Category-Level Articulated Object 9D Pose Estimation via Reinforcement Learning

Liu Liu
Hefei University of Technology
Hefei, China
liuliu@hfut.edu.cn

Jianming Du
Hefei Institutes of Physical Science, Chinese Academy of Sciences
Hefei, China
djming@iim.ac.cn

Hao Wu
Institute of Dataspace
Hefei, China
wuhao@idata.ah.cn

Xun Yang
University of Science and Technology of China
Hefei, China
xyang21@ustc.edu.cn

Zhenguang Liu
Zhejiang University
Hangzhou, China
liuzhenguang2008@gmail.com

Richang Hong
Hefei University of Technology
Hefei, China
hongrc@hfut.edu.cn

Meng Wang
Hefei University of Technology
Hefei, China
eric.mengwang@gmail.com

ABSTRACT

Human life is populated with articulated objects. Current category-level articulated object 9D pose estimation (**Art**iculated **O**bject 9D **P**ose **E**stimation, **ArtOPE**) methods usually meet the challenges of shared object representation requirement, kinematics-agnostic pose modeling and self-occlusions. In this paper, we propose a novel framework called **Art**iculated object 9D **P**ose **E**stimation via **R**einforcement **L**earning (**ArtPERL**), which formulates the category-level ArtOPE as a reinforcement learning problem. Given a point cloud or RGB-D image input, ArtPERL firstly retrieves the part-sensitive articulated object as reference point cloud, and then introduces a joint-centric pose modeling strategy that estimates 9D pose by fitting joint states via reinforced agent training. Finally, we further propose a pose optimization that refine the predicted 9D pose considering kinematic constraints. We evaluate our ArtPERL on various datasets ranging from synthetic point cloud to real-world multi-hinged object. Experiments demonstrate the superior performance and robustness of our ArtPERL. Our work provides a new perspective on category-level articulated object 9D pose estimation and has the potential to be applied in many fields, including robotics, augmented reality, and autonomous driving.

CCS CONCEPTS

• **Computing methodologies** → **Vision for robotics**.

KEYWORDS

Articulated Object, 9D Pose Estimation, Reinforcement Learning, Joint-Centric Pose Modeling

ACM Reference Format:
Liu Liu, Jianming Du, Hao Wu, Xun Yang, Zhenguang Liu, Richang Hong, and Meng Wang. 2023. Category-Level Articulated Object 9D Pose Estimation via Reinforcement Learning. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3611852

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3611852

1 INTRODUCTION

Our daily lives are filled with articulated objects, ranging from small objects like eyeglasses to larger objects like dishwashers. Unlike rigid objects that can be treated as a whole in three-dimensional space, articulated objects consist of a limited number of movable rigid parts linked by various joints and constrained by a kinematic structure. As a result, accurate estimation of the nine-dimensional (9D) pose for articulated objects is crucial in many downstream multimedia and machine vision tasks, such as hand-object interactions [12], hand reconstruction [34], robot manipulation [32], video understanding [8] or object tracking [35]. Generally, compared to instance-level six-dimensional (6D) pose estimation [24, 26, 28], category-level 9D pose estimation requires machines to understand the 3D rotation, 3D translation, and 3D scale of unseen objects, using only semantic category prior.

Under these circumstances, an increasing number of researchers are focusing on category-level **Art**iculated **O**bject 9D **P**ose **E**stimation (**ArtOPE**). [14, 17, 36]. Normalized object coordinate space (NOCS) and its variants [18, 19, 27, 29] have introduced intra-class representations, which have become the mainstream paradigm for investigating category-level articulated objects. However, this solution faces several challenges:

- It strongly requires a shared object representation for learning and optimizing per-part 9D pose, which demands a sufficiently powerful deep learning network.
- Most of these methods employ part-centric pose modeling, which predicts per-part pose separately, thereby ignoring the constraints of the kinematic structure.
- These methods are not robust to self-occlusion cases, where small-size movable parts are occluded by large-size parts under certain camera views.

Essentially, object pose estimation can be modeled as a point cloud fitting problem, where a fitting trajectory is learned to describe the process of registering the source object (point cloud in canonical space) into the target observed object (partial point cloud in camera space). Motivated by this assumption and in order to address the aforementioned challenges, we propose a novel approach namely **Art**iculated object 9D **P**ose **E**stimation via **R**einforcement **L**earning (**ArtPERL**), which formulates the problem of 9D articulation pose estimation as an iterative pose fitting task with discrete steps. To the best of our knowledge, we are the first to consider ArtOPE task as a reinforcement learning problem. To address the object representation problem, we employ a light but effective part-sensitive articulated object retrieval network that builds per-part embedding learning with the PointNet++ architecture [21]. By using precise articulated object retrieval, we can initialize the source object as a highly similar object to the observed object, thereby alleviating the strong requirement of intra-class representation learning and achieving a shared object representation-free 9D pose estimation.

Regarding the pose modeling problem, we propose a joint-centric 9D pose estimation mechanism. Specifically, our joint-centric pose definition is similar to the OMAD modeling method [33] that builds a corresponding part-joint pair, where each part's pose can be learned as a joint state representation. We focus on the three most common joint types: free joints that allow the part to move into any rotational and translational target (e.g., cabinet main body), revolute joints that cause only rotational motion (e.g., laptop hinges), and prismatic joints that allow only translational movement (e.g., drawers in a cabinet). Since the joint types and parameters reflect the kinematic structure of an articulated object, our joint-centric pose modeling and per-part pose estimation results can effectively eliminate abnormal pose predictions (anti-joint motion) and correctly follow the kinematic rule.

In terms of the self-occlusion problem, our proposed solution is to employ a kinematic tree guided pose estimation algorithm. We begin by designing an additional branch in the PointNet++ architecture to segment each rigid part. Next, we follow the order from root to leaf to estimate the 9D pose for the root part and then estimate the leaf part's pose under the predicted root pose and corresponding joint parameter, thus constraining the leaf part's pose estimation procedure with the root part estimation result and alleviating the impact of self-occlusion. Additionally, we propose an articulated 9D pose optimization method that considers both part pose and joint parameters as constraints, formulating articulated pose fitting as a combined optimization problem with given kinematic constraints.

We evaluate our ArtPERL on both point cloud observations and RGB-D images for ArtOPE task, whose datasets are ArtImage [33] and ReArtMix [18] ranging from synthetic dataset to real-world dataset. To further prove the strength of our proposed joint-centric pose modeling and articulation pose optimization algorithm, we also test ArtPERL in a multi-hinged articulated object Robot Arm benchmark that is built by [18]. We believe the extensive experiments show the superior performance and robustness of ArtPERL compared with state-of-the-arts. To summarize, our key contributions are:

- ArtPERL is a novel framework proposed to solve the problem of category-level articulated object 9D pose estimation, where the pose estimation problem is fomulated as a reinforcement learning procedure for the first time.
- To address challenges such as object representation, pose modeling, and self-occlusion, ArtPERL introduces modules such as the part-oriented articulated object retrieval network, joint-centric pose modeling, and kinematic tree guided pose estimation mechanism.
- The efficiency and robustness of ArtPERL are demonstrated through evaluation on both point cloud observations and RGB-D images for ArtOPE task, using various datasets ranging from synthetic to real-world scenarios.

2 RELATED WORK

Instance-level 6D pose estimation. Instance-level pose estimation refers to the alignment of an available 3D CAD model with an object observation (RGB or depth image). Due to the various object alignment approaches, most rigid 6D pose estimation methods can be classified into two paradigms. The first is template-based methods [3, 5], which use a rigid template to match the observed object. One classic method is the point pair feature proposed by Drost et al. [10], which uses an object voting scheme to match the observed object with a database. Vidal et al. [25] improved this method by considering neighborhoods that could potentially be affected by noise. On the other hand, feature-based methods aim to regress the object surface position for every pixel in the visual observation and use RGB information as texture features to establish stable object alignment and pose estimation [9, 20]. Wang et al. [26] proposed a new framework called DenseFusion, in which a heterogeneous architecture is used to process the RGB image and depth individually, followed by a dense fusion network that extracts pixel-wise dense feature embeddings.

Category-level 9D Pose Estimation. Recently, there has been a shift towards category-level 9D pose estimation task, where the aim is to detect an unseen object's 3D rotation, 3D translation, and 3D scale. The first proposed method, NOCS, predicts 3D-3D per-pixel correspondences between observations and canonical coordinates using a normalized space for category-level representation [27]. CASS uses a variational auto-encoder to capture both pose-independent and pose-dependent features to directly predict the 6D poses [6]. To overcome the sim-to-real gap when transferring into real-world applications, You et al. [37] design category-level PPF for robust and generalized 9D pose estimation. While most of these methods target rigid objects, some researchers are now paying attention to articulated object 9D pose. A-NCSH is an extension of NOCS, developed for single articulated object pose estimation,

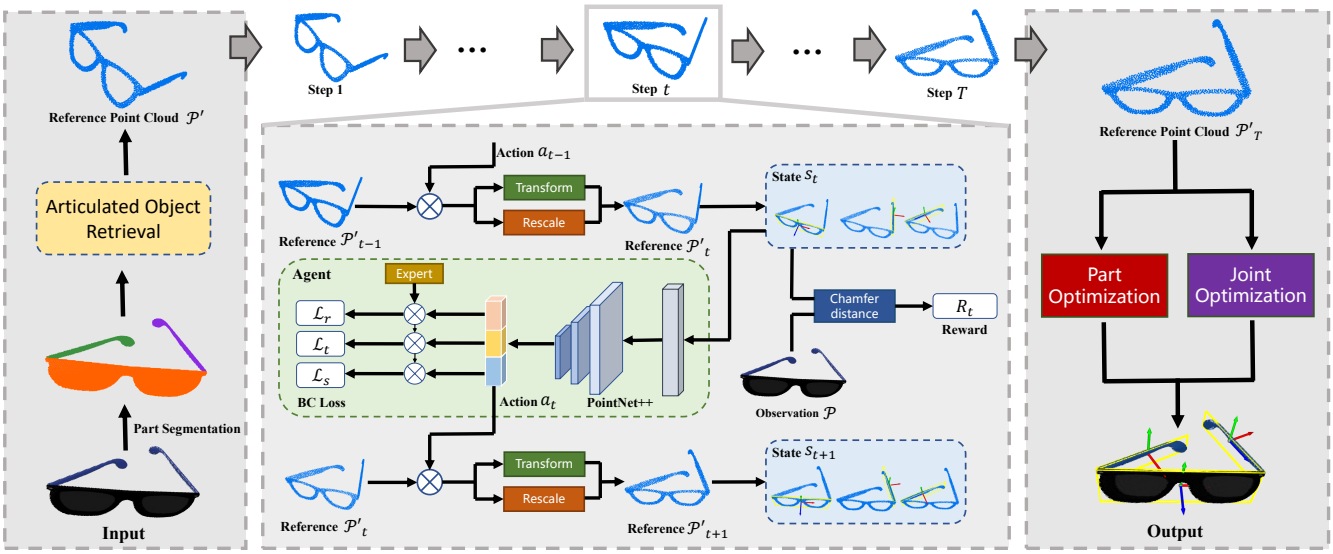

Figure 1: The ArtPERL framework. The observed object input is firstly processed by a part segmentation module for accurate part-sensitive articulated object retrieval. At each step t of reinforced agent training, we use the transformed and rescaled reference point cloud $\mathcal{P}'_t$ as input to get the predicted action by a PointNet++ architecture. We also propose a pose optimization method to further refine the predicted 9D poses.

which proposes per-part normalized coordinates for pose optimization [14]. To improve generalization, Liu et al. [18] select a pair part pose recovering strategy to alleviate the dependency of kinematic structure prior. Additionally, Xue et al. [33] propose using keypoints as an articulation modeling to speed up the inference time for accurate pose estimation. However, despite these achievements, these methods still have some unsolved issues such as object representation required and self-occlusion, and therefore, are inferior to our method.

Reinforcement Learning based Pose Estimation. Reinforcement learning is a widely used technique in many computer vision tasks, such as object detection [38] and point cloud registration [1]. In recent years, a few works attempt to employ reinforcement learning into pose estimation task. Shao et al. employ deep RL to train a policy to move an object's pose and fit into RGB images [23]. To improve the robustness, Bauer et al. [2] consider multiple object targets for pose fitting and refinement. Meanwhile, Busam et al. also model pose estimation as an action decision process [4]. An alternative use of RL is proposed in [13], where RL actions are selected as one of a pool of pose hypotheses to be refined in each iteration. To order to relief the human labor of pose annotations, Gong et al. [11] propose a meta agent learning framework for rigid object pose estimation. However, when meeting category-level articulated object 9D pose estimation problem, these methods are limited to: (1) they only consider rigid object in which the action space is much smaller than articulated object since more than one rigid parts are constrained by kinematics. (2) most of them model pose estimation as a point cloud registration problem that requires one-to-one object models for fitting, resulting in poor performance on category-level pose estimation. Therefore, we are the first to consider reinforcement learning paradigm into category-level articulated object 9D pose estimation task.

3 PROBLEM FORMULATION

To achieve robust category-level ArtOPE, our core idea is modeling pose estimation as a fitting procedure. Here, We formulate a new paradigm for the category-level ArtOPE task via the reinforcement learning technique and a new learning framework named ArtPERL. Given a 3D observed object point cloud $\mathcal{P} = \{(x_i, y_i, z_i)\}_{i=1}^{M}$ with K articulated parts, we perform predictions under unknown CAD models with a *Reinforced Agent* for (1) per-pixel semantic segmentation $\delta = \{\delta_i\}_{i=1}^{M}$, (2) per-part 3D rotation $R^{(k)}$, (3) per-part 3D translation $\mathbf{t}^{(k)}$, (4) per-part 3D amodal bounding box with scale $\mathbf{s}^{(k)}$. In total, the rotation, translation and scale together constitute a 9D pose estimation result $\{R^{(k)}, \mathbf{t}^{(k)}, \mathbf{s}^{(k)}\}_{k=1}^{K}$.

In our framework, due to the unavailable one-to-one CAD model, we use a part-sensitive articulated object retrieval to find an object of the same category and kinematic structure with high appearance similarity, which is employed as the reference object $\mathcal{P}'$ for pose estimation. Since one articulated object might contain several movable parts, the network is designed for part sensitivity that considers each part embedding for retrieval. Next, to enable the reinforced agent to accurately fit the object between that canonical space and camera space, we hypothesize the decision made by the agent at each time step is only determined by the current state and not be affected by previous states, thus the articulation 9D pose estimation problem can be formulated as a Markov Decision Process. The goal is to maximize the expected sum of future discounted rewards $V^{\pi}(s) = \mathbb{E}[\sum_{t\geq 0} \gamma_t r_t]$, where $\gamma \in [0, 1)$ is the discount factor, $r_t = r(s_t, a_t)$ is an immediate reward at time t and $a_t \sim \pi(\cdot|s_t)$ is the action generated by the policy π conditioned on s_t. In our work, when the agent generates the correct point cloud fitting trajectory from reference point cloud $\mathcal{P}'$ and observed point cloud $\mathcal{P}$, the 9D pose can be recovered from this fitting trajectory.

4 THE ARTPERL ARCHITECTURE

We propose a reinforcement learning approach ArtPERL to tackle ArtOPE task. Taking an observed partial point cloud $\mathcal{P}$ as input, our ArtPERL consists of the following components: (1) part-sensitive articulated object retrieval to find the high-similar object as a reference point cloud for fitting the pose (Sec. 4.2). (2) reinforcement learning architecture to achieve per-part 9D pose estimation with joint-centric articulation pose modeling method (Sec. 4.3). (3) an articulation 9D pose optimization with kinematic constraints to further refine the per-part 9D pose and avoid the self-occlusion problem (Sec. 4.4). The overall pipeline is illustrated in Fig. 1.

4.1 Joint-Centric 9D Pose Modeling

As discussed above, most of the articulation pose estimation methods adopt part-centric pose modeling strategy that might ignore the effect of kinematic structure and also suffer from the self-occlusion problem. In this paper, we take a joint-centric perspective to investigate articulated objects, which regards the pose estimation task as a joint state prediction task. In this way, the motion of each movable part is corresponding to the joint state changing.

In an articulated object, the kinematic tree is one kind of abstraction for the object's motion, where each joint connects two nodes: parent node and child node. Thus, when the child part moves, the motion can be modeled as that it rotates around the joint with angle θ or translates along the joint with distance d. This pose modeling strategy has two advantages: (1) we can estimate 6D pose $\{R^{(k)}, \mathbf{t}^{(k)}\}$ by only predicting joint state $\theta^{(k)}$, which reduces the degree of freedom of articulated objects. (2) we can estimate joint state by a chained estimation method that only needs to predict delta $\theta^{(k)}$ rather than absolute 6D pose.

Following the setting used in A-NCSH [14] and OMAD [33], we consider three types of joints in our work: (1) Free joint that connects the world with the root node in an articulated object, whose parent node is the world and child node is the base part. It allows the base part to move freely in the 3D space with arbitrary rotation and translation. Thus, the joint state of the free joint is defined as the 6D pose $\theta^{(free)} = (R^{(free)}, \mathbf{t}^{(free)})$. Note that there is only one free joint for an articulated object. (2) Revolute joint that connects the base part with the child part that allows the rotational motion in the articulated object. The joint state $\theta^{(rev)}$ is denoted by the relative rotation angle compared to the rest state. (3) Prismatic joint that allows the child part to move along the joint direction. The joint state $\theta^{(prs)}$ is denoted by the relative distance compared to the rest state. By defining the joint-centric pose modeling method, we can correspond part to joint one by one, and the total articulated object poses for all the K parts can be represented by a sequence of joint states $\Theta = \{\theta^{(k)}\}_{k=1}^{K}$.

4.2 Part-Sensitive Articulated Object Retrieval

Accurate articulated object retrieval is a high priority for our method. To the best of our knowledge, there is no available retrieval method specially designed for our task. We design a simple but effective Siamese Network inspired by current shape retrieval methods [15, 31]. Different from 3D shape retrieval, articulated object retrieval requires the network to be sensitive to the geometry of each articulated part. Thus, we apply the part feature learned from point clouds into Siamese Network for articulation shape retrieval. In detail, the input is the observed point cloud $\mathcal{P} = \{(x_i, y_i, z_i)\}_{i=1}^{M}$ as well as the training shape base. We randomly sample 1024 points from this point cloud and construct PointNet++ encoder-decoder architecture as the backbone for per-point feature v_i extraction. At the end of PointNet++ feature propagation layers, we build a three-layer MLP with ReLU activation function that outputs K channels for part segmentation, where K is the maximum number of rigid parts for this input articulated object. In this way, we can exploit the part segmentation results $\delta = \{\delta_i\}_{i=1}^{M}$ to achieve part-sensitive feature learning. Given the predicted segments δ, the final object feature vector of the articulated object can be computed as the sum of each part's feature vector:

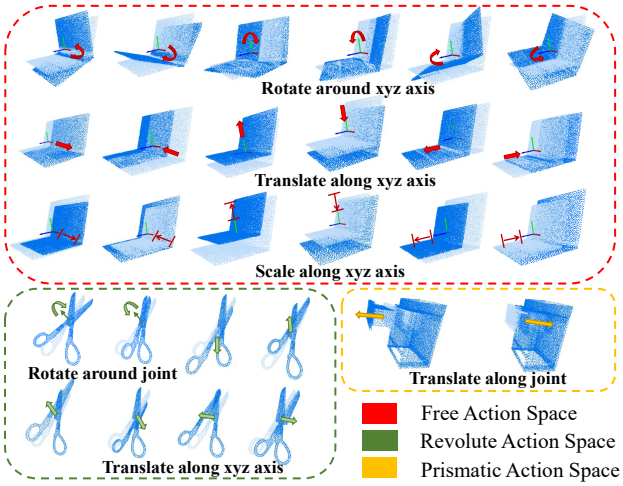

Figure 2: Action space for three joint types. Note that each action is further discretized into several intervals.

$$f = \frac{1}{K}\sum_{k=1}^{K}[\sum_{i=1}^{M} v_i \mathbb{1}(\delta_i = k)] \tag{1}$$

where $\mathbb{1}$ indicates the sign function. During the articulated object retrieval network training, our goal is to minimize the contrastive loss $\mathcal{L}_{rtv}$ defined by [7] that makes the predicted shape vector remain consistent in the same instance and increase distances between different instances. In detail, we randomly construct positive sample pairs and negative sample pairs, in which the positive sample pairs (v, v_{pos}) consist of the observed point cloud and the object in canonical space with high similarity, while the negative sample pairs (v, v_{neg}) consists of that and the random object in canonical space. Therefore, the contrastive loss $\mathcal{L}_{rtv}$ for articulated object retrieval is:

$$\mathcal{L}_{rtv} = (1-y)\frac{1}{2}[D(v, v_{pos})]^2 + y\frac{1}{2}[max(0, \epsilon - D(v, v_{neg}))]^2 \tag{2}$$

where $D(\cdot)$ indicates the Euclidean distance function, y is the label where $y = 1$ for negative sample pairs and $y = 0$ for positive sample pairs. ϵ is a hyper-parameter and we set $\epsilon = 1$. For retrieving the articulated object, we use $D(\cdot)$ to measure the object feature vector similarity.

4.3 Reinforcement Learning for ArtOPE

We train a reinforcement learning agent for pose estimation. Under our proposed joint-centric articulation pose modeling method, we use the reinforced agent to solve per-part 9D pose by the order of the kinematic tree. Formally, the pose fitting procedure starts with a feature embedding that transfers the observed articulated object point cloud into a global feature vector. Then we concatenate the observed point cloud's and reference point cloud's feature vector as the state representation s_t at action step t, which encodes all the object information of the current state. Next, the trained policy adopts the state representation to predict action vectors for the reference point cloud's transformation. Note that the number and type of action vectors depend on the joint type that is corresponding to the movable part. Finally, the resulting transformation is applied to the reference point cloud and iteratively improves the pose fitting. In each action step, we design a reward function to judge how well it performs. The individual components are discussed in more detail.

State representation. The state which encodes the knowledge of the environment should be instrumental for the agent to decide how to rotate, translate and scale the reference point cloud. Here, we build PointNet++ [21] encoder for feature embedding extraction, mapping from $M \times 3$ dimensional point cloud to $1 \times C$ dimensional state space. The concatenation of the state of the observed point cloud and that of the reference point cloud is used as state s.

Discrete action space. Previous works prove that a bad estimate in one step might lead to divergence of the whole fitting process [1], so a discrete action space could robustify the matching process well. In our ArtPERL, the action space depends on the joint types since different joint type might cause different part motions. In detail, (1) for free joint, there are three action vectors for rotation, translation and scaling, respectively, where rotation action controls the reference point cloud to rotate around x, y, z axis of the origin axis, translation action controls it to translate along x, y, z axis of the center and scaling action controls the shape of the reference point cloud be scaled along x, y, z directions. (2) for the revolute joint, there are two action vectors. One is designed for controlling the object to rotate around the given joint and the other is designed for controlling its translation. (3) for the prismatic joint, there is only one action vector that controls the object to translate along the joint direction. All the actions for three joint types are shown in Fig. 2. Note that we use a set of discrete steps for each type.

Reward function. The overall goal of ArtPERL is to align the reference point cloud with the observed point cloud. Here, we follow the reward definition widely used in point cloud registration algorithms [2], Chamfer Distance, to measure the pose fitting performance at each step. This measure is insensitive to transformations that result in the same distance between the closest points. Given the observed point cloud $\mathcal{P}$ and the accumulated transformed reference point cloud $\mathcal{P}'$, the step-wise reward is defined as:

$$R(\mathcal{P}'_t, \mathcal{P}) = \begin{cases} \eta & CD(\mathcal{P}'_t, \mathcal{P}) < CD(\mathcal{P}'_{t-1}, \mathcal{P}) \\ -0.1\eta & CD(\mathcal{P}'_t, \mathcal{P}) = CD(\mathcal{P}'_{t-1}, \mathcal{P}) \\ -\eta & CD(\mathcal{P}'_t, \mathcal{P}) > CD(\mathcal{P}'_{t-1}, \mathcal{P}) \end{cases} \tag{3}$$

where η is 0.5. $\mathcal{P}'_t$ and $\mathcal{P}'_{t-1}$ indicate the transformed reference point cloud at t and $t-1$ step. As it can be seen, we define that steps reducing chamfer distance are rewarded by η and "stop" action or steps increasing chamfer distance are penalized by -0.1η and -η. In this way, we can train a reinforced agent to achieve per-part pose estimation.

Policy training. Category-level articulated object 9D pose estimation is a complicated task, and training a reinforced agent from scratch may take long to converge and get stuck with a suboptimal policy. From the early experiments, we observe that the reference point cloud is sensitive to scaling actions which usually cause shape deformation on the object. To relieve this issue and avoid divergence during agent training, we composite Behavior Clone (BC) technique into reinforced optimization referred by [1]. In this technique, we allow the reinforced agent to have access to the action labels from the expert in every step, and we can train a robust trajectory by directly supervising the agent actions from initialized state to the ground truth 9D pose. Specifically, we randomize the agent actions at the whole trajectory and gather a replay buffer by collecting all the random trajectories and the corresponding expert actions. The gathered data and ground truth actions allow us to train and initialize the agent using the cross-entropy loss function for supervised learning. Take the free joint as an example, given the predicted pose fitting trajectory $\{(R_1, \mathbf{t}_1, \mathbf{s}_1), (R_2, \mathbf{t}_2, \mathbf{s}_2), ..., (R_t, \mathbf{t}_t, \mathbf{s}_t)\}_{t=1}^{T}$ and action labels from expert $\{(R'_1, \mathbf{t}'_1, \mathbf{s}'_1), (R'_2, \mathbf{t}'_2, \mathbf{s}'_2), ..., (R'_t, \mathbf{t}'_t, \mathbf{s}'_t)\}_{t=1}^{T}$ the training loss of behavior clone $\mathcal{L}_{bc}$ is:

$$\begin{aligned} \mathcal{L}_{bc} &= \lambda_r \mathcal{L}_r(R, R') + \lambda_t \mathcal{L}_t(\mathbf{t}, \mathbf{t}') + \lambda_s \mathcal{L}_s(\mathbf{s}, \mathbf{s}') \\ &= \sum_{t=1}^{T} [\lambda_r CE(R_i, R'_i) + \lambda_t CE(\mathbf{t}_i, \mathbf{t}'_i) + \lambda_s CE(\mathbf{s}_i, \mathbf{s}'_i)] \end{aligned} \tag{4}$$

where $\mathcal{L}_r$, $\mathcal{L}_t$ and $\mathcal{L}_s$ measure the rotation action loss, translation action loss and scaling action error respectively. λ_r, λ_t and λ_s are scale factors which are set to 0.3, 0.6 and 0.1 in our method since translation fitting policy is more difficult to converge.

After behavior clone for auxiliary training, we can update the agent by compositing reinforcement learning. Here, we employ Proximal Policy Optimization (PPO) algorithm [22] for the policy training. The loss function of PPO $\mathcal{L}_{ppo}$ is defined as:

$$\mathcal{L}_{ppo} = \mathcal{L}_{clip} + \lambda_{val} \mathcal{L}_{val} + \lambda_{etp} \mathcal{L}_{etp} \tag{5}$$

where $\mathcal{L}_{clip}$ is the clipped surrogate objective that is defined as the minimum of clipped and original advantage value function [22]. $\mathcal{L}_{value}$ is defined to measure the squared-error between the value function and the accumulated rewards $\sum_{t \geq 0} \gamma_t r_t$. $\mathcal{L}_{etp}$ is an entropy bonus that is added to ensure sufficient exploration. λ_{val} and λ_{etp} are scale factors and set to 0.3 and 0.001 in our method. The overall procedure of the composited reinforcement learning and behavior clone training is summarized in Algorithm 1.

4.4 Optimization with Kinematic Constraints

Within the reinforced agent and the retrieved reference point cloud, we can infer per-part 9D pose $\{R^{(k)}, \mathbf{t}^{(k)}, \mathbf{s}^{(k)}\}_{k=1}^{K}$ for an unseen articulated object under the joint-centric pose modeling. However, since the 9D pose for every part is estimated individually, the estimation results might lead to potentially physical errors. To cope with this issue, we propose an articulation 9D pose optimization approach that considers kinematic constraints to further refine the pose estimation results. In our method, given a part pair's 9D poses

Algorithm 1: Composited Reinforcement Learning and Behavior Clone for Agent Training

input : Observed point cloud $\mathcal{P}$, retrieved reference point cloud $\mathcal{P}'$

Initialize policy network $\pi(\cdot|s_t)$
Initialize replay buffer B
for N *trajectories* **do**
  for T *action steps* **do**
    Extract feature vectors v and v' from $\mathcal{P}$ and $\mathcal{P}'$
    Concatenate v and v' to obtain state representation s_t
    Sample action a_t and predict value $\hat{v}$ using $\pi(\cdot|s_t)$
    Receive reward r_t and transform $\mathcal{P}'$ by action a_t
  end
  Add trajectory to replay buffer B
end
Compute accumulated rewards $\sum_{t\geq 0}\gamma_t r_t$
Shuffle replay buffer
for *samples in replay buffer* B **do**
  Predict a new policy $\pi(\cdot|s_t)_{new}$ and value $\hat{v}_{new}$
  Compute behavior clone loss $\mathcal{L}_{bc}$
  Compute ppo loss $\mathcal{L}_{ppo}$ using $\pi(\cdot|s_t)$, $\pi(\cdot|s_t)_{new}$, $\hat{v}$ and $\hat{v}_{new}$
  Update agent by total loss $\mathcal{L} = \mathcal{L}_{bc} + \mathcal{L}_{ppo}$
end

$\{R^{(k1)}, \mathbf{t}^{(k1)}, \mathbf{s}^{(k1)}\}$ and $\{R^{(k2)}, \mathbf{t}^{(k2)}, \mathbf{s}^{(k2)}\}$ that are transformed by joint-centric pose represnetation $\theta^{(k1)}$ and $\theta^{(k2)}$, the energy function consists of two components: part-oriented energy E_P and joint-oriented energy E_J. Specifically, the part-oriented energy function E_P aims to separately solve $k1$ and $k2$ part 6D poses as well as 3D scales, which is defined as:

$$\begin{aligned} E_P =& CD(\mathcal{P}^{(k1)}, (\mathbf{s}^{(k1)}R^{(k1)}\mathcal{P}'^{(k1)} - \mathbf{t}^{(k1)})) \\ &+ CD(\mathcal{P}^{(k2)}, (\mathbf{s}^{(k2)}R^{(k2)}\mathcal{P}'^{(k2)} - \mathbf{t}^{(k2)})) \end{aligned} \tag{6}$$

where we can optimize poses by minimizing the chamfer distance between observed point cloud and 9D transformed reference point cloud. Then, we also introduce joint-oriented energy E_J:

$$E_J = R^{(k1)}\mathbf{u}^{(k1,k2)} - R^{(k1)}\mathbf{u}'^{(k1,k2)} \tag{7}$$

where $\mathbf{u}^{(k1,k2)}$ and $\mathbf{u}'^{(k1,k2)}$ indicate the joint directions of observed point cloud and reference point cloud. Since the 3D scale $\{\mathbf{s}^{(k1)}, \mathbf{s}^{(k2)}\}$ is kinematically independent during articulated object's motion, we fix $\{\mathbf{s}^{(k1)}, \mathbf{s}^{(k2)}\}$ and only optimize the 6D poses $\{R^{(k1)}, \mathbf{t}^{(k1)}\}$ and $\{R^{(k2)}, \mathbf{t}^{(k2)}\}$. Note that we do not optimize the joint distance because different instances might share different joint anchor points. Finally, we adopt a optimization solver to compute the refined 9D pose by optimizing the total energy function $E = E_P + E_J$. Note that in practice we optimize the joint states $\Theta = \{\theta^{(k)}\}_{k=1}^{K}$ as displacement of 9D pose and then transform it into 9D pose within the joint-centric pose modeling used.

5 EXPERIMENTS

5.1 Experimental Settings

Implementation. During the data pre-processing, input point clouds are sampled into 2,048 points and the objects in RGB-D images are also cropped and projected into the point cloud as the network inputs. The initial learning rate is 0.001 and decreases by 0.1 per 10 epochs. During training the agent by reinforcement learning, the rotation, translation and scaling actions consist of 11 step sizes per axis in positive and negative directions. The hyper-parameters are $\lambda_r = 0.3$, $\lambda_t = 0.6$, $\lambda_s = 0.1$, $\lambda_{val} = 0.3$ and $\lambda_{etp} = 0.0001$.

Datasets and Metrics. For category-level articulated 9D pose estimation, we evaluate our ArtPERL on ArtImage [33], ReArtMix [18] and RobotArm [18] datasets, which range from synthetic point cloud to real-world RGB-D image. The evaluation metrics are degree error for 3D rotation, distance error for 3D translation and 3D Intersection over Union (IoU) for 3D scale in synthetic single object observation. We also use Average Precision (AP) to measure the 9D pose estimation performance in multi-object observation where the error is less than n°, m cm distance and more than l 3D IoU. We use the bounding box results provided by [18] for object detection files to evaluate AP.

Baselines. For comparison, we firstly evaluate two deep learning based approaches: A-NCSH [14] and OMAD [33], which adopt NOCS [27] and articulation deformations for pose estimation. In terms of reinforcement learning methods, we use SporeAgent [2] algorithm as a baseline that is modified for our category-level articulated object 9D pose estimation task. Since these SporeAgent requires a complete object point cloud for registration, we add our articulated object retrieval before applying it. Note that we do not select ReArtNet [18] as a baseline because its different problem setting might cause an unfair comparison.

5.2 Experiments on Point Cloud Observations

We report the results of ArtPERL evaluated on point cloud input dataset ArtImage [33], which contains 5 categories (laptop, eyeglasses, dishwasher, scissors and drawer) of point clouds generated with the aligned models from PartNet-Mobility [30]. Table 1 shows the quantitative results of ArtImage test set. As it can be seen, the best 9D pose estimation result lies on category eyeglasses, with **4.1**°,**6.2**°,**6.0**° for rotation degree error and **0.047**,**0.095**,**0.091** for translation error. This can be explained by that the joint-centric modeling strategy used in our method can effectively solve small part 9D perception problem, while part-centric modeling relies on accurate part segmentation performance. For the 3D IoU metric, our ArtPERL could also obtain an obvious improvement compared to A-NCSH [14] or OMAD [33]. Moreover, with the benefits of joint-centric articulation modeling, the 9D poses for the instances with prismatic joints (such as drawer) show to be consistent on four parts and our ArtPERL can achieve averagely **3.5**° rotation error, which also outperforms the state-of-the-art methods. Compared with the reinforcement learning method SporeAgent [2], our method also shows a great advantage that obtains **16.7**°, **36.4**° and **39.1**° rotation improvement on three parts of eyeglasses. We can conclude that our method can fully utilize kinematic constraints in ArtOPE task rather than per-part pose processing strategy. Qualitative results are shown in Fig. 3.

Table 1: Comparison with state-of-the-arts on ArtImage dataset. The categories laptop, eyeglasses, dishwasher and scissors contain only free joint and revolute joints, and the drawer category contains free joint and prismatic joints.

Category	Method	Per-part 9D Pose		
		rotation error (°) ↓	translation error (m) ↓	3D IoU (%) ↑
Laptop	A-NCSH [14]	5.3, 5.4	0.054, 0.043	56.7,40.2
	OMAD [33]	5.4, **4.3**	0.062, 0.061	43.5,24.1
	SporeAgent [2]	14.6, 22.5	0.120, 0.148	25.1,13.5
	ArtPERL (Ours)	**4.9**, 4.7	**0.053**, **0.066**	**64.6**,**50.4**
Eyeglasses	A-NCSH [14]	**3.7**, 22.3, 23.2	0.049, 0.313, 0.324	52.5,40.2,39.6
	OMAD [33]	4.9, 7.5, 7.5	0.062, 0.103, 0.104	22.8,20.5,21.4
	SporeAgent [2]	20.8, 42.6, 45.1	0.198, 0.235, 0.236	9.5,6.4,6.2
	ArtPERL (Ours)	4.1, **6.2**, **6.0**	**0.047**, **0.095**, **0.091**	**58.6**,**46.5**,**51.7**
Dishwasher	A-NCSH [14]	4.0, 4.8	0.059, 0.123	84.3,56.2
	OMAD [33]	6.0, 6.2	0.104, 0.142	66.5,38.9
	SporeAgent [2]	12.5, 21.5	0.146, 0.184	43.8,28.6
	ArtPERL (Ours)	**3.9**,**4.3**	**0.055**,**0.079**	**89.3**,**67.6**
Scissors	A-NCSH [14]	**2**,**0**, 2.9	0.035, **0.025**	**46.5**,44.8
	OMAD [33]	3.9, 3.4	0.048, 0.039	35.6,34.5
	SporeAgent [2]	9.6, 9.7	0.087, 0.090	28.4,28.0
	ArtPERL (Ours)	2.2, **2.6**	**0.031**, 0.042	40.9,**46.3**
Drawer	A-NCSH [14]	**2.8**, 3.5, 3.9, 2.9	**0.045**, 0.155, 0.157, **0.075**	**90.2**,**81.5**,78.4,**82.7**
	OMAD [33]	4.4, 4.4, 4.4, 4.4	0.111, 0.143, 0.144, 0.115	75.8,73.4,70.2,71.3
	SporeAgent [2]	11.2, 19.6, 22.5, 18.8	0.154, 0.228, 0.275, 0.201	60.3,24.2,10.7,24.4
	ArtPERL (Ours)	3.5, **3.5**, **3.5**, 3.5	0.061, **0.112**, **0.121**, 0.104	84.8,78.6,**79.0**,81.2

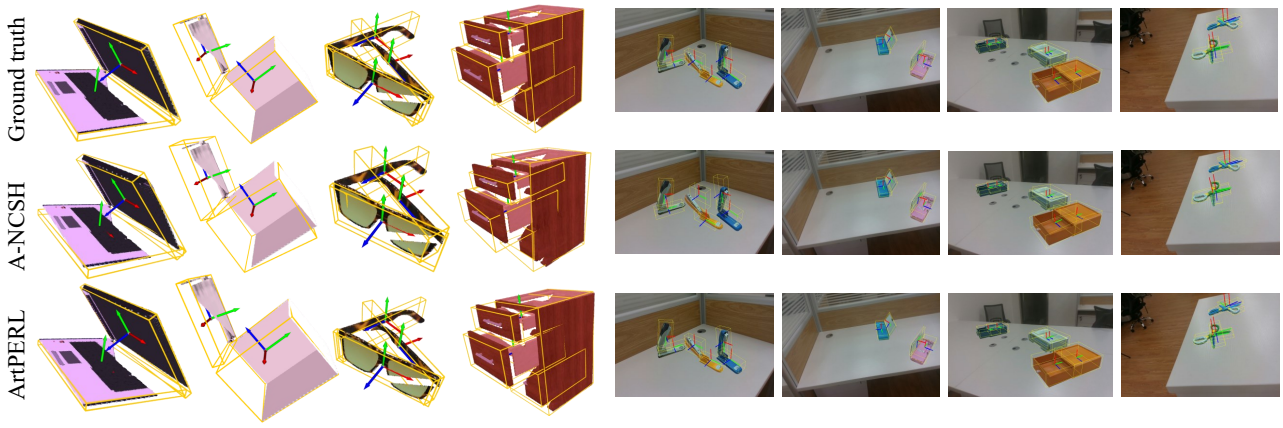

Figure 3: Qualitative results on point clouds observation (left) and RGB-D images (right)

5.3 Ablation Study

Rewards analysis. We show the received rewards and Chamfer distance curves on five categories of ArtImage during agent training in Fig. 5. We can observe that the accumulated rewards are negative at the beginning of training, but our method will receive up to **0.35** reward when the network converges. This concludes that ArtPERL can effectively train a pose-fitting agent for articulated objects. In addition, the Chamfer distances are also declined to flatten with the training progress and finally drop at approximately **0.1** to **0.15**, which reflect the successful pose fitting. Among the five categories, the dishwasher is the easiest one to train a pose estimation agent because of its simple kinematic structure (only one motion joint) and geometric shape (large-size objects).

Inference Time Analysis. We report the inference time metric to evaluate the pose estimation speed by comparing ArtPERL with A-NCSH and OMAD. Table 2 shows the experimental results. As it can be seen, since there is no dense prediction adopted in our method while A-NCSH [14] strongly employs point-to-point correspondence prediction, our ArtPERL shows much better pose estimation speed with average **1.0s** per instance. This can also be

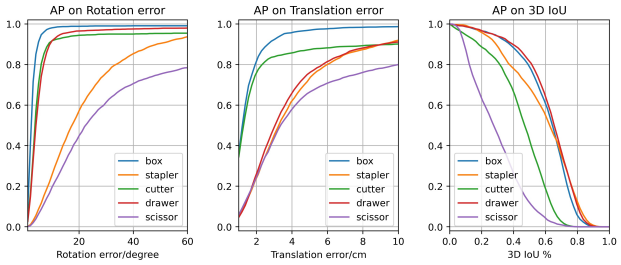

Figure 4: Pose estimation results on ReArtMix dataset

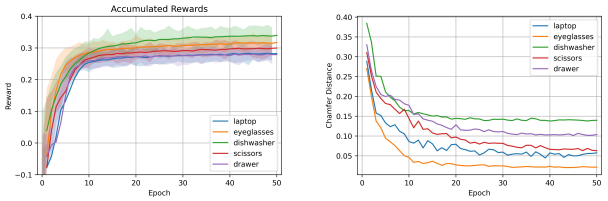

Figure 5: Accumulated rewards (left) and Chamfer distance (right) received during reinforced agent training

Table 2: Inference time comparison on ArtImage dataset

Category	A-NCSH [14]	OMAD [33]	ArtPERL
Laptop	9.0	1.6	**0.9**
Eyeglasses	11.9	2.5	**1.0**
Dishwasher	5.5	1.6	**0.9**
Scissors	6.5	1.7	**0.8**
Drawer	16.5	1.9	**1.1**

observed that the inference time of ArtPERL does not strongly rely on the number of parts, while OMAD requires per-part optimization with much time consumption.

5.4 Experiments on RGB-D Images

Fig. 4 shows the category-level articulated object 9D pose estimation results on ReArtMix dataset [18], which contains mixed reality RGB-D images where the background scenes are captured from real-world and articulated objects are rendered with real-scanned textured models. Observing from Fig. 4, we can see that our ArtPERL achieves mean AP of **48.9%**, **77.4%** and **57.8%** for rotation error 5°, translation error 5cm and 3D IoU@0.5 respectively. Specifically, the box is the category that receives the best pose estimation performance with **82.3%** AP on rotation error 5°, while scissors is the most difficult category to estimate 9D pose since their small sizes. Another impact is the articulated object detection whose results are calculated by RetinaNet [16], where the detection missing might also lower the AP values. Regarding 3D IoU, our ArtPERL can also perform well in estimating the 3D scales for most of the common articulated objects. Qualitative results are shown in Fig. 3.

5.5 Experiments on the Multi-Hinged Object

To further prove the generalization capacity of ArtPERL, we select a special case to evaluate our method: the multi-hinged articulated object. The experiment benchmark is the RobotArm dataset provided by [18] that contains RGB-D images captured from a 7-part and 6-joint Franka Panda Robot Arm for instance-level articulated object 6D pose estimation evaluation. The experimental results are demonstrated in Table 3. Our ArtPERL can perform much better on rotation and translation estimation of 7 robot arm parts compared to A-NCSH [14]. Specifically, we can receive only 5.7°,5.9°,5.8° rotation degree error in the first three parts. However, it is undeniable that our method also suffers from the effect of multi-depth structure, where accumulative errors might lead to a severe impact on ArtPERL. Qualitative results are shown in Fig. 6.

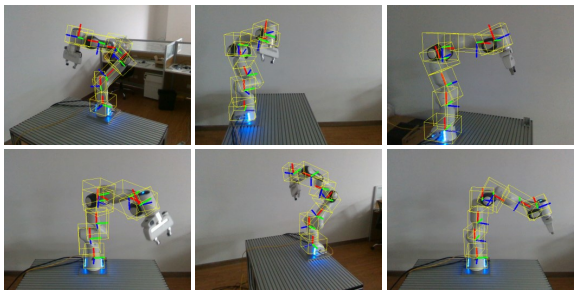

Figure 6: Qualitative results on multi-hinged robot arms

Table 3: Pose estimation results on Robot Arm dataset

Per-part Rotation Error (°)							
part ID	1	2	3	4	5	6	7
A-NCSH [14]	7.8	7.9	10.3	10.5	11.2	16.4	23.5
ArtPERL	5.7	5.9	5.8	8.4	10.1	12.5	19.6
Per-part Translation Error (m)							
part ID	1	2	3	4	5	6	7
A-NCSH [14]	0.012	0.044	0.067	0.066	0.079	0.236	0.403
ArtPERL	0.008	0.012	0.014	0.037	0.058	0.136	0.225

6 CONCLUSION

In this work, we formulate the category-level articulated object 9D pose estimation task as a reinforcement learning problem, and introduce the ArtPERL framework to tackle this issue without intra-category object representation learning. The ArtPERL designs a joint-centric articulation 9D pose to model the articulation motion, and propose a step-wise action space as well as a reward function to achieve a 9D pose fitting procedure. The predicted 9D poses are also refined by our proposed optimization strategy. Experiments demonstrate that our approach is able to obtain state-of-the-art performance on various articulated object observations.

ACKNOWLEDGEMENTS

This work is supported in part by National Natural Science Foundation of China (NSFC) under Grants 72188101, 62020106007 and 62272435, and the Major Project of Anhui Province under Grant 202203a05020011.

REFERENCES

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

[2] Dominik Bauer, Timothy Patten, and Markus Vincze. 2022. SporeAgent: Reinforced Scene-level Plausibility for Object Pose Refinement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 654–662.

[3] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. 2014. Learning 6d object pose estimation using 3d object coordinates. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*. Springer, 536–551.

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

[5] Zhe Cao, Yaser Sheikh, and Natasha Kholgade Banerjee. 2016. Real-time scalable 6DOF pose estimation for textureless objects. In *2016 IEEE International conference on Robotics and Automation (ICRA)*. IEEE, 2441–2448.

[6] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. 2020. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11973–11982.

[7] Guoxian Dai, Jin Xie, and Yi Fang. 2018. Siamese cnn-bilstm architecture for 3D shape representation learning.. In *IJCAI*. 670–676.

[8] Jianfeng Dong, Xianke Chen, Minsong Zhang, Xun Yang, Shujie Chen, Xirong Li, and Xun Wang. 2022. Partially Relevant Video Retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*. 246–257.

[9] Andreas Doumanoglou, Vassileios Balntas, Rigas Kouskouridas, and Tae-Kyun Kim. 2016. Siamese regression networks with efficient mid-level feature extraction for 3d object pose estimation. *arXiv preprint arXiv:1607.02257* (2016).

[10] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. 2010. Model globally, match locally: Efficient and robust 3D object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*. Ieee, 998–1005.

[11] Jia Gong, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. 2022. Meta agent teaming active learning for pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11079–11089.

[12] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. 2020. Hot-net: Non-autoregressive transformer for 3d hand-object pose estimation. In *Proceedings of the 28th ACM International Conference on Multimedia*. 3136–3145.

[13] Alexander Krull, Eric Brachmann, Sebastian Nowozin, Frank Michel, Jamie Shotton, and Carsten Rother. 2017. Poseagent: Budget-constrained 6d object pose estimation via reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6702–6710.

[14] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. 2020. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3706–3715.

[15] Zhaoqun Li, Cheng Xu, and Biao Leng. 2019. Angular triplet-center loss for multi-view 3d shape retrieval. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 8682–8689.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

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

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

[19] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. 2021. A-sdf: Learning disentangled signed distance functions for articulated shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13001–13011.

[20] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 2017. 6-dof object pose from semantic keypoints. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2011–2018.

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

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

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

[24] Yifei Shi, Junwen Huang, Xin Xu, Yifan Zhang, and Kai Xu. 2021. Stablepose: Learning 6d object poses from geometrically stable patches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15222–15231.

[25] Joel Vidal, Chyi-Yeu Lin, Xavier Lladó, and Robert Martí. 2018. A method for 6D pose estimation of free-form rigid objects using point pair features on range data. *Sensors* 18, 8 (2018), 2678.

[26] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. 2019. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3343–3352.

[27] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. 2019. 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*. 2642–2651.

[28] Yongming Wen, Yiquan Fang, Junhao Cai, Kimwa Tung, and Hui Cheng. 2021. GCCN: Geometric Constraint Co-attention Network for 6D Object Pose Estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2671–2679.

[29] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J Guibas. 2021. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13209–13218.

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

[31] Jin Xie, Guoxian Dai, Fan Zhu, Edward K Wong, and Yi Fang. 2016. Deepshape: Deep-learned shape descriptor for 3d shape retrieval. *IEEE transactions on pattern analysis and machine intelligence* 39, 7 (2016), 1335–1345.

[32] Haoyu Xiong, Haoyuan Fu, Jieyi Zhang, Chen Bao, Qiang Zhang, Yongxi Huang, Wenqiang Xu, Animesh Garg, and Cewu Lu. 2023. RoboTube: Learning Household Manipulation from Human Videos with Simulated Twin Environments. In *Conference on Robot Learning*. PMLR, 1–10.

[33] Han Xue, Liu Liu, Wenqiang Xu, Haoyuan Fu, and Cewu Lu. 2021. OMAD: Object Model with Articulated Deformations for Pose Estimation and Retrieval. (2021).

[34] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. 2022. OakInk: A Large-scale Knowledge Repository for Understanding Hand-Object Interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20953–20962.

[35] Xun Yang, Meng Wang, Luming Zhang, Fuming Sun, Richang Hong, and Meibin Qi. 2016. An efficient tracking system by orthogonalized templates. *IEEE Transactions on Industrial Electronics* 63, 5 (2016), 3187–3197.

[36] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. 2018. Deep part induction from articulated object pairs. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.

[37] Yang You, Ruoxi Shi, Weiming Wang, and Cewu Lu. 2022. Cppf: Towards robust category-level 9d pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6866–6875.

[38] Man Zhou, Rujing Wang, Chengjun Xie, Liu Liu, Rui Li, Fangyuan Wang, and Dengshan Li. 2021. ReinforceNet: A reinforcement learning embedded object detection framework with region selection network. *Neurocomputing* 443 (2021), 369–379.