1. Additional Implementation Details

1.1. Details on Common Affordance Distillation

As mentioned in Section 3.3, we use L_{2D-3D} to maintain 2D 3D features consistency to force the 2D and 3D features to be consistent. As depicted in Figure 1, we use E^{2D} to extract 2D features, and E^3D to extract 3D features. We use one linear layer as the domain adaptor to project 512 to d to project the 2D pixel-level features from the dimension of 512 to d, and one convolution layer as the semantic adaptor to down-sample the feature maps from the size of $H_3 \times W_3 \times 16$ to $H_2 \times W_2 \times 16$. In our model, we set $N = 20000$, $H_1 = 224$, $H_2 = 112$ and $H_3 = 223$.

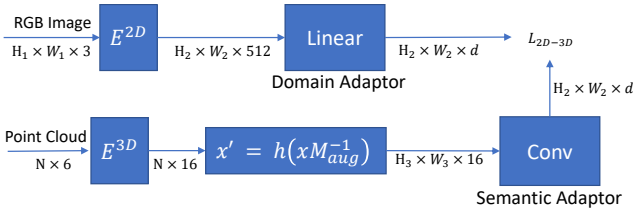

Figure 1. **Common Affordance Distillation**. **Detailed Architecture of the common affordance distillation module.** We use the Domain Adaptor to adapt 2D pixel-level features to the domain of object-part images, and the Semantic Adaptor to integrate semantic information distilled from the 2D branch into 3D point-level features.

1.2. Details on Part Instance Segmentation

The proposed actionable part perception network shares a similar architecture with PointGroup [3], and we refer readers to the original PointGroup paper for further details. In our approach, we found that setting the cluster radius to 0.03 and the cluster point number threshold to 5 yielded good segmentation results in on the GAPart [2] dataset.

The input point cloud P is initially voxelized into $100 \times 100 \times 100$ voxel grids. The backbone Sparse UNet comprises an encoder and a decoder, both with a depth of 7 (with channels of [16, 32, 48, 64, 80, 96, 112]), and outputs a point-wise feature F^{3D} with z channels, where $z = 16$. Following grouping, each mask proposal M_i is normalized, voxelized into a $50 \times 50 \times 50$ voxel grid, and then passed through the Scoring module. This module consists of a 2-depth UNet (with channels of [16, 32]) for point-wise feature extraction, an ROI Pooling layer for foreground feature merging, and a linear layer for confidence score S_i prediction.

During inference, points with binary classification scores below 0.4 are filtered out as background, and proposals with fewer than 5 points or a score lower than 0.09 are discarded. Finally, Non Maximum Suppression (NMS) with an IoU (Intersection over Union) threshold of 0.3 is applied to obtain the final segmentation masks.

1.3. Details on Part Pose Estimation

Following GapartNet [2], for each segmentation mask M_i, the point-wise feature $F^{3D}_{M_i}$ derived from F_{3D} undergoes processing via the part-oriented pose prediction network PE, which is composed of a 2-depth UNet (with channels of [16, 32]) and three Multilayer Perceptrons (3-MLP) for point-wise NPCS prediction. It should be noted that in practical implementation, we employ 9 distinct groups of 3-MLP to forecast NPCS coordinates across 9 channels, with supervision solely applied to the channel aligned with the ground truth semantic label. Since each part class has different symmetry patterns, they should be handled case by case. We also design symmetry-aware NPCS loss L_{NPCS} similar to GAPartNet [2].

During inference, upon receiving a predicted 3D part mask along with its NPCS map, we employ RANSAC [1] for outlier removal and utilize the Umeyama algorithm [6] to estimate the 7-dimensional rigid transformation. The process of joint parameter prediction is streamlined due to the unified definition of GAParts. Once the bounding box for each part is estimated, we can directly calculate the joint parameters by leveraging the definition of the GAPart. For instance, given the bounding box of a slider button, we can directly query its prismatic joint parameter, which aligns with the z-axis in the part canonical space.

1.4. Details on Data Preparation

We utilize the SAPIEN 2.0 environment [7] to generate a comprehensive dataset from the GAPartNet objects, encompassing partial point clouds, part semantic segmentation masks, part instance segmentation masks, NPCS maps, and part pose annotations, covering all requisite data for the proposed part segmentation, part pose estimation, and part-based object manipulation tasks.

Additionally, we introduce randomization in the articulated objects' joint poses and select camera positions within reasonable perspectives. Specifically, we manually define the range of camera positions for each object category to ensure favorable views of each object, avoiding perspectives from the back of a StorageFurniture, from beneath an Oven, or from excessively far or close angles. Furthermore, we apply random ambient light dimming within the range of [10%, 90%] and introduce random camera rotations within $\pm 5°$. The output image resolution is set to 800×800. For each object, we render 32 RGB images, along with segmentation masks and depth images using the built-in features of the SAPIEN environment. Additionally, we compute NPCS maps and oriented tight bounding boxes as part pose annotations for all GAParts.

Leveraging camera intrinsics, 2D RGB images, and depth images, we perform back-projection to obtain dense, partial point clouds. Subsequently, we sample 20,000 points for each dense point cloud using Farthest-Point-Sampling

(FPS). During the point cloud sampling process, we also generate corresponding ground truth annotations for semantic segmentation, instance segmentation, and NPCS maps. These 20,000-point point clouds and their annotations are precomputed offline to expedite subsequent 3D tasks.

1.5. Object Categories and Part Categories

13 Seen Object Categories. Box, Bucket, Camera, CoffeeMachine, Dishwasher, Keyboard, Microwave, Printer, Remote, StorageFurniture, Toaster, Toilet, WashingMachine.
10 Unseen Object Categories. Door, KitchenPot, Laptop, Oven, Phone, Refrigerator, Safe, Suitcase, Table, TrashCan.
9 Part Categories. Round Fixed Handle, Line Fixed Handle, Hinge Handle, Slider Button, Hinge Knob, Slider Drawer, Hinge Door, Hinge Lid, Slider Lid.

1.6. Training Procedure

Our model is trained in an end-to-end manner with maximum training epochs of 400 with an early stopping strategy. The whole training procedure takes around 31 hours on two 32G NVIDIA V100 GPUs. In order to prevent overfitting, we use position jitter, color jitter, random rotation, and random flip to make random enhancements to the point cloud to improve performance.

2. More Results of Part Segmentation

We visualize more results of part segmentation and part pose estimation in Figure 3(intra-test) and Figure 4(outer-test). We also take 64 as the feature dimension in the final model to get better performance and cost more time. The results are shown in 1.

3. More Results of Generated Images

To maintain consistency between the generated image and the original RGB image domain, we utilized ControlNet [8] and utilized depth maps sampled from GAPartNet [2] along with prompts to generate the image. This approach ensures that the generated image closely resembles the original, reducing the introduction of additional noise and distortion. The method guarantees that the generated image maintains consistency in color and texture with the original image, while also preserving its accuracy and clarity. We visualize more results of generated images in Figure 2.

4. More Details on Part-based Object Manipulation

For the 9 part categories, we used the same interaction policy as GapartNet [2]. We established our interaction environment using the SAPIEN [7] simulator, which was adapted from the ManiSkill challenge [5]. Our method was tested on four tasks: using a single Franka gripper to **open a drawer, open a door, manipulate a handle, and press a button**. These tasks showcase robot manipulation while adhering to the motion constraints of prismatic or revolute joints.

To evaluate our method, we randomly selected unseen objects containing doors, drawers, handles, and buttons from seen object categories. We considered the limitation of the single gripper and chose objects for which their segmentation and pose ground truth allowed for successful opening using our heuristics within our benchmark setting. Additionally, we assessed the cross-category generalizability of our method by randomly selecting unseen objects.

The success of these four tasks is defined as manipulating the part for 90% of the motion range within 1,000 steps with a stable stop at the end. We use 20 objects from seen categories and 20 from unseen categories to construct our benchmarks, respectively. We primarily compare our method with Where2act [4], ManiSkill [5], and GAPartNet [2]. The quantitative results of the simulation experiments are presented in Table 2. Our method demonstrates significant performance improvements over the baselines across all 4 tasks, indicating strong generalizability and validating the effectiveness of our part-pose-based manipulation policy.

dim	intra-test			outer-test		
	mAP	AP50	mIoU	mAP	AP50	mIoU
32	61.9	74.0	75.3	**30.6**	**38.7**	**35.1**
64	**63.8**	**75.3**	**80.9**	30.1	**38.7**	33.7

Table 1. **We use 64 as the feature dimension in our final model**.

References

[1] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, page 381–395, Jun 1981. 1

[2] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7081–7091, 2023. 1, 2, 3

[3] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020. 1

[4] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. 2, 3

[5] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill:

Success Rate(%)	Drawer		Door		Handle		Button	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
Where2act [4]	69.9	54.5	44.4	18.2	78.7	49.2	82.2	80.9
ManiSkill [5]	32.9	26.6	27.8	28.3	53.9	42.1	65.5	54.5
GAPartNet [2]	**95.0**	**90.0**	70.0	55.0	**90.0**	85.0	**100.0**	**95.0**
DisCo(Ours)	90.0	80.0	**80.0**	**75.0**	**90.0**	**90.0**	95.0	**95.0**

Table 2. **Results for Cross-category Object Manipulation in SAPIEN Simulator [7].**

Figure 2. **Visualization of Generated Images**. Below each image is a corresponding text prompt for generation.

Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021. 2, 3

[6] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 376–380, Apr 1991. 1

[7] 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, 3

[8] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2

Figure 3. **Part instance segmentation and pose estimation results on the intra-test**. “pc”, “pred”, “sem”, “gt”, “ins”, “bbox” stand for point cloud, prediction, semantic segmentation, ground truth, instance segmentation, prediction bounding box.

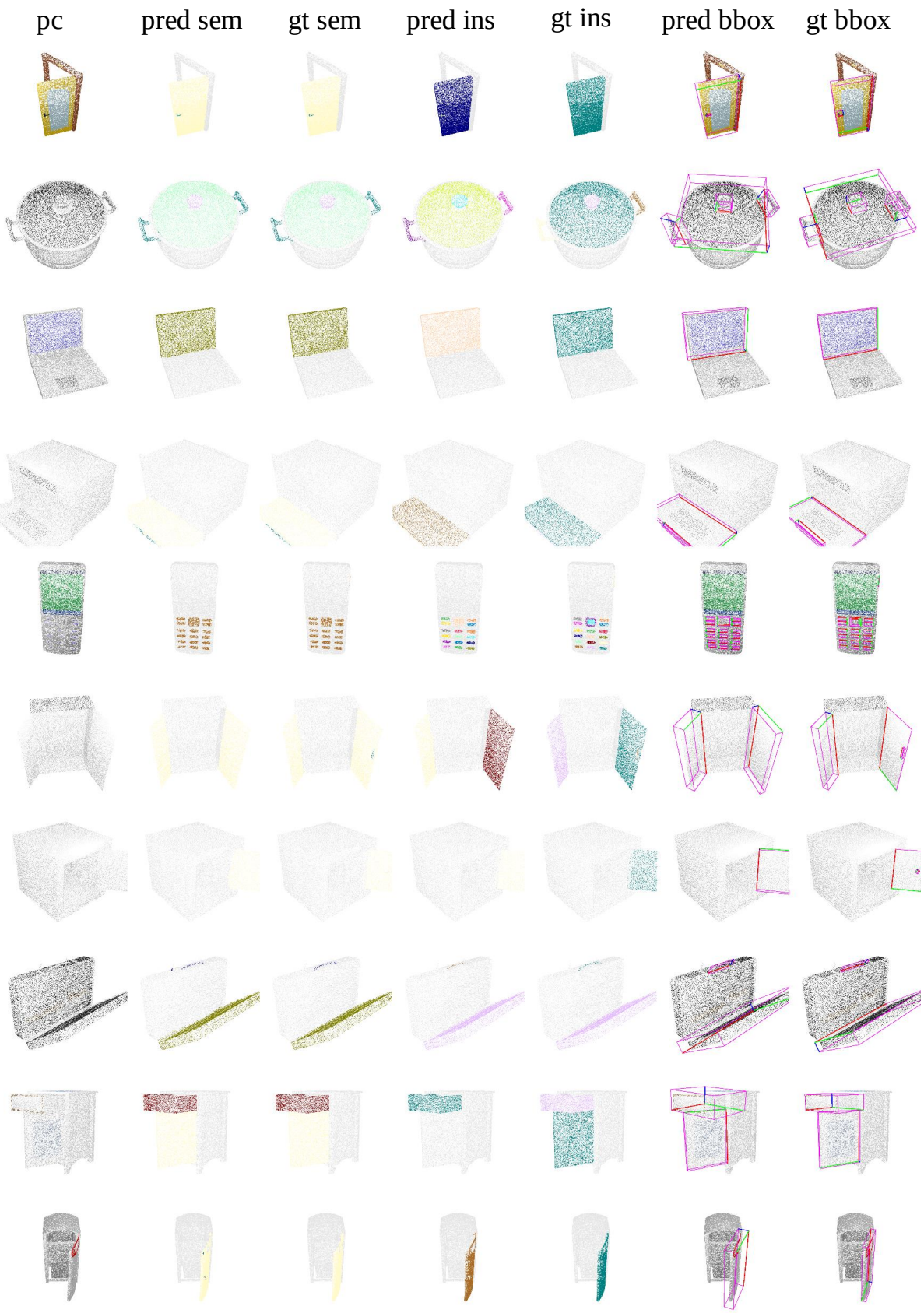

Figure 4. **Part instance segmentation and pose estimation results on the outer-test**. “pc”, “pred”, “sem”, “gt”, “ins”, “bbox” stand for point cloud, prediction, semantic segmentation, ground truth, instance segmentation, prediction bounding box.