This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

FreeArtGS: Articulated Gaussian Splatting Under Free-moving Scenario

Hang Dai[1,3*] Hongwei Fan[1,3*] Han Zhang[2,3*] Duojin Wu[1,3] Jiyao Zhang[1,3] Hao Dong[1,3†]
[1]CFCS, School of Computer Science, Peking University
[2]College of Computer Science and Technology, Zhejiang University [3]PrimeBot

Abstract

The increasing demand for augmented reality and robotics is driving the need for articulated object reconstruction with high scalability. However, existing settings for reconstructing from discrete articulation states or casual monocular videos require non-trivial axis alignment or suffer from insufficient coverage, limiting their applicability. In this paper, we introduce FreeArtGS, a novel method for reconstructing articulated objects under free-moving scenario, a new setting with a simple setup and high scalability. FreeArtGS combines free-moving part segmentation with joint estimation and end-to-end optimization, taking only a monocular RGB-D video as input. By optimizing with the priors from off-the-shelf point-tracking and feature models, the free-moving part segmentation module identifies rigid parts from relative motion under unconstrained capture. The joint estimation module calibrates the unified object-to-camera poses and recovers joint type and axis robustly from part segmentation. Finally, 3DGS-based end-to-end optimization is implemented to jointly reconstruct visual textures, geometry, and joint angles of the articulated object. We conduct experiments on two benchmarks and real-world free-moving articulated objects. Experimental results demonstrate that FreeArtGS consistently excels in reconstructing free-moving articulated objects and remains highly competitive in previous reconstruction settings, proving itself a practical and effective solution for realistic asset generation. The project page is available at: https://freeartgs.github.io/

1. Introduction

Articulated objects broadly exist and are frequently interacted with in our daily lives. Building digital replicas of interactable articulated objects not only enhances the human experience in augmented reality [41], but also reduces the sim-to-real gap [3, 11, 14, 37, 46] for robot learning. To efficiently expand the scope of graphics- and simulation-ready assets, the reconstruction system for articulated objects should achieve *high scalability* in a *simple setup*.

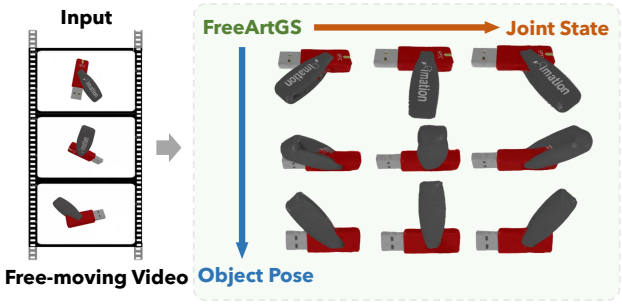

Figure 1. FreeArtGS reconstructs articulated object under the free-moving scenario, in which both joint and object pose move in an unconstrained manner.

Regarding the scalability and simplicity of reconstructing articulated objects, recent works can be separated into three lines. The first line of works [10, 16, 22] directly generates the articulated object assets from a single-view image with foundation models. These methods fail to generalize to unseen scenarios due to the scarcity of post-optimization. The second line of works [17, 20, 38] assumes that the object is captured in two articulation states (e.g., open-closed, pulled-pushed) with fixed multi-view cameras. However, these methods require alignment of the axes between the two states, limiting the real-world usage. The third line of works [15, 24, 49] reconstructs the object from casual monocular video with a static base part and moving dynamic parts. These works expose two disadvantages. First, during real-world casual capture, the pose of many articulated objects, such as scissors and pliers, may be inadvertently altered, violating the assumption of a static base part. Second, the coverage of the object is insufficient, limiting its usage in asset generation. These drawbacks motivate the need for reconstructing articulated objects in *free-moving scenario*, in which both **joint state** and **object pose** relative to the camera vary arbitrarily (Fig. 1). By reconstructing the articulated object in a free-moving manner, the object is captured with full coverage of both object pose and joint state, resulting in an interactable, simulation-ready asset.

In this paper, we introduce **FreeArtGS**, a reconstruction

*: Equal contributions. †: Corresponding author.

system for articulated objects under the free-moving scenario. Our system consists of three key modules: **(1) Free-moving Part Segmentation.** We design an optimization-based part segmentation method based on the visual cues from dense 2D tracks [6] and a pretrained feature extractor [29], without assuming a static base part or predefined motion patterns. **(2) Joint Estimation.** We estimate the articulated joints by predicting the part-to-camera transformations, and use the relative transformation between the parts to decide the joint type and axis. **(3) End-to-end Optimization.** We jointly refine the appearance, geometry, cameras, and articulation in an end-to-end manner, with ground-truth RGB, depth, and foreground masks as supervision.

To evaluate the reconstruction quality of our method under the free-moving scenario, we establish a new benchmark, FreeArt-21, including 21 free-moving articulated objects from 7 categories in the PartNet-Mobility dataset [42]. We capture the free-moving object in the simulation engine Sapien [42], and tele-operate the object poses and joint states with a VR system. We also evaluate the method on real-world objects captured by an RGB-D camera. To align the settings of the existing baselines, we further compare our method with them in the Video2Articulation-S dataset. In all three evaluation settings, our method outperforms the current baselines by a large margin.

To summarize, our contributions are threefold:

- We propose FreeArtGS, a system for reconstructing articulated objects in free-moving scenarios, where the joint state and object pose vary arbitrarily without any static base part as a reference. Our approach combines motion-based part segmentation with joint estimation and end-to-end Gaussian Splatting optimization, enabling accurate reconstruction from only a monocular RGB-D video.
- Since there is no previous benchmark on articulated object reconstruction in the free-moving scenario, to bridge this gap, we build FreeArt-21, a simulated benchmark covering 21 free-moving articulated objects from 7 categories. To mimic the free-moving setting as in the real world, we develop a VR system to manipulate the object pose and joint state in the Sapien [42] simulator.
- Experiments on our proposed benchmark FreeArt-21, Video2Articulation-S dataset, and real-world objects demonstrate that FreeArtGS consistently excels in the free-moving setting while remaining competitive in previous reconstruction settings.

2. Related Works

2.1. Articulated Object Reconstruction

Articulated object reconstruction is a long-standing problem in 3D computer vision and has been widely researched in recent years. Feed-forward network-based methods [4, 7, 10, 12, 25, 30, 33, 45] are trained on an annotated dataset, but fail to generalize to unseen objects. To improve the generalization ability, a series of methods leverage the foundation models [3, 16, 22, 26] to generate articulated object models from single-view images. While these approaches benefit from extensive pre-training on diverse datasets, they typically lack geometric consistency constraints and iterative refinement mechanisms. Another series regards articulated object reconstruction as calibrated multi-view camera capturing under two distinct articulation configurations [5, 17, 19, 20, 32, 37, 38]. Despite their geometric rigor, it remains difficult to align the axes of different states, leading to the limited practicality. To address these limitations and improve generalization, recently, a new line of methods [15, 24, 40, 49] reconstruct articulated objects from monocular RGB or RGB-D video sequences under the assumption that one part remains stationary relative to the background. These methods suffer from two fundamental limitations. First, the static base part assumption is violated in practical scenarios where users naturally manipulate objects like scissors or pliers. Second, the inability to freely repose the object during capture results in incomplete coverage. In contrast, our method operates in a free-moving setting, where the object pose and joint state can vary concurrently, eliminating these issues.

2.2. Dynamic Reconstruction

Articulated object reconstruction can be regarded as another type of dynamic reconstruction. Feed-forward methods [21, 34, 48] directly learn to reconstruct dynamic point clouds from large-scale datasets. However, they fail to recover precise motions under the free-moving setting (Fig. 3). Optimization-based dynamic reconstruction methods [2, 8, 18, 39] reconstruct temporal deformations of radiance fields, but often lack generalization ability. Recently, point tracking methods [6, 23, 27, 43, 44, 47] provide generalized priors at pixel-level resolution, enabling their application to free-moving articulated object reconstruction. However, due to the data-driven nature, the tracks inevitably contain noise and outliers. Our method uses an off-the-shelf point tracking model [6] to generate pseudo motion labels, and optimizes the articulated object to fit the labels. In this way, we combine the generalization ability of point tracking and the high accuracy of the optimization-based dynamic reconstruction in one framework.

3. Method

3.1. Overview

Given a monocular RGB-D video of a free-moving articulated object with two rigid parts $\mathcal{V} = \{\mathcal{I}_i, \mathcal{D}_i\}_{i=1}^{N}$ and foreground masks $\{\mathcal{M}_i^{fg}\}_{i=1}^{N}$, obtained using Segment Anything Model [28], our goal is to reconstruct its canonical Gaussians $\mathcal{G}_c = \{\mathcal{G}_c^0, \mathcal{G}_c^1\}$ of the two parts and joint param-

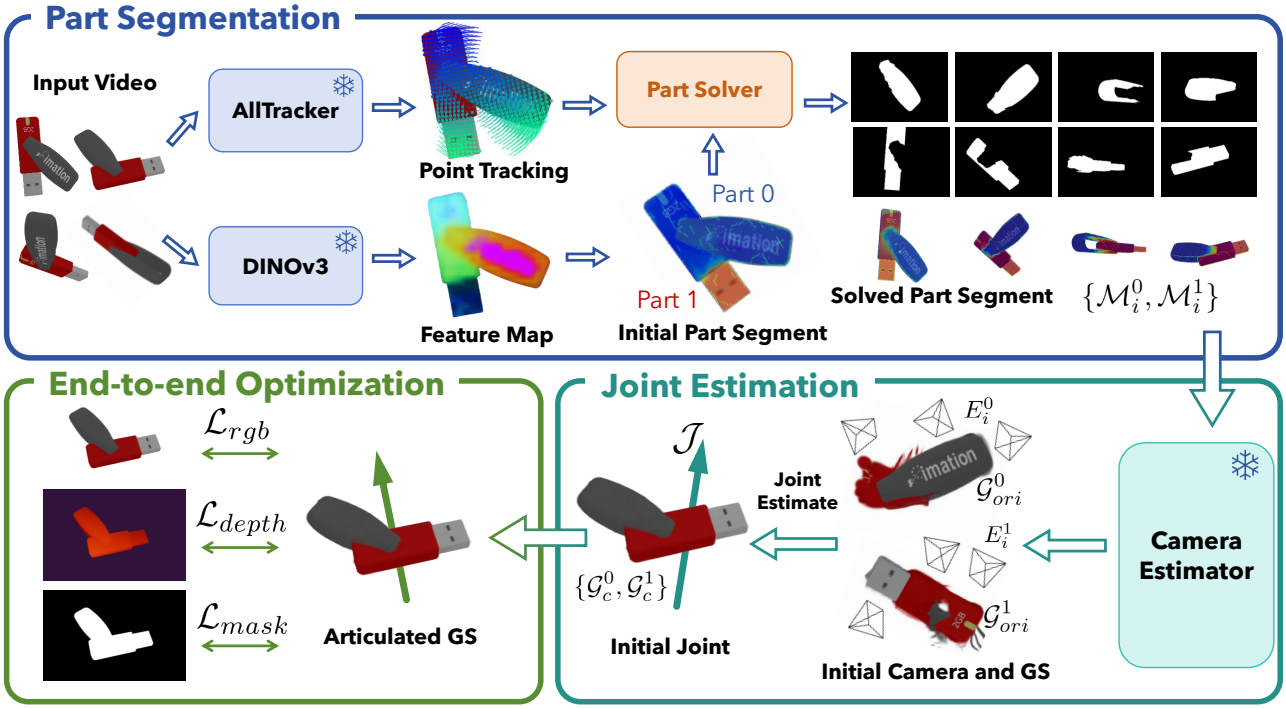

Figure 2. Overview of FreeArtGS. It consists of three modules: (1) Part Segmentation from free-moving video; (2) Joint Estimation with estimated part transforms; (3) End-to-end Optimization for articulated Gaussian Splatting.

eters $\mathcal{J}$. Thus, the object can be represented as $\mathcal{G} = \mathcal{G}_c \circ \mathcal{J}$.

As shown in Fig. 2, our framework includes three modules: part segmentation, joint estimation, and end-to-end optimization. In Sec. 3.2, our method leverages the point tracking results of the two parts to obtain their 2D part segmentation $\mathcal{M} = \{\mathcal{M}_i^0, \mathcal{M}_i^1\}_{i=1}^N$ within the foreground masks. In Sec. 3.3, we reconstruct the Gaussians of the two parts $\mathcal{G}_{ori}^0$ and $\mathcal{G}_{ori}^1$ respectively, coarsely estimate the joint parameters $\mathcal{J}$, and calibrate $\{\mathcal{G}_{ori}^0, \mathcal{G}_{ori}^1\}$ and $\mathcal{J}$ to the canonical Gaussians $\{\mathcal{G}_c^0, \mathcal{G}_c^1\}$. In Sec. 3.4, we perform blended rendering and fine-tune the Gaussians and joint parameters with an end-to-end optimization.

3.2. Free-moving Part Segmentation

Setting. We first aim to segment the articulated object into two rigid parts purely from motion. Our key assumption is that within a short temporal window, the motion of each part is well-approximated by an independent rigid transform. Specifically, for a frame pair (t, t') in the window and a tracked pixel p with valid depth, we obtain the corresponding 3D points $X_{t,p}$ and $X_{t',p}$. We seek two rigid transforms $T_{t\to t'}^0$ and $T_{t\to t'}^1$ and a soft part weight $w_{t,p} \in [0, 1]$ per point, where $w_{t,p} \approx 1$ and $w_{t,p} \approx 0$ denote the two different parts. We initialize the part weight by clustering with a feature map from DINOv3 [29], where the semantically similar points are assigned the same part label.

Part Solver. We process the RGB-D video $\mathcal{V}$ with a sliding window of size n. For each window, assume $t = 0$ for the first frame of the window, AllTracker [6] provides pixel-level 2D trajectories through n frames $\{u_{t,p}\}_{t\in[0,n-1]} \in \mathbb{R}^2$, where $p \in \mathbb{Z}^2$ is the pixel index of the tracked points in the first frame of the window and $u_{t,p}$ denotes its 2D position at the frame t in the window. Then we lift them to 3D trajectories $\{X_{t,p}\} \in \mathbb{R}^3$ with depth and camera intrinsics. With these trajectories, we further optimize the rigid transform and part weight $w_{t,p}$ for each part:

$$\mathcal{L}_{\text{main}} = \sum_p (1 - w_{t,p})\rho\left(\frac{\left\|T_{0\to t}^0 X_{0,p} - X_{t,p}\right\|}{\left\|X_{0,p} - X_{t,p}\right\| + \epsilon}\right) + w_{t,p}\rho\left(\frac{\left\|T_{0\to t}^1 X_{0,p} - X_{t,p}\right\|}{\left\|X_{0,p} - X_{t,p}\right\| + \epsilon}\right),$$

where $\rho(\cdot)$ is Huber loss and ϵ is a small constant to avoid division by zero.

Regularization. Jointly optimizing the transform and part weight may fall into a sub-optimal solution. To this end, we regularize $w_{t,p}$ to be confident and spatially coherent. First, an entropy penalty encourages near-binary assignments,

$$\mathcal{L}_{\text{ent}} = -\sum_p \Big[w_{t,p}\log w_{t,p} + (1 - w_{t,p})\log(1 - w_{t,p})\Big].$$

Second, to prevent the model from fitting the unstable

point tracking results, we build a feature-space neighbor graph $\mathcal{N}(p)$ by sampling per-pixel image features at tracked points and connecting radius-based neighbors with weights α_{pq}. A smoothness term enforces local consistency,

$$\mathcal{L}_{\text{smooth}} = \sum_{p} \sum_{q \in \mathcal{N}(p)} \alpha_{pq} \left| w_{t,p} - w_{t,q} \right|.$$

Finally, we discourage part weights from being too different from the initial weights $w_{0,p}$ with BCE loss $\mathcal{L}_{\text{init}}$,

$$\mathcal{L}_{\text{init}} = \sum_{p} \text{BCE}(w_{t,p},\, w_{0,p})\,.$$

Our objective per window is

$$\mathcal{L} = \lambda_m \mathcal{L}_{\text{main}} + \lambda_s \mathcal{L}_{\text{smooth}} + \lambda_e \mathcal{L}_{\text{ent}} + \lambda_{\text{init}} \mathcal{L}_{\text{init}}.$$

Part Segmentation. We propagate the optimized part weights $\{w_{i,p}\}$ across windows, fill the unobserved pixels via feature-space [29] neighbors, and obtain the binary part masks $\{\mathcal{M}_i^0, \mathcal{M}_i^1\}_{i=1}^N$ by thresholding the part weights at 0.5. Refer to the supplementary materials for the details.

3.3. Joint Estimation

Part-level Reconstruction. With $\mathcal{I}_i$, $\mathcal{D}_i$ and $\{\mathcal{M}_i^k\}_{k\in\{0,1\}}$, where i and k are the frame and part index, we leverage off-the-shelf pose estimators [1, 36] to calibrate each frame's part-to-camera transformations $\{E_i^k\} \in \text{SE}(3)$ for each part. We note that, though we have obtained $T_{t\to t'}^k$ in part segmentation, solving the part-to-camera transformations from all the pairs is not trivial [35], since the motion tracking labels $\{u_{t,p}\}$ contain noise and outliers. In contrast, off-the-shelf pose estimators are robust to sudden visual changes while preserving multi-view consistency. With part masks $\{\mathcal{M}_i^k\}$ and transformations $\{E_i^k\}$, we reconstruct each part $\mathcal{G}_{ori}^k$ and optimize their poses with 3DGS [13].

Pose Calibration. To unify the object-to-camera poses, we calibrate the poses of two parts to a canonical coordinate system by a rigid transform $A^k \in \text{SE}(3)$, which is given as the inverse of E_0^k in the first frame. We choose the part with the least average moving as the reference part (denoted as part 0) and the other part as the relative moving part (denoted as part 1). After calibration, we obtain the canonical Gaussians $\mathcal{G}_c^k = \mathcal{G}_{ori}^k \circ A^k$, transformation $E_i^{ref} = E_i^0 \circ A^0$ of the reference part and $E_i^{rel} = E_i^1 \circ A^1$ of the relative part. By aligning both parts, their trajectories share the same axes in the first frame, and the relative transformation of E_i^{ref} and E_i^{rel} represents the combination of joint state and axes.

Joint Type Estimation. We estimate the kinematic joint that explains the relative transformation between the two reconstructed parts. From $\{E_i^{ref}\}$ and $\{E_i^{rel}\}$, we obtain a sequence of relative part poses $\{T_i\}_{i=1}^N \in \text{SE}(3)$ of one part w.r.t. the other, and implement a light-weight solver that determines the joint type and estimates joint parameters. We decide the joint type by two cues: the overall rotation span across frames and whether translations lie nearly on a single line. A small span with strong linearity indicates a prismatic joint; otherwise, we regard the joint as revolute.

Joint Axis Estimation. For a **revolute** joint, we solve the *closed-form rotation axis* from pairwise relative rotations, recover the per-frame angle by fitting pairwise angle differences with the first frame, and solve the pivot only on the plane orthogonal to the axis to avoid degeneracy. For a **prismatic** joint, we recover the translation axis with *PCA* from $\{T_i\}_{i=1}^N$, project each translation onto the axis to obtain the displacement sequence, and keep a constant rotation.

Noise Resistance. The part poses estimated inevitably contain noise from the off-the-shelf methods. To ensure robustness, there are two key designs in joint estimation. First, instead of directly estimating the joint from the absolute T_i, we solve the pairwise relative transform $T_{i\to(i+1)}$ between neighboring frames. Second, we filter the outlier transforms with a threshold of 2σ, where σ is the standard deviation of the translations of poses in 3D space. These choices make the solver stable under small tracking errors and occasional outlier frames, while remaining fast and light-weight.

3.4. End-to-end Optimization

Joint Formulation. Starting from an estimated joint on the canonical poses detailed in 3.3, we perform an axis-aware end-to-end optimization that jointly refines appearance, geometry, cameras, and articulation. Denoting I as the identity matrix, we parameterize the target part by either a revolute joint with unit axis u, pivot o, and per-frame angle θ_i, or a prismatic joint with axis u and displacement d_i:

$$T_i = \begin{cases} T(\theta_i; u, o) = \big[R(u,\theta_i) \mid (I - R(u,\theta_i))o\big], \\ T(d_i; u) = \big[I \mid d_i u\big]. \end{cases}$$

Blended Rendering and Optimization. We apply the rigid part transformations on the canonical Gaussians $\mathcal{G}_c$ for frame i, and then perform alpha-blend on the Gaussians according to part weights $w = \{w_i\} \in [0,1]$ which represent the the probability of each Gaussian belonging to both parts

$$\mathcal{G}_i = w(\mathcal{G}_c \circ I) \cup (1-w)(\mathcal{G}_c \circ \mathcal{J}_i).$$

Finally, we render frame i with a differentiable renderer

$$\hat{\mathcal{I}}_i = \mathcal{R}\big(\mathcal{G}_i, K_i, E_i^{ref}\big)$$

where K_i is the camera intrinsics.

Supervised with RGB, depth, and foreground masks, we optimize Gaussian parameters for both parts, camera poses, part weights and articulation variables $\{u, o, \{\theta_i\}\}$ or $\{u, \{d_i\}\}$ together. The full objective is

$$\mathcal{L}_{E2E} = \textstyle\sum_i \big(\mathcal{L}_{\text{rgb}}^i + \lambda_{\text{depth}}\, \mathcal{L}_{\text{depth}}^i + \lambda_{\text{mask}}\, \mathcal{L}_{\text{mask}}^i\big)$$

Table 1. Results and Ablation Studies on FreeArt-21. Metrics are reported as mean ± std over all test videos. Lower (↓) is better for the joint and geometry metrics, higher (↑) is better for the rendering metric. The best results are highlighted in **bold**.

Joint Type	Method	Axis (deg)↓	Position (cm)↓	State (deg/cm)↓	CD-w (cm)↓	CD-m (cm)↓	CD-s (cm)↓	PSNR (dB)↑
Revolute	ArticulateAnything [16]	42.00±46.48	59.38±62.19	-	-	-	-	-
	Video2Articulation [24]	20.00±28.81	16.31±28.78	27.37±19.06	2.29±3.40	10.74±16.35	1.87±1.59	-
	Ours w/o Smooth Loss	28.01±29.23	17.73±30.79	18.74±20.96	5.72±7.39	14.37±15.89	5.64±9.17	10.60±4.61
	Ours w/o Init Loss	9.35±13.50	19.58±39.17	14.64±19.90	0.75±0.90	1.90±3.11	1.14±2.60	13.07±7.26
	Ours w/o Noise Resistance	4.75±7.83	2.22±6.89	**1.30±1.09**	0.17±0.15	0.48±0.79	1.10±2.72	22.65±3.13
	Ours w/o Blended Rendering	1.72±2.38	1.88±6.23	1.88±2.54	**0.12±0.07**	0.34±0.52	1.05±2.61	22.23±2.64
	Ours	**1.04±1.03**	**0.29±0.36**	1.43±1.20	0.14±0.13	**0.28±0.36**	**0.97±2.69**	**24.02±3.15**
Prismatic	ArticulateAnything [16]	45.00±49.30	-	-	-	-	-	-
	Video2Articulation [24]	18.47±22.83	-	13.98±21.47	1.51±2.23	8.41±14.57	5.31±15.37	-
	Ours w/o Smooth Loss	26.97±23.75	-	46.91±58.06	8.21±11.16	11.82±11.37	10.16±13.62	12.63±6.46
	Ours w/o Init Loss	28.72±27.83	-	38.31±38.81	22.3±50.18	23.73±44.37	21.62±45.33	13.86±4.48
	Ours w/o Noise Resistance	11.90±26.93	-	16.78±38.86	0.87±1.02	5.85±13.24	0.90±1.50	20.18±3.36
	Ours w/o Blended Rendering	2.04±3.01	-	0.97±0.39	0.45±0.24	**0.50±0.42**	**0.28±0.13**	20.24±2.64
	Ours	**1.85±2.75**	**-**	**0.90±0.53**	**0.41±0.22**	0.67±0.34	0.30±0.13	**22.92±2.65**

where

$$\mathcal{L}^i_{\text{rgb}} = (1-\lambda_{\text{ssim}})\,\mathcal{L}_1(\hat{\mathcal{I}}_i, \mathcal{I}_i) + \lambda_{\text{ssim}}\,\mathcal{L}_{ssim}(\hat{\mathcal{I}}_i, \mathcal{I}_i)$$

$$\mathcal{L}^i_{\text{depth}} = \left|\hat{\mathcal{D}}_i - \mathcal{D}_i\right|$$

$$\mathcal{L}^i_{\text{mask}} = |A_i - \mathcal{M}_i|$$

$\mathcal{L}_1$ and $\mathcal{L}_{ssim}$ follow the definitions in [13]. This stage tightly couples appearance with kinematics and corrects small biases from the coarse joint, producing a high-fidelity articulated 3DGS.

4. Experiments

4.1. Experimental Settings

Datasets. We evaluate the reconstruction performance of our method on the following datasets:

(1) New Benchmark: FreeArt-21. As there is no existing benchmark for free-moving articulated object reconstruction, to evaluate our method, we propose FreeArt-21, a new benchmark containing 21 objects of 7 different categories (5 revolute and 2 prismatic) from the PartNet-Mobility dataset [42]. To simulate the free-moving setting, we deploy a VR system that is widely used in augmented reality and robotics, place the object in front of a fixed RGB-D camera, and teleoperate the object. Please refer to the supplementary materials for the details of our benchmark.

(2) Video2Articulation-S. Video2Articulation-S is a synthetic dataset of two-part articulated objects proposed by Video2Articulation [24]. It consists of 73 test videos across 11 categories of synthetic objects from the PartNet-Mobility dataset, where each object has a static base part. Since a static-base capture can be regarded as a special case of the free-moving setting, our method is also compatible with it.

(3) Real-world Articulated Objects. We evaluate our method on six daily articulated objects, including five revolute-joint objects and one prismatic-joint object. The free-moving videos are captured while the objects are held by hand. We fix an Orbbec Femto Bolt RGB-D camera, and the object holder stands at a distance of 30-50cm from the camera. We report both qualitative and quantitative results.

Evaluation Metrics. We report the following core metrics: (1) Joint axis error (degree): both for revolute and prismatic joints, the angle between the predicted unit axis u and ground truth u_{gt}, (2) Joint position error (cm): only for revolute joints, the Euclidean distance between predicted pivot o and ground-truth pivot o_{gt}, (3) State (degree/cm): both for revolute and prismatic joints, the absolute difference between the predicted joint state and the ground truth joint state. (4) Chamfer Distance (cm): symmetric ℓ_2 Chamfer distance between reconstructed and ground-truth surfaces. We report CD on the whole object (CD-w) and separately on the *moving* part (CD-m) and the *reference* part (CD-s). All distances are computed in the canonical state and reported in centimeters. (5) PSNR: On our FreeArt-21 dataset, we additionally report PSNR of novel views and joint states measured inside the foreground mask.

Baselines. We choose Video2Articulation [24], Robot-See-Robot-Do (RSRD) [14] and Articulate-Anything [16] as our baseline methods. Articulate-Anything is a foundation model-based method that predicts the whole URDF from only a single image input. Since it also uses PartNet-Mobility as the URDF template for retrieval, the domain gap is reduced. For Articulate-Anything, we use the GPT-4o [9] model as the vision-language model. Additionally, we provide the ID of each case to the model, which means the model only needs to predict the correct joint, rather than jointly inferring both the object mesh and the joint. RSRD first reconstructs the object with a smartphone scan and recovers the part motion from a monocular video. Since FreeArt-21 only contains free-moving object frames, we only compare its performance on Video2Articulation-S. Video2Articulation leverages the feed-forward point map models to predict the dynamics. Although not the same as

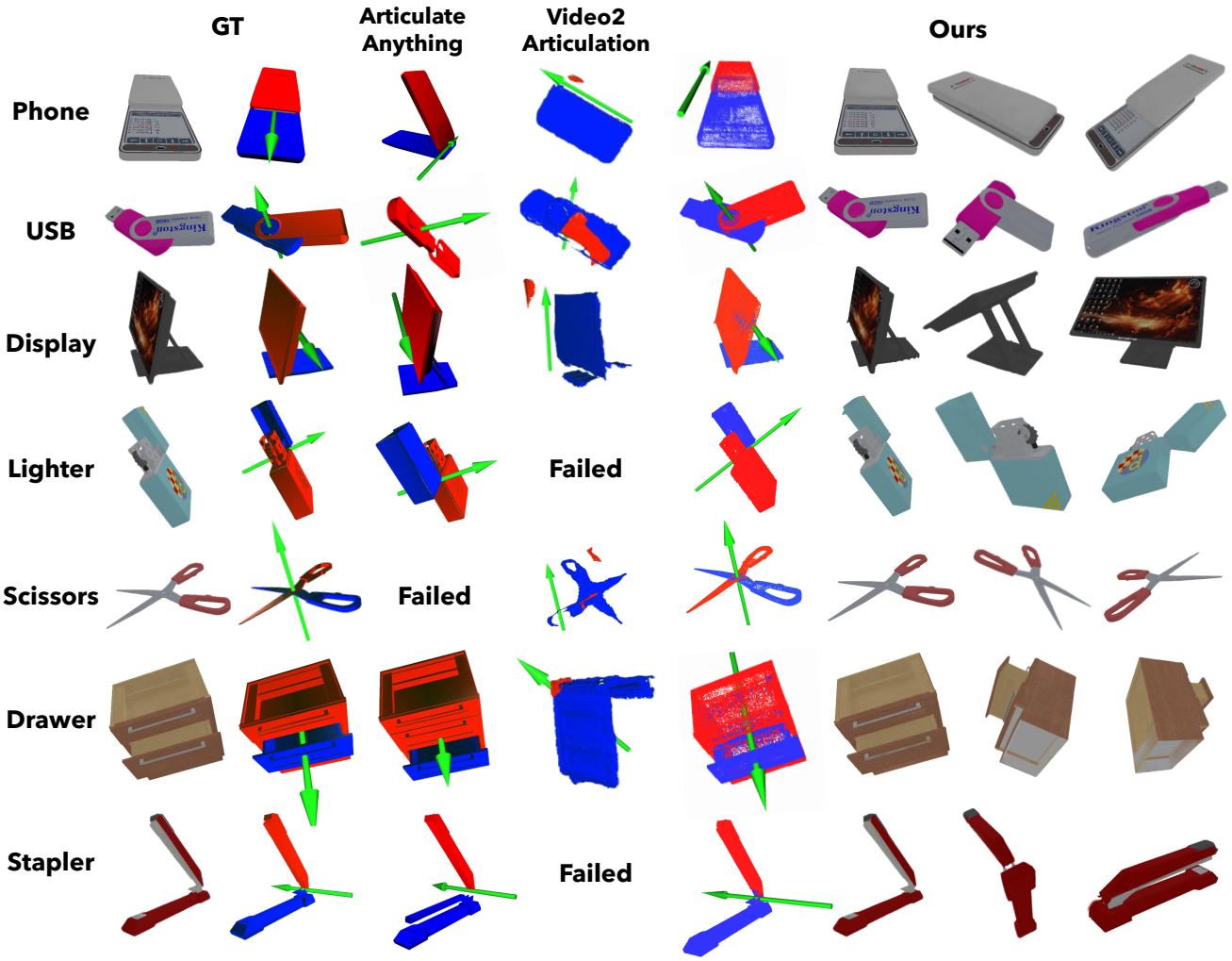

Figure 3. Qualitative Results on FreeArt-21. We visualize both the articulation and rendering results of our method. The red part and the blue part are the identified parts by our method.

the free-moving scenario, it is the latest open-source state-of-the-art method in the setting closest to ours.

4.2. Implementation Details

For all modules, we maintain a unified set of hyperparameters across all datasets, avoiding per-case tuning.

Part segmentation. For the optimization process, we employ the Adam optimizer with a learning rate of 1e-4 for the rigid transforms $T^0_{t\to t'}$ and $T^1_{t\to t'}$, and 1e-2 for the part weight $w_{t,p}$. A sliding window of 8 frames is defined, within which we optimize frame pairs anchored by the first frame, specifically $(0, i)$ for $i \in \{1, 2, \ldots, 7\}$. Each pair undergoes 100 iterations of optimization. The loss weights are: $\lambda_m = 200$, $\lambda_s = 10$, $\lambda_{init} = 5$, and $\lambda_e = 0.01$.

Joint estimation and end-to-end optimization. Our implementation is based on NeRFStudio [31] and its default parameters. During reconstruction, we optimize for 30000 iterations in both part-level reconstruction and end-to-end optimization. In the stage of part-level reconstruction, we choose $\lambda_{depth} = 1.0, \lambda_{mask} = 0.01$ while in the end-to-end optimization, $\lambda_{depth} = 1.0, \lambda_{mask} = 1.0$.

Hardware and time cost. We evaluate the running times of our method and the two baselines on a workstation equipped with an Intel i9-14900K CPU and an NVIDIA RTX 4090 GPU. Given an input RGB-D video with 100 frames and a resolution of 640×360, our method takes ∼25 minutes, including 6 minutes for part segmentation, 1 minute for joint estimation, and 18 minutes for end-to-end optimization.

4.3. Results on FreeArt-21

As shown in Table 1, across all the 21 objects in the dataset, our method achieves an average error of around 1 degree

Table 2. Results on Video2Articulation-S Dataset. We report joint estimation results (top) and geometry reconstruction results (bottom). The best results are in **bold**.

Joint Type	Method	Axis (deg)↓	Position (cm)↓	State (deg/cm)↓
Revolute	Articulate-Anything [16]	46.98±45.27	81.00±40.00	N/A
	RSRD [15]	67.06±29.22	203.00±748.00	59.02±34.38
	Video2Articulation [24]	18.34±32.09	13.00±25.00	14.32±26.35
	Ours	**1.77±2.87**	**1.31±1.81**	**3.69±6.60**
Prismatic	Articulate-Anything [16]	52.71±44.69	-	N/A
	RSRD [15]	69.91±24.07	-	70.00±48.00
	Video2Articulation [24]	13.75±18.91	-	8.00±22.00
	Ours	**0.77±2.30**	-	**1.00±2.19**
Task	**Method**	**CD-w (cm)↓**	**CD-m (cm)↓**	**CD-s (cm)↓**
Geometry Reconstruction	Articulate-Anything [16]	11.00±22.00	59.00±73.00	7.00±18.00
	RSRD [15]	339.00±2147.00	82.00±117.00	14.00±41.00
	Video2Articulation [24]	**1.00±1.00**	13.00±26.00	6.00±19.00
	Ours	1.87±2.19	**1.00±2.22**	**2.39±2.50**

in axis angle and less than 1 cm in geometry, surpassing all the baselines. FreeArtGS also achieves a plausible PSNR result, while the two baselines fail to recover the precise visual textures. Since the rendering quality is a synergy of pose estimation and visual reconstruction, incorporating a better pose estimation method may result in a higher PSNR. Figure 3 presents qualitative results across all categories. The visualization results demonstrate that our method jointly achieves high fidelity in articulation, geometry, and rendering. On challenging thin objects such as scissors, staplers, phones, and USBs, our method precisely reconstructs the geometry, consistent with Table 1. This indicates the robustness of our part segmentation module. Although using PartNet-Mobility as the asset library, Articulate-Anything [16] often fails to predict the correct part and joint axis, likely due to error accumulation in the vision-language reasoning process. Video2Articulation also performs poorly on our dataset, since Monst3R [48] fails to predict the moving part in the free-moving scenario.

4.4. Results on Video2Articulation-S

As shown in Table 2, although under a similar yet different setting, our method also surpasses all baselines on most metrics, consistent with the results on FreeArt-21. RSRD performs worst on all metrics, due to its assumption that the moving patterns of each part are unique, while for articulated objects, their motions are related by the joint transformation. Articulate-Anything also predicts incorrect assets in most cases, likely due to hallucination in the vision-language model. Regarding Video2Articulation, it should be noted that, even under its own setting, the performance of Video2Articulation is still worse than our method.

Table 3. Quantitative Results on Real-world Objects. The rotation and translation errors are clipped to 0.1° and 0.1 cm, respectively, which correspond to the smallest annotation units.

Objects	Axis (deg)↓	Position (cm)↓	CD (cm)↓	PSNR (dB)↑
Drawer	1.06	-	1.91	19.82
Bin	4.68	0.20	3.46	23.75
Case	0.10	0.10	1.89	22.77
Lid	8.92	0.40	3.36	22.39
Fan	1.53	1.30	1.78	21.95
Cabinet	0.10	3.30	2.53	23.71
Average	2.73	1.06	2.48	22.40

The main reason is that Video2Articulation depends on the predictions from a pretrained feed-forward reconstruction model, which is not robust due to the confidence threshold. Instead, our method only uses the off-the-shelf models as initialization and partial supervision. Combining the priors with optimization is key to the performance gain.

4.5. Results on Real-world Articulated Objects

We further evaluate FreeArtGS on six real-world objects, including a drawer, a trash bin, a case, a bottle lid, an electric fan, and a cabinet. As shown in Figure 4 and Table 3, our method can not only predict the correct joint type and axis, but also reconstruct precise geometry and textures across all six objects. During the data collection, some areas of the objects are inevitably occluded by human hands. However, as can be seen in the figure, our method is robust to this occlusion. There are two reasons. First, the regularization term of the part segmentation module can resist implausible part weights. Second, the end-to-end optimization from RGB-D images corrects the outlier points. These re-

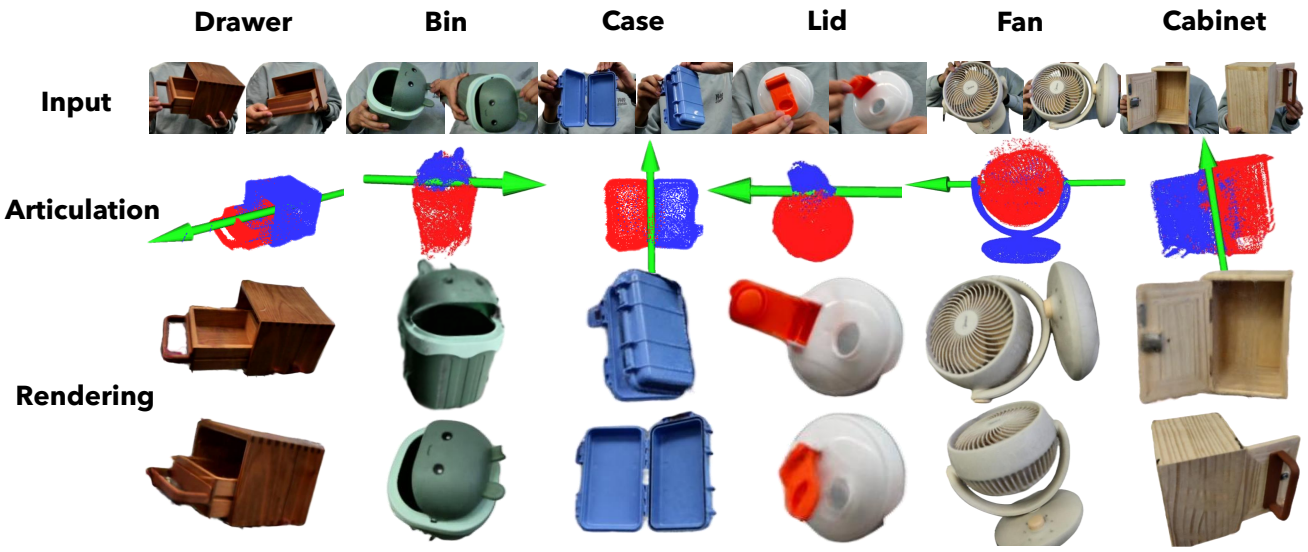

Figure 4. Qualitative Results on Real-world Objects. Our method successfully reconstructs all the objects with correct joints, geometries, and textures.

sults highlight the potential of FreeArtGS as a scalable digital twin reconstruction for real-world articulated objects.

4.6. Ablation Study

Settings. To verify the effectiveness of each component in our method, we conduct four ablation studies on the FreeArt-21 dataset. For part segmentation, we ablate both the smoothness loss and the initialization regularization term, denoted as **w/o Smooth Loss** and **w/o Init Loss**. To validate the effectiveness of noise resistance in joint estimation, we remove the outlier filtering and use absolute transforms for initialization, denoted as **w/o Noise Resistance**. For the blended rendering in end-to-end optimization, we replace the blending with hard assignment, meaning that the part weights remain fixed at 0 or 1 and will not be refined during optimization, denoted as **w/o Blended Rendering**.

Results. The results of the ablation study are shown in Table 1, from which we make the following observations: (1) Smoothness over the neighbor graph and consistency with the initialization are important for both joint and geometry in the ablation. This indicates that although the point tracking model can find the correspondence between neighboring frames, its instability may drive the part solver toward suboptimal solutions. (2) Noise Resistance of joint estimation prevents the joint from overfitting the outlier part transforms, as can be seen from the sudden degradation of the axis angles for both revolute and prismatic joints. (3) Blended Rendering improves the visual rendering quality by around 2 dB. For the few metrics in which removing this module yields slightly better results, the difference in metrics is trivial ($\sim$1mm and $\sim$0.1deg/cm). We include this module since it improves rendering quality while maintaining joint accuracy. This is consistent with its role in refining part weights during end-to-end optimization.

5. Conclusion

In this paper, we propose FreeArtGS, a novel method for reconstructing free-moving articulated objects from monocular RGB-D videos. Our method first segments the free-moving parts by combining an optimization-based method with point-tracking priors. Based on the estimated part segments and transformations, it then infers the joint type and axis by fitting the relative motion between parts. Finally, a 3DGS-based end-to-end optimization jointly refines the joint parameters, geometry, and appearance. Experiments demonstrate the robustness and effectiveness of our method in both simulated and real-world settings. With the growing need to rapidly expand articulated digital twins for augmented reality and robotics, our method provides a promising solution with fewer constraints and higher scalability.

FreeArtGS still has several limitations. First, our method currently assumes a two-part articulated object; extending it to multi-part structures by sequentially capturing moving parts remains an important direction. Second, relying on multiple off-the-shelf priors can lead to cascading error accumulation. A potential solution lies in developing a unified feed-forward model to simultaneously predict joints, poses, geometry, and textures. Third, our framework requires monocular RGB-D input. While extending it to RGB-only sequences by predicting continuous video depth is a natural progression, it currently faces challenges regarding depth accuracy. We leave these as future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62136001). We would like to thank Jinghang Wu from Peking University for technical support.

References

[1] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987. 4

[2] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2

[3] Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024. 1, 2

[4] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: 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*, pages 7081–7091, 2023. 2

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

[6] Adam W Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, et al. Alltracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5253–5262, 2025. 2, 3

[7] 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*, pages 21201–21210, 2023. 2

[8] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4220–4230, 2024. 2

[9] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 5

[10] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[11] Zhao Jin, Zhengping Che, Zhen Zhao, Kun Wu, Yuheng Zhang, Yinuo Zhao, Zehui Liu, Qiang Zhang, Xiaozhu Ju, Jing Tian, et al. Artvip: Articulated digital assets of visual realism, modular interaction, and physical fidelity for robot learning. *arXiv preprint arXiv:2506.04941*, 2025. 1

[12] Yuki Kawana and Tatsuya Harada. Detection based part-level articulated object reconstruction from single rgbd image. *Advances in Neural Information Processing Systems*, 36:18444–18473, 2023. 2

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 4, 5

[14] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *8th Annual Conference on Robot Learning*. 1, 5

[15] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *Conference on Robot Learning*, pages 587–603. PMLR, 2025. 1, 2, 7

[16] 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. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 5, 7

[17] Shengjie Lin, Jiading Fang, Muhammad Zubair Irshad, Vitor Campagnolo Guizilini, Rares Andrei Ambrus, Greg Shakhnarovich, and Matthew R. Walter. Splart: Articulation estimation and part-level reconstruction with 3d gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8841–8851, 2025. 1, 2

[18] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21136–21145, 2024. 2

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

[20] 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 *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2

[21] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3r: Aligned monocular depth estimation for dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22820–22830, 2025. 2

[22] Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code genera-

tion. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2

[23] Tuan Duc Ngo, Peiye Zhuang, Evangelos Kalogerakis, Chuang Gan, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. In *The Thirteenth International Conference on Learning Representations*. 2

[24] Weikun Peng, Jun Lv, Cewu Lu, and Manolis Savva. itaco: Interactable digital twins of articulated objects from casually captured rgbd videos, 2025. 1, 2, 5, 7

[25] Shengyi Qian, Linyi Jin, Chris Rockwell, Siyi Chen, and David F Fouhey. Understanding 3d object articulation in internet videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1599–1609, 2022. 2

[26] Xiaowen Qiu, Jincheng Yang, Yian Wang, Zhehuan Chen, Yufei Wang, Tsun-Hsuan Wang, Zhou Xian, and Chuang Gan. Articulate anymesh: Open-vocabulary 3d articulated objects modeling. *arXiv preprint arXiv:2502.02590*, 2025. 2

[27] Frano Rajič, Haofei Xu, Marko Mihajlovic, Siyuan Li, Irem Demir, Emircan Gündoğdu, Lei Ke, Sergey Prokudin, Marc Pollefeys, and Siyu Tang. Multi-view 3d point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 59–68, 2025. 2

[28] 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. In *The Thirteenth International Conference on Learning Representations*. 2

[29] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 2, 3, 4

[30] Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel Chang. Opdmulti: Openable part detection for multiple objects. In *2024 International Conference on 3D Vision (3DV)*, pages 169–178. IEEE, 2024. 2

[31] 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 *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 6

[32] Haowen Wang, Zhen Zhao, Zhao Jin, Zhengping Che, Liang Qiao, Yakun Huang, Zhipeng Fan, Xiuquan Qiao, and Jian Tang. Sm 3: Self-supervised multi-task modeling with multi-view 2d images for articulated objects. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12492–12498. IEEE, 2024. 2

[33] Junbo Wang, Wenhai Liu, Qiaojun Yu, Yang You, Liu Liu, Weiming Wang, and Cewu Lu. Rpmart: Towards robust perception and manipulation for articulated objects. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7270–7277. IEEE, 2024. 2

[34] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 2

[35] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 4

[36] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–617, 2023. 4

[37] 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 *CVPR*, 2024. 1, 2

[38] Di Wu, Liu Liu, Zhou Linli, Anran Huang, Liangtu Song, Qiaojun Yu, Qi Wu, and Cewu Lu. Reartgs: Reconstructing and generating articulated objects via 3d gaussian splatting with geometric and motion constraints, 2025. 1, 2

[39] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024. 2

[40] Mingxuan Wu, Huang Huang, Justin Kerr, Chung Min Kim, Anthony Zhang, Brent Yi, and Angjoo Kanazawa. Predict-optimize-distill: A self-improving cycle for 4d object understanding. *arXiv preprint arXiv:2504.17441*, 2025. 2

[41] Hongchi Xia, Entong Su, Marius Memmel, Arhan Jain, Raymond Yu, Numfor Mbiziwo-Tiapo, Ali Farhadi, Abhishek Gupta, Shenlong Wang, and Wei-Chiu Ma. Drawer: Digital reconstruction and articulation with environment realism. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21771–21782, 2025. 1

[42] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020. 2, 5

[43] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 2

[44] David Yifan Yao, Albert J Zhai, and Shenlong Wang. Uni4d: Unifying visual foundation models for 4d modeling from a single video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1116–1126, 2025. 2

[45] Qiaojun Yu, Junbo Wang, Wenhai Liu, Ce Hao, Liu Liu, Lin Shao, Weiming Wang, and Cewu Lu. Gamma: Generalizable articulation modeling and manipulation for articulated objects. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5419–5426. IEEE, 2024. 2

[46] Qiaojun Yu, Xibin Yuan, Junting Chen, Dongzhe Zheng, Ce Hao, Yang You, Yixing Chen, Yao Mu, Liu Liu, Cewu Lu,

et al. Artgs: 3d gaussian splatting for interactive visual-physical modeling and manipulation of articulated objects. *arXiv preprint arXiv:2507.02600*, 2025. 1

[47] Bowei Zhang, Lei Ke, Adam W Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. *arXiv preprint arXiv:2504.14717*, 2025. 2

[48] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 2, 7

[49] Hongyi Zhou, Yulan Guo, Xiaogang Wang, and Kai Xu. Monomobility: Zero-shot 3d mobility analysis from monocular videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8800–8809, 2025. 1, 2