Toward Real-World Category-Level Articulation Pose Estimation

Liu Liu, Han Xue, Wenqiang Xu, Haoyuan Fu, and Cewu Lu, *Senior Member, IEEE*

***Abstract*— Human life is populated with articulated objects. Current Category-level Articulation Pose Estimation (CAPE) methods are studied under the single-instance setting with a fixed kinematic structure for each category. Considering these limitations, we aim to study the problem of estimating part-level 6D pose for multiple articulated objects with unknown kinematic structures in a single RGB-D image, and reform this problem setting for real-world environments and suggest a CAPE-Real (CAPER) task setting. This setting allows varied kinematic structures within a semantic category, and multiple instances to co-exist in an observation of real world. To support this task, we build an articulated model repository ReArt-48 and present an efficient dataset generation pipeline, which contains Fast Articulated Object Modeling (FAOM) and Semi-Authentic MixEd Reality Technique (SAMERT). Accompanying the pipeline, we build a large-scale mixed reality dataset ReArtMix and a real world dataset ReArtVal. Accompanying the CAPER problem and the dataset, we propose an effective framework that exploits RGB-D input to estimate part-level pose for multiple instances in a single forward pass. In our method, we introduce object detection from RGB-D input to handle the multi-instance problem and segment each instance into several parts. To address the unknown kinematic structure issue, we propose an Articulation Parsing Network to analyze the structure of detected instance, and also build a Pair Articulation Pose Estimation module to estimate per-part 6D pose as well as joint property from connected part pairs. Extensive experiments demonstrate that the proposed method can achieve good performance on CAPER, CAPE and instance-level Robot Arm pose estimation problems. We believe it could serve as a strong baseline for future research on the CAPER task. The datasets and codes in our work will be made publicly available.**

***Index Terms*— Articulation estimation, category-level 6D pose, real-world, articulation parsing.**

I. Introduction

ARTICULATED objects are pervasive in our everyday life. Unlike rigid objects which can be regarded as a whole when moving in 3D space, articulated objects are usually composed of several rigid parts that are linked by different kinds of joints, e.g. revolute, prismatic, screw, etc. In comparison with rigid objects, the diverse kinematic structures endow the articulated object higher Degree of Freedom (DoF), making the estimation of articulated object pose challenging.

Recently, the Category-level Articulated object Pose Estimation (CAPE) task has drawn increasing attention [1], [2]. Since the mechanism of estimating the articulation status from a single-view observation (e.g. RGB-D image, point cloud) can benefit downstream research and applications, such as scene understanding, robot manipulation, and VR/AR. However, currently, the task is generally studied under a single-instance setting with a synthetic point cloud, where the articulated object has a known and fixed kinematic structure for each category. Apparently, this assumption does not hold for many real-world cases as shown in Fig. 1. Specifically, (1) the synthetic object point cloud may have a domain gap for real-world applications, (2) daily objects may have different kinematic structures within a semantic category, e.g. drawers with different numbers of columns, (3) multiple objects may co-occur in a single observation. Given the gap between the current research direction and the real-world requirements, we extend the CAPE task by considering all the issues and reformulate the problem setting as CAPE-Real (short for **CAPER**). To support the CAPER task, we proposed a novel RGB-D based dataset **ReArtMix** which contains the objects from our proposed articulated model repository named **ReArt-48** for training a baseline framework to address the CAPER task.

Manuscript received May 27, 2021; revised October 13, 2021; accepted November 29, 2021. Date of publication January 5, 2022; date of current version January 12, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700800, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102, in part by the National Natural Science Foundation of China under Grant 61772332, and in part by the Shanghai Qi Zhi Institute under Grant SHEITC 2018-RGZN-02046. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Andrea Fusiello. *(Liu Liu and Han Xue contributed equally to this work.) (Corresponding author: Cewu Lu.)*

Liu Liu, Han Xue, Wenqiang Xu, and Haoyuan Fu are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: liuliu1993@sjtu.edu.cn; xiaoxiaoxh@sjtu.edu.cn; vinjohn@sjtu.edu.cn; simon-fuhaoyuan@sjtu.edu.cn).

Cewu Lu is with the Qing Yuan Research Institute and the MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Shanghai Qi Zhi Institute, Shanghai 200030, China (e-mail: lucewu@sjtu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2021.3138644, provided by the authors.

Digital Object Identifier 10.1109/TIP.2021.3138644

Nomenclature

CAPE	Category-level Articulation Pose Estimation.
CAPER	Category-level Articulation Pose Estimation-Real.
FAOM	Fast Articulated Object Modeling.
SAMERT	Semi-Authentic MixEd Reality Technique.
APN	Articulation Parsing Network.
NOCS	Normalized Object Coordinate Space.
URDF	Unified Robot Description Format.
A-NCSH	Articulation Normalized Coordinate Space Hierarchy.
NAOCS	Normalized Articulated Object Coordinate Space.
NPCS	Normalized Part Coordinate Space.
$\mathcal{P}$	Point Cloud.
$\mathcal{S}$	Part Segments.
δ	Joint Type.

q Joint Location.
u Joint Axis.
R 3D Rotation.
t 3D Translation.
H Joint Heatmap.
V Offset Vector.

When constructing the dataset ReArtMix, two major challenges exist: First, there is no suitable public model repository for real articulated objects, current popular articulated model repositories either contain only synthetic models [3], [4] or support only instance-level task [5]. Second, collecting RGB-D training data with articulated pose annotations is cost-prohibitive. Therefore, we take a synthetic path to create the dataset. In detail, we build a **Re**al-world **Art**iculated model repository named **ReArt-48** which contains 48 different scanned models under 5 categories. The part segmentation and joint properties of the scanned models are annotated through a **F**ast **A**rticulated **O**bject **M**odeling (**FAOM**) pipeline. In our FAOM, we design an offline annotation interface that integrates part region segmentation by polygon, joint properties annotation with multi-view refinement, and animation rendering that helps verification. In FAOM, each real scanned articulated object can be modeled within 15 minutes. To the best of our knowledge, FAOM is the first articulated object modeling pipeline and promote efficiently building large-scale real-world datasets in an easy way. Once obtaining the annotated object models, we composite them with real-world background RGB-D images in a physically plausible manner and automatically generate a large-scale mixed reality dataset **ReArtMix** along with the annotations required (e.g. object part segmentation, part pose, joint properties, etc.) by our proposed **S**emi-**A**uthentic **M**ix**E**d **R**eality **T**echnique (**SAMERT**). To prove ReArtMix can effectively reduce domain gap when transferring to real-world scenarios, we also build a fully real-world dataset **ReArtVal** for validation. The quantitative results are reported in Table II. In comparison with annotating real-world images totally by a human ($\sim$ 2 min/image), the FAOM-SAMERT pipeline can save a proliferation of human labors for image capturing and annotation ($\sim$ 0.2 sec/image).

Accompanying the dataset ReArtMix, we propose a learning framework inspired by Normalized Object Coordinate Space (NOCS) [6]. Our method can utilize both RGB and depth information, handle multiple instances in a single forward pass. It consists of an RGB-D based object detector, a point-cloud based Articulation Parsing Network and a Pair Pose Estimation Network that can adapt to varied kinematic structures for detected instances. Specifically, for each detected articulated instance, our method applies an Articulation Parsing Network (APN) that exploits part segmentation module to analyze the kinematic structure. Next, each connected part pairs from the instance is fed into a PointNet++ encoder-decoder architecture to predict pair NOCS map as well as joint properties, e.g. joint axis and joint location. Finally, based on the NOCS map and joint prediction, we recover part-level 6D pose by a pair pose optimization mechanism. To evaluate the framework, we test our method on the proposed datasets ReArtMix and ReArtVal with Average Precision on rotation error, translation error, 3D bounding box IoU, joint angle error and joint distance error as metrics, and also test on CAPE setting of samples rendered with PartNet-Mobility [3] and a real-world Robot Arm dataset.

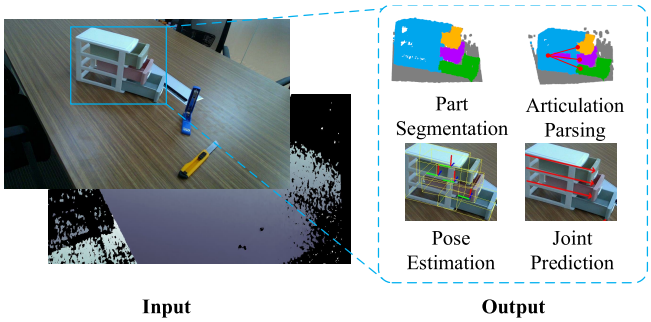

Fig. 1. In real-world scenarios, the category-level articulated object pose estimation problem will face more challenges including varied kinematic structures within a semantic category, and multiple instances within a single RGB-D image.

Our contributions can be summarized in three folds:

(1) we bring the previously proposed CAPE problem towards real-world setting as CAPER problem, which is more realistic and complicated as we consider multiple unseen objects with varied kinematic structures in a semantic category.
(2) We collect a real-world articulated model repository ReArt-48 and propose a FAOM-SAMERT pipeline to make the preparation of the real-world-like training dataset feasible. We also build a full annotated real-world dataset ReArtVal to validate the performance of our method trained on ReArtMix.
(3) We propose an effective framework to address the CAPER problem that could deal with multiple instance occurrences and unknown kinematic structure issues. Experiments show that our method could serve as a strong baseline for the task.

II. Related Work

A. Instance-Level 6D Pose Estimation

There are plenty of works regarding instance-level 6D pose estimation task, which aims to predict objects' 3D location and 3D rotation given available 3D CAD models [7]–[9]. According to the different ways of object alignment, these past works could be roughly categorized into template-based methods and feature-based methods. Generally, template techniques [10], [11] exploit a rigid template to predict 6D pose by matching observed partial point clouds with corresponding 3D models with registration algorithms such as ICP. Template-based methods work well on detecting and recognizing texture-less objects but might fail for highly occluded instances. The other family, feature-based methods [12]–[14] aim to regress the object surface position for every pixel in the image and establish the 2D-3D correspondences by extracted local features. Under the powerful enough texture features, these methods could handle the 6D pose estimation well.

B. Category-Level 6D Pose Estimation

Instance-level pose estimation aims to predict objects' 3D rotation and translation given 3D object models [9], [15]. On the contrary, the goal of category-level pose estimation is to predict an input instance's pose and location relative to category-specific representation. The first proposed method is to predict 3D-3D per-pixel correspondences between observations and canonical coordinates using a normalized space for category-level representation [6]. Besides, [16] present to directly optimize the predicted rotation, translation, and scale simultaneously in a single monocular image. In addition to the static image-based estimator, [17] propose the first category-level pose tracker, which adopts geometric or semantic keypoints to estimate the interframe motion. Among these methods, they mainly focus on rigid pose estimation while our task aims to estimate articulation pose in the real world.

C. Articulated Object Pose Estimation

Articulation estimation has been studied for decades, where previous works can be broadly categorized into interaction-based, video-based, and single-view-based. Interaction-based methods are mostly studied in the Robotics community, which adopt feedback from manipulation actions to characterize a distribution of articulation models [18]. The interaction-based approaches require some ways to interact with the object, which limits the applicability. On the other hand, a video recorded with a moving articulated object can be a good source to estimate its motion property in an image sequence [19]. With a simplified setting, instead of a clip of video that Yi *et al.* take a pair of unsegmented shape representations as input to predict correspondences, 3D deformation flow and part-level segmentation [20]. With the development of deep learning techniques, the single-view-based CAPE setting is becoming possible. A-NCSH [1], as an extension of NOCS, is developed for single articulated object pose estimation. However, it holds an obvious limitation that requires fixed kinematic structure as prior information for each input object while our CAPER setting allows multiple instances and various kinematic structures.

D. Datasets for Articulation Estimation

An unavoidable challenge for learning the articulation estimation model is the lack of large-scale training data with sufficient fully annotated category, instance, mask and 6D pose per part. To solve this issue, several works focus on generating training data rendered with rigid synthetic object models [21]. Considering articulated objects, PartNet-Mobility [3] is built with virtual CAD models with unreal texture, which hinders the development towards real-world applications. More recently, RBO dataset [5] owns only 14 objects captured from the real world and none of them belongs to the same category. Therefore, we build a new realistic articulated object dataset and propose an automatic way to generate large-scale training samples.

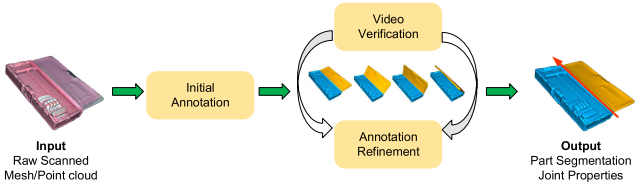

Fig. 2. **Fast Articulated Object Modeling** pipeline. We use an iterative method to refine the initial articulation annotation.

III. Problem Statement

As mentioned earlier, the CAPER problem setting advances the CAPE setting in two aspects: (1) multi-instance in an image. This setting helps us to define an end-to-end deep learning task to analyze articulated object pose, which is more in line with the real-world environment compared with CAPE setting. (2) objects of the same semantic category have different kinematic structures. This setting aims to solve the unseen objects with various structures.

In CAPER problem, given a single RGB-D image I as input, a CAPER model will firstly learn to detect N objects with bounding box $\mathcal{O} = \{(u_1^i, v_1^i, u_2^i, v_2^i)\}_{i=1}^N$ with left top (u_1^i, v_1^i) and right bottom (u_2^i, v_2^i) coordinates as well as their corresponding categories $\mathcal{C} = \{c^i\}_{i=1}^N$. In dealing with each detected instance, we project the instance patch of i^{th} instance to local point cloud $\mathcal{P}^i = \{(x_j^i, y_j^i, z_j^i, r_j^i, g_j^i, b_j^i)\}_{j=1}^M$ which has M points and predict segmentation for unknown number of parts $\mathcal{S} = \{s_j\}_{j=1}^M$, where s_j is the index of one of the maximum K parts, as well as kinematic structure for the input instance, given the segmentation $\mathcal{S}$. Then, based on the kinematic structure, we randomly sample the part pairs k_1 and k_2 that are connected by joint, and predict: (1) part-level NOCS map $\mathcal{P}'$ describing canonical representation $\mathcal{P}' = \{\mathbf{p}'_j = (x'_j, y'_j, z'_j) \in \mathbb{R}^3\}_{j=(k_1,k_2)}$ (we use symbol $'$ to define the coordinate in NOCS space); (2) joint properties that describes joint type $\delta^{(k_1,k_2)}$, location $\mathbf{q}^{(k_1,k_2)}$ and axis $\mathbf{u}^{(k_1,k_2)}$. Finally, given these prediction results, we recover the 3D rotation $\{R^{(k_1)}, R^{(k_2)}\}$ and 3D translation $\{\mathbf{t}^{(k_1)}, \mathbf{t}^{(k_2)}\}$ for the part pairs.

IV. Datasets

As there exists no dataset to fully support the CAPER task, we construct the ReArtMix dataset by taking a mixed reality approach to reduce the human labor for annotation. Firstly, we collect scanned models of common real objects with varied kinematic structures and categories, and annotate part segmentation along with joint properties of the objects using the proposed FAOM pipeline (Sec. IV-A). Then, we compose these articulated models with real-world background RGB-D images to obtain training samples with full annotations (Sec. IV-B). Last but not the least, we build up a real dataset for validation (Sec. IV-C).

A. FAOM, Fast Articulated Object Modeling

1) Model Repository and Annotation: We scanned 48 hand-scale objects from 5 common categories (such as box and

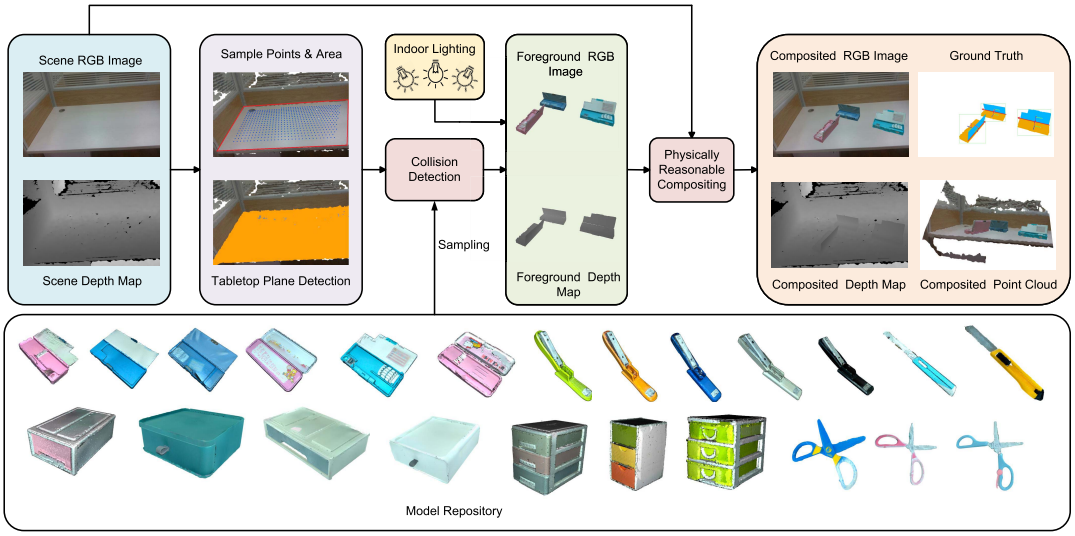

Fig. 3. Our **Semi-Authentic MixEd Reality Technique (SAMERT)**. We combine articulated models scanned from real objects with RGB-D images captured from real scenes to generate composed real-world RGB-D images. Plane detection and collision detection are performed to make the synthetic data physically reasonable.

stapler) in our daily life with EinScan Pro 2020[1] (version 3.6.0.5). To convert these scanned models to articulated models, instead of adopting the traditional 3D reverse engineering software[2] which is tedious and requires expertise, we propose a fast modeling method FAOM illustrated in Fig. 2. During real-world object scanning, we split all the objects into two groups. The first one is those can be disassembled, e.g. drawer rack and columns. We scan the parts of these models separately and then align them into global coordinates together. The other one is those cannot be disassembled. We scan them in a fully open way and segment them into several movable parts. In this way, we can efficiently obtain a large number of real-scanned models and annotate them in the FAOM pipeline. FAOM provides a user interface that integrates the annotation functions of part segmentation with predicted initialization, joint properties annotation with multi-view images, and animation verification on the whole object.

The annotated articulated objects are described with the widely used Unified Robot Description Format (URDF) [22], an XML file format to describe all elements of articulated objects with chain or tree structure, including joint properties and part meshes. The base link is the origin of the kinematic tree. Finally, we collect 48 real-world articulated models with full rich annotations to build our ReArt-48 model repository. Some of the models are illustrated in Fig. 4.

2) Comparison With Other Model Repositories: As shown in Table I, our object models have full features required for the CAPER task. Compared to other real-world model repositories, such as RBO [5], our ReArt-48 holds two advantages: (1) the number of objects is more than twice the object number of the RBO dataset. (2) ReArt-48 contains 5 categories and each category has around 10 models, which supports category-level articulated object analysis research. Therefore, rendering with our articulated models could own much more details which provide powerful information for us to estimate 6D pose in the real world.

[1]https://www.einscan.com

[2]https://www.3dsystems.com/software/geomagic-design-x

Fig. 4. Some of **ReArt-48** model repository. We consider box, stapler, cutter, scissors and drawer in our ReArt-48.

TABLE I

COMPARISON WITH OTHER POPULAR MODEL REPOSITORIES

Model repository	Articulation	Real	Category-level	Object Num
ShapeNet[23]			✓	>50K
YCB[24]		✓		21
LineMod[25]		✓		15
Shape2Motion[4]	✓		✓	2,440
PartNet-Mobility[3]	✓		✓	2,346
RBO[5]	✓	✓		14
Ours	✓	✓	✓	48

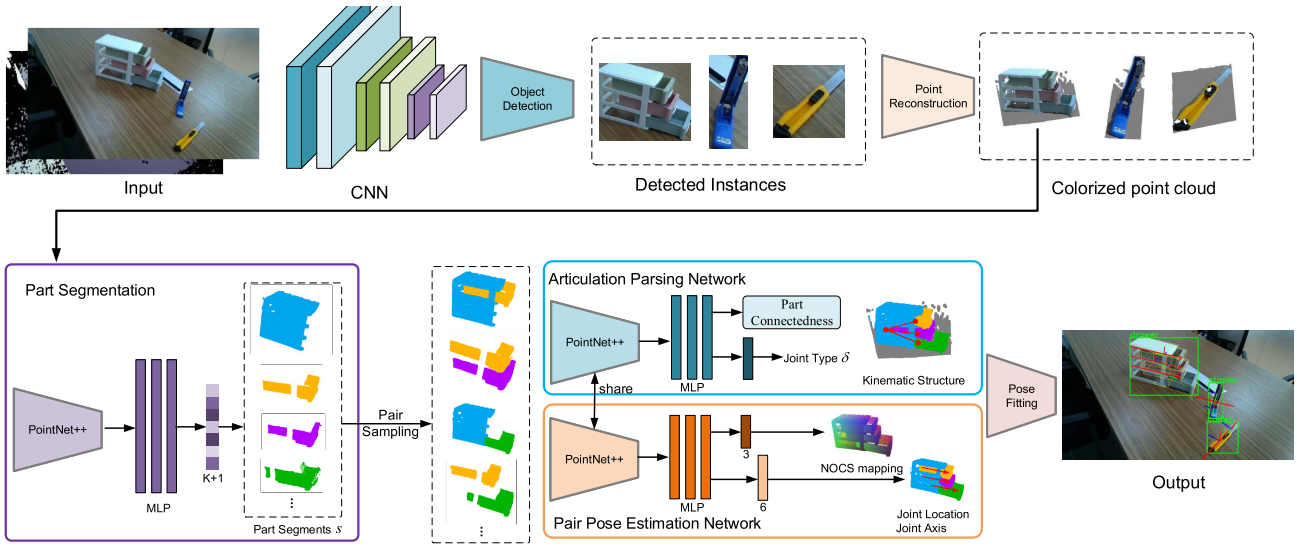

Fig. 5. Our method for real-world articulated object pose estimation. There are four modules: object detection, part segmentation, Articulation Parsing Network and Pair Pose Estimation Network. In our framework, we detect each articulated object from an RGB-D image and predict each instance's part-level NOCS and joint properties.

B. SAMERT, ReArtMix Dataset Generation

The training data for the CAPER task requires realistic visual quality and high-precision annotations. However, the accurate annotation for the part-level pose is time-consuming, which is prohibitive for large-scale dataset preparation. To address this, we propose a novel **Semi-Authentic MixEd Reality Technique (SAMERT)**, whose pipeline is illustrated in Fig. 3, based on Unity Engine[3] of version 2019.3 to automatically generate such training dataset with articulated models and pre-collected real-world RGB-D background images. These background images are snapshots from 20 different tabletop scenes with $\sim$ 200 viewpoints per scene. Given the real-scanned object models and background images, we present a *physically reasonable compositing* strategy to render RGB-D images with full annotations. Firstly we randomly place the articulated objects with random scale and joint state onto the desktop plane of the background image in 3D camera space, which is predicted by PlaneRCNN [26] and RANSAC algorithm. Part-level object pose and joint states are tracked and recorded, as well as the collision status among objects. The physics engine will guarantee the physically plausible requirements, that is, no floating objects and intersections between objects. We also randomly generate multiple directional lights with different orientations and colors to imitate indoor lighting. Compared with another simulation-to-real data generation pipeline, such as Wang *et al.* [6], our SAMERT holds several advantages: Firstly, we render articulated objects to generate numerous samples, which might cause much more difficulty in data generation, such as self-collision. Secondly, we adopt a physics engine Unity that fully considers the physical properties of our objects to avoid collision in the rendered images while Wang *et al.* [6] only consider visual rendering. Finally, our SAMERT does not require any extra background scene annotations, e.g. 3D plane and location sampling points, which save a proliferation of human labors.

With the SAMERT process, we generate 100K RGB-D images that are rendered on our collected over 1K background scenes, of which 90K are set aside for training and 10K for validation. The render scenes contain various tabletop backgrounds and are captured using Realsense D435i camera, in which the scenes involve office, school, home and so on. Among these images, 37 articulated models in ReArt-48 are used to generate training images while the rest of 11 objects are selected as unseen for validation. With the real scanned models and real background scenes, our synthetic semi-authentic data could drastically reduce the gap between virtuality and reality, which can be quantitatively verified with the real validation set described next.

C. ReArtVal, Real Data Acquisition for Validation

To validate the performance of our method in the real world, we also build a fully real dataset in the form of video sequences. For each category, we capture over 6K RGB-D frames in 6 real-world tabletop scenes using RealSense D435i camera.[4] In terms of data annotation, we propose a semi-automatic part-level 6D pose annotation pipeline referred to LabelFusion [27], in which the annotation process can be simplified into two steps: initial frame annotation and automatic ground truth generation by RGB-D registration. Finally, we capture over 6K RGB-D frames with full annotations (bounding box, part segmentation, part-level 6D pose, joint properties) to build our real-world dataset.

V. Method

We propose a framework to address the CAPER task. The key challenge is to handle multiple instances within a

[3]https://unity.com

[4]https://www.intelrealsense.com/depth-camera-d435/

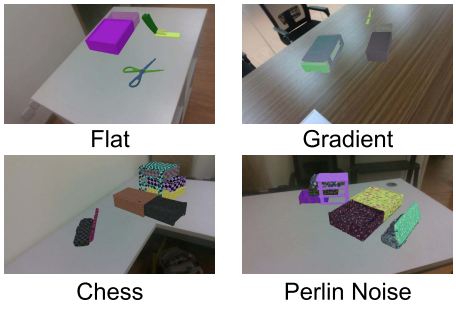

Fig. 6. ReArtMix texture augmentation. We use Flat, Gradient, Chess and Perlin Noise for augmenting the textures in our ReArtMix to ensure generalization to real-world.

single image and varied kinematic structures (unknown joint properties) within a semantic category. The multi-instance problem can be tackled by object detection (Sec. V-A). And an Articulation Parsing Network is designed to handle the detected instance with unknown kinematic structures (Sec. V-B). Finally, we sample the part pairs from the parsed articulation structure and predict the NOCS map as well as joint property for per-part pose recovering (Sec. V-C and Sec. V-D). The training is formulated as a multi-task learning problem (Sec. V-E). The overall pipeline is displayed in Fig. 5.

A. Object Detection

High detection performance is a high priority for our method. Here we adopt an anchor-based object detector RetinaNet [28] for dense detection with RGB-D image, where the input contains 6 channels (3 for RGB and 3 for XYZ). Since unseen articulated objects usually hold various appearances in texture, we adopt texture augmentation [29] to improve the appearance diversity of the training set. Specifically, we randomly generate flat colors, gradients of colors, chess patterns, and Perlin noise as object textures shown in Fig. 6.

B. Articulation Parsing Network

For each detected bounding box $\mathcal{O}^i$, we crop the corresponding region in the input RGB-D image and transform the patch into colorized point cloud $\mathcal{P}^i = \{(x_j^i, y_j^i, z_j^i, r_j^i, g_j^i, b_j^i)\}_{j=1}^M$. Then we design an Articulation Paring Network (APN) to analyze and parse the kinematic structure of the detected instance. In APN module, we train a PointNet++ [30] encoder-decoder architecture for feature extraction. At the end of PointNet++ feature propagation layers, we build three multi-layer perceptions (MLPs) with ReLU activation function that outputs $K+1$ channels for part segmentation, where K is the maximum number of rigid parts in our dataset and 1 indicates the background. At the same time, we also split the input colorized point cloud $\mathcal{P}^i$ into several semantic parts based on the predicted segments $\{\mathcal{P}^{i,(k)}, k = 1, 2 \dots K\}$. Each point cloud segment would be fed into a parallel PointNet++ encoder network and extract part descriptor $(F)^{(k)}$ with fixed dimensions for kth rigid part. Finally, we build three extra MLPs for these part descriptors and predict binary part connectedness. In this way, our APN module could parse the unknown articulation structure for the detected instance.

C. Pair Articulation Pose Estimation

1) Pair NOCS Prediction: After articulation parsing, we could obtain the sampled part pairs $\mathcal{P}^{i,(m)}$ and $\mathcal{P}^{i,(n)}$ that are correctly connected by joint. Based on these part pairs, we build our Pair NOCS prediction branch that estimates part-level NOCS mapping for articulation pose fitting. The part-level NOCS map is defined for each separate rigid part rather than the whole object, in which we define the rest state for every part and normalized the shape of the rest state into $[0, 1]$ and centered into $(0.5, 0.5, 0.5)$. In pair NOCS prediction, we also build a PointNet++ architecture with the end of 3 channels to densely predict the normalized coordinate $\mathbf{p}'_j$ in NOCS space for each observed part.

2) Joint Prediction: Current methods such as A-NCSH [1] for the CAPE setting require known kinematic structure as prior knowledge. In our method, given the parsed articulated object, we could automatically obtain the kinematic structure without known joint properties. For the input connected part pairs $\mathcal{P}^{(k_1)}$ and $\mathcal{P}^{(k_2)}$, we build another branch at the end of PointNet++ to predict the joint property connecting the input part pairs. In our joint prediction module, we aim to predict three types of information for each kinematic joint: joint type $\delta^{(k_1,k_2)}$ (also known as kinematic way), joint location $\mathbf{q}^{(k_1,k_2)}$ and joint axis $\mathbf{u}^{(k_1,k_2)}$. Due to the part pairs point cloud input, we could densely predict joint property in our framework. Specifically, we introduce two new branches into the end of PointNet++ with 2 and 6 channels, in which 2 indicates that the joint type prediction aims to classify each part into 2 kinematic way δ_j (prismatic or revolute) and 6 channels are used to regress the joint location $\mathbf{q}_j$ and joint axis $\mathbf{u}_j$. For joint axis $\mathbf{u}^{(k_1,k_2)}$, we average the $\mathbf{u}_j$ on points from part pairs k_1 and k_2:

$$\mathbf{u}^{(k_1,k_2)} = \frac{\sum_{j=1}^{M} \mathbf{u}_j \mathbb{1}(s_j = k_1) + \mathbf{u}_j \mathbb{1}(s_j = k_2)}{\sum_{j=1}^{M} \mathbb{1}(s_j = k_1) + \mathbb{1}(s_j = k_2)} \tag{1}$$

Simultaneously, we follow the voting scheme to estimate the joint location (anchor point) $\mathbf{q}^{(k_1,k_2)}$. Different from [1] that predict joint location in NOCS space, we vote for the $\mathbf{q}^{(k_1,k_2)}$ directly in camera space. We build three parallel MLP branches with 1, 3 and 2 channels to densely predict per-point heatmap, offset vector and support point classification respectively. To be specific, for input pair part point cloud $\mathcal{P}^{(k_1,k_2)}$, we exploit the point position $\mathbf{p}_j = (x_j, y_j, z_j)$ to predict the heatmap $\mathbf{H}_j^{(k_1,k_2)}$:

$$\mathbf{H}_j^{(k_1,k_2)} = 1 - \frac{\|(\mathbf{p}_j - \mathbf{q}^{(k_1,k_2)}) \times \mathbf{u}^{(k_1,k_2)}\|}{\sigma} \tag{2}$$

where σ is the distance threshold that defines neighbor radius. Here we only consider the points where the distance to the joint is smaller than σ. These points are named as support points and we use another MLP to achieve a binary classification for these support points $\mathcal{P}^{s,(k_1,k_2)}$.

In order to vote for the joint location $\mathbf{q}^{(k_1,k_2)}$, we also perform a per-point offset vector $\mathbf{V}_j^{(k_1,k_2)}$ that defines unit direction from point $\mathbf{p}_i$ to the joint:

$$\mathbf{V}_j^{(k_1,k_2)} = \frac{(\mathbf{p}_j - \mathbf{q}^{(k_1,k_2)}) \times \mathbf{u}^{(k_1,k_2)}}{\mathbf{H}_j^{(k_1,k_2)}} \tag{3}$$

Finally, we adopt a voting scheme to obtain the joint location $\mathbf{q}^{(k_1,k_2)}$ by:

$$\mathbf{q}^{(k_1,k_2)} = \frac{1}{\mathcal{N}_s}\sum\nolimits_{j=1}^{M} \mathbb{1}(\mathbf{p}_j = \mathbf{p}_j^s)(\mathbf{p}_j + \sigma \mathbf{V}_j^{(k_1,k_2)}(1 - \mathbf{H}_j^{(k_1,k_2)})) \tag{4}$$

where $\mathcal{N}_s$ indicates the number of support points.

D. Pair Pose Optimization

For pair connected parts with predicted NOCS map and joint properties, we further optimize the part-level 6D pose following the pose fitting algorithm with kinematic constrains [1]. Specifically, given the point clouds of part pairs $\mathcal{P}^{(k_1)}$ and $\mathcal{P}^{(k_2)}$ as well as their corresponding NOCS map $\mathcal{P}'^{(k_1)}$ and $\mathcal{P}'^{(k_2)}$, we first separately recover the 3D rotation and translation R^{k_1}, t^{k_1} and R^{k_2}, t^{k_2} by minimizing the energy function E_p for fitting the NOCS canonical representation into observed points in camera space:

$$E_p = \frac{1}{\mathcal{N}_{k_1}}\|\mathcal{P}^{(k_1)} - (s^{k_1}R^{k_1}\mathcal{P}'^{(k_1)} + t^{k_1})\|^2 + \frac{1}{\mathcal{N}_{k_2}}\|\mathcal{P}^{(k_2)} - (s^{k_2}R^{k_2}\mathcal{P}'^{(k_2)} + t^{k_2})\|^2 \tag{5}$$

where s^{k_1} and s^{k_2} are the normalized scale factors for part k_1 and k_2 that are initialized by Umeyama algorithm [31]. Then we also consider kinematic constraints by further optimizing pose with predicted joint properties. For the joint axis $\mathbf{u}^{(k_1,k_2)}$ that connects the part pairs k_1 and k_2, we minimize the energy function with kinematic constraints E_k:

$$E_k = \|R^{k_1}\mathbf{u}^{(k_1,k_2)} - R^{k_2}\mathbf{u}^{(k_1,k_2)}\|^2 \tag{6}$$

Finally, the pose optimization is to minimize the total energy function $E = E_p + E_k$, which is the sum of per-part energy E_p and kinematic constrained energy E_k. Similar to [1] and [6], we also use RANSAC for outlier removal.

E. Multi-Task Loss Function

For articulated object detection task, we use focal loss [28] and smoothL1 loss for bounding box classification and regression $\mathcal{L}_{det} = \mathcal{L}_{cls} + \mathcal{L}_{reg}$. In terms of pose estimation, the total loss for articulated object estimation $\mathcal{L}_{pose}$ is the sum of losses from Articulation Parsing Network $\mathcal{L}_{APN}$ and those from Pair Pose Estimation Network $\mathcal{L}_{PPEN}$, where:

$$\mathcal{L}_{APN} = \lambda_1\mathcal{L}_{seg} + \lambda_2\mathcal{L}_{connect} + \lambda_3\mathcal{L}_{type} \tag{7}$$

$$\mathcal{L}_{PPEN} = \lambda_4\mathcal{L}_{nocs} + \lambda_5\mathcal{L}_{supp} + \lambda_6\mathcal{L}_{heatmap} = +\lambda_7\mathcal{L}_{offset} + \lambda_8\mathcal{L}_{axis} \tag{8}$$

where the eight multiplication factors λ_1, λ_2, λ_3, λ_4, λ_5, λ_6, λ_7, λ_8 are set to be 1, 1, 0.5, 10, 1, 0.5, 1, 0.2. To find proper hyper-parameters, we use the cross-entropy loss for part connectedness classification $\mathcal{L}_{connect}$, joint type loss $\mathcal{L}_{type}$ and segmentation task $\mathcal{L}_{seg}$. Then L2 is adopted for NOCS map loss $\mathcal{L}_{nocs}$, joint prediction related loss $\mathcal{L}_{heatmap}$, $\mathcal{L}_{offset}$ and $\mathcal{L}_{axis}$. Finally, we use IoU loss for joint support points classification $\mathcal{L}_{supp}$. Specifically, we initialize these hyper-parameters followed by A-NCSH [1], and 0.5 times to 1.5 times initial values as search range, then divide the ReArtMix dataset into ten folds to find the optimal hyper-parameters by cross-validation method.

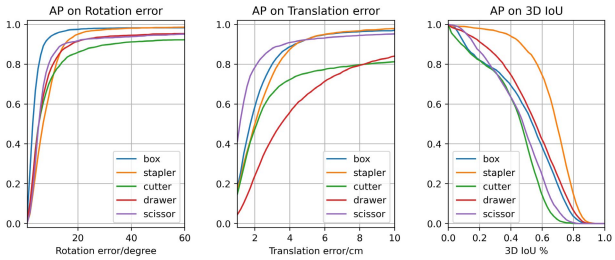

Fig. 7. Articulated pose estimation results on ReArtMix dataset.

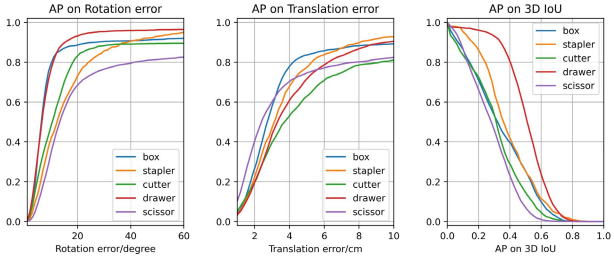

Fig. 8. Articulated pose estimation results on ReArtVal dataset.

VI. Experiments

A. Experiment Setup

1) Implementation: We use RetinaNet [28] with ResNet-50 backbone [32] and FPN [33] as our object detector. We use the SGD optimizer with the momentum of 0.9 to train the detector with total training epoch 8. During training PointNet++, we use Adam optimizer with an initial learning rate of 0.001 and 16 batch size. Both object detector and PointNet++ pose estimator are trained on an Intel(R) Xeon(R) CPU E5-2678 v3 @2.50GHz desktop and 4 TITAN RTX GPUs (24GB memory).

2) Baselines: Since no other methods are designed targeting at CAPER setting, we use the CAPE methods NPCS, NAOCS and A-NCSH in our dataset, where the A-NCSH requires kinematic structure and joint type as known information while our method estimates articulation pose without any priors. In addition, we also propose an ablated version of our method to help compare performance. In this baseline, we adopt direct regression for joint axis and location prediction rather than voting scheme.

3) Metrics: In evaluation with CAPER setting, we report Average Precision (AP) over all the parts of each category, for which the error is less than 5cm and 10cm for translation,

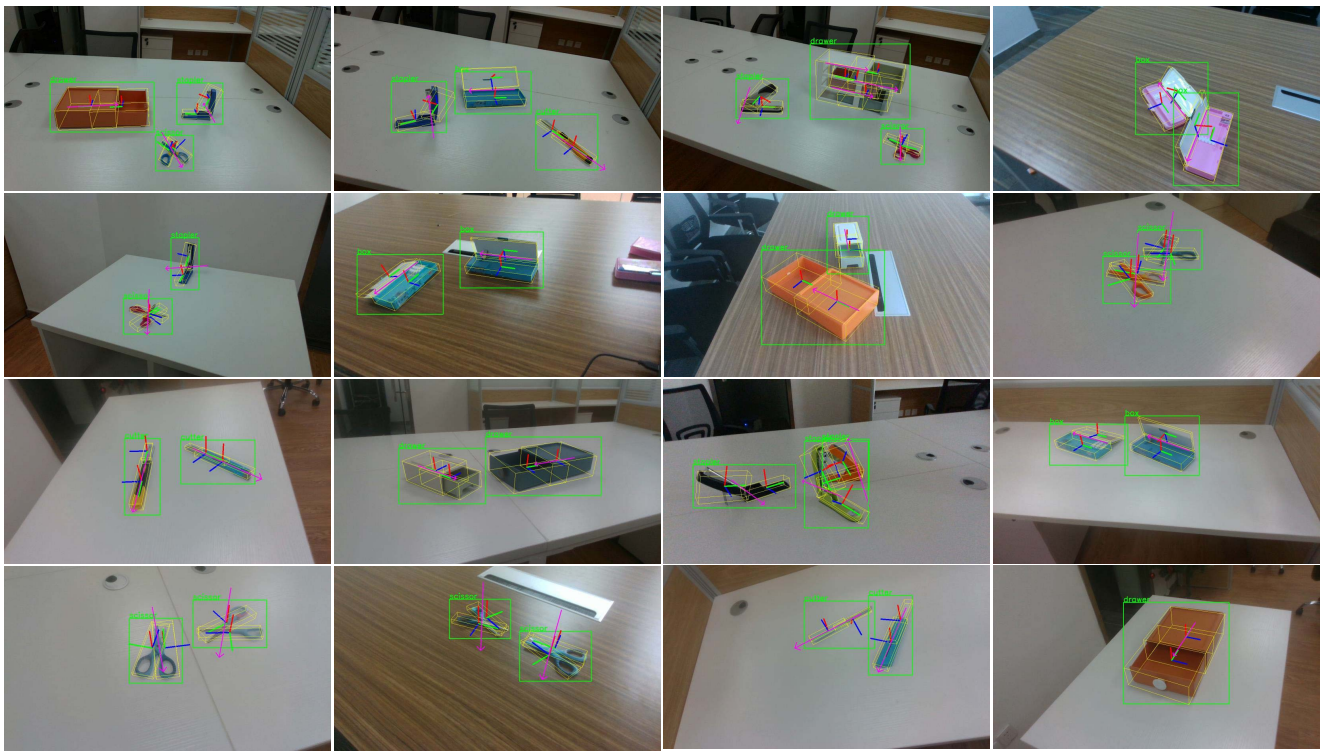

Fig. 9. Qualitative results on ReArtVal dataset.

5° and 10° for rotation. We also average the AP over various error thresholds with $AP_{1°:10°}$ for rotation and $AP_{1cm:10cm}$ for translation. We also use 3D IoU 0.5 and 0.7 as the threshold to report AP and $AP_{0.5:0.7}$. In terms of joint evaluation, we use AP under angle error 5° and 10° of joint axis, and distance error 5cm and 10cm of joint location. When evaluating with CAPE setting, we use pose accuracy at 10°, 10cm, and 3D IoU 0.7 since this setting does not require object detection.

B. Results With CAPER Setting

We report the results of our method training on the ReArtMix training set while testing on the ReArtMix test set and ReArtVal set respectively. In ReArtMix test set, our method achieves mean AP with **50.6%**, **86.7%** and **59.7%** for rotation error 5°, translation error 5cm and 3D IoU@0.5. Specifically, our method performs **77.6%** AP on rotation error 5° for box, which outperforms other categories. On the contrary, AP on translation for the drawer is much worse than others with only **63.3%** for translation error 5cm. This could be explained by the larger average size of drawers in our dataset, which increases the difficulty in estimating their precise locations. See more details in Fig. 7.

When testing on ReArtVal set on Fig. 8, there appears a drop for all the categories compared to ReArtMix evaluation. In real-world data ReArtVal, our method could obtain average **34.4%**, **84.6%** and **53.2%** for rotation error 5°, translation error 5cm and 3D IoU 0.5. In Table II, comparing with A-NCSH that uses ground truth kinematic structure and joint type, we could still obtain a comparable performance on drawer and box with only **1.3%** and **0.5%** AP drop for rotation error 10° as well as **1.8%** and **2.5%** drop for translation error 10cm. This indicates that our method could partly address the pose estimation issue in the real world. In terms of joint prediction performance, Table III shows the performance comparison. As it could be observed, when voting joint location in camera space, our method could obtain a comparable performance (**86.5%** AP for 5cm distance error) with A-NCSH that uses ground truth kinematic structure. Besides, we also achieve **35.7%** and **70.7%** AP for 5° and 10° joint angle error. Qualitative results are illustrated in Fig. 9.

C. Ablation Study

1) Regression Versus Voting Scheme: We compare the results using per point voting strategy with a direct regression strategy in Table IV. On the ReArtVal dataset, the dense prediction per-point/pixel method is consistently better than direct regression, which achieve **3.0%**, **4.2%** and **2.2%** on pose estimation evaluation. Besides, for joint prediction, our method using the voting scheme can also obtain better performance compared with direct regression method, with **1.6%** on joint axis prediction and **1.1%** on joint location prediction. Therefore, we can conclude that dense voting might be a better way in our CAPER task because direct regression largely relies on powerful global part feature representation.

2) Performance of Articulation Parsing: Articulation parsing is one of the core components of our work. To discuss the performance of the APN module, we illustrate the prediction accuracy on Joint Connectedness and Joint Type in Table V. As it can be observed, our APN module could achieve **0.78** joint connectedness accuracy on Drawer, which holds various

TABLE II

POSE ESTIMATION PERFORMANCE COMPARISON ON REARTVAL DATASET. WE COMPARE OUR METHOD WITH NAOCS, NPCS AND A-NCSH, IN WHICH A-NCSH REQUIRES GROUND TRUTH JOINT TYPE AND KINEMATIC STRUCTURE SO IT IS THE UPPER BOUND OF OUR METHOD

Category	Method	Joint Type GT	Kinematic Structure GT	Rotation Error			Translation Error			3D IoU		
				$AP_{1:10}$	AP_5	AP_{10}	$AP_{1:10}$	AP_5	AP_{10}	$AP_{0.5:0.7}$	$AP_{0.5}$	$AP_{0.7}$
Box	NAOCS[1]	-	-	46.7	42.8	82.5	72.6	86.3	90.1	-	-	-
	NPCS[1]	-	-	48.1	44.8	86.1	76.4	87.4	92.6	30.4	50.5	10.2
	A-NCSH[1]	✓	✓	51.3	48.7	89.6	78.7	91.6	95.3	34.8	54.3	14.6
	Ours	-	-	49.6	47.5	89.1	78.1	90.7	92.8	32.6	52.8	12.9
Stapler	NAOCS[1]	-	-	19.3	12.1	42.8	68.6	79.9	89.2	-	-	-
	NPCS[1]	-	-	22.5	18.6	45.7	70.9	84.8	95.3	47.2	66.8	27.0
	A-NCSH[1]	✓	✓	25.2	20.8	50.3	75.2	86.8	96.9	49.6	69.2	28.5
	Ours	-	-	23.9	18.7	48.1	73.6	85.7	95.5	47.8	67.4	26.7
Cutter	NAOCS[1]	-	-	28.6	26.9	50.3	65.4	74.3	87.0	-	-	-
	NPCS[1]	-	-	29.5	28.7	51.8	65.2	75.8	87.4	11.8	26.3	9.7
	A-NCSH[1]	✓	✓	32.7	31.5	56.7	68.5	78.6	90.8	16.7	29.0	13.5
	Ours	-	-	31.3	30.6	54.5	66.8	77.2	87.7	14.6	27.4	10.3
Drawer	NAOCS[1]	-	-	46.9	44.1	81.0	68.2	80.4	91.2	-	-	-
	NPCS[1]	-	-	48.9	42.7	83.3	70.1	83.4	93.9	59.1	78.2	30.3
	A-NCSH[1]	✓	✓	51.8	48.1	89.5	73.4	86.9	95.9	63.8	82.0	34.1
	Ours	-	-	49.8	46.7	88.2	72.0	84.3	94.1	61.6	80.4	33.2
Scissor	NAOCS[1]	-	-	26.3	22.6	58.9	70.5	81.8	88.6	-	-	-
	NPCS[1]	-	-	27.3	22.9	60.4	72.8	82.8	89.2	16.1	36.4	4.6
	A-NCSH[1]	✓	✓	31.2	30.1	64.1	77.8	85.9	92.4	20.5	38.9	7.8
	Ours	-	-	29.8	28.5	63.8	76.5	85.2	92.0	19.3	38.0	5.3
mean	NAOCS[1]	-	-	33.5	29.7	63.1	69.0	80.5	89.2	-	-	-
	NPCS[1]	-	-	35.2	31.5	65.4	71.0	82.8	91.6	32.9	51.6	16.3
	A-NCSH[1]	✓	✓	38.4	35.8	70.0	74.7	85.9	94.2	37.0	54.6	19.7
	Ours	-	-	36.9	34.4	68.7	73.4	84.6	92.4	35.1	53.2	17.6

TABLE III

JOINT PREDICTION PERFORMANCE ON REARTVAL DATASET. NOTE THAT ONLY PRISMATIC JOINTS ARE DEFINED IN DRAWER SO THERE IS ONLY EVALUATION ON JOINT AXIS ERROR FOR DRAWER

Category	Method	Joint Type GT	Kinematic Structure GT	Joint Axis		Joint Location	
				AP_5	AP_{10}	AP_5	AP_{10}
Box	A-NCSH[1]	✓	✓	49.7	91.3	92.5	96.4
	Ours	-	-	49.5	91.4	92.6	96.3
Stapler	A-NCSH[1]	✓	✓	20.2	49.6	88.1	97.2
	Ours	-	-	19.7	49.4	87.5	97.9
Cutter	A-NCSH[1]	✓	✓	32.7	57.4	79.3	89.8
	Ours	-	-	31.8	57.2	79.8	90.1
Drawer	A-NCSH[1]	✓	✓	48.3	90.2	87.3	-
	Ours	-	-	46.5	88.7	85.1	-
Scissor	A-NCSH[1]	✓	✓	31.3	66.9	87.6	94.2
	Ours	-	-	31.1	67.1	87.9	94.6
mean	A-NCSH[1]	✓	✓	36.4	71.1	86.9	94.4
	Ours	-	-	35.7	70.7	86.5	94.7

kinematic structures in this category. This indicates that our method can effectively analyze the structure of unseen objects. On the other hand, the articulation parsing performance also helps our method reach upper bound performance (A-NCSH with ground truth kinematic structure and joint type).

D. Results With CAPE Setting

We also evaluate our method on a public synthetic articulated model repository PartNet-Mobility [3] for the CAPE task. We select 91 models from three categories, including drawer, refrigerator, and trashcan with various kinematic structures. We render these models in Unity to generate synthetic images with these models. For each category, we have 5,000 images for training and 1,000 images for testing.

In Table VI, we report the performance of our method along with some strong baselines, namely NPCS, NAOCS and A-NCSH [1], in which A-NCSH uses joint properties as ground truth so it performs as the upper bound of our method. As it could be seen, our method shows a good performance that achieves average **76.2%**, **27.8%** and **33.5%** accuracy for 10°, 10cm and 3D IoU of 0.7. For the A-NCSH method, we could also obtain a comparable result on drawer and refrigerator, where there are only **2.7%** and **2.4%** accuracy disparity on rotation error 10°. Table VI also shows the joint accuracy comparison of our method in joint angle error 5° and 10°. Compared with A-NCSH that predicts joint property with known kinematic structure, our method also presents passable results with **89.6%**, **70.5%** and **78.3%** in joint angle accuracy. Qualitative results are illustrated in Fig. 10.

E. Results With Instance-Level Robot Arm

For instance-level articulation pose estimation task, we evaluate our method on a robot arm with 7 parts and 6 joints. We capture 63 videos with over 25K frames for Franka Robot Arm with random robot arm poses and split these videos into 51 for training and 12 for testing. To automatically annotate these frames, we calibrate the camera extrinsic matrix for each video and then simulate the robot arm with poses as well as a

TABLE IV

PERFORMANCE COMPARISON ON OUR METHOD USING DIRECT REGRESSION AND VOTING SCHEME FOR JOINT AXIS AND LOCATION PREDICTION. FOR POSE ESTIMATION RELATED RESULTS, WE REPORT AP FOR ROTATION ERROR 5°, TRANSLATION ERROR 5CM AND 3D IOU 0.5. FOR JOINT ESTIMATION, WE REPORT AP FOR JOINT AXIS ERROR 5° AND JOINT LOCATION ERROR 5CM

Category	Method	Pose Estimation			Joint Prediction	
		Rotation	Translation	3D IoU	Axis	Location
Box	ours-Reg	46.4	86.5	51.2	48.2	91.0
	ours-Vote	47.5	90.7	52.8	49.5	92.6
Stapler	ours-Reg	15.2	82.6	65.6	17.5	86.0
	ours-Vote	18.7	85.7	67.4	19.7	87.5
Cutter	ours-Reg	28.5	75.7	24.5	29.8	79.1
	ours-Vote	30.6	77.2	27.4	31.8	79.8
Drawer	ours-Reg	43.8	81.5	77.0	45.2	-
	ours-Vote	46.7	84.3	80.4	46.5	-
Scissor	ours-Reg	23.5	84.1	36.2	30.2	87.2
	ours-Vote	28.5	85.2	38.0	31.1	87.9
mean	ours-Reg	31.4	80.4	51.0	34.1	85.4
	ours-Vote	34.4	84.6	53.2	35.7	86.5

TABLE V

PERFORMANCE ON ARTICULATION PARSING. NOTE THAT THE SCISSOR IN OUR DATASET ONLY HAVE TWO RIGID PARTS AND REVOLUTE JOINT SO THE PREDICTION ACCURACY ON JOINT CONNECTEDNESS AND JOINT TYPE IS 1.00

Category	Prediction Accuracy	
	Joint Connectedness	Joint Type
Box	0.76	0.89
Stapler	0.84	1.00
Cutter	1.00	0.85
Drawer	0.78	1.00
Scissor	1.00	1.00

TABLE VI

PERFORMANCE COMPARISON ON PARTNET-MOBILITY DATASET. NOTE THAT WE DO NOT REPORT JOINT ACCURACY FOR NAOCS AND NPCS BECAUSE JOINT INFORMATION IS NOT DEFINED IN THESE TWO METHODS. BESIDES, ONLY PRISMATIC JOINTS ARE DEFINED IN DRAWER SO THERE IS ONLY EVALUATION ON JOINT AXIS ERROR FOR DRAWER

Category	Method	Joint Type GT	Kinematic Structure GT	Pose Accuracy			Joint Accuracy	
				10°	10cm	$3D_{70}$	10°	10cm
Drawer	NAOCS[1]	-	✓	70.6	15.0	-	-	-
	NPCS[1]	-	✓	71.9	36.8	52.4	-	-
	A-NCSH[1]	✓	✓	90.4	42.3	52.5	92.3	-
	Ours	-	-	87.7	40.9	51.2	89.6	-
Refrigerator	NAOCS[1]	-	✓	60.5	7.2	-	-	-
	NPCS[1]	-	✓	64.6	19.0	18.1	-	-
	A-NCSH[1]	✓	✓	69.4	21.2	20.3	73.4	26.5
	Ours	-	-	67.8	19.7	17.6	70.5	25.8
Trashcan	NAOCS[1]	-	✓	65.4	12.4	-	-	-
	NPCS[1]	-	✓	66.5	20.1	33.2	-	-
	A-NCSH[1]	✓	✓	77.9	24.3	33.5	82.1	28.4
	Ours	-	-	73.2	22.9	31.7	78.3	26.8

camera in Unity Engine to capture the full annotations, such as segmentation map, per-part 6D pose and joint property in camera space.

We report the performance of our method on our robot arm dataset in Table VII and Table VIII. As it could be seen, our method can perform well on estimating per-part pose with **98.8%** to **48.4%**, and **99.6%** to **25.5%** in part 1 to part 7 at rotation error 10° and translation error 10cm. In addition, we could also obtain good performance for joint prediction with maximum **94.6%** and **95.7%** at joint angle error 5° and distance error 5cm. In detail, due to the multi-depth structure of the robot arm, our method might not perform well on the end part and end joint, with only **16.4%** and **8.0%** for pose error as well as **41.1%** and **8.4%** for joint error. This indicates that when the kinematic structure deepens, the accumulative error would be more obvious. Qualitative results are shown in Fig. 11.

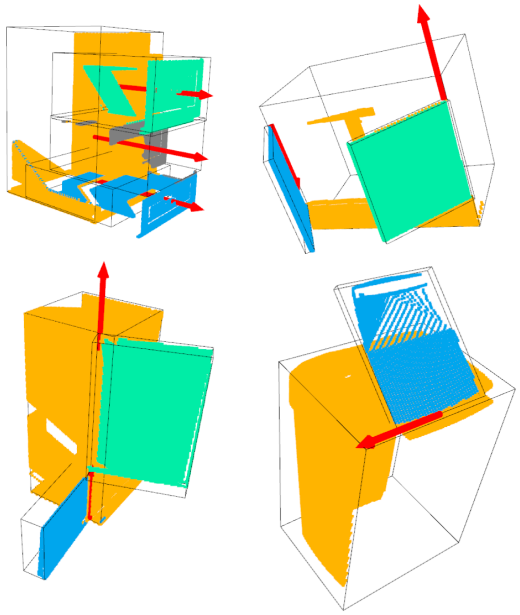

Fig. 10. Qualitative results on PartNet-Mobility.

TABLE VII

POSE ESTIMATION RESULTS ON ROBOT ARM DATASET. WE EVALUATE POSE ACCURACY ON ROTATION ERROR 5° AND 10°, TRANSLATION ERROR 5CM AND 10CM, 3D IOU 0.5 AND 0.7

Part ID	Rotation		Translation		3D IoU	
	5°	10°	5cm	10cm	0.5	0.7
1	95.5	98.8	97.7	99.6	96.8	68.1
2	73.8	88.0	94.5	96.0	75.2	63.6
3	62.0	83.1	79.9	90.1	68.1	59.8
4	59.8	82.5	76.5	89.8	78.7	64.7
5	45.3	80.1	43.1	78.0	65.7	51.4
6	36.2	73.3	48.0	83.4	73.6	59.5
7	16.4	48.4	8.0	25.5	67.7	43.9
mean	55.6	79.2	63.9	80.3	75.1	58.7

TABLE VIII

JOINT PREDICTION RESULTS ON ROBOT ARM DATASET. WE EVALUATE JOINT ACCURACY ON ANGLE ERROR 5° AND 10°, DISTANCE ERROR 5CM AND 10CM. NOTE THAT FOR FRANKA ROBOT ARM, ADJACENT PARTS ARE LINKED BY ONE JOINT SO THERE ARE 6 JOINTS IN TOTAL

Joint ID	Axis		Location	
	5°	10°	5cm	10cm
1	94.6	96.1	95.7	99.6
2	67.6	88.7	88.9	94.1
3	69.3	86.8	88.1	91.3
4	53.9	83.4	78.2	93.5
5	61.2	86.6	85.1	94.2
6	41.1	75.0	8.4	21.5
mean	64.6	86.1	74.1	82.4

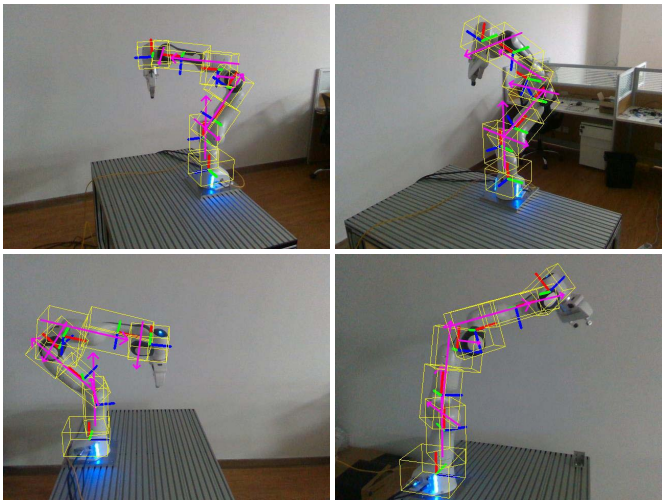

Fig. 11. Qualitative results on Robot Arm. There are 7 parts and 6 joints in our Robot Arm dataset.

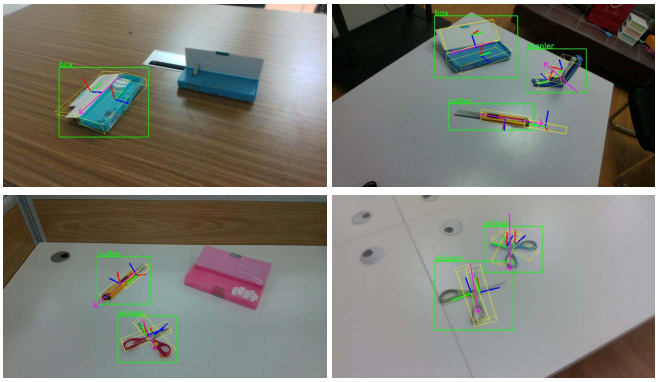

Fig. 12. Failure cases of our method on ReArtVal dataset.

F. Failure Cases and Limitations

The failure cases of our method on ReArtVal are demonstrated in Fig. 12. We summarize the failure of our method on CAPER task into the following reasons: (1) Detection missing. There exist a domain gap between mixed reality data and real-world data that influence detection accuracy. (2) Quality of depth image. The depth camera holds its limitation on an inaccurate depth map, especially when the scissor or cutter lays flat on the table.

VII. Conclusion

In this paper, we extend the CAPE task and formulate the CAPER problem for real-world articulation pose estimation. Accompanying the task setting, we provide a full package of solutions including the FAOM-SAMERT pipeline to semi-automatically build the dataset for the CAPER, the effective framework that could deal with various kinematic structures, and multiple-instance occurrence issues. We hope the proposed CAPER task can help the researchers to rethink the CAPE task setting, and the proposed dataset generation pipeline and learning framework can serve as a strong baseline for future research.

References

[1] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3706–3715.

[2] L. Yi, H. Huang, D. Liu, E. Kalogerakis, H. Su, and L. Guibas, "Deep part induction from articulated object pairs," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–15, Jan. 2018.

[3] F. Xiang *et al.*, "SAPIEN: A SimulAted part-based interactive ENvironment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11097–11107.

[4] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu, "Shape2Motion: Joint analysis of motion parts and attributes from 3D shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8876–8884.

[5] R. Martín-Martín, C. Eppner, and O. Brock, "The RBO dataset of articulated objects and interactions," *Int. J. Robot. Res.*, vol. 38, no. 9, pp. 1013–1019, Aug. 2019.

[6] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6D object pose and size estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2642–2651.

[7] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3828–3836.

[8] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 292–301.

[9] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1521–1529.

[10] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 536–551.

[11] Z. Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6DOF pose estimation for textureless objects," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2441–2448.

[12] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2011–2018.

[13] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *Int. J. Comput. Vis.*, vol. 66, no. 3, pp. 231–259, 2006.

[14] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim, "Siamese regression networks with efficient mid-level feature extraction for 3D object pose estimation," 2016, *arXiv:1607.02257*.

[15] C. Wang *et al.*, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3343–3352.

[16] F. Manhardt *et al.*, "CPS++: Improving class-level 6D pose and shape estimation from monocular images with self-supervised learning," 2020, *arXiv:2003.05848*.

[17] C. Wang *et al.*, "6-PACK: Category-level 6D pose tracker with anchor-based keypoints," 2019, *arXiv:1910.10750*.

[18] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme, "Active articulation model estimation through interactive perception," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 3305–3312.

[19] Q. Liu, W. Qiu, W. Wang, G. D. Hager, and A. L. Yuille, "Nothing but geometric constraints: A model-free method for articulated object pose estimation," 2020, *arXiv:2012.00088*.

[20] L. Yi, H. Huang, D. Liu, E. Kalogerakis, H. Su, and L. Guibas, "Deep part induction from articulated object pairs," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–15, Jan. 2019.

[21] J. Papon and M. Schoeler, "Semantic pose using deep networks trained on synthetic RGB-D," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 774–782.

[22] W. Meeussen, J. Hsu, and R. Diankov, "Unified robot description format (URDF)," Willow Garage, Menlo Park, CA, USA, Tech. Rep., 2009. [Online]. Available: https://wiki.ros.org/urdf

[23] A. X. Chang *et al.*, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[24] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015, pp. 510–517.
[25] S. Hinterstoisser *et al.*, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 858–865.
[26] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, "PlaneRCNN: 3D plane detection and reconstruction from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4450–4459.
[27] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake, "Label fusion: A pipeline for generating ground truth labels for real RGBD data of cluttered scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.
[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[29] J. Borrego, A. Dehban, R. Figueiredo, P. Moreno, A. Bernardino, and J. Santos-Victor, "Applying domain randomization to synthetic data for object category detection," 2018, *arXiv:1807.09834*.
[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++ deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.
[31] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, Apr. 1991.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

Liu Liu received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015, the M.S. degree from The University of Manchester, Manchester, U.K., in 2016, and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2020. He is currently a Postdoctoral Researcher with Shanghai Jiao Tong University. His research interests include computer vision, deep learning, and embodied AI.

Han Xue received the B.S. degree from Shanghai Jiao Tong University, China, in 2021, where he is currently pursuing the M.S. degree with the Computer Science Department. His research interests include computer vision and robotics.

Wenqiang Xu received the B.S. degree from the University of Science and Technology of China in 2015 and the M.S. degree from Shanghai Jiao Tong University, China, in 2019, where he is currently pursuing the Ph.D. degree with the School of Electronic Information and Electrical Engineering. His research interests include computer vision and robotics.

Haoyuan Fu received the B.S. degree from Shanghai Jiao Tong University, China, in 2017, where he is currently pursuing the M.S. degree with the Computer Science Department, SEIEE. His research interests include robotics, simulation environment, and computer vision.

Cewu Lu (Senior Member, IEEE) received the B.S. degree from the Chongqing University of Posts and Telecommunications, China, in 2006, the M.S. degree from the Institute of Electronics, Chinese Academy of Sciences, China, in 2009, and the Ph.D. degree from The Chinese University of Hong Kong. He was a Research Fellow with the Stanford University AI Laboratory, USA, from 2014 to 2015. He is currently a Professor with Shanghai Jiao Tong University. His research interests include robotics AI and computer vision. He served as the Associate Editor for IROS 2021, the Area Chair for CVPR 2020/ICCV 2021, a Senior Program Committee Member for AAAI 2020/2021, the Program Chair for CVM2018, and a Reviewer in AI for the journal *Nature and Science*. He was also selected as the MIT TR35—MIT Technology Review, 35 Innovators under 35, China.