Articulated Object Understanding from a Single Video Sequence

Arslan Artykov Clémentin Boittiaux Vincent Lepetit
LIGM, École des Ponts et Chaussees, IP Paris, CNRS, France
arslan.artykov@enpc.fr, clementin.boittiaux@enpc.fr, vincent.lepetit@enpc.fr

1. Generating Motion Hypotheses

As explained in the main paper, to generate hypotheses on the joints and their parameters, we rely on 3D trajectories of points lying on the object surface. We denote these 3D trajectories $T_i = \{N_i^t\}_{t_b^i \leq t \leq t_e^i}$. Each trajectory has its own time interval $[t_b^i; t_e^i]$, depending on how far the point could be tracked along the input video, starting from frame t_o^i. t_o^i is the time step of the frame we use to start the 2D track for trajectory T_i.

We generate good hypotheses for the articulated motions by randomly selecting one trajectory T_i, randomly pick a type of articulated motion—either prismatic or revolute, and compute the joint parameters for this trajectory. We then check if there are many other trajectories that can also be explained by these joint parameters. When it is the case, we keep the computed joint parameters as a good hypothesis. We remove the trajectories that can be explained by this hypothesis, and we iterate until we get the desired number H of hypotheses.

We detail here how we compute the joint parameters from a random trajectory, and how exactly we check that many other trajectories can be explained by them. This procedure is strongly inspired by RANSAC.

1.1. Prismatic Joints

If the randomly selected type of motion is "prismatic", we pick two random points $N_i^{t_1}$ and $N_i^{t_2}$ on trajectory T_i, with $t_2 > t_1$. This gives us the potential direction $d = N_i^{t_1} N_i^{t_2} / \|N_i^{t_1} N_i^{t_2}\|$ of the motion. From d, we can compute the amount of translation a_t for each time step $a_t = d^\top . (N_i^t - N_i^{t_o^i})$ by projecting point N_i^t on the line. Estimates a_t are however noisy in practice so we consider only their average $\overline{a_t}$. Using this average also allows us to extend the motion beyond the time interval $[t_b^i; t_e^i]$ of trajectory T_i.

Pair $d, \overline{a_t}$ is our motion hypothesis. To check if it explains other trajectories, we apply the estimated motion to the origin points $N_j^{t_o^j}$ of the other trajectories T_j. This predicts the 3D positions of these points in the other time steps according to the estimated motion:

$$\hat{N}_j^t = N_j^{t_o^j} + \overline{a_t}(t - t_o^j)d \,. \tag{1}$$

If $\hat{N}_j^t$ is close to its observed position N_j^t, i.e., if $\|\hat{N}_j^t N_j^t\| < \epsilon$ with ϵ a small threshold, this counts as an inlier. If the total number of inliers is sufficiently large, we keep these joint parameters as a good hypothesis. In practice, we require at least %5 of overall trajectories to be inliers.

1.2. Revolute Joints

If the randomly selected type of motion is "revolute", we proceed similarly, except that instead of a translation, we need to consider a 3D rotation.

In this case, our motion hypothesis will be represented as a triplet made of a rotation axis r, a pivot point P, and an average amount of rotation $\overline{\alpha_t}$.

To find a rotation axis and pivot point given a random trajectory T_i, we need to pick 3 points on T_i. Let's denote these 3 points by A, B, C.

We construct a 3D coordinate system (P, u, v, r), where P will be the pivot of the joint and center of circle going through A, B, and C. r is the revolution axis and the normal to plane going through A, B, and C:

- Take $u_1 = B - A$, $w_1 = (C - A) \wedge u_1$, $u = u1/||u_1||$, $r = w_1/||w_1||$, $v = r \wedge u$.
- 2D coordinates of B and C in 2D plane $((0,0), u, v)$ are $b = ((B - A).u, 0)$ and $c = ((C - A).u, (C - A).v)$.
- 2D coordinates of P in 2D plane $((0,0), u, v)$ are $p = (b_x/2, h)$ with $b_x = (B - A).u$ and h to be determined.
- h can be found by noting that the distance from p to c is the same as the distance from p to $(0,0)$:

$$(c_x - b_x/2)^2 + (c_y - h)^2 = (b_x/2)^2 + h^2 \,, \tag{2}$$

so

$$h = \frac{(c_x - b_x/2)^2 + c_y^2 - (b_x/2)^2}{2c_y} \,. \tag{3}$$

Finally, $P = A + (b_x/2)u + hv$.

Like for the translation, we use the average amount of rotation, i.e., the average $\overline{\alpha_t}$ of the amounts of rotation α_t's

for t between t_b^i and t_e^i. α_t can be computed as:

$$\alpha_t = \angle N_i^{t_o} P N_i^t = \text{atan2}(s, c) \,, \tag{4}$$

where

$$\begin{aligned} s &= \|(N_i^t - P) \wedge (N_i^{t_o} - P)\| / (||N_i^t - P||)(||N_i^{t_o} - P||)) \\ c &= (N_i^t - P)(N_i^{t_o} - P) / (||N_i^t - P||)(||N_i^{t_o} - P||)) \end{aligned} \tag{5}$$

To count the number of inliers, for each point N_j^t of the other trajectories T_j, we predict its position $\hat{N}_j^t$ according to the motion parameters we just estimated:

$$\hat{N}_j^t = P' + R(N_j^{t_o^j} - P') \,, \tag{6}$$

where R is the 3D rotation of axis r and angle $(t - t_o^j)\overline{\alpha_t}$ and $P' = P + ((N_j^t - P).r)r$. If $||N_j^t \hat{N}_j^t|| < \epsilon$ is lower than some threshold, we increment the number of inliers by one.

2. Efficiently identifying the correct hypotheses

In the main paper, we introduce the Bayesian Information Criterion (BIC):

$$\text{BIC}(\mathcal{C}) = k(\mathcal{C}) \ln(n) + \lambda \mathcal{L}(\mathcal{C}) \,, \tag{7}$$

where $\mathcal{C}$ is a combination of hypotheses in $\mathcal{H}$, $\mathcal{L}(\mathcal{C})$ is a loss function, $k(\mathcal{C})$ is the total number of parameters in $\mathcal{C}$, n is the number of observations, and λ is a hyperparameter, fixed for all experiments.

We seek combination $\hat{\mathcal{C}}$ that minimizes BIC($\hat{\mathcal{C}}$).

$k(\mathcal{C})$ is the sum of parameters of hypotheses h in $\mathcal{H}$:

$$k(\mathcal{C}) = \sum_{h \in \mathcal{C}} k(h) \,. \tag{8}$$

As explained in the main paper, $k(h) = 3$ if h is a prismatic motion and $k(h) = 5$ if h is a revolute motion

We use as loss $\mathcal{L}(\mathcal{C})$ the distance between the observed trajectories $\{N_i^t\}$ and the predicted trajectories $\{\hat{N}_i^t\}$ that were used to generate the hypotheses in $\mathcal{H}$.

We have:

$$\mathcal{L}(\mathcal{C}) = \sum_{h \in \mathcal{C}} \sum_{t_b^i \le t \le t_e^i} \|N_i^t - \hat{N}_i^t\|^2 + \sum_{h \in \mathcal{H} \setminus \mathcal{C}} \sum_{t_b^i \le t \le t_e^i} \|N_i^t - \overline{N_i}\|^2, \tag{9}$$

where $T_i = \{N_i^t\}_{t_b^i \le t \le t_e^i}$ is the 3D trajectory that was used to generate hypothesis h (we do not show the dependence of T_i on h in the equations to avoid making the notations more cumbersome).

$\mathcal{L}(\mathcal{C})$ is a sum over all the trajectories that generated the hypotheses in $\mathcal{H}$. The first term is the part of the loss function for the trajectories that generated hypotheses in $\mathcal{C}$: $\hat{N}_i^t$ is the position of N_i^t predicted by hypothesis h for time step t. The second term is the part of the loss function for hypotheses that are not in $\mathcal{C}$. If h is not in $\mathcal{C}$, we assume that this means trajectory T_i corresponds to a point that is *not* moving. We take $\overline{N_i}$ as an estimate of the actual position of the 3D points in trajectory T_i.

To compute BIC($\mathcal{C}$) efficiently, we introduce $\text{B}_1(h)$ and $\text{B}_1(h)$, which can be precomputed. We take:

$$\text{B}_1(h) = k(h) \ln(n) + \lambda \sum_{t_b^i \le t \le t_e^i} \|N_i^t - \hat{N}_i^t\|^2 \,, \tag{10}$$

and

$$\text{B}_2(h) = \lambda \sum_{t_b^i \le t \le t_e^i} \|N_i^t - N_i^{t_o^e}\|^2 \,. \tag{11}$$

It can be seen that BIC($\mathcal{C}$) can be computed as

$$\text{BIC}(\mathcal{C}) = \sum_{h \in \mathcal{H}} \mathbb{1}_{h \in \mathcal{C}} \text{B}_1(h) + (1 - \mathbb{1}_{h \in \mathcal{C}}) \text{B}_2(h) \,, \tag{12}$$

To find the combination that minimizes BIC($\mathcal{C}$), we simply compute its value for all 2^H possible combinations of hypotheses, and keep the combination that yields the lowest value.

We still have to explain how to compute n, the number of observations. This number is 3 times the number of observed points used in the loss function—3 times as each point has 3 coordinates. n can thus be computed as:

$$n = 3 \sum_{h \in \mathcal{H}} (t_e^i - t_b^i + 1) \,. \tag{13}$$

References

[1] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building Digital Twins of Articulated Objects from Interaction. In *CVPR*, 2022. 6

[2] Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In *ICLR*, 2024. 6

[3] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-Level Reconstruction and Motion Analysis for Articulated Objects. In *ICCV*, 2023. 6, 7

[4] Yijia Weng, Bowen Wen, Jonathan Tremblay, Valts Blukis, Dieter Fox, Leonidas Guibas, and Stan Birchfield. Neural Implicit Representation for Building Digital Twins of Unknown Articulated Objects. In *CVPR*, 2024. 6, 7

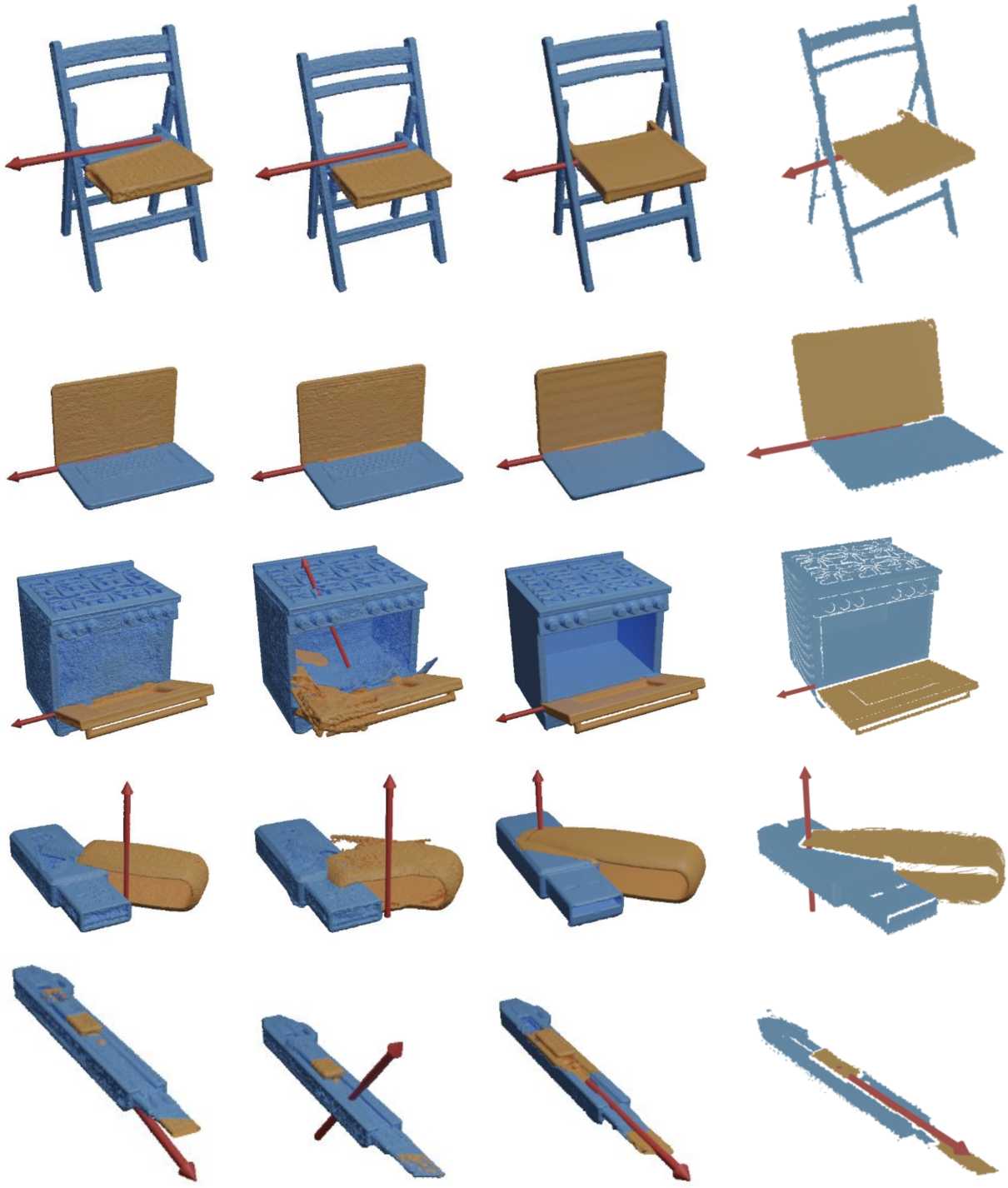

Figure 1. **Qualitative results on the Two-Part Objects dataset.** We show joint estimation, mesh reconstruction and part segmentation results.

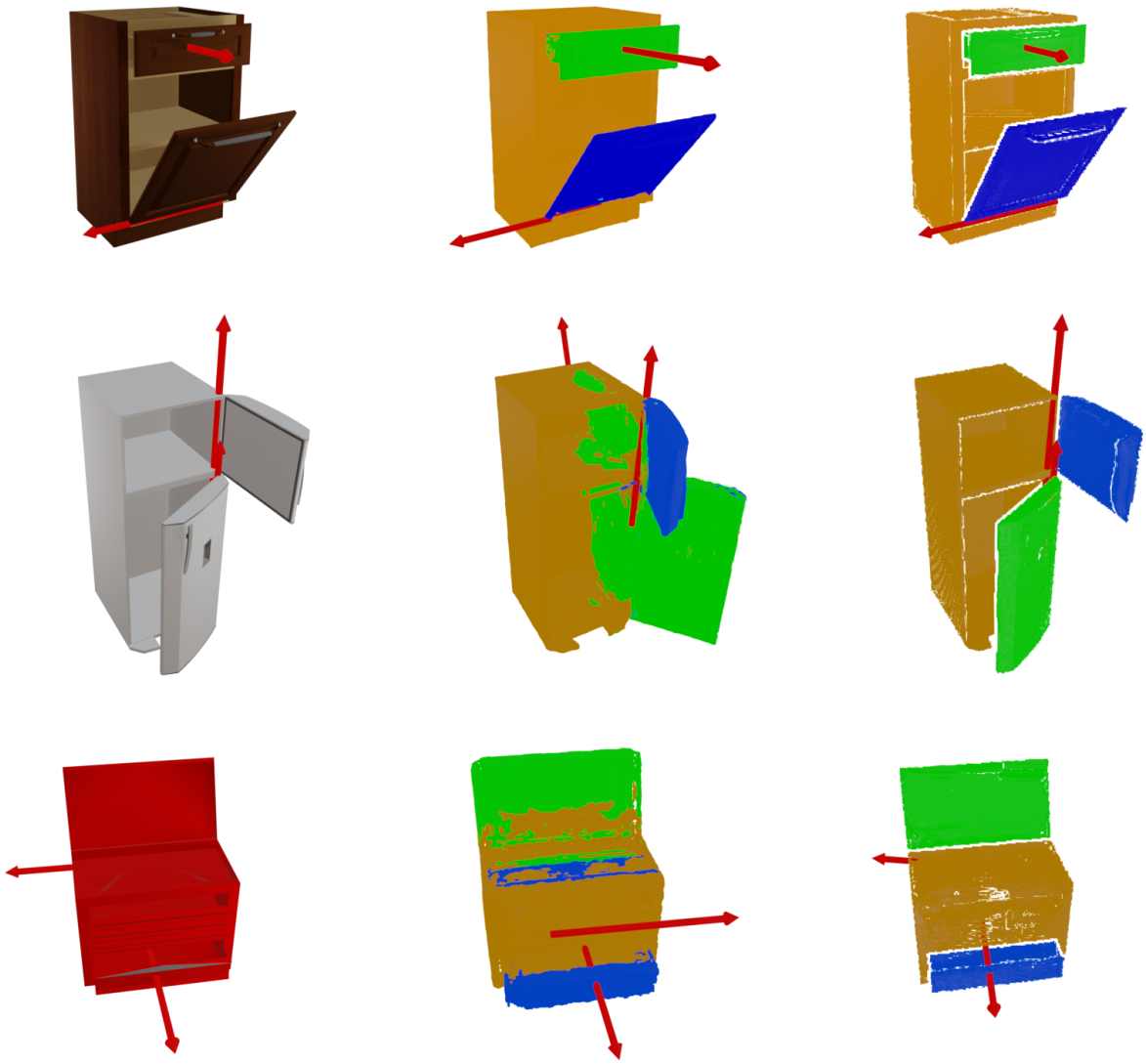

Figure 2. **Qualitative results on the Storage-47254, Fridge-10489, and Box-102377 objects from the multi-part object dataset.** We show joint estimation, mesh reconstruction and part segmentation results. Refer to the supplementary video for the animated mesh results.

Figure 3. **Qualitative results on the chair, storage, and laptop objects from the real objects dataset.** We show joint estimation, mesh reconstruction and part segmentation results. As shown in the figures, Articulate-Anything retrieves an incorrect mesh for the storage object and inaccurately predicts both the joint type and axis. Similarly, it predicts an incorrect joint axis for the laptop and an incorrect motion direction for the chair. It is important to note that Articulate-Anything does not perform geometry reconstruction; instead, it retrieves meshes from a predefined database. Therefore, our objective here is to evaluate its performance in mesh retrieval and joint prediction. Refer to the supplementary video for the animated mesh results.

		Two-Part						
		FoldChair	Fridge	Laptop†	Oven†	USB	Blade	All
Axis Ang	Ditto [1]	89.35	89.30*	3.12	0.96	89.77	79.54*	58.67
	PARIS [3]	8.08$_{\pm 13.2}$	9.15$_{\pm 28.3}$	0.02$_{\pm 0.0}$	0.04$_{\pm 0.0}$	0.13$_{\pm 0.2}$	15.38$_{\pm 14.9}$	5.47$_{\pm 9.43}$
	PARIS* [3]	15.79$_{\pm 29.3}$	2.93$_{\pm 5.3}$	0.03$_{\pm 0.0}$	7.43$_{\pm 23.4}$	0.71$_{\pm 0.8}$	41.28$_{\pm 31.4}$	11.33$_{\pm 15.03}$
	CSG-reg [4]	0.10$_{\pm 0.0}$	0.27$_{\pm 0.0}$	0.47$_{\pm 0.0}$	0.35$_{\pm 0.1}$	11.78$_{\pm 10.5}$	7.64$_{\pm 5.0}$	3.44$_{\pm 2.6}$
	3Dseg-reg [4]	-	-	2.34$_{\pm 0.11}$	-	-	9.40$_{\pm 7.5}$	-
	Articulate-Anything [2]	F	**0.00**$_{\pm 0.0}$	**0.00**$_{\pm 0.0}$	**0.00**$_{\pm 0.0}$	90.00$_{\pm 0.0}$	**0.00**$_{\pm 0.0}$	18.00$_{\pm 0.0}$
	DTA [4]	**0.03**$_{\pm 0.0}$	0.07$_{\pm 0.0}$	0.06$_{\pm 0.0}$	0.22$_{\pm 0.0}$	0.11$_{\pm 0.0}$	0.27$_{\pm 0.0}$	**0.13**$_{\pm 0.0}$
	Ours	0.34$_{\pm 0.0}$	1.54$_{\pm 0.0}$	0.30$_{\pm 0.0}$	1.46$_{\pm 0.0}$	**0.06**$_{\pm 0.0}$	0.15$_{\pm 0.0}$	0.64$_{\pm 0.0}$
Axis Pos	Ditto [1]	3.77	1.02*	0.01	0.13	5.41	-	2.59
	PARIS [3]	0.45$_{\pm 0.9}$	0.38$_{\pm 1.0}$	**0.00**$_{\pm 0.0}$	**0.00**$_{\pm 0.0}$	2.36$_{\pm 3.4}$	-	0.80$_{\pm 1.33}$
	PARIS* [3]	0.25$_{\pm 0.5}$	1.13$_{\pm 2.6}$	**0.00**$_{\pm 0.0}$	0.05$_{\pm 0.2}$	3.35$_{\pm 3.1}$	-	1.20$_{\pm 1.6}$
	CSG-reg [4]	0.02$_{\pm 0.0}$	**0.00**$_{\pm 0.0}$	0.20$_{\pm 0.2}$	0.18$_{\pm 0.0}$	0.01$_{\pm 0.0}$	-	0.07$_{\pm 0.2}$
	3Dseg-reg [4]	-	-	0.10$_{\pm 0.0}$	-	-	-	-
	Articulate-Anything [2]	F	2.71$_{\pm 0.0}$	0.06$_{\pm 0.0}$	0.39$_{\pm 0.0}$	5.46$_{\pm 0.0}$	-	1.72$_{\pm 0.0}$
	DTA [4]	**0.01**$_{\pm 0.0}$	0.01$_{\pm 0.0}$	**0.00**$_{\pm 0.0}$	0.01$_{\pm 0.0}$	**0.00**$_{\pm 0.0}$	-	**0.01**$_{\pm 0.0}$
	Ours	0.06$_{\pm 0.0}$	0.24$_{\pm 0.0}$	0.02$_{\pm 0.0}$	0.43$_{\pm 0.0}$	0.05$_{\pm 0.0}$	-	0.16$_{\pm 0.0}$
Part Motion	Ditto [1]	99.36	F	5.18	2.09	80.60	F	46.81
	PARIS [3]	131.66$_{\pm 78.9}$	24.58$_{\pm 57.7}$	**0.03**$_{\pm 0.0}$	**0.03**$_{\pm 0.0}$	64.85$_{\pm 84.3}$	0.34$_{\pm 0.2}$	36.92$_{\pm 36.85}$
	PARIS* [3]	127.34$_{\pm 75.0}$	45.26$_{\pm 58.5}$	**0.03**$_{\pm 0.0}$	9.13$_{\pm 28.8}$	96.93$_{\pm 67.8}$	0.36$_{\pm 0.2}$	46.51$_{\pm 38.3}$
	CSG-reg [4]	**0.13**$_{\pm 0.0}$	0.29$_{\pm 0.0}$	0.35$_{\pm 0.0}$	0.58$_{\pm 0.0}$	10.48$_{\pm 9.3}$	0.05$_{\pm 0.0}$	1.98$_{\pm 9.3}$
	3Dseg-reg [4]	-	-	1.61$_{\pm 0.1}$	-	-	0.15$_{\pm 0.0}$	-
	Articulate-Anything [2]	-	-	-	-	-	-	-
	DTA [4]	0.16$_{\pm 0.0}$	**0.09**$_{\pm 0.0}$	0.08$_{\pm 0.0}$	0.11$_{\pm 0.0}$	**0.11**$_{\pm 0.0}$	**0.00**$_{\pm 0.0}$	**0.09**$_{\pm 0.0}$
	Ours	0.19$_{\pm 0.0}$	1.01$_{\pm 0.0}$	2.57$_{\pm 0.0}$	0.15$_{\pm 0.0}$	0.54$_{\pm 0.0}$	0.12$_{\pm 0.0}$	0.76$_{\pm 0.0}$
CD-s	Ditto [1]	33.79	3.05	0.25	2.52	2.64	46.90	14.86
	PARIS [3]	9.16$_{\pm 5.0}$	3.65$_{\pm 2.7}$	0.16$_{\pm 0.0}$	12.95$_{\pm 1.0}$	2.69$_{\pm 0.3}$	1.19$_{\pm 0.6}$	4.97$_{\pm 1.6}$
	PARIS* [3]	10.20$_{\pm 5.8}$	8.82$_{\pm 12.0}$	0.16$_{\pm 0.0}$	3.18$_{\pm 0.3}$	1.95$_{\pm 0.5}$	1.40$_{\pm 0.7}$	4.29$_{\pm 3.22}$
	CSG-reg [4]	1.69	1.45	0.32	3.93	1.95	0.59	1.66
	3Dseg-reg [4]	-	-	0.76	-	-	66.31	-
	Articulate-Anything [2]	-	-	-	-	-	-	-
	DTA [4]	0.18$_{\pm 0.0}$	0.60$_{\pm 0.0}$	0.32$_{\pm 0.0}$	4.66$_{\pm 0.0}$	2.19$_{\pm 0.0}$	0.55$_{\pm 0.0}$	1.42$_{\pm 0.0}$
	Ours	**0.15**$_{\pm 0.0}$	**0.21**$_{\pm 0.0}$	**0.09**$_{\pm 0.0}$	**0.70**$_{\pm 0.0}$	**0.11**$_{\pm 0.0}$	**0.25**$_{\pm 0.0}$	**0.25**$_{\pm 0.0}$
CD-m	Ditto [1]	141.11	0.99	0.19	0.94	15.88	195.93	59.17
	PARIS [3]	8.99$_{\pm 7.6}$	7.76$_{\pm 11.2}$	0.21$_{\pm 0.2}$	28.70$_{\pm 15.2}$	5.32$_{\pm 5.9}$	25.21$_{\pm 9.5}$	12.70$_{\pm 8.27}$
	PARIS* [3]	17.97$_{\pm 24.9}$	7.23$_{\pm 11.5}$	0.15$_{\pm 0.0}$	6.54$_{\pm 10.6}$	10.17$_{\pm 6.9}$	117.99$_{\pm 213.0}$	61.73$_{\pm 44.48}$
	CSG-reg [4]	1.91	21.71	0.42	256.99	29.78	26.62	56.12
	3Dseg-reg [4]	-	-	1.01	-	-	6.23	-
	Articulate-Anything [2]	-	-	-	-	-	-	-
	DTA [4]	0.15$_{\pm 0.0}$	**0.27**$_{\pm 0.0}$	0.16$_{\pm 0.0}$	**0.47**$_{\pm 0.0}$	1.34$_{\pm 0.0}$	1.50$_{\pm 0.1}$	**0.65**$_{\pm 0.0}$
	Ours	**0.09**$_{\pm 0.0}$	1.44$_{\pm 0.0}$	**0.09**$_{\pm 0.0}$	18.74$_{\pm 0.0}$	**0.22**$_{\pm 0.0}$	**0.06**$_{\pm 0.0}$	3.44$_{\pm 0.0}$
CD-w	Ditto [1]	6.80	2.16	0.31	**2.51**	2.09	42.04	9.32
	PARIS [3]	1.80$_{\pm 1.2}$	2.92$_{\pm 0.9}$	0.30$_{\pm 0.1}$	11.73$_{\pm 1.1}$	2.00$_{\pm 0.2}$	0.60$_{\pm 0.2}$	3.23$_{\pm 0.62}$
	PARIS* [3]	4.37$_{\pm 6.4}$	5.53$_{\pm 4.7}$	0.26$_{\pm 0.0}$	3.18$_{\pm 0.3}$	1.78$_{\pm 0.2}$	0.58$_{\pm 0.1}$	2.62$_{\pm 2.0}$
	CSG-reg [4]	0.48	0.98	0.40	3.00	1.20	0.56	1.10
	3Dseg-reg [4]	-	-	0.81	-	-	0.78	-
	Articulate-Anything [2]	-	-	-	-	-	-	-
	DTA [4]	**0.27**$_{\pm 0.0}$	**0.70**$_{\pm 0.0}$	0.35$_{\pm 0.0}$	4.18$_{\pm 0.0}$	1.18$_{\pm 0.0}$	0.36$_{\pm 0.0}$	1.17$_{\pm 0.0}$
	Ours	0.44$_{\pm 0.0}$	0.78$_{\pm 0.0}$	**0.15**$_{\pm 0.0}$	4.91$_{\pm 0.0}$	**0.30**$_{\pm 0.0}$	**0.30**$_{\pm 0.0}$	**1.15**$_{\pm 0.0}$

Table 1. **Main metrics on the Two-Part Object dataset.** We report the means and standard deviations of the metrics for PARIS, PARIS*, DTA, and our method, computed over 10 trials. PARIS* [3] is the depth augmented version of PARIS. Objects with † belong to the seen categories that Ditto [1] was trained on. Ditto sometimes gives wrong motion type predictions, which are noted with F for joint state and * for joint axis or position. Similarly, Articulate-Anything failed on FoldChair and was noted with F. Note that Blade has only a prismatic joint and no revolute joint, which is why there is no values for the Axis Position. Articulate-Anything does not estimate part motions, nor does it perform geometry reconstruction. Instead, it retrieves object meshes from a pre-existing database. Therefore, we do not report Chamfer Distance for this method.

		Multi-Part					
		Storage-41083	Storage-44781	Fridge-10489	Storage-47254	Box-102377	All
Axis Ang 0	PARIS*-m [3]	-	-	34.52	43.26	-	38.89
	DTA [4]	36.29	6.96	8.47	**0.17**	3.03	10.98
	Ours	**0.36**	**0.42**	**2.72**	2.05	**2.25**	**1.56**
Axis Ang 1	PARIS*-m [3]	-	-	15.91	26.18	-	21.04
	DTA [4]	5.50	83.76	8.47	**0.45**	38.74	27.38
	Ours	**0.74**	**1.56**	**0.84**	1.09	**8.32**	**2.51**
Axis Ang 2	PARIS*-m [3]	-	-	-	-	-	-
	DTA [4]	-	4.81	-	-	-	4.81
	Ours	-	**0.75**	-	-	-	**0.75**
Axis Pos 0	PARIS*-m [3]	-	-	3.60	10.42	-	7.01
	DTA [4]	6.22	3.45	5.72	**0.04**	8.57	4.80
	Ours	**0.19**	**0.31**	**0.37**	0.15	**0.12**	**0.23**
Axis Pos 1	PARIS*-m [3]	-	-	1.63	-	-	1.63
	DTA [4]	-	4.84	5.72	-	-	5.28
	Ours	-	**1.21**	**0.00**	-	-	**1.21**
Part Motion 0	PARIS*-m [3]	-	-	86.21	79.84	-	83.03
	DTA [4]	83.36	59.59	7.24	**0.12**	65.91	43.24
	Ours	**0.32**	**1.47**	**3.58**	0.20	**5.29**	**2.17**
Part Motion 1	PARIS*-m [3]	-	-	105.86	0.64	-	53.25
	DTA [4]	**0.05**	93.83	40.51	**0.00**	0.73	27.02
	Ours	0.08	**2.58**	**0.11**	0.06	**0.06**	**0.58**
Part Motion 2	PARIS* [3]	-	-	-	-	-	-
	DTA [4]	-	**0.05**	-	-	-	**0.05**
	Ours	-	0.07	-	-	-	0.07
CD-s	PARIS*-m [3]	-	-	8.52	8.56	-	8.54
	DTA [4]	4.43	4.47	14.99	0.84	5.51	6.05
	Ours	**1.63**	**0.87**	**0.69**	**0.46**	**0.96**	**0.92**
CD-m 0	PARIS*-m [3]	-	-	526.19	128.62	-	327.41
	DTA [4]	64.07	389.95	368.91	0.22	111.49	186.93
	Ours	**1.39**	**0.23**	**0.55**	**0.11**	**1.80**	**0.82**
CD-m 1	PARIS*-m [3]	-	-	160.86	266.71	-	213.79
	DTA [4]	466.28	887.84	285.58	0.22	620.10	452.00
	Ours	**0.19**	**0.31**	**0.61**	**0.14**	**1.53**	**0.56**
CD-m 2	PARIS*-m [3]	-	-	-	-	-	-
	DTA [4]	-	141.56	-	-	-	141.56
	Ours	-	**1.09**	-	-	-	**1.09**
CD-w	PARIS*-m [3]	-	-	15.00	8.66	-	11.83
	DTA [4]	3.57	3.18	0.90	0.98	3.85	2.50
	Ours	**1.19**	**0.93**	**0.50**	**0.57**	**1.32**	**0.90**

Table 2. **Results on the Multi-Part Object dataset.** PARIS*-m [3] is the depth-augmented, multi-part-adapted version of PARIS. Since the code of PARIS*-m is not open-sourced, we show the results from DTA [4] for Fridge-10489 and Storage-47254. Note that Storage-44781 has 3 joints and the other objects have 2 joints. The first joints of Storage-41083, Storage-47254, and Box-102377 are prismatic, so there is no Axis Pos. for them. Our method outperforms the other methods on all metrics for the multi-part object dataset.