Supplementary Materials for
Where2Explore: Few-shot Affordance Learning for Unseen Novel Categories of Articulated Objects

Anonymous Author(s)
Affiliation
Address
email

1 Error Control

To avoid randomness and prove the universal effectiveness of our framework in different scenarios, we conduct 4 experiments with different category combinations in training, and use the average scores as the results reported in the main paper.

The 4 combinations are {cabinet, faucet, window}, {cabinet, switch, refrigerator}, {table, Faucet, refrigerator} and {table, switch, window}. We choose these category combinations as they can cover 3 representative articulation types (revolute, prismatic, and sliding joints). Table 1 shows detailed scores of each combination. **Note that AdaAfford requires additional test-time interactions with target unseen objects**, while other methods only use visual observations of those objects as inputs.

Method	F-score		Sample successful rate	
	Pushing	Pulling	Pushing	Pulling
Where2Act	21.8 / 24.1 / 26.4	6.2 / 8.5 / 9.9	12.7 / 14.6 / 16.9	3.2 / 3.6 / 6.6
AdaAfford	26.8 / 29.4 / 30.2	3.7 / 3.9 / 5.0	**29.7** / 31.6 / 36.8	8.8 / 8.5 / 9.5
Ours	**36.8 / 41.1 / 46.0**	**13.1 / 13.3 / 22.8**	28.4 / **33.7 / 38.7**	**10.9 / 11.9 / 12.6**
Combination one				
Where2Act	24.1 / 28.3 / 31.1	10.9 / 11.7 / 11.3	14.2 / 16.5 / 19.8	6.7 / 6.6 / 8.7
AdaAfford	27.6 / 30.3 / 32.0	5.4 / 5.9 / 6.0	25.1 / 33.6 / 35.2	8.7 / 9.3 / 11.3
Ours	**37.4 / 40.0 / 40.2**	**13.0 / 14.3 / 28.9**	**33.3 / 39.8 / 40.3**	**11.8 / 10.9 / 15.8**
Combination two				
Where2Act	24.4 / 26.4 / 27.6	5.3 / 6.4 / 8.7	12.6 / 14.7 / 17.6	4.2 / 5.0 / 5.6
AdaAfford	25.0 / 28.9 / 31.4	3.1 / 3.1 / 3.5	25.3 / 26.9 / 38.5	8.9 / 8.4 / 13.3
Ours	**32.1 / 35.7 / 40.7**	**11.4 / 11.5 / 18.9**	**33.7 / 37.7 / 41.5**	**9.8 / 10.7 / 17.0**
Combination three				
Where2Act	32.1 / 33.2 / 36.5	3.2 / 3.4 / 4.1	23.3 / 22.2 / 25.3	1.5 / 2.0 / 3.9
AdaAfford	30.6 / 32.1 / 33.4	2.6 / 3.1 / 3.1	28.7 / 33.1 / **37.7**	9.8 / 11.4 / 10.3
Ours	**35.8 / 37.2 / 39.5**	**10.9 / 10.9 / 16.2**	**29.8 / 37.9 /** 37.5	**13.5 / 11.6 / 14.2**
Combination four				

Table 1: Few-shot learning on novel categories using different interaction budget (1, 2, 5).

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

2 More Experimental Results and Analysis

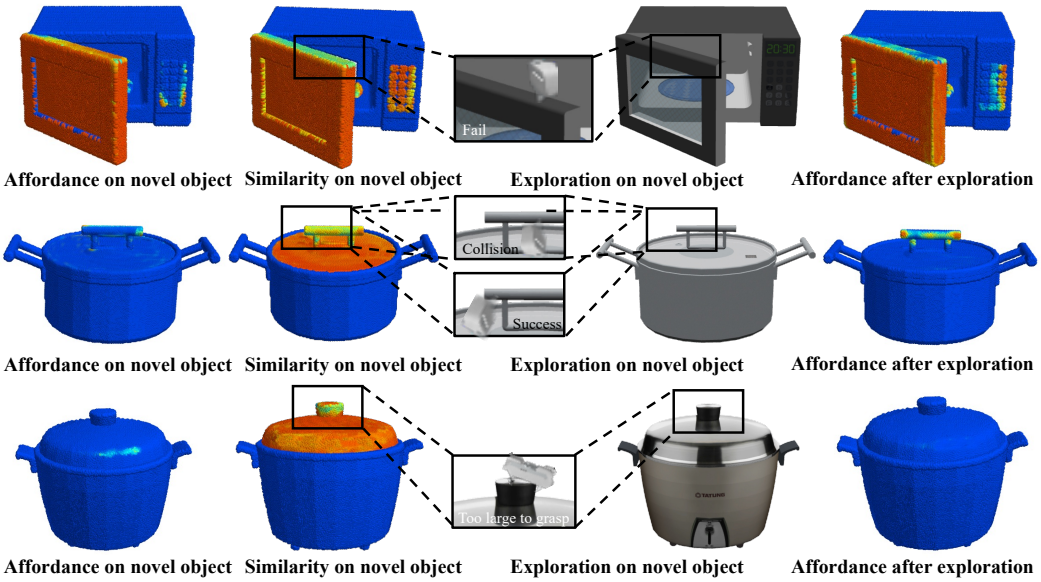

Figure 1: Visualization of pushing (top and middle rows) and pulling (bottom row) affordance with similarity prediction on novel object categories. The action directions are set to be the normal directions of each point in this visualization.

We visualize more similarity-guided exploration in Figure 1. We can see the proposed framework could explore and eliminate uncertainty in special local areas, such as the pot lid that causes collision (middle) and the handle that is too large for grasping (bottom).

3 Real-world Demonstration Details

We use a Franka Emika Panda Robot Arm to perform exploration, and an Azure Kinect DK depth camera to obtain partial point clouds. To ensure the observed motion is the part's motion instead of the entire object's motion, we require the applied force of the robot arm to be smaller than the gravity. We use pulling as our action primitive since it's more challenging and interesting than pushing.

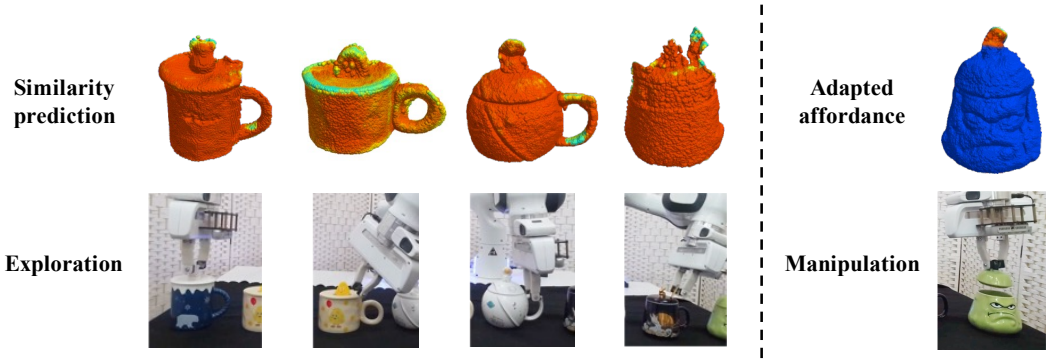

Figure 2: **Real-world Demonstration.** Our framework explores on four mugs under the guidance of similarity module, and then manipulates another unseen mug.

As shown in Figure 2, the five mugs are placed together in the workspace. We segment out the partial point could of each mug and use four of them as the input for exploration. After the few-shot learning (similarity prediction reaches a bar), we conduct manipulation on the left mug.

We have recorded the whole process of the above-described exploration via similarity, affordance adaptation and the final manipulation, please see the **introduction video** attached in supplementary.

4 Data Details

Following Where2Act [2] and AdaAfford [4], we use 959 articulated 3D objects covering 14 categories for training, few-shot learning, and testing, from the large-scale PartNet-Mobility dataset [5, 3]. The number of objects in different categories is shown in Table 2. We balance the interactions on each category by sampling more actions on categories that contains fewer objects. For example, the number of interactions on each faucet is the four times of that of cabinet.

We use approximately 81,000 interactions (on three categories) for training set, 550 interactions (on 11 categories) for few-shot set, and 77,000 interactions (on 11 categories) for testing set.

Cabinet	Faucet	Table	Window	Refrigerator	Switch	Kettle
						
345	84	101	58	44	70	29
Bucket	Trashcan	Microwave	Door	KitchenPot	WashingMachine	Box
						
36	70	16	36	25	17	28

Table 2: **Data Statistics.** Number of objects from different categories.

5 More Training Details

5.1 Hyper-parameters

We set batch size to be 32, and use Adam Optimizer [1] with 0.001 as the initial learning rate.

5.2 Computing Resources

We use PyTorch as our Deep Learning framework, and NVIDIA Tesla V100 (24GB GPU) for training and inference. Our model uses approximately 8GB memory during training.

It takes about 48 hours for training on 3 categories and 30 minutes for few-shot learning on 11 novel categories. During few-shot learning, most time is spent on collecting robot-object interactions.

5.3 Details of Few-shot learning

In each epoch of few-shot learning, the model is provided with partial point clouds of novel objects. The similarity module will propose interactions on selected objects, states and areas. The collected interactions will adapt the affordance module and the similarity module. The exploration will stop if the average score of similarity prediction exceeds a bar (0.9) or the interaction budget is reached.

When performing few-shot learning on a single category (a budget of 50 interactions in total), to avoid over-fitting to explored objects, a small portion of interactions on training categories (1% of the training data already seen by the model) are also added to the dataset during few-shot learning.

6 Code

We will release our code upon acceptance.

References

[1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *The 3rd International Conference for Learning Representations*, 2015.

[2] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021.

[3] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Yian Wang, Ruihai Wu, Kaichun Mo, Jiaqi Ke, Qingnan Fan, Leonidas J Guibas, and Hao Dong. Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 90–107. Springer, 2022.

[5] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020.