R^2-Art: Category-Level Articulation Pose Estimation from Single RGB Image via Cascade Render Strategy

Li Zhang[1, 2, 3], **Haonan Jiang**[4], **Yukang Huo**[5], **Yan Zhong**[6], **Jianan Wang**[3], **Xue Wang**[1], **Rujing Wang**[1 †], **Liu Liu**[7 †*]

[1]Hefei Institute of Physical Science, Chinese Academy of Sciences, China
[2]University of Science and Technology of China, Hefei, China
[3]Astribot, Shenzhen, China
[4]Zhejiang University of Technology, Zhejiang, China
[5]China Agricultural University. Beijing, China
[6]School of Mathematical Sciences, Peking University. Beijing, China
[7]Hefei University of Technology, Hefei, China
zanly20mail.ustc.edu.cn, 211122030127@zjut.edu.cn

Abstract

Human life is filled with articulated objects. Previous works for estimating the pose of category-level articulated objects rely on costly 3D point clouds or RGB-D images. In this paper, our goal is to estimate category-level articulation poses from a single RGB image, where we propose **R^2-Art**, a novel category-level **Art**iculation pose estimation framework from a single **R**GB image and a cascade **R**ender strategy. Given an RGB image as input, R^2-Art estimates per-part 6D pose for the articulation. Specifically, we design parallel regression branches tailored to generate camera-to-root translation and rotation. Using the predicted joint states, we perform PC prior transformation and deformation with a joint-centric modeling approach. For further refinement, a cascade render strategy is proposed for projecting the 3D deformed prior onto the 2D mask. Extensive experiments are provided to validate our R^2-Art on various datasets ranging from synthetic datasets to real-world scenarios, demonstrating the superior performance and robustness of the R^2-Art. We believe that this work has the potential to be applied in many fields including robotics, embodied intelligence, and augmented reality.

1 Introduction

Articulated objects are ubiquitous in our daily lives, spanning from small table-scale objects (*e.g.*, eyeglasses) to large-size objects (*e.g.*, dishwashers). Unlike rigid objects characterized by fixed and unchanging shapes, articulated objects consist of a limited number of interconnected rigid components (linked by various joints), which can easily conduct relative motion between the rigid parts. Kinematic constraints also make articulated object pose estimation more challenging. Accurate estimation of 6D pose for articulated objects is crucial in many downstream multimedia and machine vision tasks, such as embodied AI (Yu et al. 2023; Karrer et al. 2011), robot manipulation (Xiong et al. 2023), human-object interactions (Liu et al. 2021), and augmented reality (Amin and Govilkar 2015).

Generally, compared to instance-level 6D pose estimation, depth-based category-level 6D pose estimation requires machines to understand the 3D rotation and 3D translation of unseen objects, with the given partial observations. However, these solutions face the following challenges:

(i) Inaccurate and expensive depth information. Depth sensors are prone to introduce noise and inaccuracies, particularly with distant or low-reflectivity surfaces.

(ii) Self-occlusion. Mutual and self-occlusion can cause indistinguishable object views, leading to ambiguous refinement targets. This issue is especially common in articulated objects with multiple movable parts.

(iii) Implicit learning manner. Previous works often use implicit object representation methods, such as normalization (Wang et al. 2019) or key-point modeling (Xue et al. 2021), which involve time-consuming post-processing. These approaches overlook explicit pose representation within object categories, limiting performance in 6D pose estimation.

To deal with the first challenge, we cast the category-level object pose estimation task as a pose regression task from a single RGB image. Concretely, we use a color-sensitive backbone (DINO V2) to extract the pose features and then use a size-sensitive backbone (HRNet) to regress the depth in parallel. Considering the correlation between object pose and mask, we propose a cascade render strategy for further pose refinement.

To address the second challenge, the basic mechanism behind our method is to employ a joint-centric pose modeling mechanism. This approach estimates part poses by predicting per-joint states, such as revolute joint angles and translational joint distances. This is because joint states can still be perceived in occlusion scenarios, whereas part-centric meth-

*Li Zhang and Haonan Jiang contributed equally.
This work was done when Li Zhang was an intern at Astribot.
Corresponding authors†: Rujing Wang and Liu Liu.

ods (Zhang et al. 2024a, 2025) overlook this aspect.

To cope with the third challenge, our core idea involves predicting the disentanglement of pose (i.e., 3D rotation, 3D translation, and 1D joint state) explicitly. Concretely, our method is end-to-end and requires no post-processing during the inference phase. This is achieved by designing three parallel branches to independently regress rotation, 2D translation, depth, and joint state.

In summary, our method trains a neural network with CAD models in intra-category to derive shape, which guides pose estimation for unseen objects. Unlike part-centric methods, our approach adopts a joint-centric modeling strategy, eliminating the need to segment objects into rigid parts. Additionally, a cascade render strategy is employed for progressive refinement. Our method is simple yet effective in addressing the RGB-based pose estimation task in an end-to-end manner. Extensive experiments validate the performance of our R^2-Art across a range of datasets, from synthetic, semi-synthetic, and real-world scenarios, demonstrating its superior performance and robustness.

In total, our main contributions are threefold:

- We proposed R^2-Art, an end-to-end framework for estimating category-level articulated object 6D pose efficiently and robustly under a single 2D RGB image input.
- In our R^2-Art, we perform decoupled pose learning and then conduct prior PC transformation and deformation via a joint-centric modeling method. Moreover, a customized cascade render strategy is employed to conduct stage-wise optimation.
- The efficiency and robustness of R^2-Art are demonstrated through the evaluation on both synthetic datasets to real-world scenarios for the articulation task, e.g., ArtImage, ReArtMix, and RobotArm.

2 Related Work

Category-level 6D Pose Estimation. Recent advances in category-level 6D pose estimation focus on detecting an unseen object's 3D rotation and translation. NOCS (Wang et al. 2019) pioneered efficient pose estimation for unseen objects within the same category, inspiring subsequent improvements (Mo et al. 2021; Zhang et al. 2024b; Avetisyan, Dai, and Nießner 2019) for complex scenarios. While most methods target rigid objects, attention has shifted toward articulated object 6D pose (Liu et al. 2022c; Weng et al. 2021a). A-NCSH (Li et al. 2020) extended NOCS for articulated objects, introducing per-part normalized coordinates for pose optimization. However, despite these achievements, these methods still have drawbacks, such as ignoring the constraints of the kinematic structure and inaccurate pose estimation performance.

RGB-based Pose Estimation. RGB-based pose estimation has gained prominence in computer vision and robotics due to the high cost of acquiring depth information (Moon, Chang, and Lee 2019). Deep learning has significantly improved rigid object pose estimation, with prior works falling into three categories: (1) holistic methods (Gu and Ren 2010; Hu et al. 2024), which directly estimate poses without post-processing; (2) keypoint-based methods (Ban et al. 2024; Shen et al. 2024), which use 2D-3D correspondences and PnP/RANSAC algorithms; and (3) dense correspondence methods (Peng et al. 2019; Guan et al. 2024), which predict dense pixel-wise correspondences for pose estimation. However, due to the loss of geometric information, these methods are sensitive to appearance textures, making them less effective than depth-based approaches. Furthermore, as these methods focus on instance-level object pose estimation, they suffer from poor generalization and limited practicality.

3 Notation and Problem Statement

To achieve a robust algorithm for the Category-level RGB-based Articulation Pose Estimation task (**C-RAPE**), our key idea is utilizing point cloud prior integrated with decoupled pose representation to conduct cascade render procedure. Here, we formulate a new paradigm for the C-RAPE task with a novel cascade framework named R^2-Art. Specifically speaking, given a single 2D observed object image I (captured from an articulated object $A = \{\delta_k\}_{k=1}^{K}$) as **input** ($\{\delta_k\}$ is the k-th rigid part), our R^2-Art conducts the predictions for (i) per-part 3D rotation $R^{(k)} \in SO(3)$ (ii) per-part 3D translation $\mathbf{t}^{(k)}$. In total, the rotation and translation together constitute the final pose estimation result $T = \{R^{(k)}, \mathbf{t}^{(k)}\}_{k=1}^{K} \in SE(3)$ (**output**).

Since we can't access the one-to-one CAD model, we exploit a shape prior $\mathcal{P}$ from the intra-category of the input image that shares high appearance and kinematic structure similarity. Therefore, with the given RGB image I, the pipeline of our R^2-Art can be illustrated as follows: (i) A self-supervised model is first applied to extract off-line features from I, we first predict the 6D pose $R_{\textbf{root}}, \mathbf{t}_{\textbf{root}}$ in parallel branches and the transform the root part. (ii) Based on the joint-centric modeling strategy, we first calculate the pose of the node parts by applying the Rodrigues formula to the predicted joint state $\Theta = \{\theta^{(k)}\}_{k=1}^{K-1}$. The node parts are then transformed and deformed accordingly. This operation allows the prior to undergo a pose transformation from canonical space to camera space. (iii) To refine the articulation pose further, we propose a cascade mask render strategy that clarifies progressively advancing learning objectives at each stage. To avoid ambiguity and better clarify the variables, we use the symbol $*$ to indicate the GT variables, and $\hat{}$ to indicate the predicted variables.

4 Methodology

To conduct the Category-level RGB-based Articulation Pose Estimation (C-RAPE) task, we propose R^2-Art to solve this task. The overview pipeline is shown in Fig. 1. Concretely, taking a shape prior $\mathcal{P}$ (represented as a point cloud) and an observed RGB image I as input, our R^2-Art output 6D pose by three parallel branches (*i.e.*, rot-trans branch in blue, depth branch in purple, and joint state branch in green). Firstly, we re-modulate the features extracted by DINO V2 (Oquab et al. 2023) to predict the decoupled pose of root part, and output correction factor λ to the root depth (As illustrated in Sec. 4.1). Secondly, based on the proposed joint-centric modeling method, we get the deformed PC via

the prior (detailed in Sec. 4.2). Finally, to ensure a more refined pose prediction, we introduce the cascade render strategy with progressive constraint in Sec. 4.3. To achieve the SOTA result, we use a three-layer UNet-like network for rottrans branch, HRNet (scale-sensitive) for depth branch, and Resnet (scale-invariant) for joint state branch in our SOTA model.

4.1 Decoupled Pose Representation and Depth Regression

In this work, we use a joint-centric modeling method (it will be introduced in detail in Sec. 4.2). Here, we only focus on the root part pose (i.e., $T_{\textbf{root}} = (R_{\textbf{root}}, \mathbf{t}_{\textbf{root}})$. Intuitively, 3D translation can be represented as $\mathbf{t}_{\textbf{root}} = (x_{\textbf{root}}, y_{\textbf{root}}, z_{\textbf{root}})$, where $\phi = (x_{\textbf{root}}, y_{\textbf{root}})$ represents the 2D translation parameters, and $z_{\textbf{root}}$ denotes the camera-to-root depth. Thus, translation estimation can be divided into two independent tasks: 2D translation prediction and 1D depth estimation.

Firstly, for rotation estimation, we adopt a disentanglement scheme similar to GPV-Pose (Di et al. 2022), decomposing $R_{\textbf{root}}$ into three orthogonal vectors: α, β, and γ. Using features extracted by DINO V2 (Oquab et al. 2023), we predict two direction vectors (e.g., upward α and rightward β) for different objects, enabling more efficient learning of rotations with fewer parameters.

Secondly, for 2D translation prediction, we first remodulate the off-line features by a UNet-liked network, and then two MLPs are applied to predict the $\phi \in \mathbb{R}^2$. In this way, we could decompose articulation rotation and 2D translation estimation into two simultaneous regression-based tasks, which prevents them from influencing each other.

Thirdly, depth estimation from a single view suffers from inherent ambiguity. This mainly involves two challenges: i) Firstly, the input image does not provide any *explicit* information on the relative position of the camera and objects. ii) Directly estimating the absolute depth via the feature representations learned from the whole image is non-trivial, since it is focused on the local information without perceiving global features related to the scale of objects.

To deal with the challenge i, we introduce a new distance measure named *projection depth* d_p, which is defined as follows:

$$d_p = \sqrt{\frac{f_x \cdot f_y \cdot S_{wld}}{S_{bbox}}} \tag{1}$$

where f_x, f_y, S_{wld}, and S_{bbox} are focal lengths divided by the per-pixel distance factors (pixel) of x- and y-axes, the area of the bounding box of articulations in physics world (mm^2), and bounding box from image space ($pixel^2$), respectively. d_p could approximate the absolute depth from the camera to the object using the ratio of the actual area and the imaged area of it, given camera parameters.

To address challenge ii, We leverage the correlation between object scale and its background to improve the depth prediction performance. By concatenating RGB features from DINO V2 with depth information, our method preserves relative articulation information and enhances training stability.

In general, our final scheme is to regress correction depth λ within the enriched features from RGB and depth information. The image feature can give a clue to the depth branch about how much the λ has to be changed: The correction factor λ can adaptively change its value to better fit the object size. Because gradient can explicitly tell the network whether to amplify or reduce the predicted λ. Finally, The estimated λ is multiplied by the d_p, which becomes the final depth value (i.e., $z_{\textbf{root}} = \lambda \cdot d_p$).

4.2 Joint-Centric Representation of Articulation

A straightforward and simple modeling method for articulated objects is ***part-centric***, i.e., consider articulated objects as the concatenation and combination of multiple rigid objects. this idea is widely used in previous works (Li et al. 2020; Weng et al. 2021b), which overlooks the importance of kinematic structure and encounters challenges related to self-occlusion. In contrast, we adopt a ***joint-centric*** viewpoint to explore articulated objects in this work, treating the pose estimation task as a joint state prediction problem.

Concretely, we categorize rigid parts in an articulated object into two groups: the root part (physical root of articulations) is defined in the 3D world and can move freely without constraint, and the node part, whose motion is constrained by the joint type in the kinematic structure. Following A-NCSH (Li et al. 2020), we consider two joint types: 1) Revolute joints, enabling rotational motion, with joint state $\boldsymbol{\theta}_r$ as the relative rotation angle and parameters $\boldsymbol{\varphi}_r = (\boldsymbol{u}_r, \boldsymbol{q}_r)$, where $\boldsymbol{u}_r$ is the joint axis direction and $\boldsymbol{q}_r$ the pivot point position; 2) Prismatic joints, allowing linear motion, with state $\boldsymbol{\theta}_p$ as the relative distance and parameter $\boldsymbol{u}_p$ as the axis direction. The extended Rodrigues formula is used to convert joint states into a matrix $R^{'(k)}_{\textbf{node}}$ for the k-th part.

$$\begin{aligned} R^{'(k)}_{\textbf{node}} = &\cos\boldsymbol{\theta}_r^{(k)} U_r^{(k)} + (1 - \cos\boldsymbol{\theta}_r^{(k)}) \cdot \\ &(U_r^{(k)} \cdot Q_r^{(k)}) Q_r^{(k)} + \sin\boldsymbol{\theta}_r^{(k)} (Q_r^{(k)})^{\wedge} \end{aligned} \tag{2}$$

where $U_r^{(k)}$ denotes the normalized direction of the joint axis $\boldsymbol{u}_r^{(k)}$, $Q_r^{(k)}$ denotes a anti-symmetric matrix composed of pivot point position $\boldsymbol{q}_r^{(k)}$, symbol $^{\wedge}$ means anti-symmetric. Here the $R^{'(k)}_{\textbf{node}}$ is a matrix of size 3×4 and we stack it with the row [0, 0, 0, 1] appended at the end to get homogeneous matrix $T^{(k)}_{\textbf{node}}$ as k-th node part's pose.

For prismatic joint, given joint parameters ($\boldsymbol{u}_p^{(k)}$) and predicted joint state $\boldsymbol{\theta}_p^{(k)}$, we can also get homogeneous matrix as pose for k-th node part:

$$T^{(k)}_{\textbf{node}} = \begin{bmatrix} \mathbf{I} & \boldsymbol{\theta}_p^{(k)} \boldsymbol{u}_p^{(k)} \\ 0 & 1 \end{bmatrix} \tag{3}$$

where $\mathbf{I}$ indicates the identity matrix. Finally, we can calculate the poses of node parts by iteratively multiplying the poses from their root part. Within the joint-centric pose modeling method, we can correspond part to joint one by one. The total articulated object poses for all the K parts can be represented by a sequence of node part pose $\{T^{(k)}_{\textbf{node}}\}_{k=1}^{K-1}$ and root part pose $T_{\textbf{root}} = \{R_{\textbf{root}}, \boldsymbol{t}_{\textbf{root}}\}$.

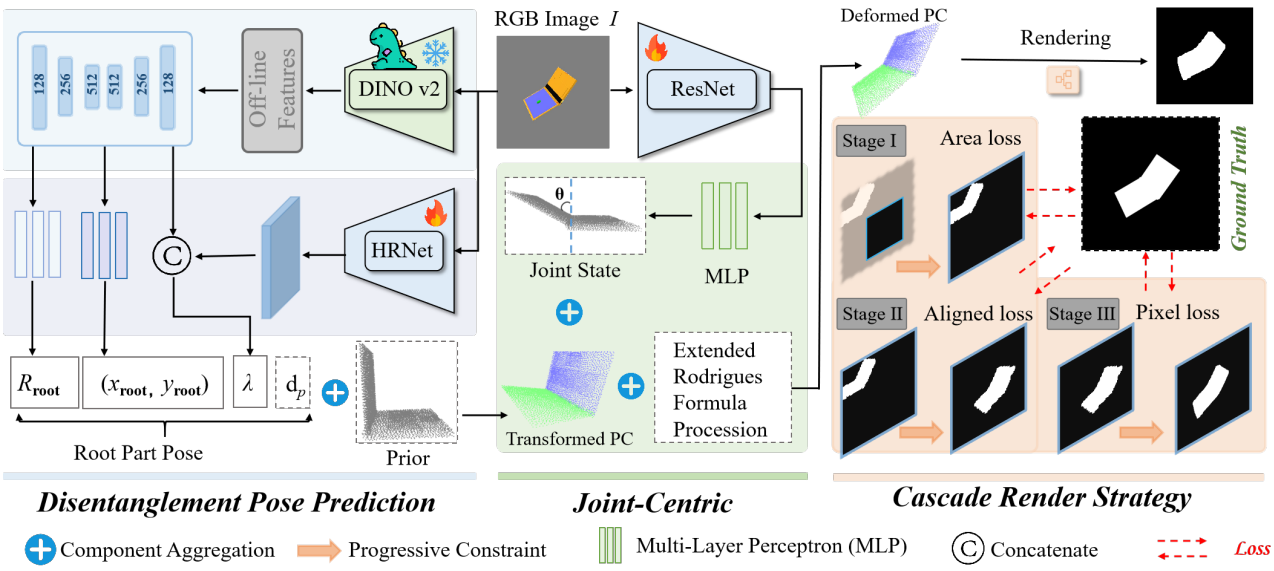


Figure 1: The pipeline of our R^2-Art. It consists of the following components: (a) Disentanglement Pose Prediction. The rot-trans branch is in blue, while depth branch is in purple. (Sec. 4.1); (b) Joint-Centric. Given the joint state, we conduct PC transformation and deformation via the Rodrigues Formula. (Sec. 4.2); (c) Cascade Render Strategy is proposed for further refinement with 3-stage progressive optimization. (Sec. 4.3), the blue solid line edge represents the imaging plane.

4.3 Cascade Render Strategy

In this work, we introduce a coarse-to-fine cascade render strategy for better optimization of our R^2-Art:

Stage I: *Coarse Level* Area-Consistency Constrain. Based on the pinhole camera model and 3D camera-to-object translation $\mathbf{t} = (x, y, z)$, the x component determines the horizontal position, y determines the vertical position in the imaging plane, and z affects the imaging size of objects. In the first stage, our primary goal is to ensure that the deformed PC can be successfully projected onto the imaging plane.

To this end, we propose an area consistency loss for allocating adaptive supervision intensities according to the rendering result. Concretely speaking, a penalty term η is introduced to L_1 loss function. η is tunable, *i.e.*, the greater rendering difference, the greater its value is. In practice, considering the correlation between the area of enclosing bounding box from rendering object mask and the available pixel number (pixel with RGB value of 255 in bbox) in rendering mask. In view of this fact, we define $\eta = \exp\{\sum p^* - \sum \hat{p}\}$, where p^* represents for the GT available pixel number, while $\hat{p}$ is the prediction.

Obviously, when the object fails to be rendered onto the imaging plane (due to the poor pose prediction), η becomes large due to the significant pixel count error. However, when the object is rendered onto the imaging plane successfully (regardless of whether it is at the correct position), the available pixel count matches that of GT, causing η to reduce to 1. Therefore, we can conclude that η is sensitive to whether the object is successfully projected. With the definition of η, the loss function of $\mathcal{L}_{area}$ is given by:

$$\mathcal{L}_{area} = \eta \cdot \| \log(S^* + 1) - \log(\hat{S} + 1) \|_1 \tag{4}$$

where $\hat{S}$ is the area of bounding box regarding predicted object rendering mask, while S^* is its GT.

Stage II: *Middle Level* Rotation and Translation Alignment. To conduct a better rotation and translation alignment, we argue that the following important geometric factors should be considered: overlapping area, center point distance, and aspect ratio. Draw inspirations from corner-based object detection works (Yang et al. 2019; Zheng et al. 2022), we focus on the corners of a bounding box. Concretely speaking, a bounding box representation can be defined by its box parameters $\Phi = (p_1, p_2) = \big((u_1, v_1), (u_2, v_2)\big) \in \mathbb{R}^4$, where p_1, p_2 are top left and bottom right corner, respectively. Our loss function includes three parts: IOU loss, overlap loss, and offset loss.

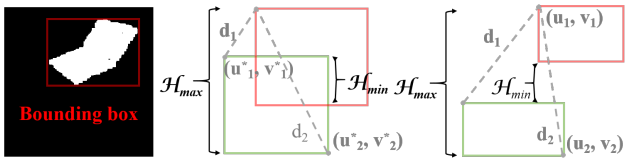
Firstly, for an improved IOU loss, we use a log-likelihood loss:

$$\mathcal{L}_{\text{IOU}} = -\log(\text{IOU}) \tag{5}$$

Secondly, we propose the bounding box overlap (BBO) loss to measure bounding box similarity by maximizing the overlap of width and height. It is a stricter constraint and puts more gradient for low overlapping bounding box. As shown in Fig. 2 (b), given the predicted box $(\hat{u}_1, \hat{v}_1, \hat{u}_2, \hat{v}_2)$ and ground truth box $(u_1^*, v_1^*, u_2^*, v_2^*)$, the AS loss simultaneously maximizes the overlap for both sides of a predicted box with its ground truth. AS loss is defined as follows:

$$\mathcal{L}_{BBO} = 2 - \{\frac{W_{min}}{W_{max}} + \frac{H_{min}}{H_{max}}\} \tag{6}$$

just as the extreme case shown in Fig. 2 (c), W_{min}, H_{min} can be negative values if bounding boxes are non-overlapping. Under this circumstance, BBO loss can still



(*a*) Rendering Result (*b*) Overlapping Case (*c*) Non-overlapping Case

Figure 2: Illustrative Diagram of Stage II Loss. It directly regresses two corner points by minimizing the normalized distance $\frac{d_1}{d_2}$ and enlarges height by minimizing the $1-\frac{H_{min}}{H_{max}}$. Note that We use red to represent the predicted bounding box and green to represent the ground truth (GT) box.

be optimized for non-overlapping cases while the traditional IOU loss fails to do that.

Thirdly, to go a further step for better corner alignment, we introduce Corner Offset (CO) loss. We find that the center loss will be roughly near zero when only the center is aligned correctly but not scale. To this end, we additionally add the Corner Offset Loss ($\mathcal{L}_{CO}$) to achieve accurate box location, which directly minimizes the normalized corner distance. Mathematically, it can be formulated as:

$$\mathcal{L}_{CO} = \frac{D(\hat{p}_1, p_1^*)}{D(p_{c_1}, p_{c_2})} + \frac{D(\hat{p}_2, p_2^*)}{D(p_{c_1}, p_{c_2})}, \tag{7}$$

where $D(\cdot,\cdot)$ is the ℓ_2 distance, $\hat{p}_1, \hat{p}_2$ are the top left and bottom right corner points of the predicted box, respectively. p_1^*, p_2^* are their GT points. p_{c_1}, p_{c_2} are used for the corner points of the smallest enclosing box covering two boxes.

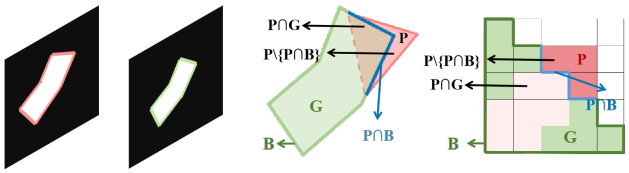
Overall, the final aligned loss $\mathcal{L}_{alg}$ can be summarized as:

$$\mathcal{L}_{alg} = \tau_1 \mathcal{L}_{\text{IOU}} + \tau_2 \mathcal{L}_{BBO} + \tau_3 \mathcal{L}_{CO} \tag{8}$$

We use a convex combination of the three losses, where τ_1, τ_2, τ_3 are 0.4, 0.3, and 0.3, respectively.

Stage III: *Fine Level* Pixel-wise Refinement. After the refinement of stage I and stage II, the predicted rendering mask exhibits only subtle differences compared to the GT mask. We decided to pursue pixel-level supervision optimization further. However, plain pixel-wise loss suffers from an imbalance of edge and non-edge samples. To overcome data imbalance, we propose the edge-aware pixel loss, which improves the BCE loss by tying it to mask prediction errors.

As shown in Fig. 3, we define P as the predicted mask (in red) and G to refer to the GT mask (in green). The key idea of edge-aware loss is to penalize pixel predictions according



(*a*) Prediction (*b*) Ground Truth (*c*) Pixel Errors (*d*) Enlarged Diagram

Figure 3: Illustrative Diagram of Stage III Loss.

Algorithm 1: Cascade Rendering Strategy

Input: Prior $\mathcal{P}$, GT transformation matrix T^*, Prediction transformation matrix $\hat{T}$, Threshold $\delta = 0.1$.
Output: Rendering Mask.
$S^* \leftarrow \mathcal{R}(T^*, \mathcal{P})$, $\hat{S} \leftarrow \mathcal{R}(\hat{T}, \mathcal{P})$; # $\mathcal{R}$ is the rendering process.
Bool In_StageI = *True* **if** $\hat{S} \notin [(1-\delta)S^*, (1+\delta)S^*]$ **else** *False*;
Bool In_StageII = *True* **if** StageI and IOU$(S^*, \hat{S}) < 1-\delta$ **else** *False*;
Bool In_StageIII = *True* **if** not In_StageI and not In_StageII **else** *False*;
repeat
  if In_StageI **then**
    $\mathcal{L}_{total} = \mathcal{L}_{area}$;
    if In_StageII **then**
      $\mathcal{L}_{total} = 0.1\mathcal{L}_{area} + \mathcal{L}_{alg}$;
      if In_StageIII **then**
        $\mathcal{L}_{total} = 0.1\mathcal{L}_{area} + 0.1\mathcal{L}_{alg} + \mathcal{L}_{pxl}$;
      end if
    end if
  end if
  Optimize $\mathcal{L}_{total}$ and then update the weights of three branches $\leftarrow$ loss backward;
until *convergence*;

to the beyond-edge errors. Therefore, the beyond-edge loss ω is given by the number of pixels in P but not in G:

$$\omega = |P \setminus \{P \cap G\}| = |P| - |P \cap G| \tag{9}$$

Obviously, if the predicted mask perfectly correlates with GT mask boundaries, ω should be zero. when $\omega > 0$, it means the predicted mask has a beyond-edge error. Afterward, define B as the set of GT boundary pixels (in dark green). The pixels that are both in the boundaries of P and G can be given by $P \cap B$ (in dark blue), and the proposed edge-aware pixel loss $\mathcal{L}_{pxl}$ can be formulated as:

$$\mathcal{L}_{pxl} = -\sum_i (1+\zeta)(a_i^* \log \hat{a}_i + (1-a_i^*)\log(1-\hat{a}_i)) \tag{10}$$

where $\zeta = \omega$ if i-th pixel $\mathtt{P}_i$ satisfies $\mathtt{P}_i \in P \cap B$ for error pixel prediction, otherwise $\delta = 0$. a_i is the i-th pixel value. In this way, a larger beyond-edge error leads to higher loss values, producing stronger gradients during backpropagation. Additionally, by weighting only the GT boundary pixels, we implicitly address data imbalance.

★ The overall cascade render strategy training procedure is summarized in Algorithm 1.

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.

Datasets and Baselines. We evaluate our R^2-Art on ArtImage (Xue et al. 2021), ReArtMix (Liu et al. 2022c), and

Category	Method	Modalities	Per-part Pose		
			ADD (mm) ↓	AUC (%) ↑	3D IoU (%) ↑
Laptop	A-NCSH (Li et al. 2020)	RGB + Depth	79.5, 68.3	62.9, 63.3	56.7, 40.2
	OMAD (Xue et al. 2021)	RGB + Depth	96.2, 93.1	63.2, 62.1	43.5, 24.1
	AKBNet (Liu et al. 2022a)	RGB + Depth	96.3, 71.3	63.8, 63.4	53.4, 36.8
	ContactArt (Zhu et al. 2024)	RGB + Depth	49.2, 62.1	71.3, 68.5	66.7, 42.1
	R^2-Art (Ours)	RGB	**37.5**, **49.2**	**75.3**, **70.3**	**74.6**, **49.2**
Eyeglasses	A-NCSH (Li et al. 2020)	RGB + Depth	78.3, 452.1, 463.1	65.2, 63.5, 63.0	52.5, 40.2, 39.6
	OMAD (Xue et al. 2021)	RGB + Depth	86.5, 145.6, 148.3	63.2, 61.5, 61.5	22.8, 20.5, 21.4
	AKBNet (Liu et al. 2022a)	RGB + Depth	73.5, 453.2, 516.3	62.1, 61.1, 60.8	48.9, 37.8, 36.1
	ContactArt (Zhu et al. 2024)	RGB + Depth	68.8, 118.1, 121.8	68.1, 66.2, 66.6	55.6. 47.8, 43.8
	R^2-Art (Ours)	RGB	**63.4, 103.5, 106.5**	**71.2, 70.3, 69.5**	**61.3, 59.3, 58.4**
Dishwasher	A-NCSH (Li et al. 2020)	RGB + Depth	121.9, 127.8	60.2, 59.6	84.3, 56.2
	OMAD (Xue et al. 2021)	RGB + Depth	134.5, 143.7	57.8, 56.8	66.5, 38.9
	AKBNet (Liu et al. 2022a)	RGB + Depth	131.5, 138.5	56.6, 55.8	82.8, 54.6
	ContactArt (Zhu et al. 2024)	RGB + Depth	95.3, 99.7	63.1, 61.2	54.2, 67.3
	R^2-Art (Ours)	RGB	**83.5, 87.5**	**66.3, 64.5**	**87.1, 71.3**
Scissors	A-NCSH (Li et al. 2020)	RGB + Depth	66.3, 70.1	65.2, 64.2	46.5, 44.8
	OMAD (Xue et al. 2021)	RGB + Depth	71.2, 73.9	62.4, 61.9	35.6, 34.5
	AKBNet (Liu et al. 2022a)	RGB + Depth	72.1, 74.5	62.2, 61.3	38.3, 37.1
	ContactArt (Zhu et al. 2024)	RGB + Depth	53.6, 57.8	71.2, 69.9	46.7, 45.0
	R^2-Art (Ours)	RGB	**45.6, 48.3**	**73.5, 72.1**	**47.5, 45.6**
Drawer	A-NCSH (Li et al. 2020)	RGB + Depth	101.3, 145.5, 173.6, 143.6	66.3, 63.9, 60.1, 64.5	90.2, 81.5, 78.4, 82.7
	OMAD (Xue et al. 2021)	RGB + Depth	121.3, 139.5, 188.8, 156.3	65.7, 58.9, 59.3, 63.4	75.8, 73.4, 70.2, 71.3
	AKBNet (Liu et al. 2022a)	RGB + Depth	98.5, 132.2, 163.6, 138.5	68.8, 62.1 63.5, 62.1	85.9, 78.6, 77.6, 79.0
	ContactArt (Zhu et al. 2024)	RGB + Depth	81.3, 112.3, 145.8, 113.9	72.3, 63.9, 68.5, 63.2	88.3, 78.6, 76.4, 79.9
	R^2-Art (Ours)	RGB	**72.3, 99.3, 126.3, 97.8**	**78.8, 69.9, 67.8, 71.3**	**91.3, 82.5, 79.6, 83.5**

Table 1: 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 bold. The up or down arrows indicate higher or lower values corresponding to better results.

RobotArm (Liu et al. 2022c) datasets, ranging from synthetic to real-world scenarios. For performance comparison, we evaluate four RGB + Depth based approaches: A-NCSH (Li et al. 2020), OMAD (Xue et al. 2021), AKBNet (Liu et al. 2022b), and ContactArt (Zhu et al. 2024). All baselines take colorized 3D point clouds as input while our R^2-Art only requires a single RGB image to estimate category-level articulation pose.

Implementation Details. During the data pre-processing, the input RGB images are scaled into 224×224 resolution and the prior point clouds are downsampled into 2,048 points. The number of total training epochs is 200. All the experiments are implemented on an NVIDIA GeForce RTX 3090 GPU with 24GB memory.

5.1 Comparison with the SOTA Methods

We compare our results with the classical methods in the synthetic dataset ArtImage to verify the effectiveness of our R^2-Art. The quantitative results are shown in Tab. 1. Overall, we get the best pose estimation result lies in category *laptop*, with $\mathbf{37.5}mm, \mathbf{49.2}mm$ for ADD. This can be explained by the proposed cascade render strategy can outperform objects with similar size and shape at per-part level. Moving to the 3D IoU metric, our prediction errors are significantly better at each part compared to the OMAD and AKBNet. More importantly, compared with the classic articulated pose estimation method A-NCSH, our method also beats it with $\mathbf{42.0}mm, \mathbf{19.1}mm$ regarding *laptop*. Besides, qualitative results are also provided for better comparison in Fig. 4 (left), we can conclude that our prediction keeps more in step with GT compared to the state-of-the-art. The quality improvement achieved by R^2-Art is attributable to the effective utilization of the decoupled pose representation and joint-centric modeling strategy.

Index	Cascade Render Strategy?			ADD (mm)	AUC (%)
	Area	Alignment	Pixel		
I				180.5	51.2
II	✓			148.2	57.4
III	✓	✓		120.5	67.2
IV	✓	✓	✓	105.6	69.9

Table 2: Ablation Study. It is noted that experiments are conducted on the category *Drawer*.

5.2 Ablation Study

Cascade Render Strategy. As mentioned in Sec. 4.3, we use a cascade render strategy for pose optimization. Ablation experiments are conducted to investigate the impact of losses at different stages on performance. Note the category *drawer* is chosen for validation since it is representative due to the most complex structure in ArtImage. Results are shown in Tab. 2 (I-IV). We can conclude that: 1) Compared with I and IV, the latter outperforms the former by $\mathbf{74.9}mm$ on ADD error. The necessity of our cascaded strategy has been validated, as it significantly improves performance. 2) Compared with I to III, the loss at each stage continuously improves performance indeed, progressively approaching the results of the GT mask. This validates the rationality behind the cascade render strategy.

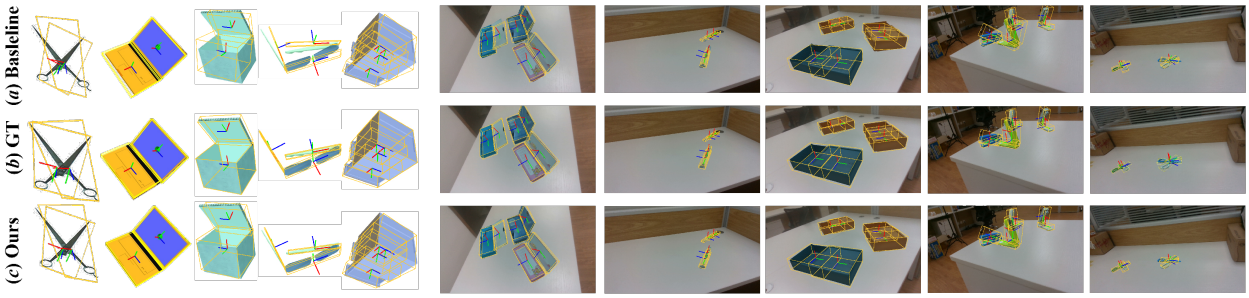


Figure 4: Qualitative results on the synthetic dataset (left) and semi-synthetic scenario (right). It is noted that the left is ArtImage and the right is ReArtMix.

Category	Method	Per-part Pose		
		ADD (mm)	AUC (%)	3D IOU(%)
Box	A-NCSH (Li et al. 2020)	80.2, 85.1	62.4, 55.6	62.1, 59.3
	R^2-Art (Ours)	**30.2**, **37.2**	**78.3**, **76.2**	**80.2**, **81.3**
Stapler	A-NCSH (Li et al. 2020)	75.6, 90.2	66.5, 55.1	40.1, 35.6
	R^2-Art (Ours)	**43.6**, **48.5**	**74.2**, **72.5**	**88.8**, **79.5**
Cutter	A-NCSH (Li et al. 2020)	91.4, 95.6	58.6, 55.7	38.5, 36.2
	R^2-Art (Ours)	**33.3**, **39.3**	**80.3**, **79.7**	**88.2**, **87.7**
Scissors	A-NCSH (Li et al. 2020)	100.3, 99.2	58.2, 59.6	37.5, 34.9
	R^2-Art (Ours)	**69.6**, **70.2**	**79.6**, **75.2**	**69.9**, **67.2**
Drawer	A-NCSH (Li et al. 2020)	95.2, 89.8	64.5, 58.9	62.3, 59.2
	R^2-Art (Ours)	**35.3**, **47.3**	**76.6**, **82.9**	**84.3**, **81.2**

Table 3: Pose estimating results on ReArtMix dataset.

Per-part ADD (mm)							
Part ID	1	2	3	4	5	6	7
A-NCSH (Li et al. 2020)	57.6	62.2	69.7	75.5	83.2	89.4	96.9
R^2-Art (Ours)	**37.6**	**39.2**	**45.3**	**45.6**	**48.9**	**52.6**	**60.8**
Per-part AUC (%)							
Part ID	1	2	3	4	5	6	7
A-NCSH (Li et al. 2020)	72.8	69.2	67.2	63.4	59.6	58.3	54.4
R^2-Art (Ours)	**80.2**	**75.6**	**72.3**	**69.9**	**64.2**	**62.5**	**59.2**

Table 4: Quantitative results on RobotArm dataset.

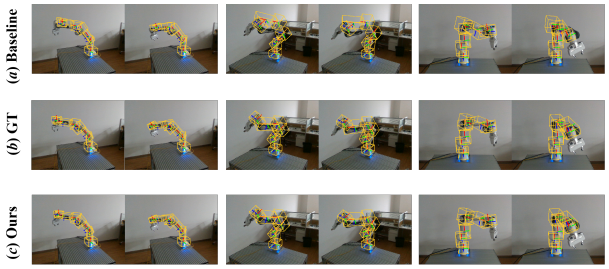


Figure 5: Qualitative results on 7-part RobotArm dataset.

5.3 Generalization Capacity

Experiments on Semi-Synthetic Scenarios. We assess our method for dataset ReArtMix, which incorporates semi-synthetic scenarios. The detailed results are shown in Tab. 3. Our method shows the best performance on category *Box* with only **30.2**mm, **37.2**mm ADD error, and **78.3**%, **76.2%** AUC. This can be explained by that the small-size objects might be easier to be optimized by our cascade render strategy. We show the qualitative results of the five categories in Fig. 4 (Right).

Experiments on Real-world Scenarios. We also train and evaluate R^2-Art using the 7-part RobotArm dataset in real-world scenarios. Note that the baseline (A-NCSH) is trained on RGB-D images, whereas our method is trained solely on RGB images. As shown in the quantitative results (Tab. 4), our method performs well in estimating per-part poses, achieving an average ADD error of **46**mm and **68**% AUC across parts 1 to 7. Additionally, compared to the baseline, R^2-Art improves pose estimation performance with ADD errors of **37.6**mm, **39.2**mm, and **45.3**mm. Although accumulative errors are observed in deeper multi-depth structures (5-th, 6-th, and 7-th parts), our method remains more robust than the baseline as the kinematic structure deepens, leading to more pronounced accumulative errors. The qualitative results are shown in Fig. 5.

6 Conclusion

In this work, the main focus is to study category-level articulation pose estimation with a single RGB image. To deal with the C-RAPE problem, we propose the R^2-Art framework, which seamlessly integrates the observed 2D RGB image to cope with this task without the necessity of depth input for pose estimation during inference. Our framework introduces a decoupled pose representation and explicitly regresses the depth. Considering the kinematic constraints, we propose a customized joint-centric modeling method to alleviate the self-occlusion problems. To further refine the predicted pose, we elaborate a cascade render strategy with a stage-wise stricter optimization. Experimental results demonstrate that our method not only achieves state-of-the-art pose estimation performance on the synthetic dataset (ArtImage) but also exhibits strong generalization capabilities on semi-synthetic (ReArtMix) and real-world articulated object datasets (RobotArm).

Acknowledgements

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

References

Amin, D.; and Govilkar, S. 2015. Comparative study of augmented reality SDKs. *International Journal on Computational Science & Applications*, 5(1): 11–26.

Avetisyan, A.; Dai, A.; and Nießner, M. 2019. End-to-end cad model retrieval and 9dof alignment in 3d scans. In *Proceedings of the IEEE/CVF International Conference on computer vision*, 2551–2560.

Ban, S.; Fan, J.; Zhu, W.; Ma, X.; Qiao, Y.; and Wang, Y. 2024. Real-time Holistic Robot Pose Estimation with Unknown States. *arXiv preprint arXiv:2402.05655*.

Di, Y.; Zhang, R.; Lou, Z.; Manhardt, F.; Ji, X.; Navab, N.; and Tombari, F. 2022. 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*, 6781–6791.

Gu, C.; and Ren, X. 2010. Discriminative mixture-of-templates for viewpoint classification. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V 11*, 408–421. Springer.

Guan, R.; Tu, W.; Li, Z.; Yu, H.; Hu, D.; Chen, Y.; Tang, C.; Yuan, Q.; and Liu, X. 2024. Spatial-Spectral Graph Contrastive Clustering with Hard Sample Mining for Hyperspectral Images. *IEEE Transactions on Geoscience and Remote Sensing*, 1–16.

Hu, D.; Liu, S.; Wang, J.; Zhang, J.; Wang, S.; Hu, X.; Zhu, X.; Tang, C.; and Liu, X. 2024. Reliable Attribute-missing Multi-view Clustering with Instance-level and feature-level Cooperative Imputation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 1456–1466.

Karrer, T.; Wittenhagen, M.; Lichtschlag, L.; Heller, F.; and Borchers, J. 2011. Pinstripe: eyes-free continuous input on interactive clothing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1313–1322.

Li, X.; Wang, H.; Yi, L.; Guibas, L. J.; Abbott, A. L.; and Song, S. 2020. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3706–3715.

Liu, L.; Xu, W.; Fu, H.; Qian, S.; Yu, Q.; Han, Y.; and Lu, C. 2022a. AKB-48: A Real-World Articulated Object Knowledge Base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14809–14818.

Liu, L.; Xu, W.; Fu, H.; Qian, S.; Yu, Q.; Han, Y.; and Lu, C. 2022b. AKB-48: a real-world articulated object knowledge base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14809–14818.

Liu, L.; Xue, H.; Xu, W.; Fu, H.; and Lu, C. 2022c. Toward real-world category-level articulation pose estimation. *IEEE Transactions on Image Processing*, 31: 1072–1083.

Liu, Y.; Yiu, C.; Jia, H.; Wong, T.; Yao, K.; Huang, Y.; Zhou, J.; Huang, X.; Zhao, L.; Li, D.; et al. 2021. Thin, soft, garment-integrated triboelectric nanogenerators for energy harvesting and human machine interfaces. *EcoMat*, 3(4): e12123.

Mo, K.; Guibas, L. J.; Mukadam, M.; Gupta, A.; and Tulsiani, S. 2021. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6813–6823.

Moon, G.; Chang, J. Y.; and Lee, K. M. 2019. Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; and Bao, H. 2019. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4561–4570.

Shen, F.; Ye, H.; Zhang, J.; Wang, C.; Han, X.; and Wei, Y. 2024. Advancing Pose-Guided Image Synthesis with Progressive Conditional Diffusion Models. In *The Twelfth International Conference on Learning Representations*.

Wang, H.; Sridhar, S.; Huang, J.; Valentin, J.; Song, S.; and Guibas, L. J. 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.

Weng, Y.; Wang, H.; Zhou, Q.; Qin, Y.; Duan, Y.; Fan, Q.; Chen, B.; Su, H.; and Guibas, L. J. 2021a. 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.

Weng, Y.; Wang, H.; Zhou, Q.; Qin, Y.; Duan, Y.; Fan, Q.; Chen, B.; Su, H.; and Guibas, L. J. 2021b. CAPTRA: CAtegory-level Pose Tracking for Rigid and Articulated Objects from Point Clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Xiong, H.; Fu, H.; Zhang, J.; Bao, C.; Zhang, Q.; Huang, Y.; Xu, W.; Garg, A.; and Lu, C. 2023. RoboTube: Learning Household Manipulation from Human Videos with Simulated Twin Environments. In *Conference on Robot Learning*, 1–10. PMLR.

Xue, H.; Liu, L.; Xu, W.; Fu, H.; and Lu, C. 2021. OMAD: Object Model with Articulated Deformations for Pose Estimation and Retrieval. *arXiv preprint arXiv:2112.07334*.

Yang, Z.; Liu, S.; Hu, H.; Wang, L.; and Lin, S. 2019. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9657–9666.

Yu, Q.; Wang, J.; Liu, W.; Hao, C.; Liu, L.; Shao, L.; Wang, W.; and Lu, C. 2023. GAMMA: Generalizable Articulation Modeling and Manipulation for Articulated Objects. *arXiv preprint arXiv:2309.16264*.

Zhang, L.; Han, Z.; Zhong, Y.; Yu, Q.; Wu, X.; et al. 2024a. VoCAPTER: Voting-based Pose Tracking for Category-level Articulated Object via Inter-frame Priors. In *ACM Multimedia 2024*.

Zhang, L.; Meng, W.; Zhong, Y.; Kong, B.; Xu, M.; Du, J.; Wang, X.; Wang, R.; and Liu, L. 2025. U-COPE: Taking a Further Step to Universal 9D Category-Level Object Pose Estimation. In *European Conference on Computer Vision*, 254–270. Springer.

Zhang, L.; Zhong, Y.; Wang, J.; Min, Z.; Liu, L.; et al. 2024b. Rethinking 3D Convolution in ℓ_p-norm Space. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zheng, T.; Zhao, S.; Liu, Y.; Liu, Z.; and Cai, D. 2022. Scaloss: Side and corner aligned loss for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3535–3543.

Zhu, Z.; Wang, J.; Qin, Y.; Sun, D.; Jampani, V.; and Wang, X. 2024. ContactArt: Learning 3d interaction priors for category-level articulated object and hand poses estimation. In *2024 International Conference on 3D Vision (3DV)*, 201–212. IEEE.