Where2Act: From Pixels to Actions for Articulated 3D Objects Supplementary Material

Kaichun Mo*[1] Leonidas Guibas[1] Mustafa Mukadam[2] Abhinav Gupta[2] Shubham Tulsiani[2]

[1]Stanford University [2]Facebook AI Research

https://cs.stanford.edu/~kaichun/where2act

A. Overview

This document provides more visualizations, results and more detailed descriptions accompanying the main paper. In summary, we include

- More details about the simulation framework and settings;
- More details about collecting interaction trials;
- Training details and computational timing;
- Visualization of the 15 object categories from SAPIEN [5] that we use in our work;
- More results on real-world data;
- More visualization of the actionability scoring predictions;
- More visualization of the action proposal predictions;
- Failure cases discussions and visualizations.

We also include a video clip (on the website) that present interactive demonstrations for the 6 types of action primitives to better illustrate the interaction dynamics and behaviors.

B. Framework and Settings: More Details

For our interactive simulation environment based on SAPIEN, we use the same set of simulation parameters for all interaction trials. We will release our simulation environment and full toolkits for the best reproducibility of our work and supporting future research. Besides the information provided in the main paper (Sec 5.1, **Environment**), we describe more detailed settings in our framework.

For general simulation settings, we use frame rate 500 fps, tolerance length 0.001, tolerance speed 0.005, solver iterations 20 (for constraint solvers related to joints and contacts), with Persistent Contact Manifold (PCM) disabled (for better simulation stability), with disabled sleeping mode (*i.e.* no locking for presumably still rigid bodies in simulation), and all the other settings as default in SAPIEN release. Following the SAPIEN suggested criterion, we also disable collision simulation between each articulated part to its direct parent node, due to usually omitted or inaccurate geometry modeling details at joint positions for ShapeNet [1] models.

For physical simulation, we use the standard gravity 9.81, static friction coefficient 4.0, dynamic friction coefficient 4.0, and restitution coefficient 0.01. For the object articulation dynamics simulation, we use stiffness 0 and damping 10. And for the robot gripper, we use stiffness 1000 and damping 400 for the free 6-DoF robot hand motion, while we use stiffness 200 and damping 60 for the gripper fingers.

For the rendering settings, we use an OpenGL-based rasterization rendering for the fast speed of simulation. We set three point lights around the object (one at the front, one from back-left and one from back-right) for lighting the scene, with mild ambient lighting as well. The camera is set to have near plane 0.1, far plane 100, resolution 448, and field of view 35°. For RGB image inputs, we downsample the obtained 448×448 images to 224×224 before feeding to the UNet backbone. For 3D partial point cloud scan inputs, we back-project the depth image into a foreground point cloud, by rejecting the far-away background depth pixels, and then perform furthest point sampling to get a 10K-size point cloud scan.

Each articulated object is approximated by convex hulls using the V-HACD algorithm [2] at the part level before simulation. The object is assumed to be fixed at its root part, with only its articulated parts movable. After loading each object to the scene and randomly initializing the starting articulated part poses, there are chances that the parts are not still due to the gravity or collision forces. Thus, we wait for 20K time steps to simulate the final rest part states until the parts are still for 5K steps, or this interaction is invalidated. We also remove interaction trials if the object parts have initial collisions, by detecting impulses bigger than 0.0001, due to unstable simulation outcomes.

*The majority of the work was done while Kaichun Mo was a research intern at Facebook AI Research.

C. Interaction Trials: More Details

In the main paper (Sec 5.1, **Action Settings**, the second paragraph), we detailedly defined our pre-programmed motion trajectories for the six types of action primitives. In the supplementary video, we further illustrate the interaction demonstrations in action. Below, we describe how the robot is driven to follow the desired motion trajectories and how to collect successful interaction trials.

The dynamics of the articulated objects and robot gripper is simulated using a velocity controller, equipped with the NVIDIA PhysX internal PID controller, that drives the gripper from one position to another, while the high-level trajectory planning is done by a simple kinematic-level computed interpolation between the starting and end end-effector poses with known gripper configurations. The robot gripper can be intialized as closed (perfectly touched) or open (0.08 unit-length apart).

For an interaction trial to be considered successful, it not only needs to cause considerable part motion along intended direction, as described in the main paper (Sec 5.1, **Action Settings**, the last paragraph), but has to be a valid interaction beforehand. First, the interaction direction should belong to the positive hemisphere along with the surface normal direction. Second, the robot gripper should have no collision or contact with the object at the initial state. Otherwise, we treat this interaction trial to be failed without simulation. Finally, for the *pushing* action primitives, we require that the first-time contact happens between the robot closed gripper and the target articulated part, to remove the case that the robot is pushing the other parts if multiple parts are very close to each other. It is also invalid if the robot hand, instead of the fingers, to first touch the part. We do not put this constraint for the *pulling* primitives, as the open gripper may touch the other parts first and then grasp the target part. For these invalid interactions, we mark them as false data points without measuring the part motion.

D. Training Details and Computational Timing

We use learning rate 0.001 and Adam optimizer. There is no image-based data augmentation. For 3D scans, we randomly down-sample point cloud inputs for augmentation. The input shapes are 224×224 for images and 10000×3 for point clouds. Each simulated interaction takes about 2-8s. With parallel computation, it takes 3-4 days to collect all offline interactions. The training takes 0.8s for each iteration and 4-5 days until convergence. As inference is a simple forward pass, it only takes 4ms to infer for a batch of 32 RGB/depth images.

E. Simulation Assets: Visualization

In Fig. 1, we visualize one example for each of the 15 object categories from SAPIEN [5] we use in our work.

F. More Results on Real-world Data

In Fig. 2, we visualize more results for directly applying our networks over real-world data.

G. Actionability Scoring Predictions: More Result Visualization

In Fig. 3, we visualize more example results of the actionability scoring module for the six types of action primitives.

H. Action Proposal Predictions: More Result Visualization

In Fig. 4, we visualize more action proposal predictions on example shapes for each action primitive.

I. Failure Cases: Discussion and Visualization

We present some interesting failure cases in Fig. 5. Please see the figure caption for detailed explanations and discussions. From these examples, we see the difficulty of the task. Also, given the current problem formulation, there are some intrinsically ambiguous cases that are generally hard for robot to figure out from a single static snapshot.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[2] Khaled Mamou, E Lengyel, and AK Peters. Volumetric hierarchical approximate convex decomposition. In *Game Engine Gems 3*, pages 141–158. AK Peters/CRC Press, 2016. 1

[3] Google Research. Garden swing. https://fuel.ignitionrobotics.org/1.0/GoogleResearch/models/GARDEN_SWING, 2020. 3

[4] Google Research. Rj rabbit easter basket blue. https://fuel.ignitionrobotics.org/1.0/GoogleResearch/models/RJ_Rabbit_Easter_Basket_Blue, 2020. 3

[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. 1, 2

[2] https://davidbaptistechirot.blogspot.com/2019/01/washing-machine-hd-png-images.html, https://www.amazon.in/SmartBuy-Collection-Selleck-Prescription-Eyeglass/dp/B083QS7N9R, http://www.allwhitebackground.com/storage-cabinets.html/download/142, https://www.colourbox.com/image/empty-aluminium-bucket-over-the-white-background-image-1709799

Figure 1. **Simulation Assets Visualization.** We visualize one example for each of the 15 object categories we use in our work.

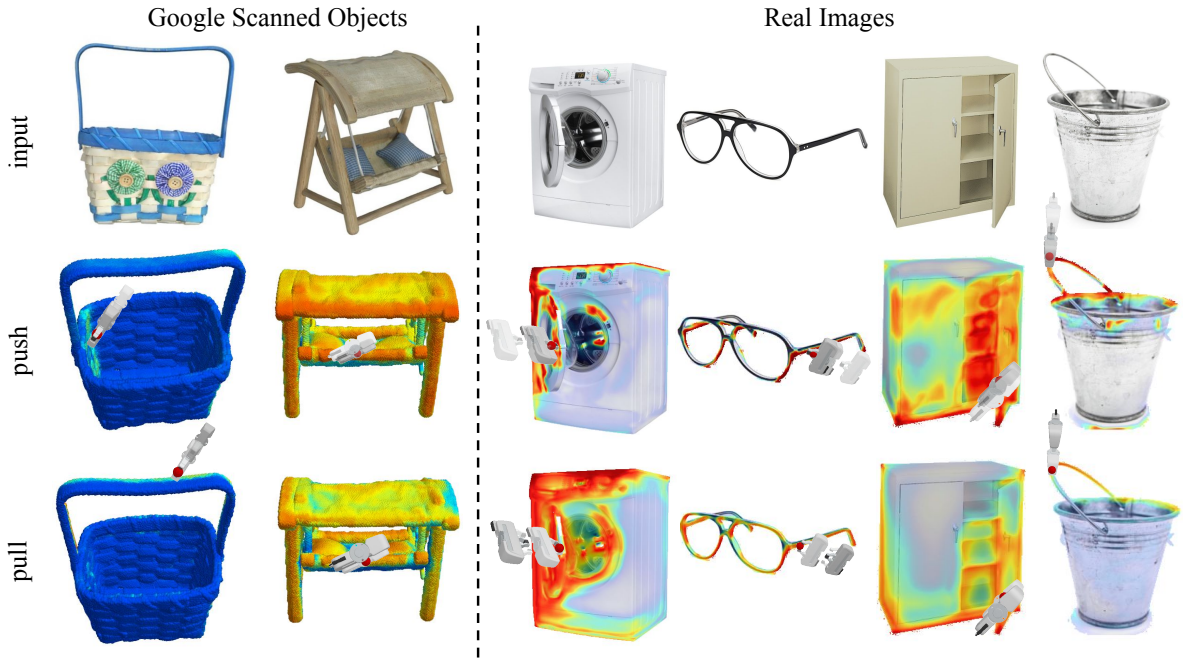

Figure 2. **More Results on Real-world Data.** We present more results on real-world data that augment Fig. 6 in the main paper. We use 3D real object scans from Google Scanned Objects [4, 3] and 2D real images from the web[2]. Here, results are shown over all pixels since we have no access to the articulated part masks. Though there is no guarantee for the predictions over pixels outside the articulated parts, the results make sense if we allow motion for the entire objects.

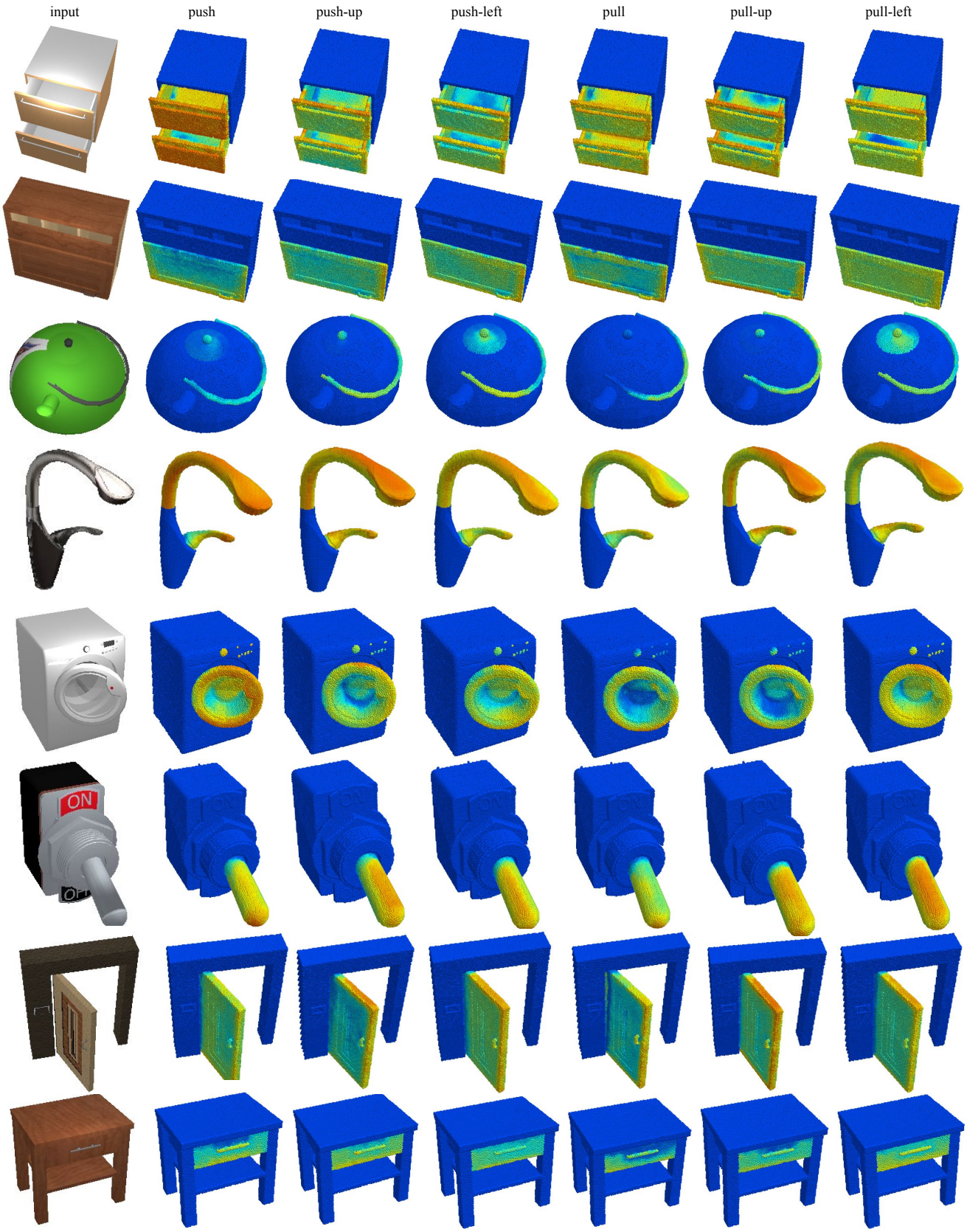

Figure 3. **Actionability Scoring Predictions.** We visualize more example predictions of the actionability scoring module for the six types of action primitives.

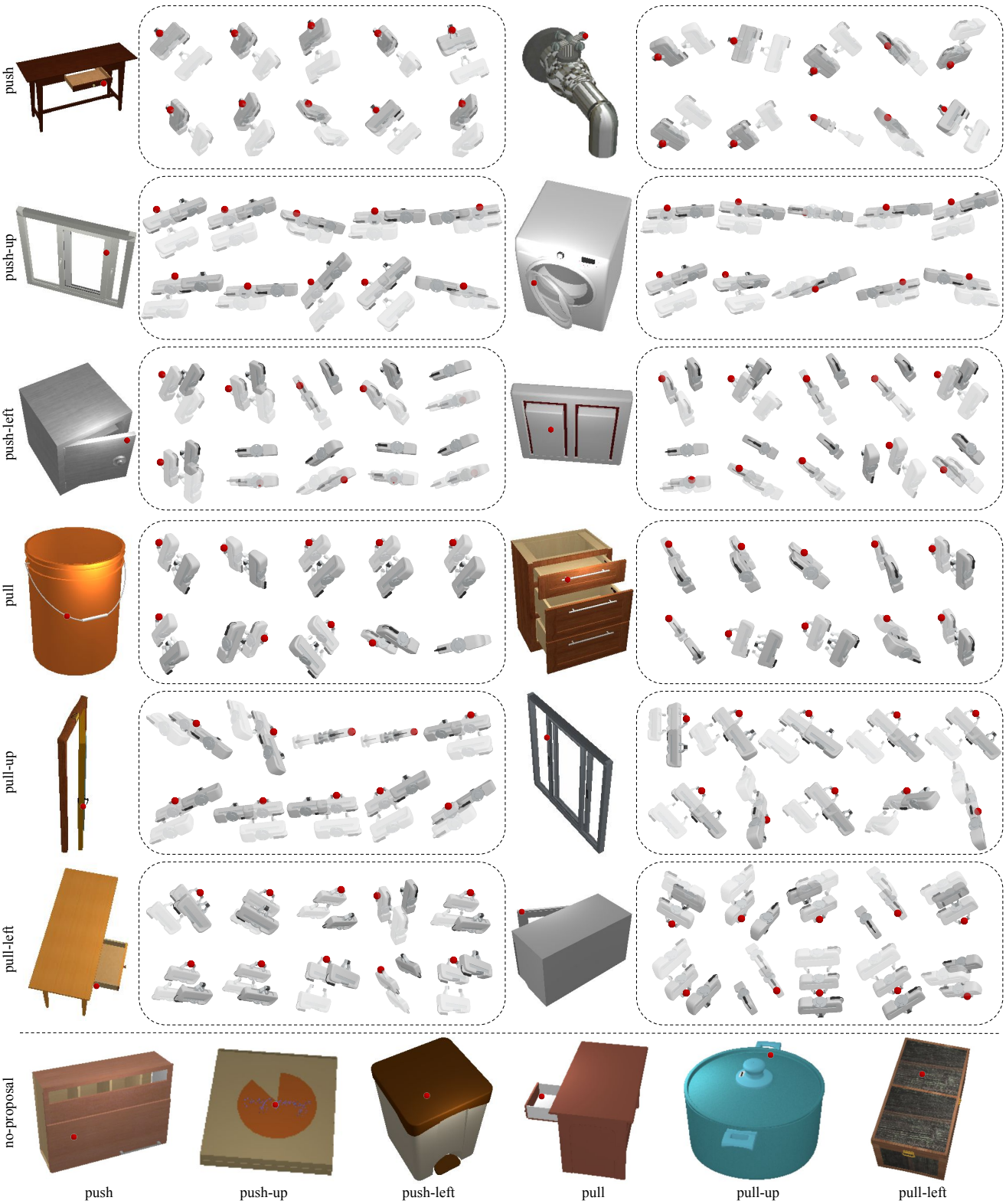

Figure 4. **Action Proposal Predictions.** We visualize the top-10 action proposal predictions (motion trajectories are 3× exaggerated) for some example testing shapes under each action primitive. The bottom row presents the cases that no action proposal is predicted, indicating that these pixels are not actionable under the action primitives.

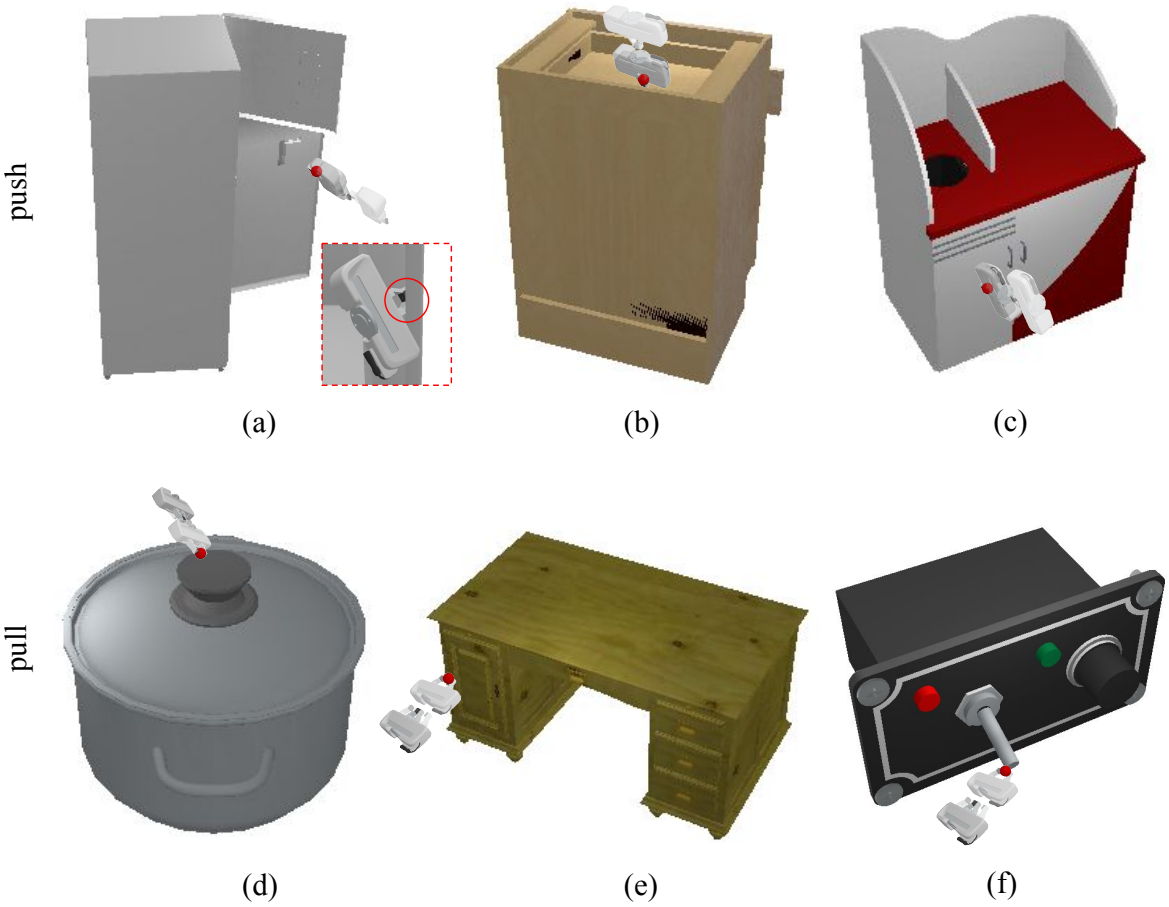

Figure 5. **Failure Cases.** We visualize some interesting failure cases, which demonstrate the difficulty of the task and some ambiguous cases that are hard for robot to figure out. For the *pushing* action, we show (a) an example of gripper-object invalid collision at the initial state, thus leading to failed interaction, though the interaction direction seems to be successful; (b) a failed interaction due to the fact that the part motion does not surpass the required amount 0.01 since the interaction direction is quite orthogonal to the drawer surface; and (c) a case that the door is fully closed and thus not pushable, though there are cases that the doors can be pushed inside in the dataset. For the *pulling* action, we present (d) a failed grasping attempt since the gripper is too small and the pot lid is too heavy; (e) a case illustrating the intrinsic ambiguity that the robot does not know from which side the door can be opened; and (f) a failed pulling attempt as the switch toggle already reaches the allowed maximal motion range.