# Articulated Object Manipulation Using Online Axis Estimation with SAM2-Based Tracking

**Xi Wang**[*,2] **Tianxing Chen**[*,1,2] **Tianling Xu**[3] **Yiting Fu**[2]

**Ziqi He**[2] **BingYu Yang**[2] **Qiangyu Chen**[2] **KailunSu**[2]

**Abstract:**

We propose a closed-loop pipeline for articulated object manipulation. We first adopt any interactive perception technique to induce slight object movements, and track the evolving manipulation process. We then segment out the point cloud of the articulated object using Segment Anything Model 2, and estimate axes to guide subsequent robotic action. Experiments show that, our method outperforms solely interactive perception methods in tasks requiring precise axis-based control.

**Keywords:** Articulated Object Manipulation, Interactive Perception, Online Axis Estimation, Point-cloud Tracking and Segmentation.

## 1 Introduction

Robotic manipulation has a wide range of applications [1, 2, 3]. Among various manipulation tasks, those involving articulated objects (*e.g.*, doors and drawers) pose significant challenges [4]. Traditional methods often rely on predefined kinematic models [5, 6] or open-loop control [7, 8], which struggle to dynamically adapt to real-world interactions due to the absence of feedback regulation, resulting in inaccuracy and inefficiency [9].

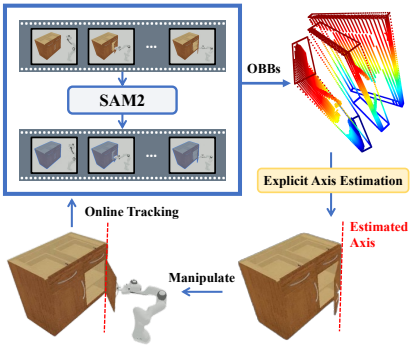


Figure 1: **The overview of our method.**

Recently, interactive perception has emerged as a promising approach to address these challenges [10, 8, 11, 12]. By actively interacting with the object, robots can gather real-time sensory data that provides insights into the structure and kinematics of objects. While these methods effectively provide information about the object's state, they overlook the evolving dynamic interactions between the robot and the articulated object over time. This omission limits the robot's ability to adapt its manipulation policy in real time as the object's state changes, leading to inefficiencies in tasks that require precise control.

To address this limitation, we propose a novel closed-loop pipeline that enhance interactive perception with online refined axis estimation, providing the robot with helpful guidance for axis-aware manipulation. The basic idea is illustrated in Fig. 1. Specifically, we leverage any interactive perception technique (*e.g.*, RGBManip [12]) to induce slight object movements and generate dynamic 3D point cloud frames. These frames are then processed using an advanced segmentation pipeline, with Grounding DINO [13] as object detector and Segment Anything Model 2 (SAM2) [14] for segmentation, which identifies and isolates the point cloud of articulated objects. By masking out the moving components of the object, we can explicitly calculate the motion axis with the oriented bounding boxes (OBBs), which in turn informs the robot's subsequent manipulation action.

Experiments show that, our method significantly outperforms open-loop methods and pure interactive perception techniques, enhancing the accuracy of manipulation tasks involving articulated

[1]The University of Hong Kong; [2]Shenzhen University; [3]Southern University of Science and Technology
[*]Equal Contribution; Project page: https://hytidel.github.io/video-tracking-for-axis-estimation/

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

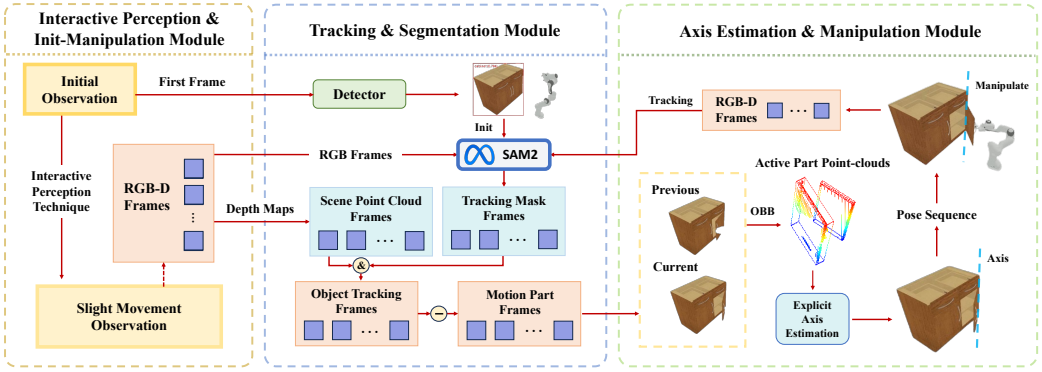


Figure 2: **Our pipeline.** An RGB-D camera captures the dynamic scene induced by the slight movement from the IPIM Module. The captured scene is then processed by the TS Module, which tracks and segments the moving part of the articulated objects. The segmented data is subsequently fed to the AEM Module for explicit axis estimation, which guides the robot's subsequent policy.

objects. Our main contributions: (i) Utilization of 2D foundation models for 3D point cloud-based manipulation; (ii) Closed-loop integration of interactive perception and online axis estimation; (iii) An explicit online motion axis estimation method based on part-level iterative dynamics.

## 2 Related Work

### 2.1 Vision-based Robotic Manipulation

Vision-based robotic manipulation heavily relies on modalities of perception, with each modality exhibiting distinct advantages and limitations. For example, while RGB images offer rich texture, they lack depth information crucial for grasping [5]. To address this, some methods combine RGB with depth maps [15, 16], though RGB-D data often suffers from noise on transparent or reflective surfaces [17]. Unlike approaches that switch across modalities [18, 19, 20], RGBManip [12] uses only multi-view RGB to estimate 6D object poses. However, its single end-effector camera can miss key interaction dynamics. Besides, point cloud-based methods, such as Where2Act [21], SAGCI-System [22] and Flowbot3D [23], excel at articulating action affordances.

Our approach strives to integrate the strengths of RGB-based methods and point cloud-based techniques by tracking RGB inputs and segmenting the point clouds corresponding to the motion part of articulated objects. In order to address the limitation of open-loop methods, we integrate an additional RGB-D camera into the scene, specifically aimed at capturing the manipulation process.

### 2.2 3D Point Cloud Segmentation Empowered by 2D Foundation Models

Image segmentation techniques (*e.g.*, Segment Anything Model (SAM) [24]) have been widely used for point cloud segmentation in robotic manipulation. For example, DeepLabv3 [25] adapts image segmentation to 3D segmentation by projecting point clouds onto 2D planes [26], which is robust at scale but loses accuracy due to projection. However, transferring video segmentation techniques (*e.g.*, SAM2 [14]) to 3D point cloud processing remains underexplored. We argue that, video segmentation models preserve more motion cues, which is more suitable for the dynamic manipulation process. Therefore, we track the manipulation process of interactive perception with SAM2 , and segment out the motion parts of articulated objects for precise control.

### 2.3 Axis Estimation for Articulated Objects

Axis estimation is critical for understanding and manipulating articulated objects, where interactive perception is widely adopted to mitigate the ambiguity inherent in single observation [27]. For example, Martín Martín and Brock [28] presented an online interactive perception system based on

task-specific priors to extract kinematic and dynamic models of articulated objects. However, previous studies on axis estimation are typically open-loop, which overlook the interaction dynamics. To address this issue, Karayiannidis et al. [29] proposes a close-loop method that utilizes force/torque sensor measurements to estimate the motion direction and the orientation of the axis, inferring the type of joint without prior knowledge of the object's kinematics. We also employ closed-loop axis estimation, but leverage the geometric prior of motion to explicitly compute the axis of the joints.

# 3 Method

We introduce a novel pipeline for articulated object manipulation, guided by explicit axis estimation derived from SAM2-based tracking. As depicted in Fig. 2, our pipeline consists of three core modules: (i) Interactive Perception & Init-Manipulation (IPIM) Module, (ii) Segmentation & Tracking (ST) Module and (iii) Axis Estimation & Manipulation (AEM) Module.

## 3.1 Interactive Perception & Init-Manipulation Module

We employ any interactive perception methods to grasp the handle and produce slight displacements to the objects. Specifically, we incorporate RGBManip [12] as the default implementation for this module. It is worth noting that, RGBManip employs the original SAM-based multi-view object pose estimation for detecting and grasping the handle. As RGBManip initiates the manipulation of articulated objects, an additional camera periodically records dynamic RGB-D data. The captured RGB data will be leveraged for subsequent segmentation and tracking tasks, while the corresponding depth data reconstructs the point cloud of the evolving dynamic scene.

## 3.2 Segmentation & Tracking Module

We utilize Grounding DINO [13] on the initial RGB frame to generate an anchor box for the target object using text prompt (*e.g.*, "`cabinet`"). The first RGB frame serves as the starting point for SAM2's tracking process [14]. The anchor box of the target object is fed into SAM2 as a box prompt, enabling it to continuously track the object and provide a mask for it throughout each moment of the dynamic manipulation process. We then segment out the portion of point cloud that represents the object from the dynamic scene with the derived masks.

We filter the raw object point cloud $P = \{p_i\}_{i=1}^{n}$ obtained above to remove outliers, ensuring that the filtered point cloud represents the region of interest of articulated object. Specifically, a point $p \in P$ is retained if and only if $|U^{\circ}(p, r)| \geq \epsilon$, where $\epsilon > 0$ is a threshold, $|\cdot|$ is the number of element in the set, and $U^{\circ}(p, r) = \{q \mid q \in P, 0 < |p - q| \leq r\}$.

We then compute the OBB for the entire articulated object on the initial frame. Subsequently, by subtracting this OBB from each frame's point cloud and applying another round of noise reduction, we obtain the point cloud representing the moving parts of the articulated objects. For implementation, the segmentation can be refined by removing the protruding handle from the cabinet's OBB to obtain a tighter OBB of the cabinet's body, leading to a more precise motion-part segmentation.

## 3.3 Axis Estimation & Manipulation Module

We calculate the axis of motion based on the segmented point cloud representing the moving parts of the articulated object identified by the ST Module. Task-specific geometric priors play an essential role in achieving accurate axis estimation: (i) for a prismatic joint, the moving parts translate along the axis; (ii) for a revolute joint, the moving parts rotate around the axis. By leveraging these task-based priors, we are able to explicitly calculate the joint's axis.

Specifically, consider a single action of robotic manipulation, where the OBBs of the initial and final point cloud of the motion components of the object are denoted as $obb_{\text{st}}$ and $obb_{\text{ed}}$, with their centers designated as $O_{\text{st}}$ and $O_{\text{ed}}$ respectively. Leveraging the geometric priors, we have: (i) **Prismatic joint:** The axis pivot is defined as $O_{\text{st}}$, and the axis direction is estimated as the direction $\overrightarrow{d} =$

Table 1: **Quantitative comparison between our method and baselines.**

| Methods | Modality | Open Door 8.6° | | Open Door 45° | | Open Drawer 15 cm | | Open Drawer 30cm | |
|---|---|---|---|---|---|---|---|---|---|
| | | Train | Test | Train | Test | Train | Test | Train | Test |
| DrQ-v2 [30] [1] | RGB | 1.8 | 2.5 | 0.8 | 0.8 | 1.9 | 1.0 | 1.4 | 0.5 |
| LookCloser [31] [1] | RGB | 1.5 | 1.25 | 0.8 | 0.8 | 0.8 | 0.0 | 0.0 | 0.0 |
| RGBManip [12] [2] | RGB | 75.0 | 82.0 | 47.0 | 47.0 | 56.0 | 64.0 | 46.0 | 45.0 |
| Where2Act [21] [1] | PCD | 8.0 | 7.0 | 1.8 | 2.0 | 5.9 | 7.5 | 1.1 | 0.6 |
| Flowbot3D [23] [1] | PCD | 19.5 | 20.4 | 6.8 | 6.4 | 27.3 | 25.8 | 16.9 | 11.3 |
| UMPNet [15] [1] | PCD | 27.1 | 28.1 | 11.0 | 10.9 | 16.6 | 18.8 | 4.4 | 5.6 |
| GAPartNet [32] [1] | PCD | 69.5 | 74.5 | 39.4 | 43.6 | 50.6 | 59.3 | 44.6 | 48.6 |
| **Ours** | RGB + PCD | **87.0** | **88.0** | **54.0** | **54.0** | **68.0** | **85.0** | **59.0** | **68.0** |

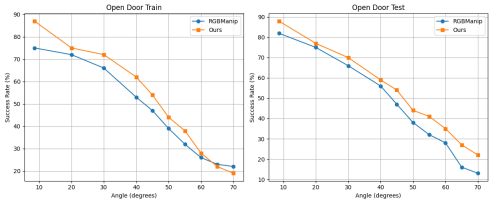


Figure 3: **Success rates of more challenging door-opening tasks.**

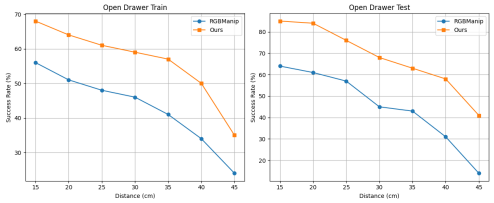


Figure 4: **Success rates of more challenging drawer-opening tasks.**

$\overrightarrow{O_{\text{st}}O_{\text{ed}}}$; (ii) **Revolute joint:** The axis pivot point is derived by identifying the intersection of the mid-perpendiculars along the longer edges in the top-down view of $obb_{\text{st}}$ and $obb_{\text{ed}}$. With the axis point $P$ established, the axis direction is ascertained by the sign of the dot product $\overrightarrow{d} \cdot \overrightarrow{t}$, where $\overrightarrow{t} = \overrightarrow{O_{\text{st}}P} \times \overrightarrow{z}$, among which $\overrightarrow{z}$ represents the positive direction of the z-axis.

We further propose an online axis estimation refinement predicated on an observation: as the manipulation of the joint assembly increases, the point cloud of its moving component becomes progressively more amenable to accurate reconstruction. The enhancement in the fidelity of the point cloud data, in turn, furnishes the axis estimation process with inputs that are increasingly precise and reliable. For implementation, after each manipulation, we invoke the ST Module and the Axis Estimation Module to ascertain the current axis estimation, in which the window of frame indices $[\text{st}, \text{ed}]$ is progressively shifted as the process unfolds, while maintaining an appropriate length of the interval $[\text{st}, \text{ed}]$. The Manipulation Module then guides the robot's subsequent actions according to the latest axis estimation. The interactive process reiterates until the robot has executed all designated actions, with the axis estimation being continuously refined throughout the procedure.

# 4 Experiment

## 4.1 Experimental Settings

We conduct experiments on 1-DoF doors and drawers. The task settings: (i) **Door-Opening:** open the door larger than $8.6° \sim 70°$; (ii) **Drawer-Opening:** open the drawer larger than 10 cm $\sim$ 45 cm.

We benchmark our methods against RGBManip [12] and other baselines (Appx. A.1) on RGBManip's training and testing set separately, and evaluate the success rates of the first 100 experiments. All methods are compared under equivalent total step sizes, though different methods may allocate step sizes differently based on their policies. Refer to Appx. A.2 for details.

## 4.2 Quantitative Results

**Basic Tasks.** Quantitative results of basic tasks in Tab. 1 show that, both our method and RGBManip almost outperform other baseline approaches, while ours consistently surpasses RGBManip.

[1] Experimental results derived from TABLE I in [12].

[2] Results reproduced using RGBManip under the settings outlined in Sec. 4.1.

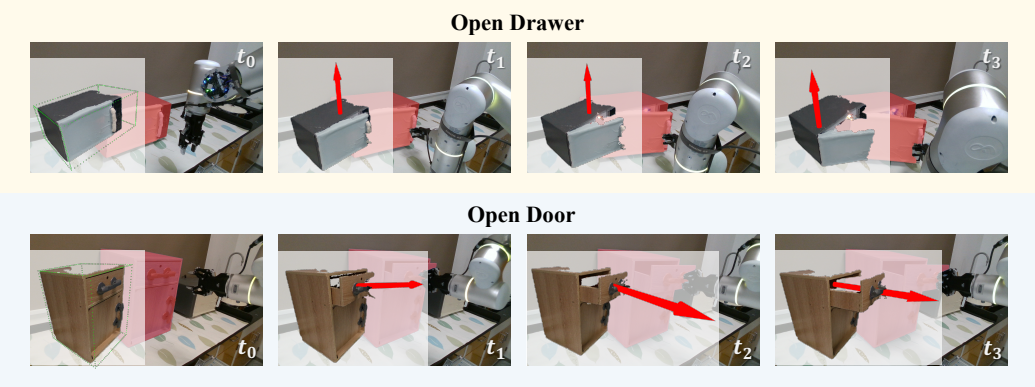


Figure 5: **Visualization of axis estimation for real-world deployment.** The initial moment and the three manipulation moments are shown, with visualization of the RGB tracking obtained from SAM2 (background), the reconstructed point cloud of the target object (bottom-left corner), the OBBs (green dashed-line boxes at $t_0$) and the axis (red arrows) estimated with our method.

**More Challenging Tasks.** We further evaluate more challenging tasks with larger opening ranges. The line charts of success rates for door-opening and drawer-opening tasks shown in Fig. 3 and Fig. 4 demonstrate that, ours consistently outperforms RGBManip with a significant enhancement in success rates, showing robustness across various types of cabinets. It indicates that, ours is more qualified for tasks requiring precise axis-based control. Refer to Appx. A.3 for granular results.

### 4.3 Real-World Deployment

To validate the effectiveness of our method in real-world deployment, we demonstrate the complete axis estimation process for the door-opening and drawer-opening tasks. Empirical results shown in Fig. 5 demonstrate that, our method remains robust despite the presence of noise in real-world point clouds. We attribute this to our point cloud augmentation method, which enables robust computation of OBBs, leading to accurate and reliable axis estimation in noisy scenarios.

### 4.4 Analysis and Discussion

**Robustness to Segmentation Quality.** During execution, from the scene RGBD camera's viewpoint, the robotic arm may occlude part of the cabinet, causing the mask to fragment into multiple components. We empirically found that, our method still produces accurate axis estimation even with incomplete point cloud segments due to robust OBB estimation, highlighting its robustness. Analogous robust segmentation is also observed when we vary real-world lighting conditions.

**Efficiency.** SAM2 reuses the per-frame processing outcomes from preceding frames, ensuring that each frame is processed only once. In addition, we deploy a sliding temporal window that obviates repeated segmentation and processing. These strategies synergisticly alleviate computational overhead. Refer to Appx. A.4 for more analysis and discussion.

## 5 Conclusion

We present a novel closed-loop pipeline for articulated object manipulation that integrates interactive perception with online axis estimation. By actively manipulating the object and tracking the evolving scene with SAM2, we segment out the motion components of the articulated object, followed by an explicit online-refined axis estimation. Experiments demonstrate the superiority of integrating axis estimation for more accurate and efficient manipulation, and indicate the promising potential to employ 2D foundation models for efficient 3D-based manipulation without 3D foundation models.

**Acknowledgments**

We thank Dr. Yao Mu and Dr. Qiaojun Yu for their invaluable advice. We also thank all reviewers for their useful comments.

## References

[1] S. Afrin, S. Roksana, and R. Akram. Ai-enhanced robotic process automation: A review of intelligent automation innovations. *IEEE Access*, 2024.

[2] P. Picozzi, U. Nocco, C. Labate, I. Gambini, G. Puleo, F. Silvi, A. Pezzillo, R. Mantione, and V. Cimolin. Advances in robotic surgery: A review of new surgical platforms. *Electronics*, 13 (23):4675, 2024.

[3] E. O. Sodiya, U. J. Umoga, O. O. Amoo, and A. Atadoga. Ai-driven warehouse automation: A comprehensive review of systems. *GSC Advanced Research and Reviews*, 18(2):272–282, 2024.

[4] H. Xiong, R. Mendonca, K. Shaw, and D. Pathak. Adaptive mobile manipulation for articulated objects in the open world. *arXiv preprint arXiv:2401.14403*, 2024.

[5] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu. Rgb matters: Learning 7-dof grasp poses on monocular rgbd images. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13459–13466. IEEE, 2021.

[6] W. Gao and R. Tedrake. kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6527–6533. IEEE, 2021.

[7] J. Wang, S. Lin, C. Hu, Y. Zhu, and L. Zhu. Learning semantic keypoint representations for door opening manipulation. *IEEE Robotics and Automation Letters*, 5(4):6980–6987, 2020.

[8] 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. In *International Conference on Learning Representations*, 2022.

[9] Y. Wang, X. Zhang, R. Wu, Y. Li, Y. Shen, M. Wu, Z. He, Y. Wang, and H. Dong. Adamanip: Adaptive articulated object manipulation environments and policy learning. *arXiv preprint arXiv:2502.11124*, 2025.

[10] Z. Jiang, C.-C. Hsu, and Y. Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5616–5626, 2022.

[11] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme. Active articulation model estimation through interactive perception. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3305–3312. IEEE, 2015.

[12] B. An, Y. Geng, K. Chen, X. Li, Q. Dou, and H. Dong. Rgbmanip: Monocular image-based robotic manipulation through active object pose estimation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7748–7755. IEEE, 2024.

[13] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[14] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[15] Z. Xu, Z. He, and S. Song. Universal manipulation policy network for articulated objects. *IEEE Robotics and Automation Letters*, 7(2):2447–2454, Apr. 2022. ISSN 2377-3774. doi: 10.1109/lra.2022.3142397. URL http://dx.doi.org/10.1109/LRA.2022.3142397.

[16] C. Wu, J. Chen, Q. Cao, J. Zhang, Y. Tai, L. Sun, and K. Jia. Grasp proposal networks: An end-to-end solution for visual learning of robotic grasps, 2020. URL https://arxiv.org/abs/2009.12606.

[17] S. S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song. Cleargrasp: 3d shape estimation of transparent objects for manipulation, 2019. URL https://arxiv.org/abs/1910.02550.

[18] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg. Dex-nerf: Using a neural radiance field to grasp transparent objects, 2021. URL https://arxiv.org/abs/2110.14217.

[19] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. Wang. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf, 2023. URL https://arxiv.org/abs/2210.06575.

[20] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu, C. Yang, D. Wang, Z. Chen, X. Long, and M. Wang. Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping, 2024. URL https://arxiv.org/abs/2403.09637.

[21] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani. Where2act: From pixels to actions for articulated 3d objects, 2021. URL https://arxiv.org/abs/2101.02692.

[22] J. Lv, Q. Yu, L. Shao, W. Liu, W. Xu, and C. Lu. Sagci-system: Towards sample-efficient, generalizable, compositional, and incremental robot learning, 2022. URL https://arxiv.org/abs/2111.14693.

[23] B. Eisner, H. Zhang, and D. Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects, 2024. URL https://arxiv.org/abs/2205.04382.

[24] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

[25] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation, 2017. URL https://arxiv.org/abs/1706.05587.

[26] T. Chen, Y. Mu, Z. Liang, Z. Chen, S. Peng, Q. Chen, M. Xu, R. Hu, H. Zhang, X. Li, et al. G3flow: Generative 3d semantic flow for pose-aware and generalizable object manipulation. *arXiv preprint arXiv:2411.18369*, 2024.

[27] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, Dec. 2017. ISSN 1941-0468. doi:10.1109/tro.2017.2721939. URL http://dx.doi.org/10.1109/TRO.2017.2721939.

[28] R. Martín Martín and O. Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors, 2014.

[29] Y. Karayiannidis, C. Smith, F. E. V. Barrientos, P. Ögren, and D. Kragic. An adaptive control approach for opening doors and drawers under uncertainties. *IEEE Transactions on Robotics*, 32(1):161–175, 2016.

[30] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning, 2021. URL https://arxiv.org/abs/2107.09645.

[31] R. Jangir, N. Hansen, S. Ghosal, M. Jain, and X. Wang. Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation, 2022. URL https://arxiv.org/abs/2201.07779.

[32] 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, 2023. URL https://arxiv.org/abs/2211.05272.

# Appendix

## A Supplementary to Experiments

### A.1 Introduction to Baselines

- **DrQ-v2 [30]:** Based on reinforcement learning (RL), it takes in the robot's state and RGB image to determine the desired 6D pose of the robot's end-effector;
- **LookCloser [31]:** A multi-perspective RL model that leverages multi-view inputs and visual transformers to amalgamate data from various angles;
- **RGBManip [12]:** Utilizes RGB-only visual input to directly estimates the 6D pose of objects from multi-view images;
- **Where2Act [21]:** Processes point-cloud data to estimate the best point of interaction for manipulation;
- **Flowbot3D [23]:** Predicts point-wise motion, or "flow", within the point cloud. The point with the highest motion magnitude is selected for interaction;
- **UMPNet [15]:** Utilizing RGB-D images, it predicts an action point in the image and projects it into 3D space using depth data;
- **GAPartNet [32]:** A pose-centric method which predicts the pose of an object's part from point-cloud data.

### A.2 Supplementary to Experimental Settings

In each task, the articulated objects (placed with limited random position and rotation) begins in a closed state. The robotic arm is initially positioned randomly in front of it, and must accomplish the designated manipulation goal — either opening the drawer or door to a specific degree or range.

To ensure fairness in comparison, all RGBManip components employ task-specific *adapose* as the pose estimator and *heuristic pose* as the controller among different tasks.

If the axis estimation fails, the action predicted in the previous step is repeated. If there is no last action, re-perform initial manipulation until step sizes are exhausted.

For simulation experiments, we set $r = 0.05$ and $\epsilon = 100$ for point cloud augmentation. For real-world deployment, we employ the D415 depth camera, and set $r = 1.3$ and $\epsilon = 1$, due to the sparsity of the point cloud reconstructed from the real-world depth map.

### A.3 Supplementary to Quantitative Results

Granular results for more challenging door-opening and drawer-opening tasks in Sec. 4.2 are presented in Tab. 2 and Tab. 3, respectively.

Table 2: **More challenging tasks for door-opening.**

| **Methods** | **20°** | | **30°** | | **40°** | | **50°** | |
|---|---|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test | Train | Test |
| RGBManip | 72.0 | 75.0 | 66.0 | 66.0 | 53.0 | 56.0 | 39.0 | 38.0 |
| **Ours** | **75.0** | **77.0** | **72.0** | **70.0** | **62.0** | **59.0** | **44.0** | **44.0** |
| **Methods** | **55°** | | **60°** | | **65°** | | **70°** | |
| | Train | Test | Train | Test | Train | Test | Train | Test |
| RGBManip | 32.0 | 32.0 | 26.0 | 28.0 | **23.0** | 16.0 | **22.0** | 13.0 |
| **Ours** | **38.0** | **41.0** | **28.0** | **35.0** | 22.0 | **27.0** | 19.0 | **22.0** |

Table 3: **More challenging tasks for drawer-opening.**

| Methods | 20 cm | | 25 cm | | 35 cm | | 40 cm | | 45 cm | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test | Train | Test | Train | Test |
| RGBManip | 51.0 | 61.0 | 48.0 | 57.0 | 41.0 | 43.0 | 34.0 | 31.0 | 24.0 | 14.0 |
| **Ours** | **64.0** | **84.0** | **61.0** | **76.0** | **57.0** | **63.0** | **50.0** | **58.0** | **35.0** | **41.0** |

### A.4 More Analysis and Discussion

The analysis section delves into the specific experimental results to elucidate the practical advantages of our online axis estimation approach in the context of articulated object manipulation. Our method's performance is grounded in the empirical data obtained from the experiments, which are analyzed below to provide a detailed understanding of the improvements achieved.

**Online Axis Estimation *vs.* Traditional Methods.** Our experiments clearly demonstrate the superiority of our online axis estimation approach over traditional methods, especially in tasks that demand precise control such as door and drawer opening. As illustrated in Tab. 1, our method achieves an impressive 87.0% success rate in the training set and 88.0% in the test set for the "Open Door" task, significantly outperforming RGBManip, which records 75.0% and 82.0% respectively. This notable enhancement stems directly from the continuous refinement of the axis estimation, empowering the robot to adapt its actions in response to the most current interaction dynamics.

Furthermore, in the more demanding versions of these tasks, depicted in Fig. 3 and Fig. 4, our method's reliance on ongoing online axis estimation reveals enhanced robustness in handling large-amplitude movements of articulated objects. The experimental data consistently show that, our method sustains higher success rates as the complexity of tasks escalates. The online axis estimation process is pivotal in this regard, enabling our system to dynamically adjust to real-time interaction feedback and ensuring precise control over articulated objects. The continuous axis refinement, aligned with the object's changing state, is essential for the manipulation's accuracy and efficiency, particularly in tasks with substantial state variations. This robust performance underscores the strength of online axis estimation in providing reliable and responsive control in sophisticated robotic manipulation scenarios.

**Consistency Across Various Manipulation Scenarios.** Our method's consistent performance across a range of manipulation scenarios highlights its robustness and versatility. The success rates in both door and drawer opening tasks, as depicted in Tab. 2 and Tab. 3, consistently show higher rates for our method, indicating that the online axis estimation is effective regardless of the specific manipulation task. This consistency is a significant advantage over methods that may perform well in one scenario but falter in others.

## B Limitations and Future Work

We point out the following limitations:

1. Our method relies on the quality of SAM2 segmentation. It is necessary to fully explore the long-term reliability and scalability of this approach across a broader spectrum of tasks and environments, *e.g.*, under occlusions or poor lighting;
2. The performance of our method is affected by the initial displacement induced by the initial manipulation module. Too small opening amplitude of the first attempt may lead to inaccurate estimation of the first axis;
3. Although bypassing the need for 3D foundation models, our method still suffers from heavy computational cost of continuous point cloud processing. More efficient pipelines for real-time deployment are expected;
4. Future works are expected to generalize task-specific prior to higher DoF articulated objects.