



U-COPE: Taking a Further Step to Universal 9D Category-Level Object Pose Estimation

Li Zhang^{1,2,5}, Weiqing Meng³, Yan Zhong⁴, Bin Kong¹, Mingliang Xu²,
Jianming Du¹, Xue Wang¹, Rujing Wang¹, and Liu Liu⁶(✉)

¹ Hefei Institute of Physical Science, Chinese Academy of Sciences, Beijing, China
zanly20@mail.ustc.edu.cn

² University of Science and Technology of China, Hefei, China

³ Anhui University, Hefei, China

⁴ School of Mathematical Sciences, National Engineering Research Center of Visual
Technology, Peking University, Beijing, China

⁵ Atribot, Shenzhen, China

⁶ Hefei University of Technology, Hefei, China

Abstract. Rigid and articulated objects are common in our daily lives. Pose estimation tasks for both types of objects have been extensively studied within their respective domains. However, a universal framework capable of estimating the pose of both rigid and articulated objects has yet to be reported. In this paper, we introduce a Universal 9D Category-level Object Pose Estimation (U-COPE) framework, designed to address this gap. Our approach offers a novel perspective on rigid and articulated objects, redefining their pose estimation problems to unify them into a common task. Leveraging either 3D point cloud or RGB-D image inputs, we extract Point Pair Features (PPF) independently from each object part for end-to-end learning. Moreover, instead of direct prediction as seen in prior art, we employ a universal voting strategy to derive decisive parameters crucial for object pose estimation. Our network is trained end-to-end to optimize three key objectives: Joint Information, Part Segmentation, and 9D pose estimation through parameter voting. Extensive experiments validate the robustness of our method in estimating poses for both rigid and articulated objects, which demonstrates the generalizability to unseen object instances, too. Notably, our approach achieves state-of-the-art performance on synthetic datasets and real-world datasets.

Keywords: Rigid objects · Articulated objects · Pose estimation

1 Introduction

In our daily lives, we encounter a diverse array of tools, equipment, and machinery, which can be broadly categorized into two main structural types: (1) rigid objects, characterized by fixed and unchanging shapes, and (2) articulated objects, consisting of interconnected components that permit relative motion between them. Accurate pose

L. Zhang and W. Meng—Contributed equally. This work was done when Li Zhang was an intern at Atribot Inc.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

A. Leonardis et al. (Eds.): ECCV 2024, LNCS 15068, pp. 254–270, 2025.

https://doi.org/10.1007/978-3-031-72684-2_15

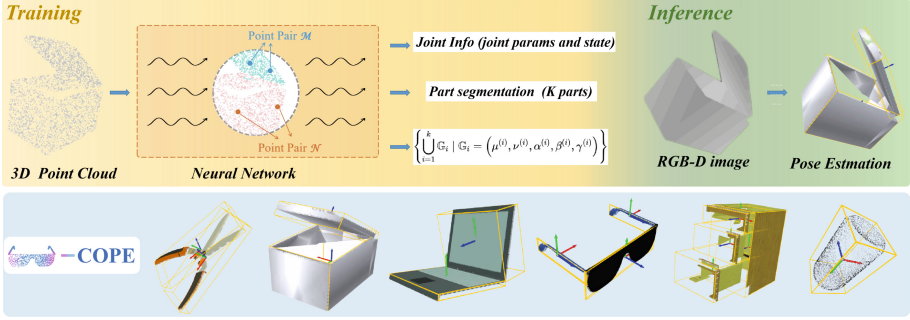


Fig. 1. Training and inference paradigm of U-COPE (top) and Quantitative results of object pose estimation (bottom). For U-COPE, we apply a partial point cloud as input, pick point pairs from each part used for the network, and get the end-to-end output. Please note that we regard a rigid object as a special articulated object, which has only one part. Its procedure is similar to the flow chart above. For visualizations, we show the 9D category-level poses (3D rotation, 3D translation, and 3D scale). The first five examples are for articulated objects from the articulated dataset ArtImage and the last example is for the rigid object from CAMERA25.

estimation for both categories is integral to numerous computer vision and robotics applications, yet it remains an area that has not received exhaustive studies, such as augmented reality [2, 4, 8, 39], 3D scene understanding [20, 22, 36, 50], and robotic manipulation [6, 16, 30, 31, 37]. While considerable progress has been made in recent years, with significant advancements in pose estimation methods for both rigid [10, 44, 47] and articulated objects [18, 24], certain challenges persist. Despite the proliferation of research efforts, two primary issues still exist in object pose estimation (Fig. 1):

The first problem is **Discontinuity on pose estimation of different objects**. Our scenarios contain various types of objects but current mainstream methods either focus only on rigid or articulated objects. There are mainly 3 challenges to a universal framework. The first challenge is *complexity*. Rigid and articulated objects mainly come from daily life, industrial equipment, robotic interaction objects, and other scenarios with very different shapes or multi-hinged parts, posing a challenge to generalizability, especially for unseen object instances. The second issue is *differences* in shape and structure, functional characteristics, and motion patterns. Rigid objects do not have joints and thus do not have flexible pose transformation similar to articulated objects, which makes them subject to different customization methods for pose estimation. The third is *deployment costs*. In practical applications, attention must be given to both rigid and articulated object pose estimation tasks, and if different frameworks are used to handle these tasks, it will involve fusion and integration issues that need to be addressed to resolve incompatibilities and consistencies between models.

The second problem is **Pose Modeling Problem**. Mainstream object pose estimation traditionally relies on two classical methods: dense prediction and key points. The former is exemplified by NOCS [44] and A-NCSH [24], which aim to predict dense correspondences between object pixels and the Normalized Object Coordinate Space (NOCS). The latter searches for features through key points matching, com-

monly employed in pose estimation [45] and tracking tasks [43,48]. However, these direct methods for estimating object pose often yield coarse results or struggle to converge and maintain meaningful features, leading to ill-posed pose modeling problems. To circumvent these issues, a preliminary voting scheme proposed by Drost [11] offers an indirect approach that utilizes point pair features to match against an object database. This voting strategy sidesteps the limitations associated with direct pose prediction methods and exhibits greater robustness owing to its $SO(3)$ invariance. Building upon this approach, we extend and generalize the scheme to the task of estimating poses for arbitrary objects (mainly including rigid and articulated objects).

In response to these challenges, we introduce a universal and end-to-end framework for both rigid and articulated object pose estimation tasks, termed U-COPE, designed to accommodate unseen object instances. Specially speaking, we first revisit and model the structural relationships inherent in rigid and articulated objects, abstracting them into a unified structure characterized by an object with K parts and $K - 1$ joints (Notably, an articulated object embodies this structure when $K \geq 2$, while it simplifies to a rigid object configuration when $K = 1$). Based on this consideration, U-COPE outputs the complete 9D pose of the object with the given partial observations. Our network first employs PointNet++ to extract useful features. A decoder then decomposes these features to generate mask vectors for part segmentation prediction. These features are also fed into the PPF encoder, which processes the point pair features (PPF [11]). The re-modulated features are used for joint information prediction and per-part pose estimation. For joint information, we use a heatmap-offset strategy for implicit joint prediction. For per-part pose estimation (Rotation (R), Translation (\mathbf{t}), Scale (\mathbf{s})), key parameters vote to simplify constraint and optimization processes. Generally speaking, our method is more similar to extending CPPF [49] into the unified rigid and articulated objects which is also involved with point pairs, but has 2 obvious differences as follows: 1) Modalities for the identification of point pairs features. Our point pairs features come from feature decoupling of a $SO(3)$ invariant network and can learn and adjust adaptively part-wise, while CPPF uses a fixed point pair feature selection approach 2) Task-driven. our framework outputs joint info, part segmentation and parameter groups related to pose estimation in an end-to-end way, while only pose estimation can be found in CPPF.

Overall, our contributions can be summarized as three folds:

- We streamline the pose estimation task for both rigid and articulated objects into a unified framework and adopt a generalized observational perspective. By decomposing both object types into K parts linked by $K - 1$ joints, we revisit the internal compositional relationships inherent in rigid and articulated objects, facilitating pose estimation for these entities.
- We propose an end-to-end, universal pose estimation framework for rigid and articulated objects. Employing a point cloud as input, we build a multi-branch optimization task with 3 outputs, *i.e.* Joint info, Part segmentation, and 9D category-level poses for each part.
- Extensive experiments demonstrate the superiority of our U-COPE to existing state-of-the-art methods. Experiments on real-world scenarios also demonstrate the gen-

eralization capacity of our method. In the meantime, our method is generalized for unseen object instances from single-depth images or partial point clouds.

2 Related Work

2.1 Rigid Object Pose Estimation

Pose estimation for rigid objects plays a pivotal role in various computer vision applications [15,41,52], including robotic tasks [1,23], virtual reality systems [7,53], and pose tracking applications [5,42,46]. It involves precisely determining the position and orientation of rigid objects within a given scene. Traditional methods [25,29,34,51] often employ feature-based techniques, where distinctive key points or landmarks are detected on the object and matched with a pre-built 3D model. These methods typically entail solving the perspective-n-point (PnP) problem to estimate the pose [9], a process sensitive to occlusions, cluttered backgrounds, and variations in lighting conditions. An alternative approach [21,24] focuses on refining the initial pose estimate by incorporating geometric priors or constraints. Iterative algorithms, such as the iterative closest point (ICP) [14], is commonly employed to optimize the pose estimate by minimizing the geometric distance between observed and model points. Additionally, methods leveraging depth information [19,32] have shown promise in enhancing the accuracy and robustness of pose estimation, particularly in scenarios featuring texture-less objects or challenging lighting conditions.

Despite the advancements in rigid object pose estimation, challenges such as handling occlusions, partial observations, and achieving real-time performance persist.

2.2 Articulated Object Pose Estimation

Category-level Object Pose Estimation was pioneered in NOCS [44], introducing an efficient scheme for estimating poses of unseen targets within the same category. Building on this foundation, subsequent works [3,13,33,38] have proposed improved methods to address challenges in more complex scenarios. However, these methods often struggle to extend seamlessly to articulated objects commonly encountered in daily life and industrial settings. Articulated objects, characterized by interconnected parts exhibiting relative motion (e.g., computers, chests of drawers, robotic arms), present unique challenges in pose estimation crucial for tasks such as robotic vision. Inspired by NOCS, Li et al. [24] introduced a novel category-level approach capable of accommodating object instances not encountered during training. In this paper, our key idea is to adopt a part-level estimation strategy. This choice reduces system complexity, eliminating the need for manual feature design or rule-based approaches, and streamlines the debugging process through an end-to-end methodology.

By end-to-end optimization of all optimization objectives, our approach facilitates the attainment of globally optimal solutions. It avoids the problem that previous two-stage approaches may need to optimize each stage separately and necessitate data transfer between stages, leading to unexpected local optima.

3 Problem Statement

To establish a universal object pose estimation framework applicable to both rigid and articulated objects, our core idea is modeling pose estimation as per-part rigid transformations. Here, we formulate a new perspective for the category-level universal object pose estimation task via a voting strategy named U-COPE. Formally, we conceptualize both objects within a universal structure consisting of K parts associated with $K - 1$ joints. Notably, an object is considered rigid only when $K = 1$. Our approach addresses the following problem statement: given a 3D observed point cloud $\mathcal{P} = \bigcup_{i=0}^N \{\mathbf{p}_i \in \mathbb{R}^{N \times 3}\}$, we conduct predictions under unknown CAD models utilizing an end-to-end framework. The outputs of our framework include 1) Joint info, comprising a pivot and a joint direction. 2) Part segmentation, denoted as $\mathbf{p}_i \in \mathbb{R}^{N \times K}$ with one-hot labels. 3) 9D pose estimation for K parts, which encompasses per-part 3D rotation $R^{(k)}$, per-part 3D translation $\mathbf{t}^{(k)}$, and per-part 3D amodal bounding box scales $\mathbf{s}^{(k)}$. These parameters are determined through a voting mechanism involving $\mu^{(k)}, \nu^{(k)}, \alpha^{(k)}, \beta^{(k)}, \gamma^{(k)}$.

Specifically speaking, in our proposed scheme, we begin with an unknown object instance from a known category as input. Utilizing a specialized encoder, we extract features from the input data. Diverging from conventional approaches, which focus solely on optimizing the pose of the current object, our framework targets the precise estimation of object poses at the per-part level. This approach necessitates simultaneous optimization of joint information and segmentation additionally. To address this, we tackle the following optimization challenges: 1) We transform the learned features into key variables conducive to predicting joint information. Heatmaps (H_i) denote the likelihood of \mathbf{p}_i acting as a pivot, while offsets (O_i) indicate the deviation of \mathbf{p}_i from the joint orientation, facilitating joint optimization. 2) Each point is encoded as a one-hot vector, reflecting its associated part within the object. 3) Given the complexities and convergence challenges of direct pose estimation methods, we introduce key parameters responsive to each part’s pose. These parameters enable accurate prediction of the 9D pose at the per-part level through our voting strategy. By adopting this approach, we effectively unify the pose estimation tasks for both rigid and articulated objects. The resulting per-part rigid transformations describe how individual parts transition from their canonical space to the camera space.

4 Methodology

4.1 Overview

Our framework is illustrated in Fig. 2. Essentially, our U-COPE comprises three interconnected modules, each serving a distinct purpose in the overall process. Firstly, we employ feature extraction from the input data, facilitating downstream network utilization. This initial step is complemented by a feature re-modulating and decoupling architecture, which plays a pivotal role in enhancing per-part pose estimation accuracy (Sect. 4.2). Afterward, our framework proceeds with joint and pose estimation through optimization within each module (detailed in Sect. 4.3). Lastly, to handle both rigid and articulated objects effectively, we incorporate a voting scheme, as discussed in Sect. 4.4, providing a comprehensive solution for both object types.

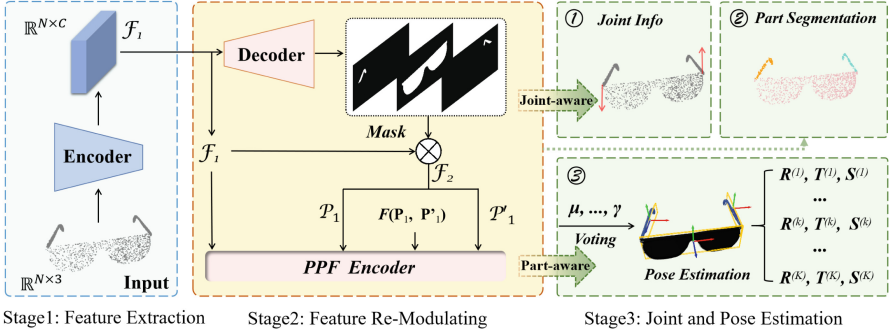


Fig. 2. The overview of our U-COPE framework. Formally, taking a 3D point cloud or partial point cloud from RGB-D images as input, our U-COPE consists of the following components: (1) **Feature Extraction** from input used for the downstream network. (2) **Feature Re-Modulating** and decoupling architecture to help per-part pose estimation (Sect. 4.2). (3) **Joint and Pose Estimation** with optimization for each branch (Sect. 4.3), which is achieved by the proposed voting scheme for both rigid and articulated objects (Sect. 4.4). Therefore, our method can output 3 targets directly, *i.e.*, Joint info, Part segmentation (K parts), and 9D pose estimation results, as described in Sect. 3.

4.2 Feature Extraction and Re-modulating

We employ either a 3D point cloud or an RGB-D image as input, which is subsequently processed by an encoder, specifically PointNet++ [35], to extract features denoted as \mathcal{F}_1 . Following this, \mathcal{F}_1 is decomposed into the mask vector of each part using a dedicated decoder. The resulting features are then combined with \mathcal{F}_1 via the Hadamard product to derive point embedding features specific to each part. These features are utilized for predicting joint information and part segmentation. Moreover, we group the point embeddings \mathcal{P}_1 and \mathcal{P}'_1 , along with the Point Pair Features (PPF) adapted from the work of Drost *et al.* [11], into the PPF encoder to optimize parameters related to pose estimation. Finally, we conduct joint and pose estimation simultaneously in the final stage: Joint information, Part segmentation (for K parts), and 9D pose estimation results, determined by a series of parameter clusters associated with pose estimation.

4.3 Joint and Pose Estimation

Joint Estimation. Since a joint comprises a pivot point \mathbf{q} to determine the location and a joint direction \vec{u} to determine the orientation, we define the joint parameters $\phi = (\mathbf{q}, \vec{u})$, where $\mathbf{q} \in \mathbb{R}^3$ and $\vec{u} \in \mathbb{R}^3$. To this end, we adopt an implicit way to predict the joint info, which is illustrated in Fig. 3. Specifically speaking, given the feature \mathcal{F}_1 extracted by the encoder in Sect. 4.2, we put it into the PPF encoder to conduct feature re-modulating process. Subsequently, for pivot point prediction, we use a three-layer MLP to generate $4N$ channels, where N channels indicate the heatmap H_i of each point \mathbf{p}_i and $3N$ channels indicate the offset between \mathbf{p}_i and its adjacent joint. During each iteration, we select the point with the highest probability (H_i) as the

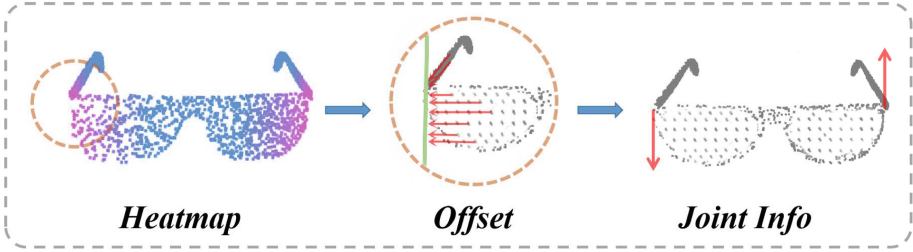


Fig. 3. Illustration for confirmation of Joint information. The color depth in the heatmap signifies the probability of a point being a pivot, with deeper red indicating higher probability and deeper blue indicating lower probability.

pivot and train \hat{p} and \hat{u} . Mathematically, the final predicted pivot \hat{p} can be formulated as Eq. 1:

$$\hat{p} = \frac{1}{N} \sum_{i=1}^N \{H_i(\mathbf{p}_i + O_i)\} \tag{1}$$

For joint direction prediction, we also adopt a three-layer MLP to regress $3N$ channels, which denotes the prediction \hat{u} . The final result of the \hat{u} can be the average prediction over all the points.

Taking k -th part as an example, the loss function used for this branch is as follows:

$$\mathcal{L}_{joint}^{(k)} = \mathcal{L}_{pvt}^{(k)}(\hat{p}, \mathbf{q}) + \mathcal{L}_{dir}^{(k)}(\hat{u}, \vec{u}) \tag{2}$$

$$\text{where } \mathcal{L}_{pvt}^{(k)}(\hat{p}, \mathbf{q}) = \|(\hat{p} - \mathbf{q}) \times \vec{u}\|_2, \quad \mathcal{L}_{dir}^{(k)}(\hat{u}, \vec{u}) = \|\hat{u} - \vec{u}\|_2 \tag{3}$$

$\mathcal{L}_{pvt}^{(k)}$ is used to supervise the joint location (\hat{p} is the predicted point, which could move arbitrarily along the joint direction, while \mathbf{q} is GT pivot), $\mathcal{L}_{dir}^{(k)}$ is used to supervise the direction of joint (\hat{u} is the predicted direction, while \vec{u} is GT direction).

Note that our output does not include such a branch when dealing with rigid objects. In other words, no joints are required for supervision and learning in this scenario.

Part Segmentation. Given a 3D point cloud $\mathcal{P} = \bigcup_{i=0}^N \{\mathbf{p}_i \in \mathbb{R}^{N \times 3}\}$, we use an encoder to extract features $\mathcal{F}_1 \in \mathbb{R}^{N \times \mathbb{C}}$, where \mathbb{C} is denoted for the channels. And then we use a particular decoder to decouple \mathcal{F}_1 into K Masks ($\{m_k \in \mathbb{R}^{N \times 1}\}_{k=1}^K$) using for K parts. Later on, we calculated Hadamard product ($\mathcal{F} \otimes m_k$) used as re-modulated features $\mathcal{F}_2 \in \mathbb{R}^{N \times \mathbb{C}}$. To this end, we transfer \mathcal{F}_1 and \mathcal{F}_2 into the PPF encoder, which is used for the supervision of Part segmentation (a point cloud with N points and K parts, defined as $\{\mathbf{p}_i \in \mathbb{R}^{N \times K}\}$). The loss function used for this branch $\mathcal{L}_{seg}^{(k)}$ is CE loss.

Similar to the preceding subsection, rigid objects don't need to be segmented in the application scenario which does not contain such a branch in the pipeline.

Pose Estimation. As described in Sect. 4.3, given a rigid or an articulated object \mathbb{O} , we can divided it into K parts, denoted as $\mathbb{O} = \{\tilde{o}_{(1)}, \dots, \tilde{o}_{(i)}, \dots, \tilde{o}_{(k)}\}$. Furthermore, In PPF encoder, we use sample point pairs features \mathcal{F}_3 to output results. It can be formulated as $\mathcal{F}_3 = \text{Concat}(\mathcal{P}_1, \mathcal{P}'_1, F(\mathbf{p}_1, \mathbf{p}_2))$, where $\mathcal{P}_1, \mathcal{P}'_1 \in \mathbb{R}^{N \times C}$ are point embeddings, $F(\mathbf{p}_1, \mathbf{p}_2) \in \mathbb{R}^{N \times C'}$ are point features. then we get $\mathcal{F}_3 \in \mathbb{R}^{N \times (2C+C')}$ using for key parameters prediction. These key parameters groups are related to posing estimation for $\tilde{o}_{(k)}$ derived from the PPF encoder, which can be defined as follows:

$$\mathbb{G}_k = \left(\mu^{(k)}, \nu^{(k)}, \alpha^{(k)}, \beta^{(k)}, \gamma^{(k)} \right) \quad (4)$$

where $k \in [1, K]$, \mathbb{G}_k denotes the beforementioned parameters used for k -th part pose estimation.

For each part, we predict its pose estimation via a voting scheme, which will be introduced in the next section. The loss function used for this branch is as follows:

$$\mathcal{L}_{pose}^{(k)} = \mathcal{L}_{tr}^{(k)} + \mathcal{L}_{rot}^{(k)} + \mathcal{L}_{scale}^{(k)} \quad (5)$$

$$\text{where } \mathcal{L}_{rot}^{(k)} = \mathcal{L}_{right}^{(k)} + \mathcal{L}_{up}^{(k)} \quad (6)$$

where $\mathcal{L}_{tr}^{(k)}$ is KLDiv loss which is used for the supervision of $\mathbf{t}^{(k)}$, $\mathcal{L}_{right}^{(k)}$, $\mathcal{L}_{up}^{(k)}$ are KLDiv loss, they are conducted for the supervision of $R^{(k)}$. we use $\mathcal{L}_{scale}^{(k)}$ to supervise the scales for $\tilde{o}_{(k)}$, implemented by MSE loss.

Finally, the joint loss function for the whole object to train our U-COPE can be formulated as:

$$\mathcal{L}_{total} = \frac{1}{K} \left\{ \lambda_1 \sum_{k=1}^{K-1} \mathcal{L}_{joint}^{(k)} + \lambda_2 \sum_{k=1}^K \mathcal{L}_{seg}^{(k)} + \lambda_3 \sum_{k=1}^K \mathcal{L}_{pose}^{(k)} \right\} \quad (7)$$

where we use a convex combination of the three losses. By default, we set $\lambda_1, \lambda_2, \lambda_3$ to 0.2, 0.2 and 0.6, respectively.

4.4 Voting Scheme

A formal description of our voting scheme is illustrated in the following:

★ **Translation voting scheme.** As depicted in Fig. 4(a₁), define the center of \mathbb{O} as \mathbf{o} , we first choose a series of point pairs (e.g., \mathbf{p}_1 and \mathbf{p}'_1) from it for random for each part $\tilde{o}_{(k)}$. Later on, we denote $\mu^{(k)}$ as the length of the projection of $\|\overrightarrow{\mathbf{p}_1 \mathbf{o}}\|$ in the direction of $\overrightarrow{\mathbf{p}_1 \mathbf{p}'_1}$, while $\nu^{(k)}$ is denoted for the length of $\|\overrightarrow{\mathbf{c} \mathbf{o}}\|$. And then, we conjecture that the part center lies on the circle (i.e., the green circle in Fig. 4(a₁)) with the center \mathbf{c} and the radius ν . Later on, candidate centers are generated on the dashed circle for an interval of $2\pi/K$ for the center voting scheme (Fig. 4(a₂)). and then we examine the votes of all the grids (i.e., $\{g^{(1)}, \dots, g^{(J)}\}$) in the space and determine the center of the

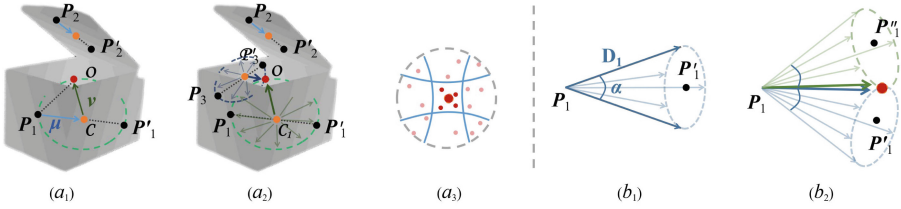


Fig. 4. Illustration for Center voting scheme (Left) and Orientation voting scheme (Right). Please note that we conduct the voting scheme on each part but only one part is on demonstrating for simplicity’s sake.

grid with the most votes as Translation ($\hat{t}^{(k)}$), which is also recognized as the geometric center of $\tilde{o}_{(k)}$ (Fig. 4(a₃)). Initialize a point pair pool $\mathbb{P} = \{\}$, for each sampled point pair \mathbf{p}_1 and \mathbf{p}_2 in point cloud, we choose qualified $g^{(j)}$ by $\|\hat{t}^{(k)} - g^{(j)}\| < \xi$ into \mathbb{P} . Initialize $\gamma_{total}^{(k)} = 0$. For \mathbf{p}_1 and \mathbf{p}'_1 in \mathbb{P} , we obtain $\alpha^{(k)}, \beta^{(k)}, \gamma^{(k)}$ by PPF encoder (each $\gamma^{(k)}$ will be added into $\gamma_{total}^{(k)}$).

★ **Orientation voting scheme.** We denote the right orientation as D_1 and the up orientation as D_2 , these two angles are invariant to arbitrary rotations. Further, the orientation of $\tilde{o}_{(k)}$ can be only confirmed when two of three axes (x, y, z) are certified. As depicted in Fig. 4(b₁), the candidate orientation vector will lie on a rounded vertebra with one degree of freedom once α is confirmed, and then we generate candidate votes for two-point pairs $\mathbf{p}_1, \mathbf{p}'_1$ and $\mathbf{p}_1, \mathbf{p}''_1$, and cast them into counters, the final prediction is the one with the most votes, which is regarded as Rotation ($\mathbf{R}^{(k)}$).

★ **Scales.** Firstly, we define the parameters \bar{s} and $s \in \mathbb{R}^3$, where \bar{s} denotes the average bounding box scales, and s is used for the bounding box scale of a particular part. To this end, γ can be defined as $\gamma = \log(s) - \log(\bar{s})$. During inference, $\hat{s}^{(k)}$ is formulated as:

$$\hat{s}^{(k)} = \exp\left(\gamma_{total}^{(k)}\right) \otimes \hat{s}^{(k)} \tag{8}$$

4.5 Joint and Part Awareness

If we use the mean \hat{u}_i of all the points as the predicted joint direction, it will get a coarse result. Note that only the sampled point pairs from the same part are useful for its pose estimation. Ablation experimental results can be seen in Sect. 5.3. To obtain more robust predictions of joint and per-part pose estimation, we introduced a scoring mechanism to refine joint and part-aware prediction. Concretely, given an object with K rigid parts and $K - 1$ joints. Accordingly, given the K part segmentation $\{M_k \mid k = 1, \dots, K\}$, for joint info prediction, the GT joint score $\{C_i^{JT} \mid i = 1, \dots, N\}$ for each point \mathbf{p}_i is 1 if the target joint is connected to the part to which \mathbf{p}_i belongs (*i.e.*, \mathbf{p}_i comes from part M_* or M_\diamond , where M_* and M_\diamond are connected with target joint), otherwise it will be

set to 0 for current joint info prediction. Mathematically, GT generation of joint score can be formulated as Eq. 9:

$$C_i^{JT} = \begin{cases} 1, & \text{if } \mathbf{p}_1 \in M_* \text{ or } \mathbf{p}_1 \in M_\diamond, \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$C_i^{PT} = \begin{cases} 1, & \text{if } (\mathbf{p}_1, \mathbf{p}'_1) \in M_k, \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Subsequently, we only use point \mathbf{p}_i with predicted score $C_i^{JT} > 0.5$ and $H_i > 0.5$ for joint info prediction, the predicted joint direction \hat{u} can be the average prediction over all the satisfied point \mathbf{p}_i .

For pose estimation, we additionally define the GT part score $\{C_i^{PT} \mid i = 1, \dots, N\}$ for each point pair $\mathbf{p}_1, \mathbf{p}'_1$ is 1 if they come from the same part, otherwise it will be set to 0 for pose estimation of the current part. Mathematically, GT generation of part score can be formulated as Eq. 10. During the inference process, we deliberately filter out certain point pairs (noise point pairs from different parts). In other words, only the point pairs $\mathbf{p}_1, \mathbf{p}'_1$ with predicted part score $C_i^{PT} > 0.5$ will be involved in Eq. 4, which is used to conduct the per-pose estimation via voting strategy.

5 Experiments

In this section, we conduct extensive experiments to compare our method with other state-of-the-art algorithms followed by some relevant analysis about our method.

5.1 Experimental Setup

Implementation. To reduce network complexity (FLOPs) and facilitate rapid network convergence, 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 during the data pre-processing. The initial learning rate is 0.001 and decreases by 0.1 per 10 epochs. During training, the rotation and translation actions consist of 11 step sizes per axis in positive and negative directions, and the scale actions also contain 11 sizes for expansion and shrinking. These experiments are implemented on a computer workstation with a GeForce GTX 3090. All experiments are conducted in the PyTorch deep learning framework.

Dataset. Our method is tailored for both rigid and articulated objects and we have evaluated it both on synthetic and real-world datasets. Specifically, we first conduct experiments on **CAMERA25** for rigid objects and **ArtImage** for articulated objects. For generalization capacity, we use the **ReArtMix** and **RobotArm** datasets.

Metrics. These following metrics are used to evaluate the performance of our algorithm on category-level pose estimation. (1) **Per-part metrics.** We evaluate rotation error measured in degrees, translation error in normalized part coordinate space, and

Table 1. Quantitative comparisons of different methods on CAMERA25. Note that the best results are highlighted in **red** color.

Method	mAP									
	3D ₇₅		3D ₅₀		3D ₅₀		3D ₇₅		3D ₅₀	
	5°	10°	5°	10°			2 cm	5 cm	2 cm	5cm
NOCS [44]	22.6	29.5	34.5	54.5	83.9	69.5	32.3	40.9	48.2	64.6
SPD [40]	47.5	61.5	56.6	75.3	93.2	83.1	54.3	59.0	73.3	81.5
Gao <i>et al.</i> [12]	23.5	28.3	33.9	53.6	81.3	66.9	30.9	38.5	46.8	61.5
MH6D [26]	23.9	28.3	33.7	54.8	75.6	40.2	25.5	31.7	31.1	40.4
IST-Net [54]	55.6	64.1	66.9	77.7	92.2	85.9	64.3	70.9	78.3	85.1
U-COPE (Ours)	57.3	66.2	68.4	79.1	93.4	86.3	65.3	71.6	78.6	86.0

3D intersection over union (IoU) of the predicted amodal bounding box for each part. Besides, we further normalize the translation for each case, which helps to compare the translation errors among parts with different sizes. (2) **3D IoU**. 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 [28] for object detection files to evaluate AP.

5.2 Comparison with the SOTA Methods

In this section, we conduct extensive experiments on synthetic datasets to verify the effectiveness of our U-COPE, quantitative results and qualitative results are both provided for better comparison.

Rigid Objects. We use several prior arts for comparison in this part. Table 1 shows the quantitative results of the CAMERA25 test set. As it can be seen, our method can achieve an mAP of 57.3, 93.4, and 65.3 for $(3D_{75}, 5^\circ)$, $(3D_{50})$ and $(10^\circ, 5\text{ cm})$ respectively. It outperforms the baseline by **34.7**, **9.5**, and **33.0** (NOCS [44]), which also indicates that our algorithm has achieved a better result. Besides, qualitative comparison results on CAMERA25 can be seen in Fig. 5 (left). It can be shown that our prediction keeps more in step with GT (especially the scale of length, width, and height) compared to state-of-the-art.

Articulated Objects. We compare our results with the classical methods in this part on ArtImage. The quantitative results are shown in Table 2. Overall, we get the best 9D pose estimation result lies on the category laptop, with $4.8^\circ, 4.1^\circ$ for rotation degree error, and **0.029**, **0.030** for translation error. we conjecture that this is because our voting strategy can outperform objects with similar size, scale, and shape at per-part level due to the more robust params groups \mathbb{G}_k in Eq. 4. Moving to the 3D IoU metric, our prediction errors are significantly better at each part compared to the OMAD and AKB-Net. More importantly, compared with the classic articulated pose estimation method

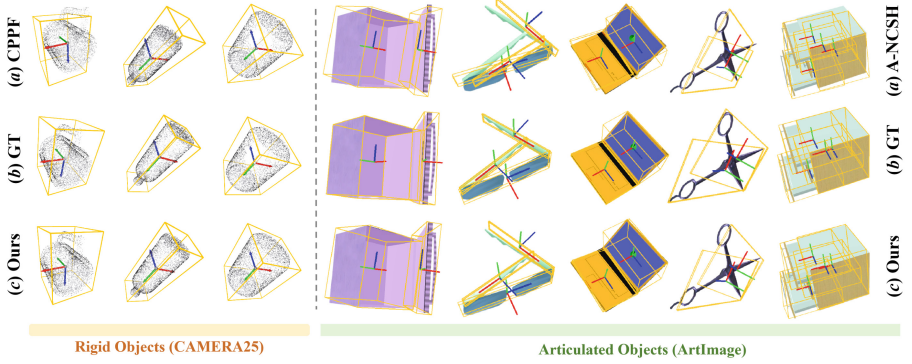


Fig. 5. Qualitative results on synthetic datasets. The baseline methods are CPPF [49] (for rigid objects) and A-NCSH [24] (for articulated objects). Please note that we show the single object and its pose visualization here rather than images for the best view. Here, CPPF [49] is used for rigid objects and A-NCSH [24] is used for articulated objects.

Table 2. Comparison with state-of-the-arts on ArtImage dataset. The categories laptop, eyeglasses, dishwasher and scissors contain only revolute joints, and the drawer category contains only prismatic joints. Note that the best results are highlighted in **red** color. The up or down arrows indicate higher or lower values corresponding to better results.

Category	Method	Per-part 9D Pose			Inference Time (s) ↓
		rotation error (°) ↓	translation error (m) ↓	3D IoU (%) ↑	
Laptop	A-NCSH [24]	5.3°, 5.4°	0.054, 0.043	56.7, 40.2	9.0
	OMAD [47]	5.4°, 4.3°	0.062, 0.061	43.5, 24.1	1.6
	AKBNet [27]	5.2°, 5.4°	0.063, 0.046	53.4, 36.8	7.4
	U-COPE (Ours)	4.8°, 4.1°	0.029, 0.030	74.6, 49.2	1.8
Eyeglasses	A-NCSH [24]	3.7°, 22.3°, 23.2°	0.049, 0.313, 0.324	52.5, 40.2, 39.6	11.9
	OMAD [47]	4.9°, 7.5°, 7.5°	0.062, 0.103, 0.104	22.8, 20.5, 21.4	2.5
	AKBNet [27]	4.3°, 23.6°, 24.2°	0.053, 0.331, 0.460	48.9, 37.8, 36.1	9.1
	U-COPE (Ours)	3.9°, 5.3°, 5.6°	0.043, 0.088, 0.088	65.4, 60.9, 61.4	2.1
Dishwasher	A-NCSH [24]	4.0°, 4.8°	0.059 , 0.123	84.3, 56.2	5.5
	OMAD [47]	6.0°, 6.2°	0.104, 0.142	66.5, 38.9	1.6
	AKBNet [27]	4.4°, 5.0°	0.075, 0.131	82.8, 54.6	4.3
	U-COPE (Ours)	3.8°, 4.5°	0.062, 0.066	87.7, 72.8	1.4
Scissors	A-NCSH [24]	2.0° , 2.9°	0.035, 0.025	46.5, 44.8	6.5
	OMAD [47]	3.9°, 3.4°	0.048, 0.039	35.6, 34.5	1.7
	AKBNet [27]	2.7°, 3.4°	0.047, 0.036	38.3, 37.1	5.2
	U-COPE (Ours)	2.4°, 2.5°	0.033, 0.023	46.9, 45.9	1.9
Drawer	A-NCSH [24]	2.8°, 3.5°, 3.9°, 2.9°	0.053, 0.155, 0.157, 0.075	90.2 , 81.5, 78.4, 82.7	16.5
	OMAD [47]	4.4°, 4.4°, 4.4°, 4.4°	0.111, 0.143, 0.144, 0.115	75.8, 73.4, 70.2, 71.3	1.9
	AKBNet [27]	3.3°, 3.8°, 4.2°, 3.7°	0.057, 0.177, 0.183, 0.096	85.9, 78.6, 77.6, 79.0	14.5
	U-COPE (Ours)	2.7°, 3.2°, 3.4°, 2.9°	0.042, 0.101, 0.122 , 0.094	86.1, 81.8, 79.1 , 80.3	1.7

A-NCSH, our method also beats it with **12.9°, 20.7°, 21.8°** regarding eyeglasses. Meanwhile, our method enjoys the competitive metric of inference time (Almost equiv-

Table 3. Quantitative results about ablation study. Please note that results are reported on eyeglasses from ArtImage.

Index	Number of Point Pairs	Per-part 9D Pose			Inference Time (s) ↓
		rotation error (°) ↓	translation error (m) ↓	3D IoU (%) ↑	
I	5,000	4.9°, 7.4°, 7.6°	0.062, 0.103, 0.112	59.3, 54.2, 55.5	1.8
II	10,000	4.3°, 5.8°, 6.4°	0.051, 0.097, 0.099	63.4, 57.8, 58.6	2.0
III	20,000	3.9°, 5.3°, 5.6°	0.043, 0.088, 0.088	65.4, 60.9, 61.4	2.1
IV	40,000	3.7°, 5.0°, 5.2°	0.039, 0.084, 0.086	66.7, 62.4, 62.9	2.6
	Awareness	Per-part 9D Pose			Inference Time (s) ↓
V	-	4.2°, 5.8°, 6.7°	0.058, 0.092, 0.105	62.9, 58.8, 56.1	2.0
VI	✓	3.9°, 5.3°, 5.6°	0.043, 0.088, 0.088	65.4, 60.9, 61.4	2.1

alent performance to OMAD), which can be explained by our end-to-end optimization strategy in contrast to the two-stage method in A-NCSH. Qualitative comparison results on ArtImage can be seen in Fig. 5 (right). As we can see, morphological differences between the poses of different objects are obvious, and some of them suffer from occlusion problems. Our method can more accurately predict the $R^{(k)}$, $\mathbf{t}^{(k)}$, $\mathbf{s}^{(k)}$ of each part of the object. The quality improvement achieved by U-COPE is attributable to the effective utilization of the universal voting strategy and joint optimization.

5.3 Ablation Study

We conduct ablation studies in this section. Quantitative results are reported in Table 3.

Numbers of Point Pairs. To demonstrate the trade-off between the numbers of point pairs and inference time, we compare this ablation study under different numbers of point pairs ranging from 5,000 to 20,000 in Table 3 (I–IV). It is not hard to recognize that the voting results for orientation and translation become more accurate and tend to saturate as the number of pair samples increases from the results. Yet, the inference time also grows fast as the number of pair samples increases. To this end, we set the number of point pairs to be 20,000 in practice, which can yield satisfying performance with a good trade-off between performance and cost.

Joint and Part Awareness. With the aid of joint and part awareness mechanism, we go a further step to filter the candidate point for joint info prediction and the sampled point pairs for per-part pose estimation (*i.e.*, the candidate point for joint info should come from the adjacent part connected to the target joint, while point pairs for per-part pose estimation should be from the target part meantime). Table 3 (V–VI) shows the ablation experiment results. It can be concluded that the proposed joint and part awareness mechanism helps to achieve better performance. We conjecture this can be attributed to the elimination of noise point pairs.

5.4 Generalization Capacity

In order to fully realize the application value of our method in practice, we go a further step to conduct generalizability validation experiments in real-world scenarios. Since

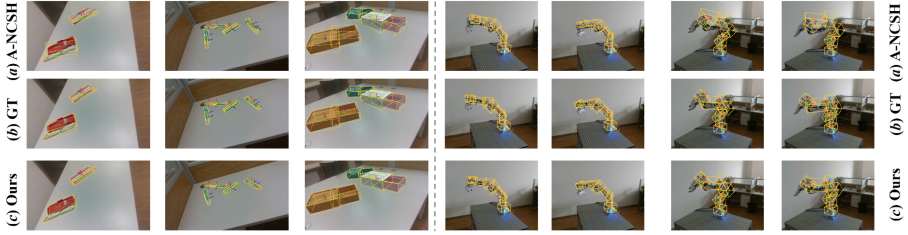


Fig. 6. Qualitative results on ReArtMix (Left), and Qualitative results on RobotArm (Right). Please zoom in for better visualization.

the articulated objects are more challenging due to the constraints of the kinematic structure, we conduct generalization experiments on ReArtMix and RobotArm datasets.

Qualitative results on ReArtMix can be seen in Fig. 6 (Left). Note that we use the right-handed coordinate system to locate the coordinates, with the green axis indicating the y-direction, the red axis indicating the x-direction, and the blue axis indicating the z-direction. As we can see, our method can more accurately predict the $R^{(k)}$, $\mathbf{t}^{(k)}$, $\mathbf{s}^{(k)}$ of each part of the object, which is closer to the results of Ground Truth. To be more concrete, our methods can do better on objects that have prismatic joints (like the Box), we think that our U-COPE can converge to the desired state due to the flatness of its features state space. Besides, we offer the qualitative results on RobotArm in Fig. 6 (Right), the results also show a good generalization.

6 Conclusion and Limitations

This paper presents a universal approach for 9D category-level pose estimation named U-COPE for both rigid objects and articulated objects. Concretely, our method uses a heatmap and offset strategy to predict the location and orientation of the joints and generate mask vectors to predict the part segmentation. By introducing a voting strategy, we output a series of parameter groups that are important in the estimation of object poses, which avoids the previous unstable convergence of the direct estimation of poses. Our U-COPE can output three kinds of targets end-to-end, *i.e.*, (1) Joint information (consisting of a pivot associated with its joint direction). (2) Part segmentation (K individual parts). (3) 9D pose for K parts calculated by universal voting strategy. To the best of our knowledge, this is the first attempt to address both rigid object and articulated object pose estimation tasks within the same framework. Experiments demonstrate that our approach is able to obtain state-of-the-art performance on both rigid and articulated object observations.

Since our idea of segmentation is encouraged by Mask R-CNN [17], the point cloud of each part will be incomplete or noisy if the segmentation result is not as good as expected. More accurate segmentation algorithms from upstream help mitigate it. In the future, we will explore more efficient feature representations, develop methods for dynamic object pose estimation, and integrate multi-modal sensor information to improve accuracy and robustness for real-world applications.

Acknowledgements. This work was supported in part by National Natural Science Foundation of China under Grant 62302143 and Anhui Provincial Natural Science Foundation under Grant 2308085QF207.

References

1. Alartartsev, S., Stellmacher, S., Ortmeier, F.: Robotic task sequencing problem: a survey. *J. Intell. Robot. Syst.* **80**, 279–298 (2015)
2. Amin, D., Govilkar, S.: Comparative study of augmented reality SDKs. *Int. J. Comput. Sci. Appl.* **5**(1), 11–26 (2015)
3. Avetisyan, A., Dai, A., Nießner, M.: End-to-end cad model retrieval and 9dof alignment in 3D scans. In: *Proceedings of the IEEE/CVF International Conference on computer vision*, pp. 2551–2560 (2019)
4. Azuma, R.T.: A survey of augmented reality. *Presence Teleoper. Virtual Environ.* **6**(4), 355–385 (1997)
5. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., Grundmann, M.: Blazepose: on-device real-time body pose tracking. *arXiv preprint [arXiv:2006.10204](https://arxiv.org/abs/2006.10204)* (2020)
6. Billard, A., Kragic, D.: Trends and challenges in robot manipulation. *Science* **364**(6446), eaat8414 (2019)
7. Burdea, G.C., Coiffet, P.: *Virtual Reality Technology*. Wiley, Hoboken (2003)
8. Carmigniani, J., Furht, B.: *Augmented Reality: An Overview*. *Handbook of Augmented Reality*, pp. 3–46 (2011)
9. Chen, H., Wang, P., Wang, F., Tian, W., Xiong, L., Li, H.: Epro-pnp: generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2781–2790 (2022)
10. Di, Y., et al.: Gpv-pose: category-level object pose estimation via geometry-guided point-wise voting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6781–6791 (2022)
11. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: efficient and robust 3d object recognition. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 998–1005. IEEE (2010)
12. Gao, G., Lauri, M., Wang, Y., Hu, X., Zhang, J., Frintrop, S.: 6d object pose regression via supervised learning on point clouds. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3643–3649. IEEE (2020)
13. Gilitschenski, I., Sahoo, R., Swarting, W., Amini, A., Karaman, S., Rus, D.: Deep orientation uncertainty learning based on a bingham loss. In: *International Conference on Learning Representations* (2019)
14. Grest, D., Woetzel, J., Koch, R.: Nonlinear body pose estimation from depth images. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) *DAGM 2005*. LNCS, vol. 3663, pp. 285–292. Springer, Heidelberg (2005). https://doi.org/10.1007/11550518_36
15. Guan, R., et al.: Contrastive multiview subspace clustering of hyperspectral images based on graph convolutional networks. *IEEE Trans. Geosci. Remote Sens.* **62**, 1–14 (2024)
16. Guo, D., Li, K., Hu, B., Zhang, Y., Wang, M.: Benchmarking micro-action recognition: dataset, methods, and applications. *IEEE Trans. Circ. Syst. Video Technol.* **34**(7), 6238–6252 (2024). <https://doi.org/10.1109/TCSVT.2024.3358415>
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969 (2017)

18. Heppert, N., et al.: Carto: category and joint agnostic reconstruction of articulated objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21201–21210 (2023)
19. Hodaň, T., Matas, J., Obdržálek, Š: On evaluation of 6D object pose estimation. In: Hua, G., Jégou, H. (eds.) ECCV 2016 Part III. LNCS, vol. 9915, pp. 606–619. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_52
20. Hou, J., Graham, B., Nießner, M., Xie, S.: Exploring data-efficient 3D scene understanding with contrastive scene contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15587–15597 (2021)
21. Hu, Y., Hugonot, J., Fua, P., Salzmann, M.: Segmentation-driven 6D object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3385–3394 (2019)
22. Jaritz, M., Gu, J., Su, H.: Multi-view pointnet for 3D scene understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
23. Khamis, A., Hussein, A., Elmogy, A.: Multi-robot task allocation: a review of the state-of-the-art. *Coop. Robots Sensor Netw.* **2015**, 31–51 (2015)
24. Li, X., Wang, H., Yi, L., Guibas, L.J., Abbott, A.L., Song, S.: Category-level articulated object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3706–3715 (2020)
25. Li, X., Weng, Y., Yi, L., Guibas, L.J., Abbott, A., Song, S., Wang, H.: Leveraging se (3) equivariance for self-supervised category-level object pose estimation from point clouds. In: *Advance in Neural Information Processing System*, vol. 34, pp. 15370–15381 (2021)
26. Liu, J., Sun, W., Liu, C., Yang, H., Zhang, X., Mian, A.: Mh6d: multi-hypothesis consistency learning for category-level 6-d object pose estimation. *IEEE Trans. Neural Netw. Learn. Syst.* (2024)
27. Liu, L., et al.: Akb-48: a real-world articulated object knowledge base. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14809–14818 (2022)
28. Liu, L., Xue, H., Xu, W., Fu, H., Lu, C.: Toward real-world category-level articulation pose estimation. *IEEE Trans. Image Process.* **31**, 1072–1083 (2022)
29. Lourakis, M., Zabulis, X.: Model-based pose estimation for rigid objects. In: Chen, M., Leibe, B., Neumann, B. (eds.) *ICVS 2013*. LNCS, vol. 7963, pp. 83–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39402-7_9
30. Mason, M.T.: *Mechanics of Robotic Manipulation*. MIT press, Cambridge (2001)
31. Mason, M.T.: Toward robotic manipulation. *Ann. Rev. Control, Robot. Auton. Syst.* **1**, 1–28 (2018)
32. Minguez, J., Montesano, L., Lamiroux, F.: Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Trans. Rob.* **22**(5), 1047–1054 (2006)
33. Mo, K., Guibas, L.J., Mukadam, M., Gupta, A., Tulsiani, S.: Where2act: from pixels to actions for articulated 3d objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6813–6823 (2021)
34. Pauwels, K., Rubio, L., Diaz, J., Ros, E.: Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2347–2354 (2013)
35. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
36. Shao, J., Loy, C.C., Kang, K., Wang, X.: Crowded scene understanding by deeply learned volumetric slices. *IEEE Trans. Circ. Syst. Video Technol.* **27**(3), 613–623 (2016)
37. Shridhar, M., Manuelli, L., Fox, D.: Cliport: what and where pathways for robotic manipulation. In: *Conference on Robot Learning*, pp. 894–906. PMLR (2022)

38. Spezialetti, R., Stella, F., Marcon, M., Silva, L., Salti, S., Di Stefano, L.: Learning to orient surfaces by self-supervised spherical CNNs. In: *Advance in Neural Information Processing System*, vol. 33, pp. 5381–5392 (2020)
39. Tang, F., Wu, Y., Hou, X., Ling, H.: 3d mapping and 6d pose computation for real time augmented reality on cylindrical objects. *IEEE Trans. Circ. Syst. Video Technol.* **30**(9), 2887–2899 (2019)
40. Tian, M., Ang, M.H., Lee, G.H.: Shape prior deformation for categorical 6D object pose and size estimation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020 Part XXI*. LNCS, vol. 12366, pp. 530–546. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_32
41. Tu, W., et al.: Attribute-missing graph clustering network. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 15392–15401 (2024)
42. Wagner, D., Schmalstieg, D.: Artoolkitplus for pose tracking on mobile devices (2007)
43. Wang, H., Davoine, F., Lepetit, V., Chaillou, C., Pan, C.: 3-D head tracking via invariant keypoint learning. *IEEE Trans. Circ. Syst. Video Technol.* **22**(8), 1113–1126 (2012)
44. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: 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*, pp. 2642–2651 (2019)
45. Wang, Y.J., Luo, Y.M., Bai, G.H., Guo, J.M.: UformPose: a u-shaped hierarchical multi-scale keypoint-aware framework for human pose estimation. *IEEE Trans. Circ. Syst. Video Technol.* **33**(4), 1697–1709 (2022)
46. Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C.: Pose flow: efficient online pose tracking. arXiv preprint [arXiv:1802.00977](https://arxiv.org/abs/1802.00977) (2018)
47. Xue, H., Liu, L., Xu, W., Fu, H., Lu, C.: Omad: object model with articulated deformations for pose estimation and retrieval. arXiv preprint [arXiv:2112.07334](https://arxiv.org/abs/2112.07334) (2021)
48. You, S., Yao, H., Xu, C.: Multi-target multi-camera tracking with optical-based pose association. *IEEE Trans. Circ. Syst. Video Technol.* **31**(8), 3105–3117 (2020)
49. You, Y., Shi, R., Wang, W., Lu, C.: CPPF: towards robust category-level 9d pose estimation in the wild. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6866–6875 (2022)
50. Yu, T., Lin, X., Wang, S., Sheng, W., Huang, Q., Yu, J.: A comprehensive survey of 3D dense captioning: Localizing and describing objects in 3D scenes. *IEEE Trans. Circ. Syst. Video Technol.* (2023)
51. Zhang, L., Du, J., Dong, S., Wang, F., Xie, C., Wang, R.: AM-ResNet: low-energy-consumption addition-multiplication hybrid ResNet for pest recognition. *Comput. Electron. Agric.* **202**, 107357 (2022)
52. Zhang, L., Xu, M., Li, D., Du, J., Wang, R.: Catmullrom splines-based regression for image forgery localization. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 7196–7204 (2024)
53. Zheng, J., Chan, K., Gibson, I.: Virtual reality. *IEEE Potentials* **17**(2), 20–23 (1998)
54. Zou, L., Huang, Z., Gu, N., Wang, G.: Learning geometric consistency and discrepancy for category-level 6D object pose estimation from point clouds. *Pattern Recogn.* **145**, 109896 (2024)