3D Implicit Transporter for Temporally Consistent Keypoint Discovery: Appendix

1. Implementation Details

We implement our models in PyTorch [2] with the Adam [1] optimizer and a mini-batch size of 10 on 4 NVIDIA A100 GPUs for 45 epochs. A learning rate of 10^{-4} is used for the first 30 epochs, which is dropped ten times for the remainder. To increase data diversity, we perform random rigid transformation and Gaussian noise for input point clouds. Perception and manipulation hyper-parameters are provided in Tab. 1.

Table 1. Perception and manipulation hyper-parameters.

N_1	N_2	$C_1/C_2/C_3/C_5/C_e$	C_4	$C_h/C_w/C_d$	λ_1/λ_2	θ_{thr}	λ
5000	128	32	256	64	1	0.1	8

Fig. 1 depicts a series of training examples comprising rendered RGB images, accompanied by corresponding point clouds, for an articulated object with motion in its constituent parts. It is pertinent to note that solely the rendered point clouds are utilized for training and testing purposes.

2. Ablation Study

Keypoint Parameters Tab. 2 provides the quantitative results on keypoint parameters. Increasing keypoint number m and Gaussian variance σ can transport more features from target to source so that the reconstruction performance improves. However, few keypoints are enough for objects with relatively small mobile parts to transport core features. In this case, the redundant keypoints may scatter on stationary parts, which could harm pose estimation on mobile parts. For our training data, $m = 6$ and $\sigma = 0.15$ are the best choice.

Volume Size We have conducted an ablation study of the impact of the volume size. The results are reported in Tab. 3. The higher volumetric resolution of feature grids improves keypoint detection performance but increases the computation cost. Therefore, we choose the voxel size of 64 to balance the memory cost and perception performance.

Table 2. Ablation study of parameters of keypoint network.

m	σ	**RR** ↑	**ACKD** ↓	**ADD** ↓
5	0.15	0.623	0.130	0.129
6	0.10	0.604	0.136	0.130
6	0.15	0.611	**0.120**	**0.109**
6	0.20	**0.601**	0.128	0.122
7	0.15	0.606	0.125	0.113
8	0.15	0.551	0.141	0.130

Table 3. The impact of volume size.

$C_h/C_w/C_d$	**RR** ↑	**ACKD** ↓	**ADD** ↓
16	0.567	0.147	0.153
32	**0.642**	0.125	0.123
64	0.611	**0.120**	**0.109**

3. Formulation of the Additional Loss

As discussed in the main paper, we incorporate an additional loss term, $\mathcal{L}_{\text{occ\_s}}$, to facilitate the source frame reconstruction for better perception results.

Via the volume features $\Phi(\mathbf{o}_s)$ of the source frame and the corresponding query set, the geometry decoder is required to predict the occupancy of the source frame, which is written as:

$$\Omega(\mathbf{q}_e, \Phi_{\mathbf{q}}(\mathbf{o}_s)) \rightarrow \text{Prob}(\mathbf{q}|\mathbf{o}_s) \quad (1)$$

Then, we use the binary cross-entropy loss to assess the dissimilarity between the decoded and the priori specified occupancy values by:

$$\mathcal{L}_{\text{occ\_s}} = \frac{1}{|Q|} \sum_{\mathbf{q} \in Q} l_{\text{BCE}} \left(\text{Prob}(\mathbf{q}|\mathbf{o}_s), \text{Prob}^{\text{gt}}(\mathbf{q}|\mathbf{o}_s) \right) \quad (2)$$

4. Training Efficiency

We compare our training time cost with UMPNET. Fig. 2 presents a comparative analysis of the training time cost required to achieve the best performance of both UMPNET and our proposed model, utilizing the same hardware (an Nvidia A100 GPU). The presented results demonstrate the

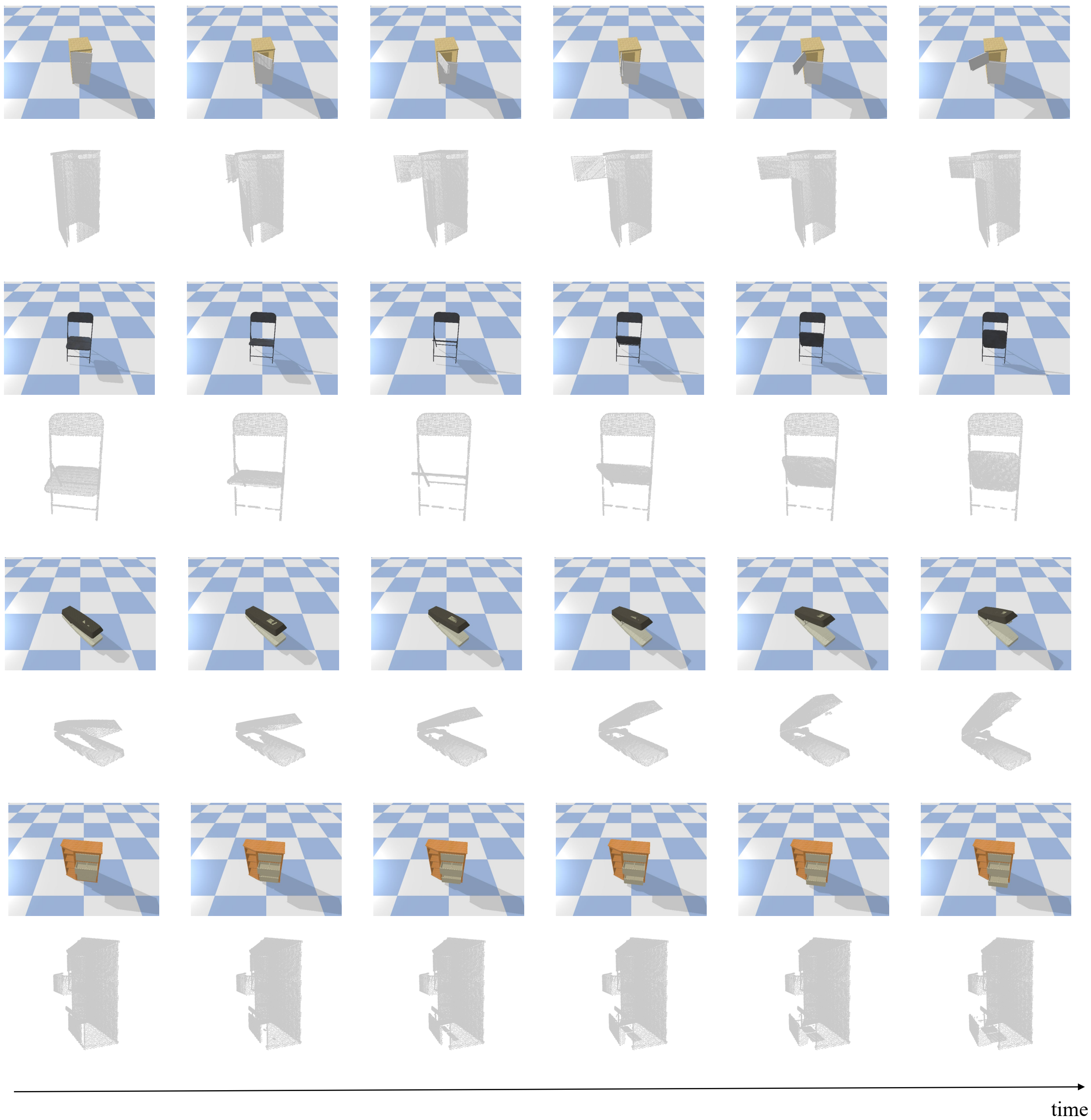

Figure 1. Examples of sequences of training data.

superior efficiency of our 3D Implicit Transporter. Our belief is that this can be attributed to the more efficient nature of sparse keypoint learning as opposed to dense affordance prediction.

5. Qualitative Results

Keypoint Consistency We show more qualitative results on keypoint temporal consistency between the same instance with different articulated states in Fig. 4. It can be seen that our method can generate more consistent keypoints than other baselines in both revolute and prismatic joints. We also provide visualizations of real objects in Fig. 6. Since the real depth image often contains artifacts caused by occlusions, depth discontinuities, or multiple reflections, we adopt the filter method as [3] used to fill holes in depth image and smooth depth values. Nevertheless, the point clouds may still be incomplete. Despite these artifacts, our method can generally detect spatiotemporally consistent keypoints. We believe the reason is that the implicit geometry decoder can represent the surface oc-

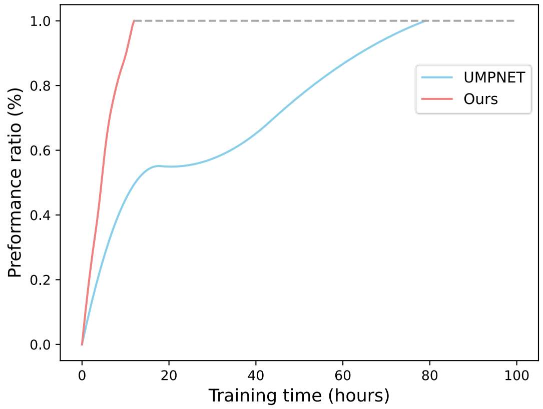

Figure 2. The training time (in hours) required to achieve the best performance of both UMPNET and ours.

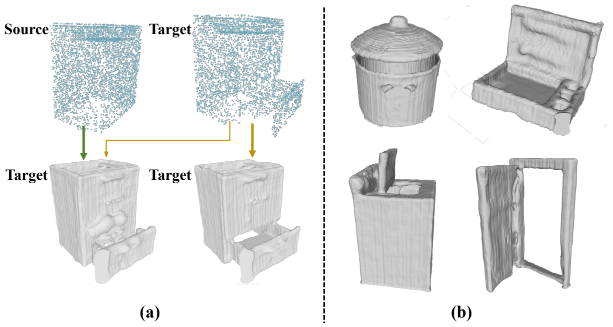

Figure 3. Surface shape reconstruction. (a) Target shape reconstruction from transported source features and target features, respectively. (b) Reconstruction results of unseen categories.

cupancy in each continuous input query point, which is robust to the density variation of point clouds. More intuitive performance can be found in the supplementary video.

Implicit Reconstruction Fig. 3-(a) shows the surface reconstruction of the target input, which is based on the transported feature from the source. It demonstrates the effectiveness of the feature transporter and the implicit geometry decoder. Fig. 3-(b) provides more reconstruction results of unseen test categories.

Goal-conditioned Manipulation As shown in Fig. 5 (simulation) and Fig. 7 (real scene), we show visualizations of the closed-loop policy taken to interact with articulated objects from their initial to goal states. More qualitative results in simulated and real scenes can be found in the supplementary video.

6. Failure Cases

If the number of points of the mobile part is too small, it is difficult for our method to detect accurate keypoints. Moreover, our manipulation strategy fails when the attached keypoint is not on the object's surface, like the point in the drawer.

References

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 1

[2] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[3] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *CVPR*. 2

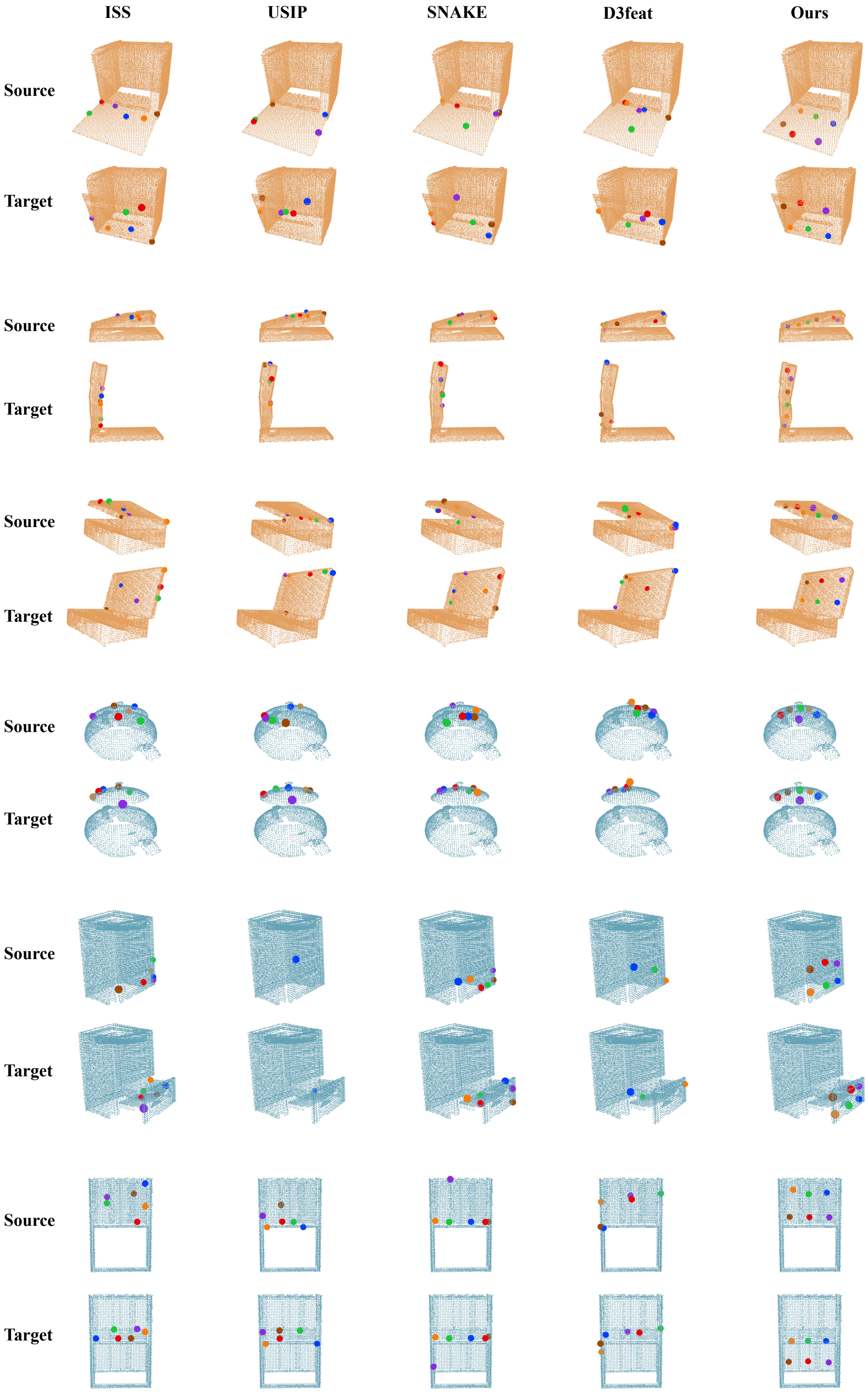

Figure 4. Keypoint temporal consistency comparison for both revolute and prismatic joints.

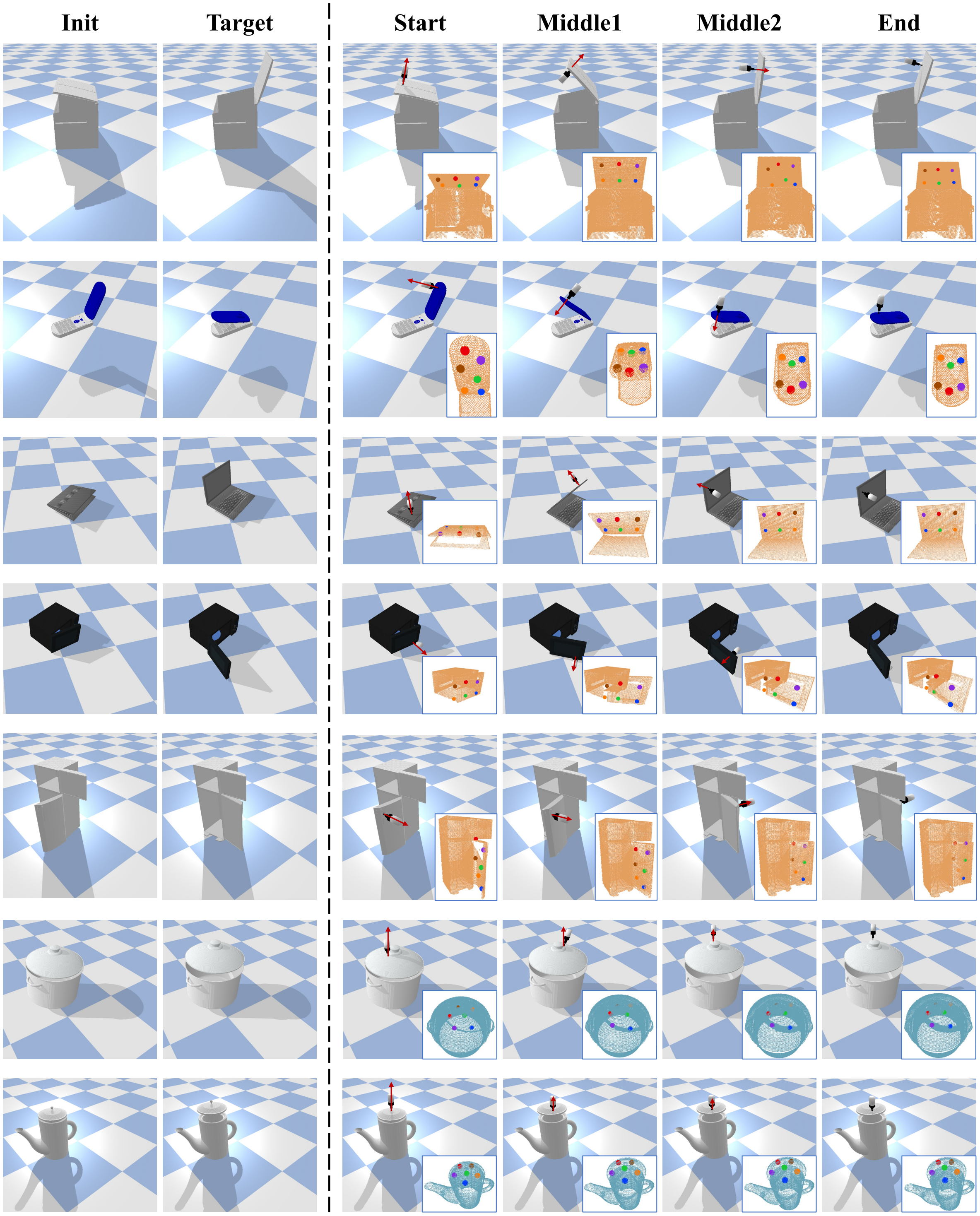

Figure 5. Visualizations of our closed-loop policy for manipulating articulated objects from initial to target states.

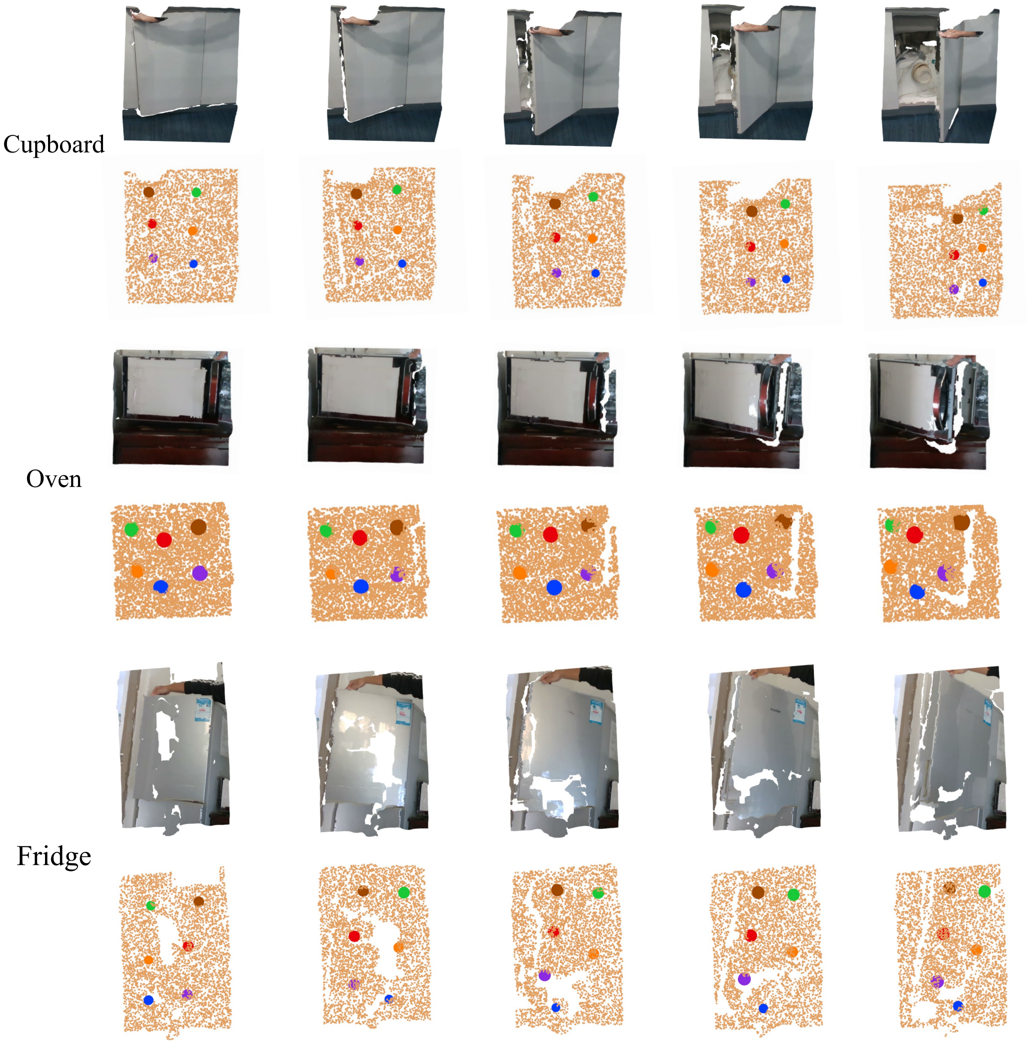

Figure 6. Keypoint consistency of real objects. The input point clouds are cropped by the human labeled bounding box in the first frame of an object video.

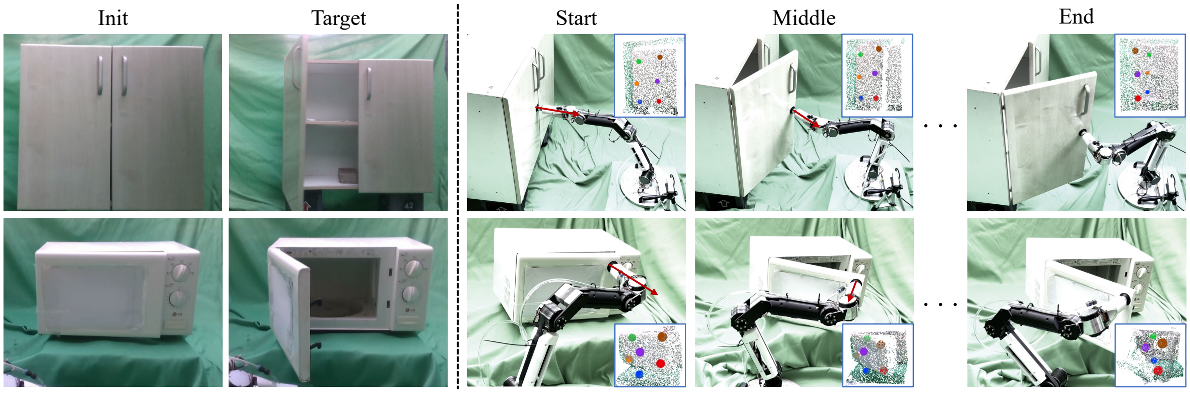

Figure 7. Qualitative results on real object manipulation.