

SPLART: Articulation Estimation and Part-Level Reconstruction with 3D Gaussian Splatting

Shengjie Lin[†], Jiading Fang[†], Muhammad Zubair Irshad[‡], Vitor Campagnolo Guizilini[‡],
Rares Andrei Ambrus[‡], Greg Shakhnarovich^{†‡}, Matthew R. Walter[†]

[†] Toyota Technological Institute at Chicago, [‡] Toyota Research Institute

{slin, fjd, greg, mwalter}@ttic.edu, {zubair.irshad, vitor.guizilini, rares.ambrus}@tri.global

Abstract

Reconstructing articulated objects prevalent in daily environments is crucial for applications in augmented/virtual reality and robotics. However, existing methods face scalability limitations (requiring 3D supervision or costly annotations), robustness issues (being susceptible to local optima), and rendering shortcomings (lacking speed or photorealism). We introduce SPLART, a self-supervised, category-agnostic framework that leverages 3D Gaussian Splatting (3DGS) to reconstruct articulated objects and infer kinematics from two sets of posed RGB images captured at different articulation states, enabling real-time photorealistic rendering for novel viewpoints and articulations. SPLART augments 3DGS with a differentiable mobility parameter per Gaussian, achieving refined part segmentation. A multi-stage optimization strategy is employed to progressively handle reconstruction, part segmentation, and articulation estimation, significantly enhancing robustness and accuracy. SPLART exploits geometric self-supervision, effectively addressing challenging scenarios without requiring 3D annotations or category-specific priors. Evaluations on established and newly proposed benchmarks, along with applications to real-world scenarios using a hand-held RGB camera, demonstrate SPLART’s state-of-the-art performance and real-world practicality. Code is publicly available at <https://github.com/ripl/splart>.

1. Introduction

Articulated objects, such as drawers, doors, and scissors, are ubiquitous in our daily lives, yet their dynamic nature poses significant challenges for 3D reconstruction—a critical task for applications in augmented/virtual reality [37, 52], robotics [1, 5, 16, 24, 51, 58], and computer vision [20, 46]. Existing methods for reconstructing articulated objects are hindered by several key limita-

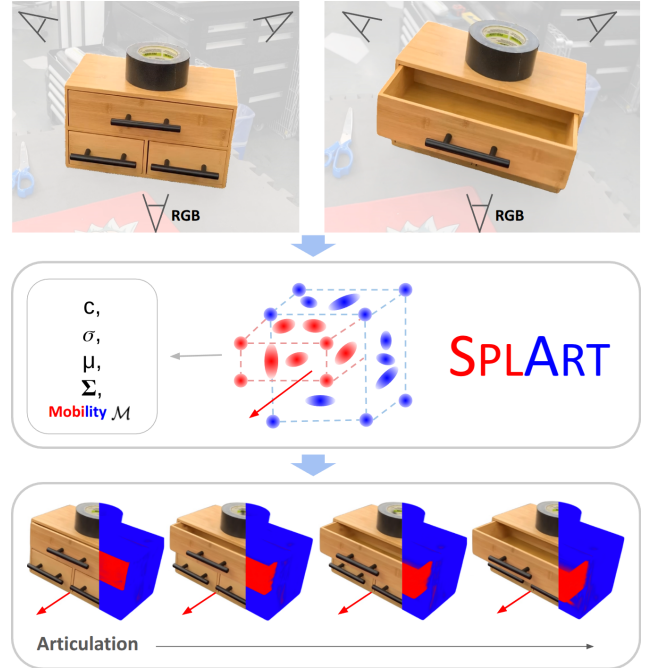


Figure 1. Given (top) RGB images of an object at two articulation states, (middle) SPLART uses 3DGS to simultaneously reconstruct its static and dynamic parts and estimate the kinematic articulation model. SPLART is then able to (bottom) render high-fidelity 3D reconstructions of the object along with part-level segmentations for novel articulation states, allowing for novel view synthesis.

tions: they often require labor-intensive supervision (e.g., part-level segmentation or articulation annotations) [23, 29, 41, 55, 57], they depend on 3D supervision that restricts practical use [23, 29, 34, 41, 61], they produce category-specific models that limit scalability [29, 41, 55, 60], or they are not capable of real-time, photorealistic rendering [6, 13, 23, 29, 30, 41, 55, 57, 60, 61]. To address these challenges, we introduce SPLART, a novel self-supervised and category-agnostic framework that leverages 3D Gaussian Splatting (3DGS) [26] to reconstruct articulated ob-

jects from minimal input—two sets of posed RGB images at distinct articulation states. SPLART reconstructs object parts and infers kinematics, enabling real-time, photorealistic rendering for novel views and articulation states.

Central to SPLART is the augmentation of 3DGS [26] to include a differentiable mobility parameter for each Gaussian, which enables a more refined segmentation of static and mobile parts through gradient-based optimization. This results in enhanced reconstruction quality, while preserving the real-time, photorealistic rendering capabilities of 3DGS—offering a speedup of more than 100× over methods [6, 30] based on neural radiance fields [39].

To enhance robustness, SPLART employs a multi-stage optimization strategy that decouples the part-level reconstruction and articulation estimation processes. Unlike end-to-end approaches prone to local optima [30], SPLART first independently reconstructs each articulation state, then estimates each Gaussian’s mobility parameter for part segmentation, and finally refines both the articulation and mobility estimates jointly. This structured approach ensures stable and accurate convergence, avoiding the stringent initialization requirements of existing methods, thereby providing a practical solution for challenging articulated structures.

Building on this foundation, SPLART leverages geometric self-supervision to eliminate the need for manual annotations or 3D supervision. By enforcing geometric consistency between reconstructions through complementary loss formulations, SPLART robustly estimates articulation parameters across diverse scenarios. This self-supervised strategy enhances scalability, enabling SPLART to reconstruct a wide range of articulated objects without relying on prior structural or categorical knowledge.

Extensive evaluations on both established and newly introduced benchmarks demonstrate SPLART’s superior articulation accuracy and reconstruction quality, surpassing state-of-the-art methods without requiring 3D supervision. Real-world experiments further validate its practicality, showcasing successful reconstructions of diverse articulated objects using only a handheld RGB camera.

In summary, this work contributes:

1. An extension of 3DGS with a differentiable mobility value per Gaussian that enables precise part segmentation using gradient-based optimization.
2. A multi-stage optimization strategy that decouples reconstruction and articulation estimation, enhancing robustness and accuracy.
3. Complementary formulations of geometric self-supervision for articulation estimation, eliminating the need for 3D supervision or laborious annotations.
4. A challenging dataset and new metrics for comprehensive evaluation of articulated object reconstruction.

2. Related Work

2.1. Data-Driven Articulation Learning

Estimating the pose and joint properties of articulated objects is crucial for robot manipulation and interaction [10, 11, 19, 31]. Recent learning-based methods [9, 13, 15, 22, 29, 33, 57, 65] infer articulation properties from point clouds via end-to-end training. For instance, Shape2Motion [57] analyzes motion parts from a single point cloud in a supervised setting, while ANCSH [29] performs category-level pose estimation but requires class-specific models. RPM-Net [63] enhances generalization across categories for part segmentation and kinematic prediction, and DITTO [23] predicts motion and geometry from 3D point cloud pairs without labels. However, these methods depend on costly 3D supervision and annotations. In contrast, our approach reconstructs accurate 3D geometry and detailed appearance, capturing articulation without 3D supervision or priors.

2.2. Representations for Object Reconstruction

Early 3D object reconstruction methods predicted point clouds, voxels, or meshes from partial observations [3, 8, 17]. Recent advances in implicit scene representations [18, 26, 38, 39, 45, 66] enable detailed geometry and appearance reconstruction via differentiable rendering [12, 44, 53, 56]. While neural fields suffer from slow rendering, 3D Gaussian Splatting (3DGS) [26] overcomes this by using explicit 3D Gaussians. We leverage 3DGS for self-supervised articulated object reconstruction from posed RGB images, achieving fast, realistic synthesis of novel views and articulations in real time.

2.3. Articulated Object Reconstruction

Recent methods leverage differentiable 3D representations [26, 39, 45] to jointly reconstruct articulated objects and infer articulation parameters. Training-based approaches use synthetic 3D data to predict joint parameters and segment parts [13, 14, 23, 25, 41, 43, 60]. Self-supervised methods optimize shape, appearance, and articulation per scene without extensive training [6, 27, 30, 32, 34, 50, 61], with some addressing multi-part objects but requiring known part counts and single-level articulation structures [6, 34, 61]. Other works enhance articulation estimation using large language or vision-language models [28, 36]. In contrast, our self-supervised method reconstructs two-part articulated objects from RGB images across articulation states using 3DGS [26]. As the first to apply 3DGS to this task without 3D supervision or pre-trained priors, it robustly handles challenging cases and achieves real-time performance.

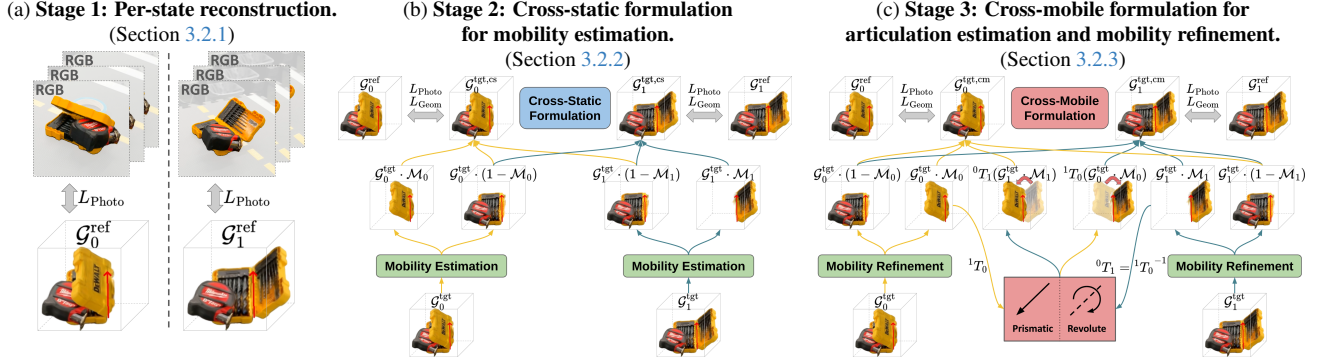


Figure 2. **Methodology overview of SPLART.** It consists of three decoupled stages for the purpose of stable optimization. **Stage 1** constructs a 3DGS for each state from posed RGB images using the photometric loss. **Stage 2** proposes cross-static formulation for mobility estimation. Intuitively, it combines static parts of both states and the mobile part of the desired state for the target Gaussians. **Stage 3** proposes cross-mobile formulation for articulation estimation. Intuitively, it combines static parts of both states, the mobile part of the desired state and the *transformed* mobile part of the other state together for the target Gaussians.

3. Methodology

3.1. Overview

Consider an arbitrary object composed of two rigid parts: a static *parent* part, and a *child* part that can move relative to its parent through either a revolute or prismatic articulation. Our objective is twofold: (1) to reconstruct the articulated object at the part level; and (2) to estimate its articulated motion. Assuming a known articulation type (i.e., either revolute or prismatic), the input to our method consists of two sets of posed RGB images (i.e., images with known camera intrinsics and extrinsics), each capturing the articulated object at one end state of the motion.

Formally, let l denote the articulation state label, where $l = 0$ and $l = 1$ correspond to the two end states of the observed articulation. For reconstruction, SPLART uses observations $\mathcal{O}_l = \{(I_l^i, P_l^i, K_l^i)\}_{i=1}^{N_l}$, $l \in \{0, 1\}$, where I_l^i is the i -th observed RGB image of the articulated object at state l , P_l^i and K_l^i denote its camera extrinsics and intrinsics respectively, and N_l represents the number of data samples for state l . Note that P_l^i is specified in a common world space for both states, while the articulated motion involves only one moving part w.r.t. the world space. SPLART models a revolute articulation with its rotation axis \mathbf{a} ($\|\mathbf{a}\| = 1$), pivot \mathbf{p} , and rotation angle θ , such that a point \mathbf{x} (in the world space) on the mobile part at state $l = 0$ will move to

$${}^1T_0\mathbf{x} = R_{\mathbf{a},\theta}(\mathbf{x} - \mathbf{p}) + \mathbf{p} \quad (1)$$

at state $l = 1$, where $R_{\mathbf{a},\theta}$ is the rotation induced by the axis-angle notation. A prismatic articulation is instead modeled by its translation axis \mathbf{a} ($\|\mathbf{a}\| = 1$) and distance d . The goal is to reconstruct the articulated object at the part level using a chosen representation while estimating the articulated motion 1T_0 , ensuring that the renderings at each articulation state align consistently with the observations.

SPLART extends the 3DGS representation for articulated objects, decoupling the goals of part-level reconstruction and articulation estimation across three stages: (1) separate reconstructions for each articulation state (Sec. 3.2.1), (2) mobility estimation using the cross-static formulation (Sec. 3.2.2), and (3) articulation estimation and mobility refinement using the cross-mobile formulation (Sec. 3.2.3). To facilitate the application of SPLART to real-world objects, we leverage modern structure-from-motion and image segmentation techniques, developing a framework that enables general users to reconstruct articulated objects in their surroundings using only images captured by a hand-held camera device (Sec. 3.3).

3.2. Decoupled Multi-Stage Optimization

Jointly performing part-level reconstruction and articulation estimation on an articulated object is prone to local minima [6, 30], making it preferable to strategically decouple the problem into multiple stages.

3.2.1. Separate per-state reconstruction

In Stage 1, two 3DGS models are separately optimized following the standard procedure, one for each end state of the articulation. Specifically, apart from the attributes from original 3DGS, each Gaussian is additionally initialized with a persistent binary state label l , equally drawn from $\{0, 1\}$. We denote the set of Gaussians representing state l as $\mathcal{G}_l^{\text{ref}}$, where *ref* emphasizes that \mathcal{G}^{ref} is the reference reconstruction unaffected by the other state. Given a data sample observed at state l , we have the optimization:

$$\min_{\mathcal{G}_l^{\text{ref}}} \Delta_I(\hat{I}_l^{\text{ref},i}, I_l^i), \quad (2a)$$

$$\hat{I}_l^{\text{ref},i} = R(\mathcal{G}_l^{\text{ref}}, P_l^i, K_l^i), \quad l \in \{0, 1\}, \quad (2b)$$

where R is the 3DGS rendering function, and Δ_I represents the photometric loss. For simplicity, the view index i and camera parameters P_l^i, K_l^i will be omitted from now on.

3.2.2. Cross-static formulation for mobility estimation

The focus of Stage 2 is mobility estimation for each Gaussian. To keep \mathcal{G}^{ref} dedicated to the single-state reconstruction, we first duplicate $\mathcal{G}_l^{\text{ref}}$ as $\mathcal{G}_l^{\text{tgt}}$ for both states (i.e., $l \in \{0, 1\}$), which is intended as the target representation that will fulfill the goals of part-level reconstruction and articulation estimation. By design, \mathcal{G}^{tgt} shares neither data storage nor gradient flow with \mathcal{G}^{ref} after its creation.

For each Gaussian in \mathcal{G}^{tgt} , we further extend its set of attributes with a mobility value $m \in [0, 1]$, initialized with 0.5. By design, m enables the break-down of a Gaussian to its static and mobile components, where each component inherits all the original Gaussian attributes except for the opacity σ . The static component has its opacity modified to $\sigma \cdot (1 - m)$, and the mobile component to $\sigma \cdot m$. For simplicity, letting \mathcal{M} be the set of mobilities for \mathcal{G} , we use the element-wise product $\mathcal{G} \cdot (1 - \mathcal{M})$ to denote the static component of \mathcal{G} , and $\mathcal{G} \cdot \mathcal{M}$ for the mobile component.

To estimate the mobilities \mathcal{M} , we employ the intuition that the static components from both states should constitute the static part of the articulated object. Formally, we introduce the *cross-static (cs) formulation*, where the static part of the articulated object is jointly represented as

$$\mathcal{G}^s = \mathcal{G}_l^{\text{tgt}} \cdot (1 - \mathcal{M}_l) \oplus \mathcal{G}_{1-l}^{\text{tgt}} \cdot (1 - \mathcal{M}_{1-l}), \quad (3)$$

where \oplus denotes concatenation. For state l , the target representation thus becomes

$$\mathcal{G}_l^{\text{tgt}, \text{cs}} = \mathcal{G}^s \oplus \mathcal{G}_l^{\text{tgt}} \cdot \mathcal{M}_l. \quad (4)$$

With this formulation, Stage 2 is further divided into two sub-stages as below.

Stage 2(a): Coarse mobility estimation via cross-static geometric consistency. To measure the geometric distance between two Gaussian sets, we design a weighted version of the Chamfer distance. Specifically, let $X = \{(x_i, w_{x_i})\}_{i=1}^M$ and $Y = \{(y_j, w_{y_j})\}_{j=1}^N$ be two sets of point-weight pairs, the weighted Chamfer distance is then:

$$\begin{aligned} \text{Chamfer}(X, Y) = & \sum_{(x_i, w_{x_i}) \in X} \tilde{w}_{x_i} \min_{(y_j, w_{y_j}) \in Y} \|x_i - y_j\|^2 \\ & + \sum_{(y_j, w_{y_j}) \in Y} \tilde{w}_{y_j} \min_{(x_i, w_{x_i}) \in X} \|x_i - y_j\|^2, \end{aligned} \quad (5)$$

where $\tilde{w}_{x_i} = w_{x_i} / \sum_{(x_i, w_{x_i}) \in X} w_{x_i}$ and \tilde{w}_{y_j} are normalized weights. For each Gaussian, we use its mobility-modified opacity σ' as the weight, essentially treating it as σ' points

overlapped at its mean position. The mobilities are then optimized by encouraging the geometric consistency formulated as follows:

$$\min_{\mathcal{M}} \text{CD}_0^{\text{cs}} + \text{CD}_1^{\text{cs}} + \lambda_m^{\text{geom}} \|\mathcal{M}\|, \quad (6a)$$

$$\text{CD}_l^{\text{cs}} = \text{Chamfer}(\mathcal{G}_l^{\text{tgt}, \text{cs}}, \mathcal{G}_l^{\text{ref}}), \quad (6b)$$

where $\text{Chamfer}(\cdot)$ denotes the weighted Chamfer distance, and $\lambda_m^{\text{geom}} \|\mathcal{M}\|$ is the regularization term that encourages smaller mobilities. Note how \mathcal{M} affects the weighted Chamfer distance by modifying the opacities, and that $\mathcal{M} = 1$ is a trivial solution without the regularization. Without photometric supervision, the mobilities obtained from Eqn. 6a are generally noisy. However, being relatively fast (taking only tens of seconds), they still serve as a good initialization for the next sub-stage.

Stage 2(b): Joint mobility and Gaussian optimization via cross-static rendering. To more accurately estimate the mobilities while jointly refining the full Gaussian attributes, cross-static rendering is performed as follows:

$$\min_{\mathcal{G}^{\text{tgt}}, \mathcal{M}_{1-l}} \Delta_I(\hat{I}_l^{\text{cs}}, I_l) + \lambda_m^{\text{photo}} \|\mathcal{M}_{1-l}\|, \quad (7a)$$

$$\hat{I}_l^{\text{cs}} = R(\mathcal{G}_l^{\text{tgt}, \text{cs}}), \quad (7b)$$

where $\lambda_m^{\text{photo}} \|\mathcal{M}_{1-l}\|$ is the mobility regularization term similar to that in Eqn. 6a.

3.2.3. Cross-mobile formulation for articulation estimation and mobility refinement

The focus of Stage 3 is to estimate the articulation parameters shared by the mobile components of all Gaussians. To this end, we employ the intuition that the mobile components from the two end states are related through the articulated motion. Formally, we introduce the *cross-mobile (cm) formulation*, where the mobile part of the articulated object at state l is jointly represented as

$$\mathcal{G}_l^m = \mathcal{G}_l^{\text{tgt}} \cdot \mathcal{M}_l \oplus {}^l T_{1-l}(\mathcal{G}_{1-l}^{\text{tgt}} \cdot \mathcal{M}_{1-l}), \quad (8)$$

where ${}^l T_{1-l}(\mathcal{G})$ denotes the transformation of Gaussians \mathcal{G} according to the articulated motion from state $1-l$ to l . For state l , the target representation thus becomes

$$\mathcal{G}_l^{\text{tgt}, \text{cm}} = \mathcal{G}^s \oplus \mathcal{G}_l^m, \quad (9)$$

where \mathcal{G}^s is defined in Eqn. 3. With this formulation, Stage 3 is further divided into three sub-stages as below.

Stage 3(a): Coarse articulation estimation via geometric consistency. Similar to Eqn. 6a, weighted Chamfer distance is used for computing the cross-mobile geometric consistency, which can be optimized over both mobilities and articulation parameters as follows:

$$\min_{\mathcal{M}, T} \text{CD}_0^{\text{cm}} + \text{CD}_1^{\text{cm}}, \quad (10a)$$

$$\text{CD}_l^{\text{cm}} = \text{Chamfer}(\mathcal{G}_l^{\text{tgt}, \text{cm}}, \mathcal{G}_l^{\text{ref}}). \quad (10b)$$

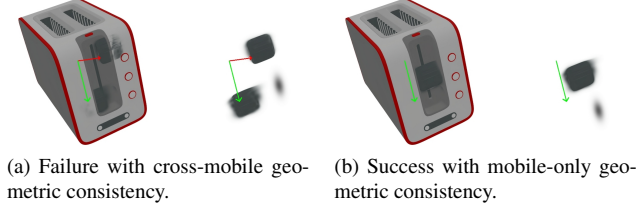


Figure 3. 103549-Toaster has a flat slider as its mobile part, similar in geometric curvature to the static component. This similarity makes the nearest-neighbor data association error-prone when computing the Chamfer distance for cross-mobile geometric consistency. On the other hand, mobile-only geometric consistency focuses only on the mobile parts, successfully handling the case.

However, we notice that this formulation is still susceptible to local minima if the mobile part is too geometrically insignificant. We qualitatively show one such failure in Fig. 3a. To remedy this, we further propose mobile-only geometric consistency, which concerns only the mobile components of both states as follows:

$$CD^m = \text{Chamfer}(\mathcal{G}_l^{\text{tgt}} \cdot \mathcal{M}_l, {}^lT_{1-l}(\mathcal{G}_{1-l}^{\text{tgt}} \cdot \mathcal{M}_{1-l})). \quad (11)$$

While successfully addressing these scenarios, mobile-only geometric consistency still falls short in other circumstances, especially when the mobile components in the two states exhibit large discrepancies. We qualitatively show one such failure in Fig. 4a. To take advantage of both formulations and to facilitate robustness in the inherently non-convex optimization of articulation parameters, we propose the following practical scheme:

1. With K^m randomized tries: articulation estimation via

$$T^m = \arg \min_T CD^m. \quad (12)$$

2. With K^{cm} randomized tries, plus another initialized with T^m : articulation estimation via

$$T^{\text{cm}} = \arg \min_T CD_0^{\text{cm}} + CD_1^{\text{cm}}. \quad (13)$$

3. Final run initialized with T^{cm} : joint articulation estimation and mobility refinement via Eqn. 10a.

Stage 3(b): Joint articulation, mobility, and Gaussian optimization via cross-mobile rendering. Like in Stage 2(b), we perform a full joint optimization over articulation parameters, mobilities, and Gaussians utilizing photometric supervision via cross-mobile rendering as follows:

$$\min_{\mathcal{G}_l^{\text{tgt}}, \mathcal{M}_{1-l}, {}^lT_{1-l}} \Delta_I(\hat{I}_l^{\text{cm}}, I_l), \quad (14a)$$

$$\hat{I}_l^{\text{cm}} = R(\mathcal{G}_l^{\text{tgt}, \text{cm}}), \quad l \in \{0, 1\}. \quad (14b)$$



Figure 4. 101713-Pen has both ends of a pen as the mobile part, with only one end visible from any given observation. Thus, the reconstructed mobile components from the two states essentially represent distinct parts, violating the intuition for mobile-only geometric consistency. On the other hand, cross-mobile geometric consistency considers the joint representation from both states as a whole, successfully estimating the articulation.

Figure 4. 101713-Pen has both ends of a pen as the mobile part, with only one end visible from any given observation. Thus, the reconstructed mobile components from the two states essentially represent distinct parts, violating the intuition for mobile-only geometric consistency. On the other hand, cross-mobile geometric consistency considers the joint representation from both states as a whole, successfully estimating the articulation.

Stage 3(c): Mobility correction via cross-mobile geometric consistency. Stage 3(b) solely relies on photometric supervision, which is limited to the training views. On rare occasions, a Gaussian may be mistakenly estimated as mobile if the articulated motion moves it out of sight from most views, effectively losing the supervision. On the other hand, geometric consistency is not affected by sight limitations, which we leverage for mobility correction as follows:

$$\min_{\mathcal{M}} CD_0^{\text{cm}} + CD_1^{\text{cm}}. \quad (15)$$

3.3. Real-World Application

For an articulated object to be reconstructed in the real world, we first collect two sets of RGB images, one for each articulation state, by imaging the object from a surrounding hemisphere. We then preprocess the data to determine the posed images that serve as the input to SPLART. This involves using SAM 2 [47] to perform foreground-background segmentation, which also removes dynamic contents of the scene. We then perform structure-from-motion (SfM) to determine camera poses using COLMAP [49] with SuperPoint [7] descriptors and SuperGlue [48] matching on the segmented backgrounds from both sets of images to construct a joint coordinate frame for the object in both states. Once we obtain the joint coordinate frame and the foreground target object is localized, we run SPLART to reconstruct the articulated objects.

4. Experiments

4.1. Datasets

PARIS PartNet-Mobility Subset. PartNet-Mobility is a large-scale dataset that provides simulatable 3D object models with part-level mobility [2, 40, 62], from which PARIS [30] selects 10 instances for experiments, 8 being

Type	Method	Success Rate \uparrow	err_a \downarrow ($\times 10^{-2}$ DEG)	err_p \downarrow ($\times 10^{-3}$)	err_r \downarrow ($\times 10^{-2}$ DEG)	err_t \downarrow ($\times 10^{-3}$)	CD_s \downarrow ($\times 10^{-3}$)	CD_m \downarrow ($\times 10^{-3}$)	CD_w \downarrow ($\times 10^{-3}$)
Revolute	PARIS	40.0%	121.90 \pm 68.45	5.27 \pm 4.55	127.80 \pm 55.12	N/A	3.48 \pm 0.39	68.44 \pm 13.03	13.11 \pm 1.16
	DTA [†]	98.8%	12.32 \pm 4.16	1.86 \pm 1.69	19.41 \pm 16.28	N/A	2.20 \pm 0.49	1.27 \pm 1.87	1.73 \pm 0.03
	SPLART	100.0%	3.70 \pm 0.38	0.40 \pm 0.07	4.85 \pm 0.33	N/A	4.08 \pm 0.32	1.06 \pm 0.78	3.59 \pm 0.21
Prismatic	PARIS	50.0%	27.97 \pm 13.09	N/A	N/A	4.28 \pm 3.02	9.21 \pm 1.94	151.78 \pm 35.00	7.99 \pm 0.49
	DTA [†]	100.0%	16.26 \pm 3.54	N/A	N/A	1.15 \pm 0.14	2.69 \pm 0.04	15.74 \pm 0.28	2.22 \pm 0.03
	SPLART	95.0%	2.36 \pm 0.44	N/A	N/A	0.33 \pm 0.04	6.02 \pm 0.20	16.77 \pm 1.14	3.69 \pm 0.14
Overall	PARIS	42.0%	103.12 \pm 57.38	5.27 \pm 4.55	127.80 \pm 55.12	4.28 \pm 3.02	4.63 \pm 0.70	85.11 \pm 17.42	12.09 \pm 1.03
	DTA [†]	99.0%	13.11 \pm 4.04	1.86 \pm 1.69	19.41 \pm 16.28	1.15 \pm 0.14	2.30 \pm 0.40	4.16 \pm 1.55	1.83 \pm 0.03
	SPLART	99.0%	3.44 \pm 0.39	0.40 \pm 0.07	4.85 \pm 0.33	0.33 \pm 0.04	4.47 \pm 0.30	4.20 \pm 0.85	3.61 \pm 0.19

Table 1. PARIS-PMS Articulation and Mesh Reconstruction Metrics. [†]DTA requires ground-truth depth.

Type	Method	PSNR \uparrow	Depth MAE \downarrow	mIoU \uparrow
Revolute	PARIS	32.21	0.093	0.955
	DTA [†]	N/A	0.031	0.941
	SPLART	43.53	0.039	0.974
Prismatic	PARIS	33.75	0.108	0.902
	DTA [†]	N/A	0.066	0.844
	SPLART	44.38	0.025	0.890
Overall	PARIS	32.52	0.096	0.942
	DTA [†]	N/A	0.038	0.922
	SPLART	43.70	0.036	0.957

Table 2. PARIS-PMS Novel View Synthesis Metrics. Results correspond to average over successful runs (see Tab. 1 for success rates). [†]DTA requires ground-truth depth.

revolute and 2 being prismatic. We refer to this dataset as PARIS-PMS. For each articulation state, PARIS provides 100 calibrated object-centric views for training and 50 for testing, sampled from the upper hemisphere. However, the released dataset lacks ground-truth depth and part segmentation maps. To address this, we follow their data generation procedure and augment PARIS-PMS with the necessary ground-truth data for depth and segmentation evaluations. Still, no test views for intermediate articulation states are provided, limiting quantitative evaluation of novel articulation synthesis.

SPLART PartNet-Mobility Subset. We curate an additional articulated object dataset from PartNet-Mobility, dubbed SPLART-PMS, to enable a more comprehensive evaluation, including the quantitative assessment of novel articulation synthesis. Our dataset comprises 22 object instances from categories not included in PARIS-PMS, with 12 revolute and 10 prismatic objects. For the test split, we also generate ground-truth depth and part segmentation maps to facilitate relevant evaluations. Crucially, while PARIS-PMS restricts articulation states to binary values, SPLART-PMS uniformly samples states from $[-0.1, 1.1]$, demanding accurate estimation of articulation, part segmentation, and object reconstruction for correct view synthesis.

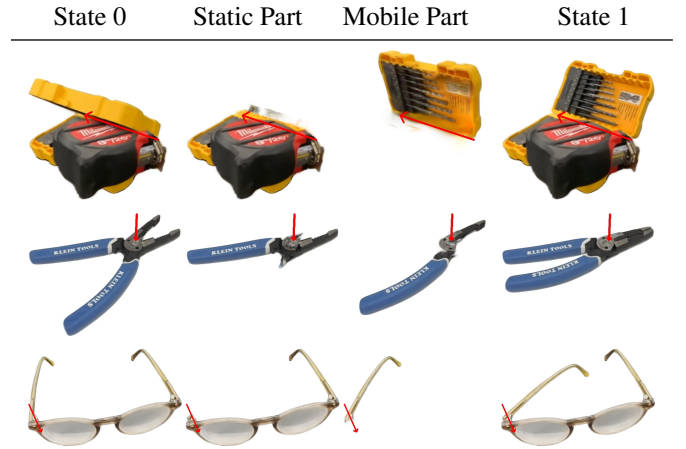


Table 3. Real-World Articulated Object Reconstructions

Our experiments reveal that SPLART-PMS poses a greater challenge than PARIS-PMS—the latter being almost perfectly solved by our method.

Real-World Dataset. To show the efficacy of SPLART on real-world usage, we collect a dataset consisting of 7 real-world objects with 9 articulations in total, including common objects like monitor, cabinet, glasses, etc. All images are captured by a hand-held phone camera. Camera parameters are calibrated following strategies detailed in Sec. 3.3. We qualitatively evaluate SPLART on this dataset.

4.2. Evaluation Metrics

Articulation Estimation. We evaluate the accuracy of articulation estimation depending on its type.

- A revolute articulation describes a rotation around some line in 3D space, parameterized by the pivot point \mathbf{p} on the line, the line’s directional axis \mathbf{a} , and the angle of rotation θ . We report the angular error err_a (in $\times 10^{-2}$ DEG) between the predicted and ground-truth axes, the geodesic distance err_r (in $\times 10^{-2}$ DEG) between the predicted and ground-truth rotations induced by the axis-angle (\mathbf{a} - θ) pair, and the pivot point error err_p . Since the pivot can

Type	Method	Success Rate \uparrow	err_a \downarrow ($\times 10^{-2}$ DEG)	err_p \downarrow ($\times 10^{-3}$)	err_r \downarrow ($\times 10^{-2}$ DEG)	err_t \downarrow ($\times 10^{-3}$)
Revolute	PARIS	18.3%	167.34 \pm 42.47	9.36 \pm 3.73	270.37 \pm 91.19	N/A
	DTA [†]	33.3%	32.76 \pm 2.38	1.28 \pm 0.39	40.80 \pm 4.66	N/A
	SPLART-2a	78.5%	2.91 \pm 0.55	2.53 \pm 2.18	52.06 \pm 32.00	N/A
	SPLART-2b	67.0%	4.62 \pm 1.69	3.92 \pm 3.07	131.99 \pm 64.45	N/A
	SPLART-3a	0.8%	296.61 \pm 0.00	2.83 \pm 0.00	427.10 \pm 0.00	N/A
	SPLART-3b	76.7%	84.13 \pm 19.13	5.11 \pm 3.03	206.98 \pm 56.30	N/A
	SPLART	77.5%	20.76 \pm 0.81	3.93 \pm 2.41	78.31 \pm 31.06	N/A
Prismatic	PARIS	39.0%	49.49 \pm 18.92	N/A	N/A	14.96 \pm 5.51
	DTA [†]	90.0%	102.77 \pm 12.48	N/A	N/A	9.25 \pm 0.73
	SPLART-2a	90.0%	25.88 \pm 23.25	N/A	N/A	1.50 \pm 1.63
	SPLART-2b	85.7%	15.63 \pm 20.40	N/A	N/A	0.73 \pm 0.58
	SPLART-3a	13.0%	15.17 \pm 0.59	N/A	N/A	2.26 \pm 0.04
	SPLART-3b	86.0%	61.00 \pm 19.76	N/A	N/A	9.43 \pm 2.37
	SPLART	95.0%	25.72 \pm 17.63	N/A	N/A	1.04 \pm 0.32
Overall	PARIS	27.7%	103.88 \pm 29.79	9.36 \pm 3.73	270.37 \pm 91.19	14.96 \pm 5.51
	DTA [†]	59.1%	77.76 \pm 8.87	1.28 \pm 0.39	40.80 \pm 4.66	9.25 \pm 0.73
	SPLART-2a	83.7%	13.85 \pm 11.36	2.53 \pm 2.18	52.06 \pm 32.00	1.50 \pm 1.63
	SPLART-2b	74.7%	9.34 \pm 9.71	3.92 \pm 3.07	131.99 \pm 64.45	0.73 \pm 0.58
	SPLART-3a	6.4%	85.53 \pm 0.44	2.83 \pm 0.00	427.10 \pm 0.00	2.26 \pm 0.04
	SPLART-3b	80.9%	73.62 \pm 19.42	5.11 \pm 3.03	206.98 \pm 56.30	9.43 \pm 2.37
	SPLART	85.5%	23.02 \pm 8.46	3.93 \pm 2.41	78.31 \pm 31.06	1.04 \pm 0.32

Table 4. SPLART-PMS Articulation Metrics. [†]DTA requires ground-truth depth.

Type	Method	PSNR \uparrow	Depth MAE \downarrow	mIoU \uparrow
Revolute	PARIS	32.75	0.125	0.897
	DTA [†]	N/A	0.053	0.886
	SplArt	36.38	0.032	0.897
Prismatic	PARIS	32.30	0.124	0.902
	DTA [†]	N/A	0.060	0.857
	SplArt	37.77	0.033	0.938
Overall	PARIS	32.51	0.125	0.900
	DTA [†]	N/A	0.058	0.867
	SplArt	37.01	0.032	0.916

Table 5. SPLART-PMS Novel View and Articulation Synthesis Metrics. Results correspond to average over successful runs (see Tab. 4 for success rates). [†]DTA requires ground-truth depth.

move arbitrarily along the axis, err_p is computed as the closest distance between the predicted and ground-truth lines induced by the axis-pivot (a-p) pair.

- A prismatic articulation describes a translation along a specific direction, parameterized by the axis \mathbf{a} of the translation direction and the translation distance d . We report the axis error err_a , as in the revolute case, and the translation error err_d , which is the distance between the predicted and ground-truth translations induced by the axis-distance pair.

Part-Level Reconstruction. The evaluation of part-level reconstruction accuracy is threefold: photometric rendering quality, geometric accuracy, and part segmentation accuracy. For all these aspects, we use novel view synthesis as the surrogate task. We perform volume rendering for each view in the test split, generating outputs that include the RGB image, depth map, and part segmentation map. Photometric rendering quality is evaluated by report-

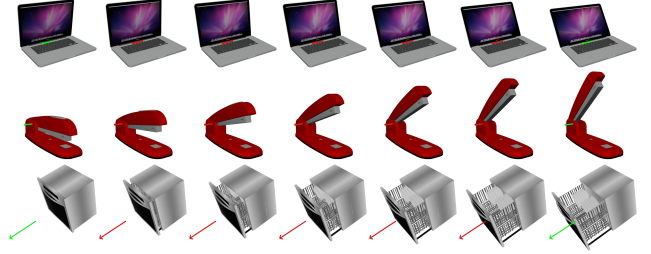


Figure 5. State interpolation from the estimated articulation.

ing image metrics PSNR. Geometric accuracy is assessed by reporting the mean absolute error of the depth map (i.e., Depth MAE). Part segmentation accuracy is evaluated using the intersection-over-union (IoU) ratio for three categories: static part (IoU_s), mobile part (IoU_m), and background (IoU_{bg}). The mean IoU (mIoU) is then reported as the average IoU across these categories. In addition, we also evaluate geometric accuracy through mesh reconstruction. To extract a mesh from 3DGS, we: (1) render depth images from uniformly sampled spherical viewpoints, (2) fuse rendered depths into a TSDF representation [4, 42], and (3) extract the mesh using the Marching Cubes algorithm [35]. For each reconstruction, we separately extract the meshes corresponding to the static, mobile, and whole parts. For evaluation, we follow the procedure used in previous works [30, 61]: 10 000 points are uniformly sampled from both the reconstructed and the ground-truth meshes, and the Chamfer distance is computed for each category—static (CD_s), mobile (CD_m), and whole (CD_w).

Impact of Randomness. Randomness can play a significant role in the joint optimization of articulation and part reconstruction [6, 30]. In order to effectively demonstrate the robustness of a method, we conduct ten trials for each scene, each with a different fixed random seed. We report results averaged over the ten runs. Additionally, we notice that for a failed run converging to a local minimum, the evaluation metrics, such as err_r , are typically orders of magnitude worse than those of a successful run. To better reflect a method’s performance, we report the number of successful runs out of the total number (i.e., ten), where a run is deemed successful if and only if each of the following criteria are met: (1) $\text{err}_a < 5$ DEG; (2) $\text{err}_p < 0.05$ for revolute articulation; (3) $\text{err}_r < 10$ DEG for revolute articulation; and (4) $\text{err}_t < 0.05$ for prismatic articulation. We then proceed to report the evaluation metrics detailed previously, *averaging only over the successful runs*.

4.3. Baselines

PARIS [30] addresses the same challenge we tackle: reconstructing the part-level shape, appearance, and motion of an articulated object from multi-view RGB images captured at two articulation states. It employs a NeRF-based

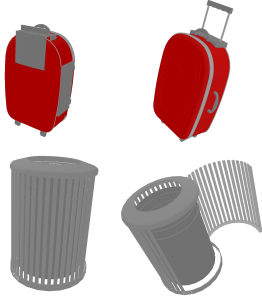
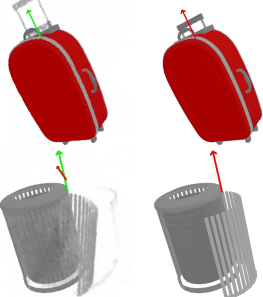
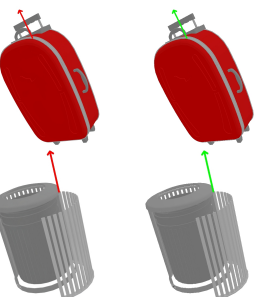
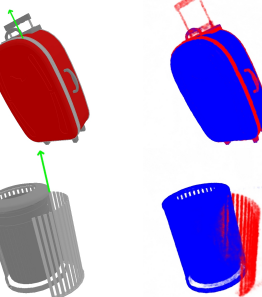
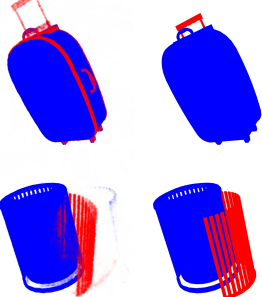
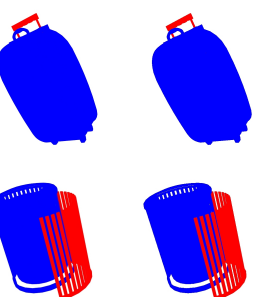
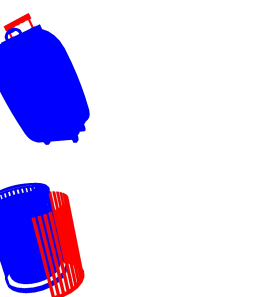
Input States	Color			Part Segmentation		
	PARIS	SPLART	Ground-truth	PARIS	SPLART	Ground-truth
						

Table 6. Qualitative comparison on part-level reconstruction and articulation estimation.

representation, modeling the object with static and mobile fields for each part, alongside a transformation to capture state transitions. These components are jointly optimized end-to-end, primarily through an image-rendering loss.

DTA [61] adopts a similar setup: reconstructing digital twins of articulated objects from multi-view RGB-D observations at two articulation states. It operates in two stages: Stage 1 optimizes neural object fields for each state using RGB-D images and extracts meshes; Stage 2 optimizes a part segmentation field and per-part motions using consistency, matching, and collision losses to determine point correspondences between states. *Note: Unlike SPLART, DTA requires ground-truth depth as input and produces plain object meshes unsuitable for photorealistic rendering.*

4.4. Experiment Results

Articulation Estimation. We quantitatively evaluate SPLART against baselines for articulation estimation accuracy, reporting scene-averaged results per articulation type on the PARIS-PMS dataset (Tab. 1) and the SPLART-PMS dataset (Tab. 4). SPLART consistently outperforms the baselines across both datasets. Additionally, we qualitatively compare SPLART with PARIS on selected scenes from these datasets in Tab. 6. In the “color” column, each image overlays the ground-truth articulation (green arrow) and the estimated articulation (red arrow), except in the ground-truth column. When the estimate aligns perfectly with the ground truth, only the red arrow appears; if significantly misaligned, only the green arrow is visible.

Part-Level Reconstruction. We assess SPLART’s part-level reconstruction accuracy against baselines, using novel view synthesis as a surrogate task. Scene-averaged results per articulation type are reported on the PARIS-PMS dataset (Tab. 2) and the SPLART-PMS dataset (Tab. 5). We also evaluate geometric accuracy through mesh reconstruction, presenting scene-averaged results per articulation type on

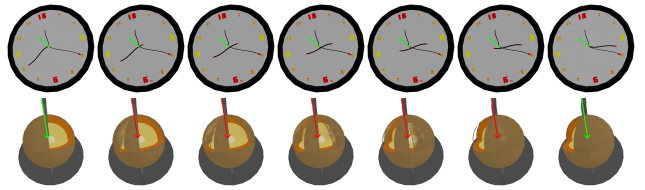


Figure 6. Failure cases due to inherent ambiguities in articulation.

the PARIS-PMS dataset (Tab. 1). Without depth supervision, which is required for the DTA baseline, SPLART matches DTA’s performance, both substantially exceeding PARIS. Qualitative comparisons with PARIS, including RGB renderings and part segmentation maps, are provided for selected scenes from both datasets in Tab. 6. Further qualitative results on articulation synthesis appear in Fig. 5.

Real-World Reconstructions. We qualitatively evaluate SPLART on a collected real-world dataset, with results shown in Tab. 3. Each image overlays the estimated articulation, visualized with a red arrow.

Ablation Studies. We perform extensive ablation studies to demonstrate the contribution of each stage. Specifically, we perform the following four ablations: (1) No geometric supervision for mobility estimation, i.e. skipping Stage 2(a), tagged SPLART-2a; (2) No photometric supervision for mobility estimation, i.e., skipping Stage 2(b), tagged SPLART-2b; (3) No geometric supervision for articulation estimation, i.e. skipping Stage 3(a), tagged SPLART-3a; (4) No photometric supervision for articulation estimation, i.e., skipping Stage 3(b), tagged SPLART-3b. Like the main experiments, we conduct 10 runs for each scene, but only on the more challenging dataset of SPLART-PMS. We report the articulation estimation results in Tab. 4.

5. Conclusion

We present SPLART, the first self-supervised method that utilizes 3D Gaussian Splatting for articulated object reconstruction from two-state RGB observations, without 3D supervision or category-specific priors. SPLART delivers robust optimization and effectively handles challenging scenarios—outperforming prior approaches while eliminating the need for 3D supervision, articulation annotations, or semantic labels. Extensive evaluations on synthetic and real-world datasets show SPLART’s articulation estimation accuracy and view synthesis quality that surpass existing methods. Real-world experiments further validate its practicality, showcasing successful reconstructions of diverse articulated objects using only a handheld RGB camera. Our framework empowers non-expert users to effortlessly create high-fidelity, articulated digital twins, supporting applications in augmented/virtual reality and robotics. However, while SPLART effectively addresses two-part articulations, it lacks direct support for multi-part articulated objects, though iteratively applying it to incremental state pairs may help. Some existing methods [6, 34, 61] handle such objects, but require known part counts and single-level articulation structures. Future research will focus on a comprehensive adaptation to multi-level articulated objects. Additionally, inherent ambiguities may arise when multiple articulations explain state differences, as shown in Fig. 6 with examples like a clock’s hand rotating on an unexpected axis or a globe’s surface spinning independently. Overall, SPLART represents a scalable, robust, and practical solution that pushes the boundaries of articulated object reconstruction, laying a strong foundation for future work.

References

- [1] Ben Abbatematteo, Stefanie Tellex, and George Konidaris. Learning to generalize kinematic models to novel objects. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2019. 1
- [2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5
- [3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2
- [4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceeding of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 303–312, 1996. 7
- [5] Andrea F. Daniele, Thomas M. Howard, and Matthew R. Walter. A multiview approach to learning articulated motion models. In *Proceedings of the International Symposium of Robotics Research (ISRR)*, 2017. 1
- [6] Jianning Deng, Kartic Subr, and Hakan Bilen. Articulate your nerf: Unsupervised articulated object modeling via conditional view synthesis. *Advances in Neural Information Processing Systems*, 37:119717–119741, 2025. 1, 2, 3, 7, 9
- [7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 224–236, 2018. 5
- [8] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–613, 2017. 2
- [9] Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 15752–15761, 2021. 2
- [10] Haoran Geng, Ziming Li, Yiran Geng, Jiayi Chen, Hao Dong, and He Wang. PartManip: Learning cross-category generalizable part manipulation policy from point cloud observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2978–2988, 2023. 2
- [11] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. GPartNet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7081–7091, 2023. 2
- [12] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. 2
- [13] Nick Heppert, Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Rares Andrei Ambrus, Jeannette Bohg, Abhinav Valada, and Thomas Kollar. CARTO: Category and joint agnostic reconstruction of articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21201–21210, 2023. 1, 2
- [14] Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3933–3939, 2023. 2
- [15] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Transactions On Graphics (TOG)*, 36(6):1–13, 2017. 2
- [16] Xiaoxia Huang, Ian Walker, and Stan Birchfield. Occlusion-aware reconstruction and manipulation of 3d articulated objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1365–1371, 2012. 1

- [17] Muhammad Zubair Irshad, Thomas Kollar, Michael Laskey, Kevin Stone, and Zsolt Kira. CenterSnap: Single-shot multi-object 3D shape reconstruction and categorical 6D pose and size estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 2
- [18] Muhammad Zubair Irshad, Sergey Zakharov, Rares Ambrus, Thomas Kollar, Zsolt Kira, and Adrien Gaidon. SHAPO: Implicit representations for multi-object shape, appearance, and pose optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 275–292, 2022. 2
- [19] Muhammad Zubair Irshad, Mauro Comi, Yen-Chen Lin, Nick Heppert, Abhinav Valada, Rares Ambrus, Zsolt Kira, and Jonathan Tremblay. Neural fields in robotics: A survey. *arXiv preprint arXiv:2410.20220*, 2024. 2
- [20] Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. ScrewNet: Category-independent articulation model estimation from depth images using screw theory. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 13670–13677, 2021. 1
- [21] Anil K Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989. 2
- [22] Hanxiao Jiang, Yongsan Mao, Manolis Savva, and Angel X Chang. OPD: Single-view 3D openable part detection. In *European Conference on Computer Vision*, pages 410–426. Springer, 2022. 2
- [23] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5616–5626, 2022. 1, 2
- [24] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3D articulated objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5003–5010, 2013. 1
- [25] Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Unsupervised pose-aware part decomposition for man-made articulated objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 558–575, 2022. 2
- [26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139–1, 2023. 1, 2
- [27] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Angjoo Kanazawa, and Ken Goldberg. Robot see robot do: Part-centric feature fields for visual imitation of articulated objects. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2024. 2
- [28] Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-Anything: Automatic modeling of articulated objects via a vision-language foundation model. *arXiv preprint arXiv:2410.13882*, 2024. 2
- [29] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3706–3715, 2020. 1, 2
- [30] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. PARIS: Part-level reconstruction and motion analysis for articulated objects. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 352–363, 2023. 1, 2, 3, 5, 7
- [31] Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Qiaojun Yu, Yang Han, and Cewu Lu. AKB-48: A real-world articulated object knowledge base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14809–14818, 2022. 2
- [32] Shaowei Liu, Saurabh Gupta, and Shenlong Wang. Building rearticulable models for arbitrary 3D objects from 4D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21138–21147, 2023. 2
- [33] Shaowei Liu, Saurabh Gupta, and Shenlong Wang. Building rearticulable models for arbitrary 3D objects from 4D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21138–21147, 2023. 2
- [34] Yu Liu, Baoxiong Jia, Ruijie Lu, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Building interactable replicas of complex articulated objects via Gaussian splatting. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 1, 2, 9
- [35] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 7
- [36] Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2Code: Reconstruct articulated objects via code generation. *arXiv preprint arXiv:2406.08474*, 2024. 2
- [37] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2633–2651, 2015. 1
- [38] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019. 2
- [39] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 1
- [40] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5
- [41] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-SDF: Learning disentangled signed distance functions for articulated shape

- representation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 13001–13011, 2021. 1, 2
- [42] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011. 7
- [43] Neil Nie, Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Structure from action: Learning interactions for articulated object 3d structure discovery. *arXiv preprint arXiv:2207.08997*, 2022. 2
- [44] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2
- [45] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 2
- [46] Sudeep Pillai, Matthew R. Walter, and Seth Teller. Learning articulated motions from visual demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2014. 1
- [47] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5
- [48] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4938–4947, 2020. 5
- [49] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 5
- [50] Chaoyue Song, Jiacheng Wei, Chuan Sheng Foo, Guosheng Lin, and Fayao Liu. REACTO: Reconstructing articulated objects from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5384–5395, 2024. 2
- [51] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011. 1
- [52] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2022. 1
- [53] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 2
- [54] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2023. 1
- [55] Wei-Cheng Tseng, Hung-Ju Liao, Lin Yen-Chen, and Min Sun. CLA-NeRF: Category-level articulated neural radiance field. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 8454–8460, 2022. 1
- [56] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2
- [57] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qingping Zhao, and Kai Xu. Shape2Motion: Joint analysis of motion parts and attributes from 3D shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8876–8884, 2019. 1, 2
- [58] Xian Wang, Luyao Shen, and Lik-Hang Lee. A systematic review of XR-based remote human-robot interaction systems. *arXiv preprint arXiv:2403.11384*, 2024. 1
- [59] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 2
- [60] Fangyin Wei, Rohan Chhabra, Lingni Ma, Christoph Lassner, Michael Zollhöfer, Szymon Rusinkiewicz, Chris Sweeney, Richard Newcombe, and Mira Slavcheva. Self-supervised neural articulated shape and appearance models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15816–15826, 2022. 1, 2
- [61] 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 *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 7, 8, 9
- [62] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5
- [63] Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen, Oliver Van Kaick, Hao Zhang, and Hui Huang. RPM-Net: Recurrent prediction of motion and parts from point cloud. *arXiv preprint arXiv:2006.14865*, 2020. 2
- [64] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting, 2024. 1

- [65] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *arXiv preprint arXiv:1809.07417*, 2018. [2](#)
- [66] Sergey Zakharov, Katherine Liu, Adrien Gaidon, and Rares Ambrus. ReFiNe: Recursive field networks for cross-modal multi-scene representation. In *Proceeding of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2024. [2](#)
- [67] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. [2](#)

SPLART: Articulation Estimation and Part-Level Reconstruction with 3D Gaussian Splatting

Supplementary Material

6. 3D Gaussian Splatting

3D Gaussian Splatting [26] (3DGS) is a method for reconstructing 3D scenes from posed images by representing the scene using Gaussian distributions in a continuous 3D space. Given a Gaussian blob parameterized by (μ, R, S, σ) , where $\mu \in \mathbb{R}^3$ denotes the position of the center (mean) of the Gaussian, $R \in \mathbb{R}^{3 \times 3}$ is a rotation matrix that denotes its orientation, $S \in \mathbb{R}^{3 \times 3}$ is the scale matrix (the scale matrix and rotation matrix determine the covariance matrix of the Gaussian), and $\sigma \in \mathbb{R}^+$ denotes its opacity. By design, the influence of a Gaussian on a point \mathbf{x} is given by

$$g(\mathbf{x}|\mu, R, S, \sigma) = \sigma \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T (RSS^T R^T)^{-1} (\mathbf{x} - \mu)\right). \quad (16)$$

In practice, 3D Gaussians are first projected to 2D given the camera view, while all Gaussians intersecting with a pixel's ray are sorted by depth for alpha-compositing. Please refer to Kerbl et al. [26] and Ye et al. [64] for more technical details. Compared to Neural Radiance Fields [39] (NeRFs), which represent a scene with radiance and density fields in the form of neural networks, 3DGS offers a much faster rendering speed (more than $100\times$ speed-up) thanks to the efficient rasterization of Gaussians.

7. Articulating the Gaussians

In SPLART, we articulate a Gaussian blob by (1) rotating by an angle θ around a line specified by (\mathbf{p}, \mathbf{a}) , where \mathbf{p} denotes the pivot point and \mathbf{a} denotes the axis direction; and (2) translating along the same line by distance d . We then need to update the parameters of the Gaussian (μ', R', S', σ') to reflect this articulation. Further, letting \mathbf{d} be the camera-to-Gaussian direction computed after the articulated motion, we want to find out the actual direction \mathbf{d}' with which to query the radiance field (i.e., a spherical function represented by Spherical Harmonics).

Let \mathbf{x}' be the new point that results from applying the articulated motion to \mathbf{x} , which follows as

$$\mathbf{x}' = R_{\mathbf{a},\theta}(\mathbf{x} - \mathbf{p}) + \mathbf{p} + d\mathbf{a}, \quad (17)$$

where $R_{\mathbf{a},\theta}$ is the matrix form of the axis-angle rotation (\mathbf{a}, θ) . By definition,

$$g(\mathbf{x}' | \mu', R', S', \sigma') = g(\mathbf{x} | \mu, R, S, \sigma) \forall \mathbf{x}. \quad (18)$$

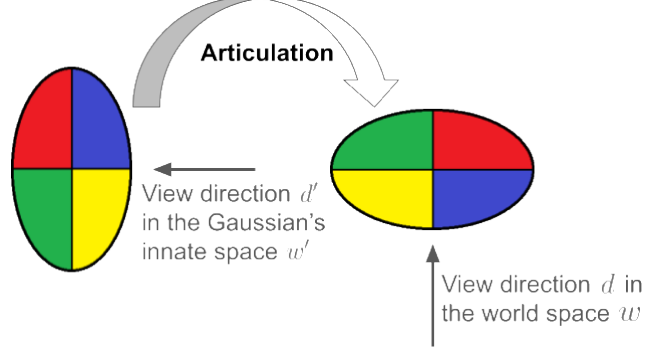


Figure 7. Since the radiance field is not modified by the articulation, we update the view direction when querying the Gaussian's radiance field in order to compensate for the articulation.

By plugging Equations 16 and 17 into Equation 18, the updated parameters follow as

$$\mu' = R_{\mathbf{a},\theta}(\mu - \mathbf{p}) + \mathbf{p} + d\mathbf{a}, \quad (19a)$$

$$R' = R_{\mathbf{a},\theta}R, \quad (19b)$$

$$S' = S, \quad (19c)$$

$$\sigma' = \sigma. \quad (19d)$$

Note that only the Gaussians are articulated, but not the radiance fields (since the coefficients of Spherical Harmonics are *not* changed). To determine the actual direction \mathbf{d}' with which to query the radiance field, imagine there is a world space w' that follows the same articulated motion, meaning that w' is stationary relative to the Gaussian blob. By definition, an articulated point \mathbf{x}' given by Equation 17 in the original world space w is still denoted as point \mathbf{x} in w' . So

$${}^wT_{w'}(\mathbf{x}) = R_{\mathbf{a},\theta}(\mathbf{x} - \mathbf{p}) + \mathbf{p} + d\mathbf{a}, \quad (20)$$

where ${}^wT_{w'}(\cdot)$ denotes the transformation from w' to w . Now, given view direction \mathbf{d} in w , the actual query direction is represented via the Gaussian's innate space, which can be computed as

$$\mathbf{d}' = {}^{w'}T_w(\mathbf{d}) \quad (21a)$$

$$= R_{\mathbf{a},\theta}^{-1}\mathbf{d}. \quad (21b)$$

See Figure 7 for an illustration.

8. Implementation

SPLART leverages *nerfstudio* [54] and *gsplat* [64], widely used open-source libraries for neural rendering and Gaus-

sian splatting, respectively. We apply consistent hyper-parameters across all experiments, spanning synthetic and real-world datasets. Optimization stages vary in termination criteria: those driven by geometric consistency—Stages 2(a), 3(a), and 3(c)—halt upon convergence, assessed by the loss function’s rate of change, whereas those guided by photometric loss—Stages 1, 2(b), and 3(b)—stop after fixed iterations of 10 000, 5000, and 10 000, respectively. By contrast, *splatfacto* (*nerfstudio*’s default 3DGS implementation) uses 30 000 iterations. In Stage 3(a), we set $K^m = K^{cm} = 3$ by default, with each optimization trial lasting 10 seconds to 1 minute on an RTX 2080 Ti. Consequently, SPLART’s total training time, approximately 20–30 minutes, is roughly twice that of *splatfacto*. For comparison, PARIS [30] requires 15–20 minutes, while DTA [61], including its LoFTR pixel-matching step, takes 35–45 minutes. At inference, rendering for novel view and articulation synthesis achieves 60–100 frames per second, varying with scene complexity due to the increased Gaussian count for intricate objects.

9. Robustness of Geometric Consistency

In Stage 3(a), we propose a practical strategy that involves multiple attempts using both mobile-only and cross-mobile geometric consistency for robust articulation estimation. For this approach to be effective, two prerequisites must be met: (1) a correct articulation should induce a lower loss than most, if not all, incorrect articulation estimates; and (2) the optimization should have a high likelihood of converging to the global optimum (indicated by the correctness of the articulation) for each randomized attempt, ensuring that a small number of trials is sufficient and robust. To verify that both prerequisites are satisfied, we design a dedicated experiment that involves 200 independent

optimization trials: 100 using mobile-only geometric consistency and 100 using cross-mobile geometric consistency, both performed based on the model checkpoint saved at the end of Stage 2(b). A trial is deemed successful if it meets a similar but slightly relaxed criterion (since the estimated articulation is still coarse at this stage). Additionally, we rank the loss of each trial in comparison to the other trials performed under the same formulation. In light of the prerequisites above, we are interested in seeing whether successful trials tend to exhibit lower loss compared to those that are unsuccessful. Figure 8 visualizes these results as a scatter plot for a set of objects from the SPLART-PMS dataset.

10. Additional Qualitative Results

We present more qualitative results, including comparison with PARIS [30] on part-level reconstruction and articulation estimation in Figure 10, and view synthesis for interpolated articulation states in Figure 9.

11. Per-Scene Quantitative Results

We report quantitative comparison of SPLART and PARIS for each scene in both PARIS-PMS and SPLART-PMS. Specifically, see Tab. 7, Tab. 9, and Tab. 10 for articulation estimation results on PARIS-PMS, revolute cases of SPLART-PMS, and prismatic cases of SPLART-PMS, respectively. For novel view and articulation synthesis, we further include SSIM [59] and LPIPS [67] metrics (in addition to PSNR [21]) for the evaluation of rendering quality, and per-category intersection-over-union ratios (in addition to mIoU): IoU_s for the static part, IoU_m for the mobile part, and IoU_{bg} for the background. See Tab. 8, Tab. 11, and Tab. 12 for results on PARIS-PMS, revolute cases of SPLART-PMS, and prismatic cases of SPLART-PMS, respectively.

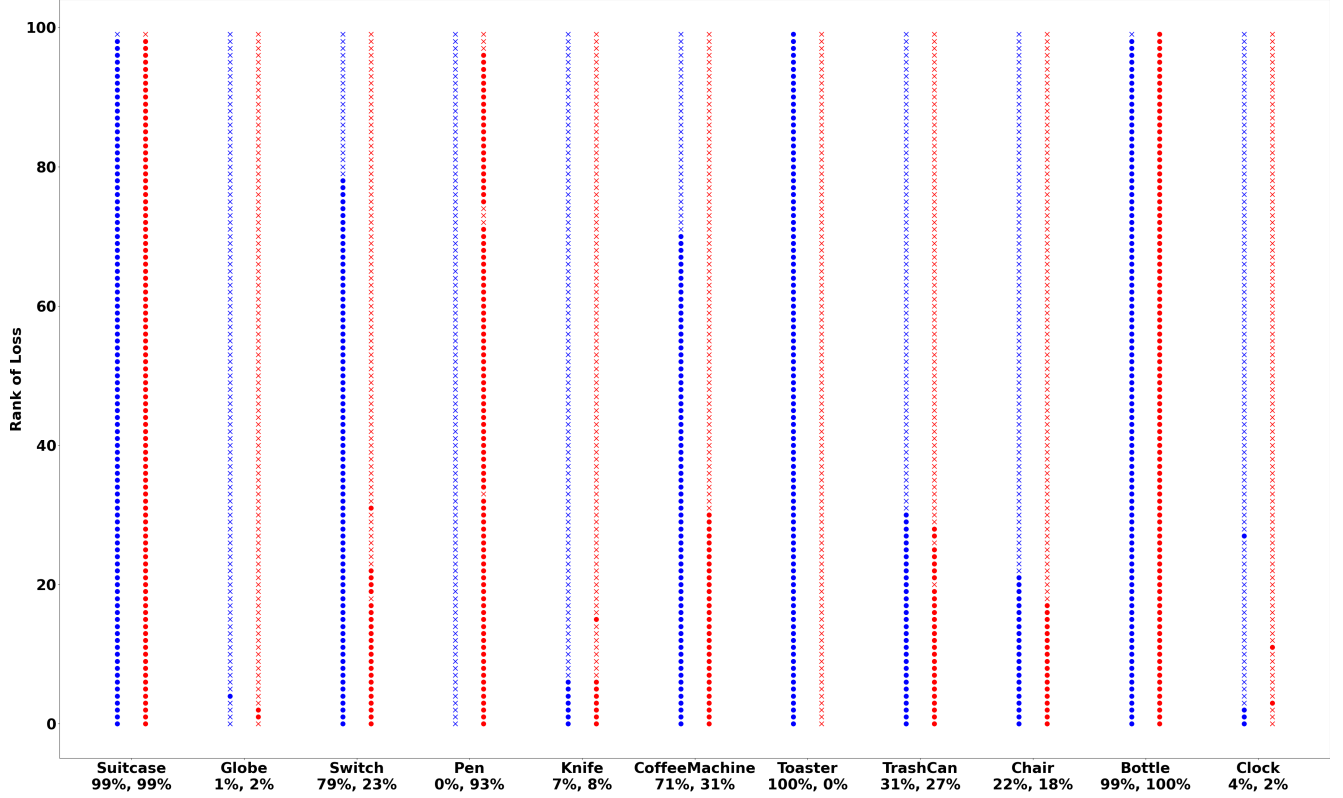


Figure 8. An analysis of the robustness of geometric consistency for articulation estimation. The horizontal axis represents different scenes, each with two columns for the two optimization formulations, where **blue** is used for optimization trials with mobile-only geometric consistency, and **red** for those with cross-mobile geometric consistency. The vertical axis represents the loss rank (lower is better). For each optimization trial, **●** denotes success, and **×** denotes failure. For each scene, the average success rates for both optimization formulations are shown under the scene name. Note that these success rates are the averages over the independent optimization trials and should not be confused with that of Stage 3(a) as a whole. We observe that successful trials **●** are often associated with lower losses than are failed trials **×**, and that both optimization formulations are necessary to ensure overall robustness (e.g., *pen* requires **cross-mobile geometric consistency**, while *toaster* requires **mobile-only geometric consistency**).

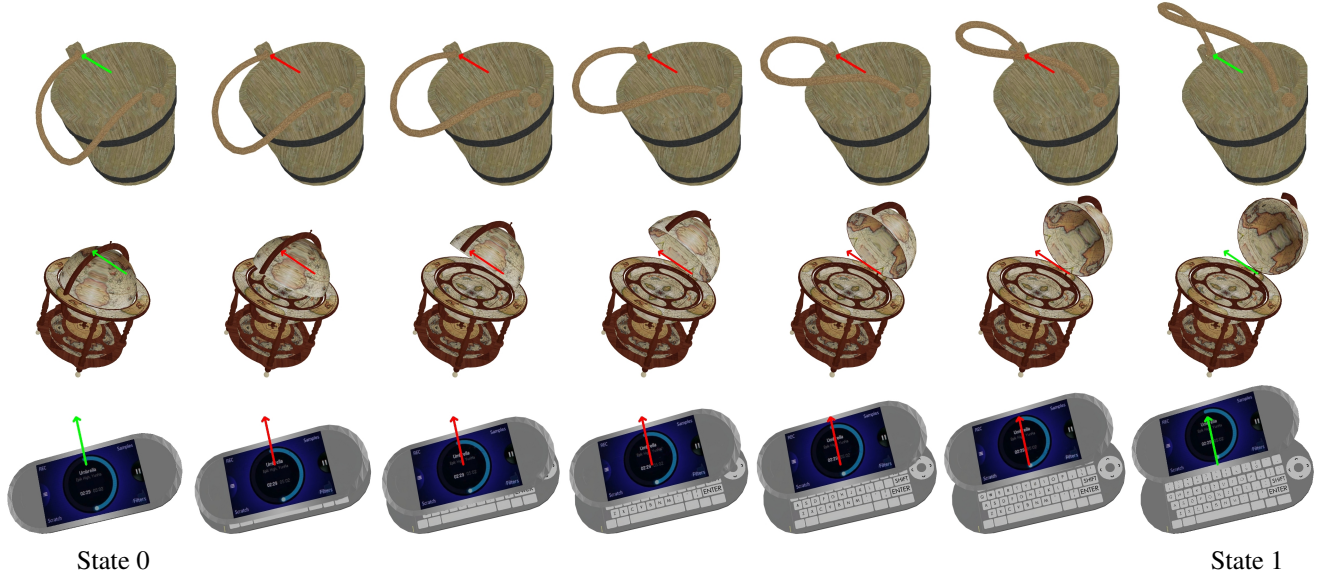


Figure 9. Additional visualizations of state interpolation for three objects along with their estimated articulation models. The left- and right-most images correspond to the ground-truth of the two end articulation states, annotated with the ground-truth articulation using **green arrows**. The intermediate images are rendered results obtained by interpolating the articulation state, annotated with both the ground-truth articulation (**green arrows**) and the estimated articulation (**red arrows**). Due to the high accuracy of the articulation estimation, the ground-truth **green arrows** are hidden by the estimated **red arrows** in the interpolated images.

Input States	Color			Part Segmentation		
	PARIS	SPLART	Ground-truth	PARIS	SPLART	Ground-truth

Figure 10. Additional qualitative comparisons of part-level reconstruction and articulation estimation.

Type	Scene	Method	Success Rate \uparrow	err_a \downarrow ($\times 10^{-2}$ DEG)	err_p \downarrow ($\times 10^{-3}$)	err_r \downarrow ($\times 10^{-2}$ DEG)	err_t \downarrow ($\times 10^{-3}$)	CD_s \downarrow ($\times 10^{-3}$)	CD_m \downarrow ($\times 10^{-3}$)	CD_w \downarrow ($\times 10^{-3}$)
Revolute	USB	PARIS	0/10	F	F	F	N/A	F	F	F
		DTA †	10/10	9.30 \pm 1.29	0.39 \pm 0.31	14.81 \pm 3.49	N/A	2.62 \pm 0.06	1.48 \pm 0.03	1.36 \pm 0.02
		SplArt	10/10	4.74 \pm 0.32	0.26 \pm 0.05	3.99 \pm 0.26	N/A	0.89 \pm 0.08	0.83 \pm 0.11	0.91 \pm 0.03
	foldchair	PARIS	3/10	103.68 \pm 15.95	4.69 \pm 4.77	195.70 \pm 67.55	N/A	0.45 \pm 0.09	45.34 \pm 4.62	14.85 \pm 1.00
		DTA †	10/10	3.34 \pm 1.43	0.50 \pm 0.31	9.72 \pm 3.61	N/A	0.18 \pm 0.00	0.14 \pm 0.00	0.26 \pm 0.00
		SplArt	10/10	5.47 \pm 0.22	0.12 \pm 0.03	8.93 \pm 0.34	N/A	1.53 \pm 0.68	0.33 \pm 0.01	0.31 \pm 0.00
	fridge	PARIS	9/10	105.51 \pm 50.87	6.04 \pm 4.17	121.25 \pm 45.78	N/A	2.97 \pm 0.19	39.11 \pm 2.07	11.62 \pm 0.54
		DTA †	10/10	6.67 \pm 2.79	0.51 \pm 0.32	12.07 \pm 2.08	N/A	0.63 \pm 0.01	0.29 \pm 0.01	0.70 \pm 0.00
		SplArt	10/10	4.14 \pm 0.56	0.06 \pm 0.05	5.81 \pm 0.34	N/A	2.23 \pm 0.05	1.21 \pm 0.03	1.95 \pm 0.04
	laptop	PARIS	10/10	117.26 \pm 94.05	6.89 \pm 5.45	99.11 \pm 51.26	N/A	0.61 \pm 0.33	32.43 \pm 2.19	15.08 \pm 1.41
		DTA †	10/10	6.65 \pm 1.73	1.33 \pm 0.52	11.81 \pm 3.62	N/A	0.31 \pm 0.00	0.14 \pm 0.00	0.34 \pm 0.00
		SplArt	10/10	3.20 \pm 0.55	0.59 \pm 0.05	4.89 \pm 0.28	N/A	0.24 \pm 0.00	0.34 \pm 0.01	0.35 \pm 0.01
	oven	PARIS	10/10	161.16 \pm 112.94	3.44 \pm 3.81	95.15 \pm 55.90	N/A	9.89 \pm 0.93	156.89 \pm 43.24	10.90 \pm 1.70
		DTA †	10/10	19.35 \pm 4.10	1.57 \pm 0.60	11.03 \pm 2.19	N/A	4.61 \pm 0.07	0.44 \pm 0.01	4.26 \pm 0.06
		SplArt	10/10	1.45 \pm 0.32	0.90 \pm 0.06	2.95 \pm 0.27	N/A	7.10 \pm 0.44	1.89 \pm 3.52	6.02 \pm 0.32
	scissor	PARIS	0/10	F	F	F	N/A	F	F	F
		DTA †	10/10	7.28 \pm 5.05	4.84 \pm 7.66	56.56 \pm 95.23	N/A	1.67 \pm 3.41	5.06 \pm 13.99	0.42 \pm 0.00
		SplArt	10/10	2.24 \pm 0.17	0.24 \pm 0.06	2.18 \pm 0.26	N/A	0.37 \pm 0.01	0.20 \pm 0.01	0.25 \pm 0.00
	stapler	PARIS	0/10	F	F	F	N/A	F	F	F
		DTA †	9/10	7.06 \pm 5.68	1.81 \pm 1.16	11.76 \pm 11.74	N/A	2.93 \pm 0.24	2.16 \pm 0.89	2.03 \pm 0.03
		SplArt	10/10	4.64 \pm 0.43	0.75 \pm 0.08	4.09 \pm 0.28	N/A	1.17 \pm 0.02	2.12 \pm 0.14	1.05 \pm 0.02
	washer	PARIS	0/10	F	F	F	N/A	F	F	F
		DTA †	10/10	38.95 \pm 11.24	3.91 \pm 2.63	27.51 \pm 8.30	N/A	4.68 \pm 0.11	0.40 \pm 0.01	4.46 \pm 0.12
		SplArt	10/10	3.76 \pm 0.48	0.24 \pm 0.15	5.95 \pm 0.67	N/A	19.14 \pm 1.27	1.59 \pm 2.40	17.92 \pm 1.22
	mean	PARIS	40.0%	121.90 \pm 68.45	5.27 \pm 4.55	127.80 \pm 55.12	N/A	3.48 \pm 0.39	68.44 \pm 13.03	13.11 \pm 1.16
		DTA †	98.8%	12.32 \pm 4.16	1.86 \pm 1.69	19.41 \pm 16.28	N/A	2.20 \pm 0.49	1.27 \pm 1.87	1.73 \pm 0.03
		SplArt	100.0%	3.70 \pm 0.38	0.40 \pm 0.07	4.85 \pm 0.33	N/A	4.08 \pm 0.32	1.06 \pm 0.78	3.59 \pm 0.21
Prismatic	blade	PARIS	0/10	F	N/A	N/A	F	F	F	F
		DTA †	10/10	25.72 \pm 4.58	N/A	N/A	0.94 \pm 0.14	0.49 \pm 0.01	31.11 \pm 0.55	0.37 \pm 0.01
		SplArt	9/10	2.04 \pm 0.61	N/A	N/A	0.22 \pm 0.04	0.44 \pm 0.01	26.60 \pm 0.85	0.44 \pm 0.01
	storage	PARIS	10/10	27.97 \pm 13.09	N/A	N/A	4.28 \pm 3.02	9.21 \pm 1.94	151.78 \pm 35.00	7.99 \pm 0.49
		DTA †	10/10	6.79 \pm 2.50	N/A	N/A	1.37 \pm 0.13	4.88 \pm 0.07	0.36 \pm 0.00	4.07 \pm 0.06
		SplArt	10/10	2.69 \pm 0.26	N/A	N/A	0.45 \pm 0.04	11.60 \pm 0.38	6.94 \pm 1.43	6.94 \pm 0.27
	mean	PARIS	50.0%	27.97 \pm 13.09	N/A	N/A	4.28 \pm 3.02	9.21 \pm 1.94	151.78 \pm 35.00	7.99 \pm 0.49
		DTA †	100.0%	16.26 \pm 3.54	N/A	N/A	1.15 \pm 0.14	2.69 \pm 0.04	15.74 \pm 0.28	2.22 \pm 0.03
		SplArt	95.0%	2.36 \pm 0.44	N/A	N/A	0.33 \pm 0.04	6.02 \pm 0.20	16.77 \pm 1.14	3.69 \pm 0.14
Overall	PARIS	42.0%	103.12 \pm 57.38	5.27 \pm 4.55	127.80 \pm 55.12	4.28 \pm 3.02	4.63 \pm 0.70	85.11 \pm 17.42	12.09 \pm 1.03	
	DTA †	99.0%	13.11 \pm 4.04	1.86 \pm 1.69	19.41 \pm 16.28	1.15 \pm 0.14	2.30 \pm 0.40	4.16 \pm 1.55	1.83 \pm 0.03	
	SplArt	99.0%	3.44 \pm 0.39	0.40 \pm 0.07	4.85 \pm 0.33	0.33 \pm 0.04	4.47 \pm 0.30	4.20 \pm 0.85	3.61 \pm 0.19	

Table 7. PARIS-PMS Articulation and Mesh Reconstruction Metrics. [†]DTA requires ground-truth depth.

Type	Scene	Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Depth MAE \downarrow	IoU _s \uparrow	IoU _m \uparrow	IoU _{bg} \uparrow	mIoU \uparrow
Revolute	USB	PARIS	F	F	F	F	F	F	F	F
		DTA [†]	N/A	N/A	N/A	0.022	0.864	0.892	0.996	0.917
		SplArt	45.70	0.996	0.0029	0.026	0.982	0.979	1.000	0.987
	foldchair	PARIS	33.48	0.936	0.0844	0.096	0.961	0.939	0.994	0.965
		DTA [†]	N/A	N/A	N/A	0.041	0.915	0.950	0.988	0.951
		SplArt	43.98	0.992	0.0119	0.037	0.981	0.980	0.999	0.987
	fridge	PARIS	32.22	0.967	0.0766	0.087	0.973	0.866	0.997	0.945
		DTA [†]	N/A	N/A	N/A	0.034	0.957	0.823	0.992	0.924
		SplArt	41.40	0.996	0.0101	0.045	0.986	0.917	0.999	0.967
	laptop	PARIS	31.64	0.964	0.0758	0.057	0.907	0.959	0.997	0.954
		DTA [†]	N/A	N/A	N/A	0.014	0.938	0.964	0.997	0.966
		SplArt	40.90	0.995	0.0096	0.026	0.931	0.975	0.999	0.968
	oven	PARIS	31.48	0.950	0.1048	0.132	0.977	0.891	0.996	0.955
		DTA [†]	N/A	N/A	N/A	0.038	0.981	0.946	0.993	0.973
		SplArt	41.59	0.993	0.0148	0.082	0.990	0.935	0.999	0.974
	scissor	PARIS	F	F	F	F	F	F	F	F
		DTA [†]	N/A	N/A	N/A	0.051	0.863	0.876	0.991	0.910
		SplArt	46.15	0.996	0.0030	0.044	0.951	0.954	0.999	0.968
	stapler	PARIS	F	F	F	F	F	F	F	F
		DTA [†]	N/A	N/A	N/A	0.035	0.902	0.894	0.994	0.930
		SplArt	44.81	0.994	0.0021	0.034	0.959	0.940	1.000	0.966
	washer	PARIS	F	F	F	F	F	F	F	F
		DTA [†]	N/A	N/A	N/A	0.016	0.991	0.888	0.998	0.959
		SplArt	43.75	0.996	0.0079	0.131	0.995	0.930	0.999	0.975
	mean	PARIS	32.21	0.954	0.0854	0.093	0.954	0.914	0.996	0.955
		DTA [†]	N/A	N/A	N/A	0.031	0.926	0.904	0.994	0.941
		SplArt	43.53	0.995	0.0078	0.053	0.972	0.951	0.999	0.974
Prismatic	blade	PARIS	F	F	F	F	F	F	F	F
		DTA [†]	N/A	N/A	N/A	0.116	0.728	0.416	0.998	0.714
		SplArt	46.15	0.998	0.0016	0.052	0.900	0.657	1.000	0.852
	storage	PARIS	33.75	0.944	0.1228	0.108	0.954	0.722	0.996	0.891
		DTA [†]	N/A	N/A	N/A	0.017	0.984	0.944	0.997	0.975
		SplArt	42.62	0.985	0.0481	0.037	0.951	0.850	0.999	0.933
	mean	PARIS	33.75	0.944	0.1228	0.108	0.954	0.722	0.996	0.891
		DTA [†]	N/A	N/A	N/A	0.066	0.856	0.680	0.997	0.844
		SplArt	44.38	0.991	0.0248	0.044	0.925	0.754	0.999	0.893
Overall		PARIS	32.52	0.952	0.0929	0.096	0.954	0.875	0.996	0.942
		DTA [†]	N/A	N/A	N/A	0.038	0.912	0.859	0.994	0.922
		SplArt	43.70	0.994	0.0112	0.052	0.963	0.912	0.999	0.958

Table 8. PARIS-PMS Novel View Synthesis Metrics. [†]DTA requires ground-truth depth.

Scene	Method	Success Rate \uparrow	err_a ($\times 10^{-2}$ DEG) \downarrow		err_p ($\times 10^{-3}$) \downarrow		err_r ($\times 10^{-2}$ DEG) \downarrow	
2230 Chair	PARIS	1/10	44.97 \pm	0.00	4.41 \pm	0.00	130.66 \pm	0.00
	DTA [†]	9/10	2.77 \pm	0.79	0.18 \pm	0.14	5.62 \pm	2.35
	SplArt	10/10	0.33 \pm	0.13	0.06 \pm	0.03	0.84 \pm	0.36
5477 Display	PARIS	8/10	86.76 \pm	27.41	3.42 \pm	3.21	133.12 \pm	43.26
	DTA [†]	10/10	1.38 \pm	0.39	0.27 \pm	0.06	2.29 \pm	0.55
	SplArt	10/10	1.91 \pm	0.43	0.49 \pm	0.25	4.10 \pm	0.75
7054 Clock	PARIS	0/10	F		F		F	
	DTA [†]	0/10	F		F		F	
	SplArt	1/10	211.22 \pm	0.00	1.06 \pm	0.00	390.40 \pm	0.00
11951 TrashCan	PARIS	0/10	F		F		F	
	DTA [†]	0/10	F		F		F	
	SplArt	10/10	1.65 \pm	0.71	21.55 \pm	16.33	473.55 \pm	352.80
100247 Box	PARIS	0/10	F		F		F	
	DTA [†]	0/10	F		F		F	
	SplArt	8/10	0.79 \pm	0.31	0.14 \pm	0.10	1.99 \pm	0.67
100460 Bucket	PARIS	0/10	F		F		F	
	DTA [†]	10/10	15.09 \pm	6.89	0.61 \pm	0.34	24.57 \pm	10.06
	SplArt	10/10	0.50 \pm	0.23	0.03 \pm	0.04	0.79 \pm	0.36
100756 Globe	PARIS	0/10	F		F		F	
	DTA [†]	0/10	F		F		F	
	SplArt	1/10	0.72 \pm	0.00	0.09 \pm	0.00	3.31 \pm	0.00
100794 Globe	PARIS	1/10	126.06 \pm	0.00	14.17 \pm	0.00	533.60 \pm	0.00
	DTA [†]	10/10	4.59 \pm	3.84	2.67 \pm	1.39	108.22 \pm	10.35
	SplArt	10/10	1.22 \pm	0.35	0.12 \pm	0.10	2.09 \pm	0.50
100882 Switch	PARIS	2/10	393.07 \pm	86.79	15.78 \pm	4.03	342.44 \pm	171.13
	DTA [†]	1/10	139.96 \pm	0.00	2.67 \pm	0.00	63.28 \pm	0.00
	SplArt	10/10	19.37 \pm	1.29	0.62 \pm	0.23	11.20 \pm	0.61
101542 Dispenser	PARIS	7/10	71.74 \pm	49.72	5.44 \pm	6.19	251.86 \pm	267.19
	DTA [†]	0/10	F		F		F	
	SplArt	8/10	0.75 \pm	0.43	0.19 \pm	0.06	11.48 \pm	0.70
102400 Knife	PARIS	0/10	F		F		F	
	DTA [†]	0/10	F		F		F	
	SplArt	5/10	9.01 \pm	5.46	22.59 \pm	11.58	37.95 \pm	15.46
103031 CoffeeMachine	PARIS	3/10	281.43 \pm	90.89	12.95 \pm	8.94	230.56 \pm	65.56
	DTA [†]	0/10	F		F		F	
	SplArt	10/10	1.69 \pm	0.40	0.22 \pm	0.14	1.98 \pm	0.58
mean	PARIS	18.3%	167.34 \pm	42.47	9.36 \pm	3.73	270.37 \pm	91.19
	DTA [†]	33.3%	32.76 \pm	2.38	1.28 \pm	0.39	40.80 \pm	4.66
	SplArt	77.5%	20.76 \pm	0.81	3.93 \pm	2.41	78.31 \pm	31.06

Table 9. SPLART-PMS Articulation Metrics on Revolute Scenes. [†]DTA requires ground-truth depth.

Scene	Method	Success Rate \uparrow	err_a ($\times 10^{-2}$ DEG) \downarrow	err_t ($\times 10^{-3}$) \downarrow
3558 Bottle	PARIS	5/10	45.83 \pm 22.38	23.32 \pm 12.29
	DTA [†]	10/10	200.52 \pm 10.32	4.60 \pm 0.27
	SplArt	10/10	3.82 \pm 0.52	0.10 \pm 0.02
12085 Dishwasher	PARIS	6/10	15.60 \pm 7.00	4.67 \pm 1.10
	DTA [†]	10/10	3.78 \pm 1.96	2.46 \pm 0.18
	SplArt	10/10	1.00 \pm 0.14	0.24 \pm 0.02
27189 Table	PARIS	3/10	23.69 \pm 10.43	21.18 \pm 3.03
	DTA [†]	10/10	15.90 \pm 2.43	2.69 \pm 0.45
	SplArt	10/10	0.14 \pm 0.05	0.14 \pm 0.01
100248 Suitcase	PARIS	0/10	F	F
	DTA [†]	10/10	192.55 \pm 30.46	10.57 \pm 1.60
	SplArt	10/10	1.22 \pm 0.58	0.10 \pm 0.04
101713 Pen	PARIS	0/10	F	F
	DTA [†]	10/10	179.16 \pm 51.33	3.76 \pm 1.04
	SplArt	10/10	244.39 \pm 172.86	7.96 \pm 2.71
102016 USB	PARIS	2/10	17.52 \pm 10.01	9.53 \pm 1.87
	DTA [†]	10/10	44.11 \pm 5.59	7.34 \pm 0.69
	SplArt	10/10	0.71 \pm 0.33	0.49 \pm 0.09
102812 Switch	PARIS	8/10	34.54 \pm 18.03	9.04 \pm 6.90
	DTA [†]	10/10	190.87 \pm 8.20	38.68 \pm 1.85
	SplArt	10/10	1.55 \pm 0.80	0.11 \pm 0.05
103042 Window	PARIS	5/10	189.87 \pm 54.52	30.48 \pm 9.55
	DTA [†]	10/10	92.93 \pm 1.64	11.92 \pm 0.23
	SplArt	10/10	1.55 \pm 0.24	0.76 \pm 0.18
103549 Toaster	PARIS	0/10	F	F
	DTA [†]	0/10	F	F
	SplArt	5/10	2.63 \pm 0.74	0.30 \pm 0.07
103941 Phone	PARIS	10/10	19.39 \pm 10.07	6.51 \pm 3.81
	DTA [†]	10/10	5.06 \pm 0.42	1.28 \pm 0.28
	SplArt	10/10	0.18 \pm 0.09	0.20 \pm 0.03
mean	PARIS	39.0%	49.49 \pm 18.92	14.96 \pm 5.51
	DTA [†]	90.0%	102.77 \pm 12.48	9.25 \pm 0.73
	SplArt	95.0%	25.72 \pm 17.63	1.04 \pm 0.32

Table 10. SPLART-PMS Articulation Metrics on Prismatic Scenes. [†]DTA requires ground-truth depth.

Scene	Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Depth MAE \downarrow	IoU _s \uparrow	IoU _m \uparrow	IoU _{bg} \uparrow	mIoU \uparrow
2230 Chair	PARIS	28.89	0.924	0.1004	0.231	0.510	0.737	0.982	0.743
	DTA [†]	N/A	N/A	N/A	0.098	0.764	0.857	0.980	0.867
	SplArt	32.08	0.972	0.0392	0.027	0.498	0.791	0.993	0.761
5477 Display	PARIS	34.34	0.943	0.0979	0.100	0.933	0.973	0.997	0.968
	DTA [†]	N/A	N/A	N/A	0.020	0.783	0.968	0.996	0.916
	SplArt	38.33	0.970	0.0379	0.020	0.804	0.971	0.999	0.924
7054 Clock	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	F	F	F	F	F	F	F	F
	SplArt	35.54	0.985	0.0187	0.027	0.994	0.811	0.999	0.935
11951 TrashCan	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	F	F	F	F	F	F	F	F
	SplArt	30.01	0.952	0.0560	0.083	0.920	0.735	0.980	0.878
100247 Box	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	F	F	F	F	F	F	F	F
	SplArt	36.96	0.968	0.0409	0.030	0.989	0.987	0.999	0.992
100460 Bucket	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	N/A	N/A	N/A	0.081	0.948	0.455	0.984	0.795
	SplArt	37.83	0.970	0.0397	0.016	0.989	0.832	0.999	0.940
100756 Globe	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	F	F	F	F	F	F	F	F
	SplArt	36.30	0.990	0.0346	0.035	0.577	0.346	0.997	0.640
100794 Globe	PARIS	29.45	0.915	0.0965	0.115	0.974	0.937	0.996	0.969
	DTA [†]	N/A	N/A	N/A	0.046	0.961	0.943	0.994	0.966
	SplArt	35.47	0.980	0.0216	0.012	0.989	0.985	0.999	0.991
100882 Switch	PARIS	40.23	0.991	0.0418	0.082	0.989	0.822	0.999	0.937
	DTA [†]	N/A	N/A	N/A	0.020	0.976	0.676	0.998	0.883
	SplArt	44.04	0.997	0.0174	0.033	0.990	0.835	0.999	0.941
101542 Dispenser	PARIS	31.32	0.966	0.0450	0.105	0.883	0.623	0.998	0.835
	DTA [†]	F	F	F	F	F	F	F	F
	SplArt	36.53	0.992	0.0153	0.020	0.869	0.585	0.999	0.818
102400 Knife	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	F	F	F	F	F	F	F	F
	SplArt	36.21	0.990	0.0215	0.041	0.986	0.874	0.997	0.952
103031 CoffeeMachine	PARIS	32.29	0.979	0.0659	0.116	0.977	0.825	0.997	0.933
	DTA [†]	F	F	F	F	F	F	F	F
	SplArt	37.20	0.993	0.0284	0.034	0.994	0.975	0.999	0.990
mean	PARIS	32.75	0.953	0.0746	0.125	0.878	0.820	0.995	0.897
	DTA [†]	N/A	N/A	N/A	0.053	0.886	0.780	0.990	0.886
	SplArt	36.38	0.980	0.0309	0.032	0.883	0.811	0.997	0.897

Table 11. SPLART-PMS Novel View and Articulation Synthesis Metrics on Revolute Scenes. [†]DTA requires ground-truth depth.

Scene	Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Depth MAE \downarrow	IoU _s \uparrow	IoU _m \uparrow	IoU _{bg} \uparrow	mIoU \uparrow
3558 Bottle	PARIS	35.75	0.981	0.0276	0.120	0.976	0.676	0.999	0.883
	DTA [†]	N/A	N/A	N/A	0.026	0.929	0.580	0.999	0.836
	SplArt	42.05	0.996	0.0076	0.034	0.994	0.968	1.000	0.987
12085 Dishwasher	PARIS	30.81	0.954	0.0927	0.111	0.964	0.929	0.994	0.962
	DTA [†]	N/A	N/A	N/A	0.113	0.873	0.773	0.982	0.876
	SplArt	35.70	0.988	0.0388	0.019	0.985	0.965	0.998	0.983
27189 Table	PARIS	29.55	0.887	0.1618	0.131	0.958	0.866	0.995	0.940
	DTA [†]	N/A	N/A	N/A	0.044	0.964	0.889	0.992	0.948
	SplArt	34.76	0.963	0.0560	0.035	0.979	0.949	0.998	0.975
100248 Suitcase	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	N/A	N/A	N/A	0.025	0.986	0.574	0.998	0.853
	SplArt	39.76	0.995	0.0104	0.038	0.995	0.934	0.999	0.976
101713 Pen	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	N/A	N/A	N/A	0.037	0.956	0.319	0.999	0.758
	SplArt	40.85	0.998	0.0135	0.028	0.981	0.789	1.000	0.923
102016 USB	PARIS	31.45	0.970	0.1005	0.191	0.731	0.583	0.994	0.769
	DTA [†]	N/A	N/A	N/A	0.157	0.710	0.553	0.979	0.747
	SplArt	35.08	0.987	0.0475	0.054	0.659	0.533	0.998	0.730
102812 Switch	PARIS	35.23	0.984	0.0433	0.102	0.987	0.908	0.998	0.964
	DTA [†]	N/A	N/A	N/A	0.024	0.970	0.616	0.996	0.861
	SplArt	38.69	0.995	0.0235	0.031	0.995	0.971	0.999	0.988
103042 Window	PARIS	29.11	0.967	0.0808	0.127	0.859	0.631	0.996	0.829
	DTA [†]	N/A	N/A	N/A	0.088	0.900	0.744	0.991	0.878
	SplArt	32.35	0.984	0.0499	0.039	0.868	0.695	0.999	0.854
103549 Toaster	PARIS	F	F	F	F	F	F	F	F
	DTA [†]	F	F	F	F	F	F	F	F
	SplArt	39.04	0.994	0.0195	0.036	0.997	0.949	0.999	0.982
103941 Phone	PARIS	34.21	0.972	0.0592	0.089	0.938	0.956	0.998	0.964
	DTA [†]	N/A	N/A	N/A	0.027	0.913	0.952	0.997	0.954
	SplArt	39.42	0.993	0.0219	0.020	0.971	0.980	0.999	0.984
mean	PARIS	32.30	0.959	0.0808	0.124	0.916	0.793	0.996	0.902
	DTA [†]	N/A	N/A	N/A	0.060	0.911	0.667	0.993	0.857
	SplArt	37.77	0.989	0.0288	0.033	0.942	0.873	0.999	0.938

Table 12. SPLART-PMS Novel View and Articulation Synthesis Metrics on Prismatic Scenes. [†]DTA requires ground-truth depth.