2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3503-7770-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/IROS58592.2024.10802368

RPMArt: Towards Robust Perception and Manipulation for Articulated Objects

Junbo Wang[1], Wenhai Liu[1], Qiaojun Yu[1], Yang You[2], Liu Liu[3], Weiming Wang[1] and Cewu Lu[1]*

***Abstract*— Articulated objects are commonly found in daily life. It is essential that robots can exhibit robust perception and manipulation skills for articulated objects in real-world robotic applications. However, existing methods for articulated objects insufficiently address noise in point clouds and struggle to bridge the gap between simulation and reality, thus limiting the practical deployment in real-world scenarios. To tackle these challenges, we propose a framework towards Robust Perception and Manipulation for Articulated Objects (RPMArt), which learns to estimate the articulation parameters and manipulate the articulation part from the noisy point cloud. Our primary contribution is a Robust Articulation Network (RoArtNet) that is able to predict both joint parameters and affordable points robustly by local feature learning and point tuple voting. Moreover, we introduce an articulation-aware classification scheme to enhance its ability for sim-to-real transfer. Finally, with the estimated affordable point and articulation joint constraint, the robot can generate robust actions to manipulate articulated objects. After learning only from synthetic data, RPMArt is able to transfer zero-shot to real-world articulated objects. Experimental results confirm our approach's effectiveness, with our framework achieving state-of-the-art performance in both noise-added simulation and real-world environments. Code, data and more results can be found on the project website at `https://r-pmart.github.io`.**

I. Introduction

Human life is populated with articulated objects, ranging from household appliances such as microwaves and refrigerators, to storage units such as safes and cabinets. Robust perception and manipulation for those objects by robots in the real world can liberate humans from mundane daily tasks. Composed of more than one rigid part connected by joints allowing rotational or translational movements, articulated objects own high degree of freedom and large state space, which makes the visual perception and downstream manipulation challenging [1]. However, such geometric structure and physical constraints also provide useful clues for their perception and manipulation (see Fig. 1 (b)).

Recently, with the development of deep learning, substantial efforts have been devoted to studying the perception and manipulation for articulated objects. Prior works adapted powerful point cloud processing networks to estimate the kinematic articulation structures and parameters [2–5], and leveraged them to produce corresponding action trajectories [6, 7]. Another line of works explored manipulation tasks by directly imitating end-to-end demonstrations or reinforcement learning [8, 9]. Despite their success, building robust and reliable robots to manipulate articulated objects within noisy observations in the real world has not yet been investigated well. To achieve this goal, two primary challenges need to be addressed. (i) Point clouds from the real world are often noisy due to bad lighting and depth camera measurement error, while real-world articulation datasets are rare and always expensive to acquire. As a result, it is essential to introduce sim-to-real techniques to bridge the gap when training only on synthetic data. (ii) Articulated object manipulation involves both semantic and physical requirements. Grasping the relevant part requires semantic understanding of the object, and the action space

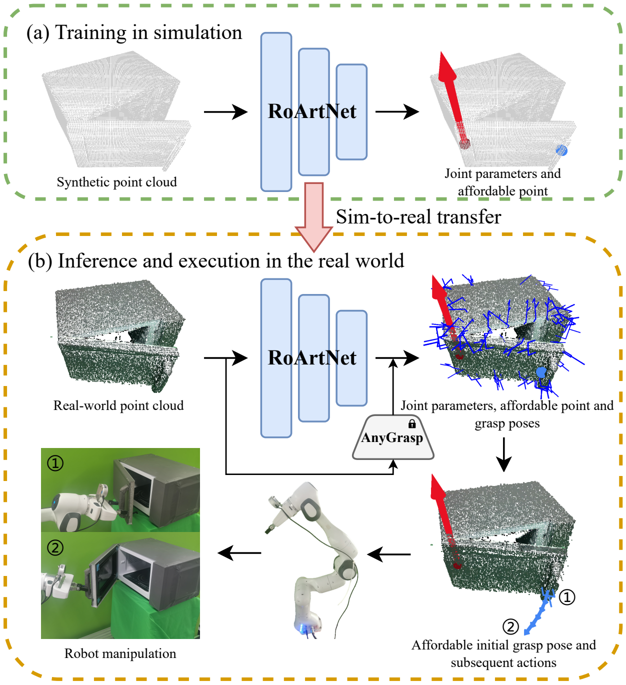

(a) Training in simulation
Synthetic point cloud
RoArtNet
Joint parameters and affordable point
Sim-to-real transfer
(b) Inference and execution in the real world
Real-world point cloud
RoArtNet
Joint parameters, affordable point and grasp poses
AnyGrasp
①
②
Robot manipulation
①
②
Affordable initial grasp pose and subsequent actions

Fig. 1: RPMArt framework to tackle the real-world articulated objects perception and manipulation. (a) During training, voting targets are generated by part segmentation, joint parameters and affordable points from the simulator to supervise RoArtNet. (b) Given the real-world noisy point cloud observation, RoArtNet can still generate robust joint parameters and affordable points estimation by point tuple voting. Then, affordable initial grasp poses can be selected from AnyGrasp-generated grasp poses based on the estimated affordable points, and subsequent actions can be constrained by the estimated joint parameters.

[1] Junbo Wang, Wenhai Liu, Qiaojun Yu, Weiming Wang, Cewu Lu are with Shanghai Jiao Tong University, China. *Cewu Lu is the corresponding author. Email: `{sjtuwjb3589635689, sjtu-wenhai, yqjllxs, wangweiming, lucewu}@sjtu.edu.cn`
[2] Yang You is with Stanford University, U.S.A. Email: `yangyou@stanford.edu`
[3] Liu Liu is with Hefei University of Technology, China. Email: `liuliu@hfut.edu.cn`

is constrained by the physical articulation joint.

To handle the above challenges, we propose a framework towards **R**obust **P**erception and **M**anipulation for **Art**iculated Objects (**RPMArt**), which learns to estimate the articulation parameters and manipulate the articulation part from the noisy point cloud as depicted in Fig. 1. We draw inspirations from BeyondPPF [10, 11], a sim-to-real 9D object pose estimation method, which formulates the problem of pose estimation as a voting process. Given the point cloud, several point tuples are sampled, and a **Ro**bust **Art**iculation **Net**work (**RoArtNet**) is trained to generate the offsets to the articulation joints and affordable points from the local features of these point tuples. For each point tuple, RoArtNet votes several target candidates. After enumerating all the candidates, the target with the most votes is regarded as the final estimation. Moreover, we introduce an articulation-aware classification scheme to make RoArtNet aware of articulated objects' unique geometric structure, facilitating better sim-to-real transfer. Finally, AnyGrasp [12] is used to propose a series of candidate grasp poses and the affordable initial grasp pose is selected based on the estimated affordable point. And the robot manipulation actions are guided by the estimated articulation joint using impedance control. RPMArt is trained only on synthetic data and is able to transfer zero-shot to real-world articulated objects. We conduct extensive experiments in both simulation and real-world environments, and achieve state-of-the-art performance.

Overall, our contributions are summarized as follows:

- We present RoArtNet, a robust articulation network that takes an articulation-aware voting approach based on local point tuple features to estimate joint parameters and affordable points robustly, facilitating effective transfer to real-world scenarios.
- We employ affordance-based, physics-guided manipulation to generate effective and robust actions executed by the robot, incorporating affordable grasp selection and articulation joint constraint.
- We conduct comprehensive experiments in both simulation and real world, and achieve state-of-the-art performance on both perception and manipulation tasks.

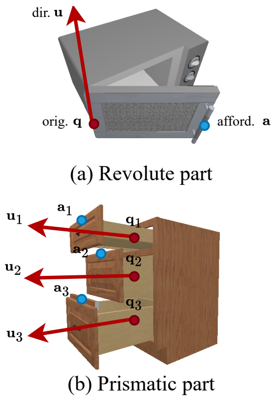
II. Related Work

Articulation perception has been studied for decades, where early stage methods often recover the poses for different parts with prior instance information available such as CAD models [13, 14]. More recently, with the development of deep learning techniques, articulation perception from raw sensory data becomes possible. Hu et al. [15] introduced a part mobility model to map the single static snapshot to dynamic units in the training set. Though this querying method can achieve motion prediction and transfer to the input object, it needs part segmentation as prior information. Shape2Motion [2] takes a two-stage method with mobility proposal and optimization networks to segment motion parts and estimate joint poses, but it is trained and tested on the whole point clouds. The following methods [3–5, 7, 16] exploit strong point cloud processing backbones [17–19] to model articulated objects from single-view point clouds. Though they achieve accurate estimation on synthetic articulated objects, their generalization to the real-world cases is not guaranteed, especially in the presence of unexpected noise. This work deals with single-view real-world point clouds, with only synthetic data used for training.

Articulated object manipulation aims to manipulate the movable part of the articulated object by a robot, and prior works can be broadly categorized into learning-based and planning-based. Some learning-based methods leverage imitation learning [8, 20] or reinforcement learning [9, 21] to learn policy from collected robot demonstrations. However, collecting high-quality demonstrations is time-consuming and expensive. Another line of learning-based methods relies on learning visual affordance heatmap [22, 23] to select contact poses and predict actions [1, 24, 25]. However, the affordance heatmap is ambiguous and hard to annotate. On the other hand, planning-based methods often compute a motion trajectory with some geometry knowledge perfectly known [26, 27] or estimated visually [28, 29]. This work lies in the planning-based methods but also learns affordance to incorporate semantic understanding.

Sim-to-real transfer is commonly needed in many real-world application fields. Although there is vast literature on rigid objects pose estimation [10, 11, 30, 31], a few works have been devoted to articulated objects perception and manipulation. Like other fields, ReArtNOCS [32] renders scanned articulated object models under different real scene backgrounds to synthesize data for training of articulation poses. However, it still does not take care of the domain gap between synthetic and real point clouds, and this tricky rendering process implicitly makes an assumption of test data distribution. This work draws inspirations from BeyondPPF [11] and wants to narrow the sim-to-real gap for articulated objects perception and manipulation.

III. Problem Formulation



(a) Revolute part

(b) Prismatic part

Fig. 2: Illustration of joint parameters and affordable points on articulated objects.

Our perception goal is to estimate the joint parameters and affordable points from an observed articulated object's 3D point cloud $P \in \mathbb{R}^{N\times 3}$ backprojected from a single depth image, where N denotes the number of points (see Fig. 2). Following other works [4, 7], we only consider 1D revolute joints and 1D prismatic joints. And we formulate the joint parameters as $\{\mathbf{u}_j, \mathbf{q}_j \mid j = 1, \ldots, J\}$, where $\mathbf{u}_j \in \mathbb{R}^3$ is the direction of the joint axis, $\mathbf{q}_j \in \mathbb{R}^3$ is the origin of the joint axis, and J denotes the number of joints an articulated object comprises. Note that the origin of prismatic joint is also considered and defined as the center of part front surface in its rest state. Unlike previous works, affordable points $\{\mathbf{a}_j \in \mathbb{R}^3 \mid j = 1, \ldots, J\}$ instead of part segmentation or part bounding boxes are estimated. The affordable point represents the affordance [22, 23] peak among the space, indicating the potential interaction between robot and object which is the most likely to succeed. Our manipulation tasks include pulling and pushing the articulation part by a robot with a two-finger parallel gripper, while ensuring that the change of joint state exceeds a specific threshold.

IV. Method

RPMArt uses RoArtNet, an articulation perception method to estimate joint parameters and affordable points from a noisy point cloud (depicted in Fig. 3), followed by an affordance-based, physics-guided manipulation pipeline (depicted in Fig. 1 (b)). First, several point tuples are sampled from the point cloud, and RoArtNet votes the joint parameters and affordable points based on these samples' local features (detailed in Sec. IV-A). Moreover, RoArtNet is supervised by an articulation-aware classification loss during training, and selects votes with high articulation scores for sim-to-real transfer during inference (detailed in Sec. IV-B). Finally, RPMArt selects an affordable initial grasp pose based on the estimated affordable point and executes subsequent actions constrained by the estimated joint parameters (detailed in Sec. IV-C).

A. RoArtNet for Point Tuple Voting

We draw inspirations from BeyondPPF [11], which is a sim-to-real rigid object pose estimation method that achieves state-of-the-art performance. Unlike most point cloud processing algorithms [18, 33, 34], we want to refrain from aggregating global features of the whole point cloud and only rely on some distinctive local patterns. Thus, given the point cloud P, we sample K point tuples from it, and each point tuple contains M points, with the first two points $\mathbf{p}_1$ and $\mathbf{p}_2$ as the major points. For each point tuple $\mathcal{T} = \{\mathbf{p}_1, \ldots, \mathbf{p}_M\}$, we extract the following features as the network input:

$$\mathcal{F}_1 = \operatorname{concat}(\{\mathbf{p}_i - \mathbf{p}_j \mid (i, j) \in \sigma^2(M)\}), \tag{1}$$

$$\mathcal{F}_2 = \operatorname{concat}(\{\max(\mathbf{n}_i \cdot \mathbf{n}_j, -\mathbf{n}_i \cdot \mathbf{n}_j) \mid (i, j) \in \sigma^2(M)\}), \tag{2}$$

$$\mathcal{F}_3 = \operatorname{concat}(\{\mathbf{s}'_i \mid i = 1, \ldots, M\}), \tag{3}$$

where concat means concatenation, $\sigma^2(M)$ represents all combinations of order 2 from M (a.k.a. M choose 2), $\{\mathbf{n}_1, \ldots, \mathbf{n}_M\}$ represents the normals of $\mathcal{T}$, and $\{\mathbf{s}'_1, \ldots, \mathbf{s}'_M\}$ is computed by MLP layers encoding the SHOT [35] features $\{\mathbf{s}_1, \ldots, \mathbf{s}_M\}$ of $\mathcal{T}$. Here, $\mathcal{F}_1$ represents the relative geometry information, while $\mathcal{F}_2$ and $\mathcal{F}_3$ contain the local context features around each point. Note that all these three features are translation invariant, while $\mathcal{F}_2$ and $\mathcal{F}_3$ are rotation invariant. Rendering under different camera poses can also make $\mathcal{F}_1$ rotation invariant. Such local features can help adapt to different situations, enhancing model robustness.

The network is implemented as a residual [36] MLP, and predicts several offsets to the joint origin $\mathbf{q}$ and joint direction $\mathbf{u}$ with respect to the major points $\mathbf{p}_1$ and $\mathbf{p}_2$:

$$\mu = \overrightarrow{\mathbf{p}_1\mathbf{q}} \cdot \frac{\overrightarrow{\mathbf{p}_1\mathbf{p}_2}}{\left\|\overrightarrow{\mathbf{p}_1\mathbf{p}_2}\right\|_2}, \tag{4}$$

$$\nu = \left\| \mathbf{q} - \left(\mathbf{p}_1 + \mu \frac{\overrightarrow{\mathbf{p}_1\mathbf{p}_2}}{\left\|\overrightarrow{\mathbf{p}_1\mathbf{p}_2}\right\|_2} \right) \right\|_2, \tag{5}$$

$$\theta = \mathbf{u} \cdot \frac{\overrightarrow{\mathbf{p}_1\mathbf{p}_2}}{\left\|\overrightarrow{\mathbf{p}_1\mathbf{p}_2}\right\|_2}. \tag{6}$$

RoArtNet also predicts offsets to the affordable point $\mathbf{a}$ for subsequent grasp pose selection (see Sec. IV-C):

$$\mu_a = \overrightarrow{\mathbf{p}_1\mathbf{a}} \cdot \frac{\overrightarrow{\mathbf{p}_1\mathbf{p}_2}}{\left\|\overrightarrow{\mathbf{p}_1\mathbf{p}_2}\right\|_2}, \tag{7}$$

$$\nu_a = \left\| \mathbf{a} - \left(\mathbf{p}_1 + \mu_a \frac{\overrightarrow{\mathbf{p}_1\mathbf{p}_2}}{\left\|\overrightarrow{\mathbf{p}_1\mathbf{p}_2}\right\|_2} \right) \right\|_2. \tag{8}$$

Once μ and ν are fixed, $\mathbf{q}$ is determined with up to one degree-of-freedom ambiguity in a circle, similar for $\mathbf{a}$ by μ_a and ν_a. Similarly, once θ is fixed, $\mathbf{u}$ lies on a conical surface with one degree-of-freedom ambiguity. Thus, during inference, we can generate multiple candidates with a constant degree interval along the circle or cone for each point tuple, and the target will emerge with the most votes, as demonstrated in Fig. 3 (c). Such voting scheme implicitly recognizes the distinctive local patterns, alleviating interference from noisy points.

For each point tuple $\mathcal{T}$, we optimize the joint origin loss $l_{\text{orig}}^{\mathcal{T}}$ and affordable point origin loss $l_{\text{afford}}^{\mathcal{T}}$ by mean squared error. Like other work [37], we optimize the joint direction loss $l_{\text{dir}}^{\mathcal{T}}$ under the classification-based case by KL divergence. Formally, the vote loss for each tuple $\mathcal{T}$ is defined as:

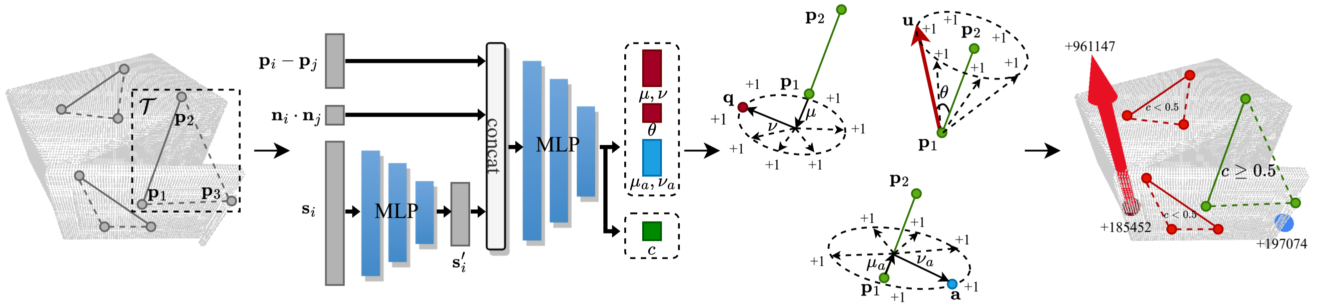
$$l_{\text{vote}}^{\mathcal{T}} = l_{\text{orig}}^{\mathcal{T}} + \lambda_d \cdot l_{\text{dir}}^{\mathcal{T}} + \lambda_a \cdot l_{\text{afford}}^{\mathcal{T}}, \tag{9}$$

where λ_d and λ_a are two weights to balance the influence of different terms.

B. Articulation Awareness

After examining point clouds of both simulated and real-world articulated objects, we find that the distinctive articulation structure, featuring a movable part connected to the base either at an angle or with an offset, is shared commonly among various articulated objects and even preserves in the real-world noisy point clouds. Such a common structure can facilitate the generalization and sim-to-real transfer of the model. To make RoArtNet aware of the articulation structure, an additional articulation score c_j is used to supervise the network during training. The ground truth articulation score $\{c_j \mid j = 1, \ldots, J\}$ of a sampled point tuple $\mathcal{T}$ is calculated based on the part segmentation $\{M_j \mid j = 0, \ldots, J\}$:

$$c_j = \begin{cases} 1, & \text{if } (\mathbf{p}_1, \mathbf{p}_2) \in (M_0, M_j) \text{ or } (\mathbf{p}_1, \mathbf{p}_2) \in (M_j, M_0) \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$



(a) Sample point tuples (b) Predict voting targets (c) Enumerate target candidates (d) Vote final results

Fig. 3: Overview of RoArtNet. First, (a) a collection of M-point tuples ($M = 3$ here as an example) are uniformly sampled from the point cloud. For each point tuple, (b) we predict several voting targets with a neural network from the local context features of the point tuple. Further, an articulation score c is applied to supervise the neural network so that the network is aware of the articulation structure. Then, (c) we can generate multiple candidates using the predicted voting targets, given the one degree-of-freedom ambiguity constraint. (d) The candidate joint origin, joint direction and affordable point with the most votes, from only point tuples with high articulation score, are selected as the final estimation.

where M_0 denotes the mask of the base. This articulation score favors the point tuples whose two major points are located separately in the target part and the base. And the articulation awareness loss $\mathcal{L}_{\text{art}}$ is defined as the binary cross entropy between c_j and predicted $\hat{c}_j$:

$$\mathcal{L}_{\text{art}} = -\frac{1}{JK}\sum_{j=1}^{J}\sum_{\mathcal{T}_k}\left(c_j^k \log \hat{c}_j^k + (1-c_j^k)\log(1-\hat{c}_j^k)\right). \tag{11}$$

During training, we only optimize the vote loss of the point tuples with articulation score as 1:

$$\mathcal{L}_{\text{vote}} = \frac{1}{JC}\sum_{j=1}^{J}\sum_{\mathcal{T}_i}\{l_{\text{vote}}^{\mathcal{T}_i} \mid c_j^i = 1\}, \tag{12}$$

where $C = \text{card}(\{l_{\text{vote}}^{\mathcal{T}_i} \mid c_j^i = 1\})$, and $\text{card}(\cdot)$ means the cardinality of a set. Therefore, our final loss is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{vote}} + \lambda_{aa} \cdot \mathcal{L}_{\text{art}}, \tag{13}$$

where λ_{aa} represents the weight of $\mathcal{L}_{\text{art}}$ term. And during inference, only the votes by point tuples with articulation score higher than 0.5 are kept for voting.

C. *Affordance-based Physics-guided Manipulation*

To start manipulating the articulated object, the robot needs to first grasp the target part. We use AnyGrasp [12] to generate a collection of grasp poses $\mathcal{G} = \{\mathbf{G}_g \in SE(3) \mid g = 1, \ldots, G\}$ given the point cloud P. In order to select one from these grasp poses that can manipulate the target part, we utilize the estimated affordable point. Currently, the grasp pose with minimum distance to the affordable point is selected. We define the affordable point for each part as the affordance [22, 23] peak among the part space, and we manually annotate the ground truth of affordable points. Typically, the affordable point lies on the edge center of the movable part, as shown in Fig. 2.

After grasping the target part, we explicitly exploit the estimated articulation joint and robot's proprioception to generate manipulation actions. In each time step t, we can sense current gripper pose $\mathbf{T}_t$ in robot base space and calculate the target pose $\hat{\mathbf{T}}_{t+1}$ with respect to the estimated articulation joint:

$$\hat{\mathbf{T}}_{t+1} = \begin{cases} \text{Rot}(\delta, \mathbf{u}, \mathbf{q}) \cdot \mathbf{T}_t, & \text{if revolute joint} \\ \text{Tr}(\delta, \mathbf{u}) \cdot \mathbf{T}_t, & \text{if prismatic joint} \end{cases}, \tag{14}$$

where $\text{Rot}(\delta, \mathbf{u}, \mathbf{q})$ represents the transformation matrix for rotating δ angle about axis $(\mathbf{u}, \mathbf{q})$, and $\text{Tr}(\delta, \mathbf{u})$ represents the transformation matrix for translating δ distance along direction $\mathbf{u}$. We employ an impedance controller [38] to realize the actuation torques for reaching target poses.

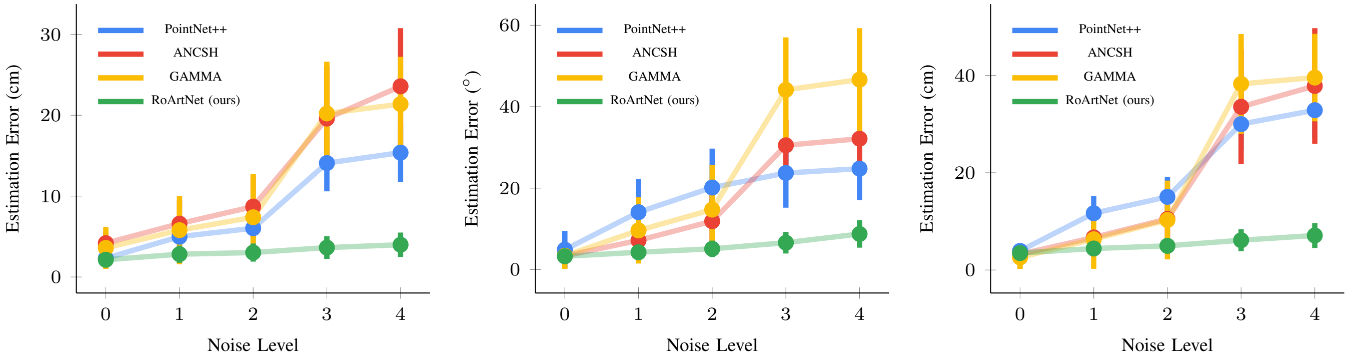
V. Experiments

We perform our experiments in both simulated and real-world environments, and validate our framework by answering the following questions: (i) Can RoArtNet robustly estimate joint parameters and affordable points from point clouds with different levels of noise? (ii) Can RPMArt still manipulate articulated objects successfully under observation noise? (iii) Can RPMArt transfer zero-shot to real-world articulated objects?

A. *Environmental Setup*

Settings. We conduct the simulated experiments in the SAPIEN simulator [39], which supports physical simulation for robots and articulated objects interaction. It also provides depth map and part-level information rendering. We use a Panda flying two-finger parallel gripper to perform the manipulation tasks. In our real-world environment, a 7-DOF Franka Emika robot arm with an Intel RealSense L515 LiDAR camera mounted on the robot's wrist is used to observe and manipulate real-world articulated objects. Computing is done on a NVIDIA A100 GPU.

Datasets. In total, we use 74 synthetic objects in 6 selected categories from PartNet-Mobility [40, 41]. And we randomly split them into training and testing instances. For each instance, we import it into SAPIEN simulator, scale it into normal object size with [0.8, 1.1] additional varying range, and randomly set joint states within the joint limit ranges. A



(a) Joint origin estimation results

(b) Joint direction estimation results

(c) Affordable point estimation results

Fig. 4: Articulation perception results. We gradually add higher level of noise to the input point clouds, and test the joint parameters and affordable points estimation performance. Lower is better. Results are averaged across six object categories. Error bars represent the standard deviation. Different noise levels are detailed in Sec. V-B. More detailed results for each category are listed on our website.

camera with 640×480 resolution is used to capture the depth map and part-level mask. We spherically sample the camera viewpoint in front of the target object, with camera looking at the center of the target object. For the spherical sampling, we set the range of the azimuth angle to [-60°, 60°], and the elevation angle to [0°, 60°]. The distance between camera and object is uniformly distributed in [0.6, 1.2] meter. In practice, we sample 40 different states for each object and 5 camera views for each state to render data. Additionally, we also collect one real object instance for each selected category, and capture its point cloud under different joint states and camera views. Note that we only use the synthetic training instances for training, and conduct evaluation over the synthetic testing instances and real-world objects.

Implementation details. We set the number of sampled point tuples K to 100,000 and each point tuple contains $M = 5$ points. And we set the loss weights $\lambda_d = 0.1$, $\lambda_a = 1.0$ and $\lambda_{aa} = 0.5$ in all our implementation.

Baselines. We compare our method to three baselines: (i) a naive PointNet++ [18] that takes the point cloud as input and directly outputs the joint parameters and affordable points; (ii) ANCSH [4] that exploits normalized coordinate space to estimate joint parameters and uses RANSAC [42] to optimize the transformation to the camera space; (iii) GAMMA [7] that learns dense projection offsets to vote joint parameters and dense clustering offsets to group part points. For ANCSH and GAMMA, we mimic their joint origin estimation and add an additional head for affordable point estimation. And we also use their perception results to finish manipulation tasks.

B. Articulation Perception Results

Metrics. We evaluate the orientation error of the joint axis direction in degrees. We evaluate the translation error of the joint axis origin using the minimum line-to-line distance in centimeters for revolute joints, and using L2 distance in centimeters for prismatic joints. And we evaluate the translation error of the affordable point using L2 distance in centimeters.

Results. To validate the robustness of our method, we test the models on the point clouds with different levels of noise. Like PointCleanNet [43], we add two types of noise to raw point clouds. For the distortion noise, a certain percentage ρ_d of points are sampled and added Gaussian noise with the standard deviation as the proportion σ_d of the original point cloud's bounding box diagonal. And for the outlier noise, a certain percentage ρ_o of points are sampled and replaced with random points that are generated by uniformly sampling among a larger bounding box, whose size is the proportion σ_o of the original bounding box, with the same center. In our experiments, five levels of $(\rho_d, \sigma_d, \rho_o, \sigma_o)$ are tested, as (0, 0, 0, 0), (0.1, 0.01, 0.001, 1.0), (0.2, 0.01, 0.002, 1.0), (0.1, 0.02, 0.001, 2.0), and (0.2, 0.02, 0.002, 2.0). Fig. 4 presents the articulation perception results. Results show that all baselines and RoArtNet achieve high estimation precision without noise added. Nevertheless, with the increasing level of noise, all three baselines exhibit a pronounced increase in estimation errors, with 5.1× to 15.3× increase. In contrast, the mean estimation error of RoArtNet increases very slowly, with 1.9× to 2.7× increase. And the baselines also have much higher standard deviation compared to RoArtNet when high level of noise is added. In addition, we also conduct an ablation study to analyze the influence of articulation awareness. We use votes from all point tuples to determine the final estimation, rather than discarding point tuples with low articulation score. The performance within the Microwave category under noise level 2 is shown in Table I. The performance degrades, especially for affordable point estimation, underscoring the robustness brought by articulation awareness.

TABLE I: Ablation study on articulation awareness.

Model	Error ↓		
	Orig. (cm)	Dir. (°)	Afford. (cm)
Ours w/o awareness	4.81±3.97	5.04±4.03	13.33±10.10
Ours (full)	**3.34**±3.11	**4.53**±3.84	**6.62**±4.88

C. Articulated Object Manipulation Results

Metrics. We run around 100 interaction trials per articulated object instance and report success rates of changing the target joint state over a threshold ratio (here we set 0.85) of specific task value (here we randomly choose a rate from [0.1, 0.7] of the joint limit).

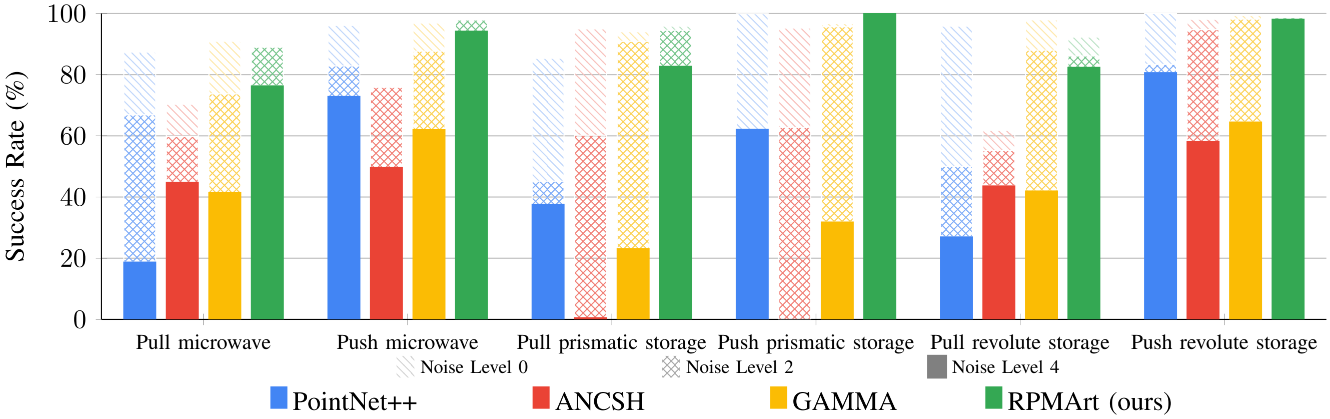


Fig. 5: Articulated object manipulation results. We report the success rate averaged among around 100 trials per object instance for each task. Higher is better. Selected noise levels are detailed in Sec. V-C. More results for other tasks are shown on our website.

Results. Like in the articulation perception experiments, we also add noise to the observed point clouds with different levels of $(\rho_d, \sigma_d, \rho_o, \sigma_o)$, as (0, 0, 0, 0), (0.2, 0.01, 0.002, 1.0), and (0.2, 0.02, 0.002, 2.0). Fig. 5 shows six example task results. It is clear that our method achieves the highest success rate under noise level 4 across all tasks. And we can observe the least degradation in performance of our method with the increase of noise. In addition, we also implement ablation studies to validate different components of our manipulation within the Microwave category under noise level 2, as presented in Table II. We first use the grasp score predicted by AnyGrasp instead of the estimated affordable point to select the initial grasp pose. The performance degrades to much lower success rates, highlighting the importance of affordance-based semantic understanding. Then, we also attempt to plan the entire trajectory at the initial stage instead of constraining by the estimated joint parameters in each time step. The success rate also decreases, indicating the necessity of physical constraints by the articulation joint.

TABLE II: Ablation studies on our affordance-based physics-guided manipulation.

Method	Success rate (%) ↑ Pull	 Push
Ours w/o affordance	38.953	29.286
Ours w/o constraint	77.326	95.714
Ours (full)	**88.953**	**97.857**

D. Real-world Experiments

To validate the ability for sim-to-real transfer of our framework, we also conduct real-world experiments using our model, trained only on synthetic data.

Articulation perception. We first collect point clouds of six real articulated objects under different conditions, including scenarios with and without background, as well as presence or absence of distractors. We capture depth images for each object with 5 uniformly selected joint states, each from 20 randomly selected camera views. Then we use our trained models to estimate the joint parameters and affordable points. The quantitative results, excluding

TABLE III: Quantitative evaluation of the performance on real-world articulation perception.

Category	Method	Error ↓ Orig. (cm)	 Dir. (°)	 Afford. (cm)
Microwave	PointNet++ [18]	4.49±3.57	9.27±5.83	15.44±4.73
	ANCSH [4]	5.10±5.52	9.17±9.56	12.71±7.93
	GAMMA [7]	**2.53**±2.90	9.91±10.67	7.24±10.19
	RoArtNet (ours)	3.83±**2.37**	**5.19**±**3.62**	**6.75**±**3.28**
Refrigerator	PointNet++ [18]	5.21±4.27	9.60±5.34	12.47±9.50
	ANCSH [4]	5.94±5.80	**8.00**±5.91	12.81±13.60
	GAMMA [7]	4.02±4.58	8.68±6.46	12.33±9.97
	RoArtNet (ours)	**2.11**±**1.70**	8.49±**4.27**	**5.85**±**2.80**
Safe	PointNet++ [18]	5.99±4.16	5.94±2.86	9.23±5.63
	ANCSH [4]	5.17±6.76	7.71±14.28	8.51±9.77
	GAMMA [7]	**3.18**±3.86	8.16±13.74	9.06±9.67
	RoArtNet (ours)	4.12±**2.43**	**5.88**±**2.77**	**8.35**±**4.39**
Storage Furniture	PointNet++ [18]	7.54±4.52	**8.78**±**4.99**	10.63±4.03
	ANCSH [4]	6.41±4.22	9.61±6.40	5.18±6.02
	GAMMA [7]	**3.48**±2.28	12.67±10.19	**4.74**±6.66
	RoArtNet (ours)	4.60±**2.05**	9.68±5.45	7.945±**3.40**
Drawer	PointNet++ [18]	8.33±3.38	**7.86**±**5.30**	10.23±4.46
	ANCSH [4]	13.85±3.76	12.14±8.03	7.72±4.70
	GAMMA [7]	**5.06**±**2.36**	14.67±6.77	**6.97**±**3.11**
	RoArtNet (ours)	5.99±3.06	11.31±5.60	7.73±5.25
Washing Machine	PointNet++ [18]	8.85±6.80	37.50±20.68	19.97±9.44
	ANCSH [4]	5.16±4.92	16.24±12.09	11.54±8.23
	GAMMA [7]	6.49±6.18	28.44±14.87	15.96±13.29
	RoArtNet (ours)	**1.58**±**1.20**	**5.60**±**2.71**	**3.25**±**0.67**

TABLE IV: Real-world articulated object manipulation results. We run 10 trials for each task and count the number of successful/half-successful/failed trials respectively, where half-successful trials include behaviors like detaching during pulling and pushing forcefully.

Tasks		PointNet++ [18]	ANCSH [4]	GAMMA [7]	RPMArt (ours)
Microwave	Pull	6/2/2	4/0/6	8/1/1	**9/1/0**
	Push	5/4/1	3/4/3	6/3/1	**7/1/2**
Refrigerator	Pull	2/1/7	1/1/8	3/1/6	**7/0/3**
	Push	0/0/10	1/0/9	2/0/8	**8/1/1**
Safe	Pull	**7/0/3**	5/2/3	5/1/4	**7/0/3**
	Push	7/0/3	**7/1/2**	**7/1/2**	**7/1/2**
Storage Furniture	Pull	1/0/9	3/1/6	2/1/7	**4/0/6**
	Push	2/2/6	**6/2/2**	2/3/5	5/2/3
Drawer	Pull	1/1/8	2/1/7	0/2/8	**2/2/6**
	Push	2/0/8	2/1/7	0/0/10	**3/2/5**
Washing Machine	Pull	0/0/10	0/1/9	0/0/10	**3/3/4**
	Push	0/0/10	0/0/10	0/0/10	**1/2/7**

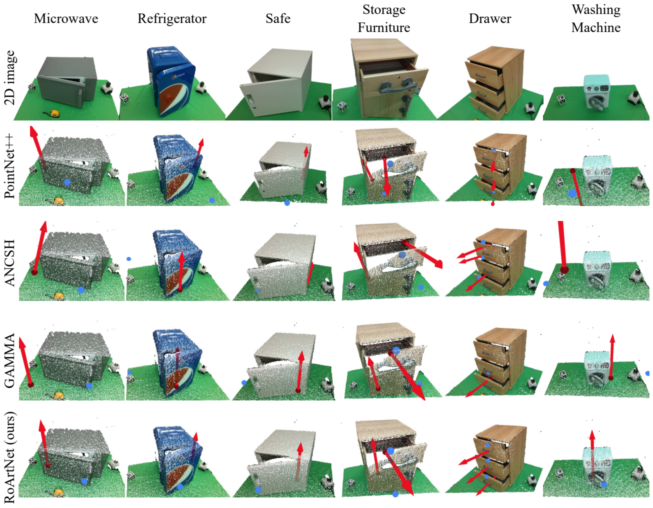


Fig. 6: Qualitative results of the performance on real-world articulation perception. Color is used only for visualization here. Red arrows represent the estimated articulation joints, and blue points represent the estimated affordable points. Zoom in for better view.

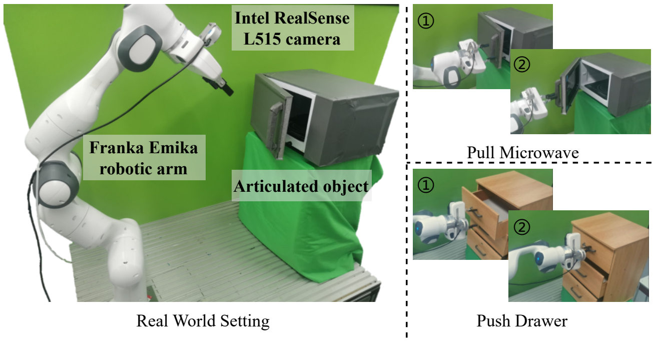


Fig. 7: Real-world manipulation experiments.

backgrounds and distractors, are shown in Table III. We also visualize the estimation results with both background and distractors included in Fig. 6. We can find that RoArtNet demonstrates more stable performance compared to other baselines. However, some performance degradation is found in the StorageFurniture and Drawer categories for RoArtNet, as well as for ANCSH and GAMMA. This could possibly be attributed to the relatively small size of parts in these two objects, where all three models somewhat rely on part segmentation to complete the estimation. Another noteworthy observation pertains to the performance on WashingMachine. Specifically, only RoArtNet successfully estimates targets accurately, while the other three baselines exhibit significantly large estimation errors. We find a potential reason that we take a relatively small washing machine toy as the object, then the influence of noisy points is relative significant.

Articulated object manipulation. We also apply the models to manipulate the real articulated objects. We run 10 trials for each task, and count the number of successful, half-successful and failed trials. Here, half-successful trials include behaviors like detaching during pulling and pushing forcefully. Table IV shows the statistics, and Fig. 7 illustrates the manipulation process. Videos are available on our website. Our method outperforms other counterparts, especially for Refrigerator and WashingMachine. Refrigerator has a glossy surface, while WashingMachine is relatively small, which makes the noise more prominent.

VI. Conclusion

We present RPMArt, a framework towards robust perception and manipulation for articulated objects. At its core, RoArtNet learns local context features from sampled point tuples to vote the joint parameters and affordable points robustly. To further improve its capability for sim-to-real transfer, articulation awareness is introduced to account for the unique geometric structure of articulated objects. Finally, we use the estimated affordable point to select the affordable initial grasp pose and generate manipulation actions guided by the estimated joint constraints. Experiments show that RPMArt achieves state-of-the-art performance in both noise-added simulation and real-world environments. Currently, RoArtNet can only achieve category-level generalization. In future work, we will explore methods that can also accomplish robust cross-category estimation.

Acknowledgements

This work was supported by the National Key Research and Development Project of China (No. 2022ZD0160102, No. 2021ZD0110704), Shanghai Artificial Intelligence Laboratory, XPLORER PRIZE grants, National Natural Science Foundation of China (No. 52305030, No. 62302143), and Anhui Provincial Natural Science Foundation (No. 2308085QF207).

References

[1] R. Wu, Y. Zhao, K. Mo, Z. Guo, Y. Wang, T. Wu, Q. Fan, X. Chen, L. Guibas, and H. Dong, "Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects," *arXiv preprint arXiv:2106.14440*, 2021.

[2] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu, "Shape2motion: Joint analysis of motion parts and attributes from 3d shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8876–8884.

[3] B. Abbatematteo, S. Tellex, and G. Konidaris, "Learning to generalize kinematic models to novel objects," in *Proceedings of the 3rd Conference on Robot Learning*, 2019.

[4] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3706–3715.

[5] H. Xue, L. Liu, W. Xu, H. Fu, and C. Lu, "Omad: Object model with articulated deformations for pose estimation and retrieval," *arXiv preprint arXiv:2112.07334*, 2021.

[6] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1647–1654.

[7] Q. Yu, J. Wang, W. Liu, C. Hao, L. Liu, L. Shao, W. Wang, and C. Lu, "Gamma: Generalizable articulation modeling and manipulation for articulated objects," *arXiv preprint arXiv:2309.16264*, 2023.

[8] H. Shen, W. Wan, and H. Wang, "Learning category-level generalizable object manipulation policy via generative adversarial self-imitation learning from demonstrations," *IEEE*

Robotics and Automation Letters, vol. 7, no. 4, pp. 11 166–11 173, 2022.

[9] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," *arXiv preprint arXiv:2302.04659*, 2023.

[10] Y. You, R. Shi, W. Wang, and C. Lu, "Cppf: Towards robust category-level 9d pose estimation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6866–6875.

[11] Y. You, W. He, M. X. Liu, W. Wang, and C. Lu, "Go beyond point pairs: A general and accurate sim2real object pose voting method with efficient online synthetic training," *arXiv preprint arXiv:2211.13398*, 2022.

[12] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, 2023.

[13] F. Michel, A. Krull, E. Brachmann, M. Y. Yang, S. Gumhold, and C. Rother, "Pose estimation of kinematic chain instances via object coordinate regression." in *BMVC*, 2015.

[14] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins, "Factored pose estimation of articulated objects using efficient nonparametric belief propagation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7221–7227.

[15] R. Hu, W. Li, O. Van Kaick, A. Shamir, H. Zhang, and H. Huang, "Learning to predict part mobility from a single static snapshot," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–13, 2017.

[16] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. 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*, 2023, pp. 7081–7091.

[17] C. M. Bishop, "Mixture density networks," 1994.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[19] B. Graham and L. Van der Maaten, "Submanifold sparse convolutional networks," *arXiv preprint arXiv:1706.01307*, 2017.

[20] P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, and C. Schmid, "Instruction-driven history-aware policies for robotic manipulations," in *Conference on Robot Learning*. PMLR, 2023, pp. 175–187.

[21] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," *arXiv preprint arXiv:2107.14483*, 2021.

[22] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, vol. 1, no. 2, pp. 67–82, 1977.

[23] M. Hassanin, S. Khan, and M. Tahtali, "Visual affordance and function understanding: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–35, 2021.

[24] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6813–6823.

[25] Z. Xu, Z. He, and S. Song, "Universal manipulation policy network for articulated objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2447–2454, 2022.

[26] L. Peterson, D. Austin, and D. Kragic, "High-level control of a mobile manipulator for door opening," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 3. IEEE, 2000, pp. 2333–2338.

[27] S. Chitta, B. Cohen, and M. Likhachev, "Planning for autonomous door opening with a mobile manipulator," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1799–1806.

[28] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to open new doors," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2751–2757.

[29] M. Arduengo, C. Torras, and L. Sentis, "Robust and adaptive door operation with a mobile robot," *Intelligent Service Robotics*, vol. 14, no. 3, pp. 409–425, 2021.

[30] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[31] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2642–2651.

[32] L. Liu, H. Xue, W. Xu, H. Fu, and C. Lu, "Toward real-world category-level articulation pose estimation," *IEEE Transactions on Image Processing*, vol. 31, pp. 1072–1083, 2022.

[33] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[34] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual mlp framework," *arXiv preprint arXiv:2202.07123*, 2022.

[35] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] X. Yang and J. Yan, "On the arbitrary-oriented object detection: Classification based approaches revisited," *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1340–1365, 2022.

[38] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE transactions on control systems technology*, vol. 13, no. 4, pp. 559–576, 2005.

[39] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.

[40] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[41] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 909–918.

[42] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[43] M.-J. Rakotosaona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, "Pointcleannet: Learning to denoise and remove outliers from dense point clouds," in *Computer graphics forum*, vol. 39, no. 1. Wiley Online Library, 2020, pp. 185–203.