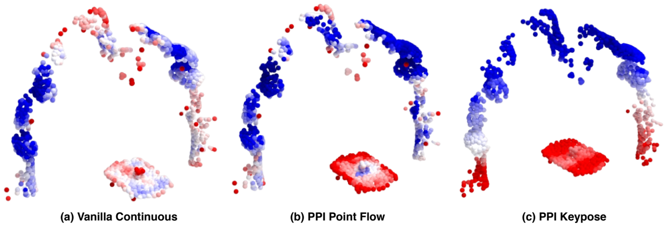

Fig. 10: **Visualization of the attention weights of interface tokens.** We use “Lift Tray” task on RLBench2 as an example. The **red** area corresponds to larger attention weight, while the **blue** area corresponds to smaller attention weight. **Left:** The attention weights of continuous tokens to the 1024 scene tokens in the vanilla continuous-action-based policy. **Middle:** The attention weights of point flow tokens to the scene tokens in PPI. **Right:** The attention weights of keypose tokens in PPI.

Further, we examine how these interfaces guide continuous action prediction. As shown in Figure 9, continuous action tokens exhibit significantly higher attention weights toward keypose and pointflow tokens than scene tokens, underscoring the critical role of interfaces in action prediction. By incorporating target gripper poses and object pointflow as interfaces, PPI not only maintains focus on task-relevant regions despite scene variations but also alleviates the learning burden on action tokens by distilling key information, rather than directly querying from the entire scene.

VI. Conclusion and Limitation

We introduce PPI, an end-to-end interface-based manipulation policy that leverages target gripper poses and object pointflow. PPI achieves state-of-the-art performance on the RLBench2 simulation benchmark and demonstrates strong effectiveness and robustness in real-world experiments.

Limitation. There remain two main limitations. First, the computational cost of visual foundation models and the diffusion process constrains efficiency. Future work will focus on accelerating diffusion sampling and adopting lightweight vision models. Second, cross-embodiment evaluation on different robotic platforms is essential to assess PPI's generalization across hardware.

References

[1] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *Arxiv*, 2024.

[2] Wen Bowen, Yang Wei, Kautz Jan, and Birchfield Stan. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *Arxiv*, 2022.

[4] 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. In *RSS*, 2023.

[5] Shivin Devgon, Jeffrey Ichnowski, Ashwin Balakrishna, Harry Zhang, and Ken Goldberg. Orienting novel 3d objects using self-supervised learning of rotation transforms. In *CASE*, 2020.

[6] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *Arxiv*, 2023.

[7] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. In *RSS*, 2022.

[8] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. In *TRO*, 2023.

[9] Markus Grotz, Mohit Shridhar, Tamim Asfour, and Dieter Fox. Peract2: A perceiver actor framework for bimanual manipulation tasks. In *Arxiv*, 2024.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[11] Stephen James and Andrew J. Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. In *IEEE Robotics and Automation Letters*, 2022.

[12] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024.

[13] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *CoRL*, 2024.

[14] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *CoRL*, 2024.

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Arxiv*, 2023.

[16] I Liu, Chun Arthur, Sicheng He, Daniel Seita, and Gaurav Sukhatme. Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation. In *CoRL*, 2024.

[17] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Arxiv*, 2023.

[18] 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 *Arxiv*, 2024.

[19] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. In *Arxiv*, 2023.

[20] Chuer Pan, Brian Okorn, Harry Zhang, Ben Eisner, and David Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. In *CoRL*, 2023.

[21] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[22] Carl Qi, Yilin Wu, Lifan Yu, Haoyue Liu, Bowen Jiang, Xingyu Lin, and David Held. Learning generalizable tool-use skills through trajectory generation. In *IROS*, 2024.

[23] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Arxiv*, 2017.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Arxiv*, 2021.

[25] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *Arxiv*, 2024.

[26] Daniel Seita, Yufei Wang, Sarthak J Shetty, Edward Yao Li, Zackory Erickson, and David Held. Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds. In *CoRL*, 2023.

[27] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *CoRL*, 2022.

[28] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Arxiv*, 2020.

[29] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced trans-

former with rotary position embedding. In *Arxiv*, 2023.

[30] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. In *IEEE Robotics & Automation Magazine*, 2012.

[31] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *ICRA*, 2023.

[32] Yang Tian, Jiyao Zhang, Zekai Yin, and Hao Dong. Robot structure prior guided temporal attention for camera-to-robot pose estimation from image sequence. In *CVPR*, 2023.

[33] Yang Tian, Jiyao Zhang, Guowei Huang, Bin Wang, Ping Wang, Jiangmiao Pang, and Hao Dong. Robokeygen: Robot pose and joint angles estimation via diffusion-based 3d keypoint generation. In *ICRA*, 2024.

[34] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *ICLR*, 2025.

[35] Chenxi Wang, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. Rise: 3d perception makes real-world robot imitation simple and effective. In *IROS*, 2024.

[36] Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, et al. Grutopia: Dream general robots in a city at scale. In *Arxiv*, 2024.

[37] Yixuan Wang, Guang Yin, Binghao Huang, Tarik Kelestemur, Jiuguang Wang, and Yunzhu Li. Gendp: 3d semantic fields for category-level generalizable diffusion policy. In *CoRL*, 2024.

[38] Yixuan Wang, Mingtong Zhang, Zhuoran Li, Tarik Kelestemur, Katherine Driggs-Campbell, Jiajun Wu, Li Fei-Fei, and Yunzhu Li. D^{3}fields: Dynamic 3d descriptor fields for zero-shot generalizable rearrangement. In *CoRL*, 2024.

[39] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 606–617, 2023.

[40] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *RSS*, 2024.

[41] Thomas Weng, Sujay Man Bajracharya, Yufei Wang, Khush Agrawal, and David Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *CoRL*, 2022.

[42] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *IROS*, 2024.

[43] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *CoRL*, 2024.

[44] Dongjie Yu, Hang Xu, Yizhou Chen, Yi Ren, and Jia Pan. Bikc: Keypose-conditioned consistency policy for bimanual robotic manipulation. In *Arxiv*, 2024.

[45] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. In *CoRL*, 2024.

[46] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. In *RSS*, 2024.

[47] Harry Zhang, Ben Eisner, and David Held. Flowbot++: Learning generalized articulated objects manipulation via articulation projection. In *CoRL*, 2023.

[48] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *RSS*, 2023.

[49] Yan Zhao, Ruihai Wu, Zhehuan Chen, Yourong Zhang, Qingnan Fan, Kaichun Mo, and Hao Dong. Dualafford: Learning collaborative visual affordance for dual-gripper manipulation. In *ICLR*, 2023.

Appendix

A. Foundation Models

PPI leverages foundation models in various aspects, either to generate ground-truth labels or serve as encoders. In this section, we will specifically describe how we utilize these foundation models, along with the exact versions and settings of the models we use.

3D Semantic Fields. To obtain semantic features from 2D images, we leverage the DINOv2 ViT-S/14 model, with a feature dimension of 384. In practice, we crop and downsample the scene point clouds to 6144 in simulations and 3072 in real-world experiments, resulting in 3D semantic fields with dimensions of 6144×387 and 3072×387, respectively, which will be further downsampled by PointNet++ in the subsequent processing. Here, the dimension of 387 includes both the semantic features and the xyz coordinates of the points.

Initial Point Sampling. In both simulation and real-world experiments, we sample 200 points on the object, which has been shown to effectively capture the object's motion. For simulation tasks, we typically select a single camera from which the target object is visible at the first timestep in each episode. We then feed a language prompt to the SwinB CogCoor version of the Grounding DINO model to obtain the bounding box for the target object from that camera. This bounding box is subsequently passed to the ViT-B SAM model, which generates a mask for the pixels corresponding to the target object.

Language. Text instructions are encoded using the CLIP ResNet50 text encoder and projected through a linear layer to generate the language token.

Object Pose Estimation. In simulation, we generate a dataset that includes the ground truth of 6D pose of the objects. During the real-world data collection process, we utilize BundleSDF[39] to obtain the meshes of unknown objects with a coarse 6D pose. With the meshes of objects, we employ the model-based method of FoundationPose[2] to enhance the precision of the objects' 6D pose. More specifically, we perform 6D pose estimation on the first frame of each demonstration, and for each subsequent frame, we use 6D pose tracking.

B. Architecture Details

Relative Attention. Our 3D denoising transformer takes four types of 3D tokens as input: point flow tokens F, keyframe action tokens $a_t^{k,i}$, continuous action tokens $a_t^{c,i}$, and scene point cloud tokens S_t. Each 3D token is represented by a latent embedding and a 3D position. Following the 3D Diffuser Actor [13], our model applies relative self-attention among all 3D tokens. We use a rotary positional embedding to encode relative positional information in the attention layers, with the attention weight between query q and key k given by:

$$e_{q,k} \propto x_q^T M(c_q - c_k) x_k$$

where x_q and x_k are the features of the query and key, and c_q and c_k are the corresponding 3D coordinates of the query and key tokens. M is a matrix function that depends only on the relative positions of the 3D coordinates of the query and key tokens. Relative attention allows us to effectively leverage the relative 3D spatial information between tokens within the transformer.

Robot State. For each gripper, the arm state consists of the end-effector position, quaternion, and gripper state, forming an 8-dimensional vector. The gripper state is either 0 (closed) or 1 (open). In our 3d denoising transformer, our method uses separate tokens for the left and right hand actions, helping the model focus on different regions. For predicting 50 continuous actions, we use 100 tokens, and for predicting 4 keyframe actions, we use 8 tokens. The robot state is tokenized with a three-layer MLP, similar to how we tokenize language and initial query points.

Network Architecture. Next, we detail the structure of our 3D denoising model as shown in Figure 11. At timestep t and denoising step i, let $a_t^{k,i}$ and $a_t^{c,i}$ be the noised keyframe action a_t^k and continuous action a_t^c, respectively. The transformer incorporates the scene tokens S_t, language tokens l, query points tokens F_0, and the noised action tokens $a_t^{k,i}$ and $a_t^{c,i}$.

All tokens in each self-attention block will first perform parallel attention with the language tokens l to capture the relevant language instructions. Next, we concatenate the query points tokens F_0 and scene tokens S_t, applying relative self-attention. The resulting tokens for F_0 pass through three MLP layers, where each token outputs a 12-dimensional vector representing the xyz coordinates of that point for the next 4 keyframes. Then, we perform relative self-attention on the noised keyframe action tokens $a_t^{k,i}$ and the pointflow tokens and the scene tokens from the previous layer. The output corresponding to the keypose token is passed through three MLP layers to map it to the action dimension, with supervision provided by the keypose noise $\epsilon_t^{k,i}$. Finally, we perform relative self-attention between the noised continuous action tokens $a_t^{c,i}$ and the detached vector from the previous layer. The output corresponding to the continuous token is also passed through three MLP layers to map it to the action dimension, with supervision provided by the continuous noise $\epsilon_t^{c,i}$.

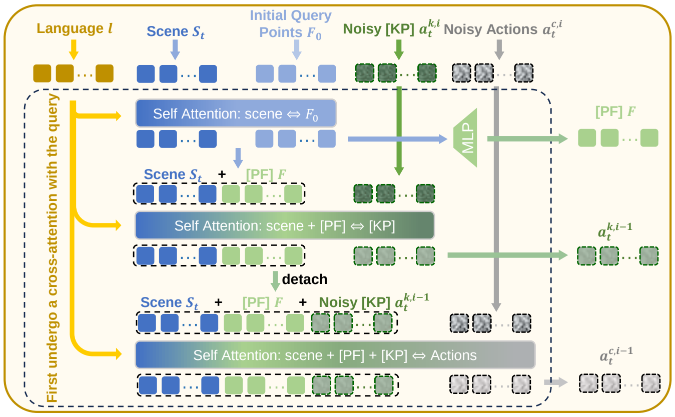
Language l
Scene S_t
Initial Query Points F_0
Noisy [KP] $a_t^{k,i}$
Noisy Actions $a_t^{c,i}$
First undergo a cross-attention with the query
Self Attention: scene ⇔ F_0
MLP
[PF] F
Scene S_t + [PF] F
Self Attention: scene + [PF] ⇔ [KP]
$a_t^{k,i-1}$
detach
Scene S_t + [PF] F + Noisy [KP] $a_t^{k,i-1}$
Self Attention: scene + [PF] + [KP] ⇔ Actions
$a_t^{c,i-1}$

Fig. 11: **Detailed architecture of 3D denoising model.**

C. Simple PPI

In the real-world, the original PPI has an average inference speed of 2.63 Hz. Since the predicted actions are dense and continuous, we excute the first few steps (e.g., 3 steps) of the 50 predicted actions during inference, which maintains performace while ensuring smooth motion. Additionally, we implemented Simple PPI to further reduce computation costs. Simple PPI achieves 10.52 Hz by reducing whole scene tokens (3072 → 512), downsampled scene tokens (512 → 64), pointflow tokens (200 → 50), DDIM denoising steps (20 → 5), and halving the number of transformer layers in real-world settings. As shown in Table VIII, this results in nearly 4× faster inference without much accuracy loss.

TABLE VIII: **Performance of Simple PPI.**

Method	SR (%) ↑ /Loc-SR (%) ↑ / Normalized Score ↑		
	Carry the Tray	Handover & Insert the Plate	Wipe the Plate
Simple PPI (10.52 Hz)	50.0 / 70.0 / 6.0	30.0 / 100 / 5.7	50.0 / 70.0 / 6.8
PPI (2.63 Hz)	**50.0 / 100 / 7.8**	**40.0 / 100 / 7.7**	**70.0 / 70.0 / 8.3**

D. More Tasks in Real-World

To demonstrate PPI's ability to handle high-precision tasks and manipulate deformable objects, we design two new real-world tasks. For high precision task **Press the Cap of Bottle**, one arm picks up a bottle and places it on a 3cm × 3cm cube, while the other precisely presses its cap (0.75cm radius). For deformable object task **Wear the Scarf**, arms are required to lift a scarf simultaneously and wrap it around a mannequin, where CoTracker is used to obtain pointflow ground truth. To showcase PPI's capability to align the scanner and bottle from different initial states and complete single-arm tasks, we design a single-arm experiment **Scan the Bottle with Single Arm** where the bottle starts in two directions: 1) facing the scanner and 2) at a 45-degree angle, requiring the scanner to rotate for a successful scan. The procedures for these three tasks are available on https://yuyinyang3y.github.io/PPI/. As shown in Table IX, our method outperforms the baseline (ACT) across all tasks.

TABLE IX: **Real-world results of three extra tasks.**

Method	SR (%) ↑ /Loc-SR (%) ↑ / Normalized Score ↑		
	Wear the Scarf	Press the Cap of Bottle	Scan the Bottle (Unimanual)
ACT	40.0 / 70.0 / 5.7	10.0 / 20.0 / 3.0	20.0 / 70.0 / 4.7
Ours	**70.0 / 80.0 / 7.7**	**40.0 / 80.0 / 6.7**	**80.0 / 100 / 9.3**

E. Baseline Implementation

In the simulation benchmark, we report the scores for DP3, ACT, PerAct^2, and 3D Diffuser Actor from their respective papers. For ACT and PerAct^2, we reproduce the results using the official code in RLBench2 and train each method for 100k steps, as suggusted in the original paper. The final scores are obtained by averaging the scores from epochs 99900, 99800, and 99700. For DP3, we directly double its action dimension and adapt it to the bimanual task setting. We use the default hyperparameter settings, prediction horizon, observation step, and other configurations from the paper, and train the model for 3000 epochs as suggested. We average the success rate of epoch 3000, 2900 and 2800 for each task. Similar to our own method, we double the number of action tokens for 3D Diffuser Actor, using separate actions to predict the left and right hand movements instead of directly doubling the action dimension. We select the checkpoint corresponding to the epoch when the eval location and rotation accuracy converged during the model training process. For simple tasks like ball lifting, we chose steps 10000, 11000, and 12000, while for more complex tasks such as handover easy, we select steps 16000, 18000, and 20000.

For real-world experiments, we implement the baselines in the same way as in simulations, but trained them with different durations due to the increased complexity of real-world tasks: ACT for 10000 epochs, DP3 for 8000 epochs (with its prediction horizon N_{act} adjusted to 10 for better performance), and 3D Diffuser Actor for 100000 steps.

F. Detailed Real-world Experiment Settings

As shown in Figure 12-15, the yellow area represents the variations during training, and the red area represents the settings during testing. We set 10 settings for each task and every method has 3 trials for each setting.

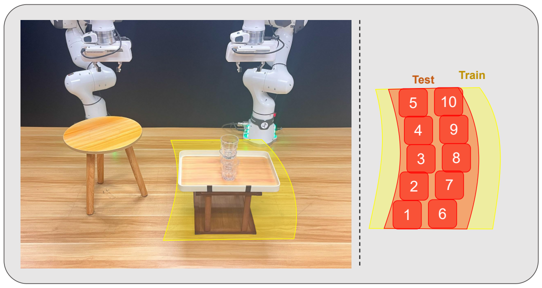

Fig. 12: **Detailed 10 settings of Carry the Tray.** The tray remains relatively stationary while the platform is moved to 10 different locations, serving as 10 distinct settings.

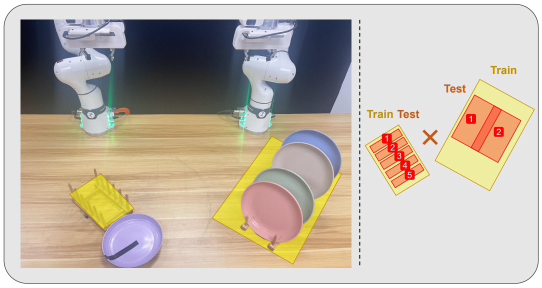

Fig. 13: **Detailed 10 settings of Handover and Insert the Plate.** On the left, the plate varies across 5 slots on the rack, while on the right, the overall setup shifts between 2 designated positions, combining to form 10 different settings.

TABLE X: **Generalization tests across all settings.**

Method	Unseen Objects			Lighting Backgrounds			Object Interference		
	Carry	Handover & Insert	Wipe	Carry	Handover & Insert	Wipe	Carry	Handover & Insert	Wipe
ACT	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗	✓/ ✓	✗/ ✗	✗/ ✗
DP3	✓/ ✗	✗/ ✓	✓/ ✗	✓/ ✓	✗/ ✓	✓/ ✓	✗/ ✗	✗/ ✗	✗/ ✗
3D Diffuser Actor	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗	✗/ ✗
Ours	✓/ ✓	✗/ ✓	✓/ ✗	✓/ ✗	✓/ ✓	✓/ ✓	✓/ ✓	✓/ ✗	✓/ ✗

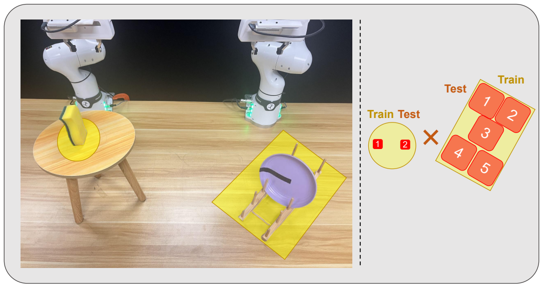

Fig. 14: **Detailed 10 settings of Wipe the Plate.** On the left, the sponge changes positions between the 2 sides of the yellow training area, and on the right, the plate moves across the 5 red areas shown, creating a total of 10 settings.

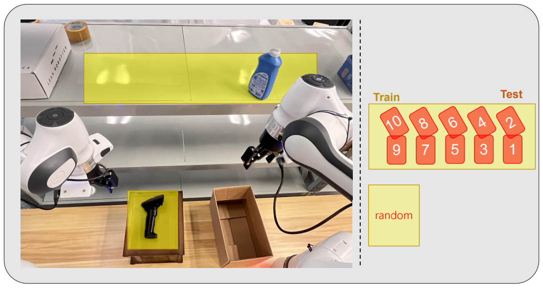

Fig. 15: **Detailed 10 settings of Scan the Bottle.** Among the 10 settings, the bottle changes its position within the 10 red areas on the shelf, with variations in both position and rotation, while the scanner is placed randomly in minor adjustments on the platform.

G. Detailed Real-world Results

We define three metrics: Success Rate (SR), Localization Success Rate (Loc-SR), and Normalized Score. And each task has several intermediate stages.

Specifically, **Carry the Tray** consists of four steps: (1) Pick the Tray, (2) Lift the Tray, (3) Place the Tray, (4) Return. The Loc_SR is recorded as 100% only if the first two steps are achieved.

Handover and Insert the Plate involves three steps: (1) Pick the Plate, (2) Handover, (3) Insert the Plate. The Loc_SR is marked as 100% only if the robot successfully picks up the plate.

Wipe the Plate consists of the following four steps: (1) Pick the Sponge, (2) Pick the Plate, (3) Wipe, (4) Place. The Loc_SR is recorded as 100% only if both the sponge and the plate are successfully picked up.

For **Scan the Bottle**, the sequence is as follows: (1) Pick the Bottle, (2) Pick the Scanner, (3) Scan, (4) Place Bottle in Box. Likewise, the Loc-SR is marked as 100% only when both the bottle and the scanner are successfully picked up.

In **Wear the Scarf**, the arms need to carry out the following steps: (1) Pick the Scarf, (2) Lift the Scarf, (3) Wear the Scarf. If the scarf is picked up successfully, the Loc_SR will be recorded as 100%.

Press the Cap of Bottle also includes three motions: (1) Pick the Bottle, (2) Place the Bottle, (3) Press the Cap. The Loc_SR will be 100% if the robot successfully picks up the bottle and places it on the cube.

For the single-arm experiment **Scan the Bottle with Single Arm**, the procedure comprises three stages: (1) Pick the Scanner, (2) Scan, (3) Place the Scanner. The Loc_SR is marked as 100% if the first stage is completed.

Additionally, the score increases by 1 point upon the completion of each stage. To facilitate comparison, the total score is then normalized to a scale of 10 points, resulting in the Normalized Score. The detailed results from the real-world experiments are available in Table XI-XVII

H. Detailed Real-world Generalization Results

In the real-world generalization experiments, there are two settings for each generalization scenario (unseen objects, lighting background changes and object interference), with each setting undergoing 3 trials. The scoring criteria (SR, Loc_SR and Score) and detailed real-world results remain consistent across these trials.

Additionally, we add all generalization tests to the first three tasks, all outperforming the baselines. There are two different settings in each of these generalization tests. **Unseen objects**: Vary tray/plate shapes/colors (*Carry/Handover*); replace sponge with rag and round plate with rectangular one (*Wipe*). **Lighting**: Dark and flickering lighting environments. **Object interference**: Introduce different distractor objects. Results are summarized in Table X.

The detailed results from the real-world generalization experiments are available in Table XVIII-XXI.

TABLE XI: **Detailed results on Carry the Tray task.**

Method	Case Index	Pick the Tray	Lift the Tray	Place the Tray	Return	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	1	✓	✓	✓	✗	0.0 / 1.0 / 3
	2	✓	✓	✓	✓	1.0 / 1.0 / 4
	3	✓	✓	✗	✗	0.0 / 1.0 / 2
	4	✓	✓	✓	✓	1.0 / 1.0 / 4
	5	✓	✓	✗	✗	0.0 / 1.0 / 2
	6	✓	✓	✓	✓	1.0 / 1.0 / 4
	7	✓	✓	✗	✗	0.0 / 1.0 / 2
	8	✓	✓	✓	✓	1.0 / 1.0 / 4
	9	✓	✓	✓	✓	1.0 / 1.0 / 4
	10	✓	✓	✗	✗	0.0 / 1.0 / 2
Simple PPI	1	✓	✓	✓	✓	1.0 / 1.0 / 4
	2	✓	✓	✓	✓	1.0 / 1.0 / 4
	3	✓	✓	✓	✓	1.0 / 1.0 / 4
	4	✗	✗	✗	✗	0.0 / 0.0 / 0
	5	✗	✗	✗	✗	0.0 / 0.0 / 0
	6	✓	✓	✗	✗	0.0 / 1.0 / 2
	7	✓	✓	✗	✗	0.0 / 1.0 / 2
	8	✓	✓	✓	✓	1.0 / 1.0 / 4
	9	✓	✓	✓	✓	1.0 / 1.0 / 4
	10	✗	✗	✗	✗	0.0 / 0.0 / 0
ACT	1	✓	✗	✗	✗	0.0 / 0.0 / 1
	2	✓	✗	✗	✗	0.0 / 0.0 / 1
	3	✓	✗	✗	✗	0.0 / 0.0 / 1
	4	✓	✓	✓	✓	1.0 / 1.0 / 4
	5	✓	✗	✗	✗	0.0 / 0.0 / 1
	6	✓	✗	✗	✗	0.0 / 0.0 / 1
	7	✓	✓	✓	✓	1.0 / 1.0 / 4
	8	✓	✓	✓	✓	1.0 / 1.0 / 4
	9	✓	✓	✓	✓	1.0 / 1.0 / 4
	10	✓	✗	✗	✗	0.0 / 0.0 / 1
DP3	1	✓	✗	✗	✗	0.0 / 0.0 / 1
	2	✓	✓	✓	✓	1.0 / 1.0 / 4
	3	✓	✓	✓	✓	1.0 / 1.0 / 4
	4	✓	✓	✓	✓	1.0 / 1.0 / 4
	5	✓	✓	✓	✓	1.0 / 1.0 / 4
	6	✓	✗	✗	✗	0.0 / 0.0 / 1
	7	✓	✗	✗	✗	0.0 / 0.0 / 1
	8	✓	✓	✓	✓	1.0 / 1.0 / 4
	9	✓	✗	✗	✗	0.0 / 0.0 / 1
	10	✗	✗	✗	✗	0.0 / 0.0 / 0
3D Diffuser Actor	1	✓	✓	✗	✗	0.0 / 1.0 / 2
	2	✓	✓	✗	✗	0.0 / 1.0 / 2
	3	✓	✓	✗	✗	0.0 / 1.0 / 2
	4	✓	✓	✗	✗	0.0 / 1.0 / 2
	5	✓	✓	✓	✗	0.0 / 1.0 / 3
	6	✓	✗	✗	✗	0.0 / 0.0 / 1
	7	✓	✗	✗	✗	0.0 / 0.0 / 1
	8	✓	✗	✗	✗	0.0 / 0.0 / 1
	9	✓	✓	✗	✗	0.0 / 1.0 / 2
	10	✓	✓	✗	✗	0.0 / 1.0 / 2

TABLE XII: **Detailed results on Handover and Insert the Plate task.**

Method	Case Index	Pick the Plate	Handover	Insert the Plate	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	1	✓	✓	✗	0.0 / 1.0 / 2
	2	✓	✓	✓	1.0 / 1.0 / 3
	3	✓	✓	✗	0.0 / 1.0 / 2
	4	✓	✓	✗	0.0 / 1.0 / 2
	5	✓	✓	✗	0.0 / 1.0 / 2
	6	✓	✗	✗	0.0 / 1.0 / 1
	7	✓	✓	✓	1.0 / 1.0 / 3
	8	✓	✓	✓	1.0 / 1.0 / 3
	9	✓	✓	✗	0.0 / 1.0 / 2
	10	✓	✓	✓	1.0 / 1.0 / 3
Simple PPI	1	✓	✗	✗	0.0 / 1.0 / 1
	2	✓	✓	✓	1.0 / 1.0 / 3
	3	✓	✗	✗	0.0 / 1.0 / 1
	4	✓	✗	✗	0.0 / 1.0 / 1
	5	✓	✓	✗	0.0 / 1.0 / 2
	6	✓	✗	✗	0.0 / 1.0 / 1
	7	✓	✓	✓	1.0 / 1.0 / 3
	8	✓	✓	✓	1.0 / 1.0 / 3
	9	✓	✗	✗	0.0 / 1.0 / 1
	10	✓	✗	✗	0.0 / 1.0 / 1
ACT	1	✗	✗	✗	0.0 / 0.0 / 0
	2	✗	✗	✗	0.0 / 0.0 / 0
	3	✓	✓	✗	0.0 / 1.0 / 2
	4	✓	✓	✓	1.0 / 1.0 / 3
	5	✓	✓	✗	0.0 / 1.0 / 2
	6	✓	✓	✗	0.0 / 1.0 / 2
	7	✓	✓	✗	0.0 / 1.0 / 2
	8	✓	✓	✗	0.0 / 1.0 / 2
	9	✓	✓	✗	0.0 / 1.0 / 2
	10	✗	✗	✗	0.0 / 0.0 / 0
DP3	1	✓	✗	✗	0.0 / 1.0 / 1
	2	✓	✓	✗	0.0 / 1.0 / 2
	3	✓	✓	✓	1.0 / 1.0 / 3
	4	✓	✗	✗	0.0 / 1.0 / 1
	5	✓	✓	✓	1.0 / 1.0 / 3
	6	✓	✗	✗	0.0 / 1.0 / 1
	7	✓	✓	✗	0.0 / 1.0 / 2
	8	✓	✗	✗	0.0 / 1.0 / 1
	9	✗	✗	✗	0.0 / 0.0 / 0
	10	✗	✗	✗	0.0 / 0.0 / 0
3D Diffuser Actor	1	✓	✗	✗	0.0 / 1.0 / 1
	2	✓	✓	✗	0.0 / 1.0 / 2
	3	✓	✗	✗	0.0 / 1.0 / 1
	4	✓	✓	✗	0.0 / 1.0 / 2
	5	✓	✓	✗	0.0 / 1.0 / 2
	6	✓	✓	✗	0.0 / 1.0 / 2
	7	✓	✓	✗	0.0 / 1.0 / 2
	8	✓	✓	✗	0.0 / 1.0 / 2
	9	✓	✓	✗	0.0 / 1.0 / 2
	10	✓	✓	✗	0.0 / 1.0 / 2

TABLE XIII: **Detailed results on Wipe the Plate task.**

Method	Case Index	Pick the Sponge	Pick the Plate	Wipe	Place	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	1	✓	✓	✓	✗	0.0 / 1.0 / 3
	2	✓	✓	✓	✓	1.0 / 1.0 / 4
	3	✓	✗	✗	✗	0.0 / 0.0 / 1
	4	✓	✓	✓	✓	1.0 / 1.0 / 4
	5	✓	✓	✓	✓	1.0 / 1.0 / 4
	6	✓	✓	✓	✓	1.0 / 1.0 / 4
	7	✓	✗	✗	✗	0.0 / 0.0 / 1
	8	✓	✓	✓	✓	1.0 / 1.0 / 4
	9	✓	✓	✓	✓	1.0 / 1.0 / 4
	10	✓	✓	✓	✓	1.0 / 1.0 / 4
Simple PPI	1	✓	✓	✓	✓	1.0 / 1.0 / 4
	2	✓	✗	✗	✗	0.0 / 0.0 / 1
	3	✓	✓	✗	✗	0.0 / 1.0 / 2
	4	✓	✓	✓	✓	1.0 / 1.0 / 4
	5	✓	✓	✓	✓	1.0 / 1.0 / 4
	6	✓	✓	✓	✓	1.0 / 1.0 / 4
	7	✓	✗	✗	✗	0.0 / 0.0 / 1
	8	✓	✗	✗	✗	0.0 / 0.0 / 1
	9	✓	✓	✗	✗	0.0 / 1.0 / 2
	10	✓	✓	✓	✓	1.0 / 1.0 / 4
ACT	1	✓	✓	✗	✓	0.0 / 1.0 / 3
	2	✗	✓	✗	✗	0.0 / 0.0 / 1
	3	✓	✗	✗	✗	0.0 / 0.0 / 1
	4	✓	✓	✗	✗	0.0 / 1.0 / 2
	5	✓	✓	✗	✗	0.0 / 1.0 / 2
	6	✓	✗	✗	✗	0.0 / 0.0 / 1
	7	✓	✗	✗	✗	0.0 / 0.0 / 1
	8	✓	✗	✗	✗	0.0 / 0.0 / 1
	9	✓	✓	✗	✗	0.0 / 1.0 / 2
	10	✓	✗	✗	✗	0.0 / 0.0 / 1
DP3	1	✓	✓	✓	✓	1.0 / 1.0 / 4
	2	✓	✓	✓	✓	1.0 / 1.0 / 4
	3	✓	✓	✓	✓	1.0 / 1.0 / 4
	4	✓	✓	✓	✓	1.0 / 1.0 / 4
	5	✓	✓	✓	✗	0.0 / 1.0 / 3
	6	✗	✓	✗	✗	0.0 / 0.0 / 1
	7	✗	✓	✗	✗	0.0 / 0.0 / 1
	8	✗	✓	✗	✗	0.0 / 0.0 / 1
	9	✓	✓	✓	✗	0.0 / 1.0 / 3
	10	✗	✓	✗	✗	0.0 / 0.0 / 1
3D Diffuser Actor	1	✓	✗	✗	✗	0.0 / 0.0 / 1
	2	✓	✓	✗	✗	0.0 / 1.0 / 2
	3	✓	✗	✗	✗	0.0 / 0.0 / 1
	4	✓	✓	✗	✗	0.0 / 1.0 / 2
	5	✓	✗	✗	✗	0.0 / 0.0 / 1
	6	✓	✓	✗	✗	0.0 / 1.0 / 2
	7	✗	✗	✗	✗	0.0 / 0.0 / 0
	8	✓	✓	✗	✗	0.0 / 1.0 / 2
	9	✓	✗	✗	✗	0.0 / 0.0 / 1
	10	✓	✓	✗	✗	0.0 / 1.0 / 2

TABLE XIV: **Detailed results on Scan the Bottle task.**

Method	Case Index	Pick the Bottle	Pick the Scanner	Scan	Place Bottle in Box	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	1	✓	✓	✓	✓	1.0 / 1.0 / 4
	2	✓	✓	✓	✓	1.0 / 1.0 / 4
	3	✓	✓	✓	✓	1.0 / 1.0 / 4
	4	✓	✓	✓	✓	1.0 / 1.0 / 4
	5	✓	✓	✓	✓	1.0 / 1.0 / 4
	6	✓	✓	✓	✓	1.0 / 1.0 / 4
	7	✗	✓	✗	✗	0.0 / 0.0 / 1
	8	✓	✓	✓	✓	1.0 / 1.0 / 4
	9	✓	✓	✓	✓	1.0 / 1.0 / 4
	10	✓	✓	✓	✓	1.0 / 1.0 / 4
ACT	1	✓	✓	✗	✗	0.0 / 1.0 / 2
	2	✗	✓	✗	✗	0.0 / 0.0 / 1
	3	✓	✓	✗	✗	0.0 / 1.0 / 2
	4	✓	✓	✓	✓	1.0 / 1.0 / 4
	5	✗	✓	✗	✗	0.0 / 0.0 / 1
	6	✓	✓	✗	✓	0.0 / 1.0 / 3
	7	✗	✓	✗	✗	0.0 / 0.0 / 1
	8	✓	✓	✗	✗	0.0 / 1.0 / 2
	9	✗	✗	✗	✗	0.0 / 0.0 / 0
	10	✗	✓	✗	✗	0.0 / 0.0 / 1
DP3	1	✓	✗	✗	✗	0.0 / 0.0 / 1
	2	✓	✓	✓	✓	1.0 / 1.0 / 4
	3	✓	✓	✗	✗	0.0 / 1.0 / 2
	4	✗	✓	✗	✗	0.0 / 0.0 / 1
	5	✗	✓	✗	✗	0.0 / 0.0 / 1
	6	✗	✓	✗	✗	0.0 / 0.0 / 1
	7	✓	✓	✓	✓	1.0 / 1.0 / 4
	8	✗	✗	✗	✗	0.0 / 0.0 / 0
	9	✓	✗	✗	✗	0.0 / 0.0 / 1
	10	✓	✓	✓	✓	1.0 / 1.0 / 4
3D Diffuser Actor	1	✗	✓	✗	✗	0.0 / 0.0 / 1
	2	✓	✗	✗	✗	0.0 / 0.0 / 1
	3	✓	✓	✓	✓	1.0 / 1.0 / 4
	4	✓	✓	✗	✓	0.0 / 1.0 / 3
	5	✗	✓	✗	✗	0.0 / 0.0 / 1
	6	✓	✗	✗	✗	0.0 / 0.0 / 1
	7	✓	✓	✓	✓	1.0 / 1.0 / 4
	8	✓	✓	✗	✗	0.0 / 1.0 / 2
	9	✗	✓	✗	✗	0.0 / 0.0 / 1
	10	✗	✓	✗	✗	0.0 / 0.0 / 1

TABLE XV: **Detailed results on Wear the Scarf task.**

Method	Case Index	Pick the Scarf	Lift the Scarf	Wear the Scarf	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	1	✓	✓	✓	1.0 / 1.0 / 3
	2	✓	✓	✓	1.0 / 1.0 / 3
	3	✓	✓	✗	0.0 / 1.0 / 2
	4	✗	✗	✗	0.0 / 0.0 / 0
	5	✓	✓	✓	1.0 / 1.0 / 3
	6	✓	✓	✓	1.0 / 1.0 / 3
	7	✓	✓	✓	1.0 / 1.0 / 3
	8	✓	✓	✓	1.0 / 1.0 / 3
	9	✗	✗	✗	0.0 / 0.0 / 0
	10	✓	✓	✓	1.0 / 1.0 / 3
ACT	1	✗	✗	✗	0.0 / 0.0 / 0
	2	✓	✓	✓	1.0 / 1.0 / 3
	3	✓	✓	✓	1.0 / 1.0 / 3
	4	✗	✗	✗	0.0 / 0.0 / 0
	5	✓	✓	✓	1.0 / 1.0 / 3
	6	✗	✗	✗	0.0 / 0.0 / 0
	7	✓	✓	✗	0.0 / 1.0 / 2
	8	✓	✓	✓	1.0 / 1.0 / 3
	9	✓	✗	✗	0.0 / 1.0 / 1
	10	✓	✓	✗	0.0 / 1.0 / 2

TABLE XVI: **Detailed results on Press the Cap of Bottle task.**

Method	Case Index	Pick the Bottle	Place the Bottle	Press the Cap	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	1	✓	✓	✓	1.0 / 1.0 / 3
	2	✓	✓	✓	1.0 / 1.0 / 3
	3	✓	✓	✗	0.0 / 1.0 / 2
	4	✗	✗	✗	0.0 / 0.0 / 0
	5	✓	✓	✗	0.0 / 1.0 / 2
	6	✗	✗	✗	0.0 / 0.0 / 0
	7	✓	✓	✗	0.0 / 1.0 / 2
	8	✓	✓	✓	1.0 / 1.0 / 3
	9	✓	✓	✓	1.0 / 1.0 / 3
	10	✓	✓	✗	0.0 / 1.0 / 2
ACT	1	✓	✗	✗	0.0 / 0.0 / 1
	2	✓	✗	✗	0.0 / 0.0 / 1
	3	✓	✗	✗	0.0 / 0.0 / 1
	4	✓	✓	✓	1.0 / 1.0 / 3
	5	✓	✗	✗	0.0 / 0.0 / 1
	6	✗	✗	✗	0.0 / 0.0 / 0
	7	✓	✓	✗	0.0 / 1.0 / 2
	8	✗	✗	✗	0.0 / 0.0 / 0
	9	✗	✗	✗	0.0 / 0.0 / 0
	10	✗	✗	✗	0.0 / 0.0 / 0

TABLE XVII: **Detailed results on Scan the Bottle with Single Arm task.**

Method	Case Index	Pick the Scanner	Scan	Place the Scanner	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	1	✓	✓	✓	1.0 / 1.0 / 3
	2	✓	✓	✓	1.0 / 1.0 / 3
	3	✓	✓	✓	1.0 / 1.0 / 3
	4	✓	✓	✓	1.0 / 1.0 / 3
	5	✓	✗	✓	0.0 / 1.0 / 2
	6	✓	✓	✓	1.0 / 1.0 / 3
	7	✓	✗	✓	0.0 / 1.0 / 2
	8	✓	✓	✓	1.0 / 1.0 / 3
	9	✓	✓	✓	1.0 / 1.0 / 3
	10	✓	✓	✓	1.0 / 1.0 / 3
ACT	1	✓	✗	✗	0.0 / 1.0 / 1
	2	✓	✗	✗	0.0 / 1.0 / 1
	3	✗	✗	✗	0.0 / 0.0 / 0
	4	✗	✗	✗	0.0 / 0.0 / 0
	5	✓	✗	✓	0.0 / 1.0 / 2
	6	✓	✓	✓	1.0 / 1.0 / 3
	7	✗	✗	✗	0.0 / 0.0 / 0
	8	✓	✓	✓	1.0 / 1.0 / 3
	9	✓	✗	✓	0.0 / 1.0 / 2
	10	✓	✗	✓	0.0 / 1.0 / 2

TABLE XVIII: **Detailed generalization experiment results on Carry the Tray task.**

Method	Generalization Type	Case Index	Pick the Tray	Lift the Tray	Place the Tray	Return	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	Normal	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Unseen Objects	Round Tray	✓	✓	✓	✓	1.0 / 1.0 / 4
		Brown Tray	✓	✓	✓	✓	1.0 / 1.0 / 4
	Lighting Changes	Dark	✓	✓	✓	✓	1.0 / 1.0 / 4
		Flickering	✓	✓	✗	✗	0.0 / 1.0 / 2
	Object Interference	Rubik's Cubes	✓	✓	✓	✓	1.0 / 1.0 / 4
		Colorful Cubes	✓	✓	✓	✓	1.0 / 1.0 / 4
ACT	Normal	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Unseen Objects	Round Tray	✗	✗	✗	✗	0.0 / 0.0 / 0
		Brown Tray	✗	✗	✗	✗	0.0 / 0.0 / 0
	Lighting Changes	Dark	✗	✗	✗	✗	0.0 / 0.0 / 0
		Flickering	✗	✗	✗	✗	0.0 / 0.0 / 0
	Object Interference	Rubik's Cubes	✓	✓	✓	✓	1.0 / 1.0 / 4
		Colorful Cubes	✓	✓	✓	✓	1.0 / 1.0 / 4
DP3	Normal	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Unseen Objects	Round Tray	✓	✓	✓	✓	1.0 / 1.0 / 4
		Brown Tray	✓	✗	✗	✗	0.0 / 0.0 / 1
	Lighting Changes	Dark	✓	✓	✓	✓	1.0 / 1.0 / 4
		Flickering	✓	✓	✓	✓	1.0 / 1.0 / 4
	Object Interference	Rubik's Cubes	✓	✗	✗	✗	0.0 / 0.0 / 1
		Colorful Cubes	✓	✓	✗	✗	0.0 / 1.0 / 2
3D Diffuser Actor	Normal	Normal Setting	✓	✓	✓	✗	0.0 / 1.0 / 3
	Unseen Objects	Round Tray	✓	✓	✗	✗	0.0 / 1.0 / 2
		Brown Tray	✓	✓	✗	✗	0.0 / 1.0 / 2
	Lighting Changes	Dark	✗	✗	✗	✗	0.0 / 0.0 / 0
		Lighting	✗	✗	✗	✗	0.0 / 0.0 / 0
	Object Interference	Rubik's Cubes	✓	✓	✗	✗	0.0 / 1.0 / 2
		Colorful Cubes	✓	✗	✗	✗	0.0 / 0.0 / 1

TABLE XIX: **Detailed generalization experiment results on Handover and Insert the Plate task.**

Method	Generalization Type	Case Index	Pick the Plate	Handover	Insert the Plate	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	Normal	Normal Setting	✓	✓	✓	1.0 / 1.0 / 3
	Unseen Objects	Rectangular Plate	✓	✗	✗	0.0 / 1.0 / 1
		Green Plate	✓	✓	✓	1.0 / 1.0 / 3
	Lighting Changes	Dark	✓	✓	✓	1.0 / 1.0 / 3
		Flickering	✓	✓	✓	1.0 / 1.0 / 3
	Object Interference	Colorful Cubes	✓	✓	✓	1.0 / 1.0 / 3
		Multi-Plate	✓	✓	✗	0.0 / 1.0 / 2
ACT	Normal	Normal Setting	✓	✓	✓	1.0 / 1.0 / 3
	Unseen Objects	Rectangular Plate	✗	✗	✗	0.0 / 0.0 / 0
		Green Plate	✓	✗	✗	0.0 / 1.0 / 1
	Lighting Changes	Dark	✗	✗	✗	0.0 / 0.0 / 0
		Flickering	✗	✗	✗	0.0 / 0.0 / 0
	Object Interference	Colorful Cubes	✓	✓	✗	0.0 / 1.0 / 2
		Multi-Plate	✓	✓	✗	0.0 / 1.0 / 2
DP3	Normal	Normal Setting	✓	✓	✓	1.0 / 1.0 / 3
	Unseen Objects	Rectangular Plate	✓	✗	✗	0.0 / 1.0 / 1
		Green Plate	✓	✓	✓	1.0 / 1.0 / 3
	Lighting Changes	Dark	✓	✓	✗	0.0 / 1.0 / 2
		Flickering	✓	✓	✓	1.0 / 1.0 / 3
	Object Interference	Colorful Cubes	✗	✗	✗	0.0 / 0.0 / 0
		Multi-Plate	✓	✗	✗	0.0 / 1.0 / 1
3D Diffuser Actor	Normal	Normal Setting	✓	✓	✗	0.0 / 1.0 / 2
	Unseen Objects	Rectangular Plate	✓	✗	✗	0.0 / 1.0 / 1
		Green Plate	✓	✗	✗	0.0 / 1.0 / 1
	Lighting Changes	Dark	✗	✗	✗	0.0 / 0.0 / 0
		Flickering	✗	✗	✗	0.0 / 0.0 / 0
	Object Interference	Colorful Cubes	✓	✗	✗	0.0 / 1.0 / 1
		Multi-Plate	✓	✗	✗	0.0 / 1.0 / 1

TABLE XX: **Detailed generalization experiment results on Wipe the Plate task.**

Method	Generalization Type	Case Index	Pick the Sponge	Pick the Plate	Wipe	Place	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	Normal	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Unseen Objects	Rag	✓	✓	✓	✓	1.0 / 1.0 / 4
		Rectangular Plate	✓	✓	✓	✗	0.0 / 1.0 / 3
	Lighting Changes	Dark	✓	✓	✓	✓	1.0 / 1.0 / 4
		Flickering	✓	✓	✓	✓	1.0 / 1.0 / 4
	Object Interference	Colorful Cubes	✓	✓	✓	✓	1.0 / 1.0 / 4
		Multi-Plate	✓	✓	✓	✗	0.0 / 1.0 / 3
ACT	Normal	Normal Setting	✓	✓	✗	✓	0.0 / 1.0 / 3
	Unseen Objects	Rag	✓	✗	✗	✗	0.0 / 0.0 / 1
		Rectangular Plate	✓	✓	✗	✗	0.0 / 1.0 / 2
	Lighting Changes	Dark	✗	✗	✗	✗	0.0 / 0.0 / 0
		Flickering	✗	✗	✗	✗	0.0 / 0.0 / 0
	Object Interference	Colorful Cubes	✓	✗	✗	✗	0.0 / 0.0 / 1
		Multi-Plate	✓	✗	✗	✗	0.0 / 0.0 / 1
DP3	Normal	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Unseen Objects	Rag	✓	✓	✓	✓	1.0 / 1.0 / 4
		Rectangular Plate	✓	✗	✗	✗	0.0 / 0.0 / 1
	Lighting Changes	Dark	✓	✓	✓	✓	1.0 / 1.0 / 4
		Flickering	✓	✓	✓	✓	1.0 / 1.0 / 4
	Object Interference	Colorful Cubes	✗	✗	✗	✗	0.0 / 0.0 / 0
		Multi-Plate	✗	✗	✗	✗	0.0 / 0.0 / 0
3D Diffuser Actor	Normal	Normal Setting	✓	✓	✗	✗	0.0 / 1.0 / 2
	Unseen Objects	Rag	✓	✓	✗	✗	0.0 / 1.0 / 2
		Rectangular Plate	✗	✓	✗	✗	0.0 / 0.0 / 1
	Lighting Changes	Dark	✓	✗	✗	✗	0.0 / 0.0 / 1
		Flickering	✓	✗	✗	✗	0.0 / 0.0 / 1
	Object Interference	Colorful Cubes	✓	✓	✗	✗	0.0 / 1.0 / 2
		Multi-Plate	✓	✓	✗	✗	0.0 / 1.0 / 2

TABLE XXI: **Detailed generalization experiment results on Scan the Bottle task.**

Method	Case Index	Pick the Bottle	Pick the Scanner	Scan	Place Bottle in Box	SR (%) ↑ /Loc-SR (%) ↑ / Score ↑
PPI	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Box	✓	✓	✓	✗	1.0 / 1.0 / 3
	Yogurt Bottle	✓	✓	✓	✗	1.0 / 1.0 / 3
ACT	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Box	✗	✓	✗	✗	0.0 / 0.0 / 1
	Yogurt Bottle	✗	✓	✗	✗	0.0 / 0.0 / 1
DP3	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Box	✗	✓	✗	✗	0.0 / 0.0 / 1
	Yogurt Bottle	✗	✓	✗	✗	0.0 / 0.0 / 1
3D Diffuser Actor	Normal Setting	✓	✓	✓	✓	1.0 / 1.0 / 4
	Box	✗	✓	✗	✗	0.0 / 0.0 / 1
	Yogurt Bottle	✗	✓	✗	✗	0.0 / 0.0 / 1