Part-Guided 3D RL for Sim2Real Articulated Object Manipulation

Pengwei Xie, Rui Chen, *Member, IEEE*, Siang Chen, Yuzhe Qin, Fanbo Xiang, Tianyu Sun, Jing Xu, *Member, IEEE*, Guijin Wang, and Hao Su

***Abstract*—Manipulating unseen articulated objects through visual feedback is a critical but challenging task for real robots. Existing learning-based solutions mainly focus on visual affordance learning or other pre-trained visual models to guide manipulation policies, which face challenges for novel instances in real-world scenarios. In this letter, we propose a novel part-guided 3D RL framework, which can learn to manipulate articulated objects without demonstrations. We combine the strengths of 2D segmentation and 3D RL to improve the efficiency of RL policy training. To improve the stability of the policy on real robots, we design a Frame-consistent Uncertainty-aware Sampling (FUS) strategy to get a condensed and hierarchical 3D representation. In addition, a single versatile RL policy can be trained on multiple articulated object manipulation tasks simultaneously in simulation and shows great generalizability to novel categories and instances. Experimental results demonstrate the effectiveness of our framework in both simulation and real-world settings.**

***Index Terms*—Deep learning in grasping and manipulation, RGB-D perception, reinforcement learning.**

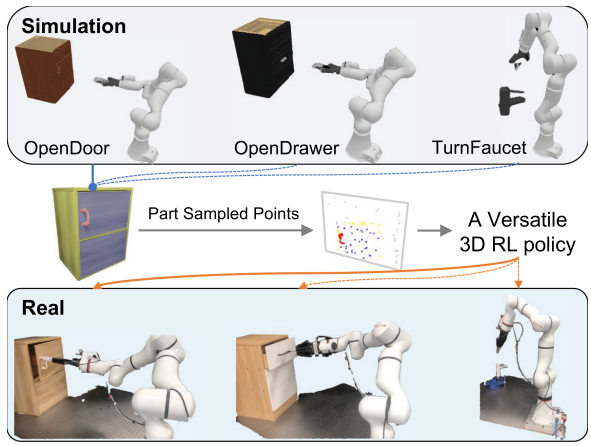


Fig. 1. Our 3D RL framework can be trained for various articulated object manipulation tasks in simulation simultaneously and efficiently. After training, without any demonstrations, the versatile RL policy can be deployed to a real robot and perform different tasks.

I. Introduction

ARTICULATED object manipulation is a fundamental problem in robotics. Unlike object grasping, which only requires stable contact between the gripper and the object [1], [2], articulated objects manipulation involves controlling the relative motion of their parts and is more challenging due to the complex kinematic structures and dynamic properties.

Recently, learning-based methods have emerged as a promising approach to learning manipulation policies from visual inputs [3], [4], [5], [6], [7], [8], [9]. Some approaches, such as [8], [9], reconstruct the object and estimate its kinematic and dynamic properties, but require interactive perception and are not practical for novel objects. Other methods [3], [4], [5], [6], [7] learn visual affordance information to guide the manipulation policies. However, these methods still face several limitations, such as the need for expert knowledge to control robot motion, the low sample efficiency of reinforcement learning (RL) algorithms, and the significant shape variations of articulated objects within and across different categories. Furthermore, the generalizability and robustness of these methods on real robots have yet to be sufficiently evaluated.

In this work, we propose an efficient part-guided 3D RL framework for articulated object manipulation without any demonstrations, as shown in Fig. 1. First, we train a part segmentation module on synthetic data and implement it to segment the 2D image into different articulated parts. Then, we transform the segmentation result into 3D points with part information, as 3D representations are more appropriate for reasoning the relationship between the robot and the object. Finally, we extract geometric features from the points to guide the RL policy training. By leveraging the strength of both the 2D segmentation network and 3D RL, our framework not only improves the sample efficiency but also enables the training of a single versatile RL policy capable of handling manipulation

Manuscript received 27 April 2023; accepted 28 August 2023. Date of publication 7 September 2023; date of current version 26 September 2023. This letter was recommended for publication by Associate Editor R. Camoriano and Editor J. Kober upon evaluation of the reviewers' comments. *(Pengwei Xie, Rui Chen, and Siang Chen contributed equally to this work.) (Corresponding author: Guijin Wang.)*

Pengwei Xie and Tianyu Sun are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: xpw18@mails.tsinghua.edu.cn; sty21@mails.tsinghua.edu.cn).

Rui Chen and Jing Xu are with the Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China (e-mail: chenruithu@mail.tsinghua.edu.cn; jingxu@tsinghua.edu.cn).

Siang Chen and Guijin Wang are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China, and also with the Shanghai AI Laboratory, Shanghai 200232, China (e-mail: csa21@mails.tsinghua.edu.cn; wangguijin@tsinghua.edu.cn).

Yuzhe Qin, Fanbo Xiang, and Hao Su are with the Department of Computer Science, University of California San Diego, San Diego, CA 92037 USA (e-mail: y1qin@eng.ucsd.edu; fxiang@eng.ucsd.edu; haosu@ucsd.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3313063, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3313063

tasks for various articulated objects with different kinematic structures.

In particular, we design a Frame-consistent Uncertainty-aware Sampling (FUS) strategy to obtain a condensed and hierarchical 3D object representation. Due to the Sim2Real gap and the variance of objects and illuminations in real-world scenarios, the 2D segmentation network trained on synthetic data cannot yield satisfactory results even with domain randomization. Therefore, we introduce a method to estimate the uncertainty of the segmentation predictions and analyze the consistency of points across consecutive frames to obtain a more reliable and accurate object representation. Compared with traditional sampling methods, our sampling method preserves the object structure with higher fidelity, resulting in a more stable manipulation policy on real robots.

The key contributions of our study are as follows.

1) We propose a generic framework for articulated object manipulation, which exploits the advantages of 2D segmentation and 3D RL. It enables learning a versatile policy to handle different articulated objects.
2) We design a novel weighted point sampling strategy considering both point uncertainty and consistency, which ensures a condensed object representation and reduces the Sim2Real gap during policy deployment.
3) We conduct various quantitative evaluations in three articulated object manipulation tasks, both in simulation and real-world experiments, proving the effectiveness of our framework.

II. Related Work

A. Learning-Based Articulated Objects Manipulation

Learning-based methods have shown remarkable progress in vision-based articulated object manipulation. Affordance learning has been proposed for articulated objects, such as [3], [4], [5], [6], [7], [10]. Recent impressive works, such as UMPNet [6] and FlowBot3D [7], first learn visual affordance and then use human-designed motion commands for execution with suction grippers. However, they may necessitate additional endeavors to address the grasping issue for other grippers. In this letter, our RL approach works in an end-to-end fashion, demonstrating excellent performance in simulation and zero-shot transfer to the real world.

[11], [12] estimate one or several keypoints from images to guide the manipulation policy. In contrast, our method utilizes part-guided sampled points, which significantly enhances robustness and leads to better generalizability on novel object instances in both simulation and real world.

B. Visual Pre-Training for Robot Learning

Recent research has demonstrated the effectiveness of visual pre-training in improving robot learning [13], [14], [15], [16], [17], [18]. Lin et al. [13] have shown that visual representations from semantic tasks highly correlate with affordance maps widely used in manipulation. Many recent works have leveraged self-supervised visual pre-training models on images to guide manipulation tasks [14], [16], [18]. However, most existing approaches [15], [17] require learning from large-scale, diverse, offline human videos, which is computationally expensive and labor intensive. Furthermore, additional real demonstrations are often needed to transfer the learned policies to real-world robot tasks [16], [17]. In contrast, our approach pre-trains the visual representations solely on synthetic data, which can be easily collected on a large scale. By incorporating domain randomization methods, our visual model achieves excellent performance on real captured images. Moreover, we lift the 2D representation to 3D sampled points, which further improves the training efficiency of RL.

III. Method

A. Overview

We model the articulated object manipulation problem as a Partially Observable Markov Decision Process (POMDP) [19], aiming to learn a policy $\pi : \mathcal{O} \rightarrow \mathcal{A}$. The observation $o \in \mathcal{O}$ includes visual features and robot states, and the action $a \in \mathcal{A}$ indicates the robot target joint positions and the gripper finger position. In our work, the agent employs a hand-centric camera to mitigate occlusion issues from third-person perspectives, showing superior generalizability for manipulation tasks in both simulation and real world [20].

As depicted in Fig. 2, our framework develops an efficient RL policy for various articulated object manipulation tasks. Firstly, we take a hand-centric RGB-D image $\mathbf{I} \in \mathbb{R}^{H \times W \times 4}$ as input. Next, our segmentation model predicts the part segmentation map $\mathbf{S} = \{0, 1\}^{C \times H \times W}$, where C denotes the number of part classes. Our Frame-consistent Uncertainty-aware Sampling (FUS) strategy selects points based on the segmentation map. Finally, geometric features are extracted from these points and fed into the RL algorithm with robot states to predict actions and corresponding critic values.

B. Part Segmentation

We aim to train an efficient part segmentation model from RGB-D images. Note that parts can also be segmented from point clouds. However, in our hand-centric setting, depth sensor noise may affect the quality of point clouds, especially when the camera is close to the object. Additionally, point cloud downsampling is necessary to enable efficient RL training. This decreases the points from small parts and degrades the segmentation performance potentially.

Part Definition: Our goal is to develop an object-irrelevant part representation that exhibits generalizability in manipulation tasks across various articulated objects. Similar to GAPartNet [21], our work defines a part as a rigid segment that shares similar affordances, facilitating generalizable and consistent interaction behavior. As shown in Fig. 2, cabinet doors and drawers consist of three parts: fixed handles, door/drawer facades, and fixed bases. Faucets are composed of handles and fixed bases. By adopting the generalizable part representation, our part-based RL method exhibits the potential to manipulate various types of articulated objects.

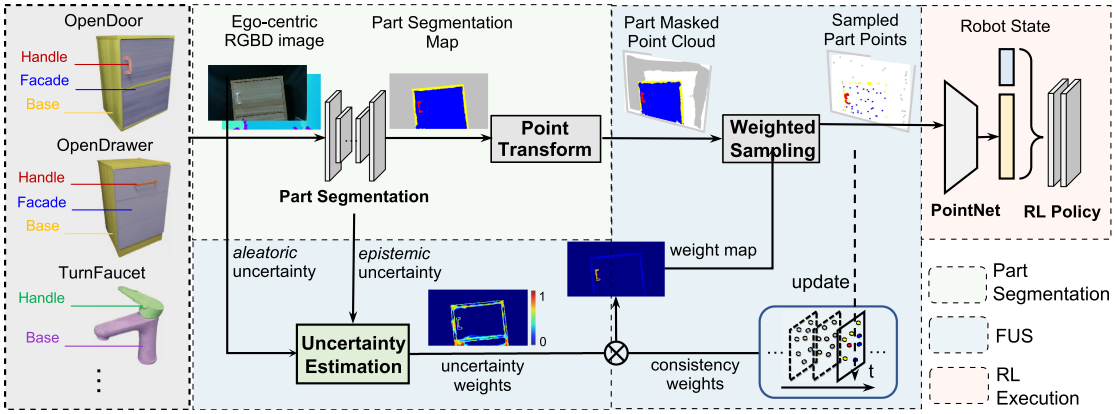


Fig. 2. Framework Overview. 1) We take hand-centric visual observation (i.e., RGB-D images) as input and predict the part segmentation map using a pre-trained segmentation network. 2) 3D part masked points are transformed from the depth image using the camera parameters. 3) Our proposed FUS strategy combines uncertainty and consistency weights to generate per-point weights. These weights are used to sample points for each part. 4) Geometric features extracted using PointNet are combined with the robot states, and fed to the RL algorithm to get the action. After the robot executes the action, a new observation is obtained, and the process iterates from the beginning.

Synthetic Data Generation: To generate a large amount of annotated data, we collect data from various categories of articulated objects in the simulation. First, We centrally position each object on a table and deploy a floating gripper fitted with a mounted camera to capture RGB-D images from diverse viewpoints. Corresponding part masks are automatically generated from the simulation without human intervention. And then, we introduce scene-level randomization to enrich the dataset, including variations in the distance between the gripper and object, camera orientation, turning angle of the articulated parts, and opening distance.

Domain Randomization: Domain randomization techniques have demonstrated great potential in bridging the gap between synthetic and real data by increasing the data's diversity [11], [12], [22]. In our data generation process, we randomize several material parameters for the object mesh models, such as metallic, roughness, and specular parameters. Additionally, random texture patterns are introduced on the surface of different objects. To further diversify the background patterns, we replace the original black background with random RGB-D images from SUNRGBD [23] and SceneNetRGBD [24]. The introduction of large variations during data generation effectively expands the distribution of synthetic data. Several data augmentation techniques are also employed in the training phase to enhance the generalizability of our segmentation network. These techniques include color jittering for RGB images, adding salt-and-pepper noise to depth images, applying random cropping and rotation, and introducing Gaussian noise. We also simulate depth sensor noise during data generation to narrow the domain gap, as previous studies have shown its effectiveness [25].

With the part masks, there are multiple approaches to guide the subsequent RL policy learning. Many image-based RL approaches [13], [20], [26] use an image encoder to extract image features. However, 2D features may not be sufficient for 3D robotic manipulation tasks that require reasoning about the geometric structure and agent-object relationships in 3D space [2], [27]. Therefore, we opt for using 3D point cloud representations. Depth pixels are transformed to 3D points in the world frame using the calibrated camera's intrinsic and extrinsic parameters. As the original point clouds can be quite large, we apply downsampling methods to accelerate RL training. To ensure the reliability and consistency of the sampled points, we introduce a Frame-consistent Uncertainty-aware Sampling strategy.

C. Frame-Consistent Uncertainty-Aware Sampling

There are various sampling strategies for point clouds. Uniformly Downsampling is the most common approach [3], [10], [27], [28]. This method first filters out table points from the point cloud in the world frame, and then uniformly downsamples the remaining points to a fixed number. However, many points of key parts may be dropped, especially when the part is small. Farthest point sampling (FPS) is a commonly employed strategy in point cloud neural processing systems [29]. However, FPS does not consider subsequent processing steps on the sampled points and may therefore yield sub-optimal performance. To maintain the versatility of our segmentation model for manipulation tasks, we do not assume the significance of any specific part. Therefore, we aim to sample the same number of points from each part, namely N_s. For optimal results, points sampled from the same mask should accurately represent its shape information and display consistency throughout different stages of the manipulation policy. Therefore, our proposed weighted sampling strategy consists of uncertainty weights and consistency weights.

Uncertainty Weights: Although our segmentation model performs well in simulation, its performance may decrease in real-world scenarios due to domain gaps between simulation and reality, even though we have implemented various domain randomization techniques. Scattered and inconsistent sampled points can result from sub-optimal segmentation outcomes, impeding the learning of subsequent policies. Inspired by [30], [31], we employ the uncertainty estimation method to measure the potential uncertainty in the Sim2Real transfer process. Specifically, we utilize Test-Time Augmentation (TTA, such as

flipping, color jittering on RGB images, and salt-and-pepper noise on depth images) to estimate *aleatoric uncertainty* (inherent observation noise), and employ Monte Carlo Dropout [30] to estimate *epistemic uncertainty* (model uncertainty).

First, given the input image $\mathbf{I}$ of the current time step, we compute the corresponding points $\mathbf{p}$ in the world frame by transforming the pixel coordinates $(\mathbf{u}, \mathbf{v})$ and depth values $\mathbf{d}$ using the calibrated camera intrinsic and extrinsic parameters. Next, we add TTA on $\mathbf{I}$ and perform K stochastic forward inferences under random dropout. Therefore, we obtain a set of softmax probability map $\{\mathbf{P}_k \in \mathbb{R}^{C\times H\times W}\}_{k=1}^{K}$. And then, we use the predictive entropy to approximate the uncertainty [31], summarized as

$$\mathbf{P}_c = \frac{1}{K}\sum_k \mathbf{P}_k^c \text{ and } \mathbf{U} = -\sum_c \mathbf{P}_c \log \mathbf{P}_c, \tag{1}$$

where $\mathbf{P}_c$ is the probability in the c-th part class at the current time step. And the uncertainty map $\mathbf{U}\in \mathbb{R}^{H\times W}$ is then normalized to [0,1]. This uncertainty map allows us to assign different sampling weights to each point. Specifically, given the part segmentation map $\mathbf{S}$ and the related normalized uncertainty score $\mathbf{U}$, the part-based uncertainty and uncertainty sampling weights can be calculated as

$$\mathbf{U}_c = \left[\mathbf{U}_{(u,v)} \middle| \arg\max \mathbf{S}_{(u,v)} = c\right], c \in \{1, \ldots, C\}, \tag{2}$$

$$\mathbf{w}_c^{ua} = \text{softmax}(\mathbf{U}_c), c \in \{1, \ldots, C\}, \tag{3}$$

where $\mathbf{U}_c \in \mathbb{R}^{N_c\times 1}$ denotes the uncertainty score vector of pixels belonging to part c and N_c is the number of pixels. We apply the softmax function to obtain the normalized sampling weights for each part.

Consistency Weights: The consistency of sampled points across frames is another key factor to consider. During the manipulation process, the points in the world frame often remain stable across consecutive frames. Even after the articulated part is moved, the points in the adjacent regions will not change significantly. Based on the above observation, we design the frame-consistent sampling weights to ensure the sampled points remain consistent in the process. First, for the initial frame, we randomly sample N_s points from each part. Then, a queue $\mathcal{Q}$ of length T_{fc} is kept to store the sampled points in the manipulation process. For the current time step, we compute the distance vector $\mathbf{d}_c \in \mathbb{R}^{N_c\times 1}$ between the part points $\mathbf{p}_c \in \mathbb{R}^{N_c\times 3}$ and the part points $\mathbf{q}_c \in \mathbb{R}^{N_s\times 3}$ from $\mathcal{Q}$ for each part:

$$\mathbf{d}_c = \left[\min_{\mathbf{q}_j\in\mathbf{q}_c} \|\mathbf{p}_i - \mathbf{q}_j\| \middle| \mathbf{p}_i \in \mathbf{p_c}\right], c \in \{1, \ldots, C\}. \tag{4}$$

As shown in Fig. 3, noticing that the points from the same part are closer, and the noisy points are unstable and usually have much larger distances to the points of the specific part, we use the distance $\mathbf{d}_c$ to calculate the frame-consistent sampling weights $\mathbf{w}_c^{fc}$ as

$$\mathbf{w}_c^{fc} = 2^{-K^{fc}\cdot \mathbf{d}_c}, \tag{5}$$

where K^{fc} denotes the decay coefficient.

Frame-consistent Uncertainty-aware Weights: Now we have uncertainty weights and consistency weights for each part, and

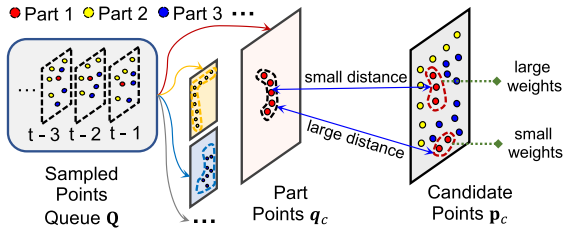


Fig. 3. We calculate consistency weights from several consecutive frames. Candidate points closer to former sampled ones from the same part are allocated larger sampling weights.

Algorithm 1: Part-Guided Articulation Manipulation Policy.

Require: $\theta \leftarrow$ parameters of the segmentation model f_θ
  while not $EpisodeComplete()$ **do**
    $\mathbf{I} \leftarrow$ the RGB-D image.
    $\mathbf{S} \leftarrow f_\theta(\mathbf{I})$, Predict the segmentation map.
    $\mathbf{p} \leftarrow PointTransform(\mathbf{I}, \mathbf{S})$.
    Get $\mathbf{w}_c$ from (6).
    $\hat{\mathbf{p}} : \bigcup_c \hat{\mathbf{p}}_c \leftarrow WeightSampling(\mathbf{p}, \mathbf{w}_c)$
    $\hat{F} \leftarrow PointNet(\hat{\mathbf{p}})$, Get geometric features.
    $a \leftarrow Actor(\hat{F} \oplus g)$, Predict action.
  end while

the total sampling weights are calculated as

$$\mathbf{w}_c = \mathbf{w}_c^{fc} \circ \mathbf{w}_c^{ua}, c \in \{1, \ldots, C\}, \tag{6}$$

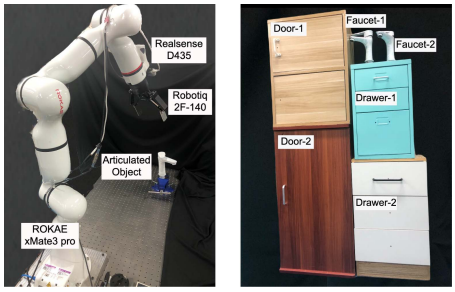
where $\circ$ denotes element-wise multiplication. Then we sample N_s points $\hat{\mathbf{p}}_c$ from each part according to $\mathbf{w}_c$. Then the combined part points $\hat{\mathbf{p}}$ are put into $\mathcal{Q}$, accounting for the calculation of $\mathbf{w}_c^{fc}$ of the next frame.

As depicted in Fig. 3, the consistency weights can be recovered when erroneous part points are removed from the queue. Furthermore, in our real experiments, we observe that the presence of individual frames with inaccurate segmentation does not significantly affect the subsequent steps.

D. RL Policy Learning

As Algorithm 1 shows, an RGB-D image $\mathbf{I}$ is first segmented into multiple parts. Then, we transform these parts into 3D points $\mathbf{p}$, incorporating their corresponding one-hot part-belonging encoding. Next, we conduct part-guided weighted sampling on the raw part points to acquire consistent part representation $\hat{\mathbf{p}}_c$. Similar to [4], [5], [27], a PointNet [32] is utilized to extract compact geometric features $\hat{F}$ from the combined part points $\hat{\mathbf{p}}$. Consequently, $\hat{F}$ is concatenated with the robot states g and fed into the RL algorithm.

Reward Shaping: To effectively train the RL policy for common articulated object manipulation, we design a versatile reward function, which includes five standard terms. 1) Approaching term: encouraging the gripper to approach the movable part. 2) Direction term: promoting manipulation of the movable part in the appropriate direction. 3) Position term: driving the gripper to move the movable part to the target position. 4) Visibility



(a) Real Setup (b) Real articulated objects

Fig. 4. Real experimental setup and 3 categories of articulated objects for our experiment. (a) Real Setup. (b) Real articulated objects.

term: encouraging the agent to maintain visual contact with the movable part. 5) Grasp term: encouraging the gripper to grasp the handles. All terms are utilized for *OpenDoor* and *OpenDrawer*. However, for the *TurnFaucet* task, the Grasp term is removed since grasping is not necessary for manipulating faucets.

IV. Experiments

To evaluate the performance of our proposed framework in both simulated and real-world scenarios, we aim to answer the following research questions: 1) How does our method compare to existing methods for articulated object manipulation tasks? 2) How effective is our framework for real-world robot applications, particularly in addressing the Sim2Real gap? 3) How does our part-based sampling strategy compare to other traditional methods regarding accuracy and stability? 4) To what extent is the versatile RL policy trained on three tasks simultaneously effective in both simulation and real-world experiments?

A. Experimental Setup and Evaluation Metrics

Experimental Setup: As Fig. 4 shows, our robot platform consists of an optical table, a 7-DOF robot (ROKAE xMate3Pro) with a 2-finger gripper (Robotiq 2F-140) mounted on its end link, an RGB-D camera (Intel RealSense D435). Besides, we randomly place the object in front of the robot arm within the robot's reachable workspace.

The simulation results presented in this study are obtained using the Sapien physics simulator [33]. We select three categories of articulated objects with a single movable part from the Partnet Mobility dataset [33] to collect synthetic data, including 40 doors, 16 drawers, and 14 faucets. In the RL policy training stage, 4 instances are selected from each category. All methods are trained for 2 million steps, and we report the training results averaged across 7 independent experimental runs with randomized seeds.

Our visual observation consists of a single RGB-D image captured from a hand-centric camera, with a size of 144×256. To facilitate the RL policy training with fast segmentation inference, we follow the streamlining design in Unet [34] and build it using MobileNetV2 [35] as encoders for its good feature representation and high efficiency. Besides, the robot states include the joint angles, gripper finger position, and end-effector positions in the robot frame. We set K, T^{fc}, and K^{fc} to 4, 3, and 40. Moreover, Soft Actor-Critic (SAC) [36] is adopted as our RL algorithm.

Evaluation Metrics: We evaluate the performance of a policy by its success rate. Following related RL-based works [11], [28], a task trial is considered a success if the movable part is moved at least a fixed range, typically set at 50%. Manipulating the movable part to its full range poses a significant challenge due to the constrained workspace imposed by the fixed arm. In our work, we set the range 50% for all tasks in simulation. In the real world, the range is 45° for doors and faucets, and 10 cm for drawers. In addition, we record the average success steps to evaluate the efficiency of a policy, which refers to the number of steps averaged by all successful trajectories.

B. Experiments in Simulation

To evaluate the effectiveness of our proposed method for articulated object manipulation tasks, we compare it against several vision-based RL baselines. Note that all baselines only differ in the visual representation, and all the other settings are kept the same.

- *Oracle:* This baseline directly obtains the ground-truth segmentation map from simulation, which is the upper bound of performance for our framework.
- *Image-based RL:* The segmentation map and the corresponding depth map are fed into an image encoder to extract visual features. We choose the image encoder from [26] due to its high efficiency for RL training. Our objective is to compare the part-guided image features with our part-guided geometric features.
- *Keypoints-based RL:* We follow [12] and estimate 3 keypoints from handles. Then we put the keypoints with the robot states into the RL policy. Different from the original work, the keypoints are estimated from hand-centric RGB-D images during the manipulation process, introducing invisibility challenges. Despite this, we achieve comparable results (estimation error is around 1.6 cm without invisibility and about 2.1 cm for all scenarios).
- *Dexpoint-based RL:* We follow the same observation as [28] in our hand-centric setting. Specifically, the filtered downsampled scene point clouds and the imagined points of our gripper are concatenated with their one-hot encoding as the PointNet input. The extracted geometric features and the robot states are then fed into the RL policy.

We evaluate all methods on three single tasks and one *HybridTask* that contains all three categories of articulated objects. For each task, we train the RL policy with one randomly selected object in each episode. To ensure a fair comparison, we evaluate the methods on one novel instance for each task and record their success rate over 50 trials.

As presented in Fig. 5, our proposed framework demonstrates superior performance in terms of training efficiency and stability compared to the other baselines for all single manipulation tasks. For the *HybridTask*, our approach stands out as the sole method capable of concurrently learning from various types of articulated objects, attributing to our generalizable and condensed part representation.

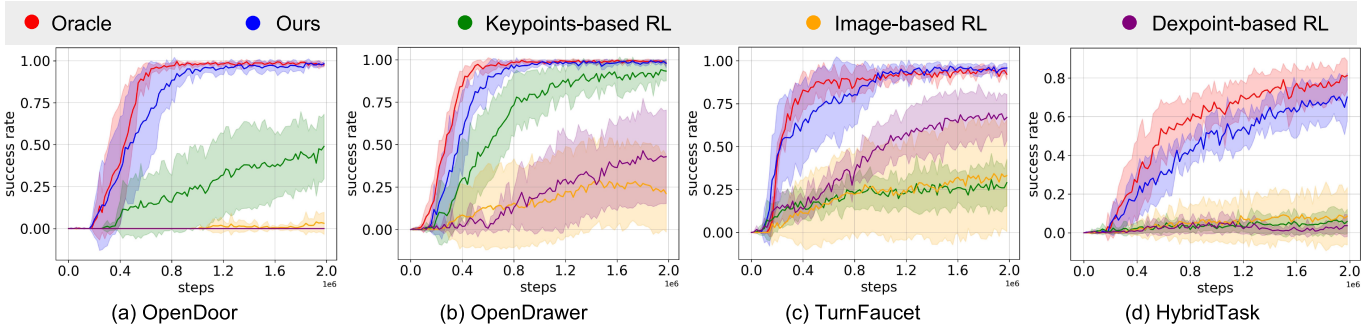


Fig. 5. Comparsions of our method with baselines on three single tasks and the *HybridTask* in simulation. The *HybridTask* involves all three categories of articulated objects. The results are averaged over 7 random seeds.

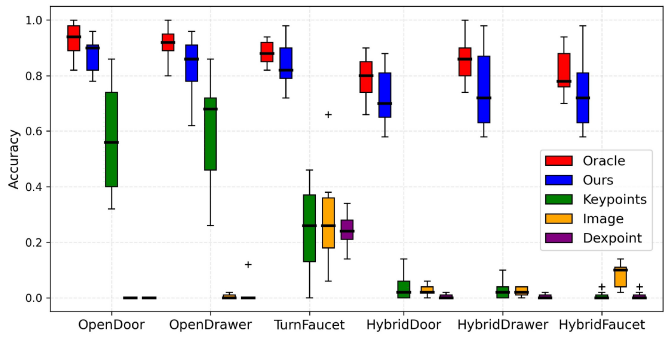


Fig. 6. Success rates of our method and baselines in simulation, averaged across 7 random seeds. *HybridDoor* represents the RL policy trained on the *HybridTask* and evaluated on the *OpenDoor* task. Similar meanings apply to *HybridDrawer* and *HybridFaucet*.

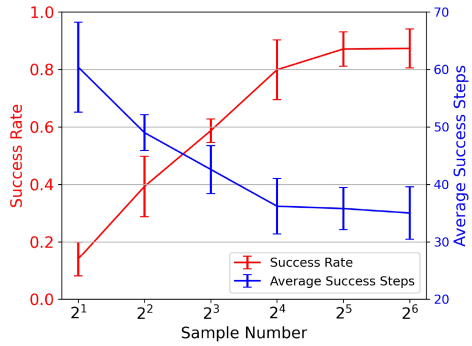


Fig. 7. Success rate (Red) and average success steps (black) for different numbers of sampling points on the *OpenDoor* task in simulation. The results are averaged over 7 random seeds.

Evaluation results is provided in Fig. 6. The results indicate that our method generalizes well on novel instances, not only for single tasks but also for the *HybridTask*. In contrast, the other baselines meet a significant performance drop facing novel instances.

Ablation Study on Sampling Number: To evaluate the impact of the sampling number N_s on our manipulation policy, we conduct an ablation study on the *OpenDoor* task. We train our method with different numbers of sampling points for each part while keeping all the other settings constant. The results are averaged over 7 random seeds.

In Fig. 7, we present the results of our ablation experiment on the *OpenDoor* task. The performance of our method improves with an increasing number of sampling points and then reaches a plateau after a certain number of points. However, note that using more points also consumes additional computing resources. In our work, we set N_s to 32 to strike a balance between performance and computational efficiency.

C. Sim-to-Real Transfer

We perform Sim2Real experiments to evaluate the performance of our proposed framework in the real world. *Part-guided Sampling Results:* Multiple sampling strategies based on the segmentation map are available. One common approach is Uniformly Downsampling used in 3D RL methods [27]. Accordingly, we downsample the scene point cloud to 1024 points in our experiments. Other strategies include Score-based Sampling, which selects the top N_s points from each part based on their semantic scores; FPS, which samples the farthest N_s points from each part; Random Sampling, which randomly samples N_s points from each part; and our proposed FUS strategy.

We compare various sampling methods and visualize the sampled points across three consecutive frames in a manipulation trajectory, as depicted in Fig. 8. The results indicate that uniformly-downsampled points may lack crucial points of small parts, such as handles, which can negatively affect the manipulation task. Score-based sampled points tend to be clustered in small circular regions, resulting in the loss of shape information of parts, and the positions of the sampled points vary rapidly across consecutive frames. Furthermore, none of the four alternative sampling methods, including FPS, Random Sampling, Score-based Sampling, and Uniformly Downsampling, can effectively handle segmentation errors and misclassified handle points, which can mislead the RL policy. In contrast, our proposed FUS strategy accurately and consistently samples points from each part, contributing to the stability of our RL policy.

Manipulation Results: To evaluate the effectiveness of our policy in the real world, we conduct trials on two doors, two drawers, and two faucets. These objects significantly vary in size, texture, and shape compared to the synthetic data. We perform 20 trials for each object, recording the success rate and average success steps. In each trial, we randomly position the object in

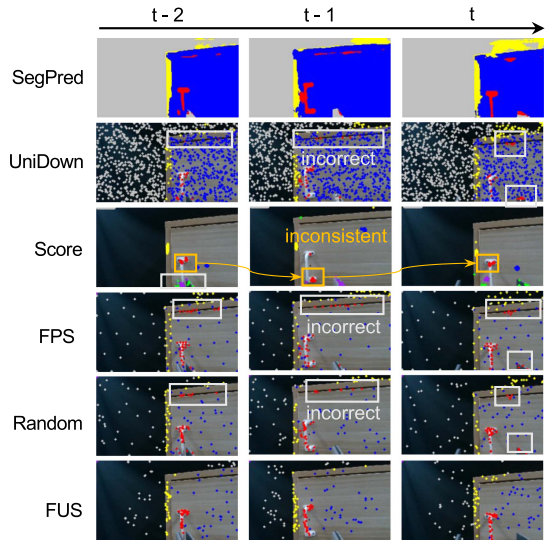


Fig. 8. The sampled points of different sampling methods on 3 consecutive real-world captured frames. The top row shows the segmentation predictions. The bottom rows show the sampled part points with different colors.

TABLE I
SUCCESS RATE AND AVERAGE SUCCESS STEPS IN THE REAL WORLD

Task		Dexpoint	Ours
Single	OpenDoor	0 / 40	**35 / 40**
		N/A	27.8 ± 3.2
	OpenDrawer	0 / 40	**32 / 40**
		N/A	28.3 ± 3.0
	TurnFaucet	8 / 40	**35 / 40**
		25.6 ± 3.6	19.9 ± 2.7
Hybrid	OpenDoor	0 / 40	**31 / 40**
		N/A	31.1 ± 3.8
	OpenDrawer	0 / 40	**27 / 40**
		N/A	30.1 ± 4.2
	TurnFaucet	0 / 40	**32 / 40**
		N/A	19.7 ± 4.5

front of the robot, ensuring the handle falls within the robot's operational space.

Table I reports the performance of our sampling-based 3D RL policy on the three articulated object manipulation tasks. Our framework achieves remarkable results on all three tasks. In particular, RL_{hybrid}, trained under the *HybridTask*, demonstrates great performance on novel instances. Despite utilizing the same domain randomization techniques, the Dexpoint-based RL method only attains a 20% success rate on the *TurnFaucet* task. This indicates that without specific part segmentation guidance, it faces challenges in manipulating articulated objects. Besides, both the Image-based RL and Keypoints-based RL methods exhibit a notable Sim2Real gap, resulting in failure to perform adequately in real-world scenarios.

To evaluate the effectiveness of our sampling strategy, we conduct ablation experiments on the *OpenDoor* task, and the results are presented in Table II. The results indicate that both uncertainty estimation and frame consistency contribute significantly to improving the performance of the framework.

TABLE II
THE SUCCESS RATE OF DIFFERENT SAMPLING STRATEGIES IN THE REAL WORLD

Methods	Success Rate
Random	24 / 40 (60.0 %)
FPS	26 / 40 (65.0 %)
FUS *w/o Uncertainty*	29 / 40 (72.5 %)
FUS *w/o Consistency*	31 / 40 (77.5 %)
FUS	**35 / 40 (87.5 %)**

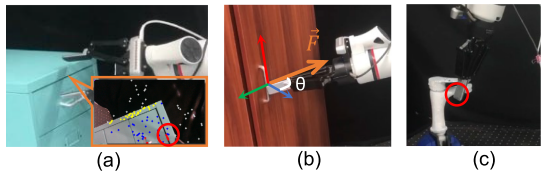


Fig. 9. Failure Cases: (a) Failing to approach due to incorrect sampled points; (b) Failing to pull due to inadequate force direction; (c) Failing to turn due to being stuck on the fixed link.

D. Limitations and Failure Analysis

Limitations: Our work assumes predefined part specifications for different types of objects, restricting its generality. Due to the hand-centric setting, we make the assumption that the handle can be seen in the initial frame. However, if the policy is misled by erroneous perception results and deviates from the correct path, it becomes challenging to relocate the handle. Additionally, our FUS strategy relies on the assumption that the object's rigid base remains stationary during the manipulation process, and it may fail if certain regions are consistently mis-segmented (e.g., the handle) in consecutive frames. Another limitation is that our method requires careful reward shaping to guide the learning process, which may not be easily extendable to more object types. Finally, as we introduce more types of articulated objects, our current training strategy becomes less effective and could benefit from improvements for more efficient training.

Failure Analysis: Some of the failure cases are shown in Fig. 9. In case (a), despite generally reliable segmentation results, the policy can be misled by erroneous segmentation of background objects as handles. In case (b), we found that inadequate grasping is a major cause of failure in our framework. Improper grasp points often result in loose contact between the gripper and the handle, leading to suboptimal manipulation behavior. To tackle this issue, incorporating additional sensing modalities, such as force-torque sensors, can provide force feedback and enhance the policy's accuracy. In case (c), the robot gripper occasionally collided with the fixed part of the object, such as the faucet base. To address this, optimizing the placement of the gripper and the hand-centric camera could be beneficial.

E. Analysis on the Versatile Policy

To explore the versatility of our RL framework, we incorporate two more types of articulated objects: laptops and kitchen pots. To exclude the influence of part segmentation errors, we employ oracle part segmentation in simulation to analyze the

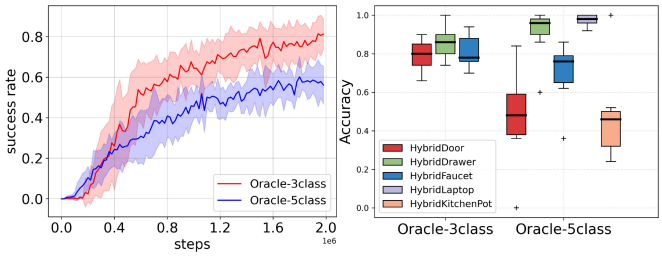


Fig. 10. Success rate of our RL policy on different types of articulated objects in the simulation. The results are averaged over 7 random seeds.

performance of the policy. The training and evaluation results are depicted in Fig. 10. On average, the success rate decreases by approximately 12% (from 82.1% to 70.3%), considering all classes. Despite a certain decline in performance, the results prove the potential of our versatile RL policy for a broader range of articulated object manipulation tasks.

V. Conclusion

In this letter, we propose an efficient part-guided 3D RL framework for articulated object manipulation tasks. The main contribution lies in the ability to learn a single versatile RL policy that can be applied across various tasks and directly deployed to novel real-world instances. Simulation and real-world experimental results demonstrate that our policy exhibits high accuracy and efficiency.

In future work, we plan to improve our part representation and extend our framework to a more general manipulation pipeline. In addition, we intend to refine the reward-tuning methods by adopting a more generalizable mechanism, which may minimize the design costs and make the overall process more efficient and adaptable.

References

[1] Y. Qin et al., "S4G: Amodal single-view single-shot SE(3) grasp detection in cluttered scenes," in *Proc. Conf. robot Learn.*, 2020, pp. 53–65.
[2] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11444–11453.
[3] K. Mo et al., "Where2act: From pixels to actions for articulated 3D objects," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 6813–6823.
[4] R. Wu et al., "VAT-mart: Learning visual action trajectory proposals for manipulating 3D ARTiculated objects," in *Proc. Int. Conf. Learn. Representations*, 2022.
[5] Y. Geng et al., "RLAfford: End-to-end affordance learning for robotic manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 5880–5886.
[6] Z. Xu, Z. He, and S. Song, "Universal manipulation policy network for articulated objects," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2447–2454, Apr. 2022.
[7] B. Eisner, H. Zhang, and D. Held, "FlowBot3D: Learning 3D articulation flow to manipulate articulated objects," in *Proc. Robotics: Sci. Syst.*, 2022.
[8] L. Ma et al., "Sim2Real^2: Actively building explicit physics model for precise articulated object manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 11698–11704.
[9] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building digital twins of articulated objects from interaction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5616–5626.
[10] Y. Wang et al., "Adaafford: Learning to adapt manipulation affordance for 3D articulated objects via few-shot interactions," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 90–107.
[11] Y. Urakami et al., "Doorgym: A scalable door opening environment and baseline agent," 2019, *arXiv:1908.01887*.
[12] J. Wang, S. Lin, C. Hu, and Y. Zhu, "Learning semantic keypoint representations for door opening manipulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6980–6987, Oct. 2020.
[13] L. Yen-Chen, A. Zeng, S. Song, P. Isola, and T.-Y. Lin, "Learning to see before learning to act: Visual pre-training for manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 7286–7293.
[14] I. Radosavovic et al., "Real-world robot learning with masked visual pre-training," in *Proc. Conf. Robot Learn.*, 2023, pp. 416–426.
[15] Y. Seo, K. Lee, S. L. James, and P. Abbeel, "Reinforcement learning with action-free pre-training from videos," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 19561–19579.
[16] A. Zhan, R. Zhao, L. Pinto, P. Abbeel, and M. Laskin, "Learning visual robotic control efficiently with contrastive pre-training and data augmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 4040–4047.
[17] Y. J. Ma et al., "VIP: Towards universal visual reward and representation via value-implicit pre-training," in *Proc. Int. Conf. Learn. Representations*, 2023.
[18] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," 2022, *arXiv:2203.06173*.
[19] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1/2, pp. 99–134, 1998.
[20] K. Hsu et al., "Vision-based manipulators need to also see from their hands," in *Proc. Int. Conf. Learn. Representations*, 2022.
[21] H. Geng et al., "GapartNet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7081–7091.
[22] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.
[23] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 567–576.
[24] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet RGB-D: Can 5 m synthetic images beat generic imagenet pre-training on indoor segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2678–2687.
[25] X. Zhang et al., "Close the optical sensing domain gap by physics-grounded active stereo sensor simulation," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 2429–2447, Jun. 2023.
[26] D. Yarats, I. Kostrikov, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *Proc. Int. Conf. Learn. Representations*, 2020.
[27] M. Liu et al., "Frame mining: A free lunch for learning robotic manipulation from 3D point clouds," in *Proc. Conf. Robot Learn.*, 2022.
[28] Y. Qin et al., "Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation," in *Proc. Conf. Robot Learn.*, 2023, pp. 594–605.
[29] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9277–9286.
[30] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.
[31] G. Wang et al., "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, 2019.
[32] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
[33] F. Xiang et al., "Sapien: A simulated part-based interactive environment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11097–11107.
[34] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.
[35] M. Sandler et al., "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
[36] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.