Part-aware Modeling of Articulated Objects using 3D Gaussian Splatting

Supplementary Material

A. Implementation Details

Part Assignment Details. As defined in Section 4.2, the part identity of a Gaussian G_i is represented by a continuous probability distribution $F(G_i) = \text{softmax}(f(\boldsymbol{\psi}_i))$. To maintain full differentiability, we employ a soft, probability-weighted strategy for applying transformations.

The final transformed position $\boldsymbol{\mu}_i^{(t)}$ of Gaussian G_i is computed as a weighted sum over all K possible part transformations $\mathcal{T} = \{T_k\}_{k=1}^K$:

$$\boldsymbol{\mu}_i^{(t)} = \sum_{k=1}^{K} p_{i,k} \left(\mathbf{R}_k^{(t)} \boldsymbol{\mu}_i^0 + \mathbf{t}_k^{(t)} \right) + \mathbf{F}_{\text{repel},i}. \quad (13)$$

Here, $p_{i,k}$ denotes the probability that Gaussian G_i belongs to part k. This formulation enables the articulation and consistency losses to jointly optimize both the part-identity embedding $\boldsymbol{\psi}_i$ and the transformation parameters $(\mathbf{R}_k, \mathbf{t}_k)$. During inference, each Gaussian is assigned the rigid transformation of its most likely part, given by $k^* = \text{argmax}_k F(G_i)$.

Part Supervision. Our method does not require explicit part-level supervision, but it does assume a user-specified upper bound on the number of possible part groups, denoted by K. Specifying K does not introduce supervision for the following reasons: (1) The model is never told which part corresponds to which semantic region; it must infer part clusters entirely through geometric and motion consistency losses. (2) The KL-based neighborhood regularization (Section 4.2) forces part probabilities to self-organize based purely on geometric affinity. Thus, the method remains fully self-supervised with respect to part identity.

We also analyze the effect of misspecifying the number of part K. Table 4 shows that under-specifying K significantly degrades accuracy, while over-specifying it causes only mild degradation. Under-specifying K forces multiple physically distinct parts to share a single rigid slot. Because each slot models only one SE(3) motion, merging parts with different joint axes produces inconsistent transformations, leading to large errors in motion estimation and geometry reconstruction. In contrast, over-specifying K introduces extra slots that receive no coherent geometric or kinematic signal. These redundant slots naturally collapse due to the part regularizer, velocity-consistency loss, and articulation constraints, resulting in only mild degradation.

Repel Point Initialization. In our formulation, repel points are placed only on the static base and used to discourage interpenetration from movable parts. We perform an ablation on the most complex object Storage (7 parts), adopting a slightly more general and stable strategy. Specifically, we first use the canonical Gaussians to identify locations where movable parts lie within a small distance threshold of the static base. We then uniformly sample $N_R = 2000$ repel points from these proximity regions, which naturally concentrates repulsion forces along potential contact interfaces. These repel points remain fixed throughout training and are not updated or pruned, preventing drift and keeping the optimization stable.

Table 4. **Specifying # parts.** Lower (↓) is better across all metrics. highlights the best-performing setting.

K	Metric	Storage (4 parts)	Oven (4 parts)	Table (4 parts)	Metric	Storage (4 parts)	Oven (4 parts)	Table (4 parts)
2		0.12	0.20	0.25		4.90	2.30	14.80
3		0.06	0.12	0.18		3.80	1.15	14.65
4	**Ang Err**	0.01	0.03	0.08	**CD_{static}**	0.68	1.01	0.56
5		0.01	0.04	0.09		0.70	1.05	0.58
6		0.02	0.05	0.10		1.72	1.20	0.65
2		0.45	0.56	-		4.20	5.30	13.00
3		0.22	0.23	-		1.12	0.48	12.40
4	**Pos Err**	0.00	0.01	-	**$CD_{movable}$**	0.07	0.11	1.95
5		0.01	0.02	-		0.28	0.22	2.45
6		0.02	0.03	-		0.39	0.34	2.70
2		0.40	0.65	0.46		4.10	7.30	6.90
3		0.45	0.32	0.23		1.95	1.62	2.60
4	**Motion Err**	0.02	0.18	0.00	**CD_{whole}**	0.80	0.95	0.51
5		0.03	0.19	0.01		1.12	1.27	0.93
6		0.04	0.20	0.02		2.84	1.99	1.55

Table 5. **Sensitivity of repel point count (N_R).** Lower (↓) is better.

Metric	$N_R = 500$	$N_R = 2000$	$N_R = 4000$
Ang Err	0.11	0.11	0.12
Pos Err	0.01	0.01	0.01
Motion Err	0.57	0.55	0.58
CD_{whole}	0.63	0.63	0.64

As shown in Table 5, performance remains stable across all tested values, with no noticeable impact on final articulation accuracy. Using too few repel points slightly increases transient overlap at early iterations, but it does not affect convergence. Increasing N_R provides no measurable benefit, confirming that our method does not depend on problem-specific tuning. Because repel points act as a soft collision prior and are not tied to any assumptions about joint type or motion, the model naturally corrects for noisy or imperfect repel placement during optimization.

Differentiability of Repulsion Forces. The repulsion update $\boldsymbol{\mu}_i^{k,(t)} \leftarrow \boldsymbol{\mu}_i^{k,(t)} + \mathbf{F}_{\text{repel},i}^k$ is implemented as a fully differentiable operation within the optimization pipeline. The displacement caused by $\mathbf{F}_{\text{repel},i}^k$ participates directly in the computation graph rather than acting as a post-processing step. Consequently, during backpropagation, gradients flow through the repulsion force term to the transformation parameters $T_k = (\mathbf{R}_k, \mathbf{t}_k)$. This effectively penalizes configurations where the optimization would otherwise drive Gaussians into repulsion zones, encouraging the learning of

Table 6. **Canonical initialization ablation.** Lower (↓) is better across all metrics. highlights best-performing strategy.

Strategy	Metrics	Table (5 parts)	Storage (7 parts)	Metrics	Table (5 parts)	Storage (7 parts)
Uniform Interpolation		0.15	0.21		1.40	1.75
Motion-Aware Per-Part β	Ang Err	0.12	0.18	CD_{static}	1.32	1.60
Motion-Aware Global β		0.03	0.11		1.18	0.61
Uniform Interpolation		0.30	0.70		2.40	4.20
Motion-Aware Per-Part β	Motion Err	0.20	0.52	$CD_{movable}$	2.15	3.00
Motion-Aware Global β		0.01	0.55		1.85	1.83
Uniform Interpolation		0.08	0.12		1.20	1.45
Motion-Aware Per-Part β	Pos Err	0.05	0.09	CD_{whole}	1.13	1.38
Motion-Aware Global β		0.00	0.01		1.10	0.63

collision-free trajectories that naturally avoid repel points while satisfying the alignment loss $\mathcal{L}_{\text{art}}$.

Stability and Force Clamping. The inverse cubic falloff defined in the main paper ($1/\|\mathbf{r}_j - \boldsymbol{\mu}_i^k\|^3$) provides strong localized gradients but poses a risk of numerical instability (gradient explosion) as the distance approaches zero. To ensure training stability, we implement two specific safeguards: **(1) Distance Clamping:** We impose a lower bound on the distance denominator. The L2 distance $\|\mathbf{r}_j - \boldsymbol{\mu}_i^k\|_2$ is clipped to a minimum value $\epsilon = 10^{-5}$. This prevents division by zero and bounds the maximum repulsive force applied to any single Gaussian. **(2) Force Magnitude Saturation:** We further limit the norm of the total force vector $\|\mathbf{F}_{\text{repel},i}^k\|$ to a maximum threshold $\tau_{\max}$ to prevent outliers from destabilizing the transformation updates in a single iteration. Thus, the effective robust force calculation is given by:

$$\mathbf{F}_{\text{repel},i}^k = \text{clip}\left(\sum_{\mathbf{r}_j \in \mathcal{R}} k_r \cdot \frac{(\mathbf{r}_j - \boldsymbol{\mu}_i^k)}{\max\left(\|\mathbf{r}_j - \boldsymbol{\mu}_i^k\|, \epsilon\right)^3}, \tau_{\max}\right), \tag{14}$$

where $\text{clip}(\mathbf{v}, \tau_{\max}) = \mathbf{v} \cdot \min(1, \tau_{\max}/\|\mathbf{v}\|)$ denotes the vector magnitude clipping operation.

Global vs. Per-part Interpolation Weighting. As described in Section 4.1, the interpolation weight β is computed once per object from the global motion richness scores $D_{0\to1}$ and $D_{1\to0}$. While this scalar coefficient is shared across all matched Gaussians, we find in practice that a global β is sufficient for the purpose of initializing a stable canonical field. This is because β is used only during initialization to place the canonical Gaussians in a reasonable configuration before the full SE(3)-based deformation module is optimized. Once training begins, each Gaussian's part membership, transformation, and geometry are updated independently, allowing the model to account for heterogeneous motion magnitudes across parts.

We additionally experiment with (i) uniform averaging and (ii) motion-aware per-part β. As shown in Table 6, both alternatives introduce instability and degrade performance. Per-part β is especially sensitive to local displacement noise and fails to reflect the actual articulation structure. In contrast, a single global β provides a simple and noise-robust prior while keeping the initialization lightweight.

Hyperparameters. For loss weighting, we set λ_{part}=0.1, λ_{art}=1.0, and λ_{phys}=0.5, with equal weights across the three physical regularizers. We set the maximum number of parts K according to category-level priors, typically 3–7. The repulsion strength is fixed to k_r=5×10^{-4}, and we sample N_R=2000 repel points from regions where canonical Gaussians of movable and static parts fall within a 1.5 unit length proximity threshold. Repel points remain fixed throughout training. The SE(3) transformations for each part are optimized jointly with Gaussian parameters using Adam with learning rate 1e−3. The canonical Gaussian initialization from the two observed states uses 30k iterations of single-state 3DGS followed by 5k iterations of canonical fusion with the global β weighting.

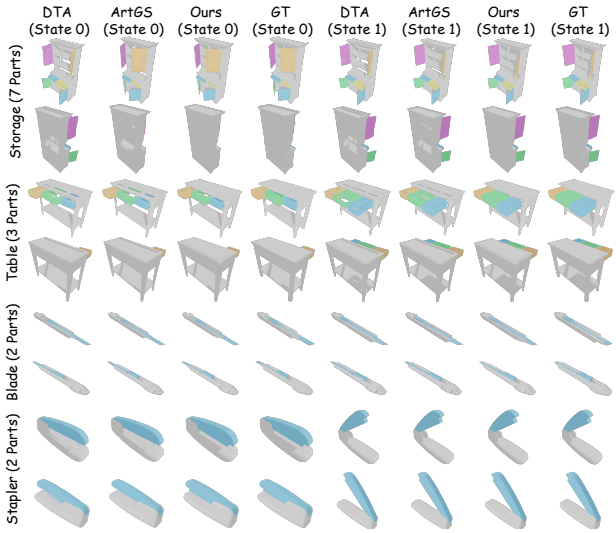

Figure 6. **Mesh visualizations**, confirming high-quality surface reconstruction and consistent part articulation.

B. Additional Qualitative Examples

Mesh Visualization. Figure 6 shows qualitative comparisons across four articulated objects, *i.e.* Storage (7 Parts), Table (3 Parts), Blade (2 Parts), and Stapler (2 Parts), under State 0 and State 1. Overall, Part^{2}GS closely matches GT in both geometry and articulation consistency across states. The improvements are especially visible for the multi-part Storage (7 Parts) and Table (3 Parts) examples.

Motion Trajectory Visualization. Figure 7 presents additional 2-part objects exhibiting diverse geometries and joint types, including rotary (scissors), prismatic (utility knife), and hinged motion (stapler, container lid). Across all examples, Part^{2}GS produces smooth and monotonically consistent motion trajectories as the articulation parameter T progresses from 0 to 1. The movable parts follow realistic kinematic paths without drifting, collapsing into the static base, or introducing geometric distortion. Notably, fine-scale geometry such as the scissor blades and the tapered cutter head remains stable throughout the motion sequence, demonstrating the robustness of our method.

Table 7. **Inference time for simple and complex objects.** Simple objects have one movable part while complex objects have multiple, denoted by their subscript (*e.g.,* Table_4 has a static base and three movable parts). highlights best performing results.

Metric	Method	Simple Objects										Complex Objects						
		Foldchair	Fridge	Laptop	Oven	Scissor	Stapler	USB	Washer	Blade	Storage	Fridge_3	Table_4	Table_5	Storage_3	Storage_4	Storage_7	Oven_4
Time (Min)	DTA	29	30	31	29	28	29	31	28	27	28	32	34	37	32	35	45	35
	ArtGS	9	8	7	7	7	7	7	8	7	8	8	8	8	8	8	8	8
	Part^{2}GS	8	9	7	8	7	8	7	8	7	9	9	8	9	8	9	10	9

Table 8. **Part^{2}GS module removal ablations** on the two most complex objects in our evaluation, Table (5 parts) and Storage (7 parts). Lower (↓) is better on all metrics. shows results with all Part^{2}GS modules while highlights severe failures by removing components of our method. Severe failures are defined as metrics that are more than 5 times worse than the full Part^{2}GS for the same object.

Objects	Methods	AngErr	PosErr	MotionErr	$\text{CD}_{\text{static}}$	$\text{CD}_{\text{movable}}$	CD_{whole}
Table (5 parts)	✗ part parameters	0.21	0.08	7.32	7.35	145.17	3.10
	✗ repel points	0.09	0.16	0.48	1.19	4.82	1.85
	✗ physical constraints	0.05	0.03	0.18	1.32	4.47	1.65
	✗ canonical init	0.14	0.06	6.32	2.47	117.25	2.62
	Part^{2}GS (all)	0.03	0.00	0.01	1.18	1.85	1.10
Storage (7 parts)	✗ part parameters	0.26	0.11	10.43	2.95	198.67	3.54
	✗ repel points	0.16	0.14	1.32	0.93	7.43	2.04
	✗ physical constraints	0.04	0.05	0.04	1.22	4.54	1.12
	✗ canonical init	22.15	0.93	19.67	0.79	442.32	1.89
	Part^{2}GS (all)	0.11	0.01	0.55	0.61	1.83	0.63

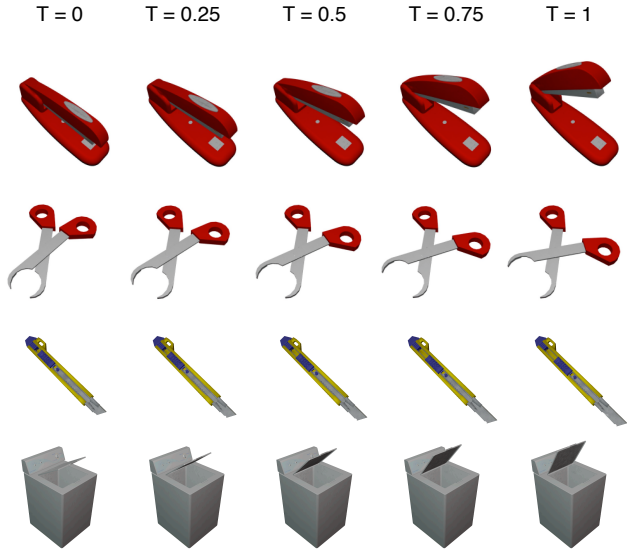

Figure 7. **Part^{2}GS qualitative results** on 2-part objects with different joints and distinct geometry structures.

C. Inference Time

Table 7 compares the per-object inference runtimes of DTA, ArtGS, and our method Part^{2}GS on both simple (one movable part) and complex (multiple movable parts) objects. On the ten simple objects, DTA requires between 28 and 31 minutes each, whereas both ArtGS and Part^{2}GS complete inference in under 10 minutes, yielding roughly a 70–75% speedup. Notably, Part^{2}GS achieves the best or tied-best time on eight out of ten simple objects, with ArtGS holding a 1min edge only on Fridge and Stapler. Despite incorporating additional part-awareness and physical constraints, our method still matches ArtGS's 8-minute inference performance on most complex objects (and only modestly increases to 10 minutes on the highest-complexity case, Storage_7). Overall, Part^{2}GS delivers state-of-the-art efficiency even with its extra inferential overhead.

D. Additional Ablations

D.1. Sensitivity Ablation

In Table 8, we further perform a module-removal ablation to quantify the sensitivity of Part^{2}GS to each design component. Starting from the full Part^{2}GS model, we sequentially disable part parameters, repel points, physical constraints, and canonical initialization.

Removing the **part parameters** leads to the most severe (three orders of magnitude) degradation across both objects. MotionErr increases by more than $700\times$ ($0.01\rightarrow7.32$) and $\text{CD}_{\text{movable}}$ by $\sim78\times$ ($1.85\rightarrow145.17$) on the 5-part Table object. On the 7-part Storage object, MotionErr rises $\sim19\times$ ($0.55\rightarrow10.43$) and $\text{CD}_{\text{movable}}$ increases over $100\times$ ($1.83\rightarrow198.67$). Angular and motion errors spike dramatically (*e.g.,* Ang Err from 0.03 to 0.21 and Motion Err from 0.01 to 7.32 on the Table object), while $\text{CD}_{\text{movable}}$ skyrockets by over $70\times$. This confirms that semantic part disentanglement is essential for stable articulation and coherent geometry recovery. Without explicit part identity supervision, the model fails to isolate and track distinct motions, leading to collapsed or entangled reconstructions.

Disabling the **repel points** has a noticeable effect on motion accuracy but limited influence on geometry quality. On the Table object, motion error increases nearly $50\times$ (from 0.01 to 0.48), while angular and positional errors also rise, suggesting that the lack of inter-part repulsion leads to ambiguity in part-specific transformations. However, CD_{whole} remains relatively stable, confirming that the Gaussian reconstruction itself is unaffected.

Table 9. **Ablations on physics-informed regularization**, on the two most complex objects in our evaluation, Table (5 parts) and Storage (7 parts). Lower (↓) is better on all metrics. highlights the best results.

Objects	Methods	AngErr	PosErr	MotionErr	CD_{static}	$CD_{movable}$	CD_{whole}
Table (5 parts)	**no physical constraints**	0.05	0.03	0.18	1.32	4.47	1.65
	+ contact loss	0.05	0.02	0.17	1.18	1.78	1.22
	+ velocity consistency	0.03	0.01	0.02	1.33	3.11	1.52
	+ vector-field alignment (Part^{2}GS)	0.03	0.00	0.01	1.22	2.22	1.41
Storage (7 parts)	**no physical constraints**	0.04	0.05	0.04	1.22	4.54	1.12
	+ contact loss	0.05	0.04	0.04	0.96	2.12	0.74
	+ velocity consistency	0.06	0.04	0.04	1.21	4.01	0.62
	+ vector-field alignment (Part^{2}GS)	0.03	0.04	0.04	1.22	3.56	0.71

Table 10. **Part^{2}GS performance by transformation type.** Evaluation across objects undergoing only translation or only rotation motions. Lower (↓) is better for all metrics.

Category	Objects	Ang	Pos	Motion	CD_{static}	$CD_{movable}$	CD_{whole}
Translation	Blade (2 parts)	0.01	–	0.00	0.03	0.06	0.04
	Storage (2 parts)	0.01	–	0.00	0.04	0.04	0.04
	Table (5 parts)	0.03	–	0.00	0.56	1.95	0.51
	Average	0.02	–	0.00	0.21	0.68	0.20
Rotation	Laptop (2 parts)	0.01	0.00	0.01	0.07	0.09	0.08
	Fridge (3 parts)	0.01	0.00	0.02	0.59	0.08	0.73
	Oven (4 parts)	0.03	0.01	0.18	1.01	0.11	0.95
	Average	0.02	0.00	0.07	0.56	0.09	0.59

Table 11. **Robustness to noisy repel-point initialization.** Lower (↓) is better on all metrics. highlights the best results.

Metric	σ_r	Foldchair (2 parts)	Stapler (2 parts)	Blade (2 parts)	Oven (4 parts)	Table (5 parts)	Storage (7 parts)
Ang Err ↓	0.00	0.01	0.01	0.01	0.03	0.30	0.11
	0.01	0.01	0.02	0.01	0.04	0.31	0.12
	0.03	0.02	0.03	0.02	0.06	0.34	0.14
	0.05	0.03	0.04	0.03	0.08	0.37	0.17
Pos Err ↓	0.00	0.00	0.01	-	0.01	0.00	0.01
	0.01	0.00	0.01	-	0.01	0.01	0.01
	0.03	0.01	0.02	-	0.02	0.02	0.02
	0.05	0.02	0.03	-	0.03	0.03	0.03
Motion Err ↓	0.00	0.01	0.00	0.00	0.18	0.01	0.55
	0.01	0.01	0.01	0.01	0.19	0.02	0.58
	0.03	0.02	0.02	0.02	0.23	0.03	0.64
	0.05	0.03	0.03	0.03	0.28	0.04	0.72
CD_{whole} ↓	0.00	0.19	1.45	0.35	0.95	1.10	0.63
	0.01	0.20	1.46	0.36	0.96	1.12	0.65
	0.03	0.21	1.48	0.38	0.98	1.14	0.67
	0.05	0.23	1.51	0.40	1.00	1.18	0.71

The **physical constraints** contribute moderate improvements, particularly in reducing $CD_{movable}$ and motion error. On both objects, removing these constraints leads to visible but not catastrophic performance drops (*e.g.,* Pos Err from 0.01 to 0.05 and $CD_{movable}$ from 1.83 to 4.54 on Storage), indicating that they provide useful geometric regularization but are not the sole factor in driving accuracy.

Finally, removing **canonical initialization** results in the most unstable training behavior. Angular error explodes from 0.11 to 22.15 on Storage, and motion error increases by over 35× on both objects. Results highlight the importance of starting from a stable, geometry-aligned canonical state to enable robust part tracking and learning.

D.2. Ablation on Physics-Informed Losses

We additionally perform ablations to quantify the impact of each physical constraint. As shown in Table 9, each physical loss meaningfully contributes to improved motion accuracy and geometry quality. **Contact loss** leads to the largest drop in geometry errors. For instance, on the Table object, which exhibits multi-axis, rotational articulation, contact loss cuts $CD_{movable}$ by more than half (4.47→1.78) and CD_{whole} by 26% (1.65→1.22), indicating far less interpenetration and more realistic results. **Velocity consistency** improves motion quality, nearly eliminating motion errors (*e.g.,* reducing Motion Err from 0.18 to 0.02). **Vector-field alignment** yields the lowest angular and positional errors, driving errors down across the board and yielding the most physically plausible, accurate articulations overall. These results demonstrate that the proposed physical constraints act in complementary ways to enable physically plausible, precise articulation and geometry reconstruction. Storage (7 parts) shows reduced inter-part penetration ($CD_{movable}$: 4.54→2.12, CD_{whole}: 1.12→0.74), while motion errors remain nearly unchanged (MotionErr = 0.04). Here, the baseline motion is already simple and prismatic, so the constraints primarily enforce geometric separation rather than further reducing dynamic error. Overall, these results indicate that the proposed constraints provide a consistent and interpretable improvement in both physical plausibility and geometric fidelity, particularly for complex, multi-axis articulations.

D.3. Translation vs. Rotation Ablation

We provide an ablation analysis for translation-only and rotation-only objects. Table 10 results show that Part^{2}GS achieves consistently low error across both motion types. Notably, objects with pure translation exhibit near-zero motion errors and lower average CD metrics, reflecting the relative simplicity of prismatic articulation. Rotational objects maintain low error as well, but with slightly higher averages due to increased articulation complexity. We also observe that rotational objects tend to have higher CD values compared to translational objects (*e.g.,* Avg. CD_{whole}: 0.59 vs. 0.20), likely due to increased geometric complexity.

D.4. Noisy Repel Points Initialization

To evaluate sensitivity to repel-point initialization, we perturb the initially generated repel points with small random 3D offsets with magnitude σ_r (*e.g.,* $\sigma_r = 0.01$ corresponds to ∼1% of the object's spatial extent). Table 11 shows performance remains stable under moderate noise.

Table 12. **Repel points robustness.** We compare **Fixed** repel points with a **Dynamic** variant that refreshes them every K=5k iterations. **Clean Init** uses default repel points; **Noisy Init** perturbs them before optimization (*e.g.*, σ_r=0.05). highlights best results.

Metric	Setting	Foldchair (2 parts)	Stapler (2 parts)	Blade (2 parts)	Oven (4 parts)	Table (5 parts)	Storage (7 parts)
Ang Err ↓	Clean + Fixed	0.01	0.01	0.01	0.03	0.30	0.11
	Clean + Dynamic	0.01	0.01	0.01	0.03	0.30	0.11
	Noisy + Fixed	0.03	0.04	0.03	0.08	0.37	0.17
	Noisy + Dynamic	0.03	0.04	0.03	0.08	0.35	0.17
Pos Err ↓	Clean + Fixed	0.00	0.01	-	0.01	0.00	0.01
	Clean + Dynamic	0.00	0.01	-	0.01	0.00	0.01
	Noisy + Fixed	0.02	0.03	-	0.03	0.03	0.03
	Noisy + Dynamic	0.02	0.03	-	0.03	0.02	0.03
Motion Err ↓	Clean + Fixed	0.01	0.00	0.00	0.18	0.01	0.55
	Clean + Dynamic	0.01	0.00	0.00	0.18	0.01	0.54
	Noisy + Fixed	0.03	0.03	0.03	0.28	0.04	0.72
	Noisy + Dynamic	0.03	0.03	0.03	0.26	0.04	0.69
CD_{whole} ↓	Clean + Fixed	0.19	1.45	0.35	0.95	1.10	0.63
	Clean + Dynamic	0.19	1.45	0.35	0.95	1.09	0.63
	Noisy + Fixed	0.23	1.51	0.40	1.00	1.18	0.71
	Noisy + Dynamic	0.22	1.43	0.39	0.99	1.16	0.69

Table 13. **Sensitivity to the number of parts.** K_{GT} denotes ground-truth number of parts K. highlights the best results.

Metric	K Setting	Foldchair (2 parts)	Stapler (2 parts)	Blade (2 parts)	Oven (4 parts)	Table (5 parts)	Storage (7 parts)
Ang Err ↓	K_{GT}	0.01	0.01	0.01	0.03	0.03	0.11
	$K_{\text{GT}}+2$	0.01	0.01	0.01	0.03	0.04	0.11
	$K_{\text{GT}}+4$	0.02	0.02	0.01	0.04	0.05	0.12
Pos Err ↓	K_{GT}	0.00	0.01	-	0.01	0.00	0.01
	$K_{\text{GT}}+2$	0.00	0.01	-	0.01	0.01	0.01
	$K_{\text{GT}}+4$	0.01	0.01	-	0.02	0.01	0.02
Motion Err ↓	K_{GT}	0.01	0.00	0.00	0.18	0.01	0.55
	$K_{\text{GT}}+2$	0.01	0.01	0.00	0.19	0.02	0.57
	$K_{\text{GT}}+4$	0.02	0.01	0.01	0.22	0.03	0.60
CD_{whole} ↓	K_{GT}	0.19	1.45	0.35	0.95	1.10	0.63
	$K_{\text{GT}}+2$	0.20	1.46	0.36	0.96	1.12	0.65
	$K_{\text{GT}}+4$	0.22	1.49	0.38	0.99	1.15	0.68

D.5. Fixed vs. Dynamic Repel Points

We compare fixed repel points with a dynamic variant that recomputes them during training. As shown in Table 12, the results are nearly identical overall, and dynamic updates provide only minor gains under noisy initialization, confirming that fixed repel points are generally sufficient and already offer a stable choice in practice.

D.6. Part Number (K) Selection

We follow standard practice in articulated modeling and set K to the number of movable parts for fair comparison with prior work, while treating it as an upper bound in practice. Beyond the mis-specification study in Table 4, we further examine a practically relevant regime of mild over-estimation in Table 13, comparing K_{GT} against $K_{\text{GT}}+2$ and $K_{\text{GT}}+4$. Results show that Part^{2}GS remains robust when K is moderately over-specified, with only small changes in articulation and reconstruction quality. Using $K_{\text{GT}}+2$ generally preserves performance across angular error, positional error, motion error, and CD_{whole}. For example, on TABLE and STORAGE, the whole-object Chamfer Distance changes only from 1.10 → 1.12 and 0.63 → 0.65, respectively. Even with $K_{\text{GT}}+4$, performance degrades only modestly on more complex objects, suggesting that redundant part slots are largely suppressed during optimization rather than causing catastrophic failure.

Table 14. **Repel-Force Ablation.** Results averaged over all objects.

Exponent p	Motion Err ↓	CD_{whole} ↓	Penetration ↓
2	0.028	0.69	0.021
3	0.020	0.66	0.009
4	0.023	0.67	0.012

Table 15. **Photometric evaluation.** Metrics averaged over observation states. highlights best performing results.

Metric	Method	Foldchair (2 parts)	Stapler (2 parts)	Blade (2 parts)	Oven (4 parts)	Table (5 parts)	Storage (7 parts)
PSNR ↑	ArtGS	32.4	33.1	31.7	30.2	29.6	28.7
	Part^{2}GS	**33.6**	**34.2**	**32.9**	**31.4**	**30.8**	**29.9**
SSIM ↑	ArtGS	0.968	0.972	0.961	0.950	0.942	0.934
	Part^{2}GS	**0.975**	**0.979**	**0.970**	**0.959**	**0.951**	**0.944**
LPIPS ↓	ArtGS	0.041	0.039	0.047	0.058	0.066	0.072
	Part^{2}GS	**0.035**	**0.033**	**0.040**	**0.051**	**0.059**	**0.064**

D.7. Repel-Force Exponent Ablation

We employ $\|\mathbf{r}-\boldsymbol{\mu}\|^3$ in Equation (7) so that the resulting repulsion vector has an inverse-square magnitude, *i.e.*, $\|\mathbf{F}\| \propto 1/d^2$ with $d=\|\mathbf{r}-\boldsymbol{\mu}\|$, while preserving its direction toward the repel point. In Table 14, we ablate the falloff exponent p in $\mathbf{F} \propto (\mathbf{r}-\boldsymbol{\mu})/\|\mathbf{r}-\boldsymbol{\mu}\|^p$ and observe that $p=3$ provides the best trade-off between preventing interpenetration and maintaining accurate motion and geometry.

E. Photometric Evaluation

We additionally report photometric metrics averaged over both observation states. As shown in Table 15, Part^{2}GS consistently outperforms ArtGS across all objects and all three metrics, indicating more accurate pixel-level reconstruction and improved perceptual quality. These gains are consistent across simpler and more challenging multi-part objects.

F. Broader Impacts

The ability to accurately reconstruct and articulate 3D objects has far-reaching implications across robotics, simulation, and digital twin technologies. Part^{2}GS contributes to this space by enabling precise, physically grounded modeling of complex articulated objects from visual observations. This can facilitate improved interaction and manipulation in embodied agents, enhance simulation fidelity in virtual environments, and support scalable generation of articulated assets for digital content creation, industrial, and educational applications. While the ability to digitize and manipulate real-world objects raises potential concerns around privacy, intellectual property, or misuse in synthetic media, our model is designed for research and educational use. We encourage responsible deployment practices aligned with consent and attribution norms. Compared to large-scale generative systems, our model is computationally lightweight and environmentally efficient, and we view its benefits in controllable, interpretable object modeling as outweighing its risks when applied ethically.