2023 IEEE International Conference on Image Processing (ICIP) | 978-1-7281-9835-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIP49359.2023.10223108

REAPER: ARTICULATED OBJECT 6D POSE ESTIMATION WITH DEEP REINFORCEMENT LEARNING

Liu Liu[1,†] *Qi Wu*[2] *Zhendong Xue*[2] *Sucheng Qian*[2] *Rui Li*[3]

[1] Hefei University of Technology, Hefei 230601, China
[2] Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China
[3] Institute of Intelligent Machines, and Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China
[†] Corresponding Author

ABSTRACT

Current articulated object pose estimation methods largely rely on dense prediction for all the input observed point cloud that suffers from huge computational costs and inference time. Besides, self-occlusion is also becoming a key problem that limits the pose estimation performance for those child parts. To solve these issues, we propose a **Re**inforcement learning based **A**rticulation **P**ose **E**stimato**R** (**ReAPER**), which integrates RL into deep neural network for per-part pose estimation. Specifically, we design the novel action space that involves the object's rotation and translation, as well as a reward function considering chamfer distance during pose fitting. To speed up the RL policy training, we employ imitation learning for policy initialization. Finally, we also introduce a new kinematic energy function to optimize the child parts' poses. Experimental results show that ReAPER could obtain state-of-the-art performance on articulated object pose estimation task.

Index Terms— Reinforcement Learning, Articulation Pose Estimation, Kinematic Optimization

1. INTRODUCTION

Accurate articulated object 6D pose perception can be beneficial for many downstream tasks, such as robot manipulation [1] and VR/AR [2]. Current 6D pose estimation methods for articulated objects usually construct deep neural networks to densely predict the corresponding points in canonical space from observed point cloud [3, 4]. However, these dense prediction methods require excessive computational consumption. In addition, observed articulated objects may suffer from the self-occlusion problem that limits the performance when estimating the parts with small shapes.

Essentially, 6D pose estimation can be treated as a partial point cloud registration task [5]. Thus, pose estimators can be made to learn a trajectory that describes the procedure to fit the source point cloud (full object in canonical space) into the target point cloud (partial object observed in camera space). Under this motivation, we propose a novel **Re**inforcement learning based **A**rticulation **P**ose **E**stimato**R** (**ReAPER**) for effective articulation pose estimation that adopts RL algorithm to learn a partial point cloud registration trajectory and estimate per-part 6D pose iteratively. During network training, we adopt a deep neural network [6, 7] to extract powerful enough feature representations of full object source point cloud and partial object target point cloud. To train the RL policy, we design the novel action space for controlling rotation and translation, as well as a novel reward function that considers chamfer distance [8] as a metric. Moreover, to speed up the training, we propose imitation learning to initialize an accurate policy in a supervision manner [9].

Another problem is that we may not obtain enough points for feature extraction when the small-shape parts are occluded under a special observation view. Here, we introduce a kinematic energy function for child part pose optimization. Specifically, our method is composed of two key steps: firstly, we employ ReAPER architecture to estimate the root part's 6D pose. Next, after fitting the root part, we use the kinematic energy function for child part pose optimization to estimate the 6D poses for the child parts. In this way, we can accurately compute and refine the 6D poses of the root and child parts, also relieving the self-occlusion issue.

To sum up, the major contributions of this paper are summarized as follows: (1) We design a novel architecture ReAPER for articulation pose estimation with novel action space and reward function. (2) We utilize imitation learning as policy initialization. (3) A kinematic optimization function is implemented to accurately compute the child parts' 6D poses. Experimental results show the effectiveness of our ReAPER in the articulation pose estimation task.

2. RELATED WORK

Articulated Object Pose Estimation. Articulation estimation has been studied for many years. In the robotics com-

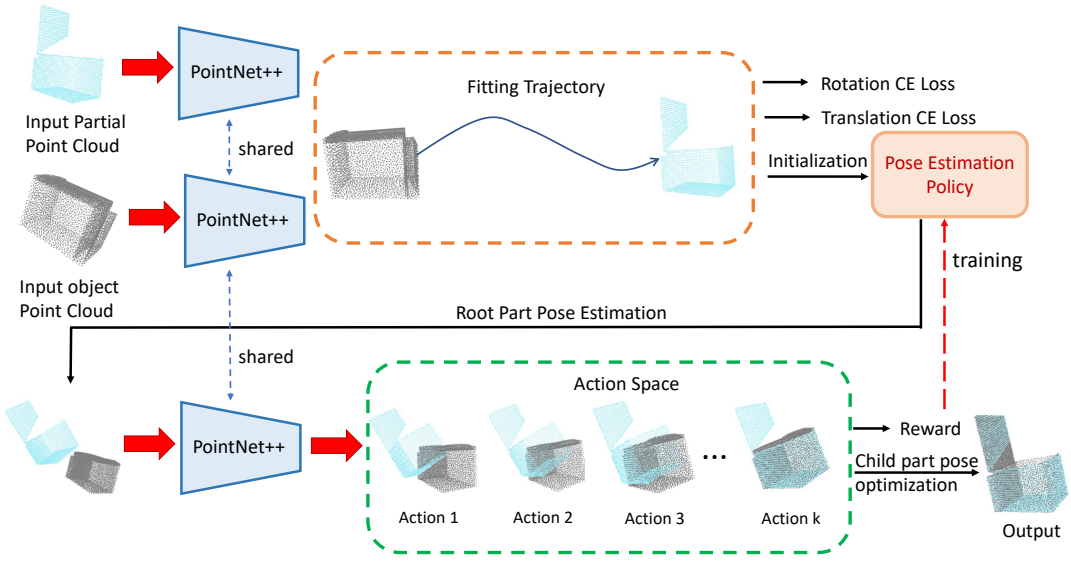

Fig. 1: Overview pipeline of ReAPER. There are three components in our method: policy initialization with imitation learning, root part pose estimation with reinforcement learning and child part pose optimization.

munity, a large number of methods adopt interaction or video observation to percept articulated objects [10, 11]. On the other hand, Wang et al. propose A-NCSH that uses dense normalized coordinates [3] to represent articulated objects and models the shape deformation implicitly with segmentation to obtain per-part pose [4]. To expand A-NCSH into the real-world articulation estimation task, Liu et al. present the extension versions ReArtNet [12] and AKBNet [13] to solve the real-world application problem. These methods adopt a dense prediction strategy that might require a large optimization complexity, and Xue et al. use ordered keypoints to obtain sparse correspondence [14]. However, the approaches above still cannot find a good trade-off between pose estimation accuracy and inference time, and also suffer from the self-occlusion problem. Thus, we propose ReAPER architecture that firstly solves articulated object pose estimation task by reinforcement learning technique.

Reinforcement Learning based Pose Estimation. There are plenty of works that use reinforcement learning for computer vision tasks, such as object detection [15] and point cloud registration [16]. In contrast, a small number of works target at solving the object pose estimation problem. Shao et al. employ deep RL to train a policy to move an object's pose and fit into RGB images [17]. Meanwhile, Busam et al. also model pose estimation as an action decision process [18]. These methods only consider rigid object pose but cannot address the articulated object pose issue, which is more challenging in the computer vision community. Referred by [16] and [5], we are the first to integrate the RL technique into solving articulated object pose estimation.

3. MATERIALS AND METHODS

ReAPER integrates RL and deep neural network for the articulation pose estimation task. In this section, we will discuss three key components in our ReAPER architecture: policy initialization, RL-based root part pose estimation, and child part pose optimization. The overview pipeline of ReAPER is illustrated in Fig. 1.

3.1. Policy Initialization with Imitation Learning

Training an RL agent for pose estimation from scratch may suffer from the unstable and inefficient training procedure as result of the slow convergence problem. To relieve this issue and speed up the training procedure, we introduce an imitation learning method for RL policy initialization. The motivation is to learn a pose-fitting trajectory and imitate the behavior of the pose estimation expert under the training labels of pose fitting in a supervision manner. In detail, as shown in Fig. 2, given the source object point cloud $\mathcal{P}$ and target partial point cloud $\mathcal{P}'$, we build a PointNet++ architecture [7] as feature extractor to process these two point clouds and output the sequential pose fitting trajectory $S = \{(R_1, \mathbf{t}_1), (R_2, \mathbf{t}_2), ..., (R_k, \mathbf{t}_k)\}$, where R_i indicates the 3D rotation and $\mathbf{t}_i$ indicates the 3D translation at the ith step. This network can be adopted as the imitation learning policy network. At the same time, since the ground truth 6D pose can be accessed in the training data, we can uniformly sample the ground truth sequential fitting trajectory $S' = \{(R'_1, \mathbf{t}'_1), (R'_2, \mathbf{t}'_2), ..., (R'_k, \mathbf{t}'_k)\}$. Finally, to train the imitation learning policy network, we use cross-entropy loss as training loss $\mathcal{L}$:

$$\mathcal{L}(S, S') = \lambda_r \mathcal{L}_r(R, R') + \lambda_t \mathcal{L}_t(\mathbf{t}, \mathbf{t}') = \lambda_r \sum_{i=1}^{k} CE(R_i, R_i') + \lambda_t \sum_{i=1}^{k} CE(\mathbf{t}_i, \mathbf{t}_i') \quad (1)$$

There are two components in this loss function where $\mathcal{L}_r$ measures the rotation error and $\mathcal{L}_t$ measures the translation error. λ_r and λ_t are scale factors and are set to be 0.3 and 0.7 since translation fitting policy is more difficult to converge.

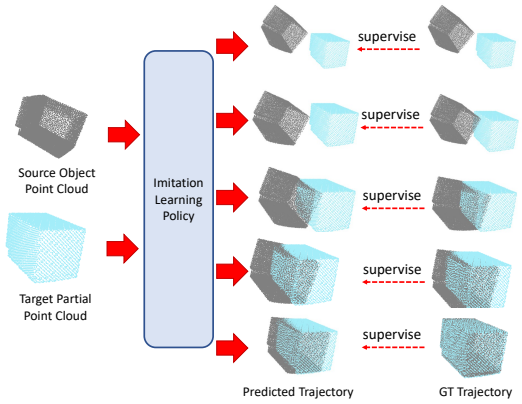

Fig. 2: The procedure of policy initialization via imitation learning technique.

3.2. Root Part Pose Estimation with RL Agent

After imitation learning initialization, we can train the RL agent for pose estimation. Due to the self-occlusion problem, we use the RL mechanism to solve the root part pose estimation task and then adopt child part pose optimization to obtain the whole 6D pose for the input articulated object. Formally, deep RL agent training consists of three components: state $\mathcal{S}$, action space $\mathcal{A}$, and reward function $\mathcal{R}$. In our ReAPER, the state $\mathcal{S}$ is represented by the feature vector outputted from the final layer of the PointNet++ architecture [7]. Given the state $\mathcal{S}$, we design 12 actions to build action space $\mathcal{A}$, in which 6 actions indicate rotation directions (X-axis positive, X-axis negative, Y-axis positive, Y-axis negative, Z-axis positive, Z-axis negative) and the other 6 actions indicate translation directions. The action space in our method is shown in Fig. 3.

A reasonable reward function takes the most essential part in our RL-based pose estimator. In our work, we design the reward function to consider the chamfer distance between the current transformed source point cloud $\mathcal{P}_i$ at ith step and target point cloud $\mathcal{P}'$. Based on this motivation, the step-wise reward function is defined as:

$$R(S_i, S') = \begin{cases} \eta & CD(\mathcal{S}_i, \mathcal{S}') \leq CD(\mathcal{S}_{i-1}, \mathcal{S}') \\ -\eta & CD(\mathcal{S}_i, \mathcal{S}') \geq CD(\mathcal{S}_{i-1}, \mathcal{S}') \end{cases} \quad (2)$$

where η is set to 0.5. $\mathcal{S}_i$, $\mathcal{S}_{i-1}$ and $\mathcal{S}'$ indicate the states of source point clouds at ith step, $i-1$th step and target point cloud. Based on this reward function, we define that steps reducing chamfer distance are rewarded by η and "stop" action or steps increasing chamfer distance are penalized by $-\eta$. In this way, we can train a RL agent to achieve root part pose estimation.

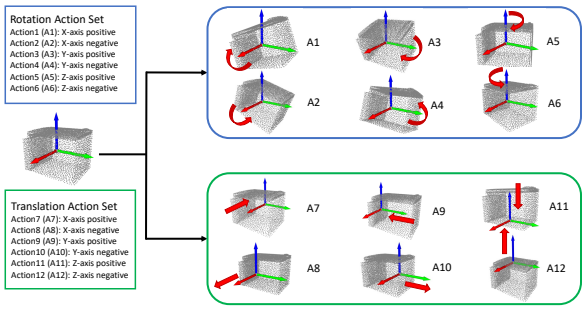

Fig. 3: The proposed action space for our ReAPER.

3.3. Child Part Pose Optimization

Within the RL agent we can obtain the root part pose estimation result. However, when employing the RL method in child part pose estimation, we might suffer from the inefficient training problem. This is because of the self-occlusion that cause weak state representation of the child part point cloud. To solve this issue, we introduce an optimization method to refine the child part pose. In detail, we initialize the kth child part pose with the the predicted root part pose $\{R_0, \mathbf{t}_0\}$. Next, given the joint information $\{\mathbf{u}_k, \mathbf{q}_k\}$ that is already defined in the source object point cloud, we build an energy function based on the kinematic structure of the input articulated object and optimize the kth child part pose:

$$E_k = \frac{1}{|N|} \sum_{x \in T(\mathcal{P}_k, \mathbf{u}_k, \mathbf{q}_k, \theta_k)} \min_{y \in \mathcal{P}_k'} \|x - y\|_2 + \frac{1}{|N'|} \sum_{y \in \mathcal{P}_k'} \min_{x \in T(\mathcal{P}_k, \mathbf{u}_k, \mathbf{q}_k, \theta_k)} \|y - x\|_2 \quad (3)$$

where $T(\mathcal{P}_k, \mathbf{u}_k, \mathbf{q}_k, \theta_k)$ indicates a transformation function that transforms source child part point cloud $\mathcal{P}_k$ around the joint $\{\mathbf{u}_k, \mathbf{q}_k\}$ by the state θ_k. The transformation function is defined based on kinematic chain of articulation [4]. To obtain the child part pose, we can optimize the energy function to get the optimal joint state θ_k^*. Finally, we can get the child part pose $\{R_k, \mathbf{t}_k\}$ by transforming the optimized joint state θ_k^* into the 6D pose format.

4. EXPERIMENTS

4.1. Experimental Settings

We use PointNet++ architecture [7] as the feature extractor, in which the dimension of the final layer is set to be 2048. All

Category	Method	Per-part 6D Pose		Joint State	Inference Time ↓
		rotation error ↓	translation error ↓	error ↓	
Eyeglasses	A-NCSH [4]	3.7°, 5.1°, 3.7°	0.035, 0.051, 0.057	4.1°, 4.5°	11.9s
	ReArtNet [12]	3.9°, 5.6°, 3.8°	0.031, 0.048, 0.055	4.2°, 4.1°	13.8s
	OMAD [14]	5.5°, 4.5°, 4.5°	0.062, 0.033, 0.034	3.9°, 4.2°	2.5s
	ReAPER (Ours)	**3.3**°, 4.6°, 4.8°	**0.028**, **0.031**, **0.033**	**3.6**°, 4.2°	**1.5s**
Oven	A-NCSH [4]	1.1°, 2.2°	0.033, 0.043	2.1°	6.6s
	ReArtNet [12]	1.5°, 2.5°	0.035, 0.052	3.0°	8.2s
	OMAD [14]	3.6°, 3.9°	0.056, 0.055	3.1°	1.6s
	ReAPER (Ours)	1.6°, **2.2**°	**0.046**, **0.041**	**2.1**°	**0.9s**
Washing machine	A-NCSH [4]	1.0°, 1.4°	0.045, 0.037	1.5°	6.8s
	ReArtNet [12]	1.8°, 2.3°	0.055, 0.051	2.8°	9.0s
	OMAD [14]	2.8°, 3.6°	0.068, 0.064	3.2°	1.8s
	ReAPER (Ours)	**0.9**°, **1.1**°	**0.032**, **0.034**	**1.4**°	**0.9s**
Laptop	A-NCSH [4]	6.7°, 4.3°	0.062, 0.044	9.7°	9.0s
	ReArtNet [12]	6.3°, 5.5°	0.049, 0.051	9.4°	10.5s
	OMAD [14]	9.6°, 9.8°	0.061, 0.064	11.2°	1.6s
	ReAPER	**4.5**°, 6.0°	**0.031**, **0.042**	**9.9**°	**0.8s**
Drawer	A-NCSH [4]	1.0°, 1.1°, 1.2°, 1.5°	0.024, 0.021, 0.021, 0.033	0.011, 0.020, 0.030	16.5s
	ReArtNet [12]	1.3°, 1.0°, 1.2°, 1.3°	0.019, 0.023, 0.022, 0.027	0.010, 0.014, 0.015	17.5s
	OMAD [14]	4.0°, 4.0°, 4.0°, 4.0°	0.020, 0.027, 0.029, 0.026	0.009, 0.015, 0.016	1.9s
	ReAPER (Ours)	1.1°, 1.5°, 1.4°, **1.3**°	**0.016**, **0.023**, **0.023**, **0.025**	0.010, **0.011**, **0.015**	2.2s

Table 1: Comparison with state-of-the-arts on PartNet-Mobility dataset. ↓ means the lower the better.

Category	Method	Per-part 6D Pose	
		rotation error ↓	translation error ↓
Laptop	OMAD [14]	5.4°, 4.3°	0.062, 0.061
	ReAPER (Ours)	**2.8**°, **3.1**°	**0.042**, **0.039**
Eyeglasses	OMAD [14]	4.9°, 7.5°, 7.5°	0.062, 0.103, 0.104
	ReAPER (Ours)	**2.9**°, **3.6**°, **3.5**°	**0.046**, **0.061**, **0.060**
Dishwasher	OMAD [14]	6.0°, 6.2°	0.104, 0.142
	ReAPER (Ours)	**2.8**°, **2.9**°	**0.054**, **0.067**
Scissors	OMAD [14]	3.9°, 3.4°	0.048, 0.039
	ReAPER (Ours)	**2.5**°, **3.1**°	**0.034**, **0.036**
Drawer	OMAD [14]	4.4°, 4.4°, 4.4°, 4.4°	0.111, 0.144, 0.143, 0.115
	ReAPER	**2.8**°, **3.3**°, **3.6**°, **3.4**°	**0.084**, **0.103**, **0.101**, **0.105**

Table 2: Comparison with state-of-the-arts on ArtImage dataset. ↓ means the lower the better.

the experiments are conducted on two popular synthetic articulated object datasets, PartNet-Mobility [19] and ArtImage [14]. We train our ReAPER agent using Adam and a batch size of 32. Experimental results are reported with rotation, translation and joint state error.

4.2. Experiments on PartNet-Mobility

Table 1 shows the performance of different methods on the articulated object pose estimation task. We can see that our ReAPER achieves state-of-the-art performance on the PartNet-Mobility dataset, which proves the effectiveness of our reinforcement learning based estimator. On some categories such as Oven and Washingmachine, our method can obtain **1.6**°, **2.2**°, **0.9**° and **1.1**° on rotation estimation with only **0.9s** testing. Compared to one of the popular methods OMAD, our ReAPER surpasses it by a large margin on not only pose estimation error but also inference time. This can be explained by that reinforcement learning largely save computational consumption compared with the optimization based methods. It is worth noting that ReAPER can also obtain a comparable performance with A-NCSH, which is a dense prediction method.

4.3. Experiments on ArtImage

We also report the results of our ReAPER on the ArtImage dataset in Table 2. As can be observed, our method largely outperforms OMAD in all the categories. For example, we achieve **2.8**° and **3.1**° rotation error on Laptop, which are **2.6**° and **1.2**° higher than the baseline method OMAD. In addition, on child part poses estimation, ReAPER can achieve **3.6**° and **3.5**° on two child parts of Eyeglasses, which is dymatically better than OMAD. This indicates that ReAPER could address the self-occlusion problem in the articulation pose estimation task.

5. CONCLUSION

In this work, we propose a reinforcement learning based articulated object pose estimation method ReAPER, which integrates the advantage of reinforcement learning into point cloud processing to achieve promising pose estimation performance. In our method, we propose a novel RL architecture to train a partial and object point cloud fitting agent for articulation pose estimation with our designed action space and reward function. To speed up the training procedure, we adopt imitation learning as policy initialization. In addition, to further relieve the influence of the self-occlusion problem, we introduce an optimization energy function to refine the child part pose. Experiments demonstrate that our ReAPER achieves promising results on the articulated object pose estimation task.

6. REFERENCES

[1] Karthik Desingh, Shiyang Lu, Anthony Opipari, and Odest Chadwicke Jenkins, "Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects," *Science Robotics*, vol. 4, no. 30, pp. eaaw4523, 2019.

[2] Catherine Taylor, Robin McNicholas, and Darren Cosker, "Transporting real world rigid and articulated objects into egocentric vr experiences," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020, pp. 622–623.

[3] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas 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.

[4] 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*, 2020, pp. 3706–3715.

[5] Dominik Bauer, Timothy Patten, and Markus Vincze, "Sporeagent: Reinforced scene-level plausibility for object pose refinement," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 654–662.

[6] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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.

[7] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[8] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin, "Balanced chamfer distance as a comprehensive metric for point cloud completion," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29088–29100, 2021.

[9] Cheng Ding, Wei Du, Jianhua Wu, and Zhenhua Xiong, "Template-based imitation learning for manipulating symmetric objects," *Mechatronics*, vol. 78, pp. 102609, 2021.

[10] Karol Hausman, Scott Niekum, Sarah Osentoski, and Gaurav S Sukhatme, "Active articulation model estimation through interactive perception," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3305–3312.

[11] Qihao Liu, Weichao Qiu, Weiyao Wang, Gregory D Hager, and Alan L Yuille, "Nothing but geometric constraints: A model-free method for articulated object pose estimation," *arXiv preprint arXiv:2012.00088*, 2020.

[12] Liu Liu, Han Xue, Wenqiang Xu, Haoyuan Fu, and Cewu Lu, "Toward real-world category-level articulation pose estimation," *IEEE Transactions on Image Processing*, vol. 31, pp. 1072–1083, 2022.

[13] Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Qiaojun Yu, Yang Han, and Cewu Lu, "Akb-48: A real-world articulated object knowledge base," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14809–14818.

[14] Han Xue, Liu Liu, Wenqiang Xu, Haoyuan Fu, and Cewu Lu, "Omad: Object model with articulated deformations for pose estimation and retrieval," 2021.

[15] Man Zhou, Liu Liu, and Rujing Wang, "Reinforcedet: Object detection by integrating reinforcement learning with decoupled pipeline," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 2778–2782.

[16] Dominik Bauer, Timothy Patten, and Markus Vincze, "Reagent: Point cloud registration using imitation and reinforcement learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14586–14594.

[17] Jianzhun Shao, Yuhang Jiang, Gu Wang, Zhigang Li, and Xiangyang Ji, "Pfrl: Pose-free reinforcement learning for 6d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11454–11463.

[18] Benjamin Busam, Hyun Jun Jung, and Nassir Navab, "I like to move it: 6d pose estimation as an action decision process," *arXiv preprint arXiv:2009.12678*, 2020.

[19] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al., "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11097–11107.