Supplementary Materials for AKB-48: A Real-World Articulated Object Knowledge Base

Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Qiaojun Yu, Yang Han, Cewu Lu†
Shanghai Jiao Tong University
{liuliu1993, vinjohn, simon-fuhaoyuan, qiansucheng, yqjllxs, lucewu}@sjtu.edu.cn
tycoer@outlook.com

1. Overview

In this supplementary material, we provide an extra description about the Articulation Knowledge Base (AKB-48) in our paper, which consists of the fast articulation knowledge modeling (FArM) pipeline, category specification and dataset analysis in Section 2. In addition, we provide more details about the AKBNet in Section 3 and experiments in Section 4. Finally, more qualitative results are shown in Section 5.

2. Articulation Knowledge Base, AKB-48, Extended

2.1. Fast Articulation Knowledge Modeling (FArM) Pipeline, Extended

We build our own object recording system with 3D sensors, which is developed with three components: EinScan Pro 2020 for scanning[1], Intel RealSense D435 for RGB-D multi-view snapshot, multi-scale rotating turntables and lift bracket. In our setup, each object can be scanned within 5 minutes. **To solve the inner hole problem during scanning the objects**, we split all the real-world object into two groups: Firstly, all the parts of the articulated object could be disassembled, e.g. drawer rack and columns. We scan these parts separately and then manually combine them together. Secondly, the parts of the articulated cannot be disassembled, e.g. box and stapler. We scan them in a full open way and then manually segment them into several movable parts. During segmentation, there might be holes at the junction of parts. To deal with this problem, we fill the holes by their curvatures.

After model acquisition by our recording system, we a 3D user interface that allows the annotators can operate on 3D shapes. The FArM interface is illustrated in Fig. 1. Here the interface integrates the functions of object alignment, part segmentation and joint annotation.

Object Alignment. At the top of the interface, we provide 7 primitive shapes for object alignment, such as cube, sphere and cylinder. Each primitive shape holds it own (x, y, z) coordinate frame. The user can select one of them according to the basic shape of the input model. During alignment, the user is required to fit the selected primitive shape to the input model controlled by keyboard, mouse and assigning the pose directly. In our modeling tool, all the primitive shapes are parametric so the user can also re-scale them to achieve better fitting result. Finally, the alignment is completed when the user is able to perfectly fit the primitive shape into the input model.

Part Segmentation. Different from the Question-Answering system adopted in PartNet [6], we provide a mesh cutting method with multi-view observation. In part segmentation, the user is able to draw arbitrary 3D polygon in input scanned mesh/point cloud to annotate initial part segmentation. In addition, we also provide pre-defined part point cloud from similar geometric shapes, which could speed up part segmentation process.

Joint Annotation. when annotating joint properties, user could give prior information such as joint type, joint limit and joint axis. In addition, multi-view RGB-D images input are also supported for annotate joint. Our FArM tool provides to animate the initially annotated part segmentation and joint properties for video verification. The animation can be paused at any time for instant adjustment.

The annotated articulated objects are described with the widely used Unified Robot Description Format (URDF) [5], an XML file format to describe all elements of articulated objects with chain or tree structure, including joint properties and part meshes. The base link is the origin of the kinematic tree.

†Cewu Lu is the corresponding author. He is the member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, and Shanghai Qi Zhi Institute, China.

[1] https://www.einscan.com

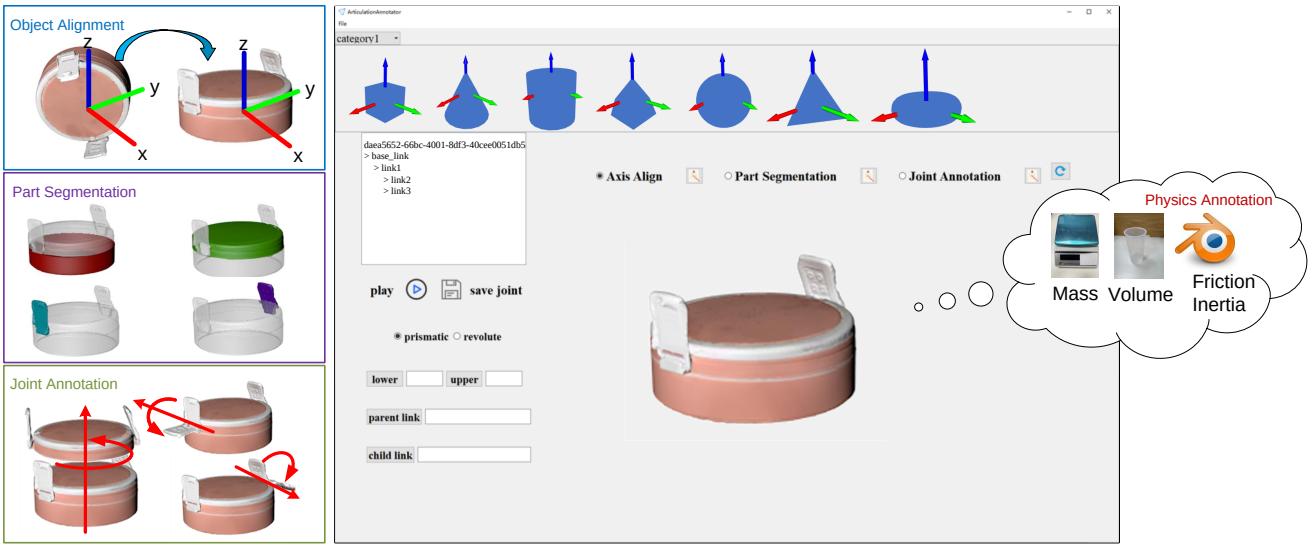


Figure 1. Our FArM interface. There are three sub-processes for articulation modeling: (1) object alignment, part segmentation and joint annotation. For physics annotation, we record per-part mass and volume in real-world, then compute inertia moment and friction using the mass and shape.

2.2. Dataset Analysis, Extended

Our AKB-48 dataset contains 48 categories and 2,037 shapes of articulated objects. Each of them provides rich appearance, semantics, structure and physics annotation. The total models can be accessed at https://liuliu66.github.io/AKB-48.

To further illustrate the advantages of AKB-48, we investigate the intra-variety in one category of our dataset. We define a shape distribution as metric to measure model shape variety. In detail, we extract Intrinsic Shape Signature (ISS) keypoints [10] from each model. The more ISS keypoints extracted, the more complicated shape is. Given these ISS keypoints, we compute the geometric distance between each keypoint pair, and then do frequency statistics for all the distances [7]. Finally, we project these histograms with t-SNE [9], as shown in Fig. 2. As it could be observed, the models in our AKB-48 hold a large shape variety in one category.

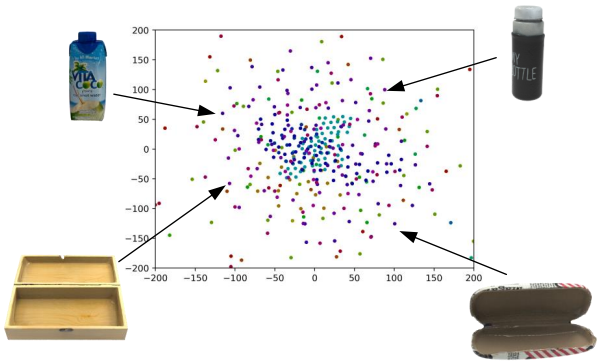


Figure 2. Object shape distribution: Visualization of t-SNE embedding of ISS histograms. A point stands for an instance and a color stands for a category.

3. AKBNet, Extended

3.1. Pose Module, Extended

Our pose module consists of two sub-modules: part sub-module and joint sub-module.

Part Sub-Module. Given the local point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ reconstructed from the input RGB-D image with detected bounding box, we use a PointNet++ [8] architecture to process the $\mathcal{P}$ for feature extraction. At the end of PointNet++, we build two parallel branches with $K+1$ and $3(K+1)$ channels for part segmentation S and per-part NOCS map [4] $\mathcal{P}' \in \mathbb{R}^{N\times 3}$ prediction, where K indicates the maximum number of parts and 1 indicates the background. The per-part NOCS map is defined for each separate rigid part on rest state. Finally, we could predict part segmentation label s_i and per-part NOCS coordinate $\mathbf{p}'_i$ on ith point.

Joint Sub-Module. Current methods such as A-NCSH [4] for the CAPE setting require fixed kinematic structure as prior knowledge. Joint sub-module aims to handle varied kinematic structures in one semantic category. Here we assume that all the parts and joints have one-to-one corre-

spondence. Therefore we can predict joint properties for each corresponding segmented part. The feature extractor is a PointNet++ architecture shared with part sub-module and we build three parallel branches for joint type classification, joint location prediction and joint axis prediction. Specifically, the joint type branch has 4 channels at the end of the MLPs, where we summarize four different joint types, including fixed, prismatic, revolute and screw. The joint location $\mathbf{q}^k$ on kth part is predicted by the heatmap and offset scheme, followed by A-NCSH [4]. The joint axis $\mathbf{u}^k$ on kth part is predicted by a voting scheme as:

$$\mathbf{u}^k = \frac{\sum_{j=i}^{N} \mathbf{u}_i^k \mathbb{1}(s_i = k)}{\sum_{i=1}^{N} \mathbb{1}(s_i = k)} \tag{1}$$

Specifically, we use the cross-entropy loss for part segmentation task $\mathcal{L}_{seg}$ and joint type classification $\mathcal{L}_{type}$. Then L2 is adopted as NOCS map loss $\mathcal{L}_{nocs}$, joint location $\mathcal{L}_{loc}$ and joint axis $\mathcal{L}_{ax}$ prediction tasks.

$$\begin{aligned}
\mathcal{L}_{seg} &= \sum_{i=1}^{M} CE(s_i, s_i^*) \\
\mathcal{L}_{nocs} &= \sum_{i=1}^{M} \mathbb{1}(s_i^* > 0) \|\mathbf{p}_i' - \mathbf{p}_i'^*\|_2 \\
\mathcal{L}_{loc} &= \sum_{i=1}^{M} \mathbb{1}(s_i^* > 0) \|\mathbf{q}_i - \mathbf{q}_i^*\|_2 \\
\mathcal{L}_{ax} &= \sum_{i=1}^{M} \mathbb{1}(s_i^* > 0) \|\mathbf{u}_i - \mathbf{u}_i^*\|_2 \\
\mathcal{L}_{type} &= \sum_{i=1}^{M} CE(\delta_i, {\delta_i}^*)
\end{aligned} \tag{2}$$

where $\mathbb{1}(s_j^* > 0)$ indicates the loss is only accounted for when the part is foreground. Finally, with predicted part segmentation, NOCS map and joint properties, we follow the pose optimization algorithm with kinematic constrains [4] to recover the 6D pose for each rigid part.

3.2. Manipulation Module, Extended

Physics Prediction Sub-Module. Apart from the reinforcement learning agent for manipulation, we also perform the physics prediction sub-module on the manipulation module of AKBNet. The input is the per-part feature vectors extracted from part segmentation results in the pose module. Then we train an extra 3-layer MLP with ReLU activation function and build three branches at the end of the layer, with 1, 6 and 1 channels respectively, in which 1 indicates the per-part mass m^k prediction, 6 indicates the per-part inertia moment $I^k = \{I_{xx}^k, I_{xy}^k, I_{xz}^k, I_{yy}^k, I_{yz}^k, I_{zz}^k\}$, and the other 1 indicates the per-part friction value μ^k. The training loss functions for these three branches are L2 loss.

4. Experiments, Extended

4.1. Experimental Setup, Extended

We use PointNet++ [8] to train our pose module and shape module. For optimizer, we adopt the Adam algorithm with an initial learning rate of 0.001 and batch size 16. The learning rate will drop by 0.7 at every 2 and 4 epochs on the pose module and shape module. Dropout 0.5 is adopted.The total training epochs are 50 and 100. During training data pre-processing, we adopt to down-sample the input point cloud with voxel size 0.005 and then randomly sample 2048 points for each instance. For data split, we use 80% of objects for training and 20% for testing. Our model is implemented on PyTorch and 4 TITAN RTX GPUs.

We use a 3-layer MLP with ReLU activation function for feature extraction on manipulation module and the number of the final channels is 4 that corresponds to 4 actions. The total training step is 1e6. Batch size is 512. Learning rate is 0.001. The optimizer is Adam.

4.2. Manipulation Module Performance, Extended

The learning curves of Reinforcement agent on opening and pulling tasks using TQC [3]+HER [1] and SAC [2]+HER [1] are illustrated in Fig. 3. We train 68 and 32 instances from AKB-48 to train the two RL algorithms with different random seeds, with each performing one evaluation rollout every 1000 environment steps. The solid curves correspond to the mean and the shaded region to the minimum and maximum returns over the five trials.

To validate the effect of physics in our AKB-48 for manipulation, we build an experiment that drives a robot arm (Franka Emika Panda[2]) to grip one of the instances in simulation. We also use a Franka Panda Arm to grip the corresponding real-world object. As illustrated in Fig. 4, we record the force feedback during the robot arm gripping the object. As it can be seen, with the physics information, the force feedback in simulation shows to be similar to the force curve in the real world. Therefore, we can conclude that the physics information annotated in AKB-48 is of importance in robotics research.

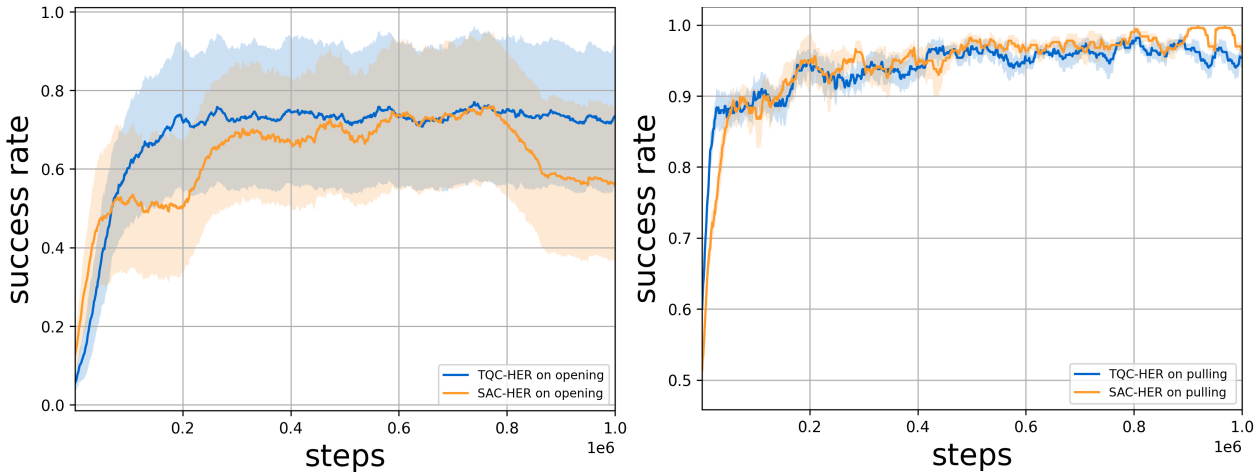
5. Qualitative Results, Extended

Qualitative results of pose module, shape module and manipulation module of AKBNet are illustrated in Fig. 5, Fig. 6 and Fig. 7.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight

[2] https://www.franka.de/



(a) Performance on opening task.

(b) Performance on pulling task.

Figure 3. Learning curves on opening and pulling manipulating tasks. We use SAC+HER and TQC+HER to train the Reinforcement Learning agent.

experience replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5055–5065, 2017. 3

[2] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018. 3

[3] Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, pages 5556–5566. PMLR, 2020. 3

[4] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3706–3715, 2020. 2, 3

[5] W Meeussen, J Hsu, and R Diankov. Unified robot description format (urdf), 2009. 1

[6] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019. 1

[7] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, 2002. 2

[8] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++ deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5105–5114, 2017. 2, 3

[9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2

[10] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 689–696. IEEE, 2009. 2

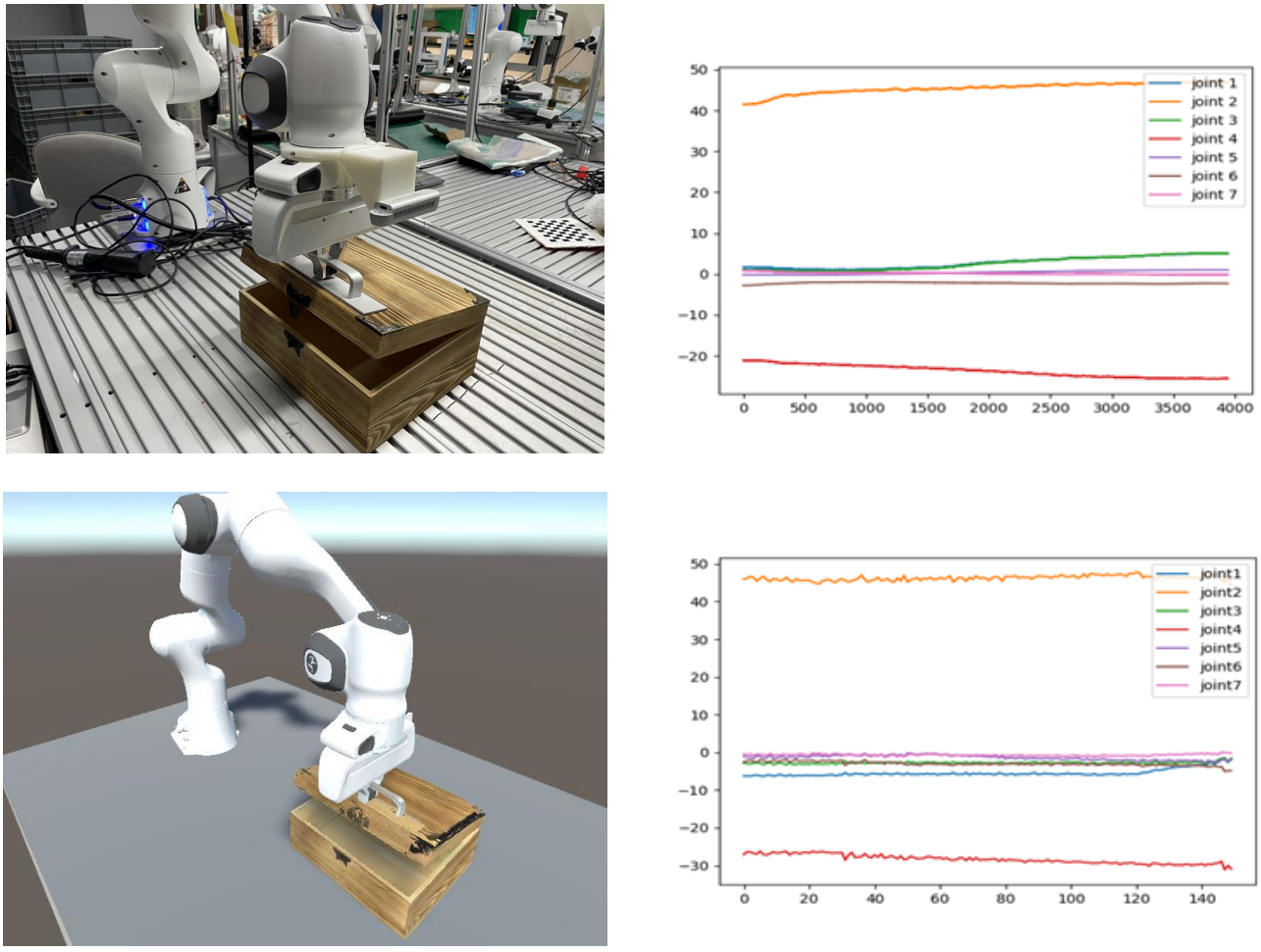


Figure 4. Force feedback during the robot arm gripping the object. We compare the force curves in the real world and the simulation with our predicted physics information. The robot arm is Franka Emika Panda.

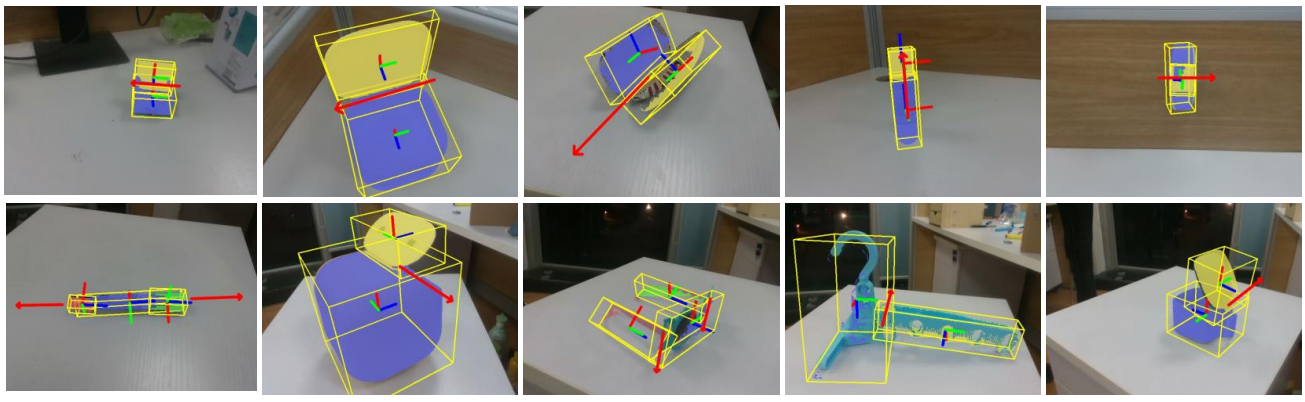


Figure 5. Qualitative results on pose module.

Figure 6. Qualitative results on shape module.

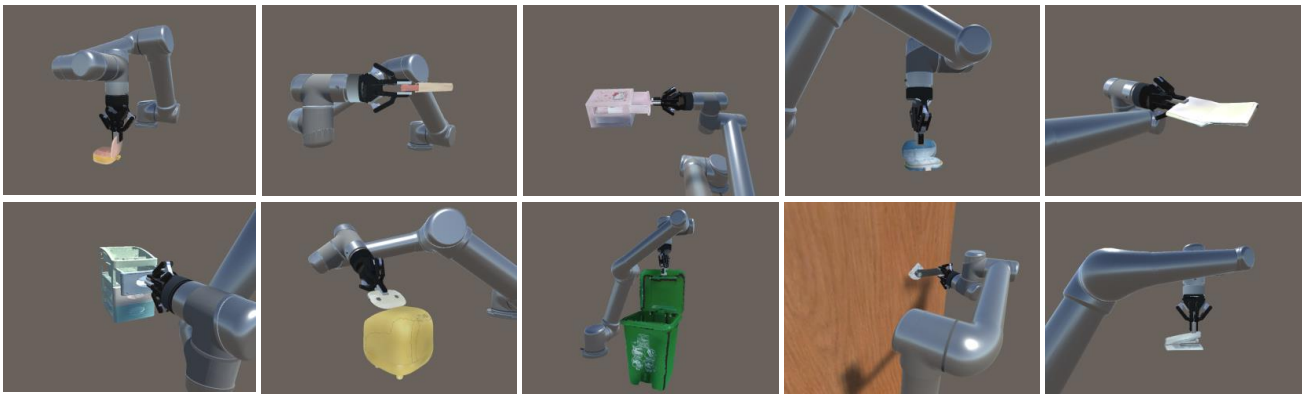


Figure 7. Qualitative results on manipulation module.