AKB-48: A Real-World Articulated Object Knowledge Base

Liu Liu Wenqiang Xu Haoyuan Fu Sucheng Qian Qiaojun Yu Yang Han Cewu Lu†
Shanghai Jiao Tong University
{liuliu1993, vinjohn, simon-fuhaoyuan, qiansucheng, yqjllxs, lucewu}@sjtu.edu.cn
tycoer@outlook.com

Figure 1. AKB-48 consists of 2,037 articulated object models of 48 categories scanned from the real world. The objects are annotated with ArtiKG, and can support a full task spectrum from computer vision to robotics manipulation.

Abstract

*Human life is populated with articulated objects. A comprehensive understanding of articulated objects, namely appearance, structure, physical property, and semantics, will benefit many research communities. As current articulated object understanding solutions are usually based on synthetic object dataset with CAD models without physics properties, which prevent satisfied generalization from simulation to real-world applications in visual and robotics tasks. To bridge the gap, we present AKB-48: a large-scale **A**rticulated object **K**nowledge **B**ase which consists of 2,037 real-world 3D articulated object models of 48 categories. Each object is described by a knowledge graph ArtiKG. To build the AKB-48, we present a fast articulation knowledge modeling (FArM) pipeline, which can fulfill the ArtiKG for an articulated object within 10-15 minutes, and largely reduce the cost for object modeling in the real world. Using our dataset, we propose AKBNet, an integral pipeline for Category-level Visual Articulation Manipulation (C-VAM) task, in which we benchmark three sub-tasks, namely pose estimation, object reconstruction and manipulation. Dataset, codes, and models are publicly available at https://liuliu66.github.io/AKB-48.*

1. Introduction

Articulated objects, composed of more than one rigid part connected by joints allowing rotational or translational movements in 3D space, are pervasive in our daily life. Knowledge about the articulated objects can be beneficial to many research communities, such as computer vision, robotics and embodied AI. Thus, many articulated object datasets have been proposed to facilitate the research, such as PartNet-Mobility [31], ReArt-48 [17], RBO [20]. However, these datasets generally focus more on the structural information (*e.g.* part segmentation, kinematic structure), but pay less attention to the appearance (*e.g.* texture, fine geometry), the physics properties (*e.g.* per-part mass, iner-

†Cewu Lu is the corresponding author. He is the member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, and Shanghai Qi Zhi Institute, China.

tial, material and friction) and semantics (*e.g*. category, affordance). While some important tasks heavily rely on these information such as object detection (*texture*) [2], 3D reconstruction (*fine geometry*) [19], object manipulation (*physical property*) [5], and so on, the lacking of such object knowledge in these datasets can prevent satisfied generalization for the learning models.

To boost the research on articulated objects, in this paper, we present **AKB-48**: a large-scale real-world **A**rticulated **K**nowledge **B**ase which includes 48 categories, 2,037 instances. For each instance, the object model is scanned from the real counterpart and refined manually (Sec. 3.2), and the object knowledge is organized to a graph, named **Art**iculation **K**nowledge **G**raph (**ArtiKG**), which contains the detailed annotations of different kinds of object attributes and properties (Sec. 3.1). To make the scanning and annotation process feasible for large datasets, we present a **F**ast **Ar**ticulation Knowledge **M**odeling (**FArM**) pipeline (Sec. 3.3). In detail, we develop an object recording system with 3D sensors and turntables, a GUI that integrates structural and semantic annotations, and standard real-world experiments for physical property annotation (Fig. 3). In this way, we can save a large amount of money and time budget for modeling real-world articulated objects (∼$3 to buy, 10-15min to annotate per object). A thorough comparison between the CAD modeling and reverse scanning can be referred to Sec. 3.2. To summarize, our pipeline can save 33 folds on the money budget and 5 folds on the time budget.

To utilize the AKB-48 for research, we propose **AKBNet**, an integral pipeline for **C**ategory-level **V**isual **A**rticulation **M**anipulation (**C-VAM**) task. To address C-VAM problem, the vision system AKBNet should be able to estimate the object pose, reconstruct the object geometry and learn the policy for manipulation at category level. Thus, it consists of three perception sub-modules:

- **Pose Module** for *Category-level Articulated Object Pose Estimation*. This module aims to estimate the per-part 6D pose of an unseen articulated object in one category. However, prior researches generally study on *kinematic category*, that is objects of a category are defined to have the same kinematic structure. Our pose module extends the concept of "category" to *semantic category*, in which the category is defined by the semantics and different kinematic structures are allowed. (Sec. 4.1)

- **Shape Module** for *Articulated Object Reconstruction*. After the pose is obtained, along with the shape code encoding from input images, we can reconstruct the shape for each part [25]. Full geometry is critical for manipulation to determine where to interact with. (Sec. 4.2)

- **Manipulation Module** for *Articulated Object Manipulation*. Once we obtain the articulation information (*e.g*. part segments, per-part pose, joint properties, full mesh, etc.) through perception, we can learn the interaction policy over the observations. We benchmark manipulation tasks with opening and pulling that are corresponding to revolute and prismatic joint respectively. (Sec. 4.3)

To evaluate the AKBNet, we report the results individually and systematically. For individual evaluation of each module, we assume the input to the module is the ground truth of the last module, while for systematical evaluation, the input is the output of the last module. Apparently, we cannot benchmark all the tasks which can be supported by the proposed AKB-48. We hope it could serve as a good platform for future articulation research in computer vision and robotics community.

Our contributions can be summarized in three folds:

- We introduce AKB-48, containing 2,037 articulated models across 48 categories, in which we adopt a multi-modal knowledge graph ArtiKG to organize the rich annotations. It can contribute to close the gap between the current vision and embodied AI researches. To the best of our knowledge, it is the first large-scale articulation dataset with rich annotations collected from the real world.

- We propose a fast articulation knowledge object modeling pipeline, FArM, which makes it much easier to collect articulated objects from the real world. Our pipeline greatly eases the cost on time and money when building real-world 3D model datasets.

- We propose an integral pipeline AKBNet for the integral category-level visual articulation manipulation (C-VAM) task. Experiments show our approach is effective both individually and systematically in the real world.

2. Related Work

3D Model Repositories and Datasets. An unavoidable challenge for analyzing 3D objects, especially for articulated objects, is the lack of large-scale training data with sufficient 3D models and full annotations. To the best of our knowledge, current 3D model repositories prefer to collect CAD models by searching from the Internet such as Trimble 3D Warehouse and Onshape [14]. ShapeNet [4] collects approximately 3 million shapes from online model repositories and categorizes them based on WordNet [22] taxonomy. But although ShapeNet contains many articulated categories, the models of ShapeNet can only be considered as rigid shapes since they do not define parts within

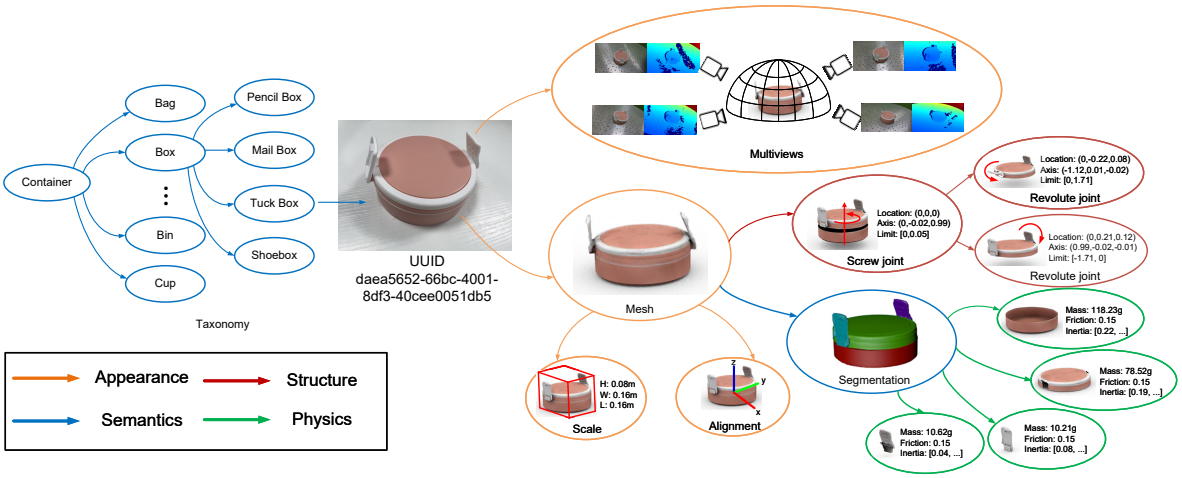

Figure 2. The Articulation Knowledge Graph (ArtiKG) defined in AKB-48 dataset. In ArtiKG, we annotate four types of knowledge: Appearance, Structure, Semantics and Physical Property. The values are rounded up to percentile in this figure for presentation.

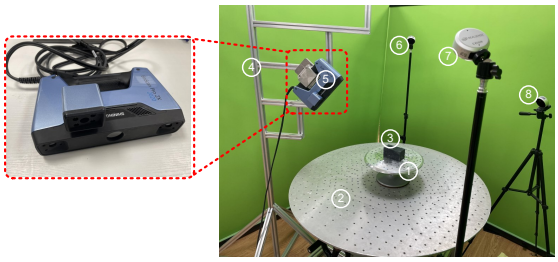

Figure 3. The task-specific model acquisition equipment. (a) 1 is a Rotating turntable for objects with multiple scales. 2 is a tracking marker. 3 is a light-absorbing item. 4 is a lift bracket. 5 is the Shining 3D scanner. 6-8 are the realsense L515 cameras for capturing multiviews of objects.

them. To deal with this problem, Mo et al. [24] first present a large-scale dataset PartNet that annotates hierarchical part semantic segmentation based on a subset of ShapeNet [4]. One critical problem in PartNet is that it pays much attention to labeling each semantic part but ignores the kinematics structures. To solve this issue, PartNet-Mobility [31] and Shape2Motion [30] further annotate joint properties on the shapes, which target at articulation research.

These datasets mostly follow the model construction paradigm from ShapeNet: collecting CAD models from the Internet and providing specific annotations for different tasks. This allows the early works (ShapeNet [4], ABC dataset [14], etc.) to quickly build large-scale object model bases. However, when the task is required to investigate new categories or kinematic structures, artists need to manually build proper CAD models from scratch, which is very time-consuming and laborious. On the other hand, current real-world researches focus on instance-level tasks so they tend to build small-scale model datasets such as YCB [3] and RBO [20]. Therefore, the data volume makes it hard to be adopted in our category-level articulation tasks, which requires generalization capacity among different instances. In this paper, we present AKB-48 as the first large-scale real-world base for articulation analysis.

Articulation-related Tasks. Articulated objects have been investigated for decades in both vision and robotics communities but hold different emphases. In vision tasks, current works tend to solve category-level object recognition, segmentation or pose estimation that focus on generalization among objects. Yi et al. [32] take a pair of unsegmented shape representations as input to predict part segmentation and deformation. For tackling with unseen objects, Li et al. [16] follow the pose estimation setting and propose a normalized coordinate space to estimate 6D pose and joint state for articulated objects. In terms of joint-centered perception tasks, several works attempt to mine joint configurations of articulated objects [11, 18, 33]. To investigate manipulation points for articulated objects from visual input, Mo et al. attempt to define six types of action primitives and predict interactions [23]. In terms of robotics community, researchers usually solve interaction or manipulation tasks to achieve articulation inference such as robot interactive perception [12], feedback by visual observation [9] and task integration [21]. Besides, some works attempt to bridge the gap between vision and manipulation but still suffer from the small-scale issue. Therefore, we propose AKBNet to deal with category-level articulation tasks.

3. Articulation Knowledge Base, AKB-48

When constructing the knowledge base, three instant questions should be answered: (1) What kinds of knowledge should we annotate on the object? (2) What objects should we annotate, those from the real or the simulated world? (3) How to annotate the object knowledge efficiently? To answer these questions, we describe the ArtiKG in Sec. 3.1, make a thorough discussion on the object selection in Sec. 3.2, and finally propose the FArM pipeline in Sec. 3.3 and provide analysis (diversity, difficulty) about the dataset in Sec. 3.4.

3.1. Articulated Object Knowledge Graph, ArtiKG

Different tasks require different kinds of object knowledge, to unify the annotation representation, we organize it into a multi-modal knowledge graph, named ArtiKG. The ArtiKG consists of four major parts, namely appearance, structure, physics, and semantics. The details are described in the following and visualized in Fig. 2.

Appearance. For each instance, we store its shape with mesh data structure along with the textures. When scanning the object from the real world, we also collect the multi-view RGB-D snapshots of the object.

Structure. The key difference between the articulated object and the rigid object is the kinematic structure. The articulated object has concepts like joint and part, which are not meaningful for the rigid object. For each joint, we annotate the joint type, parameters, and movement limits. For each part, we segment each kinematic part.

Semantics. After the basic geometric and structural information is annotated, we begin to assign the semantic information to the object in a coarse-to-fine process. We give a uuid to each instance. Then we assign the category and the corresponding taxonomy to the object according to WordNet [22]. We also label the semantic part. Though we already annotate the kinematic part, it is not quite the same as the semantic part. Take a mug with a handle, for example, the handle is not attached to the mug body through a joint, thus it is not a kinematic part, but it is a semantic part as it indicates where the human normally grabs the mug.

Physical property. Real objects exist in the physical world and typically have physical properties, which are important for accurate simulation and real-world manipulation & interaction on articulated objects. Thus, we store physical attribute annotations for our models, involving per-part mass, Per-part inertial, material and surface friction.

Discussion. In this section, we only describe the object knowledge thet should take human's effort to annotate, for those which can be calculated through algorithms or trivially inferred like surface normal, collision mesh/simplified mesh, intrinsic dimensions, are not discussed. Besides, as the annotation information is modular organized, it is convenient for new attributes to be added to the ArtiKG. Besides, though the ArtiKG is designed for articulated objects, it can also be trivially extended to rigid, and flexible objects.

3.2. Object Selection: Real-world Scanning v.s CAD Modeling

The choice between real-world scanning and CAD modeling are considered from two perspectives, namely annotation accuracy, cost on time and money.

Annotation Accuracy. According to the content of the ArtiKG, we can see objects from the real world have multiple advantages over the CAD models, such as appearance and physical property. But admittedly, the CAD model can model inner structures such as the GUNdam or the transformer, while scanning techniques focus more on the surface. Since such objects with inner structures that cannot be easily disassembled posit challenges for both artists and the scanners, we would like to update these objects when techniques are more ready. Fortunately, most daily objects can be disassembled, so the scanning techniques can properly handle them.

Cost on Time and Money. As discussed earlier, ShapeNet-like model collection paradigm is limited to large time and money cost of artists' manual CAD model building when investigating new categories or kinematic structures. On the other hand, many daily articulated objects are cheap in reality and can be scanned by a layman. We compare the average money and time budget in Table 1. For CAD modeling, it is estimated from outsourcing services in Taobao website[1]. From our survey, most artists spend more than 2 hours (over 120 minutes) to model an articulated object and the labor cost is averagely over 100 dollars for one.

	CAD modeling	Real-world Scanning
Time (min)	>120	20
Money ($)	>100	3

Table 1. Budget comparison between our real-world scanning and CAD modeling for articulated objects.

To note, we are aware that many important articulated objects in the real world are rather expensive like laptop, microwave oven, doors etc. In such cases, we either collect only the ones we can collect from the homes without rebuying, or buy one to measure the basic information and propagate to the existing simulated objects like in PartNet-Mobility [31]. For these objects, the ArtiKG is labeled as ArtiKG-sim.

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

Dataset	Appearance			Structure		Semantics		Physics		
	Num	AV	AT	Part	Joint	ST	PS	PM	PI	PF
Synthetic Model Dataset										
ShapeNet [4]	>50K	<2K	<5K	-	-	✓	-	-	-	-
PartNet [24]	>20K	<2K	<5K	✓	-	✓	-	-	-	-
Shape2Motion [30]	2K	<0.5K	<1K	✓	✓	-	-	-	-	-
PartNet-Mobility [31]	2K	<0.5K	<1K	✓	✓	✓	✓	-	-	-
Real-World Model Dataset										
YCB [3]	21	∼40K	∼90K	-	-	-	-	-	-	-
LineMod [10]	15	∼19K	∼39K	-	-	-	-	-	-	-
RBO [20]	14	∼5K	∼10K	✓	✓	-	-	-	-	-
AKB-48(Ours)	2,037	∼56K	∼110K	✓	✓	✓	✓	✓	✓	✓

Table 2. Comparison with other popular model datasets. Our AKB-48 dataset provides four types of information for rich annotations in our ArtiKG: Appearance, Structure, Semantics and Physics. **AV:** Average number of vertices. **AT:** Average number of triangles. **ST:** Semantic Taxonomy. **PS:** Per-part Semantic label. **PM:** Per-part Mass. **PI:** Per-part Inertia Moment. **PF:** Per-part Friction.

3.3. Fast Articulation Knowledge Modeling (FArM) Pipeline

Once we determine what to annotate and what object to be annotated, the remaining problem is how to make the annotation process affordable.

3.3.1 Model Acquisition Equipment.

To efficiently collect real-world articulated models, we setup a recording system, whose configuration is illustrated in Fig. 3. This apparatus is developed with three components: EinScan Pro 2020 for scanning[2], Intel RealSense D435 for RGB-D multi-view snapshot, multi-scale rotating turntables and lift bracket. In our setup, each object can be scanned within 5 minutes.

3.3.2 Articulation Modeling

After the model acquisition, we develop an articulated object modeling interface with 3D GUI for annotation guidance. Specifically, our modeling workflow split the whole process into three sub-processes:

Object Alignment. This process requires the annotator to align the scanned articulated object from camera space into canonical space which is shared within a category. To assist the alignment, we define several primitive shapes such as cube, sphere and cylinder with predefined axis, which are used to fit the targeted object.

Part Segmentation. Different from synthetic models from the Internet that often include original mesh subgroups and part information, real-world scanned models require manual segmentation for each rigid part. In our interface, we provide a mesh cutting method with multi-view observation. The annotators draw boundary polygons on the aligned watertight surface and the interface could automatically split the mesh into multiple smaller sub-components. To note, if the parts can be disassembled in the real world, we just scan each part and assemble them into an integral model.

Joint Annotation. In contrast to other object modeling pipelines, articulated objects require joint annotation that links two rigid segmented parts and describes the kinematic structure as a tree. Our interface provides an inspector window that allows the annotator to reorganize the parts into a tree structure. Then, the annotators could add joint information to each link and annotate 6D vector (3 for joint location and 3 for joint axis) in a 3D view that contains parent and child parts. To ensure the correctness of joint annotation, we provide an animation that demonstrates the motion under current joint information and the annotators could further refine the annotation.

3.3.3 Physics Annotation

Real-world articulated objects exist in the physical world and have physical properties. To enable our AKB-48 in real-world robotic manipulation & interaction tasks, we also annotate physical attribute annotations for each part of the articulated object.

Per-part Mass. We record each rigid part's weight in grams. For those objects that are inseparable on several parts, we adopt the drainage method [6] to measure volume for these parts and compute the weight by their densities according to the materials.

Per-part Inertia Moment. It is hard to obtain per-part inertia moment in the real world since scanned articulated models might contain hundreds of thousands of triangles, which is in a very complicated structure. In our method, we simplify these models with finite primitive shapes, such as cuboid

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

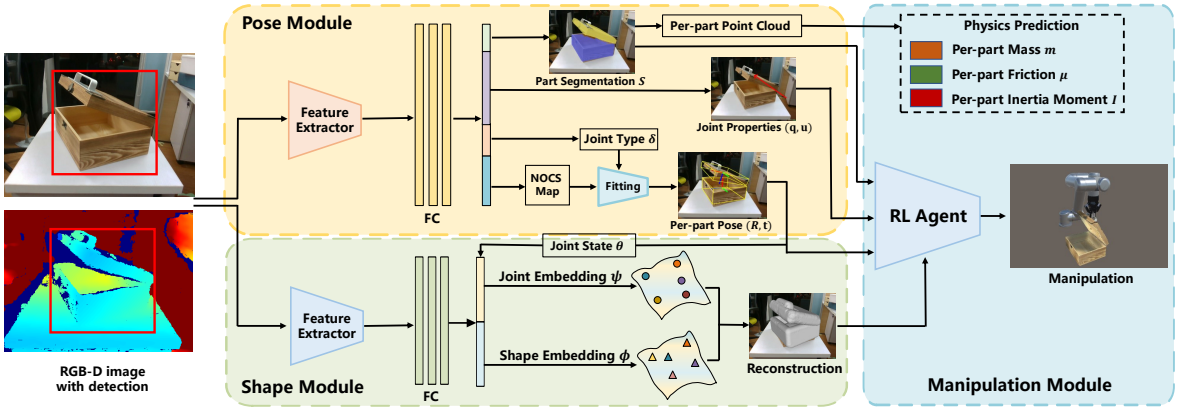

Figure 4. The overall pipeline of AKBNet. The input of AKBNet is a single RGB-D image with a detected box, and there are three components conducted: (1) Pose module for predicting per-part segmentation, 6D pose, joint type as well as joint properties. (2) Shape module for generating full mesh of the articulated object with current joint state. (3) Manipulation module for enabling the RL agent (UR5 Robot Arm with a Robotiq 85 gripper) to manipulate the object, and also predicting per-part physics information.

and cone, and then compute the inertial moment in simulation based on the combination of these primitive shapes.

Per-part Material and Friction. We also annotate the surface material and related parameters. For example, a transparent material will be annotated with the index of refraction, and normal materials will be annotated with friction coefficients. These are obtained by searching Machinery's Handbook [26].

3.4. Dataset Analysis

Object Categories. To build AKB-48 dataset, we take the following requirements into consideration: (1) *Commonality*. We require our AKB-48 could cover most of the articulated object categories in the common daily scenes, such as kitchen, bedroom and office room. (2) *Variety*. We consider the objects with a wide variety of shapes, deformability, texture and kinematic structure for one category. (3) *Usage*. The chosen objects should contain various functionalities on usage. Besides, the ability to complete manipulation performance is prioritized.

Statistics. We first compare AKB-48 with some other popular datasets in Table 2. As it is shown, our object models cover full features for real-world articulated object analysis. Specifically, compared to the synthetic model repository, we hold a much finer surface with average of around 126K triangles and real textures while synthetic models only contain thousands of triangles and synthetic textures. In terms of annotation, we provide part and joint annotations that are enough for visual articulation tasks. Furthermore, we also annotate physical information for each model that is never considered in both synthetic and real-world model repositories before. We believe the rich annotations could promote further development in articulation research. As for the model number, we have a comparable number of objects in comparison with the current largest articulated object datasets PartNet-Mobility [31], yet it comprises only CAD models. More statistics such as category specification and intra-category variety can be referred to supplementary materials.

4. AKBNet

In this section, we describe the AKBNet, an integral pipeline for C-VAM problem. In AKBNet, the input is a single RGB-D image with detected 2D bounding boxes. We build three sub-modules in AKBNet that aims to estimate per-part 6D pose (Sec. 4.1), reconstruct full geometry of articulated object (Sec. 4.2) and reason the interaction policy through the perception (Sec. 4.3). The overall pipeline of AKBNet is illustrated in Fig. 4.

4.1. Pose Module

Given an image with a detected 2D bounding box, we can obtain the partial point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$. Firstly, the input $\mathcal{P}$ is processed by a Pointnet++ [28] for feature extraction, and we build two branches at the end for predicting per-point segmentation S and part-level Normalized Object Coordinate Space [16] (NOCS) map $\mathcal{P}' \in \mathbb{R}^{N\times 3}$. To solve the unknown kinematic structure and joint type issues, we introduce three extra branches on the feature extractor to classify the joint type δ on its corresponding part k, and also to predict joint property including joint location $\mathbf{q}_i$ and joint axis $\mathbf{u}_i$. Finally, we apply the voting scheme to obtain the final joint property $\mathbf{q} \in \mathbb{R}^3$ and $\mathbf{u} \in \mathbb{R}^3$. We use cross-entropy loss for part segmentation $\mathcal{L}_{seg}$ and joint type classification $\mathcal{L}_{type}$, L2 loss for NOCS map $\mathcal{L}_{nocs}$, joint location $\mathcal{L}_{loc}$ and joint axis $\mathcal{L}_{ax}$ prediction. Taking all the loss functions into consideration, the overall loss $\mathcal{L}_{pos}$ for pose

module is:

$$\begin{aligned}\mathcal{L}_{pos} &= \lambda_{seg}\mathcal{L}_{seg} + \lambda_{nocs}\mathcal{L}_{nocs} \\ &= +\lambda_{loc}\mathcal{L}_{loc} + \lambda_{ax}\mathcal{L}_{ax} + \lambda_{type}\mathcal{L}_{type}\end{aligned} \quad (1)$$

Finally, we follow the pose optimization algorithm with kinematic constrains [16] to recover the 6D pose $\{R, \mathbf{t}\}$ for each rigid part. R denotes rotation $R \in SO(3)$ and $\mathbf{t}$ denotes translation $\mathbf{t} \in \mathbb{R}^3$.

4.2. Shape Module

Given a partial point cloud $\mathcal{P}$, the shape module aims to re-build the full geometry $\mathcal{M}_\theta$ with joint state θ. Followed by A-SDF [25], we build a feature extractor process the concatenated partial point cloud $\mathcal{P}$ and Gaussian initialized shape embedding ϕ as well as joint embedding ψ, in which ϕ indicates the shape information of the full articulated object and ψ indicates the joint state information that is shared across the same instance. we use SDF values [27] d_i as supervision and L1 loss for training the shape module F_{sha}:

$$\mathcal{L}_{sha} = \lambda_{sha}\frac{1}{N}\sum_{i=1}^{N}\|F_{sha}(p_i, \phi, \psi) - d_i\| + \lambda_\phi\|\phi\|_2 \quad (2)$$

During inference, based on the predicted shape embedding ϕ and joint embedding ψ, we follow Mu's algorithm [27] to reconstruct the full mesh $\mathcal{M}_\theta$.

4.3. Manipulation Module

The manipulation module performs two tasks: opening and pulling that are corresponding to the revolute and prismatic joints in articulation respectively. To achieve these tasks, we train two Reinforcement Learning (RL) agents (UR5 Robot Arm with a Robotiq 85 gripper) these tasks. We provide two **State Representations:** (1) object state, consisting of 6D pose $\{R, \mathbf{t}\}$, joint location $\mathbf{q}$, axis $\mathbf{u}$, full geometry $\mathcal{M}_\theta$ under current joint state θ. (2) agent state, consisting of the gripper's pose $\{R_g, \mathbf{t}_g\}$ and the gripper's width w_g. We assume that the agent can access all the information about itself so the agent state is ground truth in our method. The **Actions** include the agent's end-effector's 3D translation and the opening width of the gripper. The **Rewards** are rotation angle along the joint axis of the target part for revolute joint and translation distance of that for prismatic joint. The RL agent is trained by two popular RL baselines: Truncated Quantile Critics (TQC) [15] and Soft Actor-Critic (SAC) [8] with Hindsight Experience Replay (HER) [1] algorithm.

We also perform physics prediction in our AKBNet. Specifically, the input is a feature vector of point cloud $\mathcal{P}^k$ for kth part. We train a 3-layer MLP and build three parallel branches to predict per-part mass m^k, friction μ^k and inertia moment I^k. We use L2 loss for training the physics prediction submodule. Please refer to supplementary materials for more details.

5. Experiments

5.1. Experimental Setup

Dataset. For the pose module and shape module, we generate 100K RGB-D images with AKB-48 models for training AKBNet using SAMERT data generation scheme [17] with scenes from NOCS [29]. And we also capture 10K real-world images, in which 5K are used for fine-tuning the model and the other 5K is test set. For manipulation module, we select 68 and 32 instances for training and testing the RL agent, in which the former is used for opening task and the latter is for pulling task. During training, we use different instances at every episode.

Implementation Details. When training pose module and shape module, we use Adam optimizer with initial learning rate 0.001. Batch size is 16. The total training epochs are 50 and 100 for training these two modules. The detailed hyper-parameters are: $\lambda_{seg} = 1$, $\lambda_{nocs} = 10$, $\lambda_{loc} = 1$, $\lambda_{ax} = 0.5$, $\lambda_{type} = 1$, $\lambda_{sha} = 1$, $\lambda_\phi = 0.0001$. For the manipulation module, the hyper-parameters are: batch size is 512, learning rate is 0.001, replay buffer size is 100K, soft update coefficient is 0.05, discount factor is 0.95. We use RFUniverse [7] as the environment to train the RL agent. For more details, please refer to the supplementary materials.

Metrics. We adopt the following metrics to measure the AKBNet performance. For the pose module, We report three part-based metrics: rotation error measured in degrees, translation error measured in meters and 3D IoU for each part. We also report the joint-based metrics: angle error of joint axis measured in degrees, location error in line-to-line distance measured in meters, joint type classification accuracy (%). For the shape module, we report the average Chamfer-L1 distance [25] for reconstruction evaluation. For the manipulation module, we report success rate (%) as the metric. If the agent can grip the target part and move it through 50% of its motion range, it will be regarded as a success.

5.2. Pose Module Performance

We evaluate NPCS [16], A-NCSH [16] and AKBNet on real-world test set for category-level articulation pose estimation task. For A-NCSH baseline, we use direct regression and classification scheme to predict kinematic structure and joint type. The experimental results are illustrated in Table 3. For pose estimation, we achieve **10.0**, **0.023** and **52.7** on rotation, translation errors and 3D IoU, which

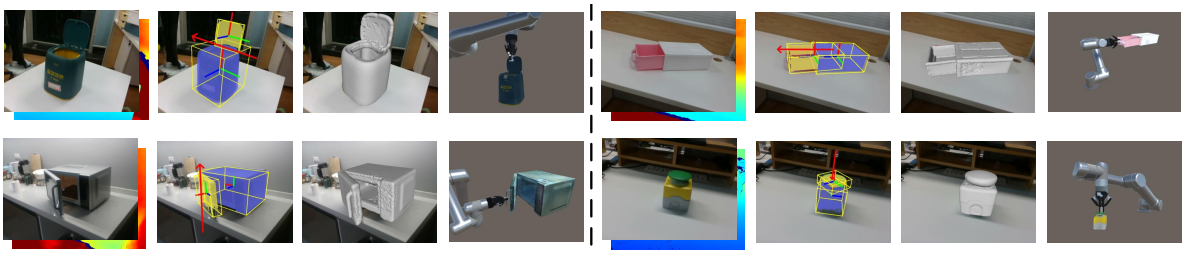

Figure 5. Qualitative results. For one instance, from left to right: input RGB-D image, output of pose module, output of shape module, manipulation demonstration.

are higher than NPCS and A-NCSH. For joint-related evaluation, we can precisely predict joint type for unseen articulated objects with **94.2%** accuracy. Besides, AKBNet achieves **8.7** and **0.019** errors in joint axis and location prediction respectively.

Method	Part-based Metrics		
	rotation↓	translation↓	3D IoU↑
NPCS [16]	12.6	0.038	48.3
A-NCSH* [16]	10.5	0.026	50.8
AKBNet	**10.0**	**0.023**	**52.7**
Method	Joint-based Metrics		
	angle↓	distance↓	type↑
NPCS [16]	-	-	-
A-NCSH* [16]	9.2	0.021	93.8
AKBNet	**8.7**	**0.019**	**94.2**

Table 3. Category-level articulation pose estimation results. ↓ means the lower the better. ↑ means the higher the better. * indicates that A-NCSH is re-implemented with the extra kinematic structure and joint type prediction modules.

5.3. Shape Module Performance

The experimental results of the shape module are illustrated in Table 4. Within ground truth joint state input, the shape module could reconstruct the articulated object with **5.6** Chamfer-L1 distance. On the other hand, we systematically evaluate the shape module given the predicted joint state, which is deduced from predicted the linked two parts' poses from the pose module. The Chamfer-L1 distance is **3.3** higher than that with ground truth joint state, indicating that the predicated poses largely affect reconstruction performance.

Mode	Chamer-L1 Distance
Joint State GT	5.6
Joint State Pre.	8.9

Table 4. Articulated object reconstruction results. Pre. means that we use the predicted joint state from the pose module.

5.4. Manipulation Module Performance

We evaluate opening and pulling tasks on the manipulation module of AKBNet using TQC+HER training algorithm compared with that using SAC+HER. Experimental results are illustrated in Table 5. With ground truth object state, AKBNet could complete opening and pulling manipulation tasks, with **68.6%** and **92.4%** success rate. However, our method might not perform well when the object state is predicted, with only 26.4% and 32.6% success rates. Qualitative results of AKBNet are illustrated in Fig. 5.

Our AKBNet can also predict physics information including per-part mass, friction and inertia moment. These predicted physics can enable force sensing for AKB-48 objects in simulation, which has the potential to realize force controlling. For more details, please refer to supplementary materials.

Method	Mode	Opening	Pulling
AKBNet+SAC [8]+HER [1]	Object State GT	53.8	**92.4**
	Object State Pre.	22.8	28.5
AKBNet+TQC [15]+HER [1]	Object State GT	**68.6**	89.7
	Object State Pre.	**26.4**	**32.6**

Table 5. Success rate (%) on articulated object manipulation task. Pre. means we use predicted object state from the pose and shape modules.

6. Conclusion and Crowd-Sourcing Data-Collection Invitation

In this paper, we present AKB-48, a large-scale articulated object knowledge and benchmark C-VAM problem for dealing with articulation problems. Admittedly, there are a few articulated object categories that might not be collected in AKB-48, although we have covered large enough categories in daily life. In the future, we will release our FArM tool for collecting more articulated objects, and it could also support any scanned shapes such as mobile reconstructor [13]. In future work, we will publish an online articulation model platform and invite crowd-sourcing data-collection to contribute to the articulation research community.

Acknowledgement This work was supported by the National Key R&D Program of China (No. 2021ZD0110700), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Qi Zhi Institute, and SHEITC (2018-RGZN-02046).

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5055–5065, 2017. 7, 8

[2] Joao Borrego, Atabak Dehban, Rui Figueiredo, Plinio Moreno, Alexandre Bernardino, and José Santos-Victor. Applying domain randomization to synthetic data for object category detection. *arXiv preprint arXiv:1807.09834*, 2018. 2

[3] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 3, 5

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 3, 5

[5] Peng Chang and Taşkin Padif. Sim2real2sim: Bridging the gap between simulation and real-world in flexible object manipulation. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 56–62. IEEE, 2020. 2

[6] John D Cutnell and Kenneth W Johnson. *Physics, Volume One: Chapters 1-17*, volume 1. John Wiley & Sons, 2014. 5

[7] Haoyuan Fu, Xu Wenqiang, Xue Han, Yang Huinan, Ye Ruolin, Huang Yongxi, Xue Zhendong, Wang Yanfeng, and Cewu Lu. Rfuniverse. 7

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018. 7, 8

[9] Karol Hausman, Scott Niekum, Sarah Osentoski, and Gaurav S Sukhatme. Active articulation model estimation through interactive perception. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3305–3312. IEEE, 2015. 3

[10] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *IEEE International Conference on Computer Vision*, 2012. 5

[11] Ajinkya Jain, Rudolf Lioutikov, and Scott Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. *arXiv preprint arXiv:2008.10518*, 2020. 3

[12] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *2008 IEEE International Conference on Robotics and Automation*, pages 272–277. IEEE, 2008. 3

[13] Matthew Klingensmith, Ivan Dryanovski, Siddhartha S Srinivasa, and Jizhong Xiao. Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields. In *Robotics: science and systems*, volume 4. Citeseer, 2015. 8

[14] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019. 2, 3

[15] Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, pages 5556–5566. PMLR, 2020. 7, 8

[16] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3706–3715, 2020. 3, 6, 7, 8

[17] Liu Liu, Han Xue, Wenqiang Xu, Haoyuan Fu, and Cewu Lu. Towards real-world category-level articulation pose estimation. *arXiv preprint arXiv:2105.03260*, 2021. 1, 7

[18] Qihao Liu, Weichao Qiu, Weiyao Wang, Gregory D Hager, and Alan L Yuille. Nothing but geometric constraints: A model-free method for articulated object pose estimation. *arXiv preprint arXiv:2012.00088*, 2020. 3

[19] Robert Maier, Kihwan Kim, Daniel Cremers, Jan Kautz, and Matthias Nießner. Intrinsic3d: High-quality 3d reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *Proceedings of the IEEE international conference on computer vision*, pages 3114–3122, 2017. 2

[20] Roberto Martín-Martín, Clemens Eppner, and Oliver Brock. The rbo dataset of articulated objects and interactions. *The International Journal of Robotics Research*, 38(9):1013–1019, 2019. 1, 3, 5

[21] Roberto Martín-Martín, Sebastian Höfer, and Oliver Brock. An integrated approach to visual perception of articulated objects. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5091–5097. IEEE, 2016. 3

[22] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 2, 4

[23] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. *arXiv preprint arXiv:2101.02692*, 2021. 3

[24] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019. 3, 5

[25] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. *arXiv preprint arXiv:2104.07645*, 2021. 2, 7

[26] Erik Oberg and Franklin Day Jones. *Machinery's Handbook*. Industrial Press, 1914. 6

[27] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 7

[28] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++ deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5105–5114, 2017. 6

[29] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 7

[30] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019. 3, 5

[31] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020. 1, 3, 4, 5, 6

[32] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *ACM Transactions on Graphics*, 37(6), 2019. 3

[33] Vicky Zeng, Timothy E Lee, Jacky Liang, and Oliver Kroemer. Visual identification of articulated object parts. *arXiv preprint arXiv:2012.00284*, 2020. 3