Neural Implicit Representation for Building Digital Twins of Unknown Articulated Objects

Yijia Weng[1,2*] Bowen Wen[1] Jonathan Tremblay[1] Valts Blukis[1]
Dieter Fox[1] Leonidas Guibas[2] Stan Birchfield[1]

[1]NVIDIA [2]Stanford University

Abstract

We address the problem of building digital twins of unknown articulated objects from two RGBD scans of the object at different articulation states. We decompose the problem into two stages, each addressing distinct aspects. Our method first reconstructs object-level shape at each state, then recovers the underlying articulation model including part segmentation and joint articulations that associate the two states. By explicitly modeling point-level correspondences and exploiting cues from images, 3D reconstructions, and kinematics, our method yields more accurate and stable results compared to prior work. It also handles more than one movable part and does not rely on any object shape or structure priors. Project page: https://github.com/NVlabs/DigitalTwinArt

1. Introduction

Articulated objects are all around us. Whenever we open a door, close a drawer, turn on a faucet, or use scissors, we leverage the complex, physics-based understanding of various object parts and how they interact. Reconstructing novel articulated objects from visual observations is therefore an important problem for robotics and mixed reality. In this work, we aim to democratize the process of building a 3D reconstruction that accurately describes an articulated object, including the part geometries, segmentation and their joint articulations, as shown in Figure 1.

The problem of generating digital twins of articulated objects has been long studied [4, 5, 7, 9, 11, 14, 23, 33, 34, 39, 40]. Two recent approaches to this problem are Ditto [11] and PARIS [16]. Both works reconstruct part-level geometries and the articulation model based on observations of the object at two joint states. Ditto is a feed-forward method that takes two multi-view fused point clouds as input. It is trained on a collection of objects from certain categories. Although Ditto shows generalizability to objects unseen during training, it is not equipped with the capability to handle arbitrary unknown objects drastically different from the training categories. PARIS takes multi-view posed images as input and optimizes an implicit representation directly from the input data without pretraining, ensuring better generalizability. However, the optimization process of PARIS depends significantly on initializations and lacks stability, as we will show in the experimental results. In addition, both PARIS and Ditto only handle two-part objects.

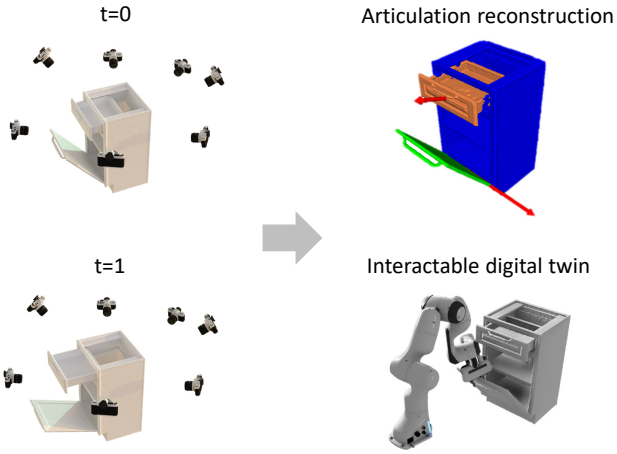

Figure 1. Our method requires two RGB-D scans of the object in each of two articulation states (left). The output is a 3D reconstruction with parts segmented, joint types identified, and joint axes estimated (top right). Note that multiple joints are allowed. The resulting digital twin can be imported into a physics-based simulator for interaction (bottom right).

In this paper, we take a step toward addressing the limitations of prior works by proposing a system with the following unique characteristics: a) the ability to handle arbitrary novel objects, regardless of the discrepancy of their motion, shape, or size from the training distribution; b) the scalability to objects with multiple moving parts; c) the robustness to initializations of the high dimensional optimization space of the articulation reconstruction problem.

Given multi-view RGB-D observations of the object at

*work done during internship at NVIDIA

two different joint states, our proposed method reconstructs articulated part meshes and joint parameters. We adopt a two-stage approach, where we first reconstruct the object at each given state with an SDF representation, and then estimate the articulation model including part segmentation and joint parameters. We explicitly derive point-level correspondences between the two states from the articulation model, which can be readily supervised by minimizing the following losses: 1) consistency between 3D local geometry from one state to the other, 2) 2D pixel correspondences from image feature matches, and, 3) physically-based reasoning in the form of minimizing articulation collisions.

We demonstrate the efficacy of our method on multiple challenging datasets, such as the dataset introduced by PARIS [16] which includes both synthetic and real scenes, We also introduce a novel synthetic dataset composed by objects with more than one joint. Extensive experiments indicate our approach generalizes to various types of objects, including those challenging ones consisting of both revolute and prismatic joints. Our method is also shown to produce more stable results than baselines under different initializations. We summarize our contributions as follows:

- We present a framework that reconstructs the geometry and articulation model of unknown articulated objects. It is per-object optimized and applicable to arbitrary articulated objects without assuming any object shape or structure priors.
- Our method decouples the problem into object shape reconstruction and articulation model reasoning. By jointly optimizing a set of loss terms on a *point correspondence field*, derived from the articulation model, we effectively leverage cues from image feature matching, 3D geometry reconstructions, as well as kinematic rules.
- Extensive evaluation on both synthetic and real-world data indicates our approach outperforms existing state-of-art methods consistently and stably.
- We demonstrate generalizability to complex unknown articulated objects consisting of more than one movable part, using only multi-view scans at two different articulation states.

2. Related Work

Articulated Object Prior Learning. A number of works leverage deep learning to train over large-scale 3D articulated assets offline in order to learn articulation priors, including part segmentation [1, 7, 8, 10, 29, 32, 33, 39], kinematic structure [1, 4, 7, 29, 33, 34, 39, 40], pose estimation [4, 14, 15, 32, 36], and articulated shape reconstruction [5, 11, 14, 23, 34]. In particular, Ditto [11] and CARTO [5] share the same objective as ours in building a full digital twin of the object, including shape reconstruction, part segmentation and articulation reasoning. Ditto [11] builds on top of PointNet++ [28] to process multi-view fused point cloud. CARTO [5] learns a single geometry and articulation decoder for multiple object categories by taking as input the stereo images. While promising results have been shown by the above methods, collecting large amount of training data is non-trivial in the real world due to the annotation complexity. The availability of 3D articulated object models is also notably limited compared to their single rigid counterparts to produce diverse synthetic training data. This results in struggling with out-of-distribution test set, as validated in our experiments. In contrast, our method does not require training on articulated assets and can be applied to a broad range of unknown articulated objects without any category restriction.

Per-Object Optimization. Per-object optimization methods perform test-time optimization to better adapt to the novel unknown objects [16, 17, 25]. By circumventing learning priors on 3D articulated assets, this type of approach can in theory generalize to arbitrarily unknown objects. Watch-it-move [25] demonstrates self-discovery of 3D joints for novel view synthesis and re-posing. However, it focuses on revolute joints and objects such as humans, quadrupeds and robotic arms as opposed to the daily-life objects considered here. [17] proposes an energy minimization approach to jointly optimize the part segmentation, transformation, and kinematics, while requiring a sequence of complete point cloud as input. Among these methods, PARIS [16] shares the closest setup considered in this work by taking two scans corresponding to the initial and final states of the unknown object and building a full digital twin. It focuses on objects with a single movable part, modeling both the static and dynamic part separately with individual neural fields. As we show, this design decision results in less robustness and efficiency, and it thus prevents generalization to more complex multi-joint objects, such as those handled by our method.

Articulation Reasoning by Interaction. Prior work [3, 6, 19] leverages physical interaction to create novel sensory information so as to reason the articulation model based on object state changes. [9, 12] pioneer the effort to introduce interactive perception into the estimation of articulation models. Follow up works further explore with hierarchical recursive Bayesian filters [21], probabilistic models [30], geometric models from multi-view stereo [9], and feature tracking [27]. Where2Act [22] presents a learnable framework to estimate action affordance on articulated objects from a single RGB image or point cloud while limited to single step interaction. AtP [3] learns interaction strategies to isolate parts for effective part segmentation and kinematic reasoning. However, most of the methods focus on learning interaction policies for effective part segmentation or motion analysis and do not aim for 3D reconstruction, which is part of the goal in this work. Recent work [6] extends Ditto [11] to an interactive setup which enables full

digital twining. Nevertheless, its dependency on pretraining shares the similar issues as the articulation prior learning methods. The assumption on perfect depth sensing without viewpoint issues also hinders direct application to noisy real-world data.

3. Method

We address the problem of building digital twins of unknown multi-part articulated objects from observations of the object at two different joint states. Specifically, we reconstruct per-part shapes and the articulation model of the object, given multi-view RGB-D observations and object masks $\{(I_v^t, \text{Depth}_v^t, \text{Mask}_v^t)\}_{v=0,\dots,V-1}$ with known camera parameters at object initial state $t = 0$ and final state $t = 1$. Typically the number of images $V \approx 100$. We also assume the number of joints is given.

Figure 2 presents an overview of our framework. We factorize the reconstruction problem into two stages with distinct focuses. Stage one (§3.1) reconstructs the object-level shape at each state, which are independent of articulation. Stage two (§3.2) recovers the articulation model including part segmentation and part motions by exploiting correspondences between per-state reconstructions.

3.1. Per-State Object Reconstruction

Given multi-view posed RGB-D images of the object $\mathcal{O}^t$ at state $t \in \{0, 1\}$, we aim to reconstruct object geometry, represented by a Neural Object Field [35] (Ω^t, Φ^t) (we omit t for simplicity in the following), where the geometry network $\Omega : \mathbf{x} \mapsto d$ maps spatial point $\mathbf{x} \in \mathbb{R}^3$ to its truncated signed distance $d \in \mathbb{R}$, and the appearance network $\Phi : (\mathbf{x}, \mathbf{d}) \mapsto \mathbf{c}$ maps point $\mathbf{x} \in \mathbb{R}^3$ and view direction $\mathbf{d} \in \mathbb{S}^2$ to RGB color $\mathbf{c} \in \mathbb{R}_+^3$.

The networks Ω and Φ are implemented with multi-resolution hash encoding [24] and are supervised with RGB-D images via color rendering loss $\mathcal{L}_c$ and SDF loss $\mathcal{L}_{SDF}$. We follow the approach of BundleSDF [35] and defer details to the appendix.

After optimization, we obtain the object mesh $\mathcal{M}^t$ by extracting the zero level set from Ω using marching cubes [18], from which we can further compute the Euclidean signed distance field (ESDF) $\tilde{\Omega}(\mathbf{x})$, as well as the occupancy field $\text{Occ}(\mathbf{x})$, defined as

$$\text{Occ}(\mathbf{x}) = \text{clip}(0.5 - \frac{\tilde{\Omega}(\mathbf{x})}{s}, 0, 1), \tag{1}$$

where s is set to a small number to make the function transition continuously near the object surface.

3.2. Segmentation and Motion Reconstruction

Given object-level reconstructions $(\mathcal{M}^t, \tilde{\Omega}^t, \text{Occ}^t, \Phi^t)$ at two different articulation states $t \in \{0, 1\}$, we aim to discover the underlying articulation model that associates them to each other, namely part segmentation and per-part rigid transformations between states. Our key idea is to derive a *point correspondence field* between states from the articulation model and supervise it with rich geometry and appearance information obtained from the first stage.

For an articulated object with M parts, we model its articulation from state t to state $t' = 1 - t$ with 1) a part segmentation field $f^t : \mathbf{x} \mapsto i$ that maps spatial point $\mathbf{x} \in \mathcal{O}^t$ from the object at state t to a part label $i \in \{0, \dots, M-1\}$, and 2) per-part rigid transformation $\mathcal{T}_i^t = (R_i^t, \mathbf{t}_i^t) \in \mathbb{SE}(3)$ that transforms part i from state t to state t'.

For differentiable optimization, instead of hard assignment f of points to parts, we model part segmentation as a probability distribution over parts. Formally, we let $P^t(\mathbf{x}, i)$ be the probability that point $\mathbf{x}$ in state t belongs to part i.

P^t is implemented as a dense voxel-based 3D feature volume followed by MLP segmentation heads. For rigid transformations, we parameterize rotations with the 6D representation as in [42] and translations as 3D vectors.

We can now derive the *point correspondence field* that maps any object point $\mathbf{x}$ from state t to its new position $\mathbf{x}^{t\to t'}$ at state t' when it moves *forward* with the motion of the part it belongs to. The field can be seen as a way to "render" the articulation model for supervision. Formally,

$$\mathbf{x}^{t\to t'} = \overrightarrow{\text{Fwd}}(\mathbf{x}, f^t, \mathcal{T}^t) = \sum_i P^t(\mathbf{x}, i)(R_i^t \mathbf{x} + \mathbf{t}_i^t). \tag{2}$$

This scheme is similar to classical linear blend skinning [13]. **Shared Motion.** We optimize two articulation models $(f^0, \mathcal{T}^0)$, $(f^1, \mathcal{T}^1)$ starting from both states. As they describe the same articulations, we share the part motions $\mathcal{T}$ to reduce redundancy and share supervision signal. Formally,

$$\mathcal{T}_i^1 = (R_i^0, \mathbf{t}_i^0)^{-1}, \forall i \tag{3}$$

Given the *point correspondence field*, we can supervise it with rich geometry and appearance information from object-level reconstructions and image observations. Specifically, we propose the following losses.

Consistency Loss. Corresponding points should have consistent local geometry and appearance at their respective states, which we can query from stage one's reconstructions. For near-surface points $\mathbf{x} \in \mathcal{X}_{surf}^t = \{\mathbf{x} \mid |\tilde{\Omega}(\mathbf{x})| < \lambda_{surf}\}$, we expect its correspondence $\mathbf{x}^{t\to t'}$ to have consistent SDF and color. Formally, we define *SDF consistency loss* l_s and *RGB consistency loss* l_c as

$$l_s(\mathbf{x}) = (\tilde{\Omega}^t(\mathbf{x}) - \tilde{\Omega}^{t'}(\mathbf{x}^{t\to t'}))^2, \tag{4}$$

$$l_c(\mathbf{x}) = \left\| \Phi^t(\mathbf{x}, \mathbf{d}) - \Phi^{t'}(\mathbf{x}^{t\to t'}, \mathbf{d}') \right\|_2^2, \tag{5}$$

where $\mathbf{d}$ denotes the direction of the ray $\mathbf{x}$ sampled from, $\mathbf{d}'$ denotes $\mathbf{d}$ transformed by $\mathbf{x}$'s part motion.

To extend supervision to points away from the surface, for which we are less confident about the reconstructed SDF or color, we enforce consistency on their occupancy values.

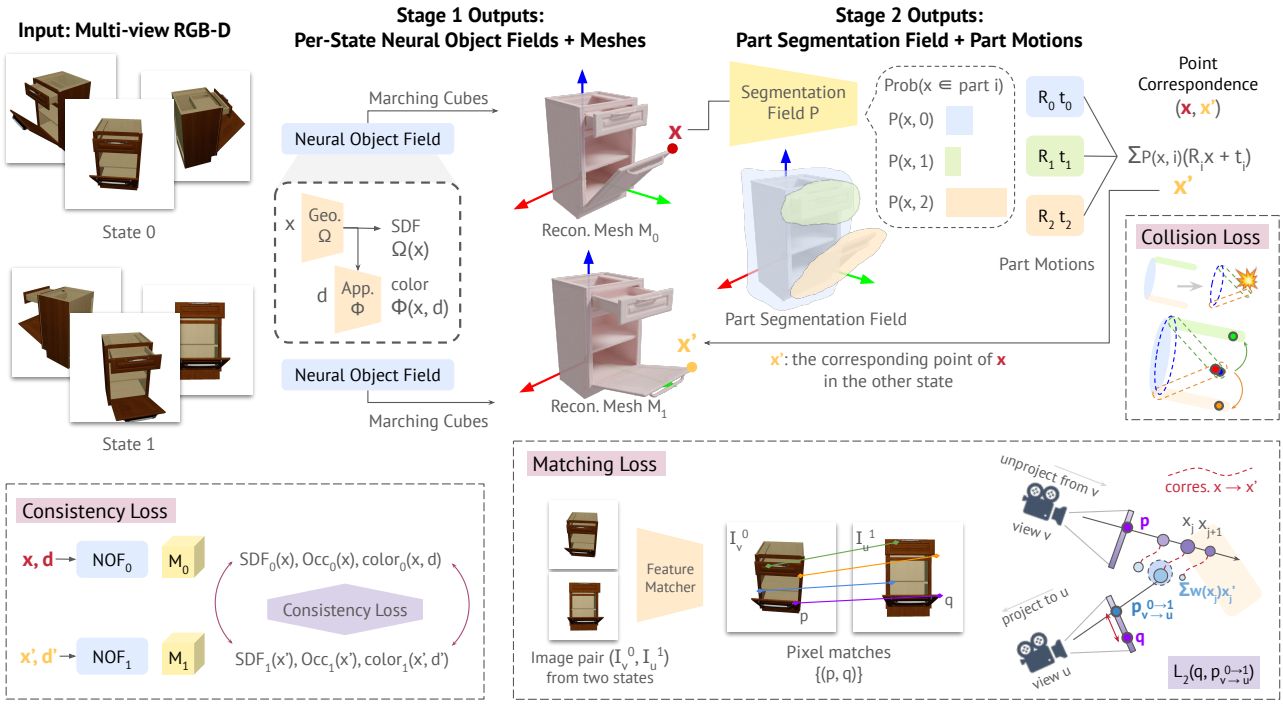

Figure 2. Overview of our method. In Stage 1, given multi-view RGB-D scans for the object at the initial and final articulation states, two neural object fields are optimized for each state. Upon learning convergence, the meshes corresponding to the two states are extracted. In Stage 2, the part segmentation field and per-part motions are optimized with three losses: consistency, matching, and collision. Together, the segmentation field and part motions yield point correspondence between the two states.

Formally, we define *occupancy consistency loss* l_o as

$$l_o = \left\| \mathrm{Occ}^t(\mathbf{x}) - \mathrm{Occ}^{t'}(\mathbf{x}^{t\to t'}) \right\|_2^2 \tag{6}$$

We enforce SDF and color consistency loss on points $\mathbf{x}$ sampled along camera rays $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ and weigh points based on their proximity to the object surface, similar to [35]. Meanwhile, we enforce occupancy consistency loss on points uniformly sampled from the unit space. Formally, we define consistency loss $\mathcal{L}_{\text{cns}}$ as

$$\mathcal{L}_{\text{cns}} = \mathbb{E}_{\mathbf{x}\in\mathcal{X}^t_{surf}} \left[w^t(\mathbf{x})(\lambda_s l_s(\mathbf{x}) + \lambda_c l_c(\mathbf{x})) \right] + \mathbb{E}_{\mathbf{x}} \left[\lambda_o l_o(\mathbf{x}) \right], \tag{7}$$

$$w(\mathbf{x}) = \mathrm{Sigmoid}(-\alpha\tilde{\Omega}(\mathbf{x})) \cdot \mathrm{Sigmoid}(\alpha\tilde{\Omega}(\mathbf{x})) \tag{8}$$

where $w(\mathbf{x})$ is a bell-shaped function that peaks at the object surface, hyperparameter α controls its sharpness, and hyperparameters $\lambda_s, \lambda_c, \lambda_o$ weigh different loss terms. However, as consistency loss is based on local descriptions, and there is a large solution space for each point, it can sometimes be challenging to arrive at the correct solution when optimizing for consistency loss alone.

Matching Loss. We propose to exploit visual cues from image observations, by leveraging 2D pixel matches across images at two states, obtained by LoFTR [31].

For image I^t_v taken from view v at state t, we select K images $\{I^{t'}_u \mid u \in \mathcal{N}_v\}$ from state t', where $\mathcal{N}_v$ are the viewpoints at t' that are closest to view v. We feed each image pair $(I^t_v, I^{t'}_u)$ to LoFTR to get L pairs of sparse and potentially noisy pixel matches $\mathcal{M}_{v,u,t} = \{(\mathbf{p}_j, \mathbf{q}_j)\}_j$.

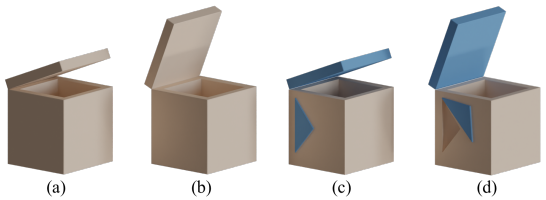

Figure 3. Motivation for collision loss. (a), (b) are the observations for the object at initial and final state respectively. Suppose the object is wrongly segmented as shown in (c), where blue represents the movable part. Moving the part with the forward motion will result in (d). In this case, wrong segmentation field still results in low consistency loss for SDF and color. Therefore, we introduce additional collision loss.

For pixel pair $(\mathbf{p}, \mathbf{q})$, let $\mathbf{r}$ be the camera ray from view v that passes through $\mathbf{p}$, the 2D correspondence of $\mathbf{p}$ at state t' from view u can be approximated with

$$\mathbf{p}^{t\to t'}_{v\to u} = \pi_u \left(\frac{\sum_{\mathbf{x}\in\mathbf{r}\cap\mathcal{X}_{surf}} w^t(\mathbf{x})\mathbf{x}^{t\to t'}}{\sum_{\mathbf{x}\in\mathbf{r}\cap\mathcal{X}_{surf}} w^t(\mathbf{x})} \right), \tag{9}$$

where π_u is a projection to view u, $w^t(x)$ is as in Eq. (8).

The matching loss is averaged over all matching pixel pairs from all image pairs:

$$\mathcal{L}_{\text{match}} = \mathbb{E}_{\substack{(\mathbf{p},\mathbf{q})\in\mathcal{M}_{v,u,t}, u\in\mathcal{N}_v, \\ v=0,\ldots,V-1, t\in\{0,1\}}} \left\| \mathbf{p}^{t\to t'}_{v\to u} - \mathbf{q} \right\|_2^2, \tag{10}$$

Collision Loss. A solution that minimizes the consistency loss may still be wrong. As illustrated in Fig. 3, a wrong segmentation still leads to a low consistency loss. The matching loss will not completely resolve the issue, either,

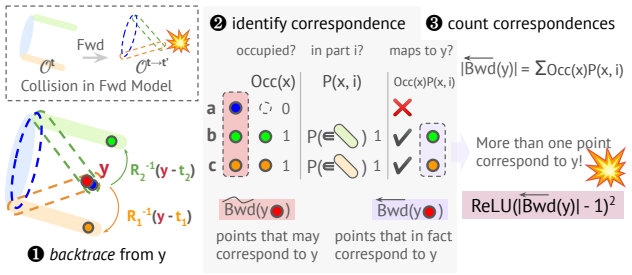

Figure 4. Illustration of the collision loss. We aim to detect and penalize collisions between parts after applying the predicted forward motion (moving the two sticks inwards). For point $\mathbf{y}$ at state t', we backtrace a set of points $\widetilde{\text{Bwd}}(\mathbf{y})$ ($\{a, b, c\}$) that may move to $\mathbf{y}$, by transforming $\mathbf{y}$ with each part's inverse motion (moving outwards following the arrows). We then check if the candidate point $\mathbf{x}_i$ obtained with part i's motion is indeed a point in part i, by looking up its occupancy and part label. Finally, we obtain the set of points $\overleftarrow{\text{Bwd}}(\mathbf{y})(\{b, c\})$ that in fact map to $\mathbf{y}$ under the articulation model, and report collision if there are more than one point that maps to $\mathbf{y}$, i.e., $|\overleftarrow{\text{Bwd}}(\mathbf{y})| > 1$.

because pixel matches can be noisy and sparse, and they mostly constrain near-surface points and do not work for points deep inside the object. On the other hand, if we look at the per-part transformed object $\mathcal{O}^{t\to t'} = \{\mathbf{x}^{t\to t'} = \overrightarrow{\text{Fwd}}(\mathbf{x}, f^t, \mathcal{T}^t)\}_{\mathbf{x}\in\mathcal{O}^t}$, as shown in Fig. 3(d), we do observe artifacts as a result of the wrong segmentation, namely the collision between the triangle and the base. Therefore, we propose to look at the entirety of $\mathcal{O}^{t\to t'}$ and check for artifacts. Fig. 4 illustrates the idea. To detect collision, we start from a point $\mathbf{y}$ at state t', and *backtrace* a set of points at state t that may *forward* to it given $(f^t, \mathcal{T}^t)$,

$$\overleftarrow{\text{Bwd}}(\mathbf{y}, f^t, \mathcal{T}^t) = \{\mathbf{x} \mid \overrightarrow{\text{Fwd}}(\mathbf{x}, f^t, \mathcal{T}^t) = \mathbf{y}\} \tag{11}$$

To simplify, we consider cases where $\mathbf{x} \in \overleftarrow{\text{Bwd}}(\mathbf{y})$ follows one of M rigid part motions. We can iterate over all possible parts and obtain a candidate set $\widetilde{\text{Bwd}}(\mathbf{y})$,

$$\overleftarrow{\text{Bwd}}(\mathbf{y}) \subset \widetilde{\text{Bwd}}(\mathbf{y}) = \{(R_i^t)^{-1}(\mathbf{y} - \mathbf{t}_i^t)\}_i$$

During training, we use $\widetilde{\text{Bwd}}(\mathbf{y})$ as an approximation.

Candidate point $\mathbf{x}_i = (R_i^t)^{-1}(\mathbf{y} - \mathbf{t}_i^t)$ corresponds to $\mathbf{y}$ only if $\mathbf{x}_i$ is on part i, which can be verified by checking occupancy $\text{Occ}(\mathbf{x})$ and part segmentation $P(\mathbf{x}, i)$. Formally, we write $\mathbf{x}_i$'s probability of corresponding to $\mathbf{y}$ as

$$a_i = P^t(\mathbf{x}_i, i) \cdot \text{Occ}^t(\mathbf{x}_i), \tag{12}$$

where $\text{Occ}(\mathbf{x})$ is defined by Eq. (1).

We count the number of points that correspond to $\mathbf{y}$ by summing contributions from all $\mathbf{x}_i$ and report collision when the result is larger than 1. Formally, we define *collision loss* $\mathcal{L}_{\text{coll}}$ as

$$\mathcal{L}_{\text{coll}} = \mathbb{E}_{\mathbf{y}}\left[\text{ReLU}(|\overleftarrow{\text{Bwd}}(\mathbf{y})| - 1)^2\right], \tag{13}$$

$$|\overleftarrow{\text{Bwd}}(\mathbf{y})| = \sum_i a_i \tag{14}$$

where $\mathbf{y}$ is uniformly sampled in the unit space.

Handling Partial Observation. In many cases, we can only observe part of the object due to limited viewpoints or self-occlusions. This may lead to hallucinated regions in object reconstructions that interfere with correspondence reasoning. Moreover, the visible portion of the movable parts vary between states, e.g., an open drawer versus a fully closed one. Points only visible in one state, e.g., those in the interior of the drawer, may not find corresponding points in the other state's reconstruction. To address this issue, we compute the visibility of point $\mathbf{x}$ by projecting it to all camera views and checking if it is in front of the depth (at the projected pixel) beyond a certain threshold ϵ. Formally,

$$\text{vis}(\mathbf{x}) = \bigvee_{v=0}^{V-1} [d_v(\pi_v(\mathbf{x})) + \epsilon > \text{dist}_v(\mathbf{x})], \tag{15}$$

where $\bigvee$ denotes logical OR, d_v denotes observed depth at view v; $\pi_v(\mathbf{x})$ denotes 2D projection; $\text{dist}_v(\mathbf{x})$ denotes the distance along the optical axis from $\mathbf{x}$ to camera origin.

Let $\mathcal{U}^t = \{\mathbf{x} \mid \neg\,\text{vis}(\mathbf{x})\}$ denote the set of unobserved points at state t. During mesh extraction at the first stage, we enforce the space to be empty at these points by setting their TSDF to 1, such that surface reconstructions only contain observed regions. We also discount the point consistency loss at $\mathbf{x}$ by a factor of w_{vis} if $\mathbf{x}^{t\to t'}$(Eq. 2) $\in \mathcal{U}^{t'}$, i.e., the predicted correspondence in the other state is not observed. w_{vis} is set to a small nonzero number to avoid learning collapse, i.e., making all points correspond to unobserved points to reduce consistency loss.

Our total loss for the second stage is defined as

$$\mathcal{L} = \lambda_{\text{cns}}\mathcal{L}_{\text{cns}} + \lambda_{\text{match}}\mathcal{L}_{\text{match}} + \lambda_{\text{coll}}\mathcal{L}_{\text{coll}} \tag{16}$$

Explicit Articulated Object Extraction Given reconstructed shape and articulation models $(\mathcal{M}^t, P^t, \mathcal{T}^t), t \in \{0, 1\}$, we can extract an explicit articulated object model. To predict joint i, we take the shared part motion $\mathcal{T}_i^0 = (R_i^0, \mathbf{t}_i^0)$ and classify joint i as prismatic if $|\text{angle}(R_i^0)| < \tau_r$, and revolute otherwise. We then project $\mathcal{T}_i^0$ to the manifold of pure-rotational or translational transformations and compute joint axes and relative joint states. For part-level geometry, we first identify the state $t^* \in \{0, 1\}$ with better part visibility, e.g. when the drawer is open instead of closed. We then compute hard segmentation $f^{t^*}(\mathbf{x}) = \arg\max_i P^{t^*}(\mathbf{x}, i)$, and extract each part mesh as $\mathcal{P}_i^{t^*} = \{\mathbf{v} \mid \mathbf{v} \in \mathcal{M}^{t^*}, f^{t^*}(\mathbf{v}) = i\}$.

4. Experiments

4.1. Datasets

PARIS Two-Part Object Dataset. PARIS [16] created a dataset of daily-life two-part articulated objects, including 10 synthetic object instances from PartNet-Mobility [37] and 2 real-world objects captured with MultiScan [20] pipeline. Each object is observed at two joint states, where only one part ("movable part") moves across states, and the other part ("static part") remains static. Observations at each state consist of RGB images and object masks captured from 100 random views in the upper hemisphere. We

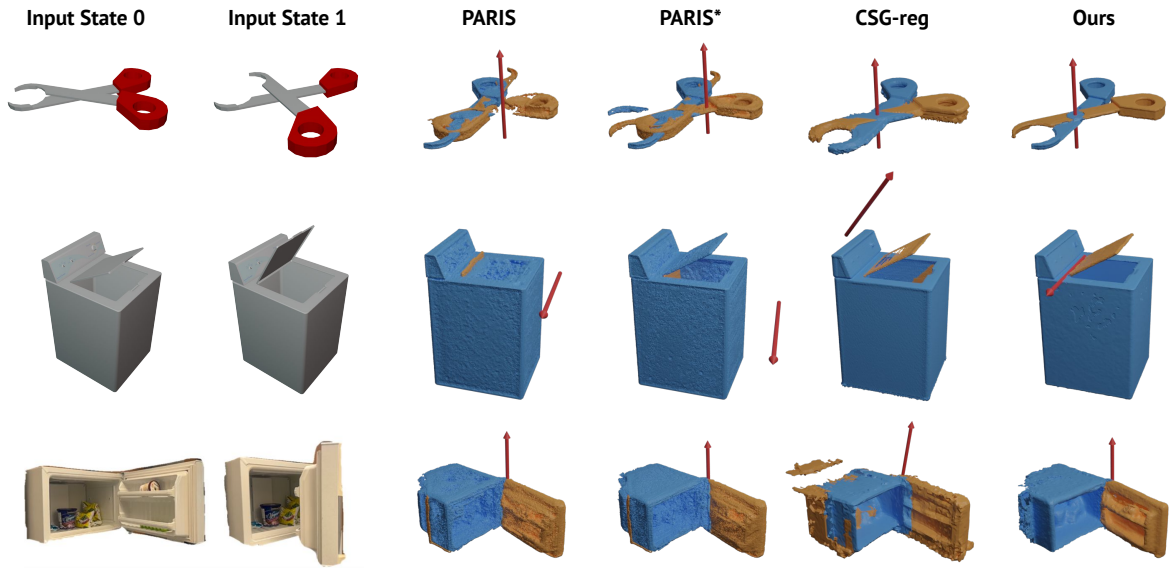

Figure 5. Qualitative results of shape reconstruction, part segmentation and joint prediction on PARIS dataset [16]. The top two rows correspond to synthetic data. The bottom row corresponds to real-world data. While PARIS and PARIS* occasionally work for these objects, depending upon the random seed, they often fail. Shown are the results from a typical trial that achieves near-average results.

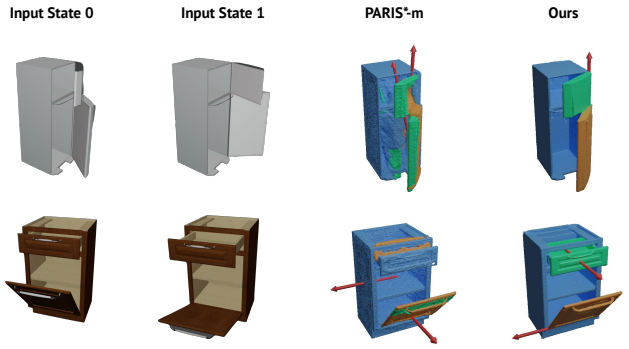

Figure 6. Qualitative results of shape reconstruction, part segmentation, and joint prediction on multi-part object dataset.

additionally rendered depth images for synthetic objects using the same camera parameters as PARIS, and retrieved depth data for real objects from raw RGB-D catpures.

Synthetic Multi-Part Object Dataset. We created 2 synthetic scenes using multi-part instances from PartNet-Mobility [37]. These objects consist of one static part and multiple movable parts. We capture them at two articulation states, where the multiple movable parts change their individual poses simultaneously across the two states. For each state, we randomly selected 100 views from the upper hemisphere and rendered RGB, depth, and object masks.

4.2. Metrics

Object- and Part-level Geometry. We evaluate object and part mesh reconstructions with bi-directional Chamfer-l_1 distance (CD), by sampling 10K points uniformly on the groundtruth and predicted meshes. We report **CD-w (mm)** for the whole object, **CD-s (mm)** for the static part, and **CD-m (mm)** for the movable part. Following [16], we report these values in millimeters.

Articulation Model and Cross-State Part Motion. We evaluate the estimated articulation model with **Axis Ang Err** (°), the angular error of the predicted joint axis for both revolute and prismatic joints, and **Axis Pos Err (0.1m)**, the minimum distance between the predicted and ground-truth joint axes for revolute joints. We also evaluate the estimated part motion between states with **Part Motion Err (° or m)** (referred to as Joint State by [16]), the geodesic distance error of predicted rotations for revolute joints, or the Euclidean distance error of translations for prismatic joints.

4.3. Baselines

Ditto [11] is a feed-forward model that reconstructs part-level meshes and the motion model (joint type, axis, and state) of a two-part articulated object given multi-view fused point cloud observations at two different joint states. It shares the same assumption as PARIS [16] that only one object part moves across states. We follow [16]'s protocol and report results from Ditto's released model pretrained on 4 object categories from Shape2Motion [33].

PARIS [16] reconstructs part-level shape and appearance as well as the motion model of a two-part articulated object, given multi-view RGB observations at two articulation states. It adopts a NeRF-based representation and performs per-object optimization such that it can be applied to arbitrary unknown objects. The object is modeled as the composition of a static part field and a mobile part field, as well as a transformation of the mobile part field that explains cross-

		Simulation											Real		
		FoldChair	Fridge	Laptop†	Oven†	Scissor	Stapler	USB	Washer	Blade	Storage†	All	Fridge	Storage	All
Axis Ang	Ditto [11]	89.35	89.30*	3.12	0.96	4.50	89.86	89.77	89.51	79.54*	6.32	54.22	1.71	**5.88**	**3.80**
	PARIS [16]	8.08±13.2	9.15±28.3	**0.02**±0.0	**0.04**±0.0	3.82±3.4	39.73±35.1	0.13±0.2	25.36±30.3	15.38±14.9	**0.03**±0.0	10.17±12.5	**1.64**±0.3	43.13±23.4	22.39±11.9
	PARIS* [16]	15.79±29.3	2.93±5.3	0.03±0.0	7.43±23.4	16.62±32.1	8.17±15.3	0.71±0.8	18.40±23.3	41.28±31.4	**0.03**±0.0	11.14±16.1	1.90±0.0	30.10±10.4	16.00±5.2
	CSG-reg	0.10±0.0	0.27±0.0	0.47±0.0	0.35±0.1	0.28±0.0	0.30±0.0	11.78±10.5	71.93±6.3	7.64±5.0	2.82±2.5	9.60±2.4	8.92±0.9	69.71±9.6	39.31±5.2
	3Dseg-reg	-	-	2.34±0.1	-	-	-	-	-	9.40±7.5	-	-	-	-	-
	Ours	**0.03**±0.0	**0.07**±0.0	0.06±0.0	0.22±0.0	**0.11**±0.0	**0.06**±0.0	**0.11**±0.0	**0.43**±0.0	**0.27**±0.0	0.06±0.0	**0.14**±0.0	2.10±0.0	18.11±0.2	10.11±0.1
Axis Ang	Ditto [11]	89.35	89.30*	3.12	0.96	4.50	89.86	89.77	89.51	79.54*	6.32	54.22	1.71	**5.88**	**3.80**
	PARIS [16]	8.08±13.2	9.15±28.3	**0.02**±0.0	**0.04**±0.0	3.82±3.4	39.73±35.1	0.13±0.2	25.36±30.3	15.38±14.9	**0.03**±0.0	10.17±12.5	**1.64**±0.3	43.13±23.4	22.39±11.9
	PARIS* [16]	15.79±29.3	2.93±5.3	0.03±0.0	7.43±23.4	16.62±32.1	8.17±15.3	0.71±0.8	18.40±23.3	41.28±31.4	**0.03**±0.0	11.14±16.1	1.90±0.0	30.10±10.4	16.00±5.2
	CSG-reg	0.10±0.0	0.27±0.0	0.47±0.0	0.35±0.1	0.28±0.0	0.30±0.0	11.78±10.5	71.93±6.3	7.64±5.0	2.82±2.5	9.60±2.4	8.92±0.9	69.71±9.6	39.31±5.2
	3Dseg-reg	-	-	2.34±0.11	-	-	-	-	-	9.40±7.5	-	-	-	-	-
	Ours	**0.03**±0.0	**0.07**±0.0	0.06±0.0	0.22±0.0	**0.11**±0.0	**0.06**±0.0	**0.11**±0.0	**0.43**±0.0	**0.27**±0.0	0.06±0.0	**0.14**±0.0	2.10±0.0	18.11±0.2	10.11±0.1
Axis Pos	Ditto [11]	3.77	1.02*	0.01	0.13	5.70	0.20	5.41	0.66	-	-	2.11	1.84	-	1.84
	PARIS [16]	0.45±0.9	0.38±1.0	**0.00**±0.0	**0.00**±0.0	2.10±1.4	2.27±3.4	2.36±3.4	1.50±1.3	-	-	1.13±1.1	**0.34**±0.2	-	**0.17**±0.1
	PARIS* [16]	0.25±0.5	1.13±2.6	**0.00**±0.0	0.05±0.2	1.59±1.7	4.67±3.9	3.35±3.1	3.28±3.1	-	-	1.79±1.5	0.50±0.0	-	0.25±0.0
	CSG-reg	0.02±0.0	**0.00**±0.0	0.20±0.2	0.18±0.0	**0.01**±0.0	0.02±0.0	0.01±0.0	2.13±1.5	-	-	0.26±0.2	1.46±1.1	-	0.73±0.6
	3Dseg-reg	-	-	0.10±0.0	-	-	-	-	-	-	-	-	-	-	-
	Ours	**0.01**±0.0	0.01±0.0	**0.00**±0.0	0.01±0.0	0.02±0.0	**0.01**±0.0	**0.00**±0.0	**0.01**±0.0	-	-	**0.01**±0.0	0.57±0.0	-	0.29±0.0
Part Motion	Ditto [11]	99.36	F	5.18	2.09	19.28	56.61	80.60	55.72	F	0.09	39.87	8.43	0.38	8.43
	PARIS [16]	131.66±78.9	24.58±57.7	**0.03**±0.0	**0.03**±0.0	120.70±50.1	110.80±47.1	64.85±84.3	60.35±23.3	0.34±0.2	0.30±0.0	51.36±34.2	2.16±1.1	0.56±0.4	1.36±0.7
	PARIS* [16]	127.34±75.0	45.26±58.5	**0.03**±0.0	9.13±28.8	68.36±64.8	107.76±68.1	96.93±67.8	49.77±26.5	0.36±0.2	0.30±0.0	50.52±39.0	**1.58**±0.0	0.57±0.1	1.07±0.1
	CSG-reg	**0.13**±0.0	0.29±0.0	0.35±0.0	0.58±0.0	0.20±0.0	0.44±0.0	10.48±9.3	158.99±8.8	0.05±0.0	0.04±0.0	17.16±1.8	14.82±0.1	0.64±0.1	7.73±0.1
	3Dseg-reg	-	-	1.61±0.1	-	-	-	-	-	0.15±0.0	-	-	-	-	-
	Ours	0.16±0.0	**0.09**±0.0	0.08±0.0	0.11±0.0	**0.15**±0.0	**0.05**±0.0	**0.11**±0.0	**0.25**±0.0	**0.00**±0.0	**0.00**±0.0	**0.10**±0.0	1.86±0.0	**0.20**±0.0	**1.03**±0.0
CD-s	Ditto [11]	33.79	3.05	0.25	**2.52**	39.07	41.64	2.64	10.32	46.90	9.18	18.94	47.01	16.09	31.55
	PARIS [16]	9.16±5.0	3.65±2.7	**0.16**±0.0	12.95±1.0	1.94±3.8	**1.88**±0.2	2.69±0.3	25.39±2.2	1.19±0.6	12.76±2.5	7.18±1.8	42.57±34.1	54.54±30.1	48.56±32.1
	PARIS* [16]	10.20±5.8	8.82±12.0	**0.16**±0.0	3.18±0.3	15.58±13.3	2.48±1.9	**1.95**±0.5	12.19±3.7	1.40±0.7	8.67±0.8	6.46±3.9	11.64±1.5	20.25±2.8	15.94±2.1
	CSG-reg	1.69	1.45	0.32	3.93	3.26	2.22	**1.95**	**4.53**	0.59	7.06	2.70	6.33	12.55	9.44
	3Dseg-reg	-	-	0.76	-	-	-	-	-	66.31	-	-	-	-	-
	Ours	**0.18**±0.0	**0.60**±0.0	0.32±0.0	4.66±0.0	**0.40**±0.1	2.65±0.0	2.19±0.0	4.80±0.0	**0.55**±0.0	**4.69**±0.0	**2.10**±0.0	**2.53**±0.0	**10.86**±0.1	**6.69**±0.0
CD-m	Ditto [11]	141.11	0.99	0.19	0.94	20.68	31.21	15.88	12.89	195.93	2.20	42.20	50.60	**20.35**	35.48
	PARIS [16]	8.99±7.6	7.76±11.2	0.21±0.2	28.70±15.2	46.64±40.7	19.27±30.7	5.32±5.9	178.43±131.7	25.21±9.5	76.69±6.1	39.72±25.9	45.66±31.7	864.82±382.9	455.24±207.3
	PARIS* [16]	17.97±24.9	7.23±11.5	**0.15**±0.0	6.54±10.6	16.65±16.6	30.46±37.0	10.17±6.9	265.27±248.7	117.99±213.0	52.34±11.0	52.48±58.0	77.85±26.8	474.57±227.2	276.21±127.0
	CSG-reg	1.91	21.71	0.42	256.99	1.95	6.36	29.78	436.42	26.62	1.39	78.36	442.17	521.49	481.83
	3Dseg-reg	-	-	1.01	-	-	-	-	-	6.23	-	-	-	-	-
	Ours	**0.15**±0.0	**0.27**±0.0	0.16±0.0	**0.47**±0.0	**0.41**±0.0	**2.27**±0.0	**1.34**±0.0	**0.36**±0.0	**1.50**±0.1	**0.37**±0.0	**0.73**±0.0	**1.14**±0.0	26.46±5.0	**13.80**±2.5
CD-w	Ditto [11]	6.80	2.16	0.31	**2.51**	1.70	2.38	2.09	7.29	42.04	**3.91**	7.12	6.50	14.08	10.29
	PARIS [16]	1.80±1.2	2.92±0.9	0.30±0.1	11.73±1.1	10.49±20.7	3.58±4.2	2.00±0.2	24.38±3.3	0.60±0.2	8.57±0.4	6.64±3.2	22.98±15.5	63.35±22.2	43.16±18.9
	PARIS* [16]	4.37±6.4	5.53±4.7	**0.26**±0.0	3.18±0.3	3.90±3.6	5.27±5.9	1.78±0.2	10.11±2.8	0.58±0.1	7.80±0.4	4.28±2.4	8.99±1.4	32.10±8.2	20.55±4.8
	CSG-reg	0.48	0.98	0.40	3.00	1.70	**1.99**	1.20	**4.48**	0.56	4.00	1.88	5.71	14.29	10.00
	3Dseg-reg	-	-	0.81	-	-	-	-	-	0.78	-	-	-	-	-
	Ours	**0.27**±0.0	**0.70**±0.0	0.35±0.0	4.18±0.0	**0.43**±0.0	2.19±0.0	**1.18**±0.0	4.74±0.0	**0.36**±0.0	3.99±0.0	**1.84**±0.0	**2.19**±0.0	**9.33**±0.0	**5.76**±0.0

Table 1. Results on PARIS dataset including both synthetic and real data. (Shown are the average ± stdev over 10 trials with different random seeds; see supplementary for details.) PARIS* [16] is augmented with depth for fair comparison under the same input modality. Objects with † are the seen categories that Ditto [6] has been trained on. Ditto sometimes gives wrong motion type predictions, which are noted with F for joint state and * for joint axis or position. Note that Blade, Storage, and Real Storage have prismatic joints, so there is no Axis Pos.

		Axis Ang 0	Axis Ang 1	Axis Pos 0	Axis Pos 1	Part Motion 0	Part Motion 1	CD-s	CD-m 0	CD-m 1	CD-w
Fridge-m	PARIS*-m [16]	34.52	15.91	3.60	1.63	86.21	105.86	8.52	526.19	160.86	15.00
	Ours	**0.16**	**0.10**	**0.01**	**0.00**	**0.11**	**0.13**	**0.61**	**0.40**	**0.52**	**0.89**
Storage-m	PARIS*-m [16]	43.26	26.18	10.42	-	79.84	0.64	8.56	128.62	266.71	8.66
	Ours	**0.21**	**0.88**	**0.05**	**-**	**0.13**	**0.00**	**0.85**	**0.21**	**3.46**	**0.99**

Table 2. Results on multi-part object dataset, averaged over 10 trials with different random seeds; see supplementary for details. PARIS*-m [16] is augmented with depth and extended to handle objects with more than two parts. Joint 1 of "Storage-m" is prismatic, so there is no Axis Pos.

	Axis Ang	Axis Pos	Part Motion	CD-s	CD-m
Proposed	0.26±0.04	0.01±0.00	0.09±0.01	0.80±0.08	0.33±0.00
W/o sharem	14.42±0.90	0.01±0.00	0.28±0.02	2.42±0.06	127.08±17.38
W/o matching	4.34±0.08	1.14±0.13	8.06±0.06	1.05±0.02	112.12±4.39
W/o rgb	0.43±0.11	0.01±0.00	0.13±0.01	0.76±0.03	0.35±0.00
W/o occ	0.42±0.14	0.01±0.00	0.12±0.02	7.16±5.73	164.33±92.62
W/o collision	0.18±0.03	0.01±0.00	0.10±0.02	11.64±0.67	191.76±2.08

Table 3. Ablation study on multi-part objects. We report results averaged over all joints/movable parts across 5 trials. *w/o shared motion* optimizes two separate sets of motion parameters for the two motion models at state 0 and 1. Other ablated versions omit the matching, RGB consistency, occupancy consistency, and collision loss, respectively.

state changes. Part fields and transformation are jointly optimized with an image rendering loss.

PARIS*. For fair comparison to our RGBD-based approach, we augmented PARIS with depth supervision following [2] and denote this version PARIS*.

PARIS*-m. The original PARIS [16] is limited to two-part objects. To make it applicable to more general P-part objects, we modified PARIS to optimize a static field and $P-1$ mobile fields, as well as their $P-1$ cross-state rigid transformations. We also augmented it with depth supervision.

CSG-reg. It reconstructs the object at each state with TSDF fusion, and applies Constructive Solid Geometry to per-state TSDF to get static (intersection) and movable (difference) parts similar to Ours-ICP baseline in PARIS. It then performs Fast Global Registration [41] and colored ICP [26] to align movable parts and estimate joint motions.

3Dseg-reg. It follows the same procedure as CSG-reg, but uses pretrained 3D object part segmentation model PAConv [38] to segment the reconstructed objects. Since PA-

Conv does not generalize well to unseen categories, we only report numbers for trained categories laptop and blade.

4.4. Experiment and Evaluation Setup.

We follow [11, 16]'s setting where part 0 is assumed to remain static ($R_0 = I, \mathbf{t}_0 = \mathbf{0}$). For multi-part objects, the number of parts is assumed known to all evaluated methods but joint types are unknown. In order to find the corresponding part for evaluation, we iterate through all possible pairs between predicted and ground-truth parts and report the best match with the smallest total chamfer distance. To remove floaters that disproportionately affect the chamfer distance, we apply a mesh clustering post-processing step to all methods, where we remove connected mesh components with less than $\tau = 10\%$ of the vertices of the largest cluster. Following [16], we transform our extracted parts with predicted motions to state $t = 0$ for evaluation.

We observed that optimization-based methods such as PARIS may yield different final results with different random initializations of the model. For comprehensive evaluation, we run all optimization-based methods 10 times with different random seeds, reporting the mean and standard deviation for each metric across the 10 trials. Please refer to the appendix for more statistics.

4.5. Results on PARIS Two-Part Object Dataset

Table 1 shows results on PARIS Two-Part Object Dataset including synthetic and real instances, summarized over 10 trials. Ditto relies on object shape and structure priors learned from training categories, which cover laptop, oven, and storage in evaluation. While it does well on seen categories, especially on shape reconstructions, a clear generalization gap can be observed in terms of unseen categories. PARIS exhibits large performance variances across trials for most instances. While it performs good reconstruction in some trials, it occasionally fails drastically, leading to overall much worse performance on both shape and articulation reconstruction. The depth supervision in PARIS* improves object-level shape reconstruction, bringing significant improvement in CD-w on challenging objects such as oven, scissor, washer from synthetic data and all real instances. At the same time, depth further complicates the optimization, leading to more failure cases and larger variance, resulting in worse average articulation predictions. Both CSG-reg and 3Dseg-reg do well on easy objects such as synthetic laptops but struggle elsewhere. Notably, segmentation errors (e.g., intersections containing the movable part of the blade, noise mistaken as movable part) easily propagate into traditional registration-based articulation estimation.

Our method is robust to initializations and consistently achieves accurate shape and articulation reconstruction across trials. We perform better than baseline methods on most instances. As shown in Fig. 5, our approach accurately reconstructs the part geometry and joint axis of real and synthetic objects, while baselines suffer from segmentation noises or complete failures.

4.6. Results on Multi-Part Objects

Table 2 summarizes results on multi-part objects across 10 trials. As shown in Fig. 6, with the increased complexity in object structure, PARIS*-m fails to correctly segment the object even coarsely and performs poorly on both shape and articulation reconstruction. PARIS' single image rendering objective fails to drive the optimization process to the correct solution. In contrast, our method achieves high-quality reconstruction with the help of rich information from 2D images, 3D geometries, and kinematics.

4.7. Ablation Study

We examine the effectiveness of our design choices on multi-part objects since they are more challenging. We report joint prediction and part reconstruction metrics averaged over all joints/movable parts and instances across 5 random trials. As shown in Table 3, sharing motions between the two articulation models significantly improves all metrics by leveraging information from both directions. Matching loss also effectively helps guide articulation reconstruction. Collision loss and occupancy consistency loss both add constraints to the whole space beyond surface region, improving dense field learning and thus resulting in higher-quality part-level reconstructions.

5. Conclusion

We presented a framework that reconstructs the geometry and articulation model of unknown articulated objects given two scans of the object at different joint states. It is per-object optimized and applies to arbitrary articulated objects without assuming any category or articulation priors. It also handles more than one movable part. Our proposed two-stage approach disentangles the problems of object-level shape reconstruction and articulation reasoning. By enforcing a set of carefully designed loss terms on a point correspondence field derived from the articulation model, our method effectively leverages cues from image feature matching, object geometry reconstructions, as well as kinematic rules. Extensive experiments indicate our approach achieves more accurate and stable results than prior work. However, challenges remain in applying the method to more general settings, e.g., where camera poses are unknown and object base parts are unaligned. How to fuse observations at different states when the occlusion pattern changes is also an interesting open problem. We leave them to future work.

References

[1] Hameed Abdul-Rashid, Miles Freeman, Ben Abbatematteo, George Konidaris, and Daniel Ritchie. Learning to infer kinematic hierarchies for novel object instances. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8461–8467. IEEE, 2022. 2

[2] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, 2022. 7, 2

[3] Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15752–15761, 2021. 2

[4] Nick Heppert, Toki Migimatsu, Brent Yi, Claire Chen, and Jeannette Bohg. Category-independent articulated object tracking with factor graphs. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3800–3807. IEEE, 2022. 1, 2

[5] 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 Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[6] Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. *ICRA*, 2023. 2, 7

[7] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017. 1, 2

[8] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas Guibas. Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[9] Xiaoxia Huang, Ian Walker, and Stan Birchfield. Occlusion-aware multi-view reconstruction of articulated objects for manipulation. *Robotics and Autonomous Systems*, 62(4): 497–505, 2014. 1, 2

[10] Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. Opd: Single-view 3d openable part detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 410–426. Springer, 2022. 2

[11] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 6, 7, 8

[12] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 5003–5010. IEEE, 2013. 2

[13] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46, 2007. 3

[14] Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Unsupervised pose-aware part decomposition for man-made articulated objects. In *European Conference on Computer Vision*, pages 558–575. Springer, 2022. 1, 2

[15] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3706–3715, 2020. 2

[16] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-level reconstruction and motion analysis for articulated objects. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 5, 6, 7, 8, 3

[17] Shaowei Liu, Saurabh Gupta, and Shenlong Wang. Building rearticulable models for arbitrary 3d objects from 4d point clouds. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[18] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 3, 1

[19] Liqian Ma, Jiaojiao Meng, Shuntao Liu, Weihang Chen, Jing Xu, and Rui Chen. Sim2real^2: Actively building explicit physics model for precise articulated object manipulation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2023. 2

[20] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 5

[21] Roberto Martín-Martín, Sebastian Höfer, and Oliver Brock. An integrated approach to visual perception of articulated objects. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 5091–5097. IEEE, 2016. 2

[22] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021. 2

[23] 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/CVF International Conference on Computer Vision*, pages 13001–13011, 2021. 1, 2

[24] 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):1–15, 2022. 3, 1

[25] 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 Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[26] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 143–152, 2017. 7

[27] Sudeep Pillai, Matthew R Walter, and Seth Teller. Learning articulated motions from visual demonstration. *RSS*, 2014. 2

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[29] Yahao Shi, Xinyu Cao, and Bin Zhou. Self-supervised learning of part mobility from point cloud sequence. In *Computer Graphics Forum*, pages 104–116. Wiley Online Library, 2021. 2

[30] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011. 2

[31] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[32] Wei-Cheng Tseng, Hung-Ju Liao, Lin Yen-Chen, and Min Sun. Cla-nerf: Category-level articulated neural radiance field. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8454–8460. IEEE, 2022. 2

[33] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019. 1, 2, 6

[34] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 15816–15826, 2022. 1, 2

[35] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 606–617, 2023. 3, 4, 1

[36] 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/CVF International Conference on Computer Vision*, pages 13209–13218, 2021. 2

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

[38] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021. 7

[39] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *ACM Transactions on Graphics*, 37(6): 1–15, 2018. 1, 2

[40] Vicky Zeng, Tabitha Edith Lee, Jacky Liang, and Oliver Kroemer. Visual identification of articulated object parts. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2443–2450. IEEE, 2021. 1, 2

[41] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 766–782. Springer, 2016. 7

[42] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 3