ArtPro: Self-Supervised Articulated Object Reconstruction with Adaptive Integration of Mobility Proposals

Supplementary Material

In this supplementary material, we first present the loss term formulations of Gaussian primitive optimization in Section A. Then, we present more two-part comparisons in Section B. In Section C and Section D, we provide more multi-part visualization and full initialization of our method on multi-part dataset. In Section E and Section F, we report additional ablation studies and computational costs. Finally, in Section G, we report the failure cases of our method.

A. Loss Terms of Gaussian Optimization

Below we present the detailed formulation of the loss terms used in the Gaussian primitive optimization, which are mentioned in Section 3.3 and 3.4 in the main paper.

The RGBD loss $\mathcal{L}_I$ and the one-sided Chamfer Distance loss $\mathcal{L}_{cd}$ encourage the transformed Gaussians to align with the underlying surface of the end state. We have

$$\begin{aligned} \mathcal{L}_D &= \log(1+ \parallel D_v(\mathcal{T}(\mathcal{G})) - \mathcal{D}_v \parallel_1) \\ \mathcal{L}_I &= \mathcal{L}_{3dgs}(\mathcal{T}(\mathcal{G})) + \mathcal{L}_D(\mathcal{T}(\mathcal{G})), \\ \mathcal{L}_{cd} &= \mathrm{CD}(\cup\{x_m\} \rightarrow \mathcal{P}^1) \end{aligned} \tag{1}$$

where $D_v(\cdot)$ denotes the rendered depth map of view v, $\mathcal{L}_{3dgs}$ is the RGB loss used in 3DGS [2], $\mathcal{G}$ is the Gaussian sets, $\{x_m\}$ is the Gaussian centers of all movable $\{\mathcal{G}_m\}$, and $\mathcal{P}^1$ is the point cloud of end state. We compute the loss $\mathcal{L}_I$ between the rendered images of transformed Gaussian $\mathcal{T}(\mathcal{G})$ and GT RGBD of end state to optimize the appearance, and use $\mathcal{L}_{cd}$ to further improve the geometric quality.

The part contrastive loss $\mathcal{L}_{pc}$ ensures each Gaussian be dominated by one distinctive part

$$\mathcal{L}_{pc} = \mathbb{E}_{\{x_i \in \mathcal{G}_m\}^M}\Big[\frac{1}{M-1}\sum_{\substack{k=1 \\ k\neq m}}^{M} P_k(x_i)\Big], \tag{2}$$

where, $\mathcal{G}_m$ denotes the m-th movable part of $\{\mathcal{G}_m\}_{m=1}^M$, $P_k(x_i)$ is the probability that the point x_i belongs to the k-th movable part.

The local smoothness loss $\mathcal{L}_{ls}$ aims to maintain consistent static part probability for the spatial neighbors $x_i, x_j \in \mathcal{G}$. It is formulated as

$$\mathcal{L}_{ls} = \mathbb{E}_{x_i \in \mathcal{G},\, x_j \in \mathrm{kNN}(x_i, \mathcal{G})} |P_s(x_i) - P_s(x_j)|, \tag{3}$$

where kNN denote k-nearest neighbors with $k = 20$. To avoid fragmented parts, we smooth the static probability field on the kNN-based neighborhood graph to obtain a locally consistent segmentation field.

The regularization term aims to encourage the compactness of the parts. It uses the mean $\{\hat{\mu}_m\}$ and variance scaling $\{\hat{s}_m\}$ of initialization $\{\hat{\mathcal{P}}_m\}$ to maintain the existence of p_m

$$\mathcal{L}_{reg} = \frac{1}{M}\sum_{m=1}^{M}\Big[\lambda_\mu |\mu_m - \hat{\mu}_m| + \lambda_s \parallel s_m - \hat{s}_m \parallel_1\Big] \tag{4}$$

where, s_m is the scaling of variance Σ_m, and $\lambda_\mu = 0.5, \lambda_s = 0.1$ is loss weight.

In summary, the objective function of Gaussian primitive optimization in the adaptive proposal integration stage is:

$$\mathcal{L} = \mathcal{L}_I + \lambda_{cd}\mathcal{L}_{cd} + \lambda_{pc}\mathcal{L}_{pc} + \lambda_{ls}\mathcal{L}_{ls} + \lambda_{reg}\mathcal{L}_{reg}. \tag{5}$$

We use $\lambda_{cd} = 0.5, \lambda_{pc} = 0.1, \lambda_{ls} = 0.02$ and $\lambda_{reg} = 1.0$ in all the experiments presented in this paper.

We further define a collision loss $\mathcal{L}_{col}$, which avoids the collisions between the movable and static parts of the transformed object. It is formulated as

$$\begin{aligned} L_{col}(\mathcal{G}_i, \mathcal{G}_j) &= \mathbb{E}_{x \in \mathcal{G}_i, y \in \mathrm{kNN}(x, \mathcal{G}_j)} |\hat{\alpha}_y|, \\ \mathcal{L}_{col} &= L_{col}(\mathcal{T}(\mathcal{G}_m), \mathcal{G}_s) + L_{col}(\mathcal{G}_s, \mathcal{T}(\mathcal{G}_m)), \end{aligned} \tag{6}$$

where $\mathcal{T}(\mathcal{G}_m)$ is the union of all movable parts of $\mathcal{T}(\mathcal{G})$, $\mathcal{G}_s$ is the static part of $\mathcal{T}(\mathcal{G})$, $\hat{\alpha}_y$ is the Gaussian opacity corresponding to center position y, and kNN is the set of k nearest neighbors with $k = 32$.

Therefore, the objective function of the post-processing refinement optimization is

$$\mathcal{L} = \mathcal{L}_I(\mathcal{G}) + \mathcal{L}_I(\mathcal{T}(\mathcal{G})) + \lambda_c\mathcal{L}_{col} \tag{7}$$

We use $\lambda_c = 0.02$ in all the experiments presented in this paper.

B. Additional Results on Two-Part Dataset

We compare with related methods including PARIS [3], ArticulatedGS [1], DTA [7], ArtGS [5] on the two-part object dataset. Since we take RGBD images as input, we apply the depth loss to PARIS [3] and ArticulatedGS [1] methods for a fair comparison. We report the quantitative evaluation results in Table 1, showing that the 3DGS-based methods, i.e. ArticulatedGS [1], ArtGS [5], and ours, achieve competitive performance. This is because static and movable parts can be well initialized with spatial clustering for two-part objects, which are further refined with the following deformable 3DGS optimization. This experiment also acts

Table 1. Results on the PARIS [3] dataset, including both synthetic and real data. Methods marked with an asterisk (*) denote versions with added depth supervision for fair comparison. Specifically, ArticulatedGS* and PARIS* are trained with an additional depth loss.

		Synthetic Objects											Real Objects		
		FoldChair	Fridge	Laptop	Oven	Scissor	Stapler	USB	Washer	Blade	Storage	All	Fridge	Storage	All
Axis Ang	PARIS* [3]	15.79	2.93	0.03	7.43	16.62	8.17	0.71	0.71	41.28	0.03	9.37	**1.90**	30.10	16.00
	ArticulatedGS [1]	6.20	0.14	15.55	**0.00**	*0.07*	0.08	0.16	0.03	0.23	0.04	2.25	4.06	48.56	26.31
	ArticulatedGS*	*0.02*	0.12	0.03	0.13	0.18	0.13	0.12	0.17	*0.04*	0.04	0.10	64.81	29.44	47.12
	DTA [7]	0.03	*0.09*	0.07	0.22	0.10	0.07	*0.11*	0.36	0.20	0.09	0.13	2.08	13.64	7.86
	ArtGS [5]	**0.01**	**0.03**	*0.01*	*0.01*	**0.05**	**0.01**	**0.04**	*0.02*	**0.03**	**0.01**	**0.02**	2.09	*3.47*	*2.78*
	Ours	0.04	**0.03**	**0.00**	*0.01*	**0.05**	*0.04*	**0.04**	**0.01**	0.07	*0.02*	*0.03*	*1.92*	**1.02**	**1.47**
Axis Pos	PARIS* [3]	0.25	1.13	**0.00**	0.05	1.59	4.67	3.35	3.28	-	-	1.79	0.50	-	0.50
	ArticulatedGS [1]	4.93	**0.00**	0.16	0.03	**0.00**	0.02	0.63	**0.00**	-	-	0.72	1.71	-	1.71
	ArticulatedGS*	**0.00**	0.02	**0.00**	0.02	**0.00**	0.02	*0.64*	*0.02*	-	-	0.09	2.71	-	2.71
	DTA [7]	*0.01*	*0.01*	*0.01*	*0.01*	*0.02*	0.02	**0.00**	0.05	-	-	*0.02*	0.59	-	0.59
	ArtGS [5]	**0.00**	**0.00**	*0.01*	**0.00**	**0.00**	*0.01*	**0.00**	**0.00**	-	-	**0.00**	*0.47*	-	*0.47*
	Ours	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	**0.00**	**0.34**	-	**0.34**
Part Motion	PARIS* [3]	127.34	45.26	0.03	9.13	68.36	107.76	96.93	49.77	0.36	*0.30*	50.52	**1.58**	0.57	1.08
	ArticulatedGS [1]	59.08	0.55	28.16	0.13	*0.05*	0.07	0.27	**0.00**	*0.04*	**0.00**	8.84	15.11	0.53	7.82
	ArticulatedGS*	0.26	0.12	0.07	0.31	0.13	0.09	0.10	0.09	**0.00**	**0.00**	0.12	39.58	0.32	19.95
	DTA [7]	0.10	0.12	0.11	0.12	0.37	0.08	0.15	0.28	**0.00**	**0.00**	0.13	*1.85*	*0.14*	*1.00*
	ArtGS [5]	**0.03**	*0.04*	*0.02*	**0.02**	**0.04**	**0.01**	**0.03**	*0.03*	**0.00**	**0.00**	**0.02**	1.94	**0.04**	**0.99**
	Ours	*0.06*	**0.03**	**0.00**	*0.03*	**0.04**	*0.05*	*0.04*	0.04	**0.00**	**0.00**	*0.03*	2.76	**0.04**	1.40
CD-s	PARIS* [3]	10.20	8.82	*0.16*	3.18	15.58	2.48	1.95	12.19	1.40	8.67	6.46	11.64	20.25	15.95
	ArticulatedGS [1]	3.17	2.02	4.27	2.31	*0.37*	1.84	1.88	5.17	0.44	2.74	2.42	36.05	75.82	55.94
	ArticulatedGS*	0.58	1.09	1.72	*1.88*	0.64	*1.45*	*1.24*	*3.58*	*0.24*	*1.93*	*1.44*	230.04	46.78	138.41
	DTA [7]	**0.18**	0.62	0.30	4.60	3.55	2.91	2.32	4.56	0.55	4.90	2.45	2.36	10.98	6.67
	ArtGS [5]	*0.26*	*0.52*	0.63	3.88	0.61	3.83	2.25	6.43	0.54	7.31	2.63	**1.64**	*2.93*	*2.29*
	Ours	0.48	**0.31**	**0.12**	**1.41**	**0.19**	**0.67**	**0.96**	**2.69**	**0.21**	**1.81**	**0.89**	*1.96*	**2.54**	**2.25**
CD-m	PARIS* [3]	17.97	7.23	0.15	6.54	16.65	30.46	10.17	265.27	117.99	52.34	52.48	77.85	474.57	276.21
	ArticulatedGS [1]	36.61	2.29	24.90	0.96	*0.35*	1.64	*1.02*	3.88	1.86	5.49	7.90	107.96	2,459.45	1,283.71
	ArticulatedGS*	0.33	0.63	2.07	1.08	0.57	1.97	1.17	*0.41*	**0.78**	0.83	*0.98*	69.23	1,578.02	823.63
	DTA [7]	*0.15*	*0.27*	*0.13*	*0.44*	10.11	1.13	1.47	0.45	2.05	*0.36*	1.66	*1.12*	*30.78*	*15.95*
	ArtGS [5]	0.54	**0.21**	*0.13*	0.89	0.64	**0.52**	1.22	0.45	*1.12*	1.02	**0.67**	**0.66**	**6.28**	**3.47**
	Ours	**0.11**	0.28	**0.07**	**0.33**	**0.18**	*0.83*	**0.37**	**0.08**	F	**0.35**	0.28 [F]	28.31	35.95	32.13
CD-w	PARIS* [3]	4.37	5.53	*0.26*	3.18	3.90	5.27	1.78	10.11	0.58	7.80	4.28	8.99	32.10	20.55
	ArticulatedGS [1]	1.06	2.12	8.52	2.13	*0.35*	1.61	1.88	4.79	0.23	2.62	2.53	77.53	995.99	536.76
	ArticulatedGS*	0.43	0.96	0.80	*1.60*	0.59	*1.48*	*1.00*	*3.08*	*0.20*	*1.75*	*1.19*	51.22	965.80	508.51
	DTA [7]	*0.27*	0.70	0.32	4.24	0.41	1.92	1.17	4.48	0.36	3.99	1.79	2.08	8.98	5.53
	ArtGS [5]	0.43	*0.58*	0.50	3.58	0.67	2.63	1.28	5.99	0.61	5.21	2.15	*1.29*	*3.23*	*2.26*
	Ours	**0.10**	**0.31**	**0.09**	**1.32**	**0.18**	**0.62**	**0.50**	**2.40**	**0.18**	**1.60**	**0.73**	**1.03**	**2.03**	**1.53**

as a sanity check for our method, validating that the over-segmentation proposals from the same movable part can be effectively merged based on their motion similarities.

Figure 1 provides the full visualization results of ArtGS and Ours on this dataset. The results demonstrate that our method can robustly integrate the proposals and reconstruct both the part probability field and motion parameters. For incorrectly initialized mobility proposals, our optimization can also obtain correct motion estimation and the final reconstructions. For the real-world objects in this dataset, our method's performance on the geometry reconstruction, especially in terms of CD metric, is limited by the quality of input depth maps. This is because our method tends to estimate the mobility parameters that align the transformed results closer to the depth map of end state ($t = 1$). Large discrepancies between the input depth maps of states $t = 0$ and $t = 1$ lead to our results containing a small number of Gaussians originating from static part, such as the Real Storage in Figure 1 and Figure 6.

C. Additional Results on Multi-Part Dataset

We present a comprehensive visual analysis to compare ArtGS [5] and our method in Figure 2 and Figure 4, which show the reconstructed articulated objects at different motion states and their articulated joints of movable parts. Compared to ArtGS, our method can robustly estimate the movable parts and motion parameters, leading to consistent and clean rendered images of the reconstructed objects.

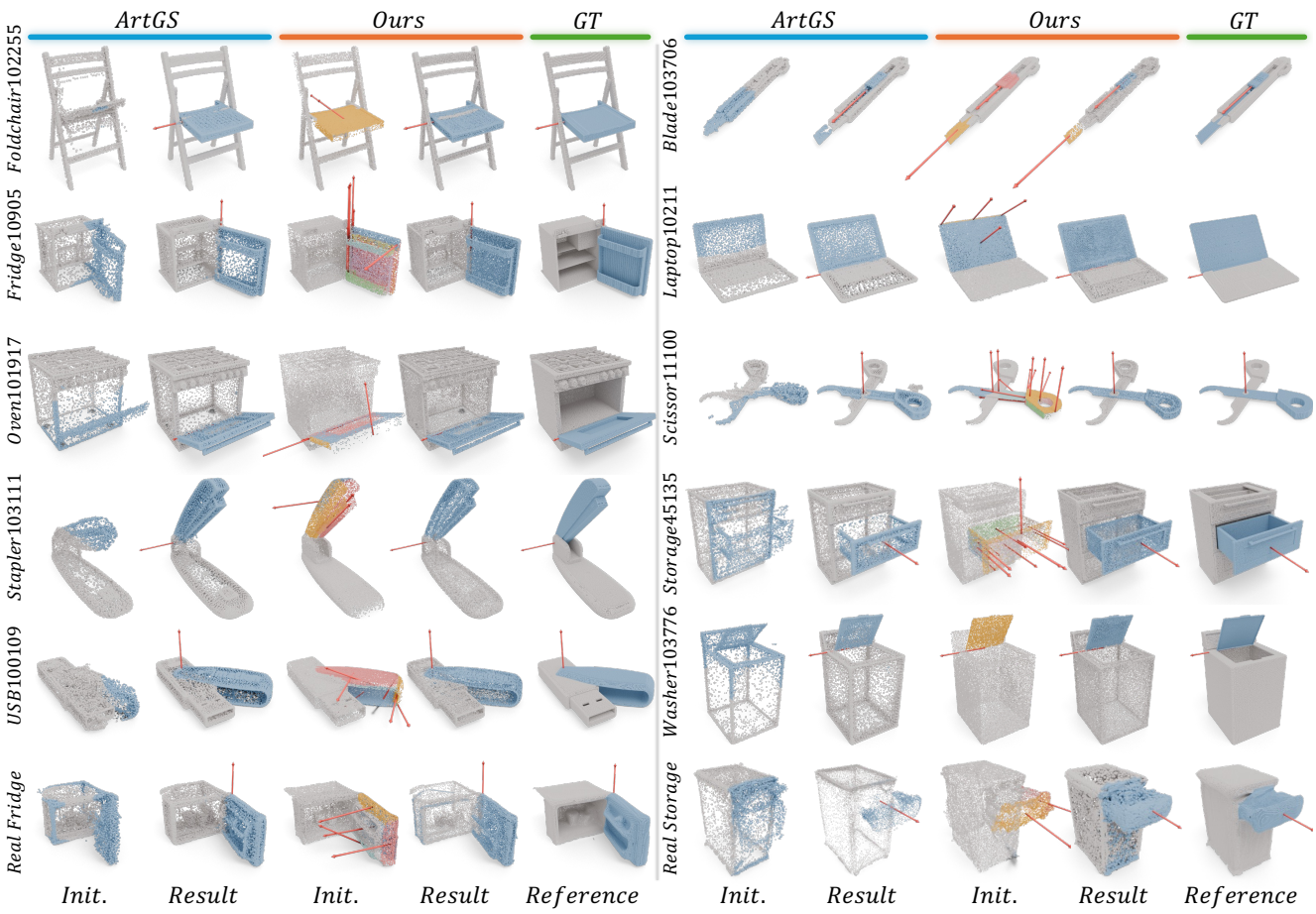

Figure 1. Additional qualitative results on PARIS dataset, including the initializations and the final reconstructions. We show the Gaussians with their center points for a better visualization of their segmentation and motion parameters.

D. More Visual Analysis on Multi-Part Dataset

We further show the visualizations of the initializations and reconstructed results of ArtGS and ours in Figure 3 and Figure 5. The results show that our mobility initialization can effectively extract the potential movable parts and reasonable motion initialization, while ArtGS suffers from unstable initialization. For over-segmented proposals, our optimization can effectively estimate their motion parameters and integrate adjacent proposals into an individual part.

We further report the visual quality metrics on our dataset in Table 2. Our method achieves better PSNR, SSIM, and LPIPS scores compared to ArtGS, while producing more accurate articulation.

Table 2. Visual quality on our dataset.

Method	PSNR	SSIM	LPIPS
ArtGS	36.10	0.980	0.037
Ours	**48.02**	**0.998**	**0.003**

Table 3. Additional ablation study on our dataset.

Case	CD-s	$\overline{\text{CD}}$-m	CD-w
w/o PM	0.85	4.26	0.68
w/o MoI	0.89	5.62	0.70
w/o MoI&PI	0.87	5.81	0.71
Full	**0.84**	**3.63**	**0.65**

E. More Ablation Study Results

To strengthen the validation of careful initialization, we report the results of "w/o MoI" (no principal axes for initialization) and "w/o MoI&PI" (no principal axes for initialization and pruning) in Table 3. We also add the results of disabling the progressive merging after over-segmentation (denoted as w/o PM) in Table 3.

F. Computation Overhead

We control the computational cost of ArtPro through two key strategies. First, we use PartField [4] to produce well-structured over-segmentation proposals rather than frag-

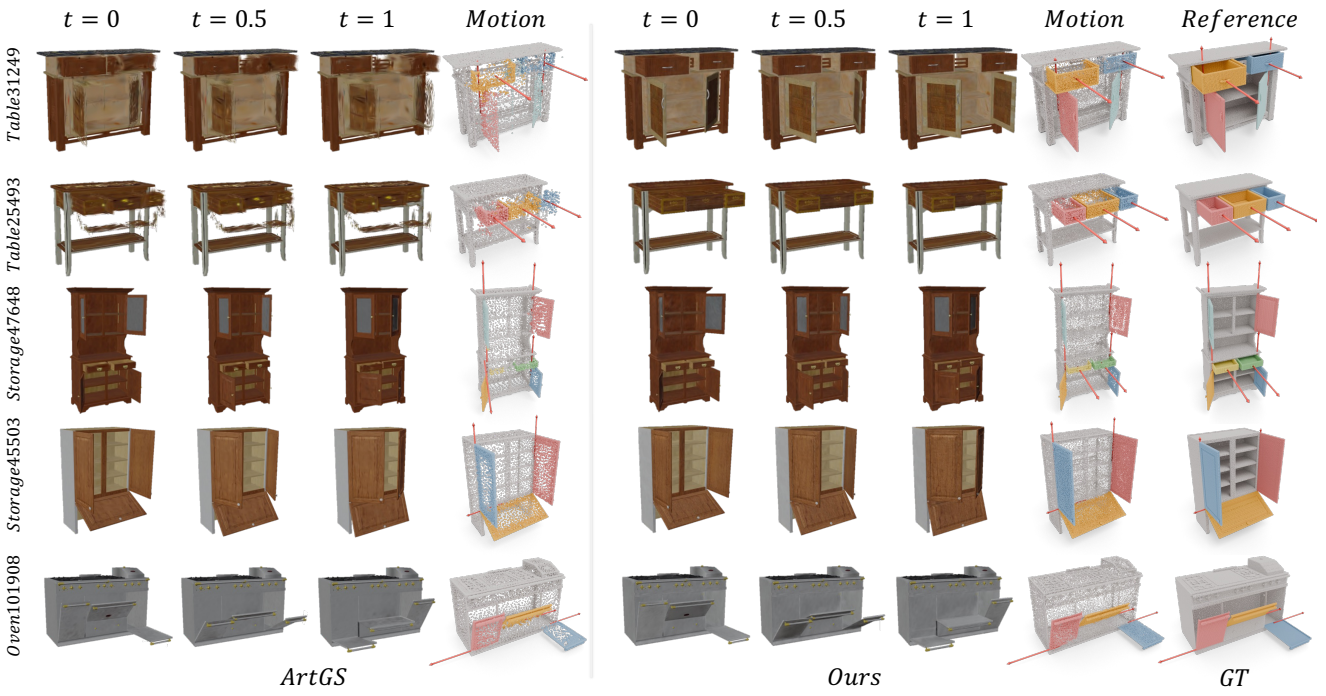

Figure 2. Additional reconstruction results on **ArtGS-Multi dataset**. We render the reconstructed articulated objects in different motion states ($t \in \{0, 0.5, 1\}$ and their motion structures for each result.

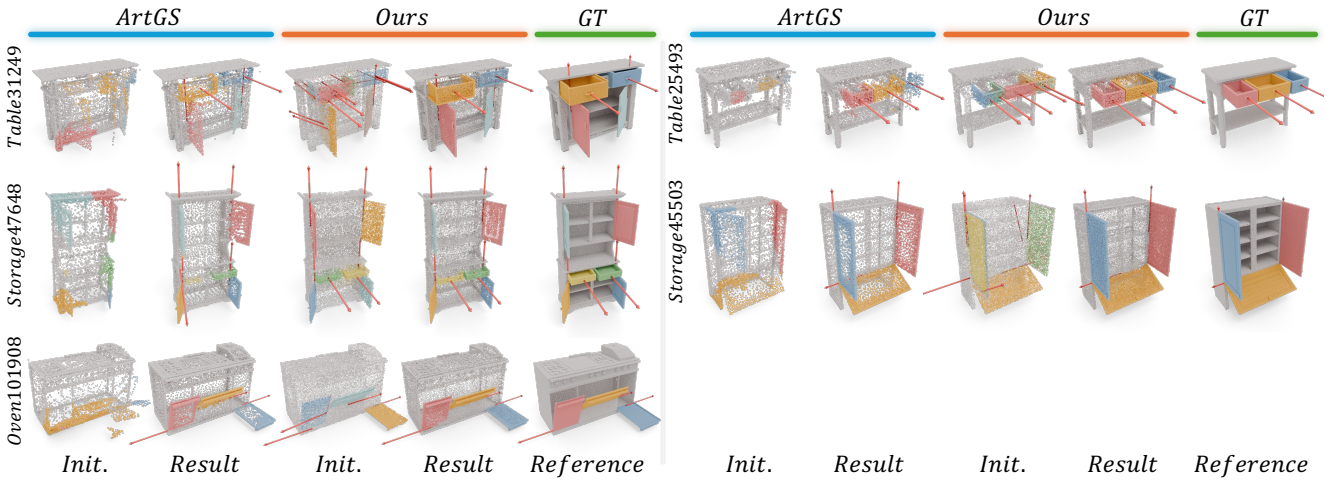

Figure 3. Additional qualitative results on **ArtGS-Multi dataset**, including the initializations and the final reconstructions. We show the Gaussians with their center points for a better visualization of their segmentation and motion parameters.

Table 4. Computation overhead comparison on our multi-part dataset. Time and memory are reported as mean±std.

Method	Time (min)	GPU Memory (GB)
DTA [7]	53.28±23.78	10.58±2.45
ArtGS [5]	10.41±0.89	2.94±0.04
Ours	16.93±2.37	2.69±0.23*

mented patches, and tend to merge the most relevant proposals in the first optimization cycle, which limits the total number of proposals and cycles. Second, the optimization of each cycle is monitored and stops early upon convergence.

Table 4 reports the average computation time and GPU memory usage across all objects in our dataset. Our method achieves a practical balance between computational cost and reconstruction quality. Although ArtGS is faster, it

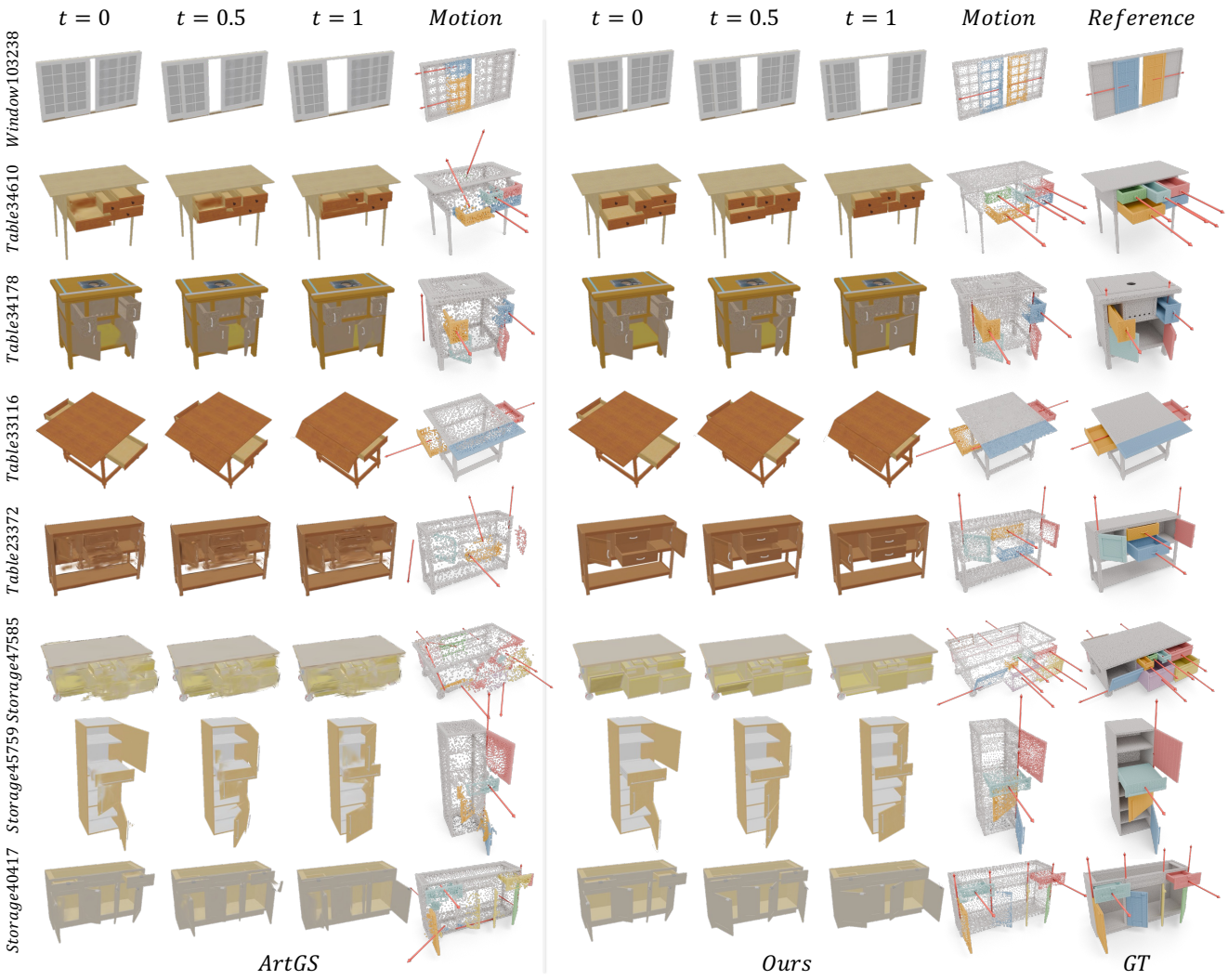

Figure 4. Additional reconstruction results on **our dataset**. We render the reconstructed articulated objects in different motion states ($t \in \{0, 0.5, 1\}$ and their motion structures for each result.

cannot robustly reconstruct complex multi-part objects, as shown in Tables 1–2 of the mainpage. DTA, on the other hand, requires substantially more time and memory while also failing on many multi-part cases.

G. Failure Cases and Robustness Analysis

Robustness to noisy depth and masks. Fig. 7 shows the robustness of our method to noisy depth maps and imprecise segmentation masks. For local geometry errors (Fig. 7(a)) and imprecise mask boundaries near drawer edges (Fig. 7(c)), our method mitigates the incorrect estimation from specific views by integrating multi-view information, achieving accurate reconstructions. However, our method would fail when the captured depth maps and masks are highly fragmented and misaligned, as in the real-scanned data provided by PARIS [3]. This can be enhanced using prior-prompted depth predictor, e.g., PriorDA [6], as we did to obtain our real-scanned data (the printer and storage).

Part boundary sensitivity. Although our approach achieves much more robust motion estimation and reconstruction of articulated objects than existing approaches, it still suffers from the common issues of the self-supervised 3DGS reconstruction framework. We show two representative failure cases in Figure 6. First, since our method lacks semantic part segmentation supervision for separating the adjacent movable parts, it sometimes causes the part boundary to encroach on neighboring part regions, such as the interior and the edge of the drawers in Figure 6.

Sensitivity to depth map quality. Second, the estimation of mobility parameters in our method primarily relies on the

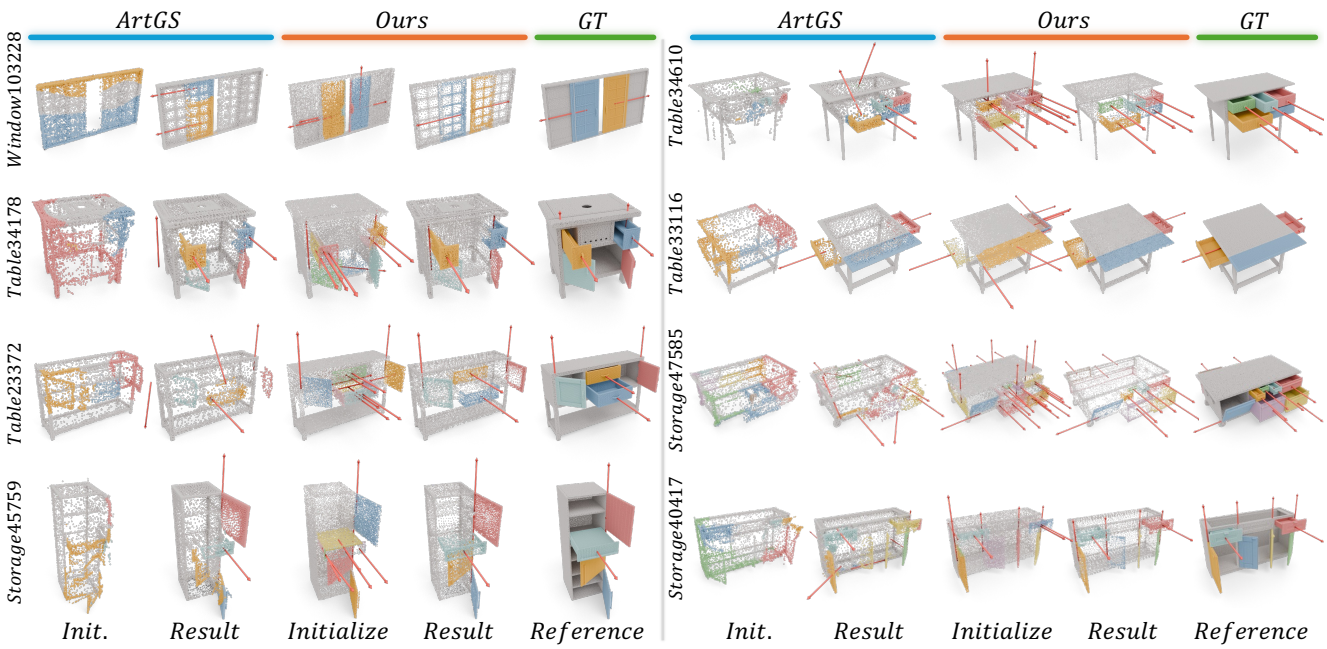

Figure 5. Additional qualitative results on **our dataset**, including the initializations and the final reconstructions. We show the Gaussians with their center points for a better visualization of their segmentation and motion parameters.

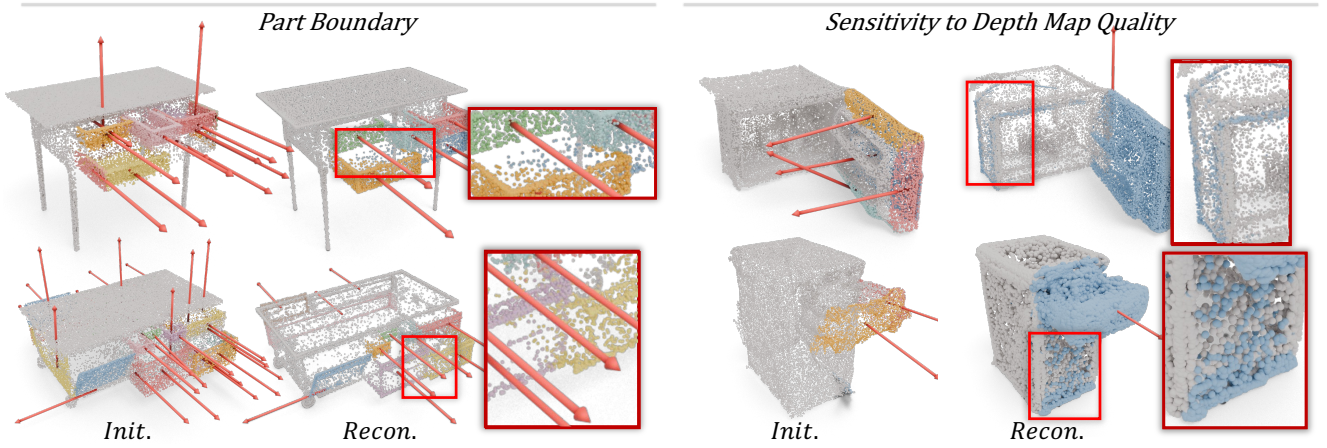

Figure 6. Failure cases. We illustrate failure cases of low-quality part boundaries. For low-quality RGBD images input, our method difficult to reconstruct the part boundaries.

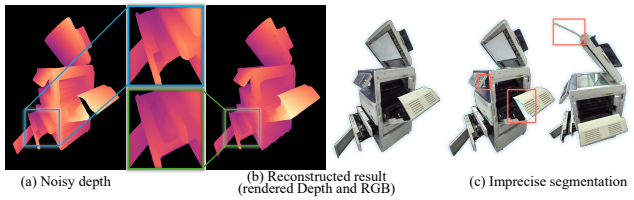

Figure 7. Influence of noisy depth and mask.

depth component of the RGBD loss $\mathcal{L}_I$ and the geometry-aware Chamfer distance loss $\mathcal{L}_{cd}$. Consequently, the accuracy of these estimations is highly dependent on the input depth map quality. In scenarios where the acquired depth maps contain significant inaccuracies, due to factors such as sensor noise, occlusions, or varying illumination conditions, the optimization process can be misled. As illustrated in Figure 6 (using real-world data from the PARIS dataset), this can cause our method to erroneously incorporate Gaussians from the static part into the estimated movable part. To achieve a lower loss value under these incorrect constraints, the optimization incorrectly transforms the movable part into the interior of the object. It is worth noting that in our real-world reconstruction experiments, the qual-

ity of the mobility results has been significantly improved by enhancing the input depth maps using the pre-trained Depth-Anything-V2 model [8] which effectively mitigates these issues.

Multi-DOF joints and non-rigid parts. Our current formulation assumes that each movable part undergoes a single rigid transformation, either revolute or prismatic. This assumption does not hold for objects with multi-DOF joints such as ball joints or compound hinges, or parts exhibiting non-rigid deformations. In such cases, our motion parameterization cannot capture the full range of part motion, leading to inaccurate articulation estimation. Future work will explore potential solutions including extending the motion model to support composite transformations or integrating learned deformation fields for non-rigid components.

References

[1] Junfu Guo, Yu Xin, Gaoyi Liu, Kai Xu, Ligang Liu, and Ruizhen Hu. Articulatedgs: Self-supervised digital twin modeling of articulated objects using 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27144–27153, 2025. 1, 2

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023. 1

[3] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-level reconstruction and motion analysis for articulated objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 352–363, 2023. 1, 2, 5

[4] Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9704–9715, 2025. 3

[5] Yu Liu, Baoxiong Jia, Ruijie Lu, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Artgs: Building interactable replicas of complex articulated objects via gaussian splatting. *arXiv preprint arXiv:2502.19459*, 2025. 1, 2, 4

[6] Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025. 5

[7] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3141–3150, 2024. 1, 2, 4

[8] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37: 21875–21911, 2024. 7