

Supplementary Material for Revisiting Articulated Parts Perception in Robot Manipulation

Xiaoqian Wu, Yejie Guo, Xiaoyang Chen, Lixin Yang, Cewu Lu*, Yong-Lu Li*

Shanghai Jiao Tong University

{enlighten, gyj123, cxy_computer, siriusyang, lucewu, yonglu_li}@sjtu.edu.cn

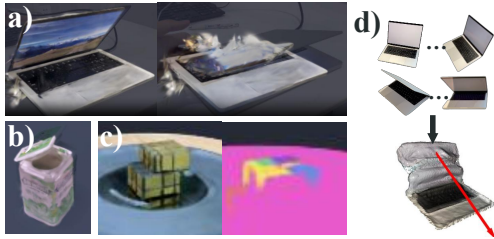


Figure 1. Failure cases for post-processing methods RSRD [5] and ArtGS [8], which are prone to errors.

1. Detailed Comparison with Existing Works

For pose-based representation, post-processing methods have emerged to reconstruct articulated objects from visual inputs. However, our method has unique advantages.

RSRD [5] uses a 4D differentiable part model to recover object motions from an object scan and a single monocular video. It is time-consuming. Reconstruction takes about 40 minutes on a single 3090 GPU following its official code, and pose estimation (10 minutes) is needed for each interaction sequence. Each time we interact with the object in another environment or camera view, we need to re-run pose estimation (10 minutes). Furthermore, this optimization-based method is prone to errors, *e.g.*, Fig. 1 (a) self-occlusion; (b) initial frame error; (c) segmentation error. Instead, our VR-GPS are quick and ensure quality with manual annotation.

There are other post-processing methods. Ditto[4] fails to process most of the VR-GPS objects, *e.g.*, book, lamp, as it is trained only on 8 categories and needs to train a network for each category. PARIS[7] and ArtGS[8] leverages neural radiance fields and 3D Gaussians to reconstruct objects and estimate joints. They excel in synthetic objects, but fail to estimate joints in real-world scenes (Fig. 1(d)) and take an extra 20 minutes to manually align two states.

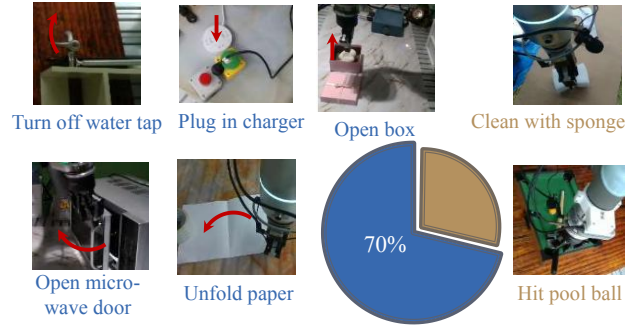


Figure 2. A large proportion of current robot tasks are related to object geometric structure. Tasks with precise force control, *e.g.*, hit a pool ball, are out of scope of this work.

2. Detailed Dataset Statistics

VR-GPS is developed in Unity and deployed on a Meta Quest 3 device, based on the existing work [1]. The virtual point coordinate is in the world frame determined during each initial configuration. During interaction, the relative transformation of the world frame and the headset is recorded. With the fixed transformation of the headset and the RealSense camera, the virtual point coordinate can finally be mapped to the camera frame. As an intermediate frame, the world frame can be located anywhere within the boundary. We also provide VR recording videos as an attachment in the supplementary material.

To collect VR-GPS dataset, we invite 8 volunteers to annotate data wearing a headset, and another 3 volunteers to check. The collected dataset has six part classes: Lid (89 objects), Lid-thin (21 objects), Lid-book (32 objects), Handle (34 objects), Door (33 objects), Drawer (25 objects). A large proportion of current robot tasks are related to the object’s geometric structure. As is illustrated in Fig. 2, among 89 complex tasks in RH20T [2], there are 70% tasks requiring geometric structure knowledge, *e.g.*, unfolding paper, plugging in a charger.

3. Geometric Structure Learning

3.1. Benchmark Details

We evaluate the model on two external datasets: HOI4D and RGBD-Art. HOI4D has 1.2K frames for Laptop, 1.4K frames for Trashcan, 2.9K frames for Safe, 0.4K frames for Bucket, 2.8K frames for Drawer. RGBD-Art has 1.1K frames for Laptop, 0.6K frames for Trashcan, 0.5K frames for Safe, 1.4K frames for Bucket, 1.4K frames for Drawer.

3.2. Implementation Details

We train our model on 2 NVIDIA H100 GPUs for a total of 100 epochs, using a batch size of 16. The initial learning rate is set to 0.0001, using a warm-up scheduler for gradual increase at the start of training. Input images are cropped and resized to 640×640 resolution, and point clouds are randomly sampled to 24,576 points before being processed by the network.

3.3. Transform Flow into GPS

We transform flow prediction into GPS for comparison under the same metric. We first sample 1024 points on the object's surface using Farthest Point Sampling (FPS) and predict their trajectories. To ensure quality and filter out static parts, we select $K = 256$ trajectories with the largest total displacements. Then the GPS is extracted as follows: For revolute objects, the rotation axis direction \mathbf{u} is computed via Principal Component Analysis (PCA) on all motion vectors $\{\mathbf{d}_{j,t}\}$, corresponding to the eigenvector with the smallest eigenvalue:

$$\mathbf{u} = \arg \min_{\|\mathbf{v}\|=1} \sum_{j,t} (\mathbf{v} \cdot \mathbf{d}_{j,t})^2, \quad (1)$$

Second, we determine a point on the axis, \mathbf{q} , using a per-trajectory voting scheme. For each of the K selected trajectories \mathcal{T}_j , we estimate an axis position candidate \mathbf{c}_j by finding the common intersection of internal motion-perpendicular lines. The final axis point is the mean of these candidates:

$$\mathbf{q} = \frac{1}{K} \sum_{j=1}^K \mathbf{c}_j, \quad \text{where } \mathbf{c}_j = \mathcal{F}(\mathcal{T}_j, \mathbf{u}). \quad (2)$$

For prismatic joints, the axis direction is instead the eigenvector with the largest eigenvalue, as motion is parallel to the axis. The axis position is the average of trajectory centers projected onto the perpendicular plane.

3.4. Transform Part Pose into GPS

We transform part pose into GPS for comparison under the same metric. With the predicted part segmentation and NPCS [6] map, we apply RANSAC [3] for outlier removal and Umeyama algorithm [9] to obtain part bounding box, and then calculate GPS from bounding box coordinates, as is shown in Fig. 3.

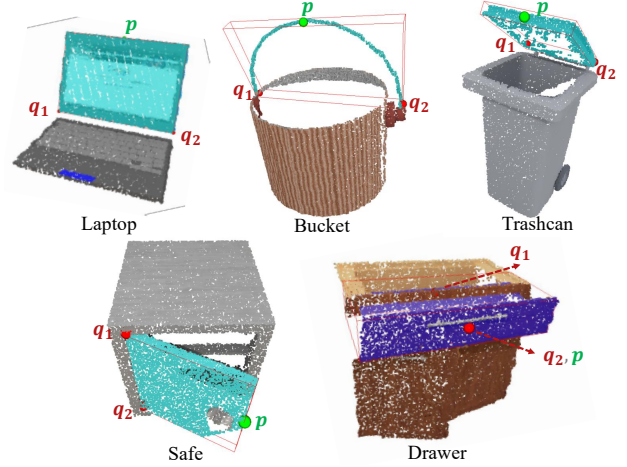


Figure 3. Object part pose and corresponding GPS.



Figure 4. The test objects in real robot experiments. We show a random view for each object.

4. Real Robot Experiments

4.1. Heuristic Policy

We test on 9 objects with diverse appearances. Their categories and part classes are: Box (Lid), Document-Box (Lid), Bucket (Handle), Door (Door), Drawer (Drawer), Notebook (Lid-book), Folder (Lid-book), Lamp (Lid-thin), Clapperboard (Lid-thin). We show a random view for each object in Fig 4.

The GPS-based heuristic policy is shown in Alg. 1. To select \mathbf{G} , GPS predictions are used for a scoring func-

Algorithm 1 Heuristic Policy

Input: Object point cloud $\mathcal{P} \in \mathbb{R}^{N \times 3}$; Time step t ;

Output: Planned robot trajectory

```
1:  $\mathcal{G} \leftarrow \text{AnyGrasp}(\mathcal{P})$ 
2:  $\{\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \hat{\mathbf{p}}\} \leftarrow \text{GPS}(\mathcal{P})$ 
3:  $\mathbf{G}, \mathbf{T}_1 \leftarrow \arg \max_{\{\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \hat{\mathbf{p}}\}}(\mathcal{G})$ 
4: for time step  $t \leftarrow 1$  to  $t$  do
5:   if Revolute joint then
6:      $\mathbf{T}_{t+1} \leftarrow \text{Rot}(\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2) \cdot \mathbf{T}_t$ 
7:   else if Prismatic joint then
8:      $\mathbf{T}_{t+1} \leftarrow \text{Trans}(\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2) \cdot \mathbf{T}_t$ 
9:   end if
10: end for
11: return  $\{\mathbf{T}_t\}_{t=1}^t$ 
```

tion \mathcal{S} of grasp proposals. For objects with revolute joint, one criterion is the angle between $\{\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \hat{\mathbf{p}}\}$ plane and $\{\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \mathbf{o}\}$ plane, where \mathbf{o} is the position of a grasp. The angle and the original grasp confidence scores are processed with z-score normalization. The final score is their weighted sum, with the coefficients 1.0, 0.25. For objects with a prismatic joint, the criterion is the distance from \mathbf{o} to the $\{\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \hat{\mathbf{p}}\}$ plane. The coefficients of the distance and the original grasp confidence scores are 1.0, 0.5.

We also provide robot manipulation videos as an attachment in the supplementary material.

4.2. Combination with Diffusion Policy

We conduct a small-scale experiment with our GPS-Policy base on RISE [10], a diffusion policy model with point cloud input. We develop GPS-Policy: we use the trained GPS model to extract GPS prediction for *the initial frame*, and encode them as additional input for RISE. *For each frame*, the policy predicts future GPS and then uses it as condition to guide action generation. The task is closing a rotation lid. Observation is recorded via a side-view RGBD camera. The policy is trained on 5 objects, with 50 demonstrations per object. We evaluate the policy on another 5 objects, conducting 10 trials each object under varying poses. We find that GPS integration boosts the success rate from 32% to 78%. Notably, GPS-Policy excels at contacting the lid at correct position and closing it via proper path. We will extend to more objects, tasks, advanced model and stronger VLA baselines in future work.

References

- [1] Sirui Chen, Chen Wang, Kaden Nguyen, Li Fei-Fei, and C Karen Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback. *arXiv preprint arXiv:2410.08464*, 2024. 1
- [2] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 653–660. IEEE, 2024. 1
- [3] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2
- [4] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke 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. 1
- [5] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. *arXiv preprint arXiv:2409.18121*, 2024. 1
- [6] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3706–3715, 2020. 2
- [7] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-level reconstruction and motion analysis for articulated objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 352–363, 2023. 1
- [8] Yu Liu, Baoxiong Jia, Ruijie Lu, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Artgs: Building interactable replicas of complex articulated objects via gaussian splatting. *arXiv preprint arXiv:2502.19459*, 2025. 1
- [9] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4): 376–380, 2002. 2
- [10] Chenxi Wang, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. Rise: 3d perception makes real-world robot imitation simple and effective. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2870–2877. IEEE, 2024. 3