博士論文

Part-level Shape, Pose and Kinematics Understanding of Man-made Articulated Objects

（関節物体に対するパーツレベルの形状・姿勢・動作に関する理解）

川名 雄樹

Acknowledgements

This research was conducted under the supervision of Prof. Tatsuya Harada at the Research Center for Advanced Science and Technology, the University of Tokyo.

Firstly, I would like to express my most profound appreciation to my supervisor Prof. Tatsuya Harada. I joined his laboratory 2019 and since then, he has always supported me with encouraging, constructive, and insightful conversations. Beyond the technical aspects, he has strongly influenced my attitude and approach to research.

I would also like to extend my sincere thanks to Prof. Takanori Fukao, Prof. Kei Okada, Assoc. Prof. Ko Yamamoto, and Lecturer Yusuke Mukuta for their constructive comments and suggestions. As sub-chief examiners of this doctoral thesis, their insightful feedback has significantly refined this work.

My comfortable life in the laboratory was supported by the help of Dr. Yusuke Mukuta, Dr. Yusuke Kurose, Dr. Lin Gu, Dr. Takayuki Osa, Dr. Westfectel Thomas, Dr. Tejero de Pablos Antonio, Dr. Hiroaki Yamane, Dr. Tomoyuki Takahata, Dr. Qier Meng, Dr. Masatoshi Hidaka, and Ms. Miyuki Kajisa. They have never spared any efforts for improving the research environment.

Discussions with my colleagues were always inspirational. While space constraints prevent me from listing all who have influenced me, I particularly thank Dr. Yusuke Mukuta, Dr. Yusuke Kurose, Dr. Antonio Tejero de Pablos, Dr. Hiroaki Yamane, Dr. Hiroharu Kato, Dr. Takuhiro Kaneko, Dr. Yang Li, Dr. Hao-Wei Yeh, Dr. Atsuhiro Noguchi, Dr. Kohei Uehara, Dr. Yusuke Mori, Takayuki Hara, Haruo Fujiwara, Kenzo Lobos Tsunekawa, and Motallebi Mohammad Reza for their contributions.

Finally, I would like to express my gratitude to my wife, Sonoka, for her unwavering support throughout my Ph.D. course.

Abstract

This thesis proposes an end-to-end, unified framework for understanding articulated objects from a single-view RGBD input. The primary focus lies in the estimation of part shape, pose, and kinematic parameters for man-made articulated objects in a scene, such as household appliances and furniture. These objects are assumed to have a base and one or more connected movable parts, excluding loop structures, separations, or structures where one movable part is attached to another.

The proposed method addresses multiple challenges associated with the understanding of articulated objects captured from a single viewpoint. The task is highly ill-posed, in terms of shape reconstruction and kinematics estimation due to partial observation of shapes and the static target without motion. The diversity and combination of part poses, shapes, kinematics, part counts, and structures presents a significant challenge in understanding 3D attributes of the articulated objects due to the arising complexity. Moreover, the need for 3D part-level supervision signals, such as kinematic parameters and part labels, makes training low data-efficient. Existing studies have failed to provide a comprehensive solution, by assuming prior knowledge for the target articulated object to manipulate for obtaining prior kinematics information, limiting part structure and counts for the target articulated objects to limit the complexity in shape reconstruction, and requiring a whole set of 3D part-level annotations for all training data.

This thesis proposes a comprehensive pipeline in response to the above problems. The pipeline takes a single RGBD image, camera intrinsics, and, optionally, foreground masks as input, and outputs the part shape, pose, kinematic parameters of individual parts, and the hierarchical structure of parts that make up the instance. This thesis further delves into unsupervised learning for data efficiency besides the supervised approach, by exploiting the fact that certain everyday articulated objects, such as ovens and washing machines, tend to have consistent part structures. The method also explores unsupervised segmentation of parts into finer semantic shapes, such as the handle of the doors, which is essential to recognize a preferable contact point for motion planning. By combining both the supervised and unsupervised approaches, the proposed pipeline is learned in a semi-supervised manner, which is the most realistic setting that we have access to the annotation for a portion of

data while reducing the amount of required annotation for better data efficiency whenever possible.

In summary, this thesis makes significant strides in understanding articulated objects from a single-view input, developing a unified framework that handles part detection, shape reconstruction, pose estimation, kinematic estimation, and segmentation of the finer shape details. Through this work, we hope to facilitate further exploration and advancement in the understanding of man-made articulated objects.

Table of contents

Chapter 1

Introduction

1.1 Background

3D understanding of objects is crucial for perceiving the environment. Humans can understand complex 3D environments from limited viewpoints only with color and depth information from stereovision. For example, by looking at the washing machine from one view, humans can instantly identify the parts that make it up, imagine their shapes, and reason about pose and kinematic constraints of those parts. Understanding these properties is essential for daily tasks to interact with such objects, such as physically grasping an object, determining how to grasp it, or identifying the appropriate grasping points. Therefore, in computer vision, understanding the 3D property of objects from limited viewpoints, especially from a single viewpoint, has been an important field to achieve intelligent machines with vision capabilities similar to humans.

While existing research has advanced in the areas of everyday objects important to human life, such as furniture, appliances, and vehicles, these studies have mainly focused on rigid objects. Man-made articulated objects as shown in Figure 1.1, with movable parts like drawers, lids, and doors, have not been extensively studied despite their abundance in our surroundings and the essential roles they play in our daily lives. 3D understanding of articulated objects is particularly important for scene interaction in AR/VR applications and scene understanding for robotics applications to manipulate objects. In this thesis, we address the challenges of shape reconstruction, pose and kinematics estimation, and part parsing of man-made articulated objects from a single view observation, thus filling the gap in the existing literature. We discuss three important elements in this section: (1) shape reconstruction, (2) pose and kinematics estimation, and (3) part parsing.

Figure 1.1 Example of man-made articulated objects.

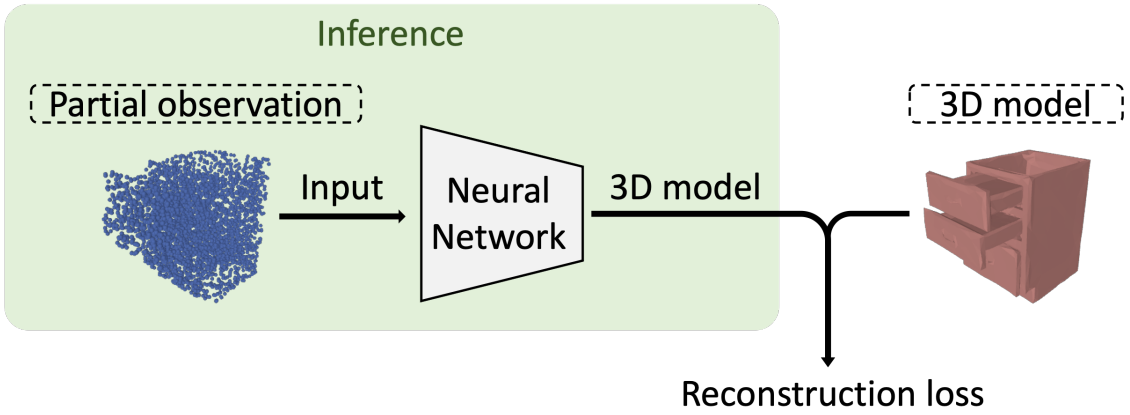


Figure 1.2 Overview of shape reconstruction from partial observation.

1.1.1 3D shape reconstruction from partial observation

Reconstruction of 3D shape from a single viewpoint has been explored in computer vision research as a shape-from-x task. Earlier works exploit partial observations such as shade [40, 44], texture [132], or focus cues [85] to reconstruct 3D shapes.

In recent years, learning-based approaches have advanced in reconstructing more comprehensive shapes from a single RGB image, sparse point cloud, or low-resolution voxels [127, 102, 35, 93, 90]. Learning-based shape reconstruction methods typically involve training a model on large datasets of 3D object shapes to generalize to unseen shapes.

1.1.2 Pose and kinematic estimation for articulated objects

Pose and kinematic estimation plays a crucial role in determining the position, orientation, and mobility constraints of an object's parts in 3D space. This estimation is vital for understanding the functionality and interactions of articulated objects. Pose and kinematic estimation for articulated objects typically involves 3D pose estimation of individual articulated parts and estimation of the corresponding joint parameters as kinematics representation. Several approaches estimate the pose and kinematics of articulated parts, including those that rely

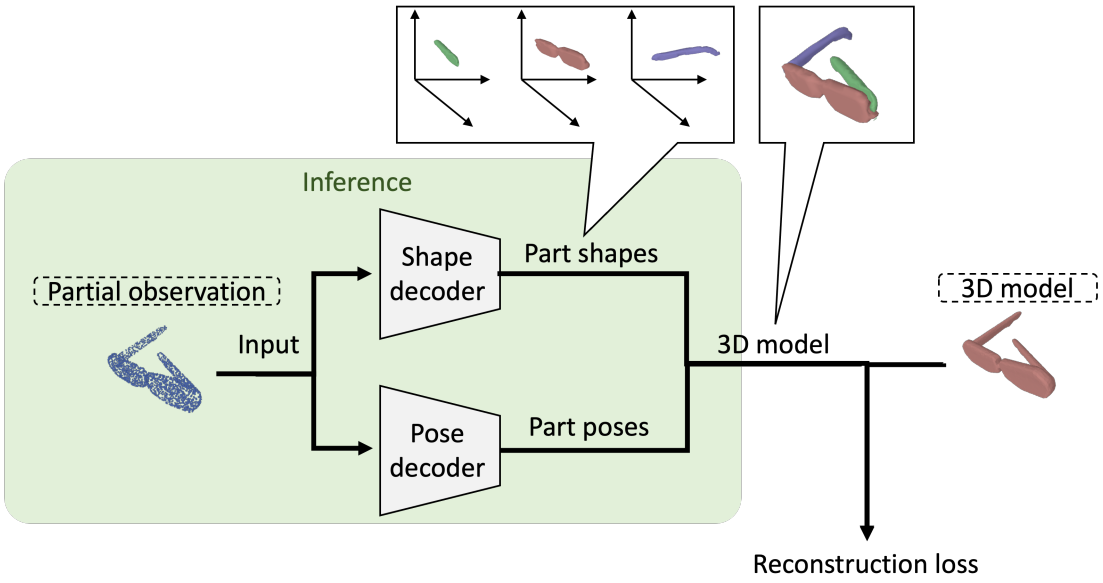


Figure 1.3 Overview of generative part parsing from partial observation.

on interaction to estimate kinematics of parts by physically manipulating articulated parts [75, 87], those that assume spatio-temporal observations of moving articulated parts to estimate underlying kinematic models [140, 47], and recent works that estimate pose and kinematics from observation of static targets at inference time, not requiring target's spatio-temporal observations or physical interaction [63, 128, 48, 32]. However, these methods require fully annotated data for supervision.

1.1.3 Part parsing

Part parsing is the process of decomposing an object into its constituent parts based on semantic understanding. In the context of articulated objects, part parsing is closely related to understanding the functional components and their relationships. The most straightforward approach learns part parsing as a segmentation task using part-level segmentation labels [101, 128, 63]. For articulated objects, motion cues can be exploited to induce part segmentation [43, 140, 49]. Another line of work employs the analysis-by-synthesis approach as a generative part decomposition task [26, 94, 19], which aims to reconstruct 3D shape as a combination of multiple primitive shapes, realizing unsupervised part segmentation through shape reconstruction. The overview of the generative part parsing is visualized in Figure 1.3.

1.2 Challenges

Addressing the following challenges will be crucial for improving the 3D understanding of articulated objects from single-view input:

1. **Shape reconstruction with exponentially increasing possible shapes**: Reconstructing objects with articulated parts presents a significant challenge due to the exponential increase in possible shapes caused by the mobility of the parts and diverse part structures. An efficient approach which can handle this variability is required.

2. **Annotation efficient pose and kinematics learning**: Existing supervised approaches rely on full supervision of pose and kinematics learning. However, part-level annotation, which is necessary for supervision, is often expensive. Therefore, it is crucial to devise unsupervised methods that minimize annotation costs. Previous methods relying on spatio-temporal observation show that pose and kinematics can be modeled by learning canonical poses and their transformations. However, articulated objects, unlike humans and animals, rely on external forces to manipulate their movable parts and lack inherent self-movement. This characteristic renders methods that rely on temporal information unsuitable for articulated objects. While interaction-based approaches have been explored to address this challenge, they require prior knowledge of the desired shape and motion.

3. **Balancing semantic part parsing and accurate shape reconstruction**: The challenge involves the need for detailed semantic shape information without expensive manual annotations. Generative part parsing realizes the unsupervised approach for part decomposition of shapes. However, these methods rely on less expressive primitive shapes to induce unsupervised part decomposition of the target shape in shape reconstruction, facing a trade-off between accurate shape reconstruction and the semantic capability of the resulted decomposition. The ideal solution requires a shape representation that achieves both semantic shape reconstruction and high reconstruction accuracy simultaneously.

Addressing these challenges will advance the field of understanding articulated objects from a single viewpoint and contribute to the development of more accurate and robust techniques for shape reconstruction, pose and kinematics estimation, and part parsing.

Chapter 2

Towards a Unified Framework for Part-level Articulated Object Understanding

In this chapter, we clarify the scope, objective, contribution, and structure of this thesis before going into the details. In Section 2.1, we clarify the objective and scope of this thesis. In Section 2.2 to 2.4, we discuss the proposed approach and system. We also show our contributions toward building this system in Section 2.5. In Section 2.6, we show the contents of each chapter.

2.1 Objective and scope

In this thesis, we propose comprehensive methods for estimating part shape, pose, kinematic parameters, and sub-part level labels for man-made articulated objects from single-view RGBD input as a semi-supervised system. We illustrate the input and output of the pipeline in Figure 2.1. The scope of this thesis is illustrated in Figure 2.2. We target man-made objects such as furniture and appliances which are static without self-motion and have various part structures, as opposed to natural articulated objects such as humans and animals, which have fixed articulated structure and can be dynamic. Within articulated objects, we assume a structure where common household appliances and furniture have a base and one or more connected movable parts with 1D prismatic or revolute joints. We do not consider loop structures, separations, or sequential joints where one movable part is attached to another movable part.

This system should have the following features:

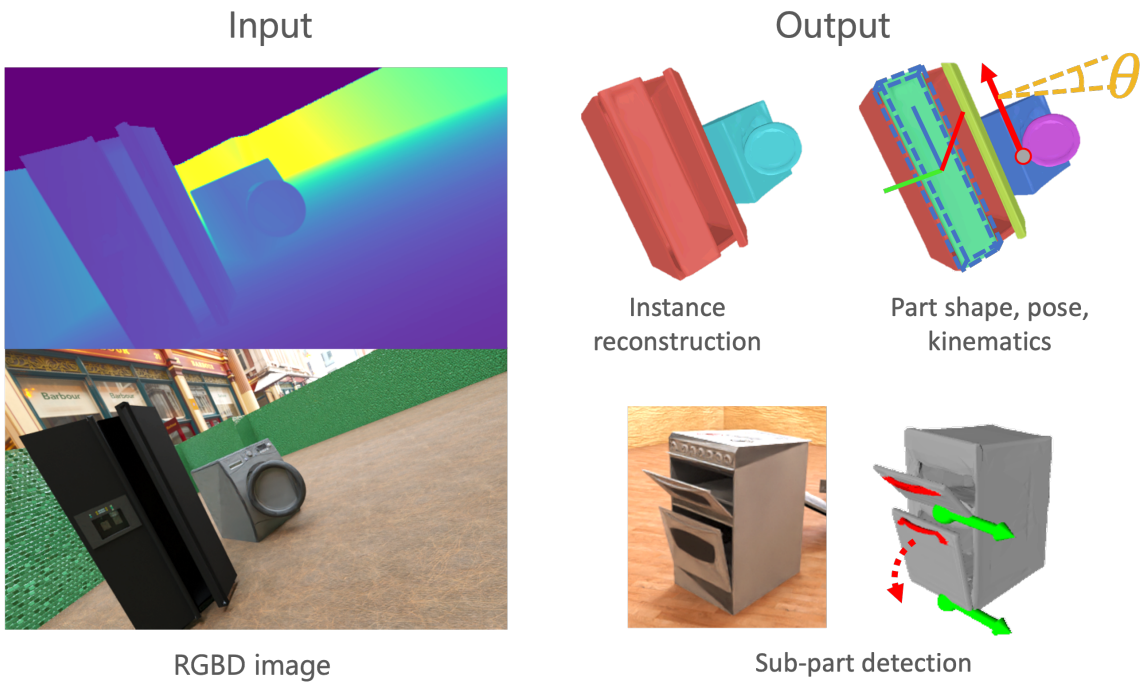


Figure 2.1 Overview of our proposed method.

1. It can perform estimation given unsegmented partial shape and color observation as a single RGBD frame, so that we do not assume interaction or prior knowledge of manipulation of articulated objects.

2. It can handle various part structures to target a wider range of real-life articulated objects.

3. It can reduce costly 3D part-level annotation.

4. It can further understand finer semantic detail beyond part-level shape, such as the handle of objects, which is needed to perform downstream tasks like grasping.

2.2 Part-level understanding of articulated objects

In this thesis, we achieve this system by part-level understanding of articulated objects by considering both supervised and unsupervised approaches. Below, we discuss how part-level representation can address the required features of the system.

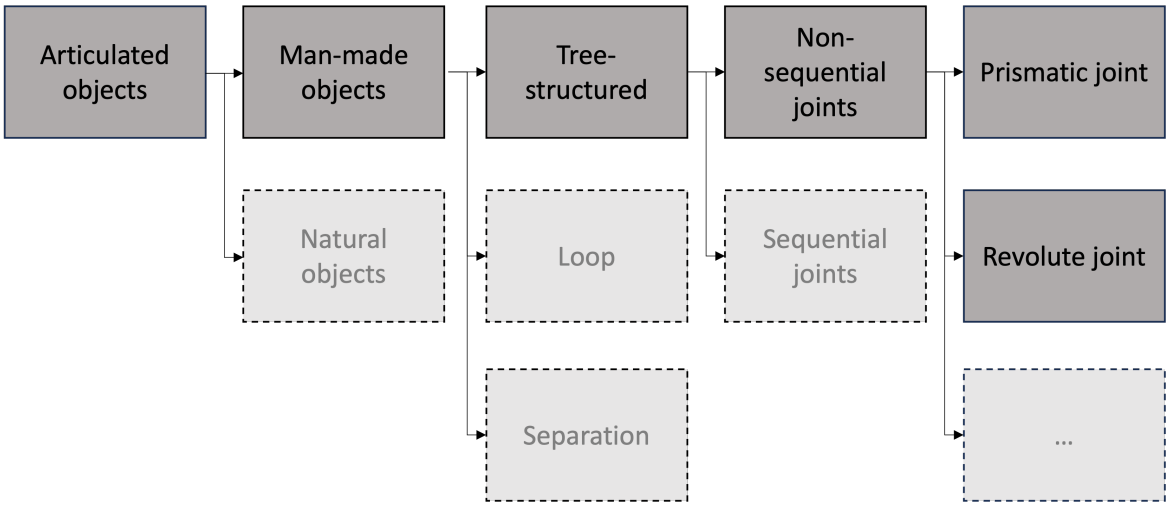


Figure 2.2 Scope of reconstruction target.

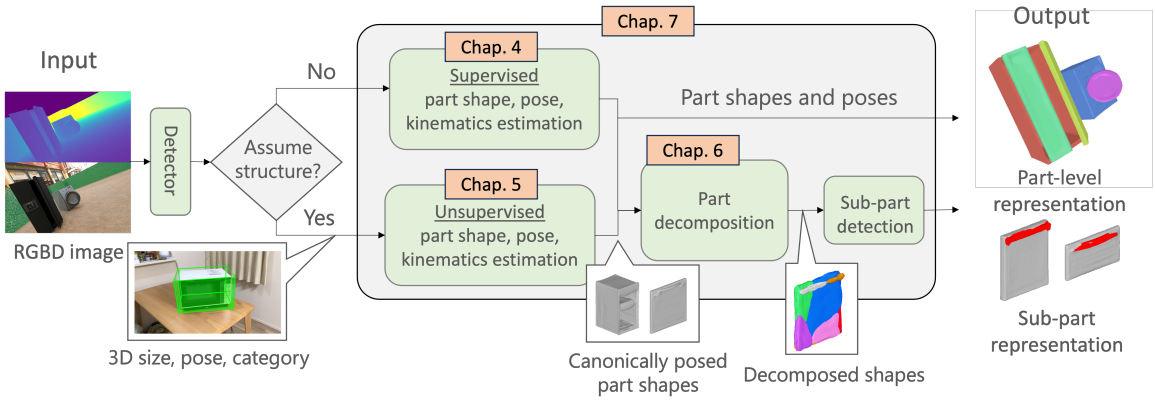


Figure 2.3 Overview of the proposed system.

2.2.1 Handling Variously Structured Articulated Objects

Articulated objects have diverse part configurations in part structures and part counts. Even within a single category, like a drawer, their part configurations significantly vary in a real environment.

Previous studies [38, 67, 83] reconstruct the whole shape without part-level understanding. However, the combination of different part poses, sizes, shapes, numbers of parts, and their layout results in exponential complexity in the task, limiting previous works' targets to simply structured articulated objects with only a few parts. Part-level representation can efficiently handle this difficulty as we decompose the task into the reconstruction of simpler part shapes.

In contrast, recent works [32, 48] on pose and kinematic estimation consider part-level understanding articulated objects. However, they do not consider shape reconstruction, and

instance detection is necessary before estimating part-level attributes. Therefore, a unified method that covers from detection, pose and kinematic estimation to shape reconstruction has not been considered so far. In Chapter 4, we introduce a single-stage, end-to-end method to address this problem as a detection-based shape reconstruction. Our proposed method takes a single RGBD image, camera intrinsic matrix, and optionally, a foreground mask recognized by a separate model as input, and outputs the shape, pose, size, joint parameters of individual parts, the category of parts, and the hierarchical structure of parts that make up the instance. The model is trained by part-level supervision on a large-scale synthetic dataset based on [133] which have variously structured articulated objects.

Moreover, we extend our detection-based approach tailored for articulated objects. For non-articulated objects, methods that perform detection to reconstruction have been considered in the previous works [88, 141]. However, articulated objects are composed of small and thin parts. In addition, while those methods assume only yaw rotation for instances, the parts we target can take various poses, making it a more challenging problem. To improve detection performance of these parts, we focus on the trajectory of movement of small and detailed parts, and by grouping individually detected parts with a common trajectory and integrating them, we reduce false positives and improve recognition accuracy. Furthermore, previous works [32, 48] required separate instance detector for part-level understanding of articulated objects. Our proposed method avoids such additional detector. Instead, our approach detects individual parts along with their association to instances in the scene. This allows a unified approach from part-level detection to instance-level understanding of target objects in a single stage. This detect-then-group approach is advantageous for articulated objects due to the countless possible part structures and vast number of shape combinations.

2.2.2 Reducing costly 3D part annotations

The previous section considered cases where the input part structure is diverse and not limited. However, in important articulated objects in daily life such as microwaves, laptops, etc., there are cases where it is practical to assume a part structure depending on the category. As mentioned earlier, with the use of a detector like previous works [32, 48], it is possible to detect instances, their global poses, sizes, and recognize their categories. However, compared to the annotation required for the automatic acquisition of global pose and size using depth sensors or AR markers, the data acquisition cost of part-level information for part poses, individual part shapes, and segmentation is high because (1) it requires manual annotation when we use scanned shapes of real objects or (2) building a large number of realistic synthetic assets with complex textured shapes requires professional artists thus expensive and not scalable. There exist unsupervised part decomposition approaches [124, 94, 19]

which decomposes target shapes into semantic parts as a set of primitive shapes which have consistent part shapes and 3D position without part-level annotation. They target rigid objects and assume the same parts across instances are reasonably fixed in the similar position in the 3D space to induce the unsupervised decomposition. However, articulated objects consist of moving, dynamic parts and they cannot estimate kinematic parameters beyond 3D location of parts.

In Chapter 5, when it is assumed that the part structure is consistent for each category, we show introducing pose and kinematics aware part representation can effectively solve unsupervised learning of part shapes, pose, and kinematic estimation without relying on 3D part annotation, but instead only using whole shape supervision. Such whole shape supervision is easily available from recent multiview shape reconstruction only requiring casual capture of target objects and able to reconstruct textured shape with fine details.

2.3 Decomposition of parts into finer semantic shapes

So far, we have seen methods for decomposing parts into shapes that are independent of part poses in both supervised and unsupervised cases. However, there are cases where more detailed semantic shape information is required. For example, when recognizing a front panel to pull out a drawer or recognizing a drawer bottom to place an object to store. Since such finer annotations are expensive, it is desirable to automatically decompose shapes into finer and consistent shapes. Furthermore, for shape recognition, it is desirable to have high reconstruction accuracy. Previous works learn such decomposition without segmentation annotation [19, 26, 124, 94] for 3D shapes, by reconstructing target shapes using simple primitive shapes as analysis-by-synthesis approach, known as generative part decomposition. However, there was a trade-off between accurate shape reconstruction [19, 26] and semantic decomposition [124, 94]. This is because if simple primitives such as cuboid are used, individual parts have semantic correspondence with the original object, but when projected onto the original shape, the segmentation becomes inaccurate at ground truth semantic shape boundaries, or the reconstruction accuracy is low due to the use of too simple shapes. On the other hand, if a large number of parts are used for learning reconstruction, the semantic meaning of individual shapes becomes less significant. Therefore, a shape representation that can achieve semantic shape reconstruction while maintaining reconstruction accuracy is necessary.

In Chapter 6, we introduce a primitive representation termed neural star domain for generative part decomposition to achieve both accurate shape reconstruction and high semantic capability. The star domain is a geometry of generalization from previous methods such as

	Single-view input	Various structures	Unsupervised	Shape reconstruction	Semantic parts
Heppert et al. [37]	✓			✓	
Huang et al. [43]		✓	✓		✓
Huang et al. [32]	✓	✓			✓
Paschalidou et al. [94]	✓		✓		✓
Chen et al. [19]	✓		✓	✓	
Ours	✓	✓	✓	✓	✓

Table 2.1 Comparison with the previous works.

convex [19, 26], cuboid [124], and superquadrics [94]. It is also suitable for representing parts with thin and detailed surfaces, such as articulated objects by optimizing the analytical surface representation made available by the star domain. Additionally, by using the capacity of the neural network to represent the star domain, our proposed shape representation can fully utilize the parameters of the neural network, compared to previous shape representations based on several hundred parameters for cuboids, superquadrics, and convex shapes.

2.4 System overview

Figure 2.3 shows an overview of our method. Our method takes an RGBD image with multiple articulated objects as input, and reconstructs instances. Moreover, each predicted instance consists of part shape, pose, kinematic parameters. Finally, each part shape is further decomposed into sub-part level shapes. Inside the system, we combine both the supervised and unsupervised approaches, making the system a semi-supervised pipeline to address both complex part configurations of articulated objects and annotation efficiency. We summarize our system compared to existing methods in Table 2.1.

2.5 Summary of contributions

To summarize our proposed method:

- When dealing with various structures and unable to assume a consistent structure for each category, we learn parts detection and reconstruction through supervised learning of part shapes and part poses (Chapter 4).
- When assuming a consistent structure for each category and learned instance detector, we reconstruct parts shapes and part pose estimation without using annotations for part shapes and part poses (Chapter 5).
- We explore the reconstruction of finer-level semantic part shapes (Chapter 6).

- We unify the above approaches with semi-supervised learning to consider the most realistic scenario of leveraging both the annotated and unannotated data (Chapter 7).

2.6 Structure of the thesis

In this thesis, we aim to develop a comprehensive understanding of articulated objects from a single-view input. Chapter 1 provides a background and motivation for this thesis. In Chapter 2, we discuss the objective and scope of this thesis, introduce core ideas of part-level understanding of articulated objects, and present the proposed system. In Chapter 3, we discuss related research comprehensively. In Chapter 4, we propose a framework for understanding variously structured articulated objects through supervised learning of part shapes and part poses. In Chapter 5, we consider a method to learn the shapes, joint parameters, and poses of individual parts without using annotations for part shapes and part poses when assuming a consistent part structure. In Chapter 6, we introduce a novel shape representation called the neural star domain, which allows for finer-level semantic part shapes while maintaining reconstruction accuracy. In Chapter 7, we introduce and demonstrate our unified pipeline. Finally, in Chapter 8, we conclude this thesis and remark on future directions.

Chapter 3

Related works

This chapter presents related work on articulated shape reconstruction, part pose and kinematics estimation, and part decomposition.

3.1 Articulated shape reconstruction

3.1.1 Natural articulated objects

A growing number of studies have tackled the reconstruction of category-specific, articulated objects with a particular kinematic structure, such as the human body and animals.

Representative works rely on the use of category-specific template models as the shape and pose prior [142, 143, 60]. They assume the target articulated objects have consistent kinematic model and shape, thus utilizing the template model as a "mean shape" of the target objects as a prior. Low dimensional parametrization of human shape has been proposed and enables reconstructing the target shape and pose without directly deforming the template [69, 9]. Recently, [138, 139] have proposed to learn articulated shapes from set of images dipicting the articulated objects from the same category and kinematic model without using templates.

Focusing on human shapes, there exists large body of works for estimating target shapes and textures from single-view input [108, 109, 65]. They utilize the category specific prior knowledges of spatial and textural information obtained from large datasets.

Another body of works reconstruct target shapes exploiting temporal and pose-independent consistency of the target object in canonical frame. Implicit field representation proposed by [27] reconstruts a part-wise implicit field given a part pose as an input to deform canonically posed shape. Another works focus on non-rigid tracking of the seen samples [11, 64] from point cloud, or from video [137, 136]. These works explicitly track trajectory of surface

points in point cloud or continous surface correspondence along temporal direction to deform canonical shape to reconstruct target shape.

In contrast, our approach focuses on man-made articulated objects with various kinematic structures even within the same category. Thus we cannot simply apply template based approaches assuming the uniform kinematic structures. Moreover, due to various part structure of target shapes, we cannot strongly rely on category-specific prior knowledge of target shapes. Also, man-made articulated objects we are interested in are static wihout self-motion. Moreover, explicit surface correspondense is not always available for man-made articulated objects especially for prismatic parts whose surface are largely occluded by base part when closed. Therefore, we cannot assume access to temporal correspondence information of the target for tracking based reconstruction. These problems neccessitate us to develop different techniques for man-made articulated objects reconstruction.

3.1.2 Man-made articulated objects

To reconstruct man-made articulated objects considering articulation under category-specific , single-view setting, [83] proposes to disentangle pose and shape for latent space modeling to tackle the complexity in shape encoding in latent space. They first encode target shape into shape latent space and also estimates part poses. Then they concatenate shape latent embedding and 1D scalar degree values per part as input for shape decoder to reconstruct target shape. In multi-view setting, [123] also disentangle pose and shape latent space and realized textured shape reconstruction by neural radiance field [79]. [130] also employs neural radiance field and further proposed to disentangle latent space into shape, pose and additionally kinematic model category, enabling few-shot reconstruction. [49] proposes to learn part-aware shape reconstruction from pair of frames with different articulation state.

To address multi-category, single-view setting, [67] proposes to employ multiple category-specific shape decoder from [83] and switch between them based on the category estimation in the previous detection stage. More recently, [38] proposed end-to-end trainable pipeline from detection to reconstruction of man-made articulated objects. They employ single shape decoder based on [83] applicable to multi-categories setting by improving latent space modeling in detection stage and latent space optimization technique in inference stage. However, they limit the targe kinematic model to have only single articulated part.

All the above approaches learn whole target shape in instance-level latent space and their shape decoder can only handle simple kinematic structures with a few parts for category-specific setting [83, 130, 123, 67] or single part for multi-category setting [38]. This is because the single shape decoder need to learn complex shape variation arising from combination of different part poses, part structures and part counts. Especially, variation of part

structures exponentially increases with more number of parts if we regard part strcuture as partitioned rectangle [104, 25], making the previous approaches not scalable to more complex part structures seen in daily life. To address this problem, in Chapter 4, we propose part-level latent-space for shape decoding and show our approach can effectively reconstruct articulatd object shapes with far more complex part structure. Moreover, even under the assumption of simple kinematic model and category-specific setting, the previous works [83, 130, 123] requires costly annotations for part segmentation or 3D poses. In Chapter 5, we show the part-wise reconstruction and pose can be learned in disentangled way only by using global shape supervision, reducing the annotation cost.

3.2 Part pose and kinematics estimation

Estimating kinematic models from interaction through optimization approach has been studied in the field of robotics [119, 51]. Recent approaches employ learning-based methods based on interaction and spatio-temporal observations [75, 87, 99, 72, 49, 42, 140, 47, 43]. However, these works require costly human manipulation or a prior knowlwdge on the interaction to the target. In this thesis, our goal is to acquire such prior knowledge without interaction or manipulation through learning-based framework for single-view input.

Another line of works infer the pose and kinematics from observation of static target, without interaction or spatio-temporal information. The early prior work [78] estimates articulated part poses given kinematic model and known part shapes from depth input, optimizing the part poses by the kinematic model constraint. Based on large synthetic dataset [133, 128], the previous works [63, 128, 1] estimates the part poses of unseen targets. More recently, detection based approaches have been proposed [48, 32] for handling more complex part structures of articualted objects. However, the piror works are fully supervised in part level, and can only handle know target shapes [78], whole shape observation [128], known kinematic models [63, 1] and limited to pose and kinematics estimation without shape reconstruction. In Chapter 4, we introduce the comprehensive framework to estimate part shape, pose and kinematic models. Moreover, in Chapter 5, we discuss an unsupervised approach which does not require part-level 3D annotation but to learn part shape, pose and kinematic models from whole shape supervision.

3.3 Part decomposition

Understanding shape as a set of primitive shapes has long been studied in computer vision [107, 8, 7]. Recent approaches learns the decomposition in an unsupervised manner

by analysis-by-synthsis approach, by learning to reconstruct target shape with learnable primitive shape representation for shape abstraction task [124, 94, 18, 33, 95] and accurate shape reconstruction [52, 19, 26]. The previous works demonstrates the few-shot part segmentation by manually annotating part labels to primitive shapes of the few reference samples. However, the previous works have trade-off between shape reconstruction accuracy and semantic capability of the decomposition; using more number of primitive shapes for higher reconstruction accuracy loses the parsimoniousness of the number of primitive shapes, leading to difficulty in manual labeling for few-shot segmentation in practice. This is because the previous works primitive shape representation has low shape representation capability due to its low dimensional parametrization, making a single primitive shape difficult to represent complex shape, requiring more primitive shapes to accurately reconstruc the target shapes. To address this trade-off, in Chapter 6, we study an expressive primitive shape representation by MLP as primitive shape in construction. This enables to represent target shape with fewer number of primitive shapes compared to the previous works while outperforming in the reconstruction accuracy.

Moreover, the previous works assumes the part locations are consistent across training samples, thus using this as an inductive bias to learn the decomposition. However, when applying part decomposition methods to articulated objects, such assumpation leads to inconsistent part decomposition which ignores the fact that the part locations are dyinamic due to different part poses and underlying part structures. In Chapter 5, we study the novel setting of the part decomposition task targeting the articulated objects, learning the consistent decomposition which considers underlying part poses and kinematics.

Chapter 4

Handling Variously Structured Articulated Objects

4.1 Introduction

Estimating object shape, pose, size, and kinematics from a single frame of partial observation is a fundamental challenge in computer vision. Understanding such properties of daily articulated objects has various applications in robotics and AR/VR.

Shape reconstruction of daily articulated objects is a challenging task. These objects exhibit a range of shapes resulting from different local part poses. More importantly, they display significant intra- and inter-category diversity in part configurations, including variations in part counts and structures. These factors together contribute to an exponentially increasing shape variation. Previous works have addressed this issue by either limiting a single model to target objects with a single articulated part [38] or employing multiple category-level models [67] to accommodate varying part counts. These approaches first detect each instance, and then model the target shape in an instance-level latent space, primarily employing A-SDF [83] for shape learning. A-SDF maps the target shape into an instance-wise latent space, and then a shape decoder outputs the entire shape of the instance. However, this approach is limited when dealing with varying part counts and structures, as the shape decoder must handle an exponentially increasing number of shape variations due to different part layout combinations [104, 25] in addition to local part poses. Consequently, addressing this variety with a single model remains a complex and unsolved task.

In this paper, we address this complexity through our novel detect-then-group approach. Our key observation is that daily articulated objects consist of similar part shapes. For example, regardless of the number of refrigerator doors, each door typically has a similar

shape, and the base part may share similarities with those from other categories, such as storage units. By detecting each part and then grouping them into multiple instances, we provide a scalable and generalizable approach for handling diverse part configurations of daily articulated objects in a scene.

Based on this concept, we propose an end-to-end detection-based approach for part-level shape reconstruction. Building upon 3DETR [80] as an end-to-end trainable detector backbone, given a single RGBD image with an optional foreground mask, our model outputs part shape, pose, joint parameters, parts-to-instance association, and instance category. Our approach employs a novel detect-then-group approach. It first detects parts and applies simple clustering of parts into instances based on learned part embeddings' distance, in contrast to the previous works using additional instance detection module [67, 38]. An overview of our approach is shown in Figure 4.1. However, we found that detection-based shape reconstruction is prone to false positives for articulated objects' thin and small parts with little overlap, which is hard to remove by NMS. Also, articulated objects often have parts of varied sizes and scales, making training with a single-shape decoder challenging. Additionally, increasing model size by end-to-end training from detection to reconstruction makes simply enlarging the model size to improve performance undesirable. To address these challenges, we propose:(1) kinematics-aware part fusion to reduce false positives and improve detection accuracy; (2) anisotropic scale normalization for various part sizes and scales in shape learning; (3) and an output space refiner module coupled with a model-size balancing strategy with decoder for improved performance while keeping the model size. We evaluate our method on the photorealistically rendered SAPIEN [133] dataset, and our approach outperforms state-of-the-art baselines in shape reconstruction and joint parameter estimation. Furthermore, the model trained only on synthetic data generalizes to real-world data, outperforming the state-of-the-art methods on the BMVC [78] dataset.

Our contributions can be summarized as follows: (1) a novel part-level end-to-end shape reconstruction method for articulated objects from a single RGBD image; (2) a novel detect-then-group approach that simplifies the pipeline; (3) addressing detection-based reconstruction challenges with kinematics-aware part fusion, anisotropic scale normalization, and a refiner module coupled with model-size balancing; (4) superior performance on the SAPIEN [133] dataset, with the ability to generalize to real-world data from the BMVC [78] dataset.

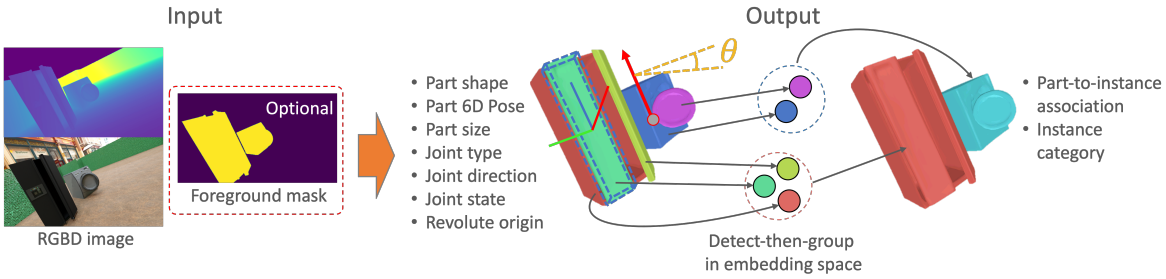


Figure 4.1 Our detection-based approach estimates part-level shape, pose, and kinematics as joint parameters. It also recovers parts-to-instance associations to handle multiple instances with various part structures and counts.

4.2 Related Work

Articulated shape reconstruction A number of works have focused on human subjects using single image input [109, 108] and utilize category-specific templates to recover deformation from a canonical shape [69, 142, 9, 143, 60]. Recent research has delved into recovering articulated shapes with unknown kinematic chains from sequences of a target in motion [92, 137]. These works make category-level assumptions about kinematic structures, with targets in observations sharing common kinematic structures. Their main focus is on natural objects such as humans and animals. In contrast, our interest is in reconstructing the shape of multi-category, daily man-made articulated objects with diverse kinematic structures using a single model. Recent years have seen the emergence of methods specifically targeting man-made articulated objects [130, 49, 123], and taking single-frame input [83, 67, 38, 53]. However, these models are constrained by either a predefined number of parts per category or the necessity of multiple models for each combination of categories and part counts. Consequently, they are unable to scale to a wide array of real-world articulated objects with varying part counts using a single model. Our approach addresses this limitation.

Pose and kinematic estimation of articulated objects predominantly, existing research on pose and kinematic estimation of articulated objects has focused on estimation from sequences [37, 47, 131, 119], necessitating multiple frames of moving targets or interaction with the environment before estimation. Estimation from a single image has also been explored [63, 78, 67], but these methods are limited to predefined part structures. A few recent studies have proposed approaches without assumptions on part structure [48, 32]. However, these methods target single instances, requiring instance detection before part-level estimation. Moreover, their focus is limited to detection, pose and kinematic estimation, whereas our work aims for shape reconstruction in an end-to-end manner.

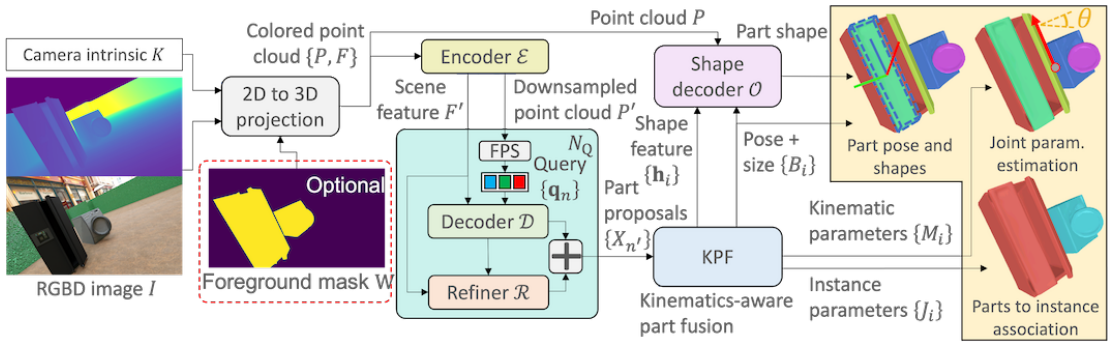


Figure 4.2 Overview of the pipeline. The input RGBD image is projected to a colored point cloud. The encoder $\mathscr{E}$ extracts scene features and a downsampled point cloud. The decoder $\mathscr{D}$ outputs a set of part proposals $\{X_n\}$ from part queryies $\{\mathbf{q}_n\}$. The refiner $\mathscr{R}$ estimates the residual of part pose and size ΔB and joint parameter ΔA for refined $\{B, A\}$. At test time, the inference is run N_Q times independently to densely sample part proposals as $\{X_{n'}\}$. KPF removes false positives in $\{X_{n'}\}$ by using kinematics-aware IoU (kIoU) to refine the prediction further. The part shape is reconstructed by the implicit shape decoder $\mathscr{O}$.

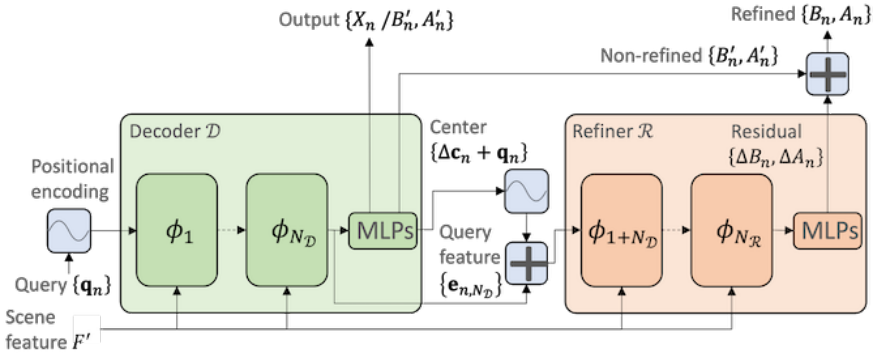


Figure 4.3 Architecture of refiner $\mathscr{R}$

Detection-based reconstruction A large body of work exists combining detection and reconstruction for multiple rigid objects in diverse settings, such as indoor scenes [121, 88, 89, 141, 125, 34], tabletop environments [45, 46], and road scenes [6, 74]. Recent works target daily articulated objects [67, 37]. However, these methods predominantly rely on an instance-level detection approach. In contrast, our work pivots towards part-level detection to effectively handle a wide variety of part structures of real-world articulated objects.

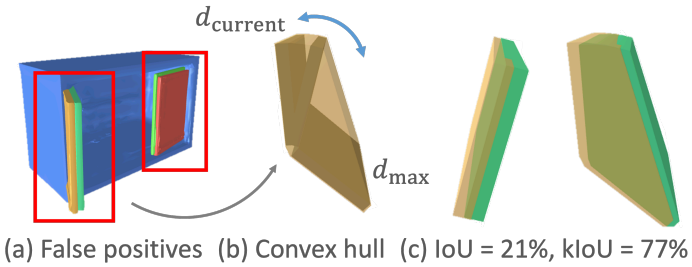


Figure 4.4 Illustration of (a) false positives, (b) Convex hull for kIoU, and (c) comparison of 3D box IoU and kIoU for overlapping parts.

4.3 Methods

4.3.1 Problem setting

Our method takes a point cloud P with N_P 3D points and color feature F lifted from a single RGBD image I of articulated objects and camera intrinsic K as input. It outputs a set of parts $G = \{i \in [N]\}$ and non-overlapping subsets of all parts $\{G_l\}_{l=1}^{N_G}$ as N_G instances, where $G = \bigcup_l \{G_l\}_{l=1}^{N_G}$. Note that we use the shorthand $\{*_i\}$ to denote an ordered set $\{*_i\}_{i=1}^{N}$ for brevity. For each part i, we estimate 6D pose and size $B \in \mathrm{SE}(3) \times \mathbb{R}^3$, part shape $\mathscr{O}$ as an implicit representation and kinematic parameters M. Our shape representation is an implicit function defined as $\mathscr{O} : \mathbb{R}^3 \to [0,1]$ with isosurface threshold $\tau_{\mathscr{O}}$, where $\{\mathbf{x} \in \mathbb{R}^3 \mid \mathscr{O}(\mathbf{x}) > \tau_{\mathscr{O}}\}$ indicates inside the shape. The kinematic parameter M consists of joint type $y \in \{0,1,2\}$ which represents fixed, revolute, and prismatic types, joint direction $\mathbf{a} \in \mathbb{S}^2$, 1D joint state d_{current} and d_{max} for the current pose and fully opened pose from the canonical pose. We also predict the revolute origin $\mathbf{v} \in \mathbb{R}^3$ for the revolute part. We define the joint parameter as $A = \{\mathbf{a}, d_{\text{current}}, d_{\text{max}}[, \mathbf{v}]\}$, thus $M = \{y, A\}$. For each instance l, we estimate the instance parameter J which consists of category u and part association defined as $\delta_{li} = \mathbb{1}(i \in G_l)$, where $\mathbb{1}$ is an indicator function.

4.3.2 Detection backbone

Our detection backbone consists of a transformer-based encoder $\mathscr{E}$ and decoder $\mathscr{D}$ based on 3DETR [80]. The encoder comprises recursive self-attention layers encoding 3D points P and color feature F into downsampled 3D point cloud P' of $N_{P'}$ points and $D_{F'}$-dimensional scene feature F'. Query locations $\{\mathbf{q}_n\}_{n=1}^{N_q} \in \mathbb{R}^3$ are randomly sampled using furthest point sampling (FPS) from P'. The decoder is composed of transformer decoder layers $\{\phi_k\}_{k=1}^{N_{\mathscr{D}}}$, considering cross-attention between queries and F' and self-attention among queries. The

decoder iteratively refines query features $\{\mathbf{e}_{n,k+1}\} = \phi_k(\{\mathbf{e}_{n,k}\}, F')$. Lastly, a set of part prediction MLPs decodes each refined query feature to produce output values. For clarity, the query index n is omitted when possible.

4.3.3 Part representation

Part pose and size We predict part pose and size B as a set of part center $\mathbf{c} \in \mathbb{R}^3$, rotation $\mathbf{R} \in \mathrm{SO}(3)$ and size $\mathbf{s} \in \mathbb{R}^3$ for each query. We predict $\mathbf{c}$ as an offset $\Delta\mathbf{c}$ from $\mathbf{q}$, added to the query coordinates, i.e., $\mathbf{c} = \mathbf{q} + \Delta\mathbf{c}$.

Part shape We employ a shared-weight, single implicit shape decoder $\mathscr{O}$ for performing part-wise shape reconstruction by taking point clouds arsound the detected regions where parts are identified. Given the diversity in shape and pose, and anisotropic scaling of the parts we focus on, it is challenging to learn shape bias with a single shape decoder. Therefore, we propose anisotropically normalizing the side lengths of the shape decoder's input and output to a unit scale to perform reconstruction. We define the input point cloud $P_{\mathscr{O}}$ as follows:

$$P_{\mathscr{O}} = \{(\mathbf{RS})^{-1}(\mathbf{p} - \mathbf{c}) \mid \mathbf{p} \in P, \max(|(\mathbf{RS})^{-1}(\mathbf{p} - \mathbf{c})|) \leq 0.5\}. \tag{4.1}$$

where $\mathbf{S}$ denotes diagonal matrix of scale $\mathbf{s}$. We define the output occupancy value at $\mathbf{x} \in \mathbb{R}^3$ as $o_{\mathbf{x}} = \mathscr{O}((\mathbf{RS})^{-1}(\mathbf{x} - \mathbf{c}) \mid P_{\mathscr{O}}, \mathbf{h})$, where $\mathbf{h} \in \mathbb{R}^{D_{\mathbf{h}}}$ is a part shape feature modeled by a part prediction MLP. Given that the input point cloud $P_{\mathscr{O}}$ includes background, the geometry of the target part shape can be ambiguous. To address this issue, we train the detector backbone and shape decoder end-to-end, by inputting $\mathbf{h}$ as shape geometry to the shape decoder so that $\mathbf{h}$ informs the shape decoder of the foreground target shape. We utilize a shape decoder architecture with a lightweight local geometry encoder [98] to spatially associate input points $P_{\mathscr{O}}$ with output occupancy values $o_{\mathbf{x}}$, which are both defined in normalized space.

Part kinematics We predict a 4-dimensional vector $\mathbf{y}$ as a probability distribution over part joint types. This includes a 'background' or 'not a part' type for instances where predicted part proposals might not contain a part. The revolute origin $\mathbf{v} = \mathbf{q} + \Delta\mathbf{v}$ is predicted similarly to the part center $\mathbf{c}$, with an offset $\Delta\mathbf{v}$. Joint states d_{current} and d_{max} are modeled by separate part prediction MLPs for revolute and prismatic types, and we only supervise the output corresponding to the ground truth joint type. A single MLP is used for joint direction $\mathbf{a}$ for both revolute and prismatic types.

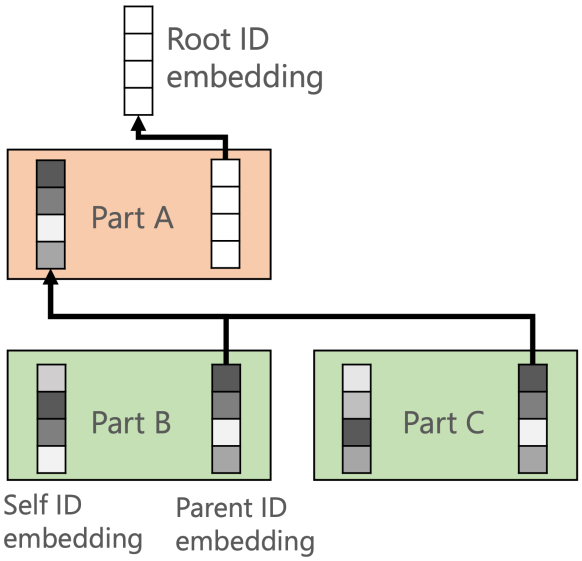


Figure 4.5 Illustration of the embedding-based sequential association. Part A shows the base part of the instance, and Parts B and C are the child parts.

4.3.4 Instance representation

We form a collection of parts as a single instance $G_l = \{i \in G \mid \forall i, j \in G, i \neq j \Rightarrow \|\mathbf{z}_i - \mathbf{z}_j\| < \tau_\mathbf{z}\}$, where the distance between the $D_\mathbf{z}$-dimensional embeddings, $\mathbf{z}_i$ and $\mathbf{z}_j$, of each part pair within the group is kept below a specified threshold $\tau_\mathbf{z}$. We predict an $N_\mathbf{u}$-dimensional vector $\mathbf{u}$ as a probability distribution over instance categories per part. At test time, we predict the instance category by taking the category prediction with the highest confidence from the category predictions of the parts belonging to the same instance.

Extension to sequential joints Although the part structure with sequentially connected parts is out of the scope of this Chapter, we show that the proposed part grouping strategy is applicable to such part configuration. Inspired by the linked list data structure, we employ two embeddings per part for grouping; $\mathbf{z}_i^s$ represents the self-identity of the part and the other embedding $\mathbf{z}_i^p$ represents the identity of its parent part. We formulate this association as:

$$j^*|i = \underset{j^* \in G \setminus i}{\operatorname{argmin}} \|\mathbf{z}_i^p - \mathbf{z}_j^s\| \tag{4.2}$$

where $j^*|i$ denotes part i's parent part j^*. The base part has a parent identity embedding pointing to a special embedding $\mathbf{0}$ representing the scene root. We illustrate this embedding-based sequential association in Figure 4.5.

4.3.5 Refiner $\mathscr{R}$

Query features are iteratively refined in the decoder $\mathscr{D}$ in the feature space [80]. Increasing the number of decoder layers can enhance performance at the expense of a larger model size [80]. However, it is crucial to avoid increasing the model size to enable efficient end-to-end training from detection to shape reconstruction. To improve model performance while maintaining the model size, we found cross-refining both output space and feature space to be effective. We introduce a refiner $\mathscr{R}$, which has an identical architecture with the decoder $\mathscr{D}$. The refiner $\mathscr{R}$ serves to refine the prediction in output space by reallocating a portion of decoder layers from decoder $\mathscr{D}$ to $\mathscr{R}$. Assuming there are $N_{\mathscr{D}+\mathscr{R}}$ decoder layers in the original model, we reallocate $N_{\mathscr{R}}$ decoder layers to the refiner $\mathscr{R}$. Consequently, $N_{\mathscr{D}} = N_{\mathscr{D}+\mathscr{R}} - N_{\mathscr{R}}$ layers are used for query feature refinement in decoder $\mathscr{D}$. The refiner $\mathscr{R}$ refines part pose and size B' and joint parameter A' from $\mathscr{D}$ by predicting residuals ΔB and ΔA to produce refined prediction B and A, respectively. The architecture of the refiner $\mathscr{R}$ is shown in Figure 4.3.

4.3.6 Kinematics-aware part fusion (KPF)

Articulated objects often consist of small and thin parts. Especially for unsegmented inputs, only a small portion of the point cloud represents such parts after subsampling. To ensure a sufficient number of queries cover such parts and their surrounding context, at test time, we randomize the sampling positions of the queries and independently carry out inference and NMS for the input N_Q times, obtaining a collection of densely sampled part proposals from N_Q inference runs. This process is termed Query Oversampling (OQ). However, OQ can increase false positives and degrade detection performance. To mitigate this, we propose Part Fusion (PF), inspired by Weighted Box Fusion (WBF) [117], to merge overlapping part proposals, thereby reducing false positives and improving detection accuracy. Unlike WBF, which fuses only 2D bounding parameters, PF fuses all the predicted parameters of part proposals, with an average weight defined as objectness $1 - \mathbf{y}_{\text{bg}}$. However, using IoU with 3D bounding boxes as overlapping metrics yields overly small values for thin structured parts, even when they are proximate. This results in PF failing to merge redundant parts, leading to false positives, as illustrated in Figure 4.4 (a). To overcome this challenge, we propose a kinematics-aware IoU (kIoU) based on the observation that a redundant part pair exhibits significantly overlapping trajectories. To calculate kIoU, we construct a convex hull for each part in the pair using the 24 vertices of three bounding boxes based on size and canonical pose, current pose, and fully opened pose using the predicted part pose and size B and joint parameter A, as show in Figure 4.4 (b). Then, we calculate the IoU between

the two hulls. The comparison between the 3D bounding box IoU and kIoU is depicted in Figure 4.4 (c). Our overall method, termed Kinematic-aware Part Fusion (KPF), includes: (1) executing inference and NMS using kIoU to gather part proposals multiple times, (2) conducting PF using kIoU to update these proposals iteratively, and (3) removing proposals with low objectness. Note that KPF is non-differentiable and is disabled during training, with $N_Q = 1$.

We show the detail of KPF algorithm in 1. In Algorithm 1, we present the details of kinematics-aware part fusion. We gather part proposals through N_Q independent inference runs, achieved by randomly sampling query positions $\{\mathbf{q}_n\}$ with the furthest point sampling (FPS). The 'model' in the algorithm represents the decoder $\mathscr{D}$ and the refiner $\mathscr{R}$, and $\mathbf{y}_{\text{fg}}$ represents objectness, defined as $1 - \text{bg}$.

After part proposals are generated, we apply Non-Maximum Suppression (NMS) with 3D bounding box IoU and threshold by objectness $1 - \mathbf{y}_{\text{bg}}$ before applying NMS with kinematics-aware IoU (NMS-kIoU) for performance reasons. PF-kIoU denotes the part fusion (PF) process using kIoU, which is based on the Weighted Box Fusion (WBF) approach as described in [12]. The procedure is further detailed in Algorithm 2.

Unlike the conventional WBF, our algorithm fuses all parameters in the part proposal rather than only the 2D bounding box parameters. Furthermore, our algorithm iteratively runs until convergence, as demonstrated from line 11 to line 15 in Algorithm 1. C represents clusters of overlapping part proposals identified by kIoU. We employ a simple arithmetic weighted average for all elements except the rotation matrix in a part proposal for fusion, denoted as weighted-average.

For the rotation matrix $\mathbf{R} \in \text{SO}(3)$, we initially derive a weighted average for the 3×3 matrices, labeled as $\mathbf{R}_{\text{wa}} \in \mathbb{R}^{3\times3}$. We then compute the weighted-averaged rotation matrix by minimizing the Frobenius norm to $\mathbf{R}_{\text{wa}}$. This is accomplished as $\mathbf{R} = \text{argmin}_{\mathbf{R}} \|\mathbf{R} - \mathbf{R}_{\text{wa}}\|_F$, following the 3D rotation library RoMa [12].

The value of $\mathbf{y}'_{\text{fg},n'}$ denotes the scaled objectness as a confidence score, considering the number of independent inferences N_Q. If the number of part proposals in a cluster is fewer than the independent inference runs N_Q, it indicates that only a limited number of independent inferences predict it. In such cases, we adjust the corresponding confidence score $\mathbf{y}'_{\text{fg},n'}$ by scaling it down by the ratio of the number of part proposals in the cluster ($|C[n']|$) to the number of independent inferences $T = N_Q$. Conversely, we scale up the confidence score if a cluster contains more part proposals than the independent inferences N_Q. For the initial run of part fusion (line 13 of Algorithm 1), we set $T = N_Q$ to fuse N_Q inferences. We assign $T = 1$ for subsequent runs, assuming we apply part fusion to a single inference run fused by the previous part fusion.

Algorithm 1 Kinematic-aware part fusion (KPF)

Input: Subsampled point cloud P', scene feature F'
Output: Set of detected parts $\mathscr{X} = \{X_i\}$

$\mathscr{X} \leftarrow \emptyset$
for N_Q times **do**
  $\{\mathbf{q}_n\} \leftarrow \text{FPS}(P')$
  $\mathscr{X}' \leftarrow \text{model}(\{\mathbf{q}_n\}, F')$
  $\mathscr{X}' \leftarrow \text{NMS}(\mathscr{X}', \tau_{\text{IoU}})$
  $\mathscr{X}' \leftarrow \{X_n \mid n \in [\mathscr{X}'], \mathbf{y}_{\text{fg},n} > \tau_{\text{obj}}\}$
  $\mathscr{X}' \leftarrow \text{NMS-kIoU}(\mathscr{X}', \tau_{\text{IoU}})$
  $\mathscr{X} \leftarrow \mathscr{X}' + \mathscr{X}$
end for
count $\leftarrow 0$
repeat
  $\mathscr{X}_{\text{old}} \leftarrow \mathscr{X}$
  $\mathscr{X} \leftarrow \text{PF-kIoU}(\mathscr{X}_{\text{old}})$
  count $\leftarrow$ count $+ 1$
until $\mathscr{X} = \mathscr{X}_{\text{old}}$ or count $= \tau_{\text{count}}$
$\mathscr{X} \leftarrow \{X_n \mid n \in [\mathscr{X}], \mathbf{y}_{\text{fg},n} > \tau_{obj_final}\}$
return $\mathscr{X}$

4.3.7 Set matching and training loss

Set matching We base our end-to-end training of detection to reconstruction by utilizing 1-to-1 matching between part proposals and ground truth, using bipartite matching [14] for loss calculation. Similarly to [80], we define a matching cost between a predicted part and a ground truth part as $C_{match} = \lambda_1 \|\mathbf{B} - \mathbf{B}^{\text{GT}}\|_1 + \lambda_2 \|\mathbf{c} - \mathbf{c}^{\text{GT}}\|_1 - \lambda_3 \mathbf{y}_y + \lambda_4 (1 - \mathbf{y}_{\text{bg}})$, where $\mathbf{B}$ defines eight vertices of cuboid defined by part pose and size B, $\mathbf{y}_y$ defines the joint type probability given the ground truth label y, and $1 - \mathbf{y}_{\text{bg}}$ defines the foreground probability. Deviating from [80], we use the L1 distance of eight vertices of a cuboid instead of the GIoU [105] of cuboids in the first term to avoid the costly calculation of enclosing hull for 3D rotated cuboids.

Training losses For each pair of prediction and ground truth, we define part loss as

$$\begin{aligned}\mathscr{L}_{\text{part}} = \frac{1}{N}\sum_{i}^{N} &\|\mathbf{B}_i - \mathbf{B}_i^{\text{GT}}\|_1 + \|I - \mathbf{R}_i^T \mathbf{R}_i^{\text{GT}}\|_F^2 + \mathbb{E}_{\mathbf{x}\sim\mathbb{R}^3}\text{BCE}(o_{\mathbf{x},i}, o_{\mathbf{x},i}^{\text{GT}}) + \|d_{\max,i} - d_{\max,i}^{\text{GT}}\|_1 \\ &+ \|d_{\text{current},i} - d_{\text{current},i}^{\text{GT}}\|_1 - \mathbf{a}_i^T \mathbf{a}_i^{\text{GT}} + \text{PL}(\mathbf{v}_i, \mathbf{v}_i^{\text{GT}}, \mathbf{a}_i^{\text{GT}}) + \text{CE}(\mathbf{y}_i, \mathbf{y}_i^{\text{GT}}) + \text{CE}(\mathbf{u}_i, \mathbf{u}_i^{\text{GT}})\end{aligned} \tag{4.3}$$

Algorithm 2 Part Fusion with kIoU (PF-kIoU)

Input: Set of part proposals $\mathscr{X}$
Output: List of updated part proposals $\mathscr{X}'$

$C \leftarrow$ Empty list, $\mathscr{X}' \leftarrow$ Empty list
Sort $\mathscr{X}$ in desceding order by objectness
for $\forall X \in \mathscr{X}$ **do**
  match $\leftarrow$ False
  for $\forall X' \in \mathscr{X}'$ **do**
    if $\text{kIoU}(X, X') > \tau_{\text{kIoU}}$ **then**
      matched $\leftarrow$ True
      $n' \leftarrow$ index of X' in $\mathscr{X}'$
      Append X to $C[n']$
      $\mathscr{X}'[n'] \leftarrow$ weighted-average$(C[n'])$
      Break
    end if
  end for
  if not match **then**
    Append X to $\mathscr{X}'$ and C
  end if
end for
for $\forall n' \in [\mathscr{X}']$ **do**
  $\mathbf{y}'_{\text{fg},n'} \leftarrow \mathbf{y}_{\text{fg},n'} \frac{|C[n']|}{T}$
end for
$\mathscr{X}' = \{X'_{n'} | n' \in [\mathscr{X}'], \mathbf{y}'_{\text{fg},n'} > \tau_{\text{scaled}}\}$
return $\mathscr{X}'$

All loss terms have equal weights. The first term is a disentangled L1 loss described in [114] to optimize **B**. This loss is replicated three times by using only one of the predicted three components (**R**,**c**,**s**) for **B**, while replacing the other three with their ground truth values. We also found that directly optimizing rotation, as in the second term of the loss, leads to smaller rotation loss during training. BCE and CE denote binary cross-entropy and cross-entropy loss, respectively. PL denotes the point-line distance between the revolute origin $\mathbf{v}_i$ and the ground truth joint axis defined by $\mathbf{v}_i^{\text{GT}}$ and $\mathbf{a}_i^{\text{GT}}$. For unrefined prediction B' and A', we define the same loss as $\mathscr{L}_{\text{part}}$ except for $o_{\mathbf{x}}$, $\mathbf{y}_i$ and $\mathbf{u}_i$ denoted as $\mathscr{L}'_{\text{part}}$. During training, we use ground truth B for Eq.4.1 to avoid noisy prediction adversely affecting shape learning.

We also define instance loss $\mathscr{L}_{\text{instance}}$ for learning part-to-instance association with modified improved triplet loss [20] defined as:

$$\mathscr{L}_{\text{instance}} = \lambda_{\text{intra}} \sum_{i \in G} \frac{1}{|G_{l|i}|} \sum_{j \in G_{l|i} \setminus i} [\eta_{ij} - \tau'_{\mathbf{z}}]_+ + \frac{1}{N} \sum_{i \in G} [\max_{j \in G_{l|i} \setminus i} \eta_{ij} - \min_{j' \in G \setminus G_i} \eta_{ij'} + 3\tau'_{\mathbf{z}}]_+ \quad (4.4)$$

where $G_{l|i} = \{j \in G_l \mid \delta_{li} = 1\}$ and $\eta_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|$ denotes the L2 distance between the part-to-instance association embeddings of the i-th and j-th parts. The first term enforces the distance between the two embeddings of two different parts belonging to the same instance below $\tau'_{\mathbf{z}}$ and the second term ensures the distance between embeddings of parts belonging to different instances is larger than $3\tau'_{\mathbf{z}}$. However, computing all combinations of triplets for the second term is operationally complex for part-wise supervision. To streamline this, we instead opt to maximize the difference between the upper bound and the lower bound of inter- and intra-instance distances of the embeddings. In practice, we replace max and min with their soft approximations defined as $\text{LogSumExp}(\eta_{ij})$ and $-\text{LogSumExp}(-\eta_{ij'})$ for smooth gradient propagation. During inference, we use a threshold $\tau_{\mathbf{z}} = \frac{1}{2}(3\tau'_{\mathbf{z}} + \tau'_{\mathbf{z}})$ to determine if two parts belong to the same instance based on the distance between their embeddings. For extending to the sequential part structures, we replace η_{ij} with $\eta^*_{ij} = \|\mathbf{z}^p_i - \mathbf{z}^s_j\|$. In the experiments in Section 4.4, we use η_{ij} except in Section **??**.

The total loss we minimize is $\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{part}} + \mathscr{L}'_{\text{part}} + \mathscr{L}_{\text{instance}}$. At training time, we use the same part prediction MLPs to predict part parameters at every layer in the decoder. We compute the $\mathscr{L}_{\text{total}}$ for each layer independently and sum all the losses. We only use the part parameter predicted from the last decoder layer at test time. The comprehensive loss formulation is as follows:

$$\mathscr{L}_{\text{total}} = \sum_{k=1}^{N_{\mathscr{D}}} \mathscr{L}_{\text{part},k} + \mathscr{L}'_{\text{part},k} + \mathscr{L}_{\text{instance},k}. \tag{4.5}$$

4.3.8 Implementation detail

Architecture Our approach strictly follows the hyperparameters of the masked-encoder described in [80]. We employ three self-attention transformer layers with self-attention masks between points in the point cloud, and each point only attends to others within a specified radius. The input point cloud to the encoder consists of $N_P = 32768$ points, which are subsequently subsampled to $N_{P'} = 2048$ points in the encoder. We set the scene feature dimension $D_{\mathbf{F}}$ a value of 256, following [80].

For the decoder, we maintain the hyperparameters from [80], except for the number of decoder layers $N_{\mathscr{D}}$. The decoder layers for the decoder $\mathscr{D}$ and the refiner $\mathscr{R}$ are set to $N_{\mathscr{D}} = 6$ and $N_{\mathscr{R}} = 2$, respectively. We match the query embedding dimension with the scene feature dimension for addition and set the number of queries to $N_q = 128$ during training. For testing, N_q is set to 512, unless stated otherwise, with independent runs for query oversampling (QO) for kinematics-aware part fusion (KPF) with $N_Q = 10$ independent inference runs.

As for the part prediction Multi-Layer Perceptrons (MLPs), we follow the box prediction MLP hyperparameters in [80], except for the number and output dimension of each MLP. The MLPs in the refiner for residual prediction strictly follow the part prediction MLPs hyperparameters in the decoder but without dropout. We set the dimension for shape feature $\mathbf{h} = 128$, and the dimension for part-to-instance association embedding $\mathbf{z} = 32$.

With respect to the ConvONet [98] architecture in the shape decoder, the local point encoder consists of five MLPs of width 128, and the volume encoder has three MLPs with widths of 16, 32, and 64. We employ the tri-plane version of the volume encoder for volume representation, using only the xy and yz planes to save GPU memory. In the implicit shape decoder, we employ four fully-connected layers with a hidden dimension of 128 and a leaky ReLU activation.

Training details We set the weights for the matching cost $\mathscr{C}_{\text{match}}$ at $\lambda_1 = 8, \lambda_2 = 10, \lambda_3 = 1, \lambda_4 = 5$. Each loss term in the total loss $\mathscr{L}_{\text{total}}$ carries equal weight. We use the AdamW [71] optimizer with a base learning rate of 9e-4 with a cosine scheduler down to a learning rate of 1e-6. The warm-up period is set to nine epochs, and mixed precision is used during training.

All models were trained on two A100 GPUs, each with 40GB GPU memory. We set the batch size to 26 per GPU, and it took approximately two days to complete 500 epochs. Weight decay was set at 0.1, with gradient clipping with an L2 norm of 0.1, as per [80].

We implemented on-the-fly sampling of 3D points and corresponding occupancy values to train the shape decoder. Half of these points are sampled uniformly from the space $[-0.5, 0.5]^3$. Additionally, a quarter of the points are sampled around the surface with a Gaussian-distributed random offset along the surface normal direction, with a standard deviation of 0.1. The remaining quarter is sampled with a standard deviation of 0.01. Each part has 128 points sampled for occupancy values.

In training the foreground segmentation model, we used the AdamW optimizer and a cosine scheduler, identical to the approach used in training the previous models. This model was also trained on two A100 GPUs with 40GB GPU memory and a batch size of 26 per GPU. Training took approximately one day for 500 epochs.

We add noise to the depth map during training, following the approach in [73]. We incorporate a depth map filter for the model tested on real-world data to mitigate the flying pixel effect on the object edge as suggested by [129]. This filter is also applied during testing for real-world data. Following the projection of the depth map to 3D points, we introduced random scaling within a range of $\pm 15\%$, and random rotation along the surface normal direction within $\pm 30°$. The point cloud was zero-centered for both training and testing.

	F-Score@80% ↑	F-Score@90% ↑	CD@5% ↑	CD@1% ↑	IoU@25% ↑	IoU@50% ↑
A-SDF-GT [83]	65.49	47.69	74.60	39.76	36.62	10.81
A-SDF-GT-2 [83]	68.81	52.55	75.91	43.58	38.59	10.43
Ours-BG	74.22	**68.80**	75.71	**58.61**	40.06	9.80
Ours	**74.77**	68.38	**77.39**	56.53	**41.35**	**11.63**

Table 4.1 Shape mAP results on SAPIEN [133] dataset.

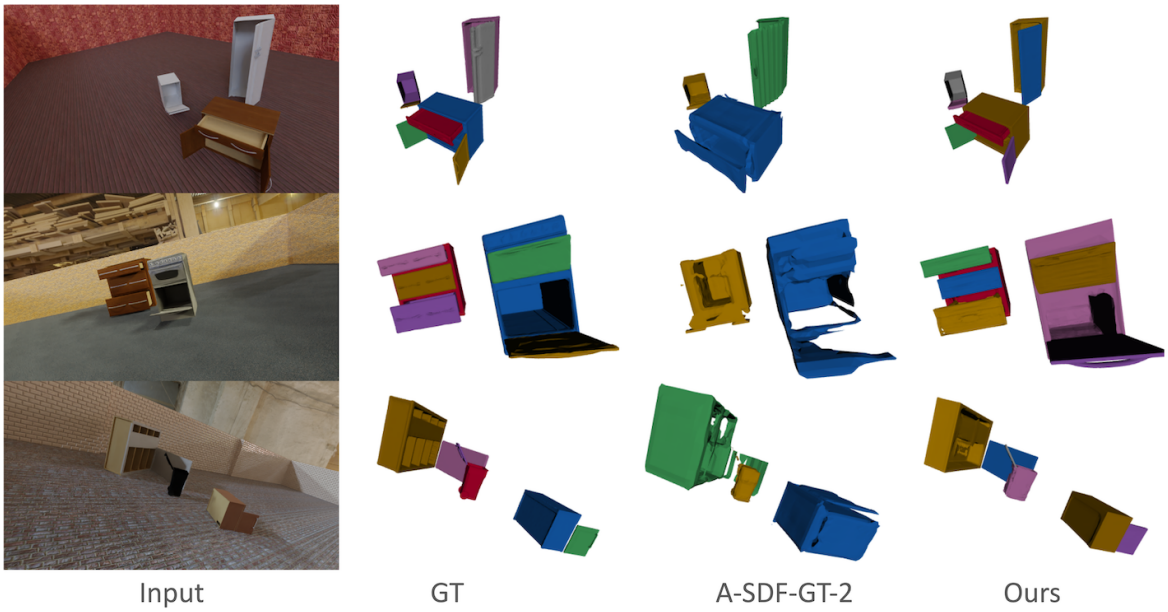


Figure 4.6 Qualitative results on SAPIEN [133] dataset.

Color augmentation during training follows the original code of [80]. For the model using a foreground mask, we augment the mask by randomly displacing the foreground pixels around the border and performing successive dilation and erosion to simulate noise on inferred masks during training.

Mesh generation We sample occupancy values on 64^3 voxel grids per part, then extract the surface mesh using the marching cubes method [70] at an isosurface level $\tau_{\mathcal{O}} = 0.3775 = \text{Sigmoid}(-0.5)$. For surface mesh reconstruction evaluation, we store an instance mesh as a union of part meshes and apply quadratic decimation [31] to reduce the number of faces to 10000. For volumetric IoU evaluation, where we evaluate the union of IoU for each part, we stored a part mesh and applied quadratic decimation to reduce the number of faces to 5000.

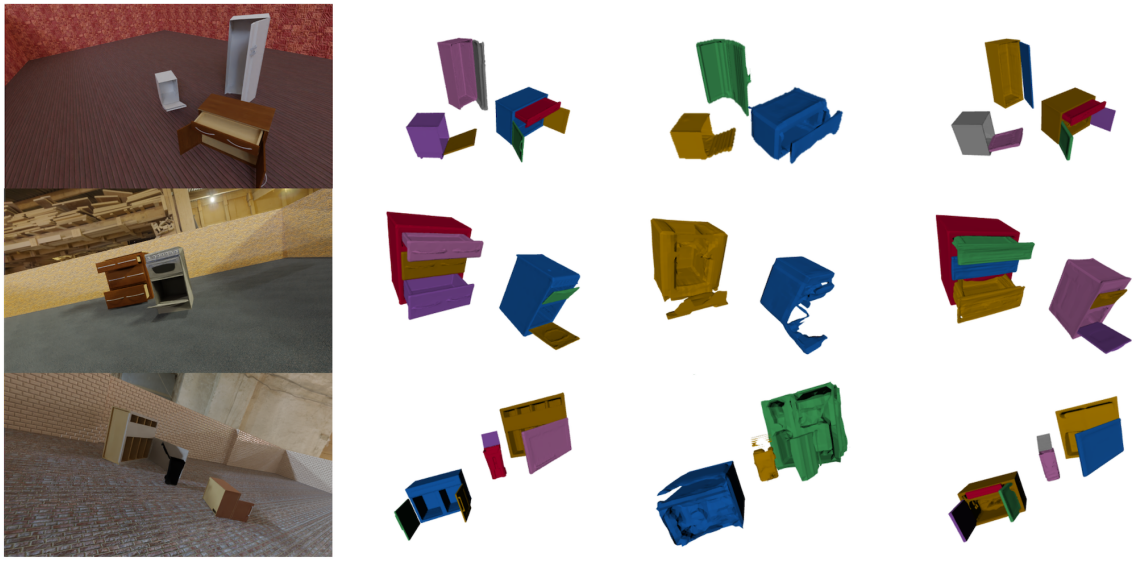


Figure 4.7 Qualitatative results on SAPIEN [133] from novel viewpoint.

4.4 Experiments

4.4.1 Datasets

We evaluate our method on both synthetic and real-world data. We use the SAPIEN [133] dataset for synthetic data evaluation, following recent works on articulated shape reconstruction [53, 130, 38]. We select ten categories with representative articulation types, a sufficient number of CAD models, and various part structures across categories. We then randomly construct room geometry (wall and floor) and place one to four instances per scene. Each instance is generated by applying random horizontal flips, random anisotropic resizing of side lengths, and random articulation of CAD models. The camera pose is sampled randomly, covering the upper hemisphere of the scene. Instances with severe truncation from the view frustum or occlusion with other instances are ignored during training and evaluation, ensuring that least one instance is visible in a view. For the training split, we randomize the textures of parts and room meshes. We use the original textures from the SAPIEN dataset for the test split. We generated 188,726 images for training and validation. Due to computational and time constraints, we used 20,000 images for training and kept the rest for validation usage. Also, we generated 4,000 images for the test split. Image size is 360×640 in height and width. The data overview is shown in Table 4.2.

For real-world data, we use the BMVC [78] dataset for quantitative evaluation. We use cabinet class which includes both prismatic and revolute parts and has the same part count

Category	Joint type		Part count			Instance count	
	Rev.	Pris.	Most freq.	Min	Max	Train	Test
Dishwasher	✓	✓	1	1	2	4997	1032
Trashcan	✓	✓	1	1	2	5037	1017
Safe	✓		1	1	1	4795	1029
Oven	✓		1	1	3	4951	1006
Storage	✓	✓	1	1	14	4907	973
Table	✓	✓	1	1	9	4790	999
Microwave	✓	✓	1	1	2	4832	934
Frige	✓		2	1	3	5089	924
Washing	✓		1	1	1	5042	995
Box	✓		1	1	4	4823	979

Table 4.2 Overview of synthetic data from SAPIEN [133] dataset. Rev. and Pres. denote revolute and prismatic joints, respectively.

and similar object shape as those used in our training and baseline models. The data contains two sequences of RGBD frames, capturing the same target objects with different part poses from various camera poses. We also test our method on RGB images taken with an iPhone X and depth maps generated from partial front views of the scene using Nerfacto [120].

Toy data For an ablation experiment on the effect of part count distribution in the dataset, we use the toy dataset where each CAD model is generated procedurally. Specifically, given uniformly sampled numbers of articulated parts ranging from one to six, part layout is randomly generated with a random assignment of either a revolute or prismatic joint to each part. We also randomize the size and pose of each part. We generated 10,000 images for training and 2,000 images for testing. For simplicity of the experiment only focusing on the variation of part count distributions, we use a single texture and shape. Note that the size of each part is randomized, and the size of the whole shape is also randomized depending on randomly sampled part count and part layout of the CAD models. We name this dataset *procedural dataset*. We visualize the generated data in 4.8.

In this section, we also verify the extendability of metric-learning-based part grouping to man-made articulated objects with sequentially connected parts. Due to the lack of such data in SAPIEN [133] and BMVC [78] datasets, we generate a point cloud toy dataset containing multiple instances with sequentially connected parts in a scene. Each object has one base

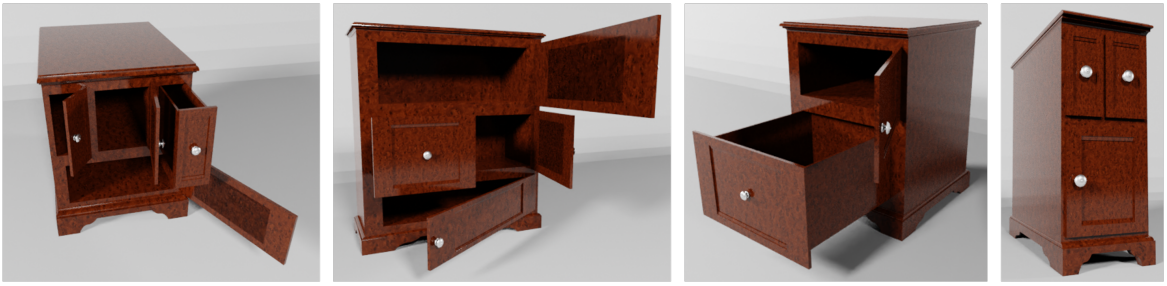


Figure 4.8 Visualization of procedurally generated CAD models of articulated objects.

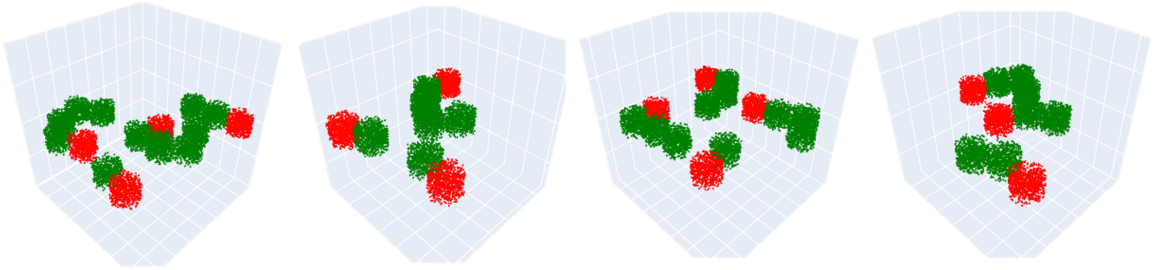


Figure 4.9 Visualization of toy data with sequentially connected parts. Red color indicates the base parts and green color indicates the sequentially connected child parts.

part and sequentially connected parts. For simplicity of the experiment solely focusing on part grouping capability in a set prediction setting, we fix the size and shape of all parts. The base part and child parts have different RGB features for ease of detection. The number of non-base parts are randomly sampled between one and four. Note that we assume each part has up to one child part. Each scene has up to four instances. We name this dataset *sequential dataset*. We visualize the data in Figure 4.9.

4.4.2 Metrics

Shape evaluation For shape evaluation, following detection-based object reconstruction studies [89, 121, 37], we report mAP combined with standard shape evaluation metrics denoted as metric@threshold. We use F-Score [122], Chamfer distance (CD), and volumetric IoU (IoU) with multiple matching thresholds. F-Score proposed in [122] evaluates percentage of correctly reconstructed mesh surface. Specifically, it calcurates harmonic mean of two percentages: surface points whose distance from neareset points on the other surface are within the threashold measuread from ground truth surface to reconstructed surface and opposite direction, respectively. We use 1% of ground truth part's diameter as a threshold and we sample 10000 points for both ground truth surface and reconstructed mesh surface. In shape mAP, we use the percentage of F-Score as threashold to evaluate the match against

ground truth. For chamfer distance, we use L2-chamfer distance in our evaluation. Our volumetric IoU is calcularted on 64^3 voxel grids. The proposed method outputs part wise meshes, thus we evauate union of each IoU result of a part by logical OR.

The shape mAP evaluation includes all the predicted parameters except for part kinematic parameters, serving as a comprehensive proxy for their quality. For part-wise detection with part pose and size, we use the average L2 distance between the corresponding eight corners of bounding boxes between the ground truth and prediction as a matching metric of mAP, similar to the ADD metric [39], denoted as mAP@threshold. The distance is normalized by the ground truth part's diameter, with a proportion of the diameter used as the threshold. Since IoU-based metrics can yield values close to zero for thin structured parts, even when they are reasonably proximate and slightly off the ground truth, and do not consider part orientation, we use the above matching metrics to better analyze part-wise detection. Note that in shape mAP with chamfer distance, we regard a reconstructed mesh matches to the ground truth if the chamfer distance is below the proportion of ground truth part's diameter as a threshold.

Kinematics evaluation For part kinematics evaluation, we evaluate the absolute error on joint state (State), Orientation Error (OE) for joint direction, and Minimum Distance (MD) for rotation origin following [48] for the detected parts with matched ground truth. To compare 2D detection result of OPD [48] and our 3D detection, we evaluate the interserction of matched ground truth. We chose detection result of mAP@50% of segmentation result for OPD and our mAP@70%. Note that, we use the average L2 distance between the corresponding eight corners of bounding boxes between ground truth and prediction as a matching metric of mAP similar to ADD metric [39]. The distance is normalized by the ground truth part's diameter, and we use 70% as a threshold here. This results in 8501 detections for OPD, 10431 detections for Ours-BG, 10651 for Ours excluding base parts, and intersection has 7126 matches for the result reported in Section 4.4.5. Threashold 80% has more similar number of detections of ours (9449 for Ours-BG, and 9720 for Ours) to OPD but it has less number of interserctions (6549) thus we chose 70% as our mAP's threashold.

4.4.3 Baselines

By default, we denote the proposed method that takes the foreground mask as 'Ours', and the model taking unsegmented input as 'Ours-BG'. To the best of our knowledge, no prior work operates on exactly the same problem setting as ours. For shape reconstruction, we benchmark against the state-of-the-art category-level object reconstruction method A-SDF [83], the closest to our approach. We follow the most recent work setup [38] in evaluation.

For kinematic evaluation, we compare our method with the state-of-the-art single-view part kinematic estimation method OPD [48]. Note that we input OPD with an RGBD image to align input modality and modify it to output joint state. Below, we discuss further details on the baselines.

A-SDF [83]

Data preparation We follow the description in the original A-SDF paper [83] for data preparation. After normalizing the global pose and scale, we ensure the base part's position does not change regardless of random articulation during training. In addition, for categories with multiple parts, we maintain consistent part ordering. We automate this by assigning part positions in canonical poses to a cell in the $5 \times 5 \times 5$ voxel, which discretizes the positions. The 1D flattened cell positions of the voxel determine the part order. To limit the training time, we randomly sample a maximum of 1500 instances per category for the training split.

Model The model implementation uses publicly shared author code. The original A-SDF only targets revolute parts. We normalize the prismatic joint state by the maximum joint state to fall within the $[0, 1]$ range. We then multiply this range by 135, ensuring joint states for revolute and prismatic parts are within the same range to extend the model for prismatic parts.

A-SDF is a category-level approach that works with a fixed number of parts, which becomes a problem when evaluated on dataset containing instances with varied part counts in a category. Similarly to our approach, OPD [48] can handle multiple part counts without assuming the predefined part number. They evaluate their method against baselines targeting a fixed number of parts by training the baselines for a category's most common part count.

We follow this protocol but aim to favor the baseline A-SDF more by training up to two models per category. The first model is for the most common part count, and the second, if applicable, is for the next most common part count. The second model is used when a category has more than two different part counts in the test split.

This setting differs from OPD's method, which only uses the most common part count. During testing, we use the model with the most frequent part count for instances with untrained part counts. We call the baseline model trained for the most frequent part count A-SDF-GT and the one using the next most frequent part count A-SDF-GT-2. The baselines and per category part counts, excluding the base part, are summarized in Table 4.3.

	Dishwasher	Trashcan	Safe	Oven	Storage	Table	Microwave	Frige	Washing	Box
Most freq.	1	1	1	2	1	1	1	2	1	1
Next Most freq.	n/a	2	n/a	n/a	2	n/a	2	1	n/a	4

Table 4.3 The most frequent and the next most frequent part count for each category that baseline A-SDF models are trained on.

Training We employ the publicly shared author code for training. During training, each part's articulation varies randomly from 0°to 135°, sampled at 5°intervals as suggested in [83]. We dynamically sample 3D points and their corresponding signed distance values from the randomly articulated watertight meshes of parts during training. The number of sample points and the ratio of uniformly sampled points to points near the mesh surface follow the original A-SDF paper [83]. A-SDF model training uses part label supervision and takes approximately ten days on a single V100 or A100 GPU per model.

Mesh generation For mesh generation, we use the publicly shared author code as well. A-SDF assumes no background and normalized global pose and size for input. During testing, we use the ground truth instance segmentation mask for each instance in a scene to isolate instance-wise foreground points. We also normalize the depth map to the input space using the ground truth pose and size. The surface mesh is extracted using the provided implementation, which samples signed distance values on 64^3 voxel grids and extracts surface mesh using marching cubes [70]. Additionally, we apply the quadratic decimation [31] to limit the number of faces to 10000. We then project the generated mesh back to the original scale and pose in the scene using the ground truth values.

OPD [48]

Data Following the original paper [48] and the publicly shared author code and dataset, we generate ground truth data. The original data includes a semantic label of part type, one of drawer, lid, or door. However, our synthetic dataset does not have corresponding labels. Thus we replace the semantic label with the joint type. Consistent with the original paper, we exclude the fixed type part from the ground truth, targeting only the revolute and prismatic parts for detection.

Model We employ the OPDRCNN-C model from [48], which predicts kinematic parameters in camera coordinates. For input modality alignment, the model uses RGBD input. We have adapted the model to produce a continuous 1D joint state; instead of supervising the

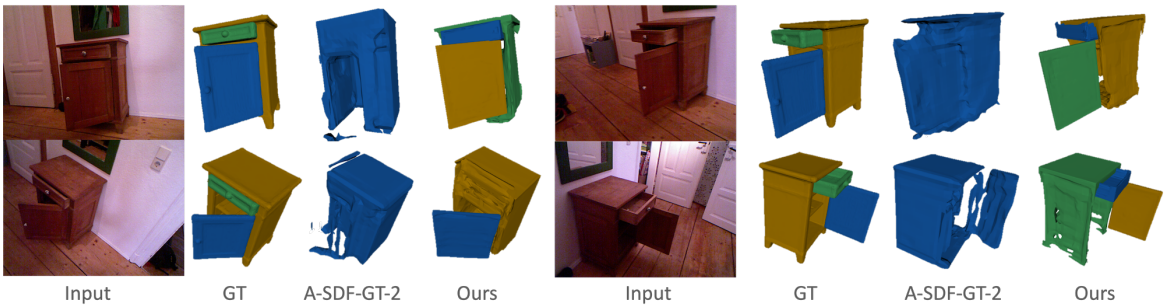


Figure 4.10 Qualitative results on the BMVC [78] dataset.

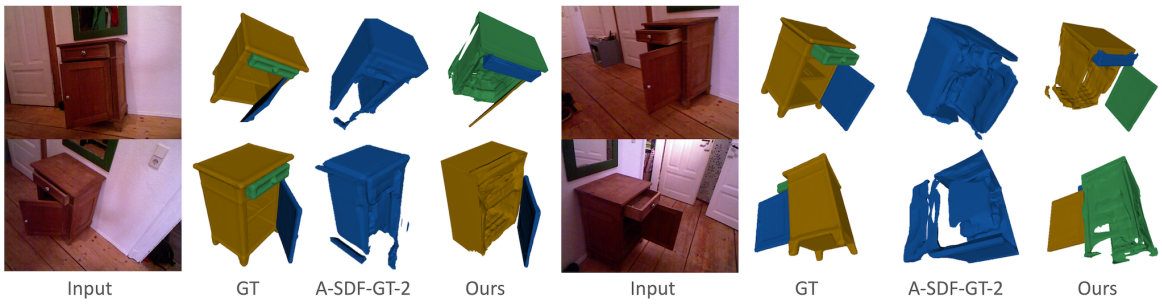


Figure 4.11 Qualitatative results on BMVC [78] from novel viewpoint.

joint state prediction head with binary joint state supervision, we use continuous 1D joint state supervision. We duplicate the MLP head to separately predict joint states for revolute and prismatic joints. We supervise only the output corresponding to the ground truth joint type. Furthermore, we guide the joint axis prediction with reference to the floor's normal direction, as we experimentally found this improves detection performance.

Training We follows the author's implementation for training code, maintaining the same settings and hyperparameters for the OPDRCNN-C model.

Testing As described in Sec. 4.4, we match the ground truth using part segmentation evaluation with an IoU threshold of 50% following the author's implementation. We then evaluate the predicted joint parameters against the matched ground truth.

4.4.4 Shape reconstruction

We show the shape mAP result in Table 4.1. Our method outperforms A-SDF [83] with ground truth in all metrics, and Ours-BG outperforms in the majority of metrics. We show the qualitative results in Figure 4.6. Our models effectively reconstruct multiple instances with diverse part counts and structures, outperforming A-SDF which struggles with

reconstructing articulated parts. We attribute our method's superior performance to its part-level representation, which facilitates easier learning of various part structures. Moreover, our shape-decoder is less affected by shape variations arising from the combination of part structures and poses. Corresponding to Figure 4.6, we visualize the qualitatative results from novel viewpoint on SAPIEN [133] dataset in Figure 4.7

4.4.5 Kinematic estimation

OPD [48] performs 2D detection evaluated by 2D IoU, while ours on 3D by L2 distance between 3D bounding box corners. To make kinematics estimation results of OPD and ours comparable, we experimentally choose the matching threshold of our mAP as 70% so that a similar number of detected parts with OPD's result with 2D segmentation mAP@50%. Then, we select the intersection of true positive detected parts by OPD and ours. The result is shown in Table 4.5. Our methods outperform OPD significantly. We attribute the reason to our method operating on 3D point clouds while OPD is on 2D. Thus, our method is more robust to various textures and lighting conditions and makes it easier to reason about 3D kinematics. Note that our focus here is kinematics estimation after the *detection* step. Thus, superiority in detection performance is not our focus.

4.4.6 Ablation studies

In the following ablation studies, we validate each proposed component in a challenging setting where the input to the encoder has an unsegmented background using Ours-BG.

Kinematics-aware part fusion We show quantitative results in Table 4.6. Besides mAP, we also show precision considering false positives. QO denotes test-time query oversampling, and PF indicates part fusion. As a baseline, we first disable all components (w/o QO, PF, kIoU) using the same number of queries $N_q = 128$ during training and add each component one by one. Introducing our proposed kIoU on top of QO and PF outperforms the baseline in shape reconstruction mAP and part detection while preserving similar precision. We visualize the effectiveness of the KPF module in Figure 4.12. In the provided comparison, 'w/o KPF' denotes disabling QO, PF and kIoU. We see that KPF enhances the detection and pose estimation of small parts. We also show the qualitative results on the proposed kIoU in Figure 4.13. Applying QO and PF alone leads to false positives of thin parts, which kIoU effectively reduces. It indicates that the proposed KPF module improves overall detection performance while suppressing false positives.

Figure 4.12 Qualitative results on kinematics-aware part fusion (KPF).

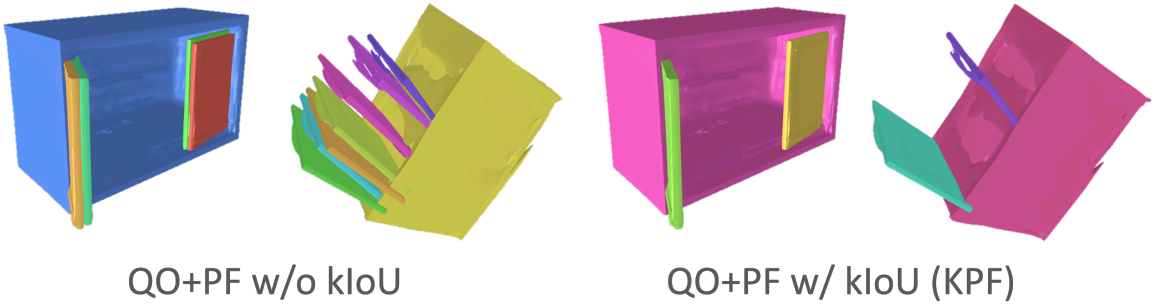


Figure 4.13 Qualitative results on kinematics-aware IoU (kIoU).

Anisotropic scaling and end-to-end training for shape learning We validate the effect of anisotropic scale normalization and end-to-end training in shape mAP evaluation. In Table 4.7, w/o AS denotes using *isotropic* scaling by normalizing the *maximum* side length to one instead of *all* sides. w/o SF denotes not passing shape feature **h** but training shape decoder $\mathscr{O}$ separately. Disabling each component degrades performance. Especially disabling anisotropic scaling significantly drops the performance, as the single shape decoder is tasked to decode various sizes of target shapes.

Ratio of decoder layers in the refiner $\mathscr{R}$ We investigate the relationship between the proportion of decoder layers in $\mathscr{R}$ and performance in shape mAP. The result is shown in Figure 4.17. We vary the ratio of decoder layers in the refiner $N_{\mathscr{R}}/N_{\mathscr{D}+\mathscr{R}}$ from 0% to 75%. Allocating a portion (25%) of decoder layers to the refiner improves performance with the same number of decoder layers while reducing excessive decoder layers from the decoder degrades performance.

Comparison on the number of learnable parameters in the refiner $\mathscr{R}$ We evaluate the effectiveness of the refiner $\mathscr{R}$ in enhancing performance while maintaining a comparable

	Fscore@80%	Fscore@90%	CD1@5%	CD1@1%	IoU@25%	IoU@50%	Params.
$N_{\mathscr{D}} = 16$	73.25	68.27	74.76	**58.64**	**40.74**	**10.57**	14.33M
$N_{\mathscr{D}} = 8$	71.88	65.38	73.68	54.79	35.31	8.20	9.05M
$N_{\mathscr{D}} = 6, N_{\mathscr{R}} = 2$	**74.22**	**68.80**	**75.71**	58.61	40.06	9.80	10.25M

Table 4.4 Ablation on the refiner $\mathscr{R}$ for shape mAP and the number of parameters.

model size. Table 4.4 shows the quantitative results on the shape mAP and the number of learnable parameters of the Ours-BG model, with the number of decoder layers in the decoder $N_{\mathscr{D}} = 6$ and in the refiner $N_{\mathscr{R}} = 2$. We compare this model against the model without a refiner and having the same total number of decoder layers $N_{\mathscr{D}} = 8$, and another model with a doubled number of decoder layers $N_{\mathscr{D}} = 16$ without the refiner. The result shows that using the refiner achieves comparable performance in shape mAP with the model with a doubled number of decoder layers $N_{\mathscr{D}} = 16$ with a smaller model size.

Effect of the refiner $\mathscr{R}$ to the kinematic estimation The effect of the refiner $\mathscr{R}$ on kinematic estimation is shown in Figure 4.14. We conduct evaluations on joint parameter estimation. These evaluations focus on the intersection of true positive detections from both the model with and without the refiner with various mAP thresholds. Except for the joint axis orientation error at mAP@90%, the refiner improves the joint parameter estimation.

Effect of data distribution in terms of part counts To investigate the effect of the data distribution in terms of part counts, we trained our models on two train splits with controlled distribution of part counts. We generated such dataset with procedurally generated toy data from the *procedural dataset* as described in Section 4.4.1. The first train split contains fewer instances with part counts more than four to simulate a dataset having a fewer number of instances with many joints. We call this split *train baseline*. The other split has more instances with part counts bigger than four. We call this split *diverse train*. The distribution of the data in terms of the number of joints per instance as part counts is visualized in Figure 4.19 (a). The green bar shows the distribution of the test split. We show the quantitative evaluation result in terms of part detection accuracy in Figure 4.19 (b). Training on the *diverse train* split significantly improves the detection accuracy for instances with more parts compared to the *baseline train* split. This result suggests that our model is capable of understanding complex articulated object shapes with more part counts when the training dataset has a sufficient amount of such data.

	State ↓	OE ↓	MD ↓
OPD [48]	19.23°/17.10cm	29.68 °	38.11cm
Ours-BG	4.30°/5.27cm	3.97°	**6.43**cm
Ours	**3.85**°/**4.06**cm	**3.66**°	6.58cm

Table 4.5 Joint parameter estimation results.

4.4.7 Real-world data

We verify the generalizability of our approach, which is trained only on synthetic data to real-world data. Here, we include foreground masks as inputs to mitigate the domain gap from background. We present the quantitative results on the BMVC [78] dataset in Table 4.8. As only one instance is present in the scene, we evaluate with shape metrics without mAP. The CD value is multiplied by 100. Our method outperforms A-SDF [83]. We observe reasonable generalization as shown in Figure 4.10. We visualize the qualitative results from novel viewpoint in Figure 4.11.

4.4.8 Sequential joints

We verify the applicability of the grouping approach formulated in Equation 4.2 to sequential joints. For this experiment, we use the *sequential dataset* as described in 4.4.1. We show the qualitative result in Figure 4.16. Each cuboid in the figure indicates the detection of the part. Red and green colors on the top faces of each cuboid indicate the base parts and non-base parts. The association of parent-child association is visualized as yellow and blue colors. No cuboid with yellow color means the estimated parent is the scene root. As shown in the figure, the proposed grouping successfully identifies the correct parent. Especially, even when the candidate parent parts are at the same distance with reference to the base part as shown in (b), the correct parent part is estimated. In (c), since the selected part is the base part, it correctly estimates its parent as scene root. We also quantitatively evaluate the accuracy of the parent identification by assigning prediction and ground truth by bipartite matching as described in Section 4.3.7 to exclude the effect of false positives and negatives to the accuracy. We found the proposed approach accurately identifies the parent-child association with an accuracy of 99.63%. This result suggests the potential application of the proposed part grouping strategy to sequential part structures.

	mAP@90 ↑	Precision ↑	F-Score@90%↑
w/o QO, PF, kIoU	38.87	**52.29**	67.43
w/o PF, kIoU	36.48	32.65	65.99
w/o kIoU	40.78	49.64	66.24
All (Ours-BG)	**41.09**	51.64	**68.8**

Table 4.6 Ablation on KPF module.

	Fscore@80%	Fscore@90%	CD1@5%	CD1@1%	IoU@25%	IoU@50%
w/o AS	59.37	38.83	69.84	24.10	25.72	5.76
w/o SF	71.04	60.49	73.83	43.14	25.17	0.64
All (Ours-BG)	**74.22**	**68.80**	**75.71**	**58.61**	**40.06**	**9.80**

Table 4.7 Ablation on shape learning.

4.5 Limitation

In this section, we discuss the failure cases and limitations of this approach.

4.5.1 Physical constraint violation

The proposed approach currently does not consider the physical constraints between parts during training. Although the KPF module removes redundant parts as post-processing, physically implausible false positives still occur, as shown in Figure 4.20 (a), for parts without overlapping trajectories. Introducing regularization, such as physical violation loss [141], on such implausible configurations would alleviate this problem.

4.5.2 Data imbalance

The kinematic models that the supervised model can handle depend greatly on the distribution of the training data; it struggles with part structures and counts that are only included in small numbers.

Objects with many parts Our methods tend to perform worse for objects with more parts. For instance, a single object with many parts is separately reconstructed as two instances, as visualized in Figure 4.18. As shown in Figure 4.18, the dataset contains a very small number of training data for such an object with many parts; the method tends to fail on such objects.

	F-Score ↑	CD ↓	IoU ↑
A-SDF-GT-2 [83]	83.25	1.73	12.26
Ours	**97.51**	**0.56**	**27.96**

Table 4.8 Shape reconstruction results on the BMVC [78] dataset.

Ambiguous opening direction Our methods do not explicitly consider the ambiguity in the opening direction of parts. As shown in Figure 4.20 (b), the model fails to estimate the correct axis direction w.r.t. the base part for the closed part of the oven. This is because the dataset includes parts with different opening directions but similar shapes when closed. In this case, the data ground truth joint direction is horizontal w.r.t the base part of the oven; however, such data consists of only 18.75% of training data, and the rest are in the vertical direction, which is the same as the falsely predicted joint direction. Such ambiguity can be addressed by explicit uncertainty modeling, as in [1, 29], and integrating finer 2D visual cues for 3D reasoning [125, 62, 17] for localizing knobs and handles, which are informative for the opening direction.

4.6 Conclusion

We presented an end-to-end trainable part-level shape reconstruction method for multiple articulated objects from a single RGBD image. We have demonstrated that our method successfully tackles the major limitation of previous works, which are unable to handle objects with various part counts using a single model, by employing a novel detect-then-group approach. We have also shown that the proposed kinematic part fusion (KPF) module effectively handles small parts as challenging targets while suppressing false positives for detection-based reconstruction. Our method outperformed the state-of-the-art baselines in shape reconstruction and kinematics estimation on the SAPIEN [133] dataset. Furthermore, on the BMVC [78] dataset and casually captured images, we demonstrated that the model trained on synthetic data reasonably generalizes to real-world data.

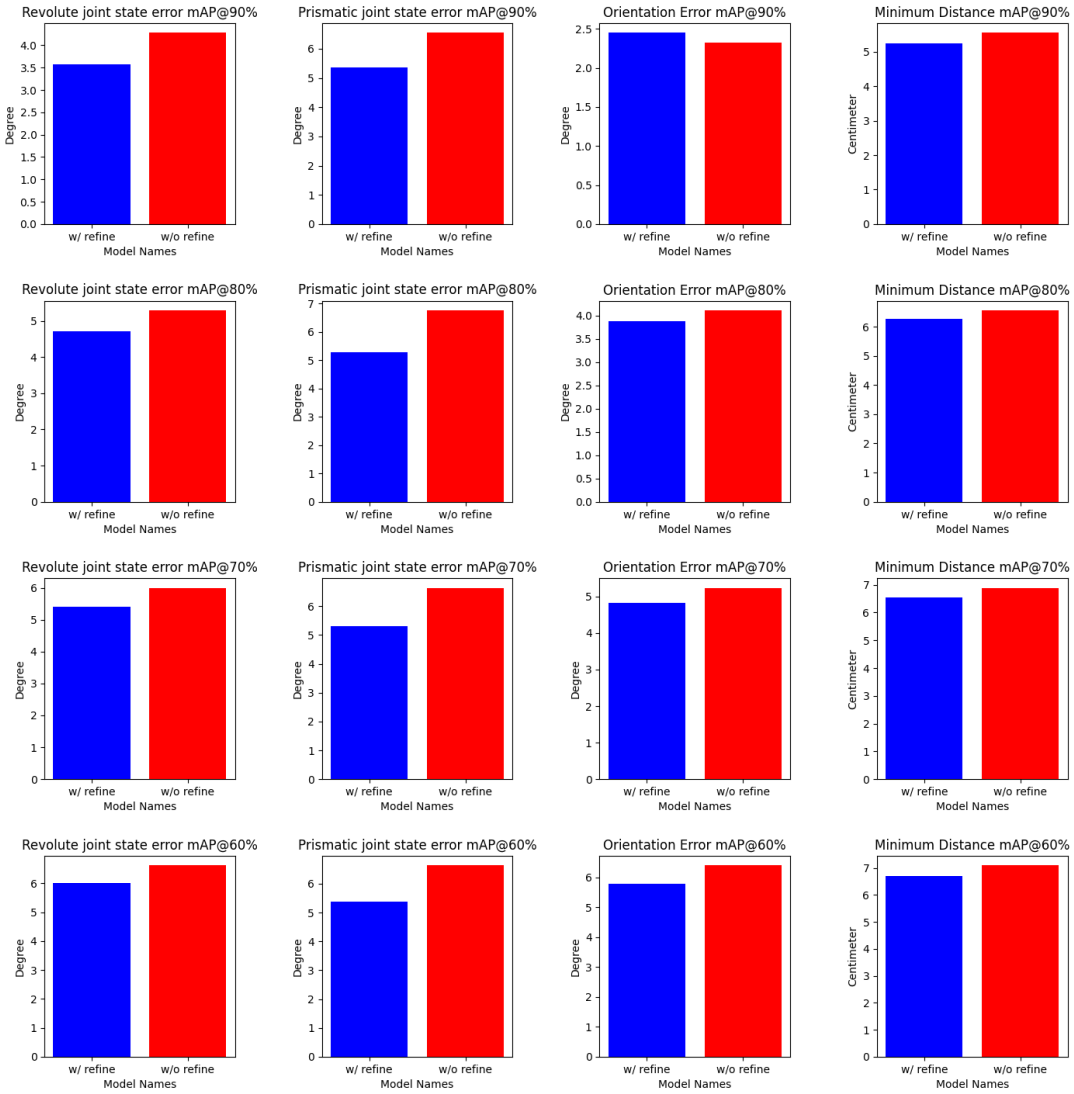
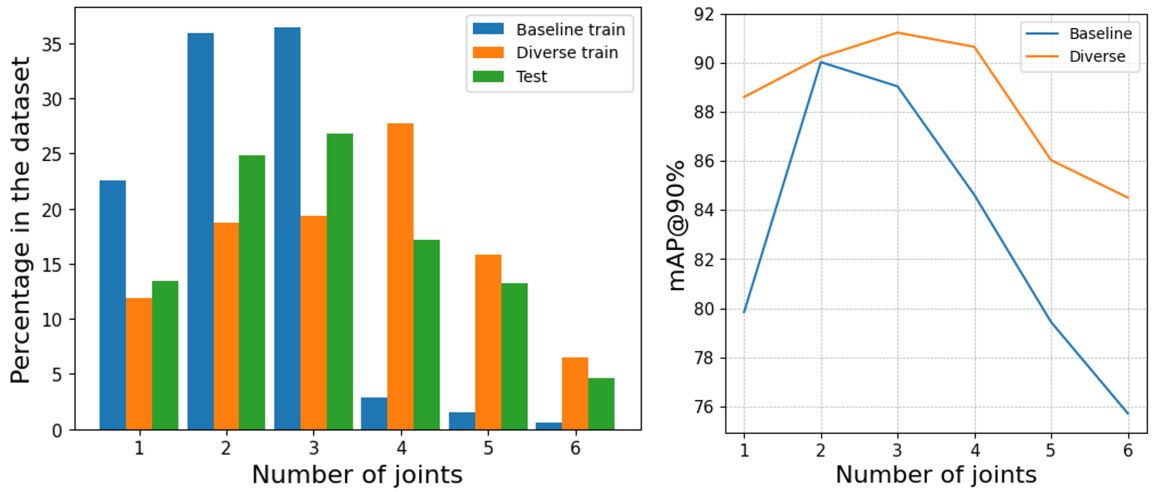
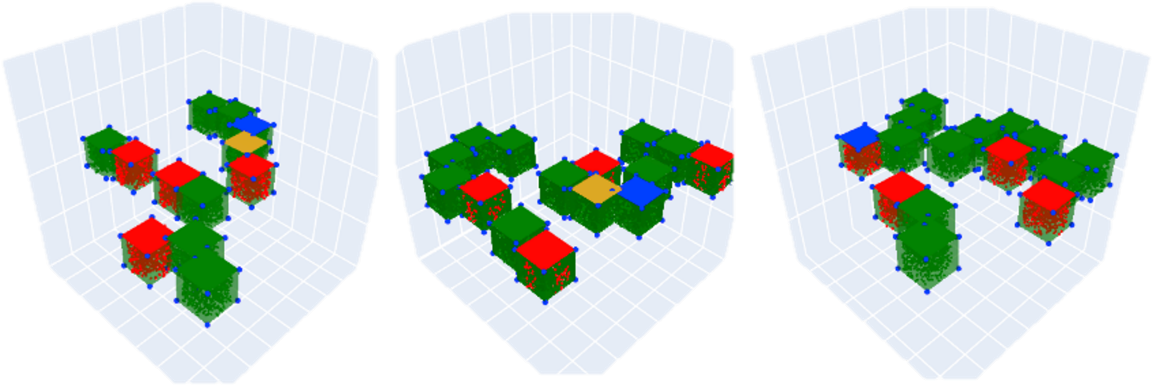


Figure 4.14 Effect of the refiner to joint parameter estimation.



(a) (b)

Figure 4.15 (a) Distribution of data in terms of the number of joints as part counts for the two train splits and the test split. (b) Average part detection accuracy for each part count per instance in terms of the number of joints.



(a) (b) (c)

Figure 4.16 Visualization of the sequential part association. Cuboids indicate the detection of the part, and red and green colors on the top faces of the cuboids show the base parts and non-base parts. The yellow color shows the estimated parent part of the part with blue color. In (c), since the estimated parent of the part is the scene root, no part is colored with yellow.

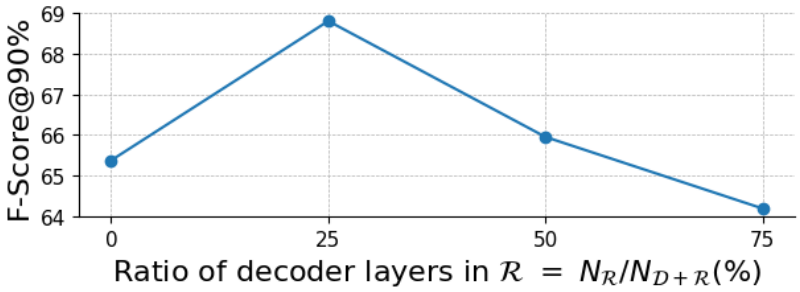


Figure 4.17 Ablation on refiner $\mathscr{R}$.

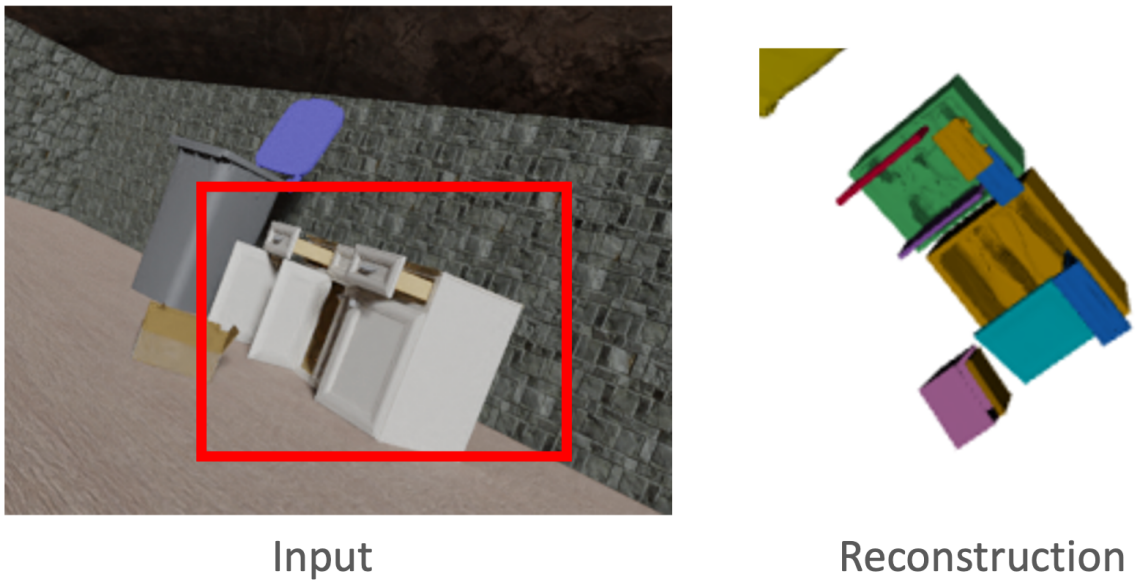


Figure 4.18 Failure case on an object with many parts.

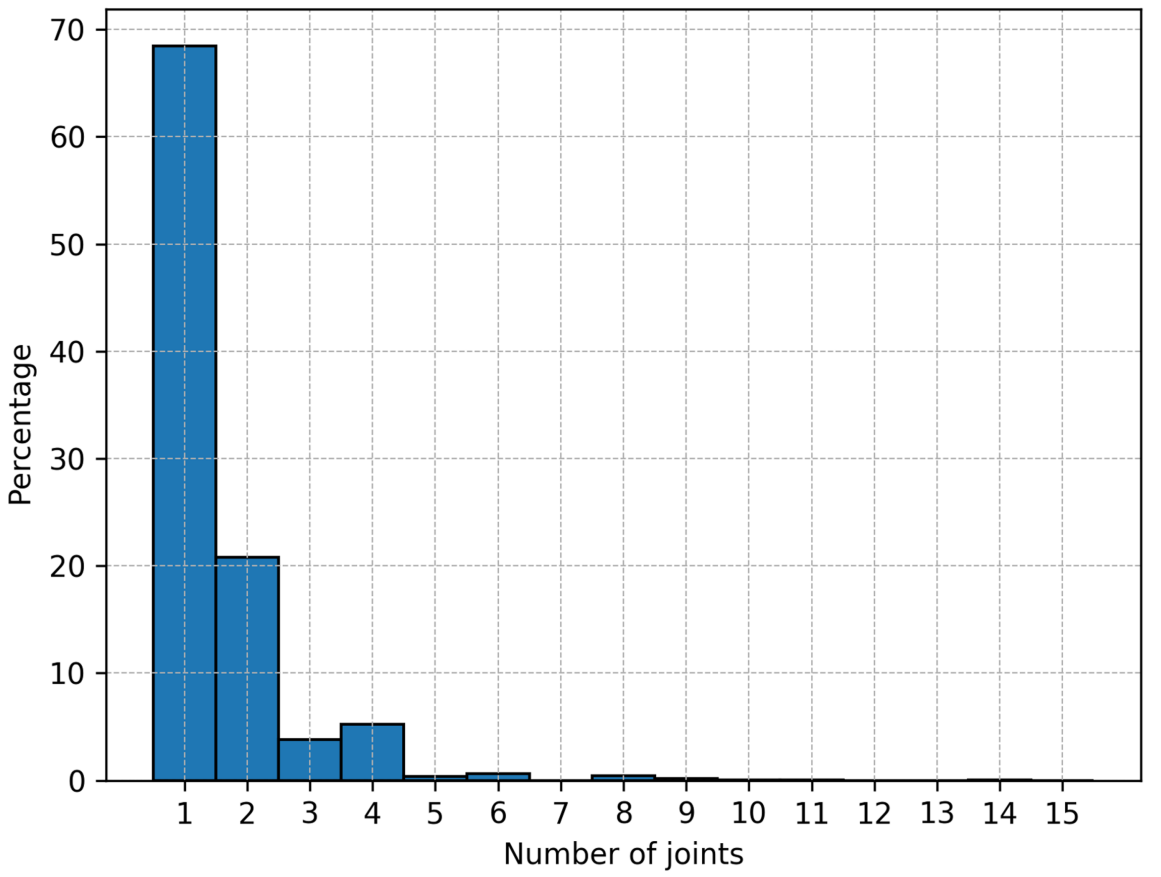


Figure 4.19 Distribution of objects in the dataset by the number of parts.

Figure 4.20 Visualization of failure cases.

Chapter 5

Exploiting consistent part structure for unsuperiviesd learning

5.1 Introduction

Our daily life environments are populated with man-made articulated objects, ranging from furniture and household appliances such as drawers and ovens to tabletop objects such as eyeglasses and laptops. Humans are capable of recognizing such objects by decomposing them into simpler semantic parts based on part kinematics. Researchers have shown that even very young infants learn to group objects into semantic parts using the location, shape, and kinematics as a cue [118, 116, 134], even from a single image [112, 59]. Although humans can naturally achieve such reasoning, it is challenging for machines, particularly in the absence of rich supervision.

3D part-level understanding of shapes and poses from a single frame observation has wide range of applications in computer vision and robotics. Learning to represent complex target shapes with simpler part components as a generative approach enables applications such as structure modeling [82, 106] and unsupervised 3D part parsing [124, 94, 19, 96]. The previous unsupervised approaches have mainly focused on non-articulated objects. Because they exploit the consistent part location as a cue to group shapes into semantic parts, these approaches are unsuitable for decomposing articulated objects when considering the kinematics of *dynamic part locations*. For part pose, modeling kinematic structures as joint parameters has various applications, such as motion planning in robotic manipulation [1] and interaction with environment in augmented reality [10]. There exists a large body of works for discriminative approaches dedicated to man-made articulated objects for part pose estimation in addition to part segmentation. However, they require explicit supervision, such

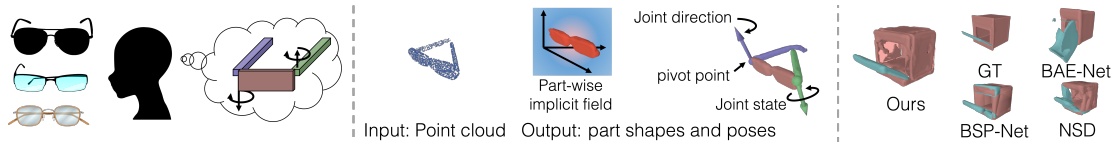


Figure 5.1 (Left) Even through independent observations, infants can build a mental model of the articulated object for part parsing based on its kinematics. (Middle) Likewise, we propose an unsupervised generative method that learns to parse the single-frame, unstructured 3D data of articulated objects and predict the part-wise implicit fields as abstracted part shapes as well as their part poses as joint parameters. (Right) Our approach outperforms the previous works in consistent part parsing for man-made articulated objects.

as segmentation labels and joint parameters [29, 1, 140, 133, 63]. Removing the need for such expensive supervision has been an important step toward more human-like representation learning [4].

In this study, as a novel problem setting, we investigate the unsupervised part decomposition task for man-made articulated objects with mechanical joints, considering part poses as *joint parameters*, in an *unsupervised fashion*. Specifically, we consider the revolute and prismatic parts with one degree-of-freedom joint state because they cover most of the kinematic types that common man-made articulated objects have [133, 1, 78]. This task aims to learn consistent part parsing as a generative shape abstraction approach for man-made articulated objects with various part poses from single-frame shape observation. An overview is shown in Figure 5.1. Recent part decomposition studies have focused on novel part shape representations for *shape reconstruction*. In contrast, we focus on *part parsing* and *part pose modeling* as a first step to expand the current generative part decomposition's applications to man-made articulated objects in novel ways, such as part pose consistent segmentation and part pose estimation as joint parameter prediction. To realize the task, we identify the two challenges; (1) for pose-aware part decomposition, the model must consider the kinematics between possibly distant shapes to group them as a single part and (2) has to disentangle the part poses from shape supervision. A comparison with previous studies is presented in Table 5.1.

To address these challenges, we propose PPD (unsupervised Pose-aware Part Decomposition) that takes an unsegmented, single-frame point cloud with various underlying part poses as an input. PPD predicts abstracted part-wise shapes transformed using the estimated joint parameters as the part poses. We train PPD as an autoencoder using single-frame shape supervision. PPD employs category-common decoders to capture category-specific rest-posed part shapes and joint parameters. Learning to transform the rest-posed shapes properly disentangles shape and pose, and (2) constraining the position of the parts by the joint parameters forces shapes in distant space that share the same kinematics to be

	Part segmentation	Joint parameter estimation	Generative	Unsupervised
ANSCH [63]	✓	✓		
A-SDF [83]	✓		✓	
Nueral Parts [96]	✓		✓	✓
Ours	✓	✓	✓	✓

Table 5.1 Overview of the previous works. We regard a method as unsupervised if the checked tasks can be learned only via shape supervision during training.

recovered as the same part. We also propose a series of losses to regularize the learning process. Furthermore, we employ non-primitive-based part shape representation and utilize deformation by part poses to induce part decomposition, in contrast to previous works that employ primitive shapes and rely on its limited expressive power as an inductive bias.

Our contributions are summarized as follows: (1) We propose a novel unsupervised generative part decomposition method for man-made articulated objects based on part kinematics. (2) We show that the proposed method learns disentangled part shape and pose: a non-primitive-based implicit field as part shape representation and the joint parameters as the part poses, using single-frame shape supervision. (3) We also demonstrate that the proposed method outperforms previous generative part decomposition methods in terms of semantic capability (parsimonious shape representation, consitent part parsing and interpretability of recovered parts) and show comparable part pose estimation performance to the supervised baseline.

5.2 Related works

5.2.1 Unsupervised part decomposition.

Existing unsupervised generative part decomposition studies mostly assume non-articulated objects in which the part shapes are in a fixed 3D location [124, 95, 18, 19, 26, 52], or also targeting human body and hand shapes without considering part pose [96]. They induce part decomposition by limiting the expressive power of the shape decoders by employing learnable primitive shapes. Closest work of ours is BAE-Net [18], whose main focus is consistent part parsing by generative shape abstraction. It also employs a non-primitive-based implicit field as the part shape representation, similar to ours. However, it still limits the expressive power of the shape decoder using MLP with only three layers. In contrast, our approach assumes parts to be dynamic with the consistent kinematics and induces part decomposition through

rigid transformation of the reconstructed part shapes with the estimated part poses to make the decomposition pose-aware.

5.2.2 Articulated shape representation.

A growing number of studies have tackled the reconstruction of category-specific, articulated objects with a particular kinematic structure, such as the human body and animals. Representative works rely on the use of category-specific template models as the shape and pose prior [69, 142, 9, 143, 60]. Another body of works reconstruct target shapes without templates, such as by reconstructing a part-wise implicit field given a part pose as an input [27] or focusing on non-rigid tracking of the seen samples [11]. The recent work [83] targets man-made articulated objects and supervised part shape reconstruction. In contrast, our approach focuses on man-made articulated objects with various kinematic structures. Our approach learns the part shapes and poses during training, without any part label and pose information either as supervision or input, and is applicable to unseen samples.

5.2.3 Part pose estimation.

In discriminative approaches, a number of studies have focused on the inference of part poses as joint parameters [63, 133, 1] targeting man-made articulated objects. These approaches require expensive annotations, such as part labels and ground-truth joint parameters. Moreover, they require category-specific prior knowledge of the kinematic structure. In contrast, our model is based on generative approach and is category agnostic. Moreover, it only requires shape supervision during training. A recent work [43] assumes an unsupervised setting where multi-frame, complete shape point clouds are available for both input and supervision signals during training and inference. Whereas our approach assumes a single-frame input and shape supervision, it also works with partial shape input during inference. Note that, in this study, the purpose of part pose estimation is, as an auxiliary task, to facilitate consistent part parsing. It is not our focus to outperform the state-of-the-art supervised approaches in part pose estimation.

5.3 Methods

In our approach, the goal is to represent an articulated object as a set of semantically consistent part shapes based on their underlying part kinematics. We represent the target object shape as an implicit field that can be evaluated at an arbitrary point $\mathbf{x} \in \mathbb{R}^3$ in 3D space as $O : \mathbb{R}^3 \rightarrow [0,1]$, where $\{\mathbf{x} \in \mathbb{R}^3 \mid O(\mathbf{x}) = 0\}$ defines the outside of the object,

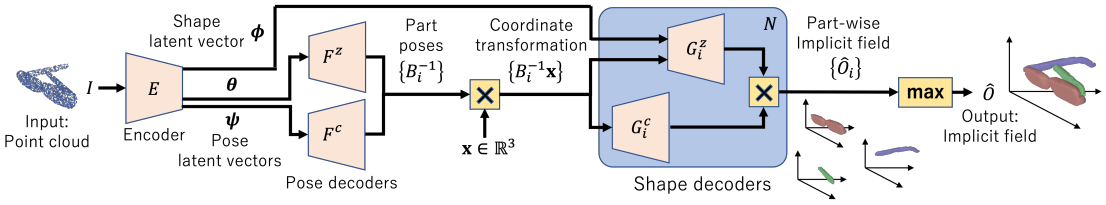


Figure 5.2 Model overview. PPD infers implicit field $\hat{O}$ based on part poses $\{B_i\}$ and part-wise implicit fields $\{\hat{O}_i\}$ given input point cloud I. The category-common decoders F^c and $\{G_i^c\}$ capture part pose biases and part shape priors in constant latent vectors. Instance-dependent decoders F^z and $\{G_i^z\}$ model input specific components. Constraining the instance-dependent decoders by the category-common biases and the priors in the proposed approach realizes unsupervised part decomposition and joint parameter learning. Note we shorthand $\{*_i\}$ to denote an ordered set $\{*_i\}_{i=1}^N$ for brevity.

$\{\mathbf{x} \in \mathbb{R}^3 \mid O(\mathbf{x}) = 1\}$ the inside, and $\{\mathbf{x} \in \mathbb{R}^3 \mid O(\mathbf{x}) = 0.5\}$ the surface. Given a 3D point cloud $I \in \mathbb{R}^{P\times 3}$ of P points as an input, we approximate the object shape using a composite implicit field $\hat{O}$ that is decomposed into a collection of N parts. The i-th part has an implicit field $\hat{O}_i(\mathbf{x} \mid I)$ as part shape and part pose $B_i \in \mathrm{SE}(3)$. We ensure that O is approximated as $O(\mathbf{x}) \approx \hat{O}(\mathbf{x} \mid I, \{B_i\}_{i=1}^N)$ through the losses.

An overview of PPD is shown in Figure 5.2. PPD employs an autoencoder architecture, and is trained under single category setting. Given a point cloud I, the encoder derives the disentangled shape latent vector $\boldsymbol{\phi} \in \mathbb{R}^m$ and the two pose latent vectors $\boldsymbol{\theta} \in \mathbb{R}^n$ and $\boldsymbol{\psi} \in \mathbb{R}^o$. Category-common pose decoder F^c captures joint parameter biases given $\boldsymbol{\psi}$. Instance-dependent pose decoder F^z models residual joint parameters to the biases given $\boldsymbol{\theta}$. The part-wise category-common shape decoder G_i^c captures category-common shape prior. Given $\boldsymbol{\phi}$ and conditioned by G_i^c, instance-dependent shape decoder G_i^z infers residual shape details of the target shape to decode a part-wise implicit field $\hat{O}_i$. We discuss the details about F^z and F^c in Section 5.3.1, and G_i^z and G_i^c in Section 5.3.2.

5.3.1 Part pose representation

We characterize part pose B_i by its part kinematic type and joint parameters. Each part kinematic type $y_i \in \{\text{fixed}, \text{prismatic}, \text{revolute}\}$ is manually set as a hyperparameter. The joint parameters consist of the joint direction $\mathbf{u}_i \in \mathbb{R}^3$ with the unit norm and joint state $s_i \in \mathbb{R}^+$. Additionally, the "revolute" part has the pivot point $\mathbf{q}_i \in \mathbb{R}^3$. We refer to the joint direction and pivot point as the joint configuration. For the "fixed" part, we set B_i as an identity matrix because no transformation is applied. For the "prismatic" part, we define $B_i = T(s_i\mathbf{u}_i)$, where $T(\cdot)$ represents a homogeneous translation matrix given the translation

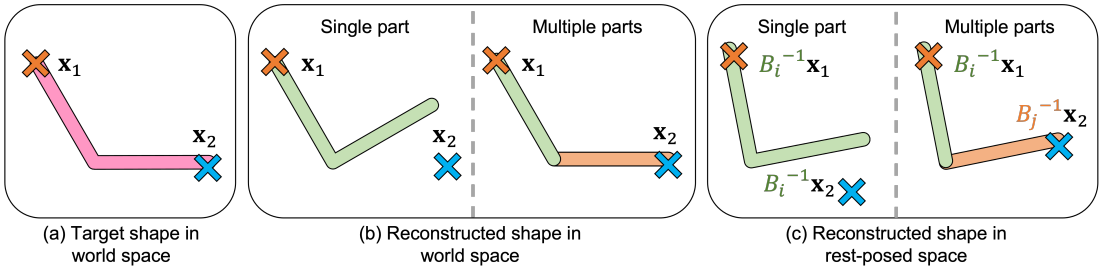


Figure 5.3 Illustration of part decomposition induction. "Single part" indicates that the model is degenerated to use only a single part to reconstruct the whole target shape. "Multiple parts" indicates that part decomposition is correctly induced. In (b), the "single part" model misclassifies query point $\mathbf{x}_2$ as outside, in contrast to $\mathbf{x}_1$. As shown in (c), a single part pose $\{B_i\}$ cannot correctly transform both query points inside the rest-posed shape. The "multiple parts" model successfully classifies both query points using different part poses per part. Minimizing the reconstruction loss incentivizes the model to use multiple parts and appropriate part types for $\{B_i\}$.

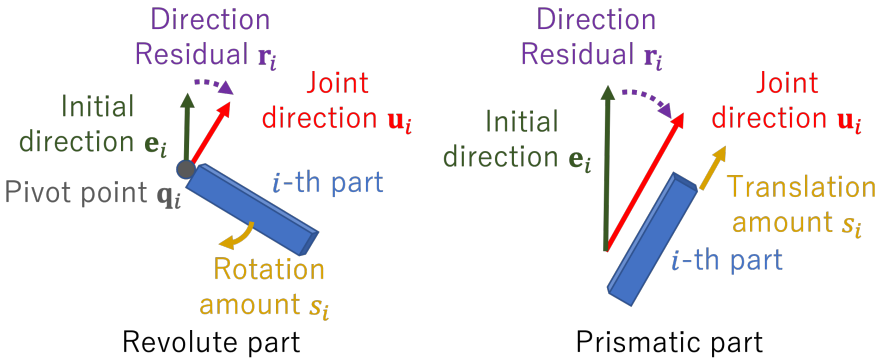


Figure 5.4 Geometric relationship between the joint parameters.

in $\mathbb{R}^3$, and s_i and $\mathbf{u}_i$ represent the translation amount and direction, respectively. For the "revolute" part, we set $B_i = T(\mathbf{q}_i)R(s_i, \mathbf{u}_i)$, where $R(\cdot)$ denotes a homogeneous rotation matrix given the rotation representation, and s_i and $\mathbf{u}_i$ represent the axis-angle rotation around the axis $\mathbf{u}_i$ by angle s_i. In human shape reconstruction methods using template shape, its pose is initialized to be close to the real distribution to avoid the local minima [50, 60]. Inspired by these approaches, we parametrize the joint direction as $[\mathbf{u}_i; 1] = R(\mathbf{r}_i)[\mathbf{e}_i; 1]$, where $\mathbf{e}_i$ is a constant directional vector with the unit norm working as the initial joint direction as a hyperparameter and $\mathbf{r}_i \in \mathbb{R}^3$ represents the Euler-angle representation working as a residual from the initial joint direction $\mathbf{e}_i$. This allows us to manually initialize the joint direction in a realistic distribution through $\mathbf{e}_i$ by initializing $\mathbf{r}_i = \mathbf{0}$. Figure 5.4 illustrates the joint parameters.

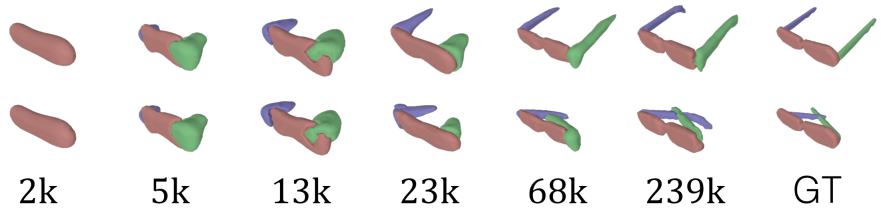


Figure 5.5 Visualization of the training process. The numbers in the figure show the training steps.

Based on our observations, we assume that the joint configuration has a category-common bias, while the joint state strongly depends on each instance. This is because the location of each part and the entire shape of an object can constrain the possible trajectory of the parts, which is defined by the joint configuration. To illustrate this idea, we propose to decompose the joint configuration into a category-common bias term and an instance-dependent residual term denoted as $\mathbf{r}_i = \mathbf{r}_i^c + \mathbf{r}_i^z$ and $\mathbf{q}_i = \mathbf{q}_i^c + \mathbf{q}_i^z$, respectively. We employ the category-common pose decoder $F^c(\mathrm{qt}(\boldsymbol{\psi}))$, which outputs $\{\mathbf{r}_i^c \mid i \in \mathbb{A}^p\}$ and $\{\mathbf{q}_i^c \mid i \in \mathbb{A}^r\}$, where $\mathbb{A}^p = \{i \in [N] \mid y_i \neq \text{fixed}\}$, $\mathbb{A}^r = \{i \in [N] \mid y_i = \text{revolute}\}$, $\boldsymbol{\psi}$ denotes a pose latent vector, and $\mathrm{qt}(\cdot)$ is a latent vector quantization operator following VQ-VAE [103]. Directly regressing the pose parameter value is known to be difficult. Prior work [128] classifies the value into the discretized value ranges and regresses the residual within the classified range. Motivated by this, vector quantization for F^c models discrete modality of the joint parameter, and F^z regresses the residual. The operator $\mathrm{qt}(\cdot)$ outputs the nearest constant vector to the input latent vector $\boldsymbol{\psi}$ among the N_{qt} candidates. Instead of using a single constant vector, the model selects a constant vector among multiple constant vectors to capture the discrete, multi-modal category-common biases. We also employ an instance-dependent pose decoder $F^z(\boldsymbol{\theta})$ that outputs $\{s_i \mid i \in \mathbb{A}^p\}$, $\{\mathbf{r}_i^z \mid i \in \mathbb{A}^p\}$, and $\{\mathbf{q}_i^z \mid i \in \mathbb{A}^r\}$. We constrain the possible distribution of the joint configuration around the category-common bias by the loss function explained in Section 5.3.3. This constraint incentivizes the model to reconstruct the instance-dependent shape variation by the joint state, which constrains the part location along the joint direction. This kinematic constraint biases the model to represent the shapes having the same kinematics with the same part. The previous works [52, 26, 94] do not impose such a constraint on the part localization, thus learned part decomposition is not necessarily consistent under different poses.

5.3.2 Part shape representation

We propose a non-primitive-based part shape representation that is decomposed into the category-common shape prior and instance-dependent shape details. We employ MLP-based decoders to model a part-wise implicit field. We capture the category-common shape prior using the category-common shape decoder $G_i^c(\mathbf{x})$. Because G_i^c does not take a latent vector from the encoder, it learns an input-independent, rest-posed part shape template as the category-common shape prior. We also employ an instance-dependent shape decoder $G_i^z(\mathbf{x} \mid \boldsymbol{\phi})$ to capture the additional instance-dependent shape details conditioned with the shape prior. We formulate a part-wise implicit field $\hat{O}_i$ as follows:

$$\hat{O}_i(\mathbf{x} \mid I) = \sigma(G_i^z(\mathbf{x}, \boldsymbol{\phi})\hat{O}_i^c(\mathbf{x})) \tag{5.1}$$

where $\sigma(\cdot)$ represents the sigmoid function and $\hat{O}_i^c(\mathbf{x}) = \sigma(G_i^c(\mathbf{x}))$. For brevity, we omit I in $\hat{O}_i$ and simply denote it as $\hat{O}_i(\mathbf{x})$. Given the part poses $\{B_i\}$ as part-wise locally rigid deformation, we formulate $\hat{O}$ as the composition of $\{\hat{O}_i\}$ defined as $\hat{O}(\mathbf{x} \mid I, \{B_i\}) = \max_i\{\hat{O}_i(B_i^{-1}\mathbf{x})\}$. As in the piecewise rigid model of [27], coordinate transformation $B_i^{-1}\mathbf{x}$ realizes locally rigid deformation by B_i of the part-wise implicit field by querying the rest-posed indicator. Note that, although we set the maximum number of parts N, the actual number of parts used for reconstruction can change; it is possible that some parts do not contribute to the reconstruction because of the *max* operation or simply because $\hat{O}_i < 0.5$ for all 3D locations.

In Equation 5.1, we experimentally found that conditioning G_i^z by $\hat{O}_i^c$ through multiplication rather than addition effectively prevents G_i^z from deviating largely from G_i^c. Since a shape can exist at positions only where both G_i^z and G_i^c are large through multiplication, for each part, G_i^c defines a category-common shape, and G_i^z provides a shape that reflects input-dependent details around the category-common shape, visualized in Figure 5.6. This conditioning induces the unsuperivsed part decomposition. We illustrate the idea in Figure 5.3. Considering reconstructing the target shape by single i-th part, since the multiplication makes it difficult to output shapes that deviating largely from the category-common prior shape, the large shape variations of target shapes are expressed by B_i regarded as the global pose of the reconstructed shape. However, the large shape variations in target shapes are due to the various local poses of multiple part shapes. Therefore, the large shape variations of target shapes cannot be expressed only by the single part and its part pose B_i. Thus, as an inductive bias of the unsupervised part decomposition, the model is incentivized to use a composition of multiple parts to express the shape variations due to various local part poses. We visualize the learning process in Figure 5.5. First, the model learns high indicator

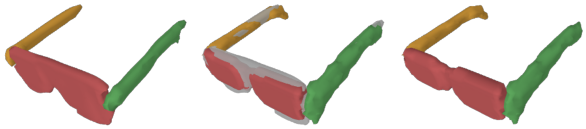


Figure 5.6 Relationship between G_i^c, G_i^z and $\hat{O}$. (a) Category-common shape $\hat{O}^c = \max_i\{\sigma(G_i^c(\cdot))\}$. (b) $\max_i\{\sigma(G_i^z(\cdot))\}$ overlaid with $\hat{O}^c$. (c) Predicted shape $\hat{O}$.

values in spatial locations of static parts with high probabilities of space occupancy in any instance. Next, part decomposition is induced to accommodate various target shapes' part poses, generating multiple dynamic parts. Indicator values in the spatial locations with less displacement by different part poses (e.g., near pivot points of revolute parts) first exceed the iso-surface threshold. Then, the model simultaneously optimizes part pose estimation and shape reconstruction during training as an analysis-by-synthesis approach.

5.3.3 Training losses

5.3.4 Shape losses.

To learn the shape decoders, we minimize the reconstruction loss using the standard binary cross-entropy loss (BCE) defined as:

$$\mathscr{L}_{\text{reconstruction}} = \lambda_{\text{reconstruction}}\text{BCE}(\hat{O}, O) + \lambda^c_{\text{reconstruction}}\text{BCE}(\hat{O}^c, O) \tag{5.2}$$

where $\hat{O}^c(\mathbf{x} \mid B) = \max_i\{\hat{O}_i^c(B_i^{-1}\mathbf{x})\}$, and $\lambda_{\text{reconstruction}}$ and $\lambda^c_{\text{reconstruction}}$ are the loss weights. The second term in Equation 5.2 is essential for stable training; it facilitates fast learning of $\{G_i^c\}$, so that $\{G_i^z\}$ can be correctly conditioned in the early stage of the training process. Moreover, because we consider the locally rigid deformation of the shape, the volumes of the shape before and after the deformation should not be changed by the intersection of parts; we formulate this constraint as follows:

$$\begin{aligned}\mathscr{L}_{\text{volume}} = \lambda_{\text{volume}}\Bigg(&\mathbb{E}_{\mathbf{x}}\Big[\text{ReLU}(\max_i\{G_i^z(B_i^{-1}\mathbf{x}, \boldsymbol{\phi})\})\Big] \\ &- \mathbb{E}_{\mathbf{x}}\Big[\text{ReLU}(\max_i\{G_i^z(\mathbf{x}, \boldsymbol{\phi})\})\Big]\Bigg)^2\end{aligned} \tag{5.3}$$

5.3.5 Joint parameter losses.

For the joint parameters $\mathbf{q}_i$ and $\mathbf{r}_i$, we prevent an instance-dependent term from deviating too much from the bias term, we regularize them by the loss:

$$\mathscr{L}_{\text{deviation}} = \lambda_{\text{deviation}} \left(\frac{1}{N^r} \sum_{i \in \mathbb{A}^r} \|\mathbf{q}_i^z\| + \frac{1}{N^p} \sum_{i \in \mathbb{A}^p} \|\mathbf{r}_i^z\| \right) \tag{5.4}$$

where $N^r = |\mathbb{A}^r|$, $N^p = |\mathbb{A}^p|$, and $\lambda_{\text{deviation}}$ is the loss weight. Moreover, we propose a novel regularization loss that constrains the pivot point with the implicit fields. We assume that the line in 3D space, which consists of the pivot point and joint direction, passes through the reconstructed shape. The joint should connect at least two parts, which means that the joint direction anchored by the pivot point passes through at least two reconstructed parts. We realize this condition as follows:

$$\mathscr{L}_{\text{pivot}} = \frac{\lambda_{\text{pivot}}}{N^r} \sum_{i \in \mathbb{A}^r} \left(\min_{\mathbf{x} \in \mathbb{S}_{\text{GT}}} \|\mathbf{q}_i - \mathbf{x}\| + \frac{1}{2} \left(\min_{\mathbf{x} \in \mathbb{S}_i} \|\mathbf{q}_i - \mathbf{x}\| + \min_{\mathbf{x} \in \mathbb{S}_{i,j}} \|\mathbf{q}_i - \mathbf{x}\| \right) \right) \tag{5.5}$$

where $\mathbb{S}_{\text{GT}} = \{\mathbf{x} \in \mathbb{R}^3 \mid O(\mathbf{x}) = 1\}$, $\mathbb{S}_i = \{\mathbf{x} \in \mathbb{R}^3 \mid \hat{O}_i(B_i^{-1}\mathbf{x}) > 0.5\}$, $\mathbb{S}_{i,j} = \{\mathbf{x} \in \mathbb{R}^3 \mid \hat{O}_j(B_j^{-1}\mathbf{x}) > 0.5,\ j \in \mathbb{A}^r \setminus i\}$, and λ_{pivot} is the loss weight. Note that $\mathscr{L}_{\text{pivot}}$ is self-regularizing and not supervised by the ground-truth part segmentation. We illustrate the formulation in Figure 5.7. To reflect the diverse part poses, we prevent the joint state s_i from degenerating into a static state. In addition, to prevent multiple decomposed parts from representing the same revolute part, we encourage the pivot points to be spatially spread. We realize these requirements by the loss defined as:

$$\mathscr{L}_{\text{variation}} = \frac{1}{N^p} \sum_{i \in \mathbb{A}^p} \left(\frac{\lambda_{\text{variation}^{\text{s}}}}{\text{std}(s_i)} + \lambda_{\text{variation}^{\text{q}}} \sum_{j \in \mathbb{A}^r \setminus i} \exp\left(- \frac{\|\mathbf{q}_i - \mathbf{q}_j\|}{v} \right) \right) \tag{5.6}$$

where $\text{std}(\cdot)$ denotes the batch statistics of the standard deviation, v is a constant that controls the distance between pivot points, and $\lambda_{\text{variation}^{\text{s}}}$ and $\lambda_{\text{variation}^{\text{q}}}$ are the loss weights. Lastly, following the loss proposed in [103], the pose latent vector $\boldsymbol{\psi}$ is optimized by the loss:

$$\mathscr{L}_{\text{quantization}} = \|\boldsymbol{\psi} - \text{sg}(\text{qt}(\boldsymbol{\psi}))\| \tag{5.7}$$

where sg denotes an operator stopping gradient on the backpropagation.

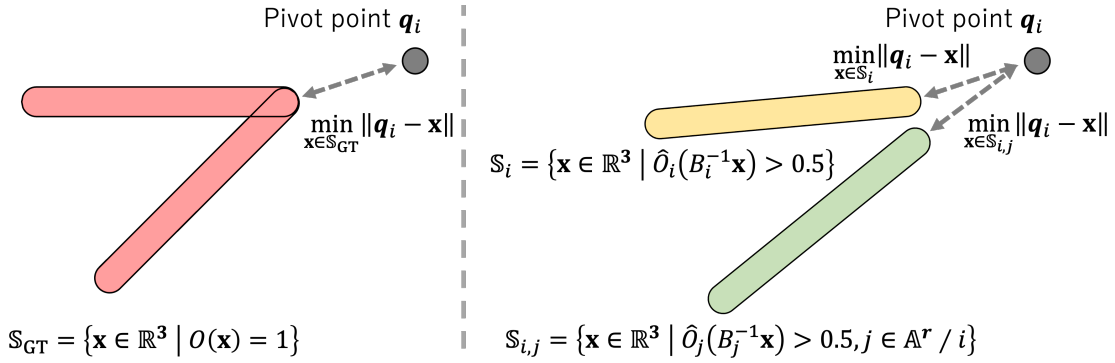


Figure 5.7 Illustration of $\mathscr{L}_{\text{pivot}}$ in 2D.

5.3.6 Adversarial losses.

Inspired by human shape reconstruction studies [16, 97], we employ the adversarial losses from WGAN-GP [36] to regularize the shape and pose in the realistic distribution. The losses are defined as:

$$\begin{aligned}\mathscr{L}_{\text{discriminator}} = \lambda_{\text{discriminator}} \Big(& \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r}[D(x)] \Big) \\ & + \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2]\end{aligned} \tag{5.8}$$

$$\mathscr{L}_{\text{generator}} = -\lambda_{\text{generator}} \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g}[D(\tilde{x})] \tag{5.9}$$

where $D(\cdot)$ is a discriminator; $\tilde{x}$ is a sample from the reconstructed shapes $\mathbb{P}_g$ transformed by the estimated joint configuration and randomly sampled joint state $\tilde{s}_i \sim \text{Uniform}(0, h_i)$, with the maximum motion amount h_i treated as a hyperparameter; x is a sample from the ground-truth shapes $\mathbb{P}_r$; $\hat{x}$ is a sample from $\mathbb{P}_{\hat{x}}$, which is a set of randomly and linearly interpolated samples between $\hat{x}$ and x; and $\lambda_{\text{generator}}$ and $\lambda_{\text{discriminator}}$ are the loss weights. As an input to D, we concatenate the implicit field and corresponding 3D points to create a 4D point cloud, following [57].

5.3.7 Implementation details

We use the Adam solvers [55] with a learning rate of 0.0001 with a batch size of 18 to optimize the sum of the losses: $\mathscr{L}_{\text{reconstruction}} + \mathscr{L}_{\text{volume}} + \mathscr{L}_{\text{quantization}} + \mathscr{L}_{\text{deviation}} + \mathscr{L}_{\text{pivot}} + \mathscr{L}_{\text{variation}} + \mathscr{L}_{\text{generator}}$ and the discriminator loss $\mathscr{L}_{\text{discriminator}}$, respectively. We set the loss weights as follows: $\lambda_{\text{reconstruction}} = 0.01$, $\lambda^c_{\text{reconstruction}} = 0.001$, $\lambda_{\text{deviation}} = 0.1$, $\lambda_{\text{pivot}} = 100$, $\lambda_{\text{variation}^s} = 0.1$, $\lambda_{\text{variation}^q} = 0.01$, $\lambda_{\text{volume}} = 1000$, $\lambda_{\text{generator}} = 0.65$, and $\lambda_{\text{discriminator}} = 0.35$. We set $v = 0.01$ in $\mathscr{L}_{\text{variation}}$ and $N_{qt} = 4$ for $\text{qt}(\cdot)$. For h_i in $\mathscr{L}_{\text{discriminator}}$, we set to $\frac{\pi}{2}$ and 0.4

the "revolute" and "prismatic" parts, respectively. Note that we experimentally found that it does not constrain the model to predict s_i larger than h_i to reconstruct the target shape.

Because we do not impose any geometric constraints on the part shapes, we set the number of parts for each part kinematics y_i as its maximum number in the datasets plus an additional one part for over-parameterization. The detail of the datasets is explained in Section 5.4.1.

We set $N = 8$, which consists of one "fixed" part, three "revolute" parts, and four "prismatic" parts. We use the same hyperparameter for all categories, without assuming the category-specific knowledge.

During the training, the max operation is substituted with LogSumExp for gradient propagation to each shape decoder.

We train our network in two stages following [19]: first, we train it on an implicit field of 16^3 grids and then on 32^3 grids. For the ground-truth implicit field, for each sample in a batch, we use 4096 3D coordinate points and their corresponding indicator values sampled from either 16^3 or 32^3 grids, depending on the training stage. This multi-stage training strategy on grids with different resolutions is inspired by [19]. We train our network on 16^3 grids in the first training stage. In addition, we set $\mathbf{r}_i = \mathbf{r}_i^c$ in the first stage. Then, we set $\mathbf{r}_i = \mathbf{r}_i^c + \mathbf{r}_i^s$ in the second stage. We determine the number of iterations for each stage according to the reconstruction loss and to the visualization of the reconstructed shapes on the validation data. For the input, we use the point cloud with 4096 points sampled from the surface of the target shape during the training. Unless otherwise noted, we use the complete shape point cloud.

It takes 2 to 3 days to train one model on a single NVIDIA V100 graphics card with 16 GB of GPU memory.

5.3.8 Model parameter initialization.

We use a sine function as a nonlinear activation function and the weight initialization strategy proposed in [115] in our shape decoders, as follows:

$$w \sim \mathrm{U}\left(-\sqrt{\frac{6}{\mathrm{IN}}}, \sqrt{\frac{6}{\mathrm{IN}}}\right) \frac{1}{30} \tag{5.10}$$

where IN is an input channel to a linear layer, U is a uniform distribution, and w is an element of the weight of a linear layer. For a linear layer that takes 3D coordinates as an input, we do not scale the weight w by $\frac{1}{30}$.

As described in the main paper, we set the number of parts used in our model as $N = 8$, which consists of one "fixed" part, three "revolute" parts, and four "prismatic" parts. Each initial joint direction $\mathbf{e}_i$ for the "revolute" and "prismatic" parts are set as follows, with z-axis as up, x-axis as forward, and y-axis as right: $+z$, $-z$ and $+y$ directions for "revolute" parts and $+x$ direction for "prismatic" parts. See Figure 5.8 for visual correspondence between shapes and axes.

5.3.9 Network architecture.

We use the PointNet [101]-based architecture from [77] as an encoder E and the one from [113] as a discriminator D. Our shape decoders $\{G_i^c\}$ and $\{G_i^z\}$ are MLP with sine activation [115] for a uniform activation magnitude suitable for propagating gradients to each shape decoder. For the category-common pose decoder F^c, we use separate networks of MLP for each kind of output variables. For the instance-dependent pose decoder F^z, we employ MLP with a single backbone having multiple output branches.

5.4 Experiments

5.4.1 Datasets.

Synthetic data In our evaluation, we follow the recent part pose estimation studies targeting man-made articulated objects for the synthetic datasets and the object categories covering various part kinematics: oven, eyeglasses, laptop, and washing machine categories from Motion dataset [128], and the drawer category from SAPIEN dataset [133].

Each category has a fixed number of parts with the same kinematic structure. We generate 100 instances with different poses per sample, generating 24k instances in total. We split our training and test data according to the per-category data split approach introduced in [63]. We ensure that the test split contains at least six samples per category, except for the laptop category; therefore, the average split ratio is approximately 8:2. For the laptop category, we use 11 samples in the test split to make the split ratio comparable with those of the other categories. The number of samples in each split per category is presented in Table 5.2.

Following [77], we generate the ground-truth implicit field by the volumetric fusion of 100 depth images of a mesh object. For the mesh object, we sample 100 instances with randomly sampled part poses for each sample. For the pose sampling, we uniformly sample the rotation amount for each joint for the revolute joints. For the revolute joints of all categories except the eyeglasses category, we sample the rotation amount between 0° and 135°. For the eyeglasses category, we sample between 0° and 90°. For the prismatic joints of

the drawer category, we sample the translation amount between 0 and the maximum amounts of the joints written in the URDF files of each sample in the SAPIEN dataset [133]. After we sample a part pose for each instance, we articulate the sample in its canonical pose (the rotation amount and translation amount were set to 0° and 0, respectively) using the sampled motion amount and ground-truth joint configuration. The canonically posed shape and the randomly posed shape of the same sample are shown in Figure 5.8. Finally, we normalize the size and location of the instances following [77]. Specifically, we normalize the instances with the maximum extent collected from the instances generated from the same sample.

Real data To verify the transferability of our approach trained on synthetic data to real data, we use the laptop category from RBO dataset [76] and Articulated Object Dataset [78], which is the intersecting category with the synthetic dataset.

5.4.2 Baselines

We compare our method with the state-of-the-art unsupervised generative part decomposition methods with various characteristics: BAE-Net [19] (non-primitive-based part shape representation), BSP-Net [19] (primitive-based part shape representation with part localization by 3D space partitioning), NSD [52] and Neural Parts [96] denoted as NP (primitive-based part shape representation with part localization in $\mathbb{R}^3$). For the part pose estimation, we use NPCS [63] as the supervised baseline. NPCS performs part-based registration by iterative rigid-body transformation, which is a common practice in articulated pose estimation of rigid objects.

We use the author-provided implementations for all the baselines. We explain the additional detail of the training for the baselines below.

BSP-Net [19] Because the models in the author-provided codes of the other part decomposition baselines (BAE-Net [18] and NSD [52]) are trained on 32^3 grids, we also trained BSP-Net on up to 32^3 grids, compared to the 64^3 grids in the original implementation. For training on the eyeglasses category, we could not successfully train the model even with different random seeds with the provided training script. After several trials, we experimentally found that scaling ground-truth indicator values by four for the first 20,000 iterations produced good initialization of the model. On the basis of this finding, we first pre-trained the model using the scaled ground-truth indicator values for 20,000 iterations for the eyeglasses category; then, we trained the model with the provided training script.

NSD [52] and Neural Parts [96] The model defined in the author-provided code takes an RGB image as an input, which is a more challenging setting for 3D shape reasoning than 3D shape input. We replace the image encoder of the original implementation with the same PointNet-based encoder used in our approach for a fair comparison.

NPCS [63] In the experiment described in Section 5.4.6 in the main paper, we modified the original implementation of NPCS to use complete shape point clouds instead of partial point clouds of the depth map as an input with training from scratch, to remove the unnecessary performance degradation caused by pose ambiguity arising from the barely visible articulated part.

5.4.3 Metrics.

Part segmentation For the quantitative evaluation of the consistent part parsing as a part segmentation task, we use the standard label IoU, following the previous studies [18, 19, 26, 52]. As our method is unsupervised, we follow the standard initial part labeling procedure using a training set to assign each part a ground-truth label for evaluation purposes following [26, 52].

First, for each surface point sampled from the ground-truth part mesh of the instance of the training set, we determine the nearest reconstructed part and vote for the ground-truth part label of that point. Next, we assign each reconstructed part to the part label that has the highest number of votes. Finally, for each surface point sampled from the instance in the test split, we determine the nearest reconstructed part surface and assign the part label of the reconstructed part.

To visualize part segmentation, similar to [124], We first measure the distance between a barycentric point of a ground-truth mesh face to the surface of each part. Then we assign a mesh face the label of the part with the shortest distance to the barycentric point. Lastly, we color each face according to the obtained label.

Part pose evaluation For the part pose evaluation, we evaluate the 3D motion flow of the deformation from the canonical pose to the predicted pose as the endpoint error (EPE) [135], which is a commonly used metric for pose estimation of articulated objects [128, 11]. We scale it by 100 in experiment results.

	Drawer	Eye-glasses	Oven	Laptop	Washing machine
Training	24	35	30	73	39
Test	6	7	7	13	6
# of parts	(1 3 0)	(1 0 2)	(1 0 1)	(1 0 1)	(1 0 1)

Table 5.2 Number of samples per category in each data split. Each sample is augmented by transforming its part pose to generate 100 instances. Numbers in a parethnesis in the last row indicates the ground-truth number of fixed, prismatic and revolute type parts.

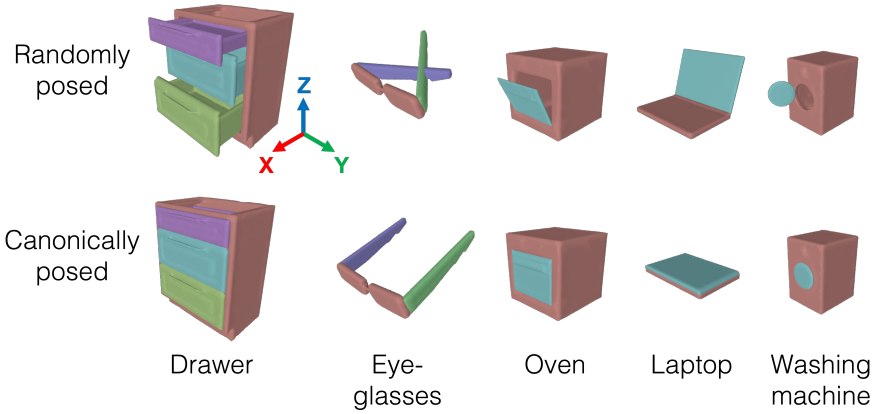


Figure 5.8 Visualization of the canonically posed and randomly posed ground-truth meshes of each category. The colors correspond to the different ground-truth part labels. The colored arrows show the axis directions of the coordinate system used in this paper.

5.4.4 Semantic capability

We evaluate the semantic capability of our approach in part parsing. As part decomposition approaches aim to learn 3D structure reasoning with *as small a number of ground-truth labels as possible*, it is preferable to obtain the initial manual annotations with as *few numbers of shapes* as possible. This requirement is essential for articulated objects, which have diverse shape variations owing to the different articulations. As our approach is part pose consistent, we only need a minimal variety of instances for the initial manual labeling. To verify this, we evaluate the part segmentation performance using only the canonically posed (joint states were all zero) samples in the training set.

We visualize the qualitative part segmentation results of the proposed approach in Figure 5.9, and the part segmentation results given various part poses in Figure 5.10.

The show the quantitative results in Table 5.3. Our model uses a much smaller number of parts than BSP-Net [19]; however, it still performs the best. This shows that our model

	Drawer	Eye-glasses	Oven	Laptop	Washing machine	mean	# of parts
BAE [18]	6.25*	11.11*	73.06	25.11*	80.30	39.17	1.42/8
BSP [19]	66.31	**70.69**	81.65	76.68	87.92	76.65	27.50/256
NSD [52]	38.39	42.11	74.67	74.44	89.11	63.75	10
NP [96]	60.57	64.69	**85.41**	86.23	74.65	74.31	5
Ours	**74.73**	66.18	82.07	**86.81**	**95.15**	**80.99**	4.16/8

Table 5.3 Part segmentation performance in label IoU. Higher is better. The starred numbers indicate the failure of part decomposition and that only one recovered part represents the entire shape. The average and the predefined maximum numbers of recovered parts or primitives are shown before and after the slash, in the last column.

is more parsimonious, and each part has more semantic meaning in part parsing. The segmentation results compared against the baselines are visualized in Figure 5.11.

To eliminate differences in the number of parts and primitives for each method, Table 5.4 shows the result when each method's maximum number of parts and primitives is aligned to $N = 8$. Our method outperforms the previous works by a large margin.

We also visualize the generated part shapes in Figure 5.12. We can see that a single part shape successfully reconstructs the complex target shape, such as disconnected shapes that a single primitive shape cannot express. Also, our part shapes are more semantic and interpretable than the previous works. This demonstrates the advantage of using non-primitive-based part shape representation. As we can see in the improved part segmentation performance, our approach realizes semantically more consistent part decomposition without a complicated mechanism such as grouping primitive shapes based on part kinematics.

Part segmentation using all the training samples In the above evalution, we show that our method works most efficiently by requiring instances with only a limited variety of poses for the initial annotations. We use canonically posed shapes, visualized in Figure 5.8, in the training set for the initial annotations. This section reports the evaluation setting where annotations of all training instances are available for the initial annotation, which is a favorable setting for the baselines. However, the annotation cost can be much higher in reality than in the previous setting.

The results are shown in Table 5.6. Even under this setting favorable for the previous works, our method performs comparably with the state-of-the-art part decomposition method BSP-Net [19] using 256 primitives. It is not surprising that using many primitives achieves fewer part segmentation errors because, even when one primitive is inconsistently assigned to

	Label IoU ↑
BAE [18]	39.17
BSP [19]	66.79
NSD [52]	59.46
NP [96]	70.71
Ours	**80.99**

Table 5.4 Label IoU with the aligned number of primitives and parts for all methods ($N = 8$).

the ground-truth part, the impact on the label IoU is smaller. This is because a smaller portion of the evaluation points becomes erroneous compared with the model using fewer parts or primitives. Note that our research focuses on representing ground-truth articulated parts with consistently the same reconstructed parts by considering the part kinematics, unlike BSP-Net and the other baselines, which can assign different sets of primitives to the same articulated parts without considering the underlying part pose. To show the effectiveness of considering the part kinematics, we show the performance drop from using all training instances to using only the canonically posed instances in the table under the heading "Difference." We can see that our approach has the second best drop with the comparable number with Neural Parts [96], yet higher part parsing performance. This shows that considering the part kinematics contributes to label efficiency by reducing the necessary initial annotation to perform well on the unsupervised part segmentation of articulated objects.

5.4.5 Disentanglement between the part shapes and poses.

Because our approach disentangles shape supervision into part shapes and poses, it realizes pose-aware part decomposition. To verify the learned disentanglement, we visualize the interpolation results of part shapes and joint states as part poses in Figure 5.17. In the middle row, we show the shape interpolation between the source and the target while fixing the joint state s_i of the source to maintain the same part pose. The shape is smoothly deformed from the source to the target maintaining the original pose. In the bottom row, we interpolate the joint state s_i between the source and the target; the joint state changes from the source to the target maintaining the shape identity of the source shape. Our model successfully disentangles the part shapes and poses, unlike previous methods as shown in the top row.

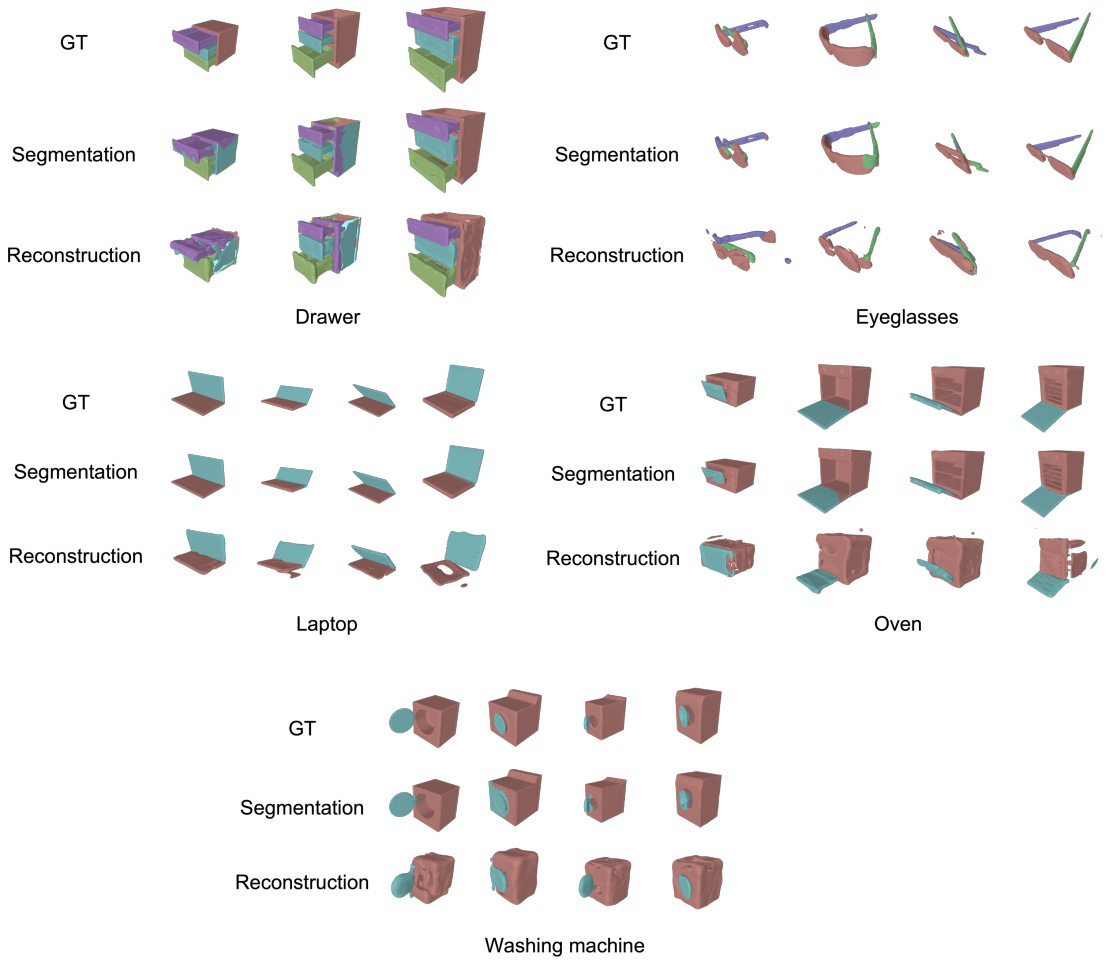


Figure 5.9 Visualization of the part segmentation results of the proposed approach with various samples. For drawer category, the different between some GT shapes are subtle (e.g., difference in handle shapes), we pick the three samples with distinct shape difference to avoid confusion.

5.4.6 Part pose estimation

To validate whether the predicted part decomposition is based on the reasonable part pose estimation, we quantitatively evaluate part pose estimation. Because we train our model without specifying a canonically posed shape, we use the part pose transformations between the target instance and the canonically posed instance of the same sample as the estimated part pose to align with the prediction of the supervised baseline, NPCS [63]. Note that NPCS assumes that part segmentation supervision and ground-truth of part-wise rigid-body transformations as part pose are available during training, and part kinematic type per part is known, which we do not assume. Therefore, NPCS offers an upper bound for our

	Drawer	Eye-glasses	Oven	Laptop	Washing machine	mean
NPCS [63] (Supervised)	1.598	1.087	2.702	0.751	1.594	1.546
Ours (Unsupervised)	3.452	2.631	3.360	2.546	2.529	2.903

Table 5.5 Part pose estimation performance in EPE. Lower is better. NPCS is trained with ground-truth for both part labels and part-wise rigid-body transformations as part pose, offering an upper bound for our unsupervised approach.

	Drawer	Eye-glasses	Oven	Laptop	Washing machine	mean (All)	mean (Canonical)	Difference (All - Canonical)	# of parts
BAE [18]	6.25	11.11	73.01	25.11	80.32	39.16	39.17	-0.01	1.42/8
BSP[19]	70.29	**74.96**	**89.40**	86.21	**95.28**	**83.23**	76.65	6.58	27.50/256
NSD [52]	38.56	44.06	74.63	74.40	89.01	64.13	63.75	0.39	10
NP [96]	60.56	64.75	85.33	86.22	74.72	74.32	74.31	**0.01**	5
Ours	**74.83**	66.25	82.06	**86.80**	95.18	81.02	**80.99**	0.04	4.16/8

Table 5.6 Part segmentation performance. We use all the instances in the training set to assign a label to each part as well as to the primitives. “Canonical" denotes the mean label IoU only using the canonically posed instances of the training for the label assignment. “Difference" shows the performance drop from the setting that uses all the instances in the training set to the setting that uses only the canonically posed instances. The average and the predefined maximum numbers of recovered parts or primitives are shown before and after the slash, in the last column. Our method achieves the same level of the label efficiency with Neural Parts with higher part segmentation performance.

unsupervised approach. We present the evaluation results in Table 5.5. Our method is comparable with NPCS, with the same order of performance. Note again that we are not attempting to outperform supervised pose estimation methods; rather, we aim to show that our unsupervised approach can decompose parts based on reasonable part pose estimation.

We also evaluate the accuracy of our joint parameter estimation. Note that, because we train our model in an unsupervised fashion, the part kinematic types of the ground-truth and the assigned reconstructed part do not necessarily match. Moreover, multiple reconstructed parts may be assigned to one ground-truth part. Therefore, we choose EPE as the primary evaluation metric for part pose estimation due to its kinematic type agnostic property and calculation based on point correspondence between prediction and ground-truth, rather than part-level correspondence. To avoid the problem of part pose evaluation in unsupervised learning described above, we evaluate the accuracy of joint parameter estimation by considering the prediction is correct when the following three conditions are

	Drawer	Eye-glasses	Oven	Laptop	Washing machine	mean
# of assigned parts	1.0	1.0	1.0	1.0	1.0	1.0
Part type accuracy	89.50	83.25	100.0	92.14	100.0	91.46

Table 5.7 Part assignment evaluation. The first row shows the number of reconstructed parts assigned to the ground-truth parts, and the second row shows the accuracy of part kinematic type matches between the ground-truth and the assigned reconstructed parts for dynamic part types.

all satisfied. (1) One reconstructed part is assigned to one ground-truth dynamic part. (2) The part kinematic type is the same between the ground-truth and the assigned reconstructed part. (3) The error of the joint parameters against the ground-truth is less a threshold. This evaluation method is more challenging than EPE because of the influence of (1) and (2) above, besides the prediction error of the joint parameters. We evaluate joint state accuracy and joint direction accuracy. Only for the revolute part, we also evaluate joint axis distance accuracy, defined as the line to line distance between the ground-truth and the predicted line segments consisting of the pivot point and the joint direction.

Figure 5.13 shows the evaluation results with varying error thresholds. We show the results of NPCS only as a reference; NPCS is a supervised model and assumes that the part segmentation is available during training, and the part kinematic types are also known. In contrast, our method learns both part segmentation and part kinematic type in an unsupervised fashion. Since NPCS does not estimate the pivot point, we only show the results of our method for joint axis distance accuracy. As for the joint state, we see reasonable accuracy of 70.80% for revolute parts on average when the threshold is less than 10 degrees and 79.43% when the threshold is 15 degrees. For the “prismatic" part of the drawer, our method outperforms the NPCS when the threshold is less than 0.1. For joint direction estimation, in three out of five categories (eyeglasses, laptop, and oven), our method is comparable or outperforming NPCS. In Table 5.7, we also show the number of reconstructed parts assigned to the ground-truth parts and the accuracy of part kinematic type of dynamic parts matches between the ground-truth and the assigned reconstructed parts. In all categories, the model correctly assigns one part. Moreover, even without part type supervision, our model successfully predicts correct part types with high accuracy of 91.46%. Improving the unsupervised learning of joint parameters under shape supervision is an interesting research direction.

	$\mathscr{L}_{\text{volume}}$	$\mathscr{L}_{\text{deviation}}$	$\mathscr{L}_{\text{pivot}}$	$\mathscr{L}_{\text{variation}}$	$\mathscr{L}_{\text{generator}}$	VQ	CS	CP	Full
Label IoU ↑	72.20	73.21	74.27	65.29	70.14	72.78	55.67	71.35	**80.99**
EPE ↓	4.362	6.628	9.250	6.676	7.276	10.772	8.827	7.219	**2.988**

Table 5.8 Ablation study of the losses and the proposed components: VQ, CP and CS indicates disabling the use of multiple constant vectors introduced in Section 5.3.1, the category-common pose decoder, and the category-common shape decoders, respectively. "Full" means using all the losses and the components.

	Label IoU ↑	EPE ↓
Complete	80.99	2.903
Depth	80.65	3.203

Table 5.9 Comparison between the point cloud input types: complete shape and depth map.

5.4.7 Ablation studies

We evaluate the effect of the proposed losses, the multiple constant vectors for multi-modal category-common pose bias learning, and the category-common decoders on part segmentation and part pose estimation. We disable each loss and component one at a time. We only use the corresponding instance-dependent decoder(s) when disabling the category-common decoders for pose and shape. The results are shown in Table 5.8. Enabling all losses and the components performs the best. Particularly, disabling the category-common shape decoders significantly degrades both label IoU and EPE. This indicates that learning category-common shape prior is essential to perform proper part decomposition and to facilitate part pose learning, which is the core idea of this study. We visualize the qualitatie results of turning off the category common decoders in Figure 5.15. Colors indicate the part IDs. When we remove caninical shape decoder (CS), we frequently find unsuccessful decomposition; a single part spans two GT parts (purple), and the shape deformation accounts for the shape variation. The boxes show the parts with their joint states set to 0. With CS, the variation is expressed by part poses with successful decomposition. Without canonical pose decoder (CP), similarily with CS, we find that the model degenerates to express shape variation by different part poses by shape deformation. During the training, we find turning off $\mathscr{L}_{\text{pivot}}$ for correct pivot point localization makes the traning difficult to converge to preferable decomposition, especially for eyeglasses category. As visualized in Figure 5.16, with $\mathscr{L}_{\text{pivot}}$, the pivot point locates the proper position between parts even at the early stage of the training. However, without $\mathscr{L}_{\text{pivot}}$, the pivot points are off from the reconstructed shape.

5.4.8 Depth map input and real data

Because PPD's decoders do not assume a complete shape as an input, it works with depth map input. Following BSP-Net [19], we train a new encoder that takes a depth map captured from various viewpoints as a partial point cloud and replace the original encoder. We minimize the mean squared error between the output latent vectors of the original and the new encoders so that their output are close for the same target shape. The results are shown in Table 5.9. The depth map input performs comparably to the complete point cloud input. We also verify that our model trained on synthetic depth maps reasonably generalizes to real data, as shown in Figure 5.14.

5.5 Failure cases and limitation

As illustrated in Figure 5.18 (a), mixing different kinematics model during unsupervised learning is left for future work.

Also, our method tend to result in incosistent part decomposition when size is diverse, as shown in 5.18 (b). We found our drawer category is more challenging to converge well than the other categories, resulting in degenerated quantitative performance. Our canonical shape decoder learns category-specific mean part shapes. The decoder also models canonical part locations for the prismatic part, unlike a revolute part location modeled by its pivot point. Thus, deviating largely from the mean part shapes and large part location difference caused by the size difference weakens the effectiveness of the canonical shape decoder, leading to the semantically less consistent part decomposition. Due to the small number of parts used by our model, misclassifying a single part could drop the segmentation performance significantly ($74.83 \rightarrow 60.97$). Note that even such a model performs comparably with the leading primitive-based part decomposition method [96] when the number of parts is aligned, as shown in Table 5.4. One possible extension to tackle this problem can be learning to model size in addition to the shape and pose for each part and letting the canonical shape decoder learn part shape in normalized space in terms of part pose and size.

Because pose-aware part decomposition without explicit supervision is a highly ill-posed task, as a limitation, our method requires manual initialization of part types and joint directions for each part to stabilize the training process, as described in Section 5.3.1. Although the manual initialization, part decomposition induction by pose constraints with joint parameters, and the proposed losses contribute to stabilizing the training process, different model initialization and stochastic training may result in different part decomposition results due to the unsupervised nature of the approach and the ill-posed target problem. In

our experiments, we tried a few random seeds when part decomposition failed in the early stages of the traning for the models reported in Table 5.3 of the main paper.

5.6 Conclusion

We propose a novel unsupervised generative part decomposition method, PPD, for man-made articulated objects considering part kinematics. We show that the proposed method learns the disentangled representation of the part-wise implicit field as the decomposed part shapes and the joint parameters of each part as the part poses. We also show that our approach outperforms previous generative part decomposition methods in terms of semantic capability and show comparable part pose estimation performance with the supervised baseline.

As shown in qualitative results, our generative method achieves reasonable part shape reconstruction reflecting target shape variations sufficient to induce part decomposition and challenging joint parameter learning. As a limitation, our method currently fails to capture details of the target shapes up to the primitive-based previous works [52, 19], focusing on the shape reconstruction performance rather than part pose consistency. Also, joint parameter learning requires manual initialization of joint direction and part types for each part. The future work will address the above limitation.

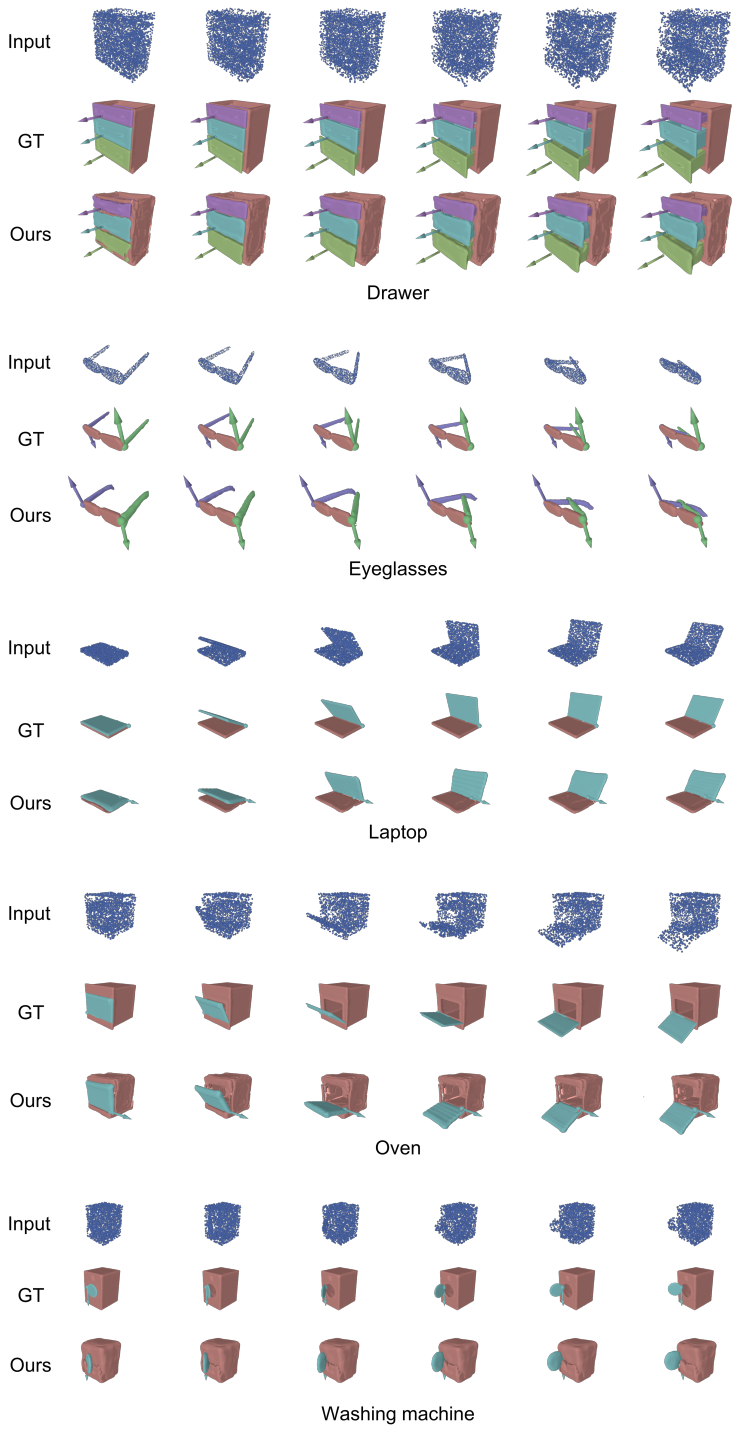


Figure 5.10 Visualization of the part segmentation results given input shapes with various part poses. Arrows in the figure indicate the ground-truth or predicted joint directions.

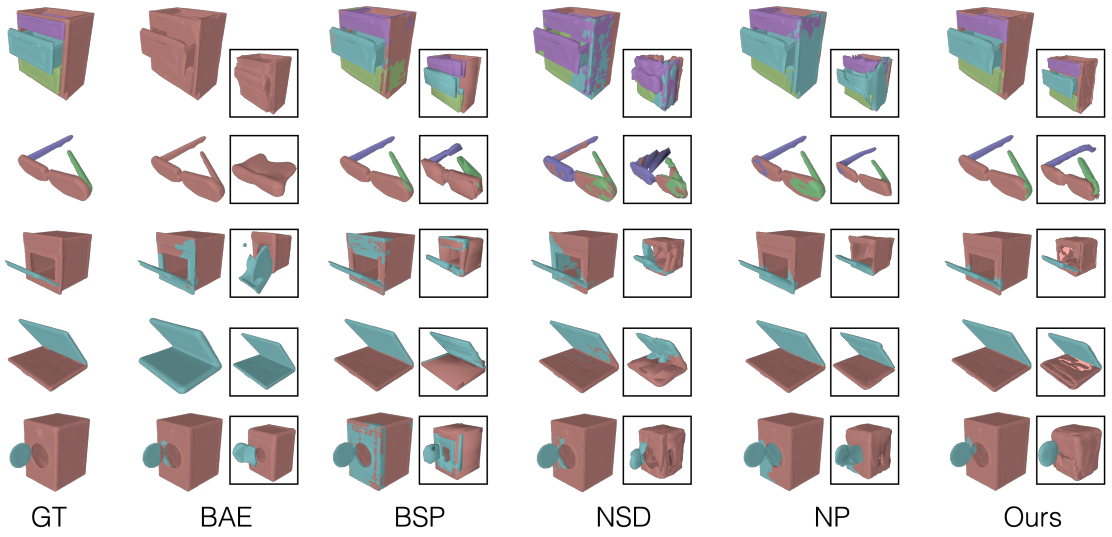


Figure 5.11 Qualitative result of the part segmentation compared to the baselines. Reconstructed shape in mesh is shown inside a box. The same color indicates the same segmentation part.

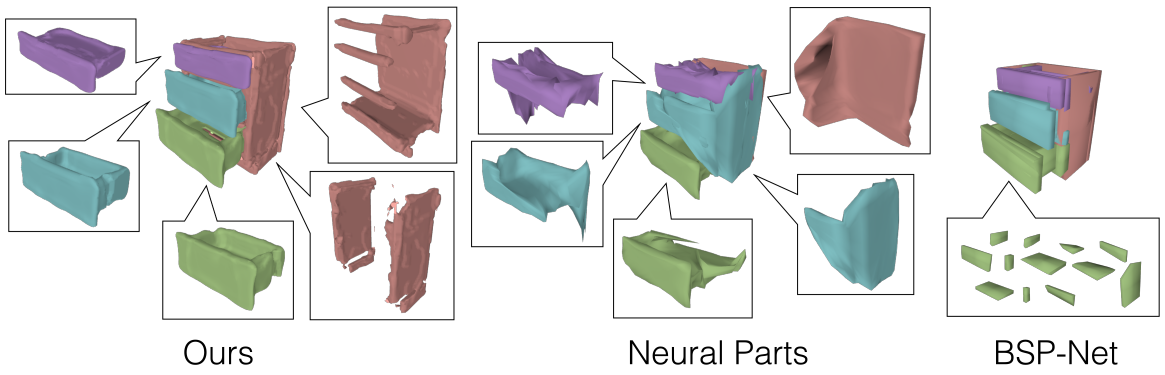


Figure 5.12 Visualization of parts and primitives. The boxes represent the parts or primitives used to reconstruct the semantic parts.

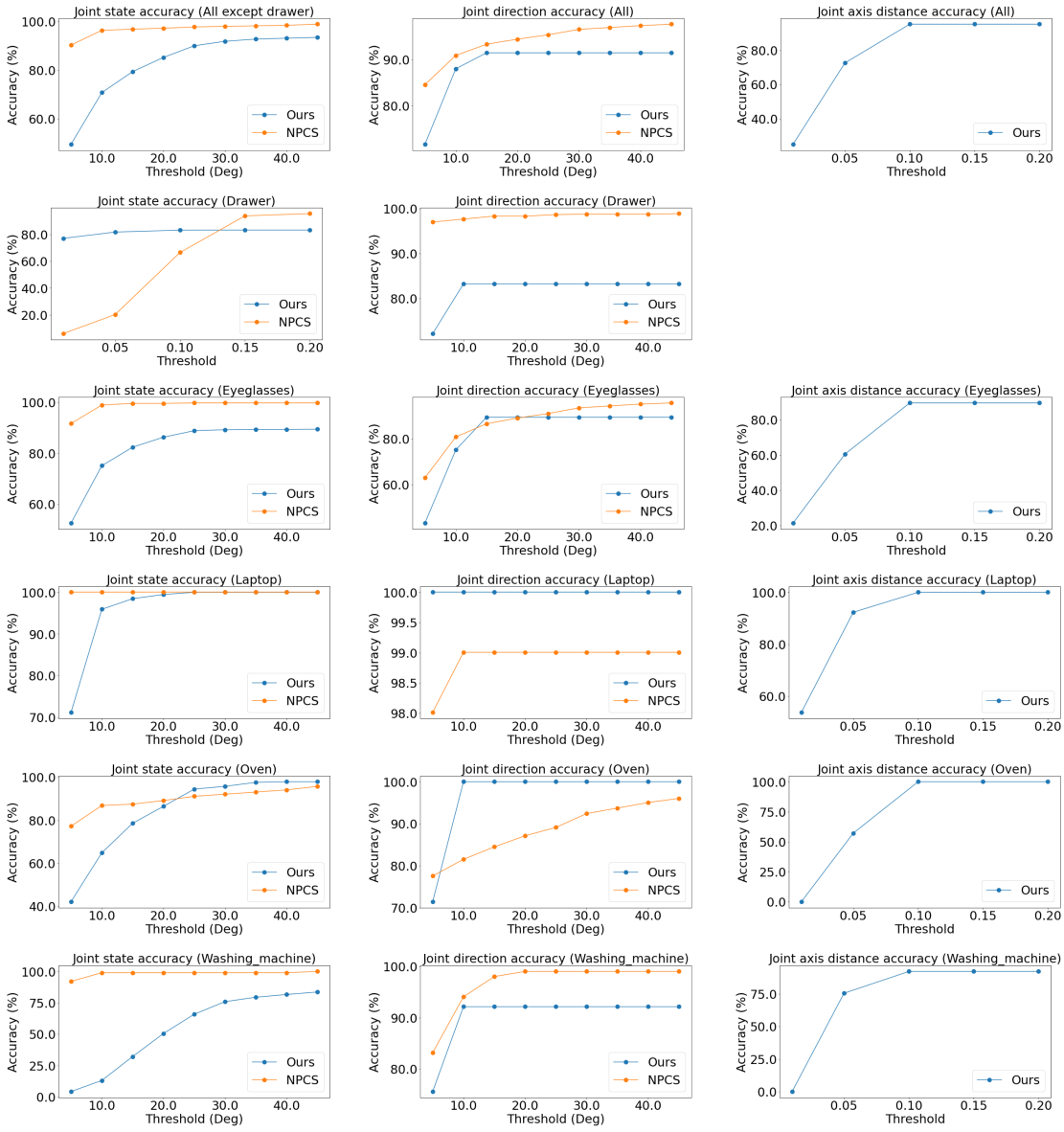


Figure 5.13 Joint parameter estimation performance.

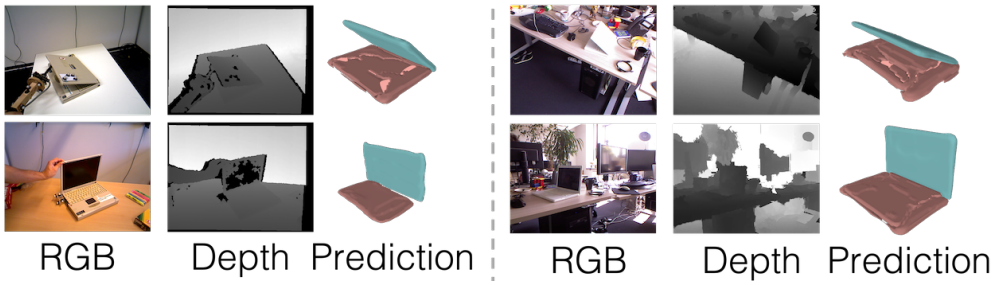


Figure 5.14 Real depth map input. (Left) RBO dataset [76] and (Right) Articulated Object Dataset [78].

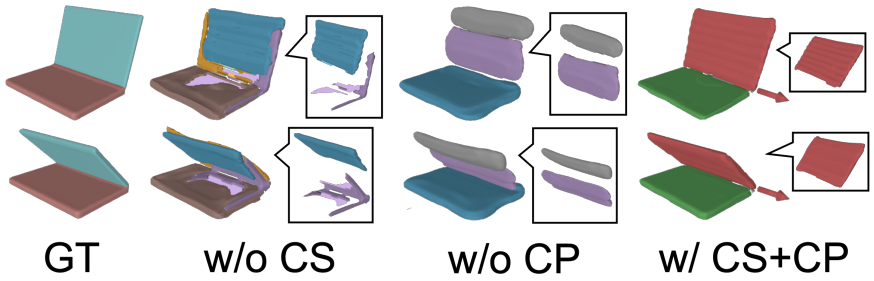


Figure 5.15 Qualitative ablation of CS and CP. The arrow indicates the predicted revolute direction.

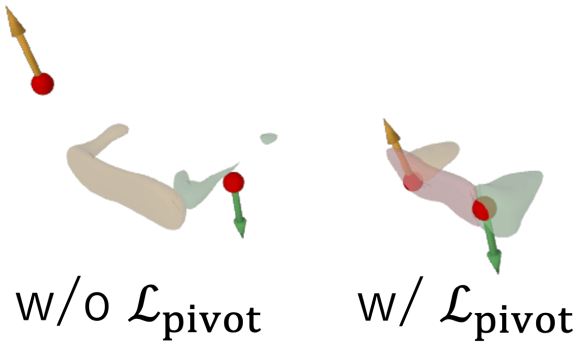


Figure 5.16 Qualitative ablation on $\mathcal{L}_{\text{pivot}}$ at training step 1.3k. The red sphere shows the pivot point, and the arrow indicates the joint direction.

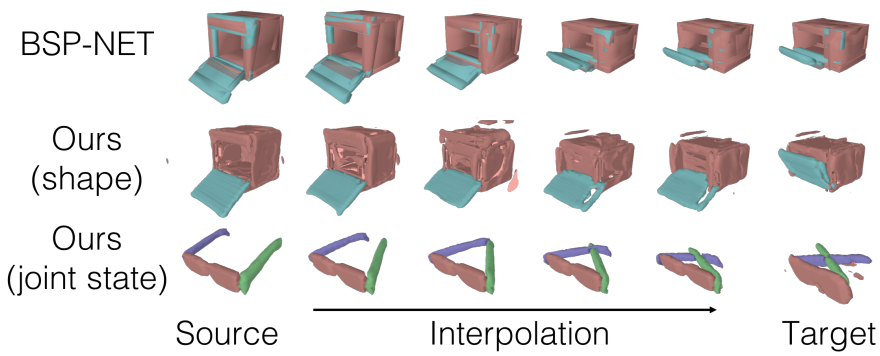


Figure 5.17 Interpolation in terms of disentangled part shapes and joint states as part pose.

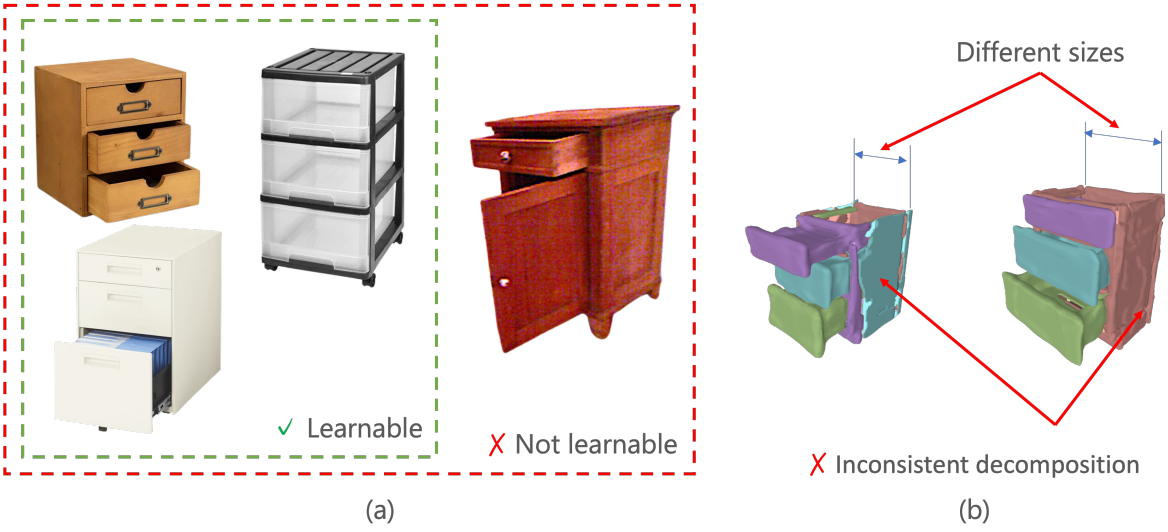


Figure 5.18 Limitation and failure cases. (a) Consistent kinematics model as limitation for unsupervised learning. (b) Failure cases on categories with large size difference.

Chapter 6

Unsupervised Decomposition of Shape into Finer Semantic Parts

6.1 Introduction

Understanding 3D objects by decomposing them into simpler shapes (termed primitives) has been widely studied in computer vision [107, 8, 7]. Decomposing 3D objects into parsimonious and semantic primitive representations is important for understanding their structure. Constructive solid geometry [61] uses combinations of primitives to reconstruct complex shapes.

Recently, learning-based approaches have been adopted to primitive based approaches [19, 26, 28, 94, 95, 91, 124]. It has been demonstrated that these approaches enable a semantically consistent part arrangement in various shapes. Moreover, the use of implicit representations allows the set of primitives to be represented as a single collective shape by considering a union [19, 26, 33]; this can improve the reconstruction accuracy during training.

However, the expressiveness of primitives, particularly those with closed shapes, has been limited to simple shapes (cuboids, superquadrics, and convex shapes). Although primitives can learn semantic part arrangements, the semantic shapes of the parts cannot be learned using existing methods. In addition, although the union of primitive volumes could be represented by implicit representations in previous studies, the lack of immediate access to the union of primitive surfaces during training results in complex training schemes [19, 26, 33].

It is challenging to define a primitive that addresses all these problems. State-of-the-art expressive primitives with explicit surfaces do not have implicit representations [35, 28], and thus they cannot efficiently consider unions of primitives to represent collective shapes.

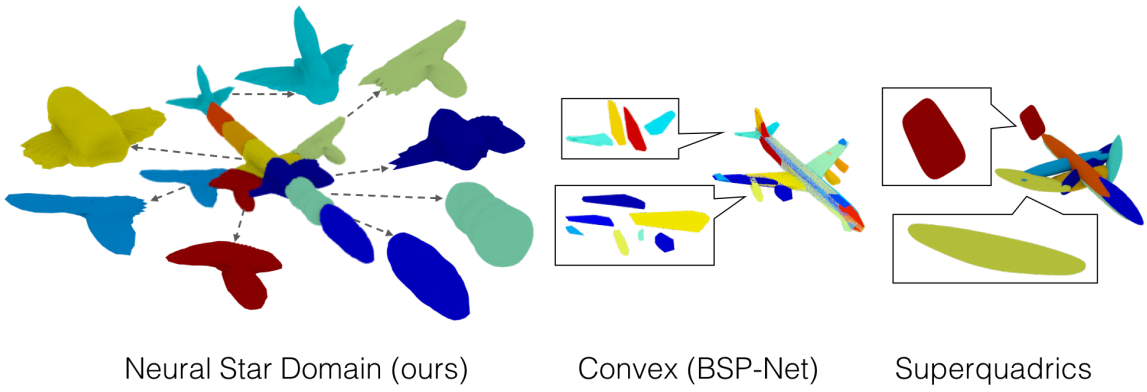


Figure 6.1 Overview of proposed approach. The primitives have a more meaningful and wider shape variety compared with those in previous studies.

Leading primitive representations by convex shapes [19, 26] with implicit representations involve a tradeoff regarding the number H of half-space hyperplanes defining a convex. Using more hyperplanes yields more expressive convex shapes at the expense of a quadratically growing computation cost in extracting differentiable surface points. A naive implementation costs $O(H^2)$ to filter the surface points of a convex from the hyperplanes.

To address these issues, we propose a novel primitive representation termed neural star domain (NSD) that learns shapes in a star domain by using neural networks. A star domain is a group of arbitrary shapes that can be represented by a continuous function defined on the surface of a sphere. As it can express concavity, we can regard it as a generalized shape representation of convex shapes. The learned primitives are visualized in Figure 6.1. Moreover, we can directly approximate star-domain shapes using neural networks owing to their continuity. We demonstrate that the complexity of the shapes that can be represented by an NSD is equivalent to the approximation ability of the neural network. In addition, as it is defined on the surface of a sphere, a primitive can be represented in both implicit and explicit forms by transforming it between spherical and Cartesian coordinates. The proposed approach is compared with those in previous studies in Table 6.1.

The contributions of this study can be summarized as follows: (1) We propose a novel primitive representation with high expressive power, and we demonstrate that it is more parsimonious and can learn semantic part shapes. (2) We demonstrate that the proposed primitive provides unified implicit and explicit representations that can be used during training and inference, leading to improved mesh reconstruction accuracy and speed.

	Implicit	Explicit	Semantic	Parsimonious	Accurate
DMC [66]	✓	✓	–	–	✓
SQ [94]		✓		✓	
AtlasNetV2 [28]		✓	✓	✓	✓
BSP-Net [19]	✓				✓
Ours	✓	✓	✓	✓	✓

Table 6.1 Overview of shape representations in previous studies. SQ stands for superquadrics [94]. We regard a primitive as having an explicit representation if it can access the explicit surface in *both* the inference and the training process. Moreover, a primitive representation is said to be semantic if it can reconstruct semantic *shapes* in addition to part correspondence.

6.2 Related work

Methods for decomposing shapes to primitives have been studied extensively in computer vision [107]. Some of the classical primitives used in computer vision are generalized cylinders [8] and geons [7]. In deep generative models, cuboids [124, 91] and superquadrics [94, 95] are used to realize consistent parsing across shapes. However, these methods have poor reconstruction accuracy owing to the limitations in the parameter spaces of the primitives. Thus, their application is limited to shape abstraction. Using parametrized convex shapes for improved reconstruction accuracy has been recently proposed in [19, 26]. However, as the shapes of the primitives are constrained to be convex, their interpretability is limited to part parsing. In this study, we investigate star domains as primitive representations with more expressive power than that of previously proposed primitive representations.

In computation theory, 2D polygonal shape decomposition using star domains has a long history [22, 54]. In computer vision, star domains have been used to abstract 3D shapes to *encode* shape embedding [68, 100, 23, 58] for discriminative models. In contrast, we study the application of star domains to *decode* shape embedding to accurately reconstruct 3D shapes for generative models.

Surface representation of 3D objects in the context of generative models has been studied extensively. In recent studies, the standard explicit shape representation for generative models is a mesh [35, 50, 102, 127, 35]. Meshes [30], pointclouds [28], and parametrized surfaces [124, 91, 94, 95] have been studied as explicit surfaces for primitive models. A state-of-the-art method employs a learnable indicator function for non-primitive- [77, 93] and primitive-based approaches [33, 19, 26]. However, extracting a surface mesh during inference is quite costly, as the isosurface extraction operation grows cubically for the desired meshing resolutions. An implicit representation model with fast polymesh sampling during inference

was proposed in [19]. However, owing to the lack of explicit surface representations during training, primitive-based methods with implicit representations require complicated training schemes, such as near-surface training data sampling with ray casting [33, 26], and heuristic losses to keep primitives inside the shape boundary [26], or a multi-stage training strategy to approximate explicit surfaces [19]. A notable exception that uses both implicit and explicit representations was proposed in [66]; however, this is possible by reconstructing the shape as a voxel at the cost of limited shape resolution. In this study, we propose a unified shape representation in both explicit and implicit forms at an arbitrary resolution. This is used to realize a simple training scheme with fast high-resolution mesh sampling during inference.

6.3 Methods

We first formulate the problem setting in Section 6.3.1. Subsequently, we define star domains in Section 6.3.2. In addition, we introduce NSDs to approximate shapes in star domains, with a theoretical analysis of the representation power. Using NSDs as building blocks, we describe the pipeline of the proposed approach in Sections 6.3.3, 6.3.4, and 6.3.5. Implementation details are provided in Section 6.3.6.

6.3.1 Problem setting

We represent an object shape as a set of surface points $P \subseteq \mathbb{R}^3$, and as an indicator function that can be evaluated at an arbitrary point $\mathbf{x} \in \mathbb{R}^3$ in 3D space as $O : \mathbb{R}^3 \to \{0,1\}$, where $\{\mathbf{x} \in \mathbb{R}^3 \,|\, O(x) = \tau\}$. In this equation, $\tau = 0$ defines the outside of the object, and $\tau = 1$ defines the inside. Our objective is to parametrize the 3D shape by a composite indicator function $\hat{O}$ and surface points $\hat{P}$ that can be decomposed into a collection of N primitives. The ith primitive has an indicator function $\hat{O}_i : \mathbb{R}^3 \to [0,1]$ and a surface point function defined on a sphere $\hat{P}_i : \mathbb{S}^2 \to \mathbb{R}^3$. To realize implicit and explicit shape representation simultaneously, we further require $\hat{O}$ and $\hat{P}$ to be related as $\hat{O}(\hat{p}) = \tau_o$, where $\hat{p} \in \hat{P}$, and $\tau_o \in [0,1]$ is a constant that represents the isosurface. We ensure that both the composite indicator function and the surface points are approximated as $O \approx \hat{O}$ and $P \approx \hat{P}$, respectively, through training losses.

6.3.2 Neural star domain

A geometry $U \subseteq \mathbb{R}^3$ is a star domain if $\exists \mathbf{t} \in U, \forall \mathbf{u} \in U, [\mathbf{t}, \mathbf{u}] = \{(1-v)\mathbf{t} + v\mathbf{u}, 0 \leq v \leq 1\} \subseteq U$. Intuitively, a star domain is any geometry with an origin $\mathbf{t}$ such that a straight line segment between any point $\mathbf{u}$ inside the geometry and $\mathbf{t}$ is also inside the geometry. Thus, we

can regard star domain shapes as continuous functions defined on the surface of a sphere. We denote such functions as $r : \mathbb{S}^2 \to \mathbb{R}$. The spherical harmonics expansion $\mathbb{S}^2 \to \mathbb{R}$ is a multivariate polynomial function that is also defined on the surface of a sphere. Thus, we can formulate a star domain using a spherical harmonics expansion as

$$r(\mathbf{d}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} c_{l,m} Y_{l,m}(\omega(\mathbf{d})),\ \omega(\mathbf{d}) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta), \tag{6.1}$$

where $\mathbf{d} = (\theta, \phi) \in \mathbb{S}^2$, $c_{l,m} \in \mathbb{R}$ is a constant, and $Y_{l,m}$ is the Cartesian spherical harmonic function [126]. The spherical harmonics expansion f_∞ with Cartesian spherical harmonics $Y_{l,m}$ is written as follows:

$$f_\infty(\mathbf{d}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} c_{l,m} Y_{l,m}(\omega(\mathbf{d})),\ \omega(\mathbf{d}) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta), \tag{6.2}$$

where $\mathbf{d} = (\theta, \phi) \in \mathbb{S}^2$, $c_{l,m} \in \mathbb{R}$ is a constant. Examples of $Y_{l,m}$ given (l, m) are shown below:

$$\begin{aligned}
Y_{0,0}(x,y,z) &= \frac{1}{2}\sqrt{\frac{1}{\pi}} \\
Y_{1,-1}(x,y,z) &= \sqrt{\frac{3}{4\pi}}y \\
Y_{1,0}(x,y,z) &= \sqrt{\frac{3}{4\pi}}z \\
Y_{1,1}(x,y,z) &= \sqrt{\frac{3}{4\pi}}x \\
Y_{2,-2}(x,y,z) &= \frac{1}{2}\sqrt{\frac{15}{\pi}}xy & Y_{2,-1}(x,y,z) &= \frac{1}{2}\sqrt{\frac{15}{\pi}}yz \\
Y_{2,0}(x,y,z) &= \frac{1}{4}\sqrt{\frac{5}{\pi}}(-x^2-y^2+2z^2) \\
Y_{2,1}(x,y,z) &= \frac{1}{2}\sqrt{\frac{15}{\pi}}zx & Y_{2,2}(x,y,z) &= \frac{1}{4}\sqrt{\frac{15}{\pi}}(x^2-y^2)
\end{aligned}$$

To realize the star domain primitive, we propose an NSD, which approximates r by a neural network f_{NN}, taking $\omega(\cdot)$ as input.

Approximation ability We demonstrate the universal approximation ability of the NSD to a star domain r. The following theorem implies that r can be arbitrarily approximated by an NSD.

Theorem. *Let $r : \mathbb{S}^2 \to \mathbb{R}$ be a continuous function on the surface of a sphere. Then, $\forall \varepsilon > 0$, $\exists$ an NSD $f_{NN} \circ \omega : \mathbb{S}^2 \to \mathbb{R}$ such that for any $\mathbf{d} \in \mathbb{S}^2$, we have*

$$|r(\mathbf{d}) - f_{NN}(\omega(\mathbf{d}))| < \varepsilon. \tag{6.3}$$

Proof. By the completeness of spherical harmonics [13] to a continuous function on a spherical surface, as shown in Equation (6.1), $\forall \varepsilon_1 > 0$, $\exists L \in \mathbb{N}^+$ and $c_{l,m} \in \mathbb{R}$ such that for any $\mathbf{d} \in \mathbb{S}^2$, we have

$$|r(\mathbf{d}) - r_L(\mathbf{d})| < \varepsilon_1, \text{ where } r_L(\mathbf{d}) = \sum_{l=0}^{L} \sum_{m=-l}^{l} c_{l,m} Y_{l,m}(\omega(\mathbf{d})). \tag{6.4}$$

ω can be regarded as $Y_{1,m}$ with an appropriate constant $c_{1,m}$, and from the definition of Cartesian spherical harmonics [126], each $Y_{l,m}$ with $l > 1$ can be written as a polynomial function of $Y_{1,m}$ with an appropriate constant $c_{l,m}$. Thus, r_L can be regarded as a polynomial function over ω, that is, it is continuous over ω.

By the universal approximation theorem of neural networks to a continuous function [41, 2], $\forall \varepsilon_2 > 0$, $\exists$ a neural network $f_{NN} : \mathbb{R}^3 \to \mathbb{R}$ such that for any $\omega_{\mathbf{d}} \in \{\omega(\mathbf{d}) \,|\, \mathbf{d} \in \mathbb{S}^2\}$, we have

$$|r_L(\mathbf{d}) - f_{NN}(\omega_{\mathbf{d}})| < \varepsilon_2. \tag{6.5}$$

Given Equations (6.4) and (6.5), $\forall \varepsilon > 0$, $\exists$ a neural network $f_{NN} : \mathbb{R}^3 \to \mathbb{R}$ such that for any $\mathbf{d} \in \mathbb{S}^2$, we have

$$|r(\mathbf{d}) - f_{NN}(\omega(\mathbf{d}))| < \varepsilon_1 + \varepsilon_2 = \varepsilon. \tag{6.6}$$

□

It should be noted that there exist network architectures that take the output values of trigonometric functions as input, such as HoloGAN [86]. However, the proposed approach differs in the input and output as follows: (1) By taking ω as input, the proposed approach is theoretically founded on an approximate spherical harmonic expansion. HoloGAN takes the output values of high-degree trigonometric polynomial functions as input. (2) The neural network in HoloGAN is aimed at predicting high-dimensional vectors as images, whereas the proposed approach is aimed specifically at predicting a single-dimensional radius r by approximating the star domain.

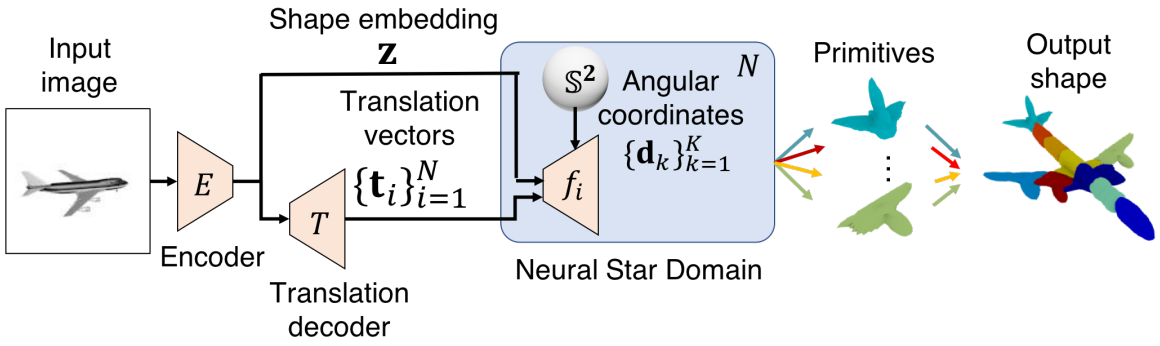


Figure 6.2 Architecture of NSDN.

6.3.3 Primitive representation

As an NSD is defined on the surface of a sphere, one can define both implicit and explicit shape representations of a primitive. For simplicity, we define an NSD $f := f_{NN} \circ \omega$ in the following sections.

Implicit representation Given the 3D location $\mathbf{x} \in \mathbb{R}^3$, an indicator function $\hat{O}_i : \mathbb{R}^3 \to [0,1]$ for the ith primitive located at $\mathbf{t}_i$ is expressed as follows:

$$\hat{O}_i(\mathbf{x};\mathbf{t}_i) = \text{Sigmoid}(\alpha(1 - \frac{\|\bar{\mathbf{x}}\|}{r^+})), \text{ where } \bar{\mathbf{x}} = \mathbf{x} - \mathbf{t}_i,\ r^+ = \text{ReLU}(f_i(G(\mathbf{x}))), \tag{6.7}$$

where α is a scaling factor that adjusts the margin of the indicator values between the inside and outside of the shape, $G : \mathbb{R}^3 \to \mathbb{S}^2$ denotes the conversion from 3D Cartesian coordinates to the spherical surface, and the ReLU operator ensures that the estimated radius is a non-negative real number. We note that $\|\bar{\mathbf{x}}\| - r^+$ can be interpreted as a singed distance function. We define the conversion from Cartesian coordinates to the surface of the sphere $G : \mathbb{R}^3 \to \mathbb{S}^2$ as

$$G(x,y,z) = (\arctan\frac{y}{x}, \arctan\frac{\sqrt{x^2+y^2}}{z^2}) \tag{6.8}$$

We define the conversion from spherical coordinates to Cartesian coordinates $G^{-1} : \mathbb{R} \times \mathbb{S}^2 \to \mathbb{R}^3$ as

$$G^{-1}(r,\theta,\phi) = (r\sin\theta\cos\phi, r\sin\theta\sin\phi, r\cos\theta). \tag{6.9}$$

Explicit representation With a slight abuse of notation, we denote the conversion from spherical coordinates to a 3D location as $G^{-1} : \mathbb{R} \times \mathbb{S}^2 \to \mathbb{R}^3$. We can sample a surface point

in the direction of $\mathbf{d}$ from the origin of the ith primitive located at $\mathbf{t}_i$ as follows:

$$\hat{P}_i(\mathbf{d};\mathbf{t}_i) = G^{-1}(r^+,\mathbf{d}) + \mathbf{t}_i, \text{ where } r^+ = \text{ReLU}(f_i(\mathbf{d})). \tag{6.10}$$

6.3.4 Neural star domain network

To represent the target shape as a collection of primitives, we define an NSD Network (NSDN), which employs a bottleneck auto-encoder architecture similar to that in [77]. An NSDN consists of an encoder E, a translation network T, and a set of NSDs $\{f_i\}_{i=1}^N$. Given an input I, the encoder E derives a shape embedding $\mathbf{z}$. Then, the translation network T outputs a set of translation vectors $\{\mathbf{t}_i\}_{i=1}^N$ from $\mathbf{z}$. The translation vectors represent the location of each primitive. The ith NSD f_i acts as a decoder and infers the radius given an angular coordinate $\mathbf{d}$, translation vectors $\mathbf{t}_i$, and a shape embedding $\mathbf{z}$. In this study, we only estimate the location as the pose of the primitives, whereas in previous studies, the scale and rotation of each primitive were additionally predicted [26, 94, 124]. We observe that learning the rotation and scale leads to unsuccessful training. An overview of the architecture is shown in Figure 6.2.

Composite indicator function To derive an implicit representation of the NSDN, we define a composite indicator function as the union of N NSD indicator functions as

$$\hat{O}(\mathbf{x};\{\mathbf{t}_i\}_{i=1}^N) = \text{Sigmoid}(\sum_{i\in[N]} \hat{O}_i(\mathbf{x};\mathbf{t}_i)). \tag{6.11}$$

To encourage gradient learning of all primitives during training, we use the sum of the indicator values over the primitives rather than the maximum value. We treat the threshold of the indicator value τ_o of the surface level of $\hat{O}$ as a hyperparameter.

Surface point extraction Owing to the unified explicit and implicit shape representation of the NSD, the NSDN can extract the union of the surface points of the primitives in a differentiable manner. We define the unified surface points as follows:

$$\hat{P} = \bigcup_i \{\hat{P}_i(\mathbf{d};\mathbf{t}_i) | \forall j \in [N \setminus i], \hat{O}_j(\hat{P}_i(\mathbf{d};\mathbf{t}_i),\mathbf{t}_i) < \tau_s, \mathbf{d} \in \{\mathbf{d}_k\}_{k=1}^K\}, \tag{6.12}$$

where K denotes the number of points sampled from the surface of the sphere, and τ_s is a hyperparameter for the threshold of the indicator value for the surface points.

Normal estimation NSD can also estimate differentiable normal vectors. Unlike methods using mesh templates, the proposed approach can derive normals at arbitrary resolution. Following [77], we derive the surface normal of the ith primitive $\hat{n}_i$ can be derived:

$$\hat{n}_i(\hat{\mathbf{p}};\mathbf{t}_i) = -\frac{\partial \hat{O}_i(\hat{\mathbf{p}};\mathbf{t}_i)}{\partial \hat{p}}, \tag{6.13}$$

where $\hat{\mathbf{p}} \in \hat{P}_i$ is the predicted surface point, $\hat{O}_i$ is the indicator function, and $\mathbf{t}_i$ is the translation vector of the ith primitive. Collective surface normal vectors $\hat{n}$ can be defined as follows:

$$\hat{n} = \bigcup_i \{\hat{n}_i(\hat{\mathbf{p}};\mathbf{t}_i) | \forall j \in [N \setminus i], \hat{O}_j(\hat{P}_i(\mathbf{d};\mathbf{t}_i);\mathbf{t}_i) < \tau_s, \mathbf{d} \in \{\mathbf{d}_k\}_{k=1}^K\}, \tag{6.14}$$

where N is the number of primitives, and τ_s is a hyperparameter for the threshold of the isosurface indicator value.

6.3.5 Training loss

To learn the parameters ■ of the NSDN, we define the *surface point loss*, which minimizes the symmetric chamfer distance between the surface points P from a training sample and those from the predicted surface points $\hat{P}$. The surface point loss is formulated as

$$L_S(■) = \mathbb{E}_{\hat{p}\sim\hat{P}} \min_{p\sim P} \|\hat{p} - p\| + \mathbb{E}_{p\sim P} \min_{\hat{p}\sim\hat{P}} \|p - \hat{p}\|. \tag{6.15}$$

We note that the surface point loss enables learning collective surfaces of primitives by accessing both implicit and explicit representations, as shown in Equation 6.12. The training loss leads to a better reconstruction than minimizing the distance between P and a simple union of the surface points of the primitives $\bigcup_{i\in[N]} \{\hat{P}_i(\mathbf{d};\mathbf{t}_i) | \mathbf{d} \in \{\mathbf{d}_k\}_{k=1}^K\}$, as in [94, 124]. This is because, ideally, the loss should measure the distance between the two sets of surface points. We also use the occupancy loss as in [77] $L_O(■) = \mathbb{E}_{\mathbf{x}\sim\mathbb{R}^3}\text{BCE}(O(\mathbf{x}), \hat{O}(\mathbf{x}))$, where BCE is the binary cross entropy. We observe that using the occupancy loss in addition to the surface point loss achieves the best reconstruction performance.

6.3.6 Implementation details

In all experiments, we use the same architecture, whereas the number of primitives N varies. N is set to 30 by default, unless stated otherwise. We use ResNet18 as the encoder E, which produces shape embedding as a latent vector $\mathbf{z} \in \mathbb{R}^{256}$ for an input RGB image by following OccNet [77]. For the translation network T, we use a multilayer perceptron (MLP) with

three hidden layers with $(128, 128, N*3)$ units with ReLU activation. For an NSD, we use an MLP with three hidden layers with $(64, 64, 1)$ units and ReLU activation. We set the margin α of the indicator function to 100. The threshold τ_o of the composite indicator function is determined by a grid search over the validation set. For example, for $N = 30$, we use $\tau_o = 0.99$. We use 0.1 for the threshold τ_s of surface point extraction. During training, we use a batch size of 20, and train with the Adam optimizer, with a learning rate of 0.0001. We set the weight of L_o and L_s as 1 and 10, respectively. For the training data, we sample 4096 points from the ground-truth pointcloud, and $400*N$ samples from the generated shape for the surface point loss L_s; moreover, we sample 2048 points from the ground-truth indicator values for the indicator loss L_o. For mesh sampling, we use a spherical mesh template.

6.4 Experiments

Dataset In the experiments, we use the ShapeNet [15] dataset. Following [77], we test the proposed approach on 13 categories of objects. In addition, we use the same samples and data split as in [77]. For 2D images, we use the rendered view provided in [21]. For the quantitative evaluation of the part semantic segmentation, we use PartNet [81] and the part labels provided in [18].

Methods We compare the proposed approach with several state-of-the-art approaches using different shape representations. Regarding primitive-based reconstruction approaches, we compare the proposed method with BSP-Net [19], CvxNet [26], and SIF [33] (implicit-representation), and with AtlasNetV2 [28] (explicit representation). As the approaches in [19, 26] represent shapes as collections of convex shapes, we regard them as a baseline for the effectiveness of the star-domain primitive representation. Regarding non-primitive-based reconstruction approaches, we compare the proposed method with OccNet [77], which is the leading implicit representation technique, and with AtlasNet [35] (explicit shape representation). Concerning AtlasNetV2, as the code provided by the author does not include a model for single-view reconstruction, we replace the provided encoder with the same ResNet18 used by NSDN and OccNet, and train the model. Furthermore, for a fair comparison with NSDN, we sample 400 points from each patch during training, and use 30 patches for AtlasNetV2, unless otherwise noted. We confirm that this leads to a slightly better reconstruction accuracy than the original configuration. For BSP-Net, we use the pretrained model described in Section 6.4.2. In Section 6.4.3, we train BSP-Net using the code provided in [19]. As BSP-Net uses different train and test splits, we evaluate it on the intersection of the test splits from [77] and [19].

		airplane	bench	cabinet	car	chair	display	lamp	speaker	rifle	sofa	table	phone	vessel	mean	time
F-score	AtlasNet [35]	67.24	54.50	46.43	51.51	38.89	**42.79**	33.04	35.75	**64.22**	43.46	44.93	58.85	**49.87**	48.57	0.008
	AtlasNetV2 [28]	54.99	50.67	31.95	39.73	29.10	33.55	28.35	22.54	62.27	30.15	45.93	51.45	39.91	40.05	0.010
	OccNet [77]	62.87	56.91	61.79	56.91	42.41	38.96	38.35	42.48	56.52	48.62	58.49	66.09	42.37	51.75	0.525
	OccNet* [77]	63.56	57.39	**63.03**	61.41	**43.61**	41.54	**41.13**	**45.39**	57.94	**49.86**	**59.62**	66.11	45.00	53.51	0.529
	SIF [33]	52.81	37.31	31.68	37.66	26.90	27.22	20.59	22.42	53.20	30.94	30.78	45.61	36.04	34.86	n/a
	CvxNet [26]	**68.16**	54.64	46.09	47.33	38.49	40.69	31.41	29.45	63.74	42.11	48.10	59.64	45.88	47.36	n/a
	BSP-Net [19]	61.91	53.12	44.75	55.24	38.57	35.68	29.98	34.04	57.28	43.89	46.42	49.18	42.76	45.60	0.014
	NSDN (ours)	67.96	**60.37**	59.26	**63.54**	43.58	41.81	38.83	43.09	63.31	48.97	57.91	**70.65**	46.49	**54.29**	0.014
CD1	AtlasNet [35]	0.104	0.138	0.175	0.141	0.209	**0.198**	**0.305**	**0.245**	0.115	0.177	0.190	0.128	**0.151**	0.175	0.008
	AtlasNetV2 [28]	0.119	0.164	0.246	0.176	0.256	0.209	0.313	0.340	**0.099**	0.210	0.221	0.131	0.159	0.203	0.010
	OccNet [77]	0.147	0.155	0.167	0.159	0.228	0.278	0.479	0.300	0.141	0.194	0.189	0.140	0.218	0.215	0.525
	OccNet* [77]	0.141	0.154	**0.149**	0.150	0.206	0.214	0.369	0.254	0.142	0.182	0.175	0.124	0.194	0.189	0.529
	SIF [33]	0.167	0.261	0.233	0.161	0.380	0.401	1.096	0.554	0.193	0.272	0.454	0.159	0.208	0.349	n/a
	CvxNet [26]	**0.093**	**0.133**	0.160	**0.103**	0.337	0.223	0.795	0.462	0.106	**0.164**	0.358	**0.083**	0.173	0.245	n/a
	BSP-Net [19]	0.128	0.158	0.179	0.153	0.211	0.224	0.332	0.269	0.126	0.190	0.190	0.153	0.189	0.192	0.014
	NSDN (ours)	0.111	0.135	0.155	0.136	**0.191**	0.205	0.320	0.251	0.118	0.177	**0.167**	0.110	0.174	**0.173**	0.014
IoU	OccNet [77]	0.571	0.485	0.733	0.737	0.501	0.471	0.371	0.647	0.474	0.680	0.506	0.720	0.530	0.571	0.525
	OccNet* [77]	0.591	**0.492**	**0.750**	**0.746**	**0.530**	0.518	**0.400**	**0.677**	0.480	**0.693**	**0.542**	0.746	0.547	**0.593**	0.529
	SIF [33]	0.530	0.333	0.648	0.657	0.389	0.491	0.260	0.577	0.463	0.606	0.372	0.658	0.502	0.499	n/a
	CvxNet [26]	0.598	0.461	0.709	0.675	0.491	**0.576**	0.311	0.620	0.515	0.677	0.473	0.719	**0.552**	0.567	n/a
	BSP-Net [19]	0.549	0.371	0.660	0.708	0.466	0.507	0.323	0.638	0.462	0.667	0.428	0.711	0.523	0.539	0.014
	NSDN (ours)	**0.613**	0.461	0.719	0.742	0.515	0.553	0.368	0.667	**0.516**	0.689	0.511	**0.760**	0.550	0.589	0.014

Table 6.2 Reconstruction performance on ShapeNet [15]. In the far right column (labeled as "time"), the per object average duration (in seconds) of mesh sampling is provided to indicate the time cost for producing an evaluated mesh. In contrast to the original implementation of OccNet [77], no data augmentation is performed. Accordingly, we also report the results of pretrained OccNet trained without data augmentation, denoted as OccNet*.

Metrics We evaluate the proposed methods in terms of reconstruction accuracy, part correspondence, and mesh sampling speed. To evaluate the reconstruction accuracy, we apply three commonly used metrics to compute the difference between the reconstruction meshes and the ground truth: (1) F-score, which, by the argument in [122], can be interpreted as the percentage of correctly reconstructed surfaces, (2) L1 chamfer distance (CD1), and (3) volumetric IoU (IoU). For all metrics, we use 100,000 sample points from the ground-truth meshes, and reconstruct shape meshes by following [77, 26]. To evaluate the part correspondence in semantic capability, we use the standard label IoU between the ground-truth part label and the predicted part label. Regarding mesh sampling speed, we measure the time in which a pipeline encodes an image and decodes mesh vertices and faces. We exclude the time for the device I/O. All speed measurements are performed on an NVIDIA V100 GPU. Moreover, for a fair comparison, we measure the time to mesh a single primitive for AtlasNet, AtlasNetV2, and BSP-Net analogously with parallel processing, because their original implementation sequentially processes each primitive for meshing, whereas ours does meshing is parallel.

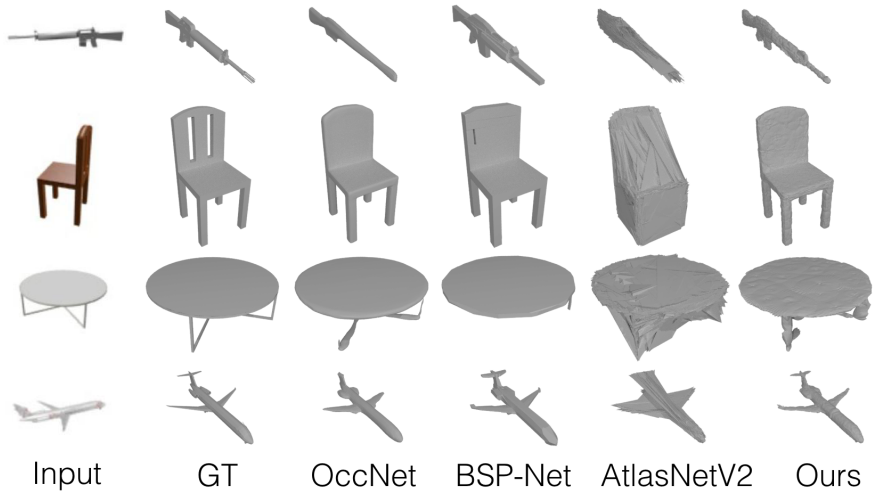


Figure 6.3 Visualization of reconstructed meshes with an RGB image input. Best viewed when zoomed in.

6.4.1 Visualization of differentiable shape and surface representations

NSD provides multiple *differentiable* shape and surface representations that are available both during training and inference: mesh, surface points, normal, indicator function (signed distance function), and texture. We show the qualitative examples in Figure 6.4.

6.4.2 Single view reconstruction

We evaluate the reconstruction performance of an NSD compared with state-of-the-art methods for an input RGB image. The quantitative results are shown in Table 6.2. Qualitative examples are shown in Figure 6.3. The number of faces of the meshes generated by NSDN and AtlasNetV2 are comparable with those by OccNet [77]. We arrive at the following conclusions: (1) NSDN consistently outperforms previous primitive-based approaches (CvxNet, SIF, BSP-Net, and AtlasNetV2) in terms of the averages of all metrics. In particular, significant improvement is observed in the F-score. (2) NSDN is relatively more effective than the leading technique (OccNet [77]), as indicated by CD1 and the F-score. It should be noted that the proposed method is comparable with OccNet, but the mesh sampling speed is distinctively faster. Details on the mesh sampling analysis can be found in Subsection 6.4.4.

Effect of losses As the surface point loss is made available by using integrated implicit and explicit representations, we evaluate the effectiveness of the proposed loss to demonstrate the advantage of the proposed representation. We use $N = 10$ for faster NSDN training

	implicit	explicit	F-score
AtlasNetV2 [28]		✓	40.05
BSP-Net [19]	✓		45.60
NSDN_O	✓		23.93
NSDN_C		✓	45.84
NSDN_S	✓	✓	50.52
NSDN_{S+O}	✓	✓	**52.27**

Table 6.3 Effects of different losses on the F-score. Check marks under the implicit and explicit columns indicate whether the loss uses the corresponding shape representation. In NSDN, O, C, and S indicate that only the occupancy loss, chamfer loss without surface point extraction, and surface point loss, respectively, are used.

to accelerate the experiments. The results are shown in Table 6.3. Using only occupancy loss leads to unsuccessful training. Using the standard chamfer loss leads to performance comparable with that of previous methods. Using surface point loss outperforms leading primitive-based techniques [19]. Additionally, using occupancy loss along with surface point loss leads to slightly higher accuracy and achieves the best results.

Analysis on expressive power of primitive shapes We quantitatively evaluate the expressive power of NSD compared with other primitives in previous studies: convexes [19] and superquadrics [94]. We evaluate the expressive power by measuring the complexity of the inferred primitive shapes. To quantify the complexity of the shape, we evaluate the discrete Gaussian curvature [24]. We use the airplane and the chair categories from ShapeNet [15] in this evaluation. For NSD, we use $N = 10$ for the number of primitives. The mean and standard deviation of the curvature measure are shown in Table 6.4. A larger mean value indicates that primitive shapes have more complex surfaces in terms of unevenness, and a larger standard deviation indicates that primitives have more diverse shapes. It can be seen that NSD has larger mean and standard deviation than the methods in previous studies. This quantitatively demonstates that NSD has more expressive power, as it learns more complex and diverse primitive shapes. Randomly sampled primitives from the airplane and chair categories are shown in Figure 6.5.

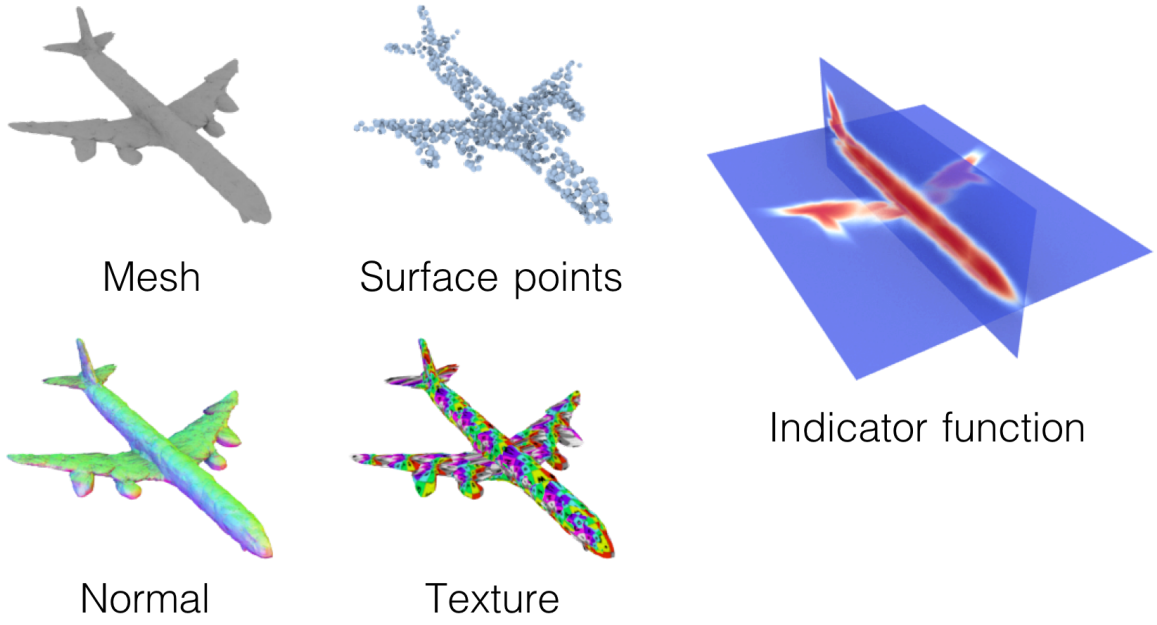


Figure 6.4 Differentiable shape and surface representations of NSD.

	mean	std
Superquadrics [94]	0.042	0.030
BSP-Net (convex) [19]	0.070	0.344
Proposed (star domain)	**0.154**	**0.351**

Table 6.4 Mean and standard deviation of discrete Gaussian curvature [24].

6.4.3 Semantic capability

We evaluate the semantic capability of the proposed approach compared with other approaches based on implicit and explicit primitive representations: BSP-Net [19] and AtlasNetV2 [28]. Following the evaluation methods in [26, 19, 94], involving varying numbers of primitives for each method, we evaluate the semantic capability of the approaches as a tradeoff between representation parsimony and semantic segmentation accuracy on part labels and reconstruction accuracy measured by the F-score. For the semantic segmentation task, labels for each ground truth point are predicted as follows: (1) For each ground truth point in a training sample, we determine the nearest primitive and vote for the part label of the point, (2) we assign each primitive a part label with the highest number of votes, and (3) for each point of a test sample, we determine the nearest primitive and assign the part label of the primitive to that point. We use four classes for semantic segmentation: plane, chair, table, and lamp. For table and lamp, we follow [19] to reduce the parts from (base,

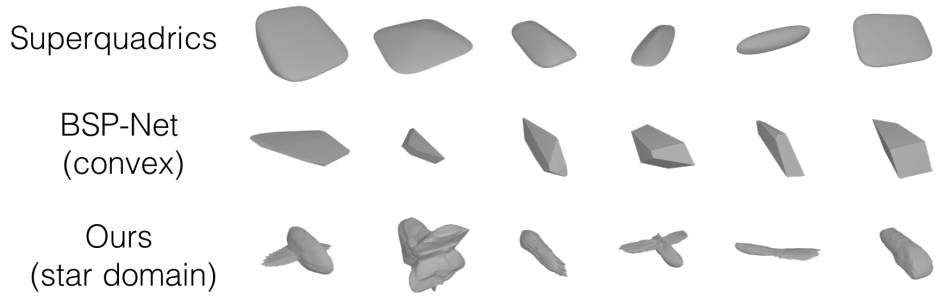


Figure 6.5 Randomly sampled primitives: superquadrics [94], convex [19], and proposed (star domain).

pole, lampshade, canopy) $\rightarrow$ (base, pole, lampshade), and analogously for table (top, leg, support) $\rightarrow$ (top, leg). The models are trained without part label supervision.

In Figure 6.10, it is seen that the proposed method consistently outperforms previous methods in terms of reconstruction accuracy regardless of the number of primitives, whereas it performs comparably in the semantic segmentation task. This demonstrates its superior semantic capability. It is comparable with the method in [19] in consistent part correspondence, but it better reconstructs target shapes. The learned primitives are shown in Figure 6.8, where it can be seen that the proposed approach is more parsimonious in reconstructing corresponding parts. We provide an additional visualization of the primitives of ours in Figures 6.6 for the plane, rifle, and chair categories from ShapeNet [15], respectively.

Effect of overlap regularization The high expressivity of NSD results in severe primitive overlap, leading to less interpretable part correspondence. To alleviate this, we investigate the effect of overlap regularization. As NSD is also an implicit representation, we adapt the decomposition loss proposed in [26] as an off-the-shelf overlap regularizer. We note that we use the L1 norm instead of the L2 norm in our formulation:

$$L_{\text{decomp}}(\Theta) = \mathbb{E}_{\mathbf{x}\sim\mathbb{R}^3}|\text{ReLU}(\sum_i \hat{O}_i(\mathbf{x};\mathbf{t}_i) - \tau_r)|, \tag{6.16}$$

where τ_r is a hyperparameter that controls the amount of overlap. The effect of overlap regularization is shown in Figure 6.9. In the visualization, there is less overlap between primitives with the regularization. A quantitative evaluation is shown in Table 6.6. In the table, we define an overlap metric (termed "overlap"), which counts the number of 3D points inside more than one primitive as follows:

$$\text{Overlap} = \mathbb{E}_{\mathbf{x}\sim\mathbb{R}^3}\mathbb{1}(\sum_i \mathbb{1}(\hat{O}_i(\mathbf{x};\mathbf{t}_i) \geq \tau_s) > 1). \tag{6.17}$$

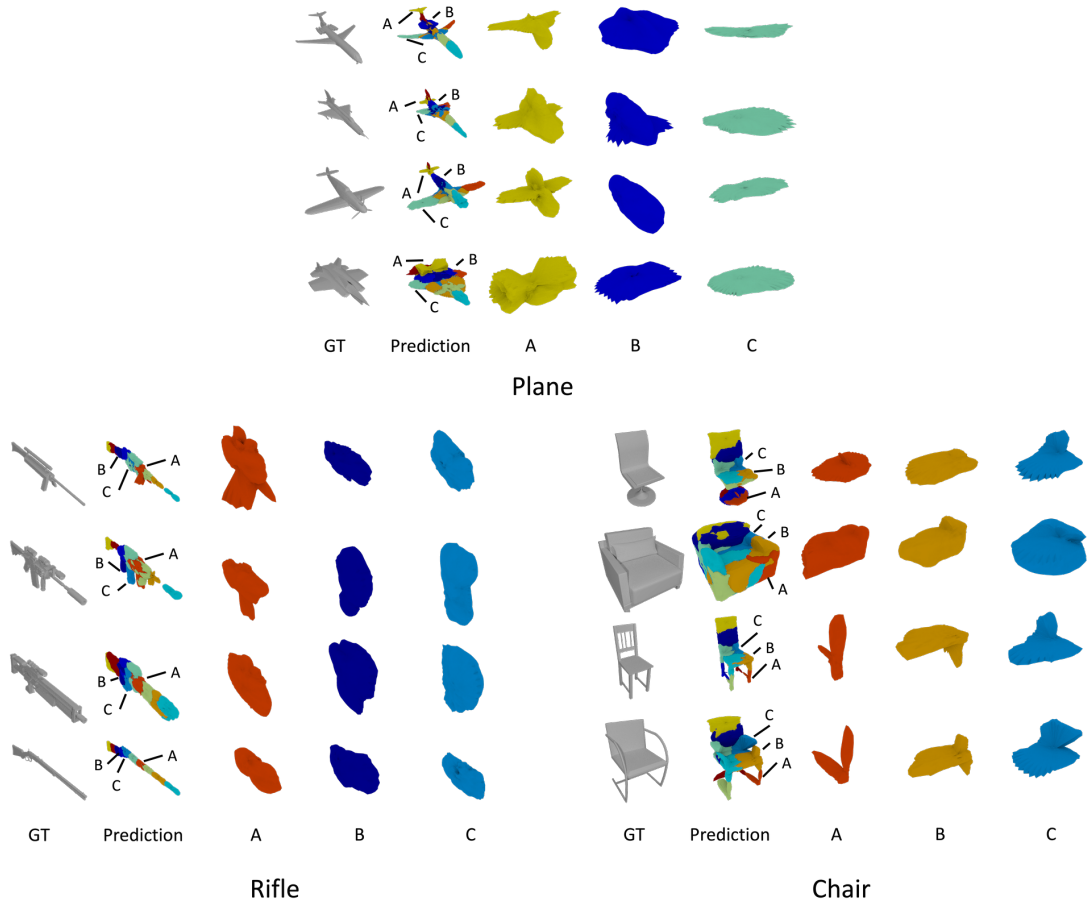


Figure 6.6 The additional visualization of the NSD primitives for plane, rifle and, chair categories from ShapeNet [15]

$\mathbb{1}$ is an indicator function. We set the loss weight of the regularizer to 10. In this experiment, we train the model for the airplane and chair categories. As the optimal τ_r varies across categories, we train the model with a single category. We use 1 and 1.2 for τ_r in the airplane and chair categories, respectively.

Applying the overlap regularizer clearly reduces the overlap, with a slight change in the F-score, and it improves the part IoU for both categories. In particular, the part IoU for the chair category significantly increases by 8%.

It should be noted that planar mesh patches as primitives [35, 28, 5] also have high expressivity and suffer from the same overlapping problems as NSD. Existing overlap regularization for this type of primitives, however, requires computationally expensive Jacobian computation [5]. Moreover, it is an indirect overlap regularization. We demonstrate that by simultaneously being highly expressive and an implicit representation, NSD allows for a computationally simpler and more direct approach to overcoming this shortcoming.

Unsupervised	89.01%
Few-shot w/ 2 ref.	87.28%
Few-shot w/ 8 ref.	88.15%

Table 6.5 Label IoU evaluation of the few-shot setting.

Figure 6.7 Qualitative comparison of unsupervised and few-shot setting.

Few-shot setting We investigate the effect of using a few reference samples with part labels in a few-shot setting. We train the model for the airplane category. To train the model without overfitting, we follow the few-shot training strategy of [18]; we first train the model for 3000 iterations only with reference samples using part labels and then finetune the model on the whole training set without part labels. We also employ the overlap regularization as discussed above. We show the quantitative result in Table 6.5. Surprisingly, using a few reference samples slightly degrades the segmentation performance. We attribute this reason to potential excessive optimization of the decomposition to the reference samples' part structure in the early stage of training, which does not lead to consistent decomposition for other samples through the successive training. Using more reference samples may alleviate this problem, as shown in the performance increase with more reference samples in the few-shot setting. We also show the qualitative result in Figure 6.7. Although there is a slight quantitative degradation, the qualitative results of the few-shot models show more evenly spread part decomposition with less overlap compared to the fully unsupervised result, leading to a visually more preferable result. We observed faster convergence to more evenly spaced primitive locations under the few-shot setting during the initial training. This observation attributes primitive shapes tend to have less overlap and become more evenly sized during the rest of the training. Balancing quantitative and qualitative performance of the decomposition result is left for future work.

Semantic part In Figure 6.1, it can be seen that a single NSD primitive (in cyan color) reconstructs the empennage. Moreover, in Figure 6.8, the wings (colored in green) and

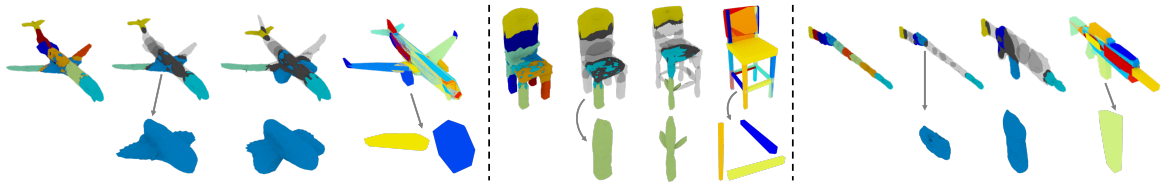


Figure 6.8 Primitives of different categories (plane, chair, and rifle). In each category, from far left: (1) reconstruction results with colored primitives, (2) top: only a few primitives are colored to indicate part correspondence with another reconstruction result on the right. Bottom: one primitive is selected and zoomed. (3) Top: another reconstruction result in the same category. Bottom: Same primitive as in the previous visualization. (4) Top: Reconstruction result of the same object with previous reconstruction by BSP-Net. Bottom: Manually selected primitives that correspond to the same semantic parts of the previous primitives. Best viewed zoomed in color.

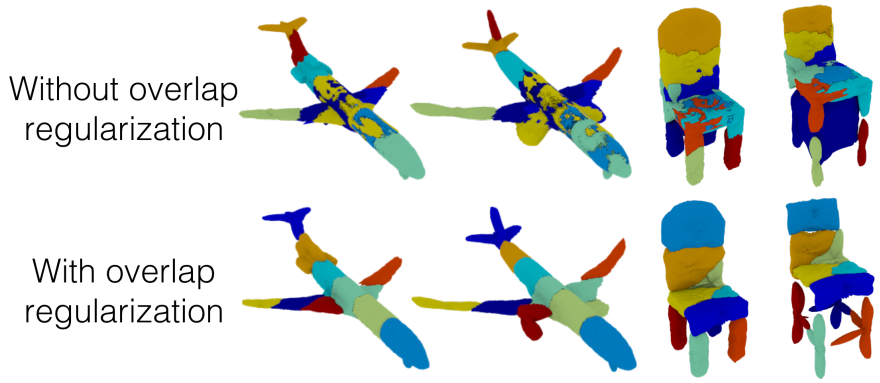


Figure 6.9 Effect of overlap regularization on primitive decomposition for the airplane and chair categories.

fuselage (colored in blue) are each reconstructed with nacelles by a single primitive. Thus, NSD can reconstruct complex shapes so that multiple parts under the same semantic part are reconstructed by one primitive. This demonstrates the expressive power of NSD in reconstructing semantic parts.

6.4.4 Mesh sampling

As the proposed method can represent surfaces in explicit forms using mesh templates, it can sample meshes significantly faster than time-consuming isosurface extraction methods. To demonstrate this, we evaluate the meshing speed and reconstruction accuracy of the proposed explicit representation compared with an implicit representation using the leading isosurface extraction method MISE [77]. For comparison, we use the same NSDN model for both

	airplane		
	Overlap	F-score	Label IoU
w/o reg.	5.810	69.55	48.15
w/ reg.	**0.445**	**69.92**	**50.90**
	chair		
	Overlap	F-score	Label IoU
w/o reg.	51.21	**35.56**	56.12
w/ reg.	**1.16**	33.92	**64.37**

Table 6.6 Effects of overlap regularization. The overlap score is scaled by a value of 1000 from the original value.

representations. The results are shown in Table 6.7. NSD can sample meshes significantly faster than MISE with comparable F-scores (see NSDN ico#2 and MISE up#1). We also investigate the number of vertices and faces on the surface over mesh sampling speeds. The proposed method can produce higher-resolution meshes significantly faster than MISE. We also use the mesh sampling speed of BSP-Net [19] as a reference for implicit representation approaches with fast mesh sampling. The proposed method is comparable with that in [19]. It should be noted that we use the result of BSP-Net only in relation to meshing speed and quality, as this method focuses on low polymesh.

6.5 Conclusion

In this study, we proposed NSD as a novel primitive representation. We demonstrated that the proposed method consistently outperforms previous primitive-based approaches and that it is the only primitive-based approach performing better than the leading reconstruction technique (OccNet [77]) in a single-view reconstruction task. Moreover, it has significantly better semantic capability. In future work, we would like to integrate texture reconstruction to extend the proposed primitive-based approach to more semantic part reconstruction.

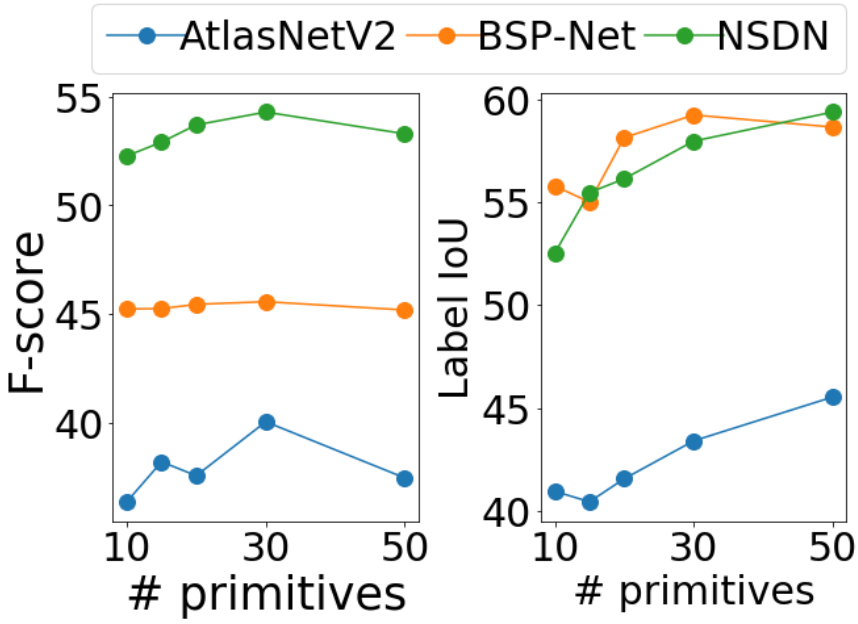


Figure 6.10 F-score and label IoU with varying number of primitives. The number of evaluated primitives is: 10, 15, 20, 30, and 50.

	#V	#F	F-score	time
NSDN ico0	2	5	34.02	0.012
NSDN ico2	30	88	42.87	0.013
NSDN ico4	478	1414	55.66	0.017
MISE up0	12	31	26.46	0.051
MISE up1	54	143	40.37	0.635
MISE up2	220	592	50.28	5.438
BSP-Net [19]	10	18	45.60	0.014

Table 6.7 Mesh sampling speed for given mesh properties. *#V* and *#F* denote the number of mesh vertices ($\times 100$) and mesh faces ($\times 100$), respectively. Ico# denotes the number of icosphere subdivisions used as the mesh template of the primitive. Up# denotes the number of upsampling steps in MISE [77]. Up0 is equal to 32^3 voxel sampling, and up2 to 128^3.

Chapter 7

Unified Pipeline for Comprehensive Understanding of Man-made Articulated Objects

7.1 Introduction

In previous chapters, we have demonstrated the proposed methods for articulated objects of diverse shapes through supervised learning (Chapter 4), for reducing 3D annotation data at the part level when structures can be assumed under unsupervised learning (Chapter 5), and for decomposing shapes under unsupervised learning (Chapter 6) for finer semantic shapes. However, in the reconstruction of articulated objects, supervised learning alone requires part-level supervision for all training data. Furthermore, unsupervised learning can reduce part-level 3D annotation when a structure can be assumed, but it is not applicable when a structure is not consistent. In addition, for applications such as planning the opening and closing of doors, it is necessary to recognize not only a part-level understanding but also more detailed shapes such as handles attached to the parts, which are suitable for grasping. However, a consistent part decomposition requires normalized part shapes in terms of part-level pose and size.

To compensate for the problems of these various methods, we propose an integrated pipeline that reduces the necessary annotation while accommodating the structure of diverse articulated objects and recognizing finer shapes of part shapes. In the unsupervised method of shape reconstruction and pose estimation of articulated objects in Chapter 6, we considered only the canonicalized shapes in terms of global pose and size as input and output, which is not applicable as is for input with background, when targeting multiple articulated objects,

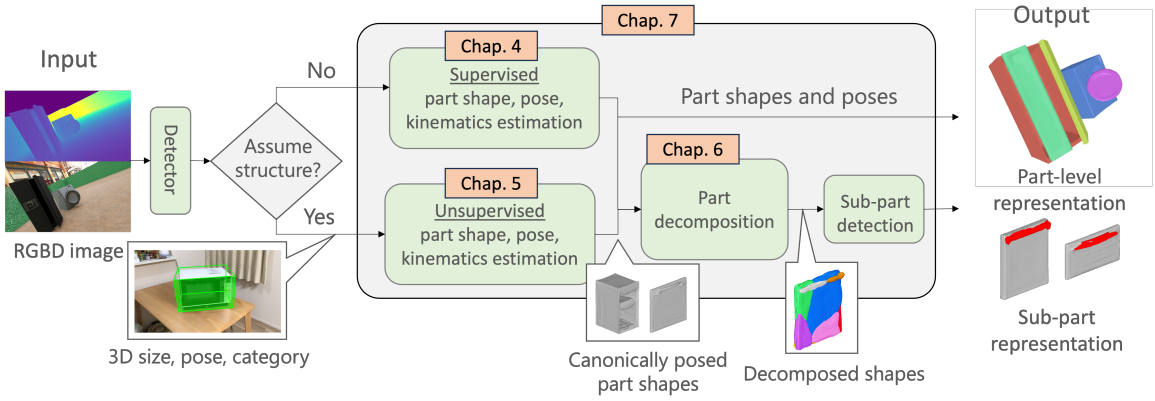


Figure 7.1 Visualization of the unified pipeline.

and when the output is in the camera coordinate space. In addition, the shape decomposition in Chapter 5 has the problem that when the target sub-part level shape is too detailed for the whole shape, it is difficult to decompose in an unsupervised manner, and there are cases where the target sub-part to be recognized does not exist in the input shape requiring to identify the presence or absence of the target shape.

This chapter proposes a method of integrating the results of unsupervised shape reconstruction and pose estimation of articulated objects targeting backgrounds and multiple articulated objects into the output in the camera coordinate system, and as an example of sub-part level understanding, proposes a method of recognizing the handle shape attached to the rotating parts by unsupervised shape decomposition. Figure 7.1 shows the relationships of the components proposed in the previous chapters in the proposed pipeline. For categories that can assume a consistent structure, the unsupervised method (Chapter 5) is applied, and for cases where a single structure cannot be assumed, we apply the supervised method handling the variously structured articulated shapes (Chapter 4). Finally, shape decomposition is performed on the normalized part shapes output from both supervised and unsupervised methods (Chapter 6). Thus, this chapter proposes a pipeline as a whole that reduces the costly part-level annotation data while accommodating articulated objects with various structures and performing sub-part level shape understanding.

7.2 Method

In the following sections, we first discuss conditioning on supervised and unsupervised approaches in the pipeline in Section 7.2.1. Then, Section 7.2.2 presents the method of integrating the unsupervised method targeting articulated objects (Chapter 5) into the pipeline,

Washing machine	Microwave	Storage	Table
17	16	346	101

Table 7.1 Number of CAD models in [133] dataset.

and Section 7.2.3 shows the method for decomposition and segmentation of the sub-part shapes of part shapes.

7.2.1 Supervised and unsupervised conditioning in the pipeline

One of the main limitations of the supervised approach discussed in Chapter 4 is that the method requires part-level 3D annotation for training. However, such annotation is costly to produce for real-world articulated objects, especially with many parts often seen in storage categories like drawers. Thus, synthetic data with automatically generated part-level annotations is widely used in previous works [63, 38, 83, 48, 32] and also in the proposed method in Chapter 4. However, curating synthetic data for complex object shape and texture is also costly [67] as it requires professional skills to create CAD models and is time-consuming to create detailed textures and shapes for realistic data. For example, the SAPIEN [133] dataset, as the most widely used synthetic dataset, only contains CAD models of the categories with more detailed texture and shapes, such as washing machine and microwave categories, ten times smaller than the categories with simpler texture and repetitive simple part shapes, such as storage and table categories. We show the number of CAD models in the SAPIEN [133] dataset in Table 7.1 and their visualization in Figure 7.2. In contrast, scanning of an object with a short video clip can recover detailed shape and texture in a few minutes with modern multiview reconstruction pipeline [110, 111, 84, 3] without special skills. Thus, learning part-level understanding from unannotated data, such as whole shape scans, is an appealing approach. Moreover, we observe that the categories with a smaller number of samples in the synthetic dataset, such as microwave and washing machine as discussed above, have relatively uniform and consistent part structures within a category due to a smaller number of parts compared to other categories like storage and refrigerator with more part counts. Thus, in the pipeline, we demonstrate the application of the unsupervised approach to categories with such consistent part structure within a category to show the complementary effectiveness of the unsupervised approach to the supervised approach.

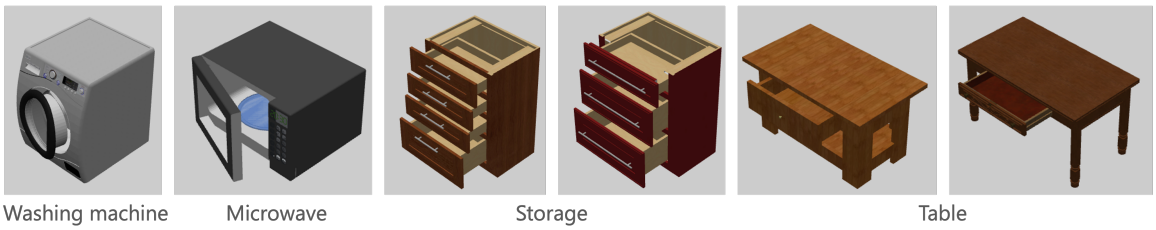


Figure 7.2 Visualization of CAD models from [133] dataset.

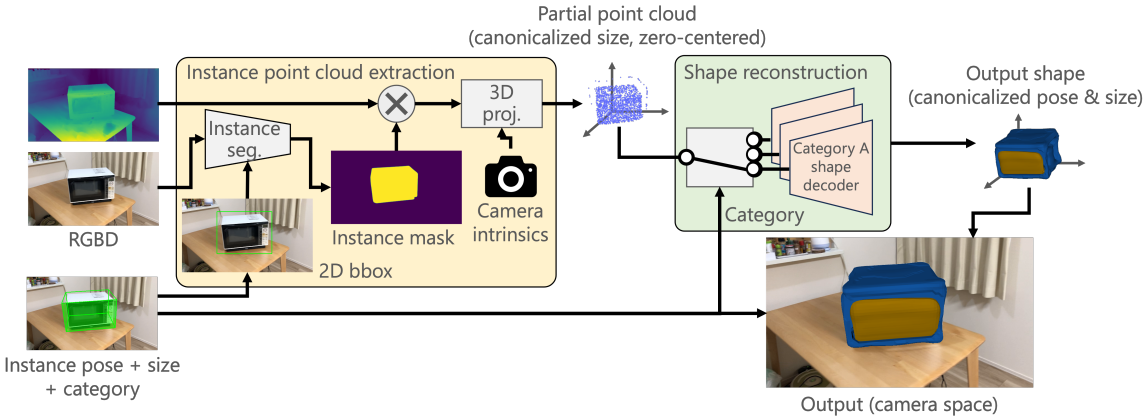


Figure 7.3 Visualization of instance point cloud extraction and camera space projection for the unsupervised approach.

7.2.2 Instance point cloud extraction and camera space projection

We first extract the foreground point cloud of the target instance by the instance point cloud extraction module shown as the "instance point cloud extraction" module in Figure 7.1. We show the module architecture in the yellow box in Figure 7.3. The module takes the RGBD image, target instance 6D pose, and size detected by the detector and outputs the partial point cloud of the target instance. The 3D cuboid represented by 6D pose and size is projected to the image plane to get 2D bounding boxes, which is used to prompt along with the input RGB image to extract the instance segmentation mask by using the instance segmentation module [56]. The depth map is segmented by the instance mask and projected to 3D space as a partial point cloud with the known camera intrinsics to form the input partial point cloud to the pose-aware part decomposition (PPD) module discussed in Chapter 5. To adapt the PPD module to the depth input, the partial point cloud encoder is trained by the distillation approach as discussed in Section 5.4.8. The output shape is then projected to the camera space by the estimated instance-level pose and size.

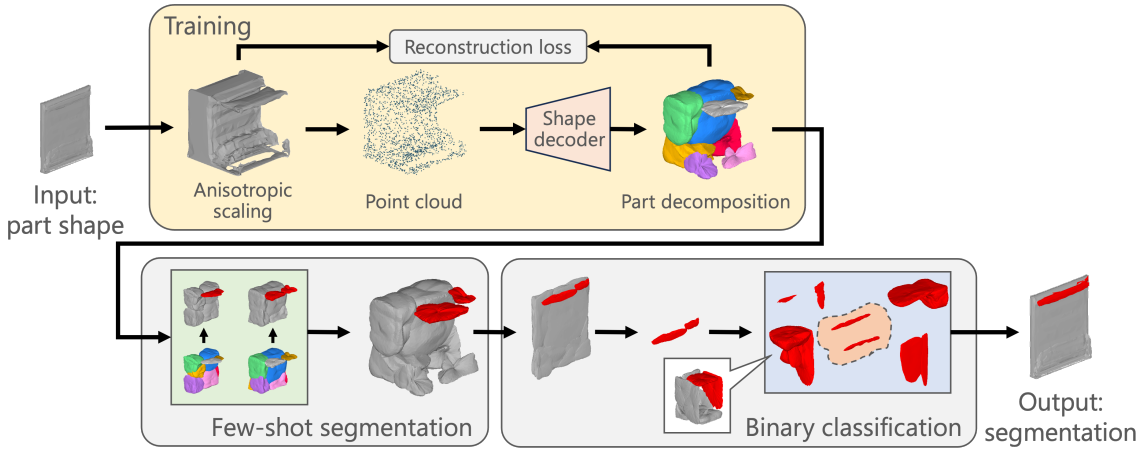


Figure 7.4 Visualization of the sub-part segmentation module.

7.2.3 Sub-part shape segmentation

We visualize the overview of the sub-part shape segmentation approach in Figure 7.4. The decomposed primitive shapes in the part decomposition approach tend to have similar sizes, as depicted in Figure 6.8. Also, the method assumes the same semantic shape is located approximately in a similar location in the canonicalized 3D space (Section 5.4.4) to induce unsupervised part decomposition. This indicates that variously posed small or thin shapes of a part shape are hard to segment out. To mitigate this issue when decomposing the part shapes, we first canonicalize the pose of the input part shape to reduce pose variation, then apply anisotropic scaling (Section 4.3.3) to normalize all side lengths of the part shape to one. This preprocessing enlarges the small parts for easier segmentation. After the part decomposition is learned, we manually annotate a few samples to identify the primitives corresponding to the target sub-part shape. We follow a similar step as discussed in Section 5.4.3. We treat the shapes reconstructed by the identified primitives as the target sub-part shape candidate. As discussed in Section 7.1, the identified primitives reconstruct the non-target sub-part shapes when the target sub-part is missing. To identify whether the primitives reconstructing the target sub-part, we train a simple binary classifier to detect the target sub-part. After classifying the target sub-part, we segment the input shape based on the reconstructed sub-part shape by primitive shapes. Again, we follow a similar step described in Section 5.4.3, for each input mesh's face center, measure the distance between the face and each primitive, and then assign the sub-part label to the face if the primitive with the shortest distance to the face belongs to the sub-part shape.

7.3 Experiment

7.3.1 Models

For the unsupervised approach, in this experiment, we target ovens with horizontal joint direction as a target category. We use the pretrained oven model introduced in Chapter 5. For sub-part segmentation, we train the neural star domain shape decoder from Chapter 6 on the output shapes of the supervised approach from Chapter 4.

7.3.2 Data

RGB images taken by a consumer smartphone and the corresponding depth maps generated from partial front views of the scene using an off-the-shelf neural radiance field model [120] are used in this experiment. Note that the input to the pipeline is the single RGB-D image. For the manual annotation of the primitives described in Section 7.2.3, we pick ten to twenty samples of the primitives reconstructing handle shapes as positive samples, and the same number of random, non-handle shapes as negative samples to build a binary classifier.

7.3.3 Reconstruction by the supervised approach

The qualitative results are shown in Figure 7.5. "Current pose" indicates the reconstruction of the target shape with the estimated pose in the input. "Fully opened" or "closed" indicates that the pose of the instance has been changed based on the estimated kinematic parameters. We can see that reasonable estimates based on the shape, posture, and kinematic parameters can be made for various objects with diverse poses. The results of simultaneously reconstructing two instances are shown in Figure 7.6. Our pipeline successfully performs the simultaneous reconstruction of multiple target instances.

7.3.4 Reconstruction by the integrated unsupervised approach

The results of the unsupervised method integrated in Section 7.2.2 are shown in Figure 7.7. The proposed method allows for the reconstruction in camera space, projected from the canonicalized space of the unsupervised model (Chapter 5). Figure 7.7 (c) shows the result combined with the supervised method (Chapter 4). We can see that consistent reconstruction is possible using a model trained without part-level annotation aligned with the results of the supervised method.

7.3.5 Sub-part segmentation

Figure 7.8 shows the results of segmentation on the sub-part shapes of the input part shape. It demonstrates that the same primitives consistently reconstruct the handle shape in two inputs and that the handle shape can be segmented from the input shape in an unsupervised manner. Combined with estimated joint parameters as visualized in Figure 7.8, the output of the pipeline can be transferred to downstream tasks such as grasping by a robot arm.

7.4 Limitation

While our method makes significant strides in the daily articulated object reconstruction task, it does have several limitations.

Unclear instance boundary Our system cannot handle cases where the instance boundary is not well defined, such as a door directly attached to a room. 3D reconstruction of room geometry with part-level shape reconstruction of articulated parts is an interesting future direction as a scene reconstruction task.

Category-specific setting of the unsupervised method for articulated objects The unsupervised approach requires a model for each category and different kinematic models by switching between multiple models as illustrated in Figure 7.9. Applying the unsupervised approaches to more categories needs to add more models, complicating the pipeline.

False negative reconstruction by sub-part decomposition The sub-part level shape segmentation by the part decomposition strongly depends on the input shape, thus it directly reflects any false positives or false negatives of the shapes in the reconstructed parts, as shown in Figure 7.10. Improving the shape reconstruction accuracies in the previous stages would alleviate this problem.

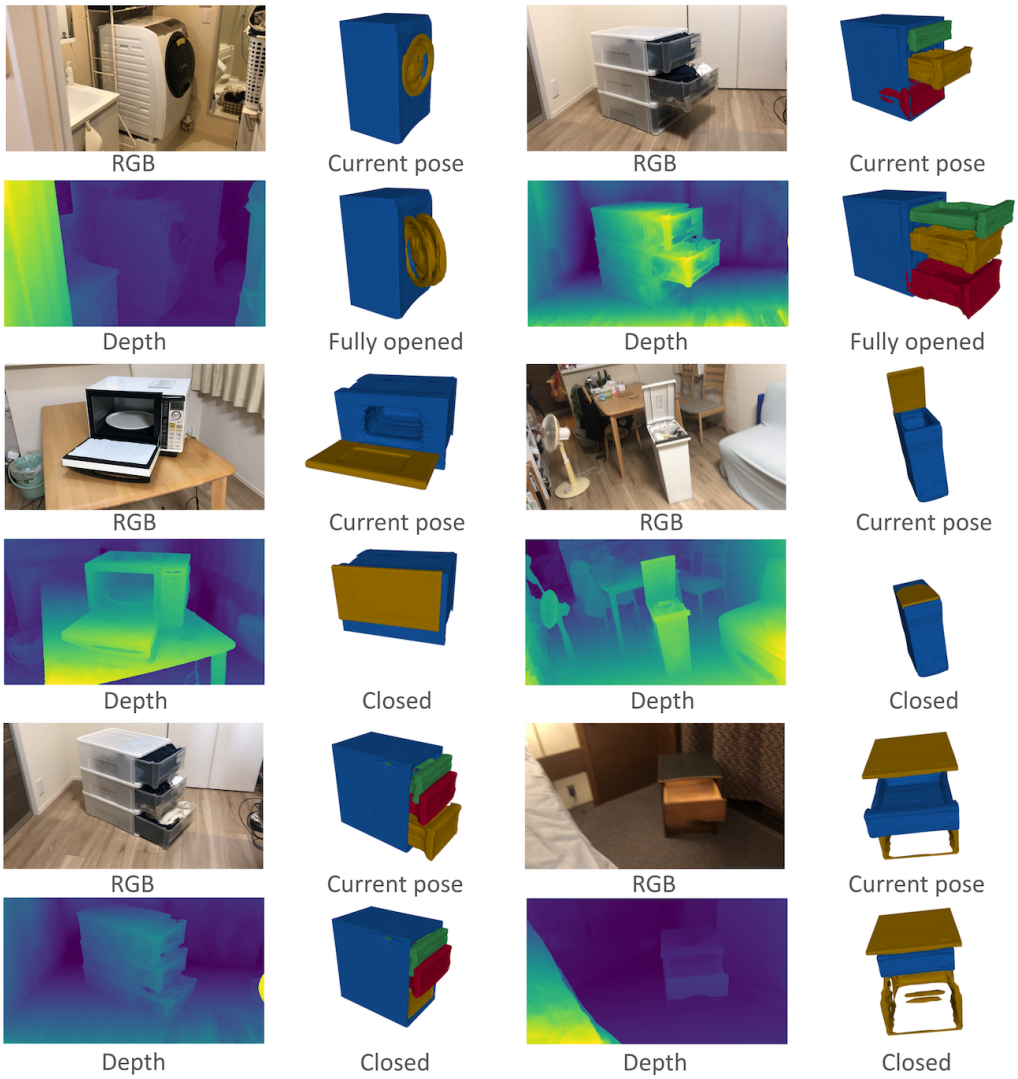


Figure 7.5 Visualization of the real-world result by the supervised approach (Chapter 4) taken by a commodity smartphone camera.

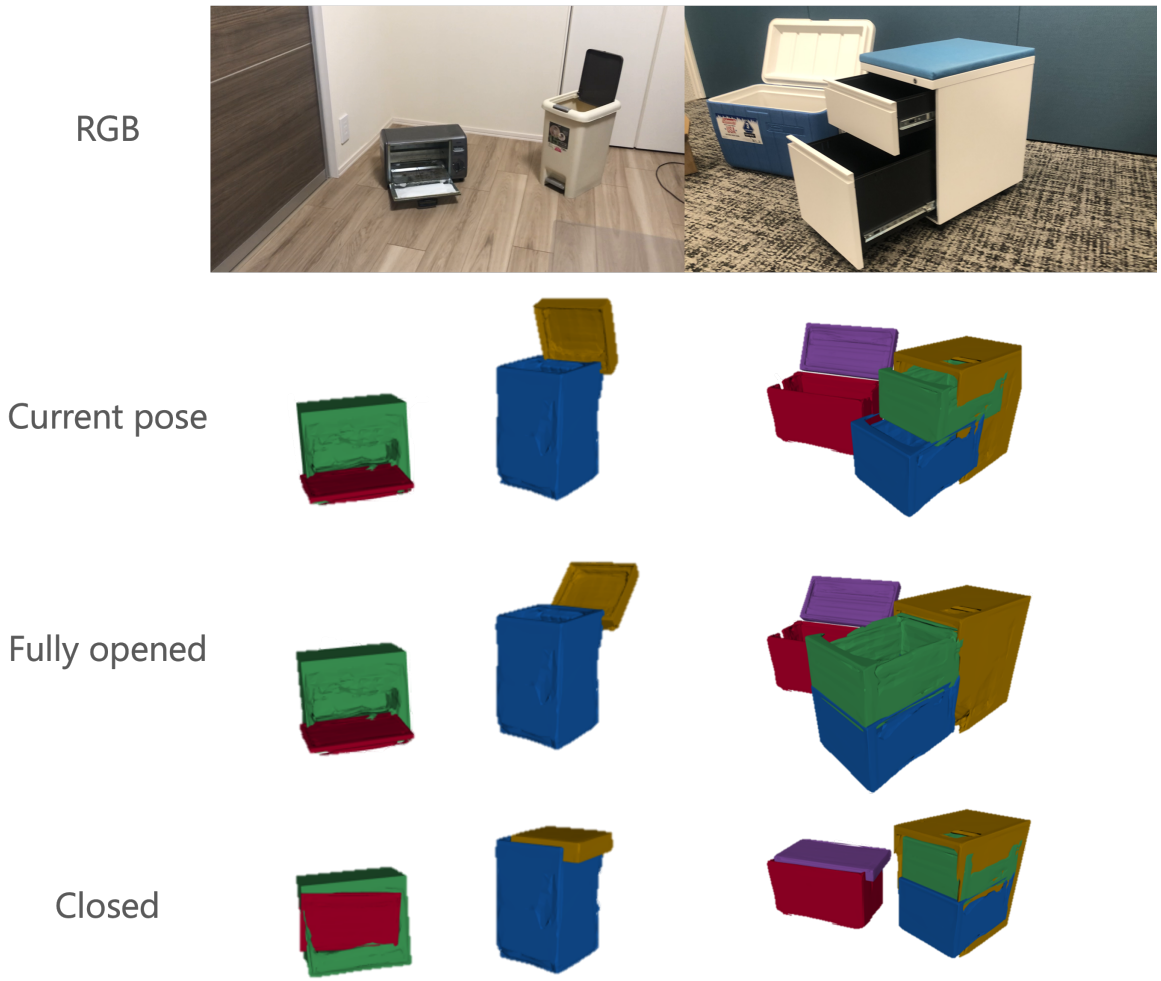


Figure 7.6 Visualization of reconstruction result by the supervised approach (Chapter 4) with multiple instances.

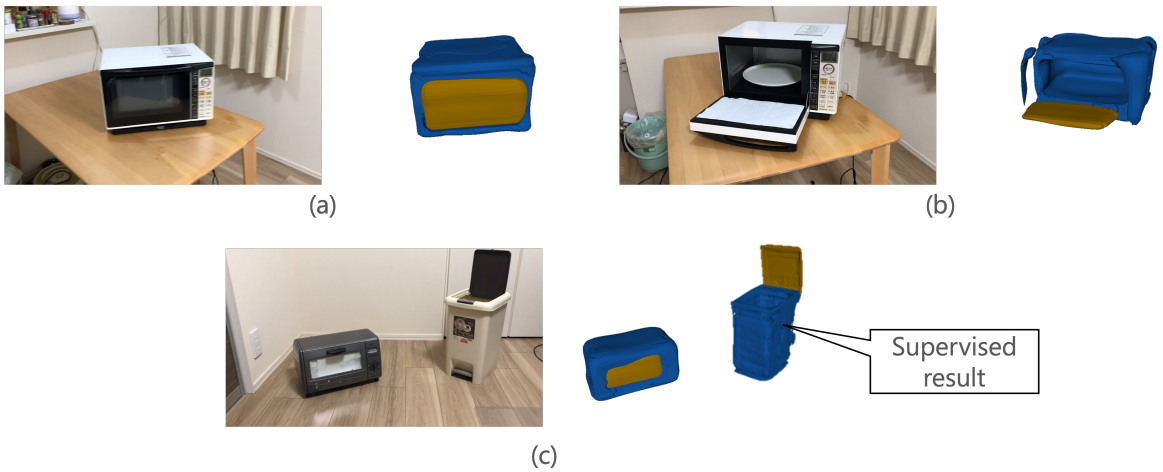


Figure 7.7 (a) and (b): Visualization of reconstruction result using the unsupervised approach (Chapter 5) (c): The unsupervised approach combined with the supervised approach (Chapter 4)

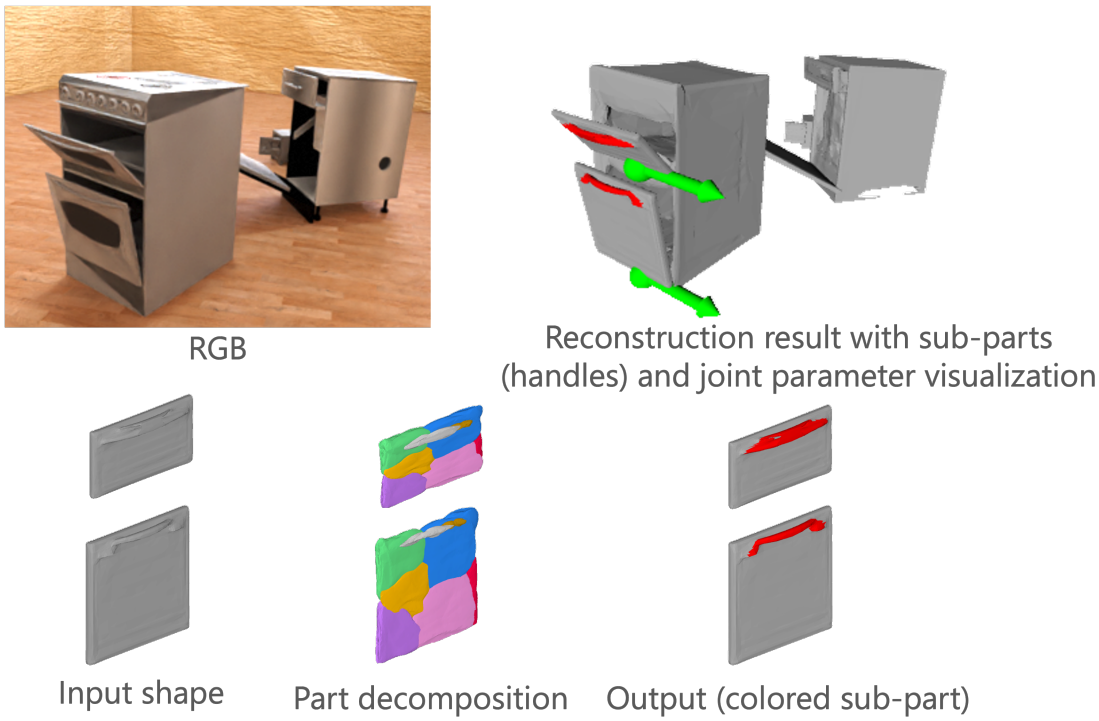


Figure 7.8 Visualization of the sub-part shape segmentation.

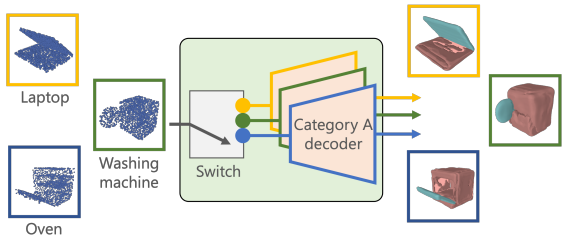


Figure 7.9 Limitation of the unsupervised approach.

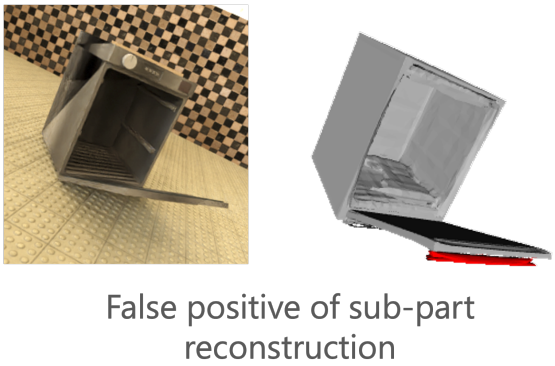


Figure 7.10 False positive reconstruction of handle shape by sub-part decomposition.

Chapter 8

Conclusion and Future Work

Understanding the shape, pose, kinematics, and finer semantic details of articulated objects at the part level from a single RGBD image has numerous applications in robotics, AR/VR. However, estimating these properties from a single image is a highly ill-posed and challenging problem. Furthermore, estimating these properties for articulated objects with diverse structures and varying numbers of parts is even more challenging. In this study, we propose a method to achieve the understanding of articulated objects' shape, pose, kinematics, and finer semantic details at the part level using both supervised and unsupervised approaches. Our method targets objects with diverse structures and varying numbers of parts. In this section, we summarize the proposed method and discuss its contributions.

In Chapter 4, we have presented the central idea of the thesis which is to develop a robust processing method for single-view input capable of handling the diverse attributes of articulated objects. The method provides a solution to issues identified in previous studies by offering a single-stage, end-to-end approach that extends from parts-level detection to instance reconstruction without assuming the part structures and counts. The proposed method has been designed to handle various articulated object attributes, with a particular emphasis on the unique challenges presented by small and thin parts. By focusing on the trajectory of movement of these small and detailed parts, we have been able to improve detection performance, reduce false positives, and enhance recognition accuracy. Importantly, the proposed approach also avoids explicitly learning part structures, instead detecting parts as individual shapes, which allows for a unified approach from detection to shape reconstruction.

In Chapter 5, we have introduced the idea of exploiting consistent part structure for unsupervised learning. In the real world, many everyday objects, such as eyeglasses, laptops, and scissors, exhibit consistent part structures. Our approach leverages this consistency, allowing for the learning of shapes, joint parameters, and poses of individual parts without

the need for manual annotations as opposed to the previous works which require part-level annotations.

In Chapter 6, we have proposed a novel primitive shape representation, the neural star domain. This approach helps to maintain high reconstruction accuracy while enabling the unsupervised decomposition of parts into finer semantic shapes. This novel shape representation, which generalizes the previous works' primitive representations, offers the ability to fully utilize the parameters of the neural network compared to previous shape representations.

Chapter 7 presents our unified pipeline, integrating the methods described in the previous chapters. This pipeline efficiently handles articulated objects with consistent part structures within the category, accommodating varying part counts and structures. Moreover, it showcases a hierarchical understanding of articulated objects, encompassing shape, pose, kinematics, and semantics at both the instance and subparts levels.

To summarize, this thesis presents three primary contributions: a method for supervised learning of part shapes and poses when unable to assume a consistent structure for each category, a method for reconstructing part shapes and estimating part pose without using annotations when assuming a consistent structure, and the development of the neural star domain for the reconstruction of finer-level semantic part shapes.

Finally, we list potential extensions beyond this study as future work:

Modeling other types of joints Although the proposed method covers a wide variety of daily man-made articulated objects, this study dealt with man-made articulated objects consisting only of revolute and prismatic joints. However, there are other types of joints, such as screw joints, ball joints, etc. Modeling each joint with different parameterization to express each constrained motion is not scalable. Therefore, an interesting direction is to explore the learnable, unified joint representation suitable in learning-based computer vision pipelines.

More complex mechanical linkage This study only deals with non-sequential joint configurations without loops, where the object consists of a single base part and all the articulated parts are attached to it. However, humans can reasonably understand and predict more complex system dynamics of unseen mechanical linkage consisting of joints with sequences, loops, constraints by other joints from perception. Extending the man-made articulated object understanding with such a challenging setting is an interesting direction.

Unified frame including natural articulated objects To the best of my knowledge, a universal framework for part-level shape and pose understanding explicitly targeting both man-made articulated objects and other types of common articulated objects, such as humans and animals in a single-view setting, has not been explored. The existing work relies on interaction or temporal information to identify kinematics and consisting parts. Without such information to understand 3D shape and kinematics requires a generic prior knowledge of the 4D world.

References

[1] Ben Abbatematteo, Stefanie Tellex, and George Konidaris. Learning to generalize kinematic models to novel objects. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2020.

[2] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[3] Ayush Baid, John Lambert, Travis Driver, Akshay Krishnan, Hayk Stepanyan, and Frank Dellaert. Distributed global structure-from-motion with a deep front-end. *arXiv preprint arXiv:2311.18801*, 2023.

[4] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356), 1992.

[5] Jan Bednarik, Shaifali Parashar, Erhan Gundogdu, and Pascal Salzmann, Mathieu and-Fua. Shape reconstruction by learning differentiable surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] Deniz Beker, Hiroharu Kato, Mihai Adrian Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Monocular differentiable rendering for self-supervised 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[7] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2), 1987.

[8] I Binford. Visual perception by computer. In *Proceedings of the IEEE Conference of Systems and Control*, 1971.

[9] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[10] Leonardo Bonanni, Chia-Hsun Lee, and Ted Selker. Counterintelligence: Augmented reality kitchen. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2005.

References

[11] Aljaž Božič, Pablo Palafox, Michael Zollhöfer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] Romain Brégier. Deep regression on manifolds: a 3D rotation case study. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2021.

[13] Charles E Burkhardt and Jacob J Leventhal. *Foundations of quantum physics*. 2008.

[14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[15] 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.

[16] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Stefan Stojanov, and James M Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[17] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. Deformable feature aggregation for dynamic multi-modal 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[18] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. Bae-net: branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[19] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[22] Vasek Chvatal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18(1), 1975.

[23] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[24] David Cohen-Steiner and Jean-Marie Morvan. Restricted delaunay triangulations and normal cycle. In *Proceedings of the Annual Symposium on Computational Geometry (SoCG)*, 2003.

[25] Jim Conant and Tim Michaels. On the number of tilings of a square by rectangles. *European Journal of Combinatorics*, 18, 2014.

[26] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[27] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[28] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[29] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins. Factored pose estimation of articulated objects using efficient nonparametric belief propagation. In *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[30] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38(6), 2019.

[31] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, 1997.

[32] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[33] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[34] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[35] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[36] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

References

[37] Nick Heppert, Toki Migimatsu, Brent Yi, Claire Chen, and Jeannette Bohg. Category-independent articulated object tracking with factor graphs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[38] Nick Heppert, Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Rares Andrei Ambrus, Jeannette Bohg, Abhinav Valada, and Thomas Kollar. Carto: Category and joint agnostic reconstruction of articulated objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[39] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2013.

[40] Berthold KP Horn. *Shape from shading: A method for obtaining the shape of a smooth opaque object from one view*. PhD thesis, Massachusetts Institute of Technology, 1970.

[41] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2), 1991.

[42] Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[43] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas J Guibas. Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[44] Katsushi Ikeuchi and Berthold KP Horn. Numerical shape from shading and occluding boundaries. *Artificial intelligence*, 17(1-3), 1981.

[45] Muhammad Zubair Irshad, Thomas Kollar, Michael Laskey, Kevin Stone, and Zsolt Kira. Centersnap: Single-shot multi-object 3d shape reconstruction and categorical 6d pose and size estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[46] Muhammad Zubair Irshad, Sergey Zakharov, Rares Ambrus, Thomas Kollar, Zsolt Kira, and Adrien Gaidon. Shapo: Implicit representations for multi object shape appearance and pose optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[47] Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[48] Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. OPD: Single-view 3D openable part detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[49] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[50] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[51] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[52] Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Neural star domain as primitive representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[53] Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Unsupervised pose-aware part decomposition for man-made articulated objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[54] J Mark Keil. Decomposing a polygon into simpler components. *SIAM Journal on Computing*, 14(4), 1985.

[55] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[56] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023.

[57] Marian Kleineberg, Matthias Fey, and Frank Weichert. Adversarial generation of continuous implicit shape representations. In *Eurographics*, 2020.

[58] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[59] Zoe Kourtzi and Nancy Kanwisher. Activation in human mt/mst by static images with implied motion. *Journal of Cognitive Neuroscience*, 12(1), 2000.

[60] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[61] David H Laidlaw, W Benjamin Trumbore, and John F Hughes. Constructive solid geometry for polyhedral objects. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1986.

[62] Hao Li, Zehan Zhang, Xian Zhao, Yulong Wang, Yuxi Shen, Shiliang Pu, and Hui Mao. Enhancing multi-modal features using local self-attention for 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[63] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[64] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[65] Tingting Liao, Xiaomei Zhang, Yuliang Xiu, Hongwei Yi, Xudong Liu, Guo-Jun Qi, Yong Zhang, Xuan Wang, Xiangyu Zhu, and Zhen Lei. High-fidelity clothed avatar reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[66] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[67] Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Qiaojun Yu, Yang Han, and Cewu Lu. Akb-48: a real-world articulated object knowledge base. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[68] Xinguo Liu, R Su, Sing Bing Kang, and Shum Heung-Yeung. Directional histogram model for three-dimensional shape similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[69] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6), 2015.

[70] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1987.

[71] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[72] Jun Lv, Qiaojun Yu, Lin Shao, Wenhai Liu, Wenqiang Xu, and Cewu Lu. Sagci-system: Towards sample-efficient, generalizable, compositional, and incremental robot learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[73] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *Proceedings of Robotics: Science and Systems (RSS)*, 2017.

[74] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[75] Roberto Martín-Martín and Oliver Brock. Building kinematic and dynamic models of articulated objects with multi-modal interactive perception. In *AAAI Spring Symposium Series*, 2017.

[76] Roberto Martín-Martín, Clemens Eppner, and Oliver Brock. The rbo dataset of articulated objects and interactions. *arXiv preprint arXiv:1806.06465*, 2018.

[77] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[78] Frank Michel, Alexander Krull, Eric Brachmann, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[79] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[80] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[81] 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 (CVPR)*, 2019.

[82] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J Mitra, and Leonidas J Guibas. Structedit: Learning structural shape variations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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

[84] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4), 2022.

[85] Shree K Nayar and Yasuo Nakagawa. Shape from focus. *IEEE Transactions on Pattern analysis and machine intelligence (T-PAMI)*, 16(8), 1994.

[86] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[87] Neil Nie, Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Structure from action: Learning interactions for articulated object 3d structure discovery. *arXiv preprint arXiv:2207.08997*, 2022.

[88] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[89] Yinyu Nie, Ji Hou, Xiaoguang Han, and Matthias Niessner. Rfd-net: Point scene understanding by semantic instance reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[90] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[91] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[92] Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch it move: Unsupervised discovery of 3D joints for re-posing of articulated objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[93] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[94] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[95] Despoina Paschalidou, Luc van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[96] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[97] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[98] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[99] Sudeep Pillai, Matthew R Walter, and Seth Teller. Learning articulated motions from visual demonstration. *Proceedings of Robotics: Science and Systems (RSS)*, 2014.

[100] Adrien Poulenard, Marie-Julie Rakotosaona, Yann Ponty, and Maks Ovsjanikov. Effective rotation-invariant point cnn with spherical harmonics kernels. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2019.

[101] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[102] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[103] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

[104] Nathan Reading. Generic rectangulations. *European Journal of Combinatorics*, 33(4), 2012.

[105] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[106] Dominic Roberts, Ara Danielyan, Hang Chu, Mani Golparvar-Fard, and David Forsyth. Lsd-structurenet: Modeling levels of structural detail in 3d part hierarchies. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[107] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[108] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[109] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[110] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[111] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[112] Nobu Shirai and Tomoko Imura. Implied motion perception from a still image in infancy. *Experimental Brain Research*, 232(10), 2014.

[113] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[114] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[115] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[116] Alan Slater, Victoria Morison, Carole Town, and David Rose. Movement perception and identity constancy in the new-born baby. *British Journal of Developmental Psychology*, 3(3), 1985.

[117] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107, 2021.

[118] Elizabeth S Spelke, Roberta Kestenbaum, Daniel J Simons, and Debra Wein. Spatiotemporal continuity, smoothness of motion and object identity in infancy. *British Journal of Developmental Psychology*, 13(2), 1995.

[119] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41, 2011.

[120] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, et al. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023.

[121] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. Point scene understanding via disentangled instance mesh reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[122] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[123] Wei-Cheng Tseng, Hung-Ju Liao, Lin Yen-Chen, and Min Sun. Cla-nerf: Category-level articulated neural radiance field. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[124] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[125] Michał J Tyszkiewicz, Kevis-Kokitsi Maninis, Stefan Popov, and Vittorio Ferrari. Raytran: 3d pose estimation and shape reconstruction of multiple objects from videos with ray-traced transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[126] Dmitriĭ Aleksandrovich Varshalovich, Anatolï Nikolaevitch Moskalev, and Valerii Kel'manovich Khersonskii. *Quantum theory of angular momentum*. 1988.

[127] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[128] 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 (CVPR)*, 2019.

[129] Jamie Watson, Oisin Mac Aodha, Daniyar Turmukhambetov, Gabriel J Brostow, and Michael Firman. Learning stereo from single images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[130] Fangyin Wei, Rohan Chabra, Lingni Ma, Christoph Lassner, Michael Zollhöfer, Szymon Rusinkiewicz, Chris Sweeney, Richard Newcombe, and Mira Slavcheva. Self-supervised neural articulated shape and appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

[132] Andrew P Witkin. Recovering surface shape and orientation from texture. *Artificial intelligence*, 17(1-3), 1981.

[133] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[134] Fei Xu and Susan Carey. Infants' metaphysics: The case of numerical identity. *Cognitive psychology*, 30(2), 1996.

[135] Zike Yan and Xuezhi Xiang. Scene flow estimation: A survey. *arXiv preprint arXiv:1612.02590*, 2016.

[136] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[137] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[138] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Lassie: Learning articulated shapes from sparse image ensemble via 3d part discovery. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[139] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Hi-lassie: High-fidelity articulated shape and skeleton discovery from sparse image ensemble. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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

[141] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[142] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[143] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild". In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

Publications

Reviewed Conference

1. Yuki Kawana, Yusuke Mukuta, Tatsuya Harada, “Neural Star Domain as Primitive Representation,” Neural Information Processing Systems (NeurIPS), 2020.

2. Yuki Kawana, Yusuke Mukuta, Tatsuya Harada, "Unsupervised Pose-aware Part Decomposition for Man-Made Articulated Objects," European Conference on Computer Vision (ECCV), 2022.

3. Yuki Kawana, Tatsuya Harada, "Detection Based Part-level Articulated Object Reconstruction from Single RGBD Image," Neural Information Processing Systems (NeurIPS), 2023.