

APPENDIX

A METRICS AND EVALUATION DETAILS

We evaluate from three perspectives consistent with the main text: (1) *part segmentation* via 3D IoU on voxelized meshes, (2) *reconstruction quality* via bi-directional Chamfer Distance (mm), and (3) *articulation estimation accuracy* via axis and motion errors.

1. PART SEGMENTATION PERFORMANCE (3D IOU)

For each predicted part and its ground-truth (GT) counterpart, we voxelize both meshes onto a shared binary occupancy grid (identical bounds and voxel size). Let \mathcal{V}^p and \mathcal{V}^g be the sets of occupied voxels. The part-level segmentation score is

$$\text{IoU} = \frac{|\mathcal{V}^p \cap \mathcal{V}^g|}{|\mathcal{V}^p \cup \mathcal{V}^g|}, \quad (8)$$

as in prior work (Nie et al., 2021).

2. RECONSTRUCTION QUALITY (CHAMFER DISTANCE, MM)

We uniformly sample points on the surfaces of the predicted and GT meshes and compute the symmetric (bi-directional) Chamfer Distance in millimeters. For point sets X and Y ,

$$\text{CD}(X, Y) = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} \|x - y\|_2 + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} \|y - x\|_2. \quad (9)$$

We report CD for the whole object (*CD-w*), the static parts (*CD-s*), and the movable parts (*CD-m*).

3. ARTICULATION ESTIMATION ACCURACY

For each dynamic joint we report three metrics:

Axis Ang Err (°). Let $\hat{\mathbf{a}}^p, \hat{\mathbf{a}}^g \in \mathbb{R}^3$ be the unit axis directions (predicted and GT). The angular error (orientation-invariant) is

$$\theta = \min \left\{ \arccos(\hat{\mathbf{a}}^p \cdot \hat{\mathbf{a}}^g), 180^\circ - \arccos(\hat{\mathbf{a}}^p \cdot \hat{\mathbf{a}}^g) \right\}. \quad (10)$$

Axis Pos Err (0.1m). Let the axes be lines $\ell^p(s) = \mathbf{o}^p + s \hat{\mathbf{a}}^p$ and $\ell^g(t) = \mathbf{o}^g + t \hat{\mathbf{a}}^g$. The shortest distance d between two 3D lines is used, and we report it in units of 0.1m by

$$\text{AxisPosErr} = 10 \times d, \quad d = \frac{|(\hat{\mathbf{a}}^p \times \hat{\mathbf{a}}^g) \cdot (\mathbf{o}^p - \mathbf{o}^g)|}{\|\hat{\mathbf{a}}^p \times \hat{\mathbf{a}}^g\|} \quad (\text{for non-parallel axes}), \quad (11)$$

and $d = \|(\mathbf{o}^g - \mathbf{o}^p) \times \hat{\mathbf{a}}^p\|$ for (nearly) parallel axes. This metric is reported for *revolute* joints only.

Part Motion (° or m). Between the start state $t=0$ and end state $t=1$, we measure the state error: (i) *revolute*: geodesic angle on $\text{SO}(3)$ between the predicted and GT relative rotations $\Delta R^p = R_1^p(R_0^p)^\top$ and $\Delta R^g = R_1^g(R_0^g)^\top$,

$$\phi = \arccos \left(\frac{\text{tr}(\Delta R^p (\Delta R^g)^\top) - 1}{2} \right) \cdot \frac{180^\circ}{\pi}; \quad (12)$$

(ii) *prismatic*: Euclidean difference between relative translations $\Delta t^p = t_1^p - t_0^p$ and $\Delta t^g = t_1^g - t_0^g$,

$$s = \|\Delta t^p - \Delta t^g\|_2 \quad (\text{meters}). \quad (13)$$

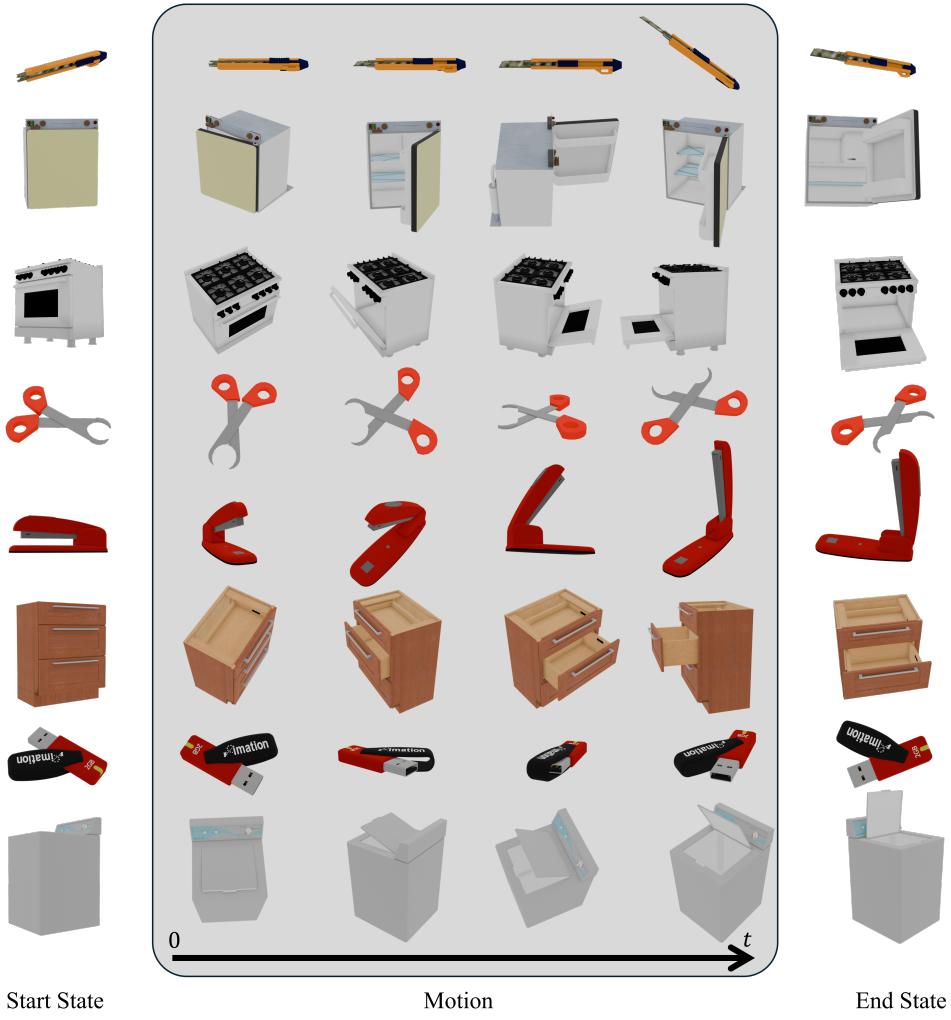


Figure A1: Examples from the Two objects dataset

B DATASETS

B.1 SYNTHETIC DATASET

We build a dataset to evaluate motion segmentation under increasing difficulty. Detailed splits and statistics are reported in Table A1, Table A2, and Table A3. Each scene contains a start (static) state, a continuous motion segment, and an end (static) state. We also provide visual examples from the dataset. Fig. A1 shows a two-object scene with 100 multi-view images for the start state, 200 for the motion segment. Fig. A2 shows three-object scene with the same counts: 100/200. Fig. A3 is a complex object scene with a longer motion segment: 100/500.

Table A1: Motion type and motion records across ten scenes on the Two-part objects dataset

Scene	Blade 103706	Fridge 10905	Oven 101917	Scissor 11100	Stapler 103111	Storage 45135	USB 100109	Washer 103776
Motion type	Translate	Rotate	Rotate	Rotate	Rotate	Translate	Rotate	Rotate
Motion	$0 \rightarrow 0.5$	$-110^\circ \rightarrow 0^\circ$	$0^\circ \rightarrow 90^\circ$	$45^\circ \rightarrow -45^\circ$	$0^\circ \rightarrow -80^\circ$	$0 \rightarrow 0.5$	$0^\circ \rightarrow -90^\circ$	$0^\circ \rightarrow -60^\circ$

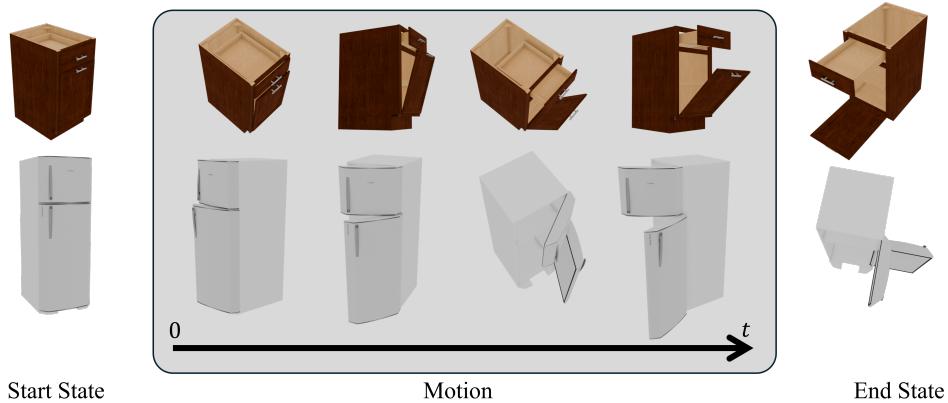


Figure A2: Examples from the Three objects dataset

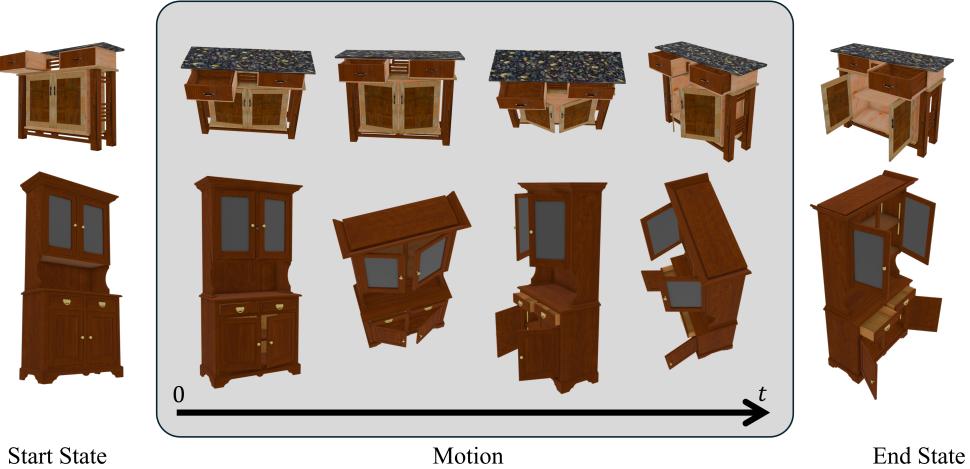


Figure A3: Examples from the Complex objects dataset

Table A2: Motion types and ranges extracted from the two scenes of the Three-part objects dataset.

Scene	Part ID	Motion Type	Range	Part ID	Motion Type	Range
Fridge 11304	0	Rotate	$0 \rightarrow -180^\circ$	1	Rotate	$0^\circ \rightarrow -90^\circ$
Storage 47024	0	Rotate	$0^\circ \rightarrow 90^\circ$	1	Translate	$0 \rightarrow 0.7$

Table A3: Motion types and ranges extracted from two additional scenes of the Complex objects dataset.

Scene	Part ID	Motion Type	Range	Part ID	Motion Type	Range
Storage 47648	0	Rotate	$0 \rightarrow 120^\circ$	1	Rotate	$0 \rightarrow -120^\circ$
	2	Rotate	$0 \rightarrow -60^\circ$	3	Rotate	$0 \rightarrow 60^\circ$
	4	Translate	$0 \rightarrow 0.1$	5	Translate	$0 \rightarrow 0.16$
Table 31249	0	Translate	$0 \rightarrow 0.38$	1	Translate	$0.35 \rightarrow 0$
	3	Rotate	$0 \rightarrow -90^\circ$	4	Rotate	$0 \rightarrow 90^\circ$

Table A4: Comparison results with Video2Articulation (Peng et al., 2025). We report four metrics, *i.e.*, Axis Ang ($^{\circ}$), Axis Pose (0.1m), CD-m (mm) and CD-s (mm). Especially, for two-part objects, we report the metrics as $\text{mean} \pm \text{std}$ across 10 trials. For three-part object, we report the mean value on different moving parts. **Fail** represents that Video2Articulation fails to detect the correct part segmentation masks, and (%) shows the failure rate. **Bold** means the better performance.

	Method	Two-part				Three-part	
		Fridge	Storage	USB	Washer	Fridge (Joint0)	Fridge (Joint1)
Axis Ang ($^{\circ}$)	Video2Articulation	3.80 ± 0.00	6.53 ± 0.00	1.89 ± 0.00	Fail (100%)	2.17	1.35
	Ours	2.70 ± 1.73	1.52 ± 0.88	0.59 ± 0.30	1.63 ± 0.90	1.67	0.68
Axis Pose (0.1m)	Video2Articulation	0.95 ± 0.00	—	0.12 ± 0.00	Fail (100%)	0.71	1.92
	Ours	0.86 ± 0.34	—	1.45 ± 0.71	1.12 ± 0.29	0.68	3.58
CD-m (mm)	Video2Articulation	8.06 ± 0.16	141.95 ± 12.02	24.68 ± 0.49	Fail (100%)	2.88	41.38
	Ours	2.21 ± 0.18	18.95 ± 2.57	0.89 ± 0.10	21.03 ± 1.02	2.12	3.85
CD-s (mm)	Video2Articulation	7.21 ± 0.12	8.66 ± 0.47	101.42 ± 0.72	Fail (100%)	44.45	44.45
	Ours	3.45 ± 0.09	7.09 ± 0.49	1.54 ± 0.14	9.25 ± 0.99	8.16	8.16

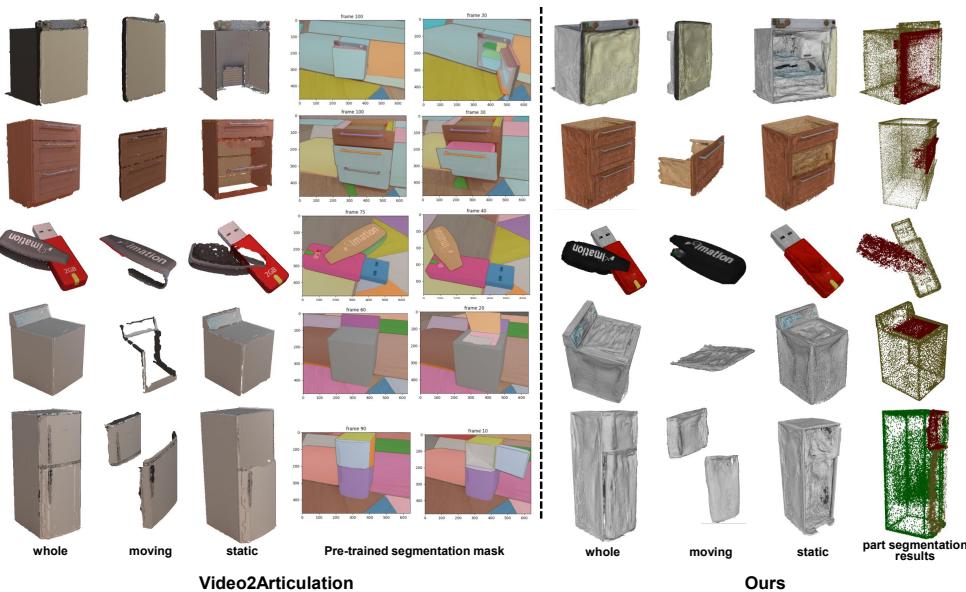


Figure A4: Visual results of Video2Articulation Peng et al. (2025) (Left) and ours (Right) (Notable, as the expensive pre-processing progress of Video2Articulation, we directly use their released dataset to reproduce Video2Articulation, there is some color difference). **Left:** From the visualizations, we observe that the pre-trained-segmentation-driven Video2Articulation method fails to predict correct moving parts when the underlying pre-trained segmentation models cannot provide reliable masks. Typical failure cases include mis-segmenting the drawer and its cabinet, or losing track of the inside of refrigerator door once it opens. **Right:** In contrast, our approach performs part mobility analysis by directly exploiting motion cues from the interaction video. After achieving clean static–dynamic separation, we apply multi-model fitting based on the inferred motion patterns, which leads to consistently more robust and accurate results.

C ADDITIONAL RESULTS

C.1 ADDITIONAL RESULTS ON OUR DATASET

We provide more visual comparisons among DTA, ArtGS and ours. As shown in Fig. A10 and Fig. A11, we compare the rendering quality with ArtGS, and compare the part segmentation performance with DTA and ArtGS. Especially, all the results are with the start state and generated with the estimated motion parameters. From the results, we can see that our method achieves more stable and accurate part mobility analysis. Furthermore, the point clouds in Fig. A11 are presented to show the geometry recovery.

C.2 ADDITIONAL RESULTS COMPARED WITH PRE-TRAINED SEGMENTATION DRIVEN METHOD

As introduced in Sec. 2, our work primarily focuses on self-contained methods, namely approaches that can independently perform part-level mobility analysis without relying on any externally pre-trained segmentation models or segmentation-mask pools. Therefore, we select PARIS, DTA, and ArtGS as our baselines. Furthermore, motivated by the inherent limitations shared by these two-state based methods, we propose AiM, that leverages motion cues from common close-to-open interaction videos to achieve part prior-free mobility analysis without any structural priors.

To further demonstrate the effectiveness of our method beyond the self-contained setting, we additionally report both quantitative and qualitative comparisons against the recent pre-trained segmentation driven approach Video2Articulation. Since Video2Articulation requires preprocessing through Monst3r Zhang et al. (2024) and AutoSeg-SAM Zrporz (2024), we directly use the overlapping subset of objects provided in their released dataset. Specifically, we evaluate on four two-part objects (Fridge-10905, Storage-45135, USB-100109, Washer-103776) and one three-part object (Fridge-11304) and reproduce the official codes using the official settings. These results could be found in Tab. A4 and Fig. A4.

Table A5: Quantitative evaluation of articulation estimation under the open-start and open-end condition. (a) Two-part; (b) Three-part; (c) Complex objects. For complex objects, we report the average of all moving parts. Due to the different magnitudes of part motion for revolute and prismatic joints, we report both of them. – indicates prismatic joints w/o rotation axis.

(a) Two-part objects

Metric	Method	Two-part objects						
		Fridge	Oven	Scissor	USB	Washer	Blade	Storage
Axis Ang	DTA	0.08 \pm 0.03	0.10 \pm 0.04	0.05 \pm 0.02	0.83 \pm 0.49	2.05 \pm 1.20	0.41 \pm 0.11	0.14 \pm 0.07
	ArtGS	0.00 \pm 0.00	0.01 \pm 0.00	0.07 \pm 0.00	0.01 \pm 0.00	0.03 \pm 0.02	0.02 \pm 0.00	0.00 \pm 0.00
	Ours	0.19 \pm 0.08	0.06 \pm 0.03	0.21 \pm 0.01	0.20 \pm 0.06	0.05 \pm 0.02	0.03 \pm 0.01	0.05 \pm 0.02
Axis Pos	DTA	0.01 \pm 0.00	0.06 \pm 0.03	0.01 \pm 0.00	0.03 \pm 0.02	3.05 \pm 4.31	–	–
	ArtGS	0.00 \pm 0.00	0.01 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	–	–
	Ours	0.04 \pm 0.02	0.05 \pm 0.04	0.00 \pm 0.00	0.02 \pm 0.01	0.02 \pm 0.01	–	–
Part Motion	DTA	0.12 \pm 0.04	0.20 \pm 0.09	0.04 \pm 0.02	0.66 \pm 0.38	12.13 \pm 11.07	0.00 \pm 0.00	0.00 \pm 0.00
	ArtGS	0.01 \pm 0.00	0.04 \pm 0.00	0.05 \pm 0.00	0.00 \pm 0.00	0.03 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00
	Ours	0.79 \pm 0.34	0.46 \pm 0.16	0.58 \pm 0.05	1.19 \pm 0.07	0.10 \pm 0.04	0.00 \pm 0.00	0.00 \pm 0.00

(b) Three-part objects

	Methods	D_0		D_1			
		Axis Ang	Axis Pos	Part Motion	Axis Ang		
Storage 47254	DTA	0.09	0.02	0.07	0.32	–	0.00
	ArtGS	0.04	0.00	0.01	0.05	–	0.00
	Ours	0.18	0.05	0.22	0.05	–	0.00
Fridge 10489	DTA	0.26	0.01	0.19	0.18	0.01	0.26
	ArtGS	0.02	0.00	0.01	0.00	0.00	0.05
	Ours	0.09	0.01	0.42	0.06	0.04	0.85

(c) Complex objects

	↓	Ang _{avg}	Pose _{avg}	Motion ^r _{avg}	Motion ^p _{avg}
				Storage 47648	Table 31249
	ArtGS	10.18	0.43	10.91	0.13
	Ours	0.08	0.24	1.62	0.03

C.3 ADDITIONAL RESULTS UNDER THE OPEN-START AND OPEN-END SETTING

As shown in Tab. A5 and Tab. A6, we evaluate all methods under the same input setting used in ArtGS and DTA, namely, the open-start and open-end configuration. Meanwhile we render the

Table A6: **Mesh reconstruction comparison under the open-start and open-end condition.** (a) Two-part objects; (b) Three-part objects; (c) Complex objects. For two-part objects, we report CD distance (mm) as mean \pm std across 5 trials. For three-part and complex objects, we only report mean value, while we report average CD for movable parts. Lower (\downarrow) is better.

(a) Two-part objects								
Metric	Method	Two-part objects						
		Fridge	Oven	Scissor	USB	Washer	Blade	Storage
CD-S	DTA	0.62 \pm 0.02	4.59 \pm 0.13	0.71 \pm 0.51	3.19 \pm 1.07	1.69 \pm 1.10	0.80 \pm 0.10	2.78 \pm 0.04
	ArtGS	0.50 \pm 0.00	4.74 \pm 0.02	0.82 \pm 0.23	2.58 \pm 0.01	0.96 \pm 0.01	0.71 \pm 0.00	4.65 \pm 0.03
	Ours	0.53 \pm 0.00	4.59 \pm 0.18	0.57 \pm 0.00	2.95 \pm 0.13	0.85 \pm 0.01	0.72 \pm 0.00	5.84 \pm 1.60
CD-m	DTA	0.30 \pm 0.01	0.47 \pm 0.01	0.46 \pm 0.13	3.52 \pm 1.92	1.38 \pm 0.65	3.28 \pm 0.33	0.40 \pm 0.00
	ArtGS	0.27 \pm 0.00	0.52 \pm 0.00	0.79 \pm 0.25	2.27 \pm 0.05	0.27 \pm 0.01	2.80 \pm 0.14	1.61 \pm 0.02
	Ours	0.25 \pm 0.00	0.63 \pm 0.06	0.49 \pm 0.00	1.33 \pm 0.04	0.32 \pm 0.01	0.75 \pm 0.01	2.94 \pm 0.31

(b) Three-part objects			(c) Complex objects		
	\downarrow	DTA	ArtGS	Ours	
Storage 47254	CD-s	1.01	0.95	1.58	
	CD-D ₀	0.49	0.25	0.18	
	CD-D ₁	1.11	0.41	0.46	
Fridge 10289	CD-s	2.66	1.97	2.12	
	CD-D ₀	3.56	1.26	1.26	
	CD-D ₁	2.78	0.76	0.69	

sequences using the motion parameters provided by ArtGS. Under this setting, both ArtGS and DTA achieve good articulation estimation, particularly because the geometric correspondence between the two states is clear and the part number is known. This also makes the articulation estimation of ArtGS stable, with almost zero variance on the evaluation of articulation estimation. Meanwhile, without structural priors as input, our AiM also achieves accurate and stable articulation estimation through dual-Gaussian-based dynamic–static disentanglement and robust motion-based Sequential RANSAC. As shown in Tab. A5, Tab. A6, our results are comparable to these optimization-based baselines, and in challenging complex-object cases, AiM is more stable. Meanwhile, AiM’s part segmentation, driven only by motion cues in interaction videos and without requiring the number of parts, remains on par with or even surpasses recent state-of-the-art ArtGS.



Figure A5: The Meta Arria glass.

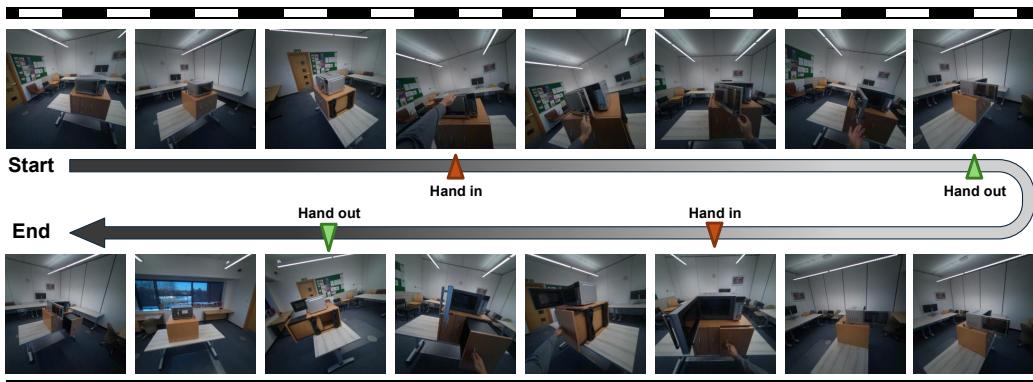


Figure A6: The captured interaction videos. In particular, we use the hand as an indicator to automatically detect the motion start (*hand in*) and end times (*hand out*), enabling a fully automated data-processing pipeline.

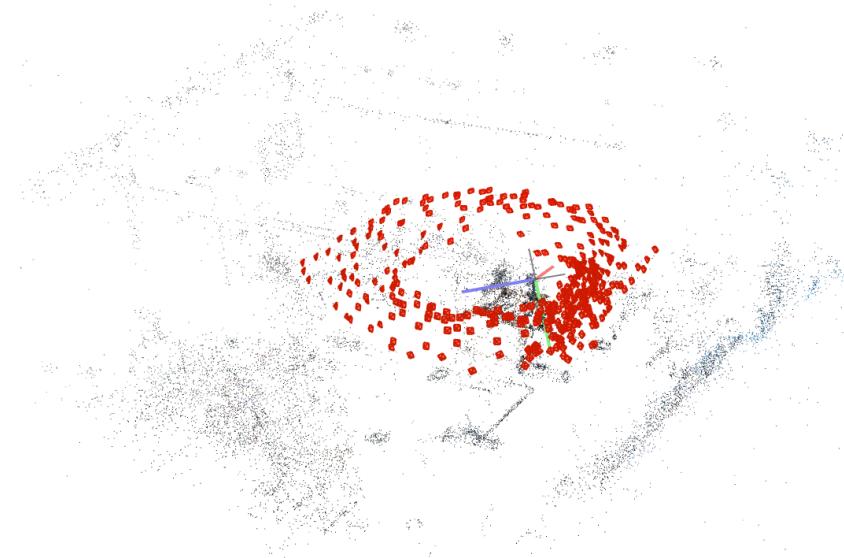


Figure A7: The estimated pose of real-world video via using COLMAP Schonberger & Frahm (2016).

C.4 ADDITIONAL RESULTS ON REAL-WORLD DATA

C.4.1 REAL-WORLD DATASET ACQUISITION

To better support natural human–object interaction during data capture, we leverage the Meta Aria glass (as shown in Fig. A5) to record the interaction video. In details, videos are recorded in real time using the device’s built-in fisheye cameras while the user observes the target object and manipulates its movable parts. As shown in the video, during the interaction process, the user first walks into the scene and observes both the surrounding environment and the articulated objects in their closed-start state. To achieve the automatic pipeline, the user then signals the beginning of interaction by using the hand to touch the target object (*i.e.*, the oven) with the hand. While manipulating the movable part, the user freely moves the head to observe the object from different viewpoints. The hand is then removed to inspect the object again. The user subsequently touches another object (*i.e.*, the storage) and repeats the same manipulation-and-observation procedure.

For data processing, we first extract the recordings from the Aria Glass. Since the RGB cameras on Aria Glass are fisheye camera, we apply the official Project Aria toolkit (Engel et al., 2023). to rectify each frame and re-project it into a pinhole camera mode. This produces a set of undistorted pinhole frames along with their corresponding timestamps (See Fig. A6). We then extract keyframes with FFmpeg and manually filter the frames to remove blurred or low-quality images. Finally, as shown in Fig. A7, the curated images are fed into COLMAP (Schonberger & Frahm, 2016) to estimate camera poses via structure-from-motion, which are used as inputs for subsequent reconstruction. Especially, we follow the process, introduced in video2articulation Peng et al. (2025), to obtain the whole articulated object (*i.e.*, oven and storage) via Grounded SAM2 Ravi et al. (2024); Ren et al. (2024) with the two text prompts: “silver microwave” and “wooden storage”. The final inputs to AiM consist of **100 start-state frames and 87 motion frames** for the oven sequence, and **77 start-state frames and 58 motion frames** for the storage sequence. Although the number of motion frames is much smaller than in our rendered dataset, the following results show that AiM still performs robustly and accurately under this limited motion input.

C.4.2 EXPERIMENTAL RESULTS ON REAL-WORLD DATA

As shown in Fig. A8 and Fig. A9, we present our results on the real-world captured sequences. AiM performs robust and accurate part mobility analysis purely from RGB inputs, without any structural prior knowledge. Notably, AiM reliably predicts the correct joint-axis direction and achieves low-error articulation estimation, such as the oven door’s nearly 85° opening motion is predicted as about

82°. Moreover, our SDMD module correctly reassigned newly revealed static regions during motion, such as the oven interior. For the storage example, despite significant occlusion from the oven and the user’s hand, AiM still produces correct part-level segmentation and articulation estimation. These results further confirm the strong motion analysis ability of dual-Gaussian representation and generalization capability of AiM in challenging real-world scenarios.

Limitations: From the real-world data, we observe that when motion introduces structural ambiguities—particularly those caused by specular reflections, such as the glass door of oven in the video—our vanilla 3DGS-based reconstruction in AiM can be affected. In future work, we plan to further address such challenges, for example by incorporating depth information to improve robustness under complex lighting and reflective surfaces. Morevoer, as discussed in Sec. 5, we will extend AiM to more diverse and larger real-world scenes in future work.

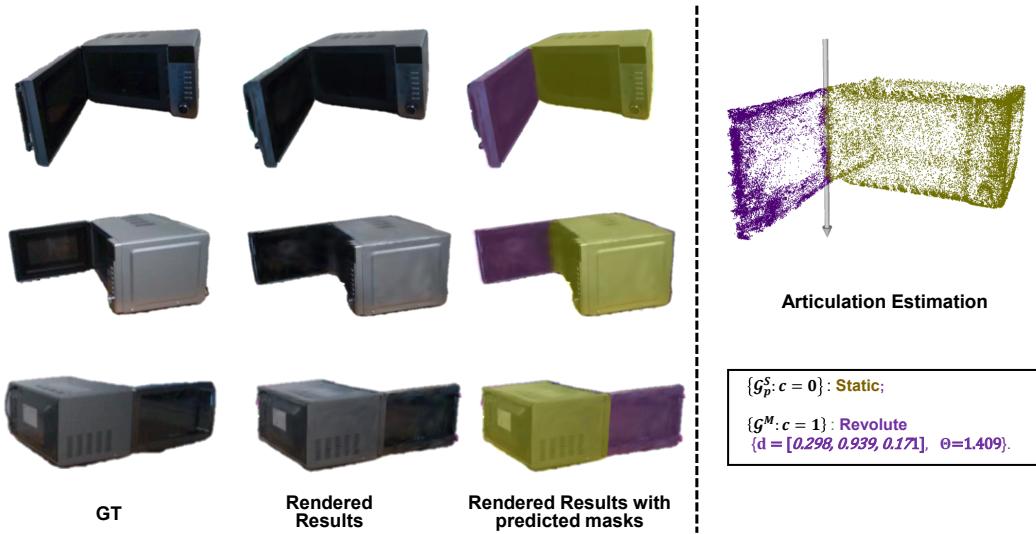


Figure A8: Qualitative results of our AiM on the real-world data of oven. **Left:** Comparison between the ground-truth views and rendered views. **Besides, we provide the rendered masks based on our dual-Gaussian representation (via directly changing the spherical harmonics of Gaussians).** Due to the strong specular reflections on the oven’s glass door, the appearance of the moving part undergoes frequent and significant changes during interaction. Despite this challenge, our dual-Gaussian representation still achieves clean dynamic–static disentanglement by relying on stable motion cues. Moreover, the SDMD module reliably reassigned the newly revealed static interior regions back to the static base as the motion unfolds, further improving the quality of disentanglement and reconstruction. **Right:** Our prior-free part mobility analysis. Based on our dual-Gaussian representation, we can easily infer the trajectories of moving Gaussians, and obtain the articulation parameters based on the optimization-free and robust sequential RANSAC without any prior structural knowledge.

D ABLATION STUDY

In Fig. A12, we present qualitative ablations by removing individual modules, highlighting the importance of each component.

E PRELIMINARY

3D Gaussian splatting 3DGS represents the scene as an explicit point-based 3D structure, enabling orders of magnitude faster reconstruction and rendering. In this work, we build on 3DGS to reconstruct the articulated objects with the part-level structures. In details, each 3D Gaussian \mathcal{G}_i is defined by a center position $\mu_i \in \mathbb{R}^3$, opacity $\sigma_i \in \mathbb{R}$, a covariance matrix Σ_i parameterized by a 3D scale $s_i \in \mathbb{R}^3$ and a rotation $r_i \in \mathbb{R}^4$, as well as spherical harmonics (SH) coefficients h_i for view-dependent color modeling (Kerbl et al., 2023). Given image captures of the scene, we optimize

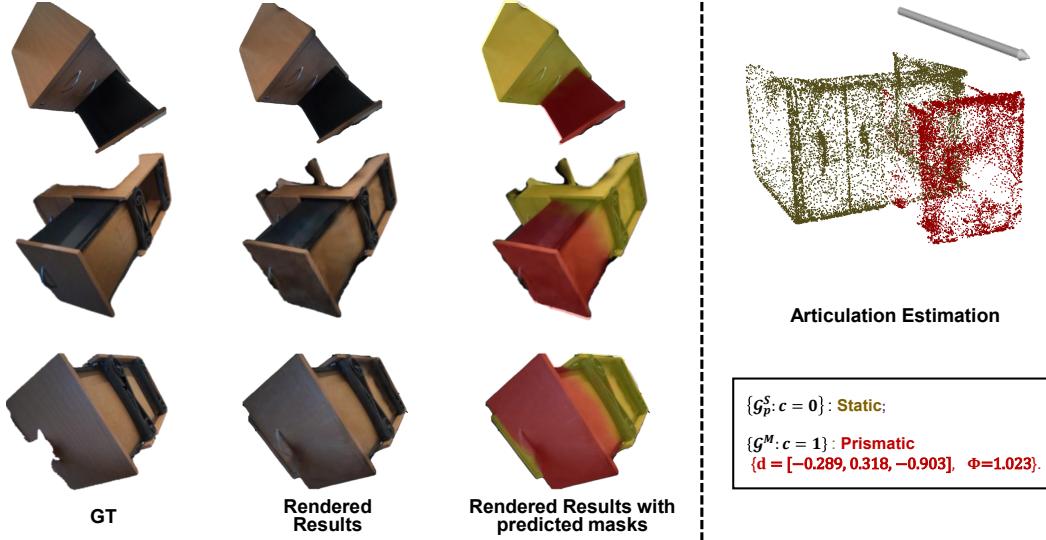


Figure A9: Qualitative results of our AiM on the real-world data of storage. **Left:** Comparison between the ground-truth views and rendered views. Besides, we provide the rendered masks based on our dual-Gaussian representation (via directly changing the spherical harmonics of Gaussians). Although large regions of the storage are occluded by the oven and the hand during the interaction video (See Fig. A6), AiM remains robust and accurately performs dynamic–static disentanglement, enabling clean separation of the static base and the moving part purely from motion cues. **Right:** Our prior-free part mobility analysis. Based on our dual-Gaussian representation, we can easily infer the trajectories of moving Gaussians, and obtain the articulation parameters based on the optimization-free and robust sequential RANSAC without any prior structural knowledge. (Notably, for prismatic joint, we do not consider the axis pose).

a collection of Gaussians $\{\mathcal{G}\} = \{\mathcal{G}_i\}_{i=1}^N$ via blending-based differentiable rendering:

$$\mathcal{I} = \sum_{i=1}^N \mathbf{c}_i T_i \alpha_i^{2D}, \quad \alpha_i^{2D}(u) = \sigma_i \exp\left(-\frac{1}{2}(u - \mu_i^{2D})^T \Sigma_i^{2D-1} (u - \mu_i^{2D})\right), \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j^{2D}), \quad (14)$$

where μ_i^{2D} and Σ_i^{2D} denote the 2D projections of the 3D center μ_i and covariance matrix Σ_i , respectively; u represents the pixel coordinate; and \mathbf{c}_i is the color of \mathcal{G}_i , determined by the SH coefficients h_i and the view direction. T_i represents the transmittance from the start of rendering to \mathcal{G}_i .

Deformable 3D Gaussian splatting. (*Original 3DGS Kerbl et al. (2023) is introduced in Appendix 5.*) To model the time-varying changes of geometry and appearance in a dynamic scene, Deformable-3DGS (Yang et al., 2024) introduces a learnable deformation field to model temporal transformations in the center positions μ , rotations r , and scales s of 3D Gaussians. This deformation field is parameterized by a multi-layer perceptron (MLP), which predicts offsets based on time and the canonical Gaussian \mathcal{G}_c . Specifically, at time step t , the deformed Gaussian \mathcal{G}_d is defined as:

$$\mathcal{G}_d(\mu_i, s_i, r_i, \sigma_i, h_i) = \mathcal{G}_c(\mu_i + \delta\mu, s_i + \delta s, r_i + \delta r, \sigma_i, h_i), \quad (15)$$

where the offsets $\delta\mu, \delta s, \delta r$ are given by $F_\theta(\gamma(t), \gamma(\mu_i))$, with F_θ representing the deformation field and γ denoting the positional encoding function. With differentiable rendering, both the Gaussian parameters and the deformation network parameters are jointly optimized. Here, we only consider the time-varying transformation of position $\delta\mu$ and rotation δr .

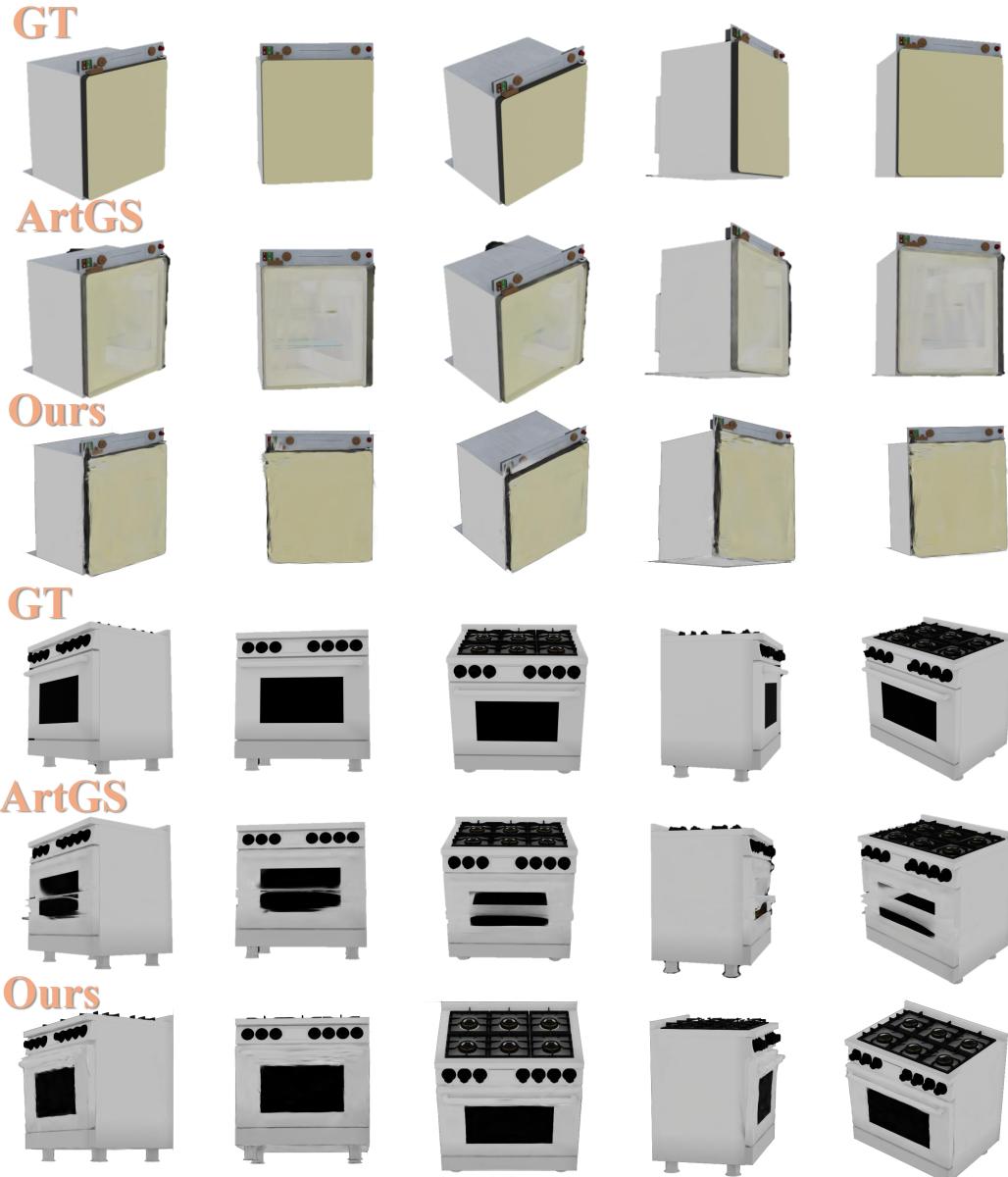


Figure A10: Rendering results based on articulation estimation parameters for start state of two-part objects: Fridge (Top) and *Oven* (Bottom). **Top:** From the visualization, it can be observed that the newly seen interior content severely influences the performance of ArtGS’s articulation estimation, causing the door located inside the body. **Bottom:** Similarly, during opening the oven, the newly seen content could not be well aligned between two states, leading the wrong axis estimation and joint type estimation. The handle moves into the oven.

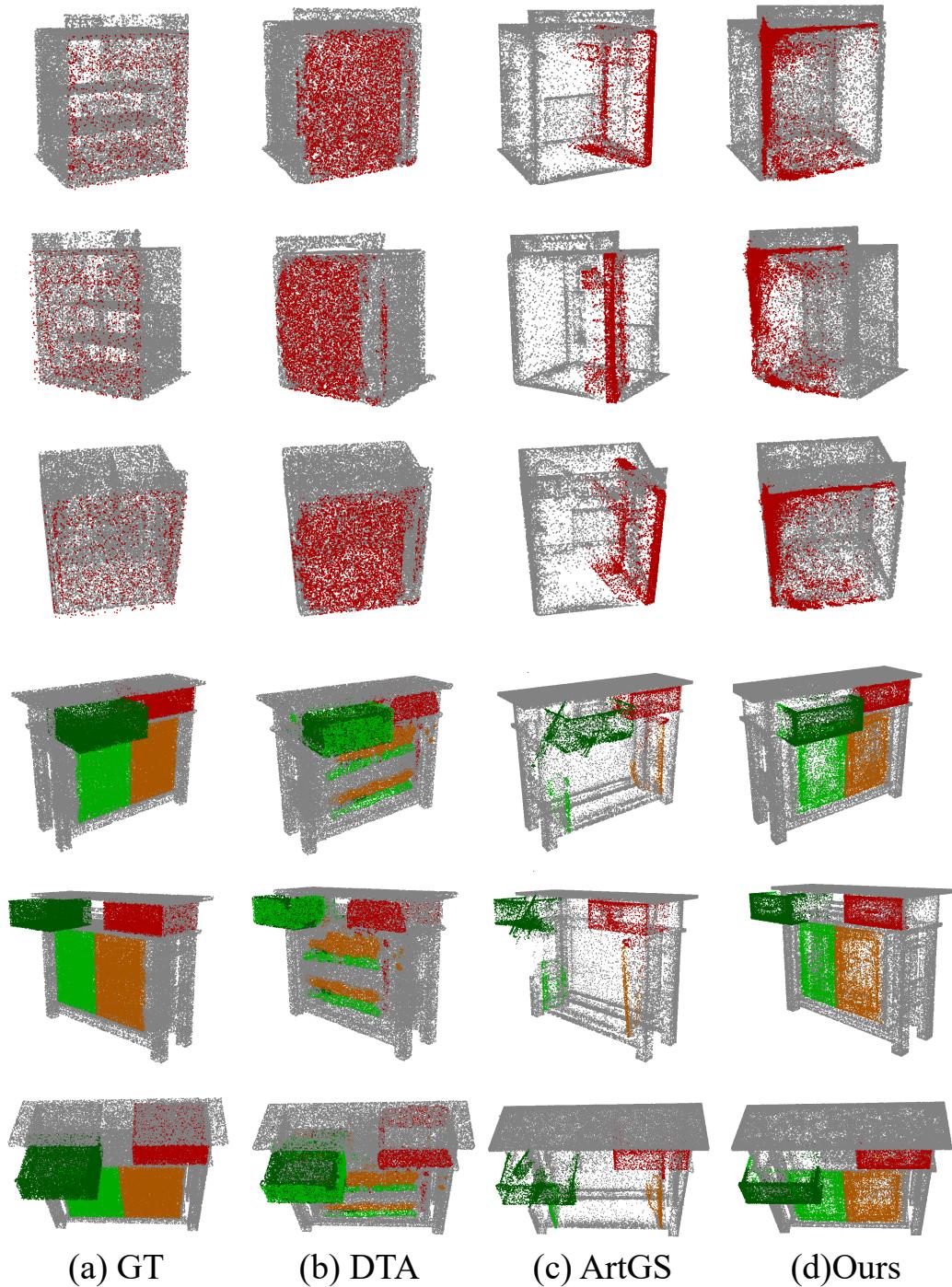


Figure A11: Qualitative comparison among DTA, ArtGS and ours.

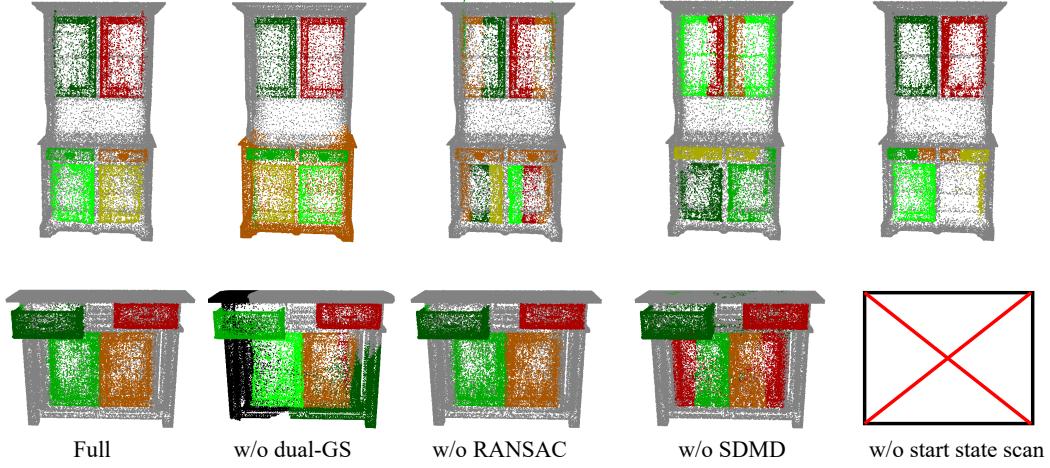


Figure A12: Qualitative Comparisons for ablation studies.



Figure A13: Rendering of our dual-Gaussian representation (**left**: static result $\{\mathcal{R}^S\}$, **right**: moving result $\{\mathcal{R}^{M,t=1}\}$). All objects ($category_{instance}$) are from PartNet-Mobility dataset (Mo et al., 2019).

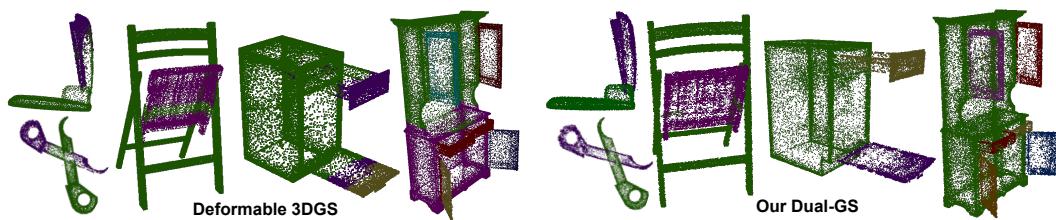


Figure A14: Part segmentation results based on Deformable 3DGS and our dual-Gaussian representation. We cluster Gaussians using the same sequential RANSAC settings; colours denote groups. Static noise in Deformable 3DGS noticeably degrades the segmentation.