GEAR: GEometry-motion Alternating Refinement for Articulated Object Modeling with Gaussian Splatting

Supplementary Material

Supplementary Material Overview

The supplementary is organized as follows: Sec. A provides implementation specifics, algorithmic strategies, and hyperparameters. Sec. B presents comprehensive quantitative and qualitative results, including runtime analysis, EM framework robustness, and detailed performances on benchmarks. Finally, Sec. C discusses the current limitations of our method and potential future extensions.

A. Implementation Details

In this section, we provide implementation specifics, algorithmic strategies, and detailed hyperparameters.

A.1. Coarse Reconstruction Module

To ensure consistent operations across different states, we establish a unified voxelization space. We adopt an adaptive voxel scale v based on object complexity. Specifically, for simple single-joint objects, we use a coarser scale ($v = 0.1$) to robustly cover and bound all potential dynamic candidates. For complex multi-joint objects, we employ a finer scale ($v = 0.01$) combined with Top-K selection to isolate spatially adjacent small parts. Dynamic regions are extracted by subtracting the static intersection (processed with a morphological dilation) from the total voxel space.

Leveraging the prevalence of planar structures (e.g., doors, lids) in articulated objects, we apply a Structure-Aware Motion Initialization strategy. Extracted components with a high PCA-based aspect ratio (greater than 3.0) are classified as planes, allowing us to approximate the rotation axis via the intersection of fitted planes. For non-planar parts, we conservatively initialize with an identity matrix to avoid over-constraining the optimization.

As demonstrated in Fig. S1, the initialization prior acts as an anchor, guiding the subsequent GEAR optimization to refine geometry and motion, ultimately converging to a high-fidelity reconstruction.

A.2. Part-Aware Optimization Refinements

To handle the specific challenges posed by articulated objects with occlusions or thin structures, we introduce two targeted refinement strategies within the training loop:

Class-Specific Voting for Prismatic Joints. Prismatic joints (e.g., drawers, sliding doors) often exhibit limited visual changes compared to revolute joints and are frequently occluded by the static main body. During the SAM Mask Aggregation phase in the E-step, standard majority voting might allow the dominant static class to overwhelm these smaller dynamic regions. To mitigate this, we apply a voting boost factor $\gamma_{\text{prism}} = 4$ to the prismatic parts identified during optimization. Specifically, when aggregating SAM mask regions, votes cast for prismatic parts are weighted by γ_{prism}, ensuring that partially occluded drawers are correctly segmented and not absorbed by the static part.

Adaptive Opacity Maintenance for Small Parts. The standard 3D Gaussian Splatting pipeline periodically resets the opacity of all Gaussians to prevent local minima and encourage densification. However, for articulated parts with thin structures or small surface areas (e.g., handles, levers), the number of initialized Gaussians is often low. A global opacity reset can cause these valid but sparse Gaussians to be aggressively pruned. We implement a protective mechanism: before each opacity reset interval, we count the number of Gaussians assigned to each dynamic part. If a part contains fewer than $N_{\text{thresh}} = 2000$ points, we skip the opacity reset step for that specific part, ensuring its structural integrity is maintained throughout the optimization.

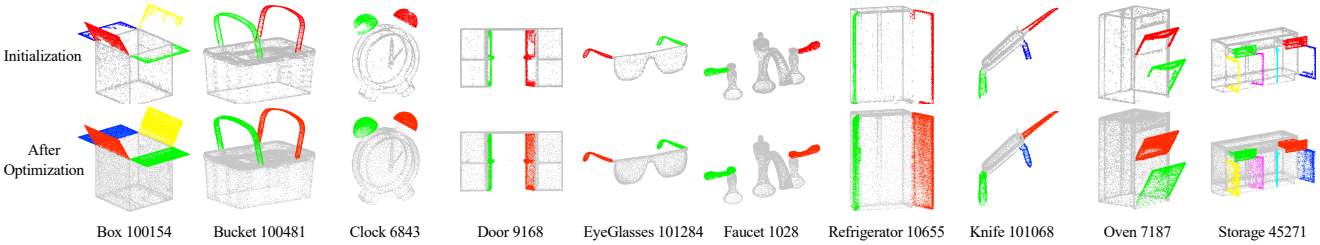

Figure S1. **Visualization of Initialization and Convergence.** Top row: The coarse point clouds derived from the voxel-based initialization module. Colors indicate different initialized parts. Bottom row: The final high-fidelity reconstruction after GEAR optimization. The comparison highlights that while the coarse module produces rough and discretized structures, it successfully captures the topological structure, guiding GEAR to converge to a photorealistic and geometrically accurate result.

A.3. Hyperparameters

We list the key hyperparameters used in GEAR in Tab. S1. The values kept consistent across all experiments unless otherwise stated.

Table S1. **Hyperparameters setting of GEAR.**

Category	Parameter	Value
Initialization	Voxel Size v	0.1 / 0.01*
	Morphological Dilation Radius	1
Training Loop	Total Iterations	30,000
	E-step Interval	500
	M-step Interval	500
	KNN Gaussian Neighbors $\mathcal{N}$	3
	Joint Classification Iteration	10,000
	Rotation Angle Threshold ϵ	10°
Loss Weights	D-SSIM Loss Weight $\lambda_{\text{D-SSIM}}$	0.2
	Depth Loss Weight λ_{Depth}	1.0
	Mask Cross-Entropy Weight λ_{SAM}	0.1
	KNN Consistency Weight λ_{KNN}	0.1

* Adaptive: 0.1 for single-joint and 0.01 for multi-joint objects.

B. Extended Experiments and Analysis

In this section, we provide comprehensive quantitative and qualitative results to further validate the efficiency, robustness, and generalizability of GEAR.

B.1. Computational Cost and Runtime Analysis

To evaluate the computational efficiency, we benchmark the runtime of GEAR against ArtGS [31], on a single NVIDIA RTX 4090 GPU.

Runtime. Our full optimization schedule (30k iterations for initialization + 30k for alternating refinement) is designed to prioritize maximum stability for extremely complex multi-joint objects. As detailed in Tab. S2, the overall processing time scales with the complexity (i.e., the number of movable parts) of the object. While the complex *Storage* object (7 parts) demands a longer convergence time (40 mins), simpler objects are typically reconstructed faster. Overall, the full pipeline averages approximately 34.0 minutes per object on the GEAR-Multi dataset.

Efficiency and Convergence. Importantly, our alternating EM framework intrinsically converges faster than joint optimization. To demonstrate its efficiency, we deploy a fast version, **Ours***, which halts optimization at 10k initialization iterations and 10k training iterations. As shown in Tab. S3, **Ours*** requires only **11.4 minutes** on average—faster than ArtGS (14.0 minutes)—yet it still substantially outperforms ArtGS across most metrics. Furthermore, GEAR matches ArtGS in VRAM efficiency ($<$ 8GB), lowering the hardware requirements.

Table S2. **Runtime Breakdown on GEAR-Multi.** Time is measured in minutes. The total computational cost scales reasonably with the number of movable parts.

Object	Parts	Initialization	training	Total Time
Box	5	13	35	48
Bucket	3	13	27	40
Clock	3	9	16	25
Door	3	12	17	29
EyeGlasses	3	10	24	34
Faucet	3	9	20	29
Knife	4	9	16	25
Oven	3	12	23	35
Refrigerator	3	13	22	35
Storage	7	14	26	40
Average	-	11.4	22.6	34.0

Table S3. Performance of the Fast Version on GEAR-Multi (Average). Our fast version (**Ours***) outperforms ArtGS in runtime while delivering substantially higher accuracy.

Method	Time (min)↓	Axis Ang↓	Axis Pos↓	Geo Dist↓	CD-s↓	CD-m↓	CD-w↓
ArtGS [31]	14.0	8.82	141.45	12.83	0.90	115.70	1.12
Ours* (Fast)	**11.4**	0.30	**0.01**	0.34	0.99	3.00	**0.95**
Ours (Full)	34.0	**0.09**	0.09	**0.08**	**0.86**	**0.71**	1.13

B.2. Effectiveness of the EM Framework

The contribution of GEAR lies in the EM-style alternating optimization framework. The ablation study in the main paper on complex multi-joint objects demonstrates that standard joint optimization struggles to converge. Here, we provide further insights into joint optimization on simpler tasks and the plugin generality of our EM framework.

Joint Optimization Performance on Simple Objects. To investigate whether the underlying GEAR representation (part-assigned Gaussians and dual-quaternion kinematics) inherently relies on the alternating strategy, we evaluate the joint optimization strategy on simpler, single-joint objects from the PARIS dataset. As shown in Tab. S4, optimizing the GEAR representation jointly yields reasonable results on

Table S4. **Alternating vs. Joint Optimization** on PARIS dataset. The results indicate that GEAR works reasonably well under joint optimization, while EM strategy behaves better.

Method	Axis Ang↓	Axis Pos↓	Geo Dist↓	CD-s↓	CD-m↓	CD-w↓
Ours (Joint)	0.11	**0.00**	0.32	2.32	2.19	1.93
Ours (EM)	**0.02**	**0.00**	**0.02**	**1.99**	**0.70**	**1.87**

Table S5. **Plugin Capability.** Applying EM framework (15k joint pre-training followed by 5k EM alternating refinement) to ArtGS [31] on the GEAR-Multi dataset reduces modeling errors.

Method	Axis Ang↓	Axis Pos↓	Geo Dist↓	CD-s↓	CD-m↓	CD-w↓
ArtGS	8.82	141.45	12.83	0.90	115.70	**1.12**
ArtGS + EM	**7.77**	**0.86**	**11.33**	**0.80**	**27.46**	1.37

Table S6. **Robustness Analysis of the EM Alternating Interval.** Evaluated on two complex 7-part objects. GEAR maintains stable, high-quality convergence across a broad range of interval settings (50 to 2000). Degeneration only occurs when the interval is excessively large (≥ 3000), mimicking the failure mode of staged optimization.

Method	Storage_47648						Storage_45271					
	Axis Ang↓	Axis Pos↓	Geo Dist↓	CD-s↓	CD-m↓	CD-w↓	Axis Ang↓	Axis Pos↓	Geo Dist↓	CD-s↓	CD-m↓	CD-w↓
ArtGS	0.14	0.02	0.62	0.67	3.70	0.70	9.33	1268.83	5.85	1.27	102.33	**1.74**
Ours-50	0.06	**0.01**	0.13	**0.53**	0.23	0.60	0.03	**0.00**	0.07	0.49	0.17	**0.89**
Ours-250	0.13	**0.01**	0.15	0.57	**0.22**	**0.59**	**0.02**	**0.00**	0.05	0.55	0.17	0.91
Ours-500 (Default)	0.07	**0.01**	**0.10**	0.61	0.32	0.60	0.03	**0.00**	**0.03**	0.50	**0.10**	3.92
Ours-2000	**0.05**	**0.01**	0.12	0.54	0.26	0.60	0.03	**0.00**	**0.03**	**0.43**	0.16	0.91
Ours-3000	0.19	**0.01**	0.24	0.55	0.24	0.60	13.17	3.02	9.54	1.00	73.78	0.94

simple objects.

This confirms that our fundamental geometric and motion representation is sound for basic tasks where error attribution ambiguity is relatively low. While joint optimization is applicable in these cases, the EM alternating strategy further reduces the errors (Angle error to 0.02 and CD-m to 0.70). More importantly, as demonstrated in the main paper, when object complexity scales up to multiple joints, the EM framework becomes necessary to prevent the optimizer from getting trapped in local minima.

Plugin Capability for Existing Methods. To demonstrate that our EM-style formulation also serves as a generalizable optimization plugin, we integrate alternating strategy into ArtGS [31]. As shown in Tab. S5, standard ArtGS struggles on GEAR-Multi dataset. To address this, we modify its training schedule: using its default joint optimization for the first 15k iterations, followed by our EM alternating refinement for the final 5k iterations (**ArtGS + EM**).

This plug-and-play refinement substantially reduces errors: the Axis Pos error drops from 141.45 to 0.86, and the CD-m decreases from 115.70 to 27.46. This validates that our alternating paradigm is not only effective for training from scratch, but also a robust approach to recovering existing methods from local minima.

B.3. Robustness of the EM Alternating Interval

To demonstrate that GEAR does not rely on exhaustive per-object hyperparameter tuning, we evaluate its sensitivity to the EM alternating interval (iterations per E/M-step) on two 7-part objects (*Storage_47648* and *Storage_45271*).

As shown in Tab. S6, GEAR stably maintains high-fidelity convergence across a wide range of intervals (50 to 2000 iterations), consistently outperforming the ArtGS baseline. Degeneration only occurs at excessively large intervals (e.g., 3000), where the process mimics a single-pass "Staged Optimization," causing early segmentation errors to irrecoverably propagate. This confirms our framework's inherent stability for diverse articulated objects.

B.4. Robustness to Imperfect 2D Mask

To evaluate the robustness of our framework against imperfect 2D priors, we analyze its performance under typical failure cases of SAM, such as over-segmentation (fragmenting parts) or under-segmentation (merging parts). As visualized in Fig. S2, SAM produces flawed 2D priors. Nevertheless, as quantitatively validated in Tab. S7, GEAR effectively filters out this segmentation noise, yielding highly accurate geometric and motion estimates on these failure-prone objects.

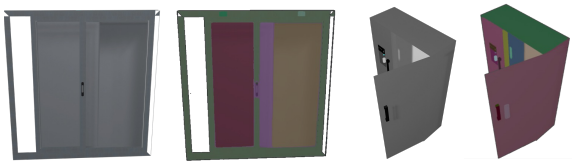

Figure S2. **Failure Cases of SAM.** Visualizations of flawed 2D masks, including over-segmentation (left) and under-segmentation (right), which serve as the challenging initial priors for our framework.

Table S7. **Performance under Flawed 2D Priors.** GEAR maintains high accuracy despite being initialized with the erroneous masks shown in Fig. S2.

Object	Axis Ang↓	Axis Pos↓	Geo Dist↓	CD-s↓	CD-m↓	CD-w↓
Window_102985	0.06	-	0.00	0.77	2.10	0.68
Refrigerator_10685	0.00	0.00	0.02	0.52	0.10	0.53

B.5. Detailed Results on PARIS Dataset

Here, we present the complete per-object breakdown in Tab. S8, which covers 10 synthetic objects and 2 real-world objects. As shown in the table, GEAR achieves the highest accuracy in motion parameter estimation across the majority of simulated objects.

B.6. Reconstruction Results

We provide a comprehensive qualitative comparison between GEAR and the baseline ArtGS [31] across diverse objects

Table S8. Detailed quantitative comparison on the PARIS dataset, averaged over 10 trials for both simulated and real objects with higher visibility. Lower (↓) is better for all metrics. The best (**Bold** with Top-1) and second-best (Top-2) results are highlighted. Baseline results for Ditto, PARIS, DTA, and ArtGS are sourced from ArtGS, objects with * are seen categories trained in Ditto.

		Simulation											Real		
		Foldchair	Fridge	Laptop*	Oven*	Scissor	Stapler	USB	Washer	Blade	Storage*	Average	Fridge	Storage*	Average
Axis Ang	Ditto	89.35	89.30	3.12	0.96	4.50	89.86	89.77	89.51	79.54	6.32	46.22	1.71	5.88	3.80
	PARIS	7.90	9.19	0.02	0.04	3.92	0.73	0.13	25.18	15.18	0.03	6.23	**1.64**	43.13	22.39
	DTA	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	0.01	0.03	**0.01**	**0.01**	0.05	**0.01**	0.04	**0.02**	**0.03**	0.02	**0.02**	2.09	**3.47**	**2.78**
	Ours	**0.00**	**0.01**	0.02	**0.01**	**0.00**	**0.01**	**0.01**	0.05	0.09	**0.00**	**0.02**	2.16	4.85	3.50
Axis Pos	Ditto	3.77	1.02	0.01	0.13	5.70	0.20	5.41	0.66	-	-	2.11	1.84	-	1.84
	PARIS	0.37	0.30	0.02	**0.00**	1.52	2.26	2.37	1.50	-	-	1.04	**0.34**	-	**0.34**
	DTA	0.01	0.01	0.01	0.01	0.02	0.02	**0.00**	0.05	-	-	0.02	0.59	-	0.59
	ArtGS	**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**	0.38	-	0.38
Geo Dist	Ditto	99.36	F	5.18	2.09	19.28	56.61	80.60	55.72	F	0.09	39.87	8.43	0.38	4.41
	PARIS	131.82	24.64	3.03	0.04	120.61	10.71	64.91	60.62	0.54	0.14	41.71	2.16	0.56	1.36
	DTA	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	**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	0.04	**0.01**	**0.01**	**0.02**	**0.01**	0.02	**0.01**	**0.03**	**0.00**	**0.00**	**0.02**	2.17	0.06	1.18
CD-s	Ditto	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	9.12	3.73	0.45	12.85	1.83	**1.96**	2.58	25.19	1.33	12.80	7.18	42.57	54.54	48.56
	DTA	**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	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	0.20	**0.44**	0.53	2.73	**0.44**	3.89	2.72	5.28	0.67	**2.98**	**1.99**	**1.50**	3.70	2.60
CD-m	Ditto	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	8.79	7.76	0.49	28.51	46.69	19.36	5.53	178.39	25.29	76.75	39.76	45.66	864.82	455.24
	DTA	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	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	**0.12**	**0.19**	**0.11**	**0.26**	**0.39**	0.56	1.39	**0.05**	1.89	2.02	0.70	0.69	16.01	8.35
CD-w	Ditto	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	1.90	2.53	0.50	**1.94**	10.20	6.30	2.31	24.71	0.44	**3.84**	5.47	22.98	63.35	43.17
	DTA	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	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	**0.23**	**0.56**	0.47	2.85	0.47	2.59	1.61	5.32	0.61	4.01	1.87	**1.00**	4.99	2.99

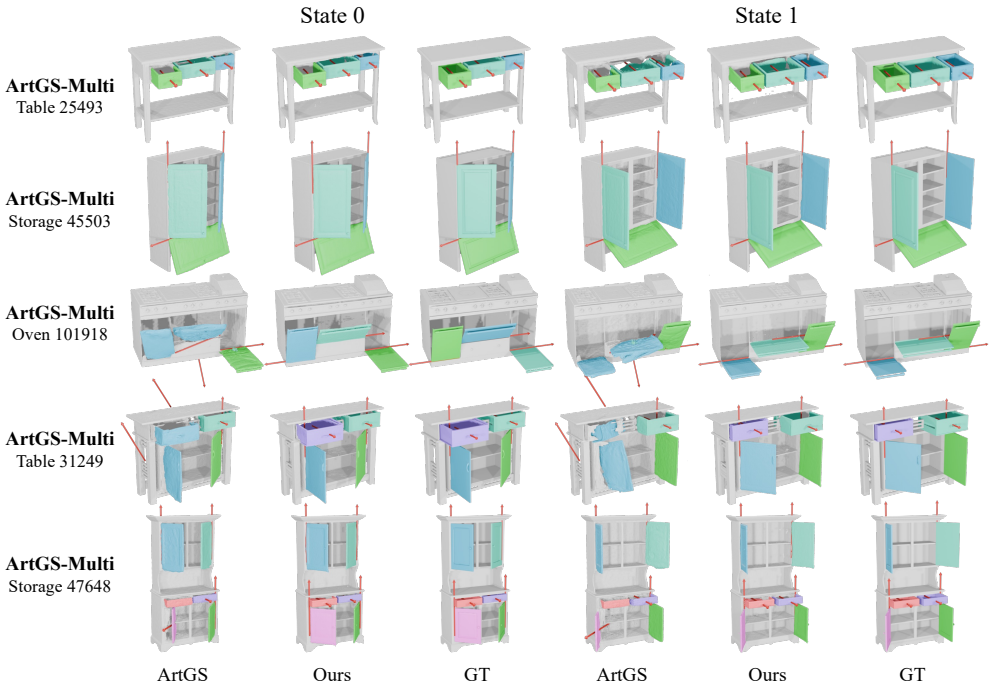

Figure S3. **Qualitative comparison on the ArtGS-Multi dataset.** We visualize all reconstruction results across two states. The columns show the results from the ArtGS baseline, Ours, and the Ground Truth (GT). As highlighted in the visualizations (e.g., *Oven 101918* and *Table 31249*), ArtGS struggles to distinguish spatially adjacent parts, leading to fused segmentation. In contrast, GEAR accurately disentangles these components and estimates precise motion axes.

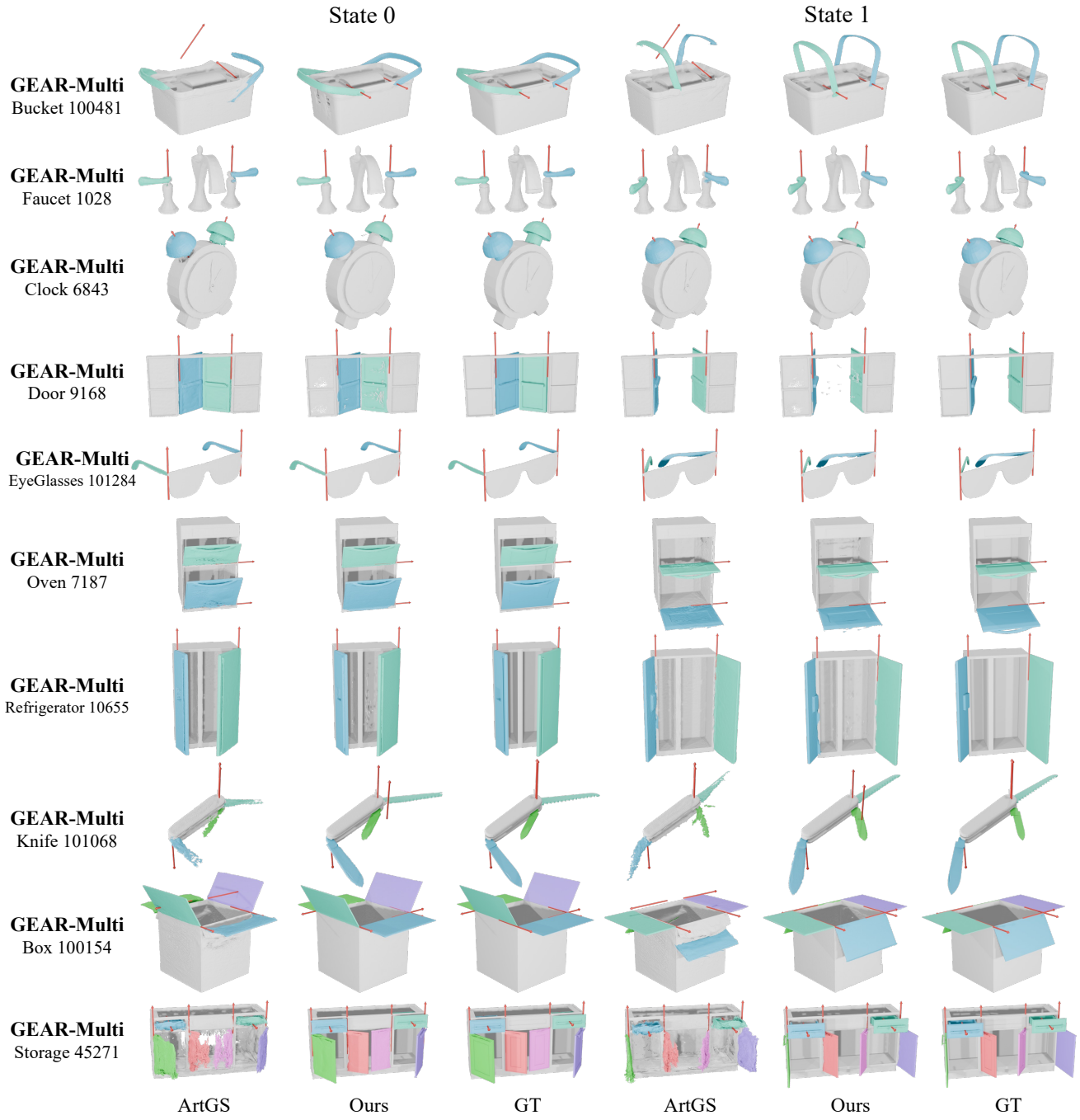

Figure S4. **Qualitative comparison on the GEAR-Multi dataset.** We present results on all 10 objects with diverse categories and topological structures. GEAR demonstrates robustness in handling objects with thin structures and small movable parts (e.g., *Bucket 100481*, *Knife 101068*), whereas ArtGS often fails to capture these fine-grained geometries or generates noisy artifacts.

from both ArtGS-Multi and GEAR-Multi datasets. As visualized in Fig. S3 and Fig. S4, GEAR successfully disentangles tightly packed adjacent parts and preserves high-frequency geometric details for thin structures (e.g., handles, knife blades), whereas the baseline often generates noisy artifacts or fused geometries.

To validate the physical correctness of our estimated motion parameters, we render the articulated objects at inter-

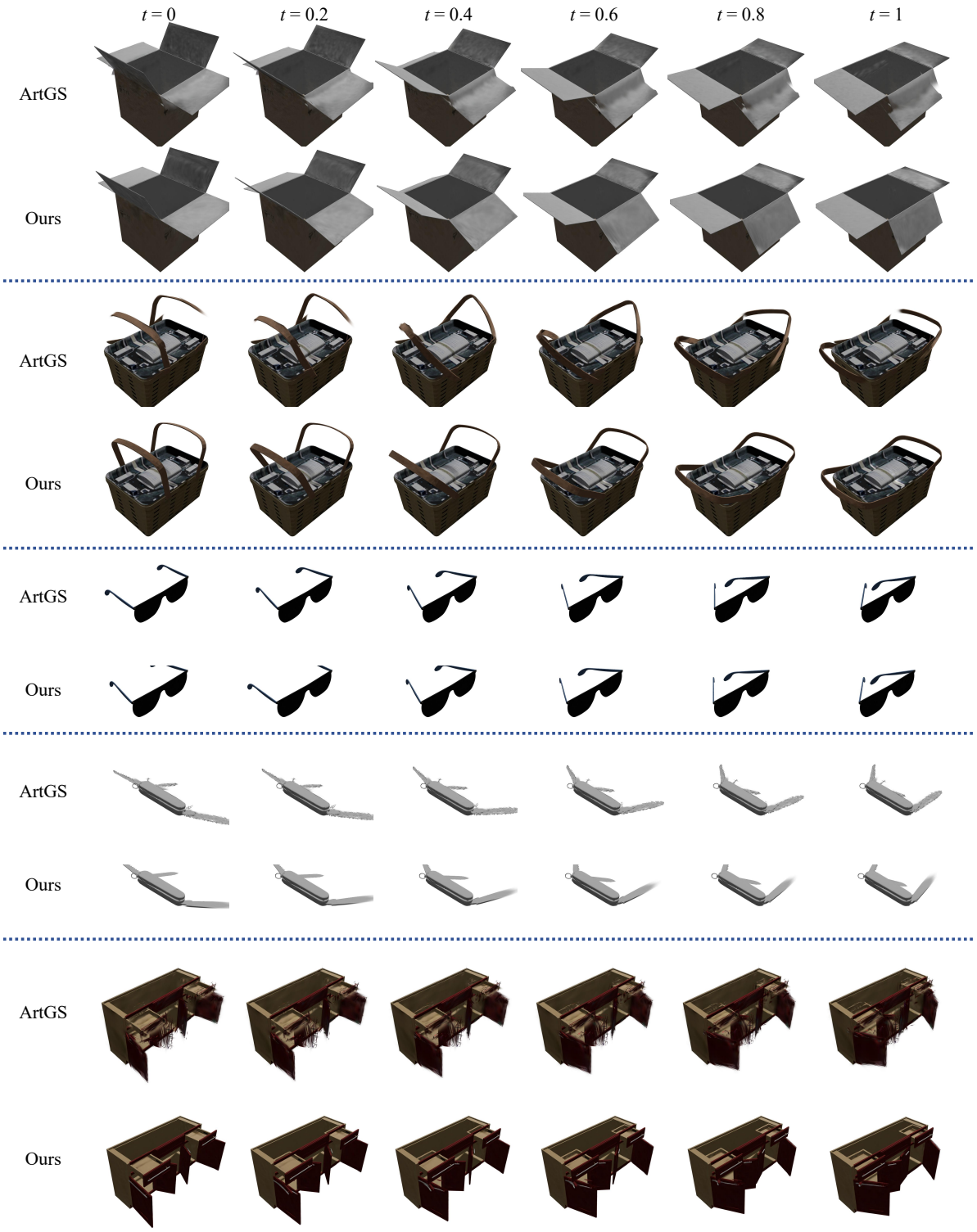

Figure S5. **Novel Joint Motion Synthesis.** We render the articulated object at intermediate states $t = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. GEAR (Bottom) preserves the geometry of the door and the contents inside the fridge consistently throughout the motion trajectory, whereas the baseline (Top) suffers from geometric distortion and blurring.

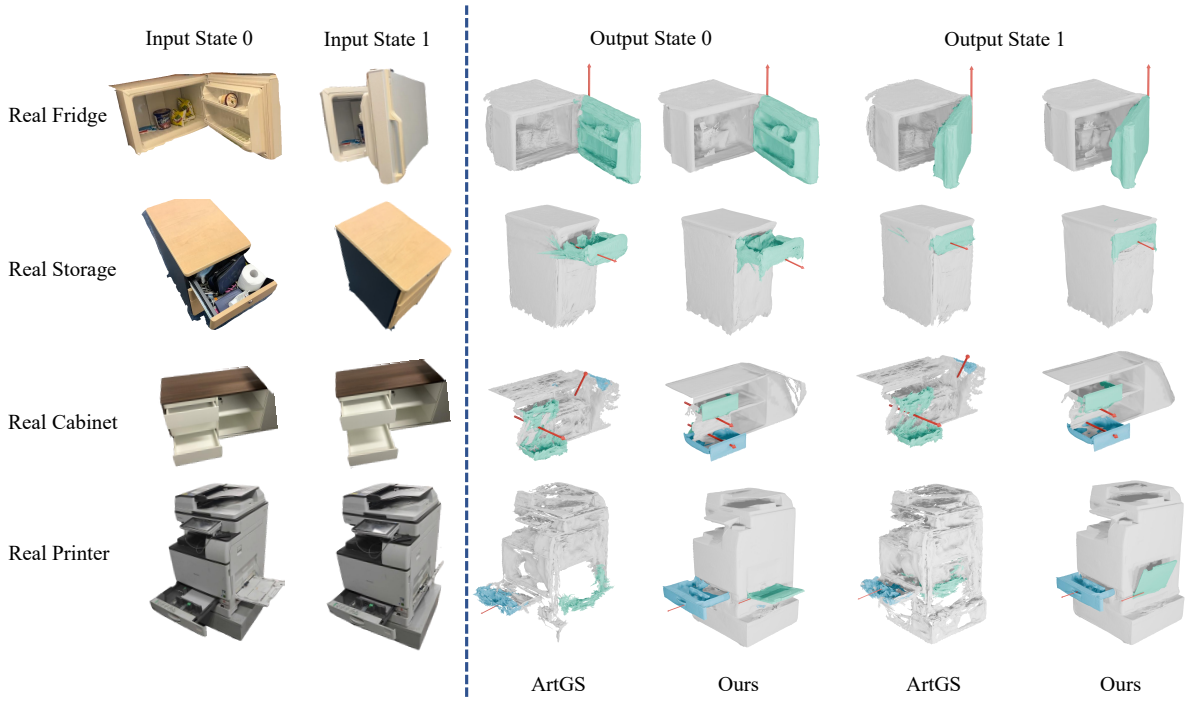

Figure S6. **Qualitative results on real-world objects.** The top two rows display single-joint objects from the PARIS dataset. The third row shows a multi-joint cabinet from the ArtGS dataset, and the bottom row depicts a printer captured in our lab. While ArtGS achieves comparable results on simple single-joint objects, it fails to reconstruct the structural details of complex multi-joint objects, leading to noisy artifacts. GEAR consistently delivers high-fidelity reconstruction and accurate part segmentation across all sources.

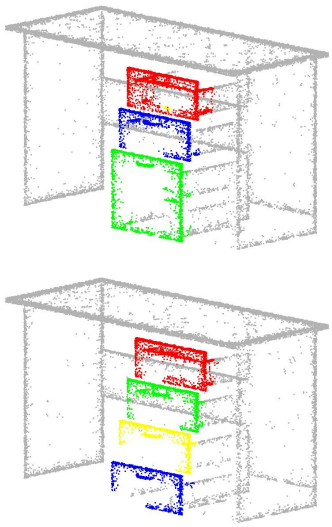

Figure S7. **Spatial Ambiguity in Initialization.** Top: When two movable parts are spatially adjacent with no gap (parallel drawers), our coarse module may fail to separate them, initializing them as a single part. Bottom: When parts are spatially staggered (offset drawers), the module successfully identifies distinct components.

mediate continuous states. As shown in Fig. S5, GEAR preserves the geometry of moving parts seamlessly throughout the entire motion trajectory without distortion.

B.7. Performance on Real-World Objects

To assess robustness against sensor noise and uncontrolled lighting, we evaluate GEAR on real-world objects. As visualized in Fig. S6, while the baseline produces reasonable reconstructions on simple single-joint objects, it exhibits significant performance degradation on complex multi-joint inputs (e.g., Real Cabinet and Real Printer), resulting in severe floating artifacts and incorrect part assignments. In contrast, GEAR preserves the integrity of planar surfaces and accurately disentangles coupled motion parts under real-world conditions.

C. Limitations

While GEAR demonstrates robust performance on a wide range of complex articulated objects, it is subject to certain limitations that present exciting avenues for future research.

C.1. Current Limitations

Spatial Ambiguity in Coarse Initialization. Our coarse reconstruction module relies on Connected Component Analy-

sis (CCA) of dynamic voxels to separate movable parts. This approach assumes that different movable parts are spatially disjoint in 3D space. However, when multiple movable parts are tightly packed or perfectly adjacent (e.g., side-by-side cabinet doors with negligible gaps), the dilated voxel grids may merge into a single connected component.

As illustrated in Fig. S7, we present a failure case with a 4-door cabinet where the two bottom doors are flush against each other. The initialization module incorrectly merges them into a single dynamic group. In contrast, we show a success case with the same object where the drawers are open in a *staggered* configuration. The spatial offset allows the CCA to correctly identify them as distinct parts. This indicates that our current initialization requires a minimal spatial separation between moving parts.

Misclassification in Extreme Articulation. GEAR relies on visual correspondence to estimate motion parameters. For extreme articulations, particularly 180-degree rotations, the visual overlap between the canonical state ($180°$) and the target state ($0°$) is minimal, and the geometric displacement is extremely large. In such scenarios, the optimization landscape is highly non-convex. We observe that the model may fall into a local minimum where it interprets the large displacement as a linear translation rather than a rotation.

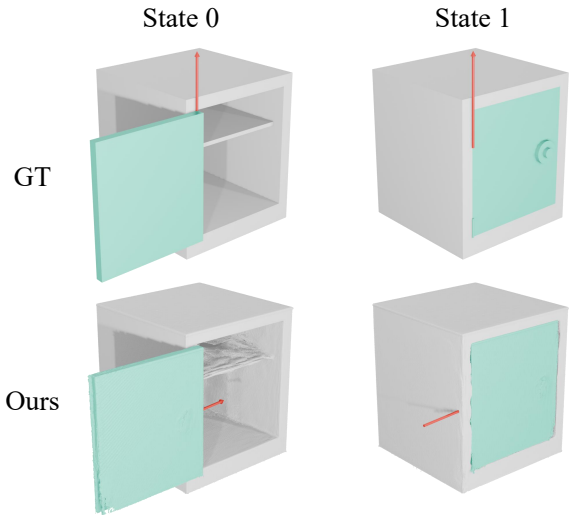

Figure S8. **Misclassification in Extreme Motion.** For a safe box door rotating 180 degrees, the large displacement and lack of intermediate visual cues lead the model to misclassify the revolute joint as a prismatic joint (visualized by the incorrect linear motion arrow).

Fig. S8 demonstrates this on a *Safe Box* object with a single revolute door opening 180 degrees. The model incorrectly identifies the joint as a prismatic joint, predicting a large translation vector instead of the correct rotation axis and angle. Integrating kinematic priors or trajectory constraints could be a potential solution.

Material Constraints. Finally, our method inherits the inherent limitations of the Gaussian Splatting representation. GEAR struggles to reconstruct articulated objects made of transparent (e.g., glass cabinets) or highly reflective materials. The view-dependent effects of such materials are difficult to model with standard spherical harmonics, leading to noisy geometry or "holes" in the reconstruction. Since GEAR uses geometric cues (depth and occupancy) for part segmentation, these rendering artifacts can propagate errors into the segmentation and motion estimation pipeline.

C.2. Future Extensions

To address these limitations and broaden GEAR's applicability, future work can explore several promising directions. First, integrating generative priors (e.g., diffusion models) could facilitate in-the-wild reconstruction from sparse observations by hallucinating unobserved geometry. Second, to resolve misclassifications in extreme motions (as shown in Fig. S8), incorporating 4DGS temporal modeling from continuous video sequences would allow the framework to explicitly track non-linear trajectories and avoid local minima. Finally, evolving our formulation into neural deformation fields would enable the simultaneous modeling of rigid part articulation and localized non-rigid dynamics for flexible objects.