AdaAfford: Learning to Adapt Manipulation Affordance for 3D Articulated Objects via Few-shot Interactions

Yian Wang[1,2⋆], Ruihai Wu[1,2⋆], Kaichun Mo[3⋆], Jiaqi Ke[1,2], Qingnan Fan[4], Leonidas Guibas[3], and Hao Dong[1,2,5†]

[1] CFCS, CS Dept., PKU
[2] AIIT, PKU
{yianwang,wuruihai,kjq001220,hao.dong}@pku.edu.cn
[3] Stanford University
{kaichun,guibas}@cs.stanford.edu
[4] Tencent AI Lab
fqnchina@gmail.com
[5] Peng Cheng Lab
https://hyperplane-lab.github.io/AdaAfford

1 More Detailed Data Statistics

Table 1 summarizes the data statistics and splits. Figure 1 visualizes some example shapes from our dataset.

2 More Details about Method

In our setting, the subsequent interactions always continue from the previous interactions. But it actually doesn't matter for our method, we can take any distribution of test-time interactions in the training process and it won't violate our design.

The input point cloud O is the current observation and might be changed if an interaction successfully moves the object.

3 More Experiment Settings

Details of Baselines For **Ours-fps**, **Where2Act-interaction** and **Where2Act-adaptation** baselines, we augment the FPS method with an actionability score. In detail, we only select the points with higher actionability scores than a preset threshold (*e.g.*, 0.5). If there are not enough such points, the threshold will be set lower until there exist at least 50 points whose actionability scores are higher than the threshold. For **Where2Act-adaptation** baseline, we first train a network to give the similarities between points. For point p_1 and point p_2, given their point features extracted by PointNet++ and the distance between them, the network outputs a similarity score $sim_{p_1p_2}$. Then, an interaction $I = (O, p, R, m)$ acting on p with action score s_u will influence point q by:

$$(r - s_u) * sim_{pq} \tag{1}$$

⋆ Equal contribution
† Corresponding author.

Fig. 1. Data Visualization. We show example shapes of object categories in our paper.

where $r = 1$ if $m > \tau$ (*e.g.*, $\tau = 0.01$) or $r = 0$, $u = (p, R)$. Specifically, if the original an actionability score of point q is a_q and the original action score of an arbitrary action $u_* = (q, R_*)$ on point q is s_{u_*}, the new action score $s^{new}_{u_*}$ and actionability score a^{new}_q would be:

$$s^{new}_{u_*} = s_{u_*} + (r - s_u) * sim_{pq} \tag{2}$$

$$a^{new}_q = a_q + (r - s_u) * sim_{pq} \tag{3}$$

To train the network to give similarity between points, similar to our method, we use the ground truth result of the action u_* as the regression target of $s^{new}_{u_*}$.

More baselines We employ several baselines using FPS method to sample interaction points, and the results show the usefulness of the proposed AIP module of our framework.

- **Ours-purefps:** that directly uses FPS method to sample interaction points without using actionability scores.
- **Ours-argfps:** that uses FPS augmented with actionability scores to select interaction points. When sampling a new point, we combine its distance to the sampled point set with its actionability score while doing FPS, as the weighted distance.
- **Where2Act-interaction:** the Where2Act method augmented with four additional interaction observations as inputs where the interaction positions are

Table 1. We first summarize the shape counts in our dataset for pushing and pulling shapes over all categories, in which there are three data splits: training data from the training categories, test data from the training categories, and data from the test categories. We use the first split to train and use the rest two for evaluation. We further show the shape counts in our two additional tasks: pulling closed door and pushing faucet (denoted as Closed Door and Faucet for brevity).

Train-Cats	Box	Microwave	Door	Faucet	TrashCan
Train / Test	20 / 8	9 / 3	23 / 12	65 / 19	52 / 17
	Kettle	Refrigerator	Switch	Cabinet	Window
	22 / 7	32 / 11	53 / 17	270 / 75	40 / 18
Total	586 / 187				
Test-Cats	Table	Washer	Bucket	Pot	Safe
	95	16	36	23	29
Total	199				
ADDL Exp.	Category	Train data	Test data		
Closed door	Cabinet	74	11		
Faucet	Faucet	15	4		

uniformly sampled over the predicted affordance heatmap using Furthest Point Sampling (FPS) and we train an additional encoding branch similar to the *Adaptive Information Encoder* to extract the additional input feature; As the Where2Act baseline only takes visual information, in this baseline, we train an end-to-end network takes not only the visual input but also 4 interactions generated by farthest-point-sample.

- **Ours-random:** a variant of our proposed method that we use randomly sampled interaction trials over the geometry instead of the AIP proposals;

4 More Results and Analysis

In Figure 2 and 3, we show more qualitative results. See the captions of these two figures for more details.

Table 2 shows the comparisons between different methods using FPS. In most cases, both **Ours-argfps** and **Ours-fps** achieve better results than **Ours-purefps**. Because in **Ours-purefps** baseline, FPS only cares about the 3D position of points discarding the point features. While **Ours-argfps** and **Ours-fps** utilize the action scores which are generated by point features and thus achieve better results. Results show that our framework gets better performance in most cases compared with those baselines, which further shows the effectiveness of our AIP module.

Compared to the **Where2Act-interaction** baseline that is fed with four interactions in one shot, our whole framework works better because our recur-

rent structure strategically and successively selects the most effective interaction trials.

Several aspects cause lower sample-success rate but don't hurt the contribution: a) There exist extreme cases in which no action will succeed (*e.g.*, door too heavy, window too slippery). b) Actions should be precise enough to complete given tasks. Minor changes in gripper poses may cause failures as the success criteria is set tightly, even when we have already selected the correct manipulation point.

Some objects in test-cat (*e.g.*, bucket) are easier to manipulate than those in the train-cat (*e.g.*, fridge), explaining that sometimes the test-cat numbers are higher than the train-cat ones.

Table 2. Quantitative Evaluations. Comparison with more baselines. Results show that our framework achieves the best performance in most cases.

		F-score (%)	Sample-Succ (%)
pushing all (train cat.)	Where2Act-interaction	72.13	31.53
	Ours-random	70.24/70.58/70.85	29.59/31.35/32.57
	Ours-purefps	66.78/69.43/70.65	28.23/31.50/29.51
	Ours-argfps	66.78/69.43/70.65	28.23/31.50/29.51
	Ours-fps	64.32/69.58/70.99	26.22/27.30/30.65
	Ours-final	**72.78**/**73.12**/**75.18**	**33.82**/**33.23**/**35.23**
pushing all (test cat.)	Where2Act-interaction	76.12	37.10
	Ours-random	75.12/76.92/76.98	30.78/30.78/29.48
	Ours-purefps	66.35/66.55/67.19	34.15/32.60/35.06
	Ours-argfps	74.04/75.03/76.63	33.11/34.54/36.49
	Ours-fps	66.17/67.27/69.08	33.64/35.19/**37.79**
	Ours-final	**77.58**/**77.63**/**78.42**	**34.97**/**36.75**/37.40
pulling all (train cat.)	Where2Act-interaction	38.28	3.89
	Ours-random	35.03/34.48/36.84	4.44/2.78/6.11
	Ours-purefps	35.46/37.54/37.35	2.78/5.56/2.78
	Ours-argfps	35.46/37.54/37.35	3.89/4.44/6.11
	Ours-fps	39.88/42.74/43.55	2.78/5.56/4.44
	Ours-final	**42.62**/**43.87**/**44.08**	**7.78**/**9.44**/**10.55**
pulling all (test cat.)	Where2Act-interaction	45.80	9.73
	Ours-random	41.97/44.88/46.11	6.13/4.78/8.26
	Ours-purefps	43.60/48.91/47.36	6.96/5.22/3.91
	Ours-argfps	45.17/47.39/50.60	**8.69**/7.22/10.00
	Ours-fps	43.67/42.77/48.33	4.35/3.91/4.78
	Ours-final	**49.51**/**50.00**/**51.33**	5.21/**7.39**/**10.45**
pulling closed door	Where2Act-interaction	66.79	9.09
	Ours-random	52.41/54.25/53.37	7.14/6.84/6.53
	Ours-purefps	53.53/59.81/67.20	6.67/7.64/10.71
	Ours-argfps	58.42/62.31/68.72	8.94/11.25/13.75
	Ours-fps	**59.79**/63.43/69.13	8.88/11.33/12.10
	Ours-final	57.83/**65.60**/**79.65**	**10.86**/**11.57**/**22.14**
pushing faucet	Where2Act-interaction	79.85	80.97
	Ours-random	72.61/76.29/79.16	61.81/79.01/80.82
	Ours-purefps	73.39/79.13/79.85	61.88/76.59/72.50
	Ours-argfps	74.66/78.30/79.61	61.42/66.65/74.75
	Ours-fps	74.19/79.36/77.95	60.44/70.12/77.41
	Ours-final	**77.42**/**83.06**/**83.83**	**65.90**/**81.66**/**82.14**

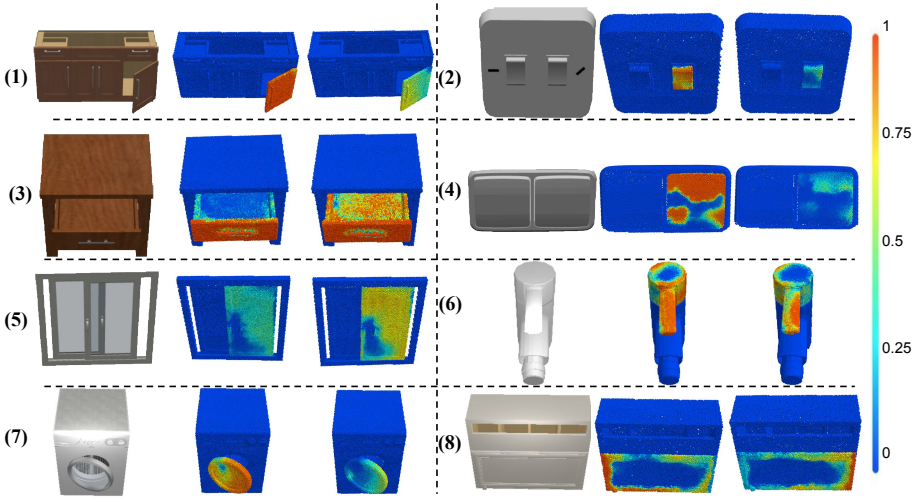

Fig. 2. We visualize more results for the adapted affordance predictions given by the AAP module conditioned on different hidden kinematic and dynamic information. From the first tor the last block, we respectively change the 1) mass of target part 2) joint friction 3) friction coefficient on the target part's surface 4) joint friction 5) friction coefficient on the target part's surface 6) rotating direction of the faucet 7) mass of target part 8) axis location of the door, and clearly see reasonable adaptions in affordance predictions.

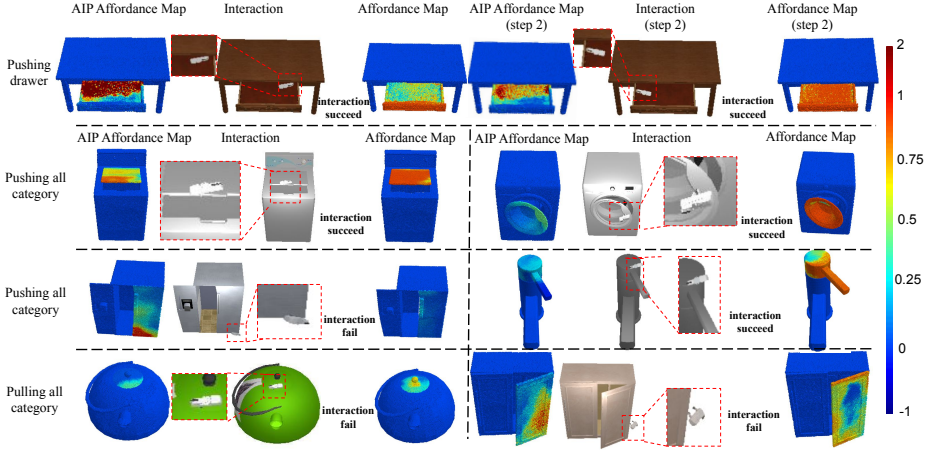

Fig. 3. We visualize more results for the interactions proposed by the AIP module and the corresponding AIP affordance map predictions. In the first row, we show the initial and the second AIP affordance maps, the corresponding proposed interactions, and the posterior affordance map predictions. In the last three rows, we present six more examples that only one interaction is needed.