AffordDP: Generalizable Diffusion Policy with Transferable Affordance
Supplementary Materials

A. Overview

This supplementary document provides additional information, results, and visualizations to supplement the main paper. Specifically, we include:

- Details on data collection;
- Information about the experimental setup;
- Descriptions of training and inference procedures;
- Additional experiment results.

B. Demonstration Collection

In this section, we provide a detailed explanation of our demonstration collection process, encompassing both simulation and real-world environments.

B.1. Simulation Data Collection

We employ CuRobo [32] as our motion planner. Given the world coordinates of the robot's base and the target end-effector pose, CuRobo is capable of computing a feasible robotic motion trajectory. Then we utilize the bounding boxes of the annotated parts provided by GAPartnet [11] to calculate the grasping center and the pull direction post-grasping. We further plan a set of waypoints along the computed pulling direction. By combining these waypoints with the rotational components of the grasp center, we leverage CuRobo to calculate a series of feasible robot action sequences. We place an RGBD camera in the simulation environment, and the entire trajectory is recorded using this camera.

B.2. Real-world Data Collection

Expert demonstrations are collected via the teleoperation system, where a human operator controls the system and the camera records the entire process, including RGB image and depth information, shown as in Fig. 5. We use a RealSense D455 RGBD camera to capture point cloud streams at a resolution of 640 × 480. We perform hand-eye calibration in the real world. The calibration process enabled us to accurately determine the transformation relationship from the camera coordinate system to the robot base coordinate system. Then, we utilize the camera intrinsic parameters and this transformation matrix to convert the RGB-D images into point clouds in the robot's base coordinate system, which facilitates subsequent policy inference.

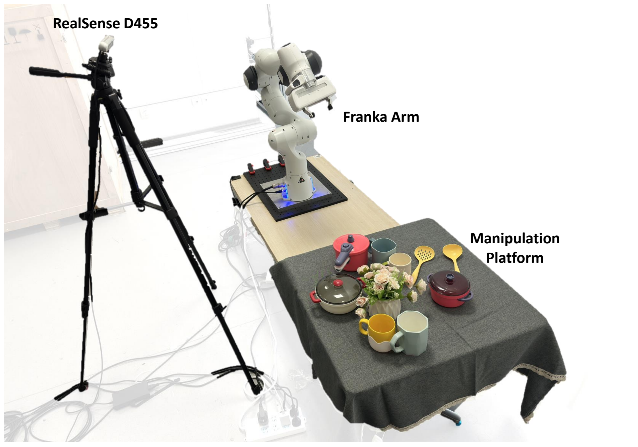

Figure 5. Real world experiment setup.

C. Experiment Details

C.1. Setup

During the data collection process, we introduced varying levels of random noise to the initial positions and rotations of the objects, as well as to the initial position of the robotic arm. Here is the unified equation to represent the noise injection process.

$$x' = x + 2\xi\mathcal{N}(0,1) - \xi. \tag{14}$$

The magnitudes of these random noises ξ correspond to different difficulty levels, as summarized in Tab. 6. For the unified policy training in simulation, we construct datasets using the Easy level. We summarize object categories and the number of object instances in Tab. 7.

	Hard	Median	Easy
Robot Position Noise	0.1	0.05	0.025
Robot Dof Noise	0.1	0.05	0.025
Object Position Noise	0.05	0.05	0.01

Table 6. Noise levels for different difficulty settings during data collection: Hard, Median, and Easy. The noise values represent the amount of random noise introduced to the robot's position, degrees of freedom, and object position.

We provide additional visualizations for our objects in real-world tasks, shown in Fig. 6. We visualize all the seen instances, unseen instances, and unseen categories.

C.2. Training

We employ a convolutional network-based diffusion model as the backbone. The visual input consists of a point cloud

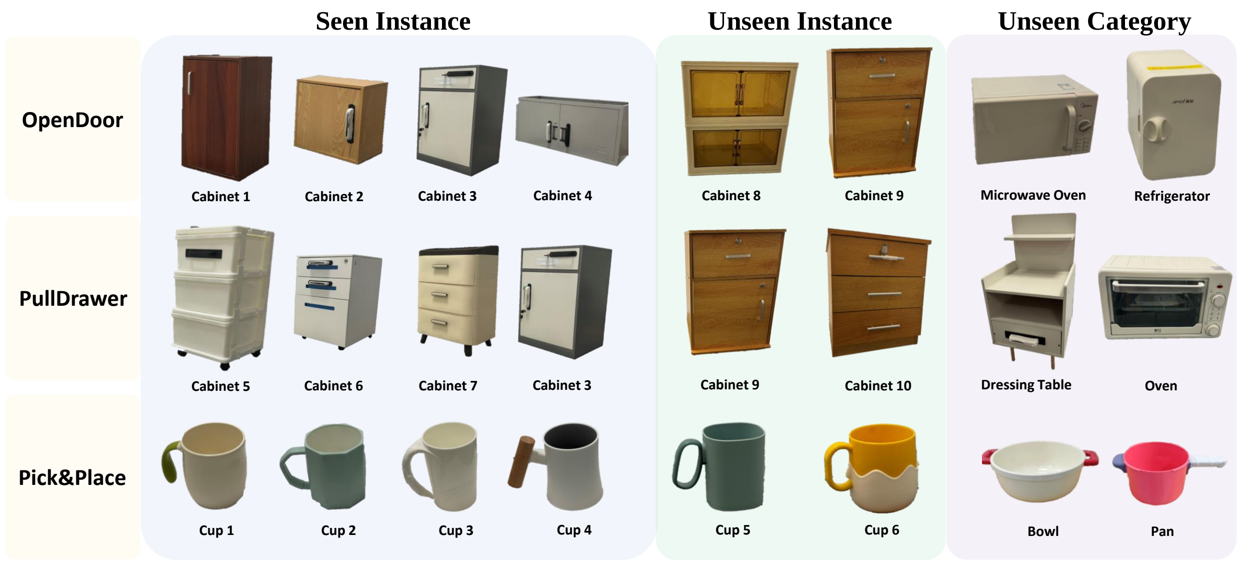

Figure 6. Objects used in Real-World Tasks, including the seen instances, unseen instances, and unseen categories.

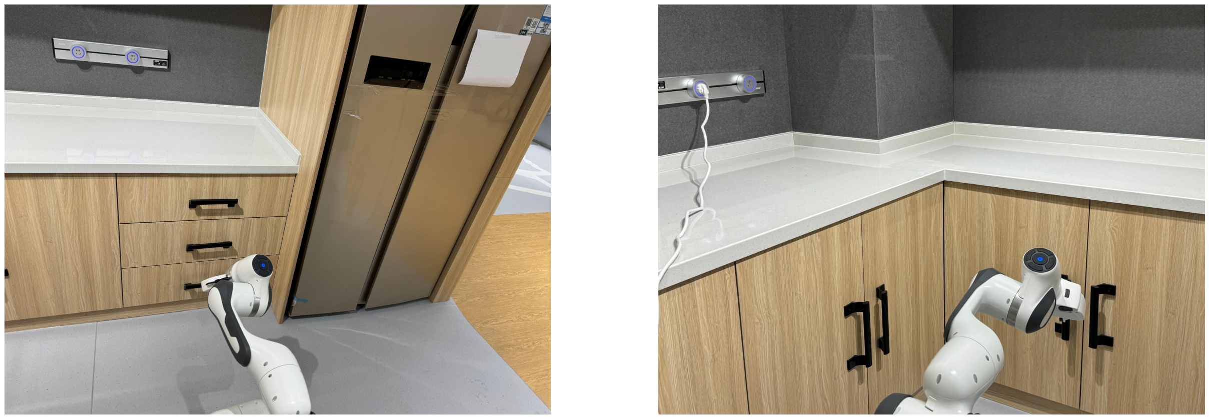

Figure 7. Visualizations of different unseen scenes. We deployed our policy to a real-world kitchen environment in a zero-shot manner, and it still demonstrated commendable generalization.

Task Category	Task Name	#Train Instances	#Test Instances
Simulation	OpenDoor	5	5
	PullDrawer	5	5
Real-world	OpenDoor	4	4
	PullDrawer	4	4
	Pick&Place	4	4

Table 7. Number of training and testing instances for different tasks under the unified policy training setting. Simulation tasks include OpenDoor and PullDrawer, with 5 training instances and 5 testing instances each. Real-world tasks include OpenDoor, PullDrawer, and Pick&Place, with 4 training instances and 4 testing instances each.

without colors, which is downsampled from the raw point cloud using Farthest Point Sampling (FPS). We consistently use 4096 points for all simulated and real-world tasks. The representations encoded from point clouds, robot poses, and affordances are concatenated to form a unified representation with a dimension of 256. The policy in the real world is trained on a single A100 40GB GPU for two days. Hyperparameters related to policy training are shown in Tab. 8.

C.3. Inference

During inference, AffordDP retrieves the most similar object from the affordance memory and transfers its static and dynamic affordances to the target object. Subsequently, the affordance, visual observation, and robot proprioception are

Hyperparameter	Default
Num epochs	4000
Batch Size	128
Horizon	16
Observation Steps	2
Action Steps	8
Num points	4096
Affordance MLP size	[64,64]
Affordance transformer size	64
Num attention heads	4
Num attention layers	4
Num train timesteps	500
Num inference steps	10
Learning Rate (LR)	1.0e-4
Weight decay	1.0e-6

Table 8. Hyperparameters of Policy Network.

fed into the policy network to predict the action. The action is then fed into the low-level controller to operate the robot.

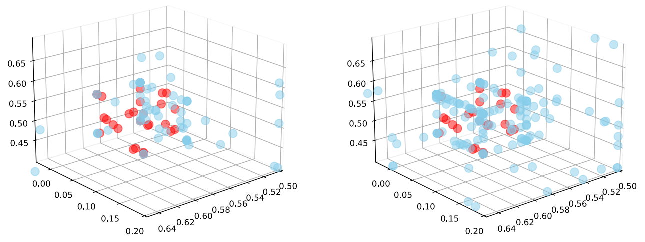

Figure 8. Spatial generalization with 30 expert demonstrations. We evaluate the performance across 1000 random positions in the 3D space. Expert demonstrations are marked as ●, and successful trials are marked as ●. In the PullDrawer task, DP3 only succeeds in regions close to the expert demonstrations. In contrast, our method generalizes effectively, covering a broader range of 3D space, including regions not represented in the expert demonstrations.

C.4. Evaluation

Here we list the evaluation criteria for each task:

- OpenDoor (sim): The task is considered successful if the robotic arm accurately grasps the specific door handle and opens the door to an angle of 30 degrees.
- PullDrawer (sim): The task is considered successful if the robotic arm accurately grasps the specific drawer handle and extends the drawer by a distance of 0.15 meters.
- OpenDoor: The task is considered successful if the robotic arm accurately grasps a specific door handle and actuates the door to a predetermined angular displacement.
- PullDrawer: The task is considered successful if the robotic arm accurately grasps a specific drawer handle and extends the drawer along a defined linear distance.
- Pick&Place: The task is considered successful if the robotic arm accurately grasps a specific region of the object and places the object onto a designated storage rack.

D. Additional experiment results

D.1. Spatial Generalization

Following DP3 [41], we utilized PullDrawer as our benchmark task in simulation, providing 30 demonstrations (visualized by ●). We randomly initialized the positions of the objects and conducted 1,000 evaluations (successful positions are visualized by ●), shown in Fig. 8. The number of successful trials using DP3 is 65, whereas our approach achieved 170 successful trials.

D.2. More Real World Experiments

We give the real-world results for each object in Tab. 9, which is supplementary to Tab. 4 in our main paper.

D.3. Semantic correspondence model selection

We compared several foundational vision models (CLIP, DINOv2, SD, and SD-DINOv2) and provided qualitative results on semantic correspondence transfer, shown as Fig. 9. The lack of sufficient semantic understanding in CLIP can lead to incorrect semantic correspondence transfer. Similarly, the SD model also exhibits transfer errors in certain tasks. In comparison to DINOv2 and SD, the SD-DINOv2 model demonstrates greater stability and exhibits smaller errors in the transfer of static affordance within the pixel space.

D.4. Unseen scene generalization

To further demonstrate the generalization capability of our method, we applied the policy zero-shot transfer to unseen scenes(kitchen environment), as shown in Fig. 7. We recalibrated the camera and cropped the extraneous points from the point cloud to execute our policy. Surprisingly, despite the stark contrast with our training scene, our method still demonstrated robust generalization capabilities. Please refer to our project website for detailed videos.

E. Limitations

The limitations of our method are mainly two-fold: scenarios where foundation models fail, such as severe occlusion or visual distortion; and tasks requiring precise force control where affordance extraction is difficult, like grasping eggs without breaking it.

	Open Door							
Methods	Cabinet 1	Cabinet 2	Cabinet 3	Cabinet 4	Cabinet 8	Cabinet 9	Microwave Oven	Refrigerator
DP	0/10	0/10	0/10	0/10	0/10	0/10	0/10	0/10
DP3	3/10	2/10	3/10	0/10	0/10	0/10	**6/10**	0/10
Ours	**6/10**	**8/10**	**10/10**	**8/10**	**6/10**	**4/10**	**6/10**	**4/10**

	Pull Drawer							
Methods	Cabinet 5	Cabinet 6	Cabinet 7	Cabinet 3	Cabinet 9	Cabinet 10	Dressing Table	Oven
DP	0/10	0/10	0/10	0/10	0/10	0/10	0/10	0/10
DP3	1/10	0/10	0/10	3/10	0/10	0/10	2/10	0/10
Ours	**6/10**	**5/10**	**7/10**	**9/10**	**5/10**	**4/10**	**5/10**	**3/10**

	Pick&Place							
Methods	Cup 1	Cup 2	Cup 3	Cup 4	Cup 5	Cup 6	Bowl	Pan
DP	0/10	0/10	0/10	0/10	0/10	0/10	0/10	0/10
DP3	**6/10**	3/10	2/10	**5/10**	3/10	3/10	0/10	3/10
Ours	**6/10**	**6/10**	**5/10**	4/10	**6/10**	**7/10**	**4/10**	**6/10**

Table 9. Success rates of different methods on real-world tasks. We evaluated the performance of different methods on various objects.

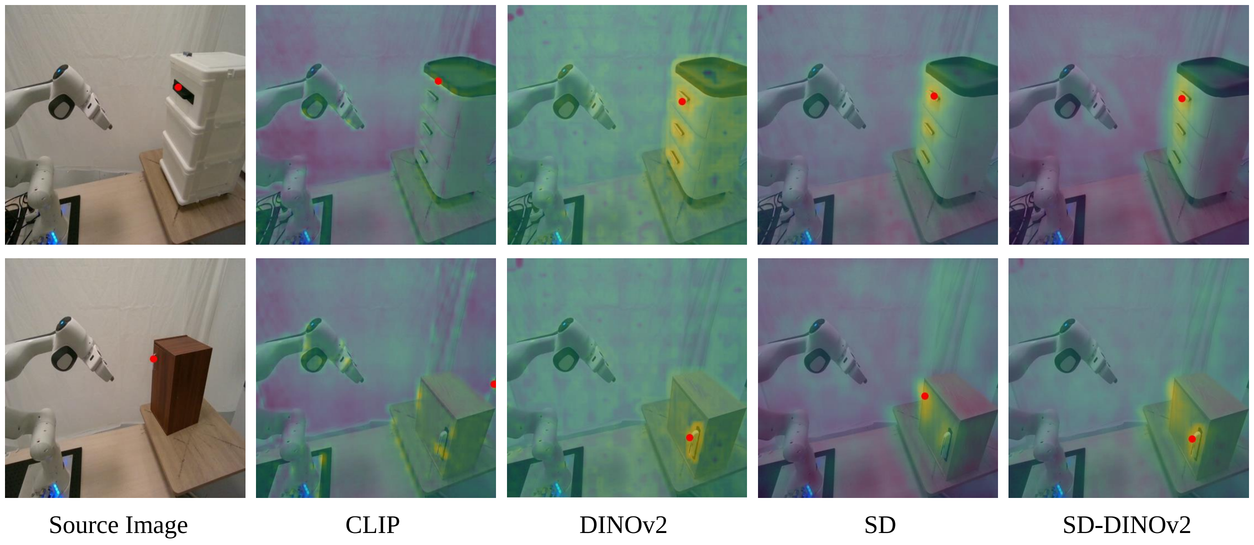

Figure 9. Comparison of semantic correspondence transfer among different foundational models. Red points ● represent the static affordance and its corresponding transferred results.