VIDEOARTGS: BUILDING DIGITAL TWINS OF ARTICULATED OBJECTS FROM MONOCULAR VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

Building digital twins of articulated objects from monocular video requires jointly recovering part geometry and articulation from entangled camera and object motion. Although 3D point tracks provide motion cues, their noise and lack of explicit part structure complicate articulation learning. To address this problem, we introduce VideoArtGS, a motion-prior-guided framework that turns these tracks into structured initialization for joint geometry-and-motion reconstruction. Our pipeline analyzes trajectories under articulation constraints, filters unreliable tracks, and groups them to initialize part centers, joint parameters, and an articulation-based deformation field. We also design a hybrid center-grid part-assignment module that represents movable parts with learnable centers and the static base with a flexible spatial grid. We then jointly optimize 3D Gaussians and the deformation field using rendering and tracking supervision. VideoArtGS achieves state-of-the-art performance in articulation estimation and mesh reconstruction on iTACO-S, iTACO-L, and VideoArtGS-20 datasets. On VideoArtGS-20, which contains objects with up to nine movable parts, mean joint-axis error decreases from 43.65° to 0.34°, a reduction of over **two orders of magnitude**. Visualizations are available at: `https://videoartgs-2026.github.io`.

1 INTRODUCTION

Articulated objects, prevalent in our daily life, are becoming a major focus in recent research for computer vision and robotics (Liu et al., 2024a). Reconstructing interactable digital twins (Weng et al., 2024) from visual observations is fundamental to advancing applications in world simulation (Geng et al., 2025; Lu et al., 2025a), robotic manipulation (Zhao et al., 2024b; Büchner et al., 2026), and scene understanding (Linghu et al., 2025; Liu et al., 2024c). By generating digital twins from video, we can significantly accelerate the development of intelligent systems, particularly by bridging the sim-to-real gap for robotic manipulation and interaction tasks (Torne et al., 2024; Kerr et al., 2024). To build powerful and generalizable robotic systems, reconstructing interactable objects from monocular video represents a critical frontier, as this would unlock the ability to learn from the vast amount of videos available online and allow robots to model the world through their own eyes.

Prior work recovers articulation through feed-forward prediction (Yuan et al., 2025; Li et al., 2025c;a), VLM-based modeling (Le et al., 2025), or reconstruction from multi-view observations in multiple discrete states (Liu et al., 2025; 2023a; Lin et al., 2025b; Yu et al., 2025b; Guo et al., 2025; Kim et al., 2025). These approaches exploit learned priors or geometric constraints, with requirements on training coverage or multi-state capture. Video-based methods have established reconstruction from observed part motion (Song et al., 2024; Kerr et al., 2024; Peng et al., 2025), with subsequent work addressing free-moving objects, functional indoor scenes, and hand–articulated-object interactions (Dai et al., 2026; Delitzas et al., 2026; Wang et al., 2026b). Reconstructing articulated objects from monocular videos simplifies data capture but poses a coupled estimation problem. Camera motion and part motion are entangled in the image observations, while object geometry and part assignments must also be recovered. In particular, estimating a part's joint motion depends on identifying its constituent points, whereas determining which points belong to the same part relies on their motion consistency.

Previous methods, such as Shape-of-Motion (Wang et al., 2025c) and ArtiPoint (Werby et al., 2025), have explored lifting 2D tracks for supervision. More recently, the advent of powerful perception models like SpatialTrackerV2 (Xiao et al., 2025) and TAPIP3D (Zhang et al., 2025a) provides 3D

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tracks, which offer richer motion information. However, noisy trajectories do not directly specify which points belong to the same articulated part or how that part moves. The question is therefore how to turn these imperfect trajectories into a useful initialization of articulated structure, in addition to using them as correspondence supervision.

To address this problem, we propose VideoArtGS, which uses motion priors to guide both initialization and optimization of articulated reconstruction. Our **motion prior guidance pipeline** analyzes trajectories under articulation motion constraints, filters inconsistent tracks, and groups the retained trajectories into coherent parts. The resulting part centers and joint parameters, together with tracking supervision, initialize an articulation-based deformation field before joint reconstruction.

To represent and refine the part structure, we introduce a **hybrid center-grid part assignment module**. Movable parts are represented by learnable centers initialized from motion clusters, whereas the static base is represented by a flexible spatial grid that accommodates irregular geometry. These complementary representations are jointly optimized with the Gaussian geometry and joint parameters. Motion guidance provides structural initialization and temporal supervision, while hybrid assignment refines the part structure during reconstruction.

With these designs, VideoArtGS achieves substantial gains in articulation estimation and part-level reconstruction. Mean revolute-axis error on iTACO-S decreases from 13.83° to 0.32° and mean joint-axis error on VideoArtGS-20 decreases from 43.65° to 0.34°. Evaluation on 711 iTACO-L videos further supports these gains at larger scale. Our ablations indicate that motion-derived part-center initialization and hybrid assignment are particularly important to this performance.

Contributions Our main contributions can be summarized as follows:

1. We propose VideoArtGS, a novel method for articulated object reconstruction from monocular video that achieves state-of-the-art performance. Notably, it reduces key error metrics by up to **two orders of magnitude** over prior work across multiple datasets.
2. We introduce a motion prior guidance framework that analyzes 3D tracking trajectories to robustly initialize the deformation field. We also design a hybrid center-grid part assignment module that accurately segments parts and benefits articulation learning, accommodating complex geometries.
3. We introduce VideoArtGS-20, covering objects with two to nine movable parts, and evaluate VideoArtGS on this dataset as well as iTACO-S and iTACO-L. Extensive comparisons establish the gains in articulation and part geometry, while ablations systematically validate our designs and point out directions for future improvement.

2 Related Work

2.1 Dynamic Scene Reconstruction

Dynamic scene reconstruction is a long-standing challenge in computer vision. A significant line of work focuses on jointly estimating camera poses and scene geometry, often represented as depth maps or point clouds. Pioneering methods like CasualSAM (Zhang et al., 2022) and Mega-SaM (Li et al., 2025b) established robust frameworks for this task.

While the above methods provide camera and geometry information, representing the dynamic scene itself has been revolutionized by 3D Gaussian Splatting (Kerbl et al., 2023). Many 4D extensions learn to deform Gaussians implicitly over time (Jung et al., 2023; Katsumata et al., 2023; Wu et al., 2024; Luiten et al., 2024; Li et al., 2024; Lu et al., 2024b; Lei et al., 2024a; Guo et al., 2024; Qian et al., 2024; Bae et al., 2024; LIU et al., 2025; Wu et al., 2025c), which excels at capturing complex, non-rigid motion but offers no explicit control over an object's underlying structure. Although some methods learn dense tracks by reconstructing videos (Wang et al., 2025c; Lei et al., 2024b), they do not model the articulated object and cannot reconstruct interactive assets from it.

Attempts to add control via superpoints (Huang et al., 2024) or physics engines (Xie et al., 2024; Jiang et al., 2024) have been made, but they either fail to extract accurate physical parameters or require impractical priors. VideoArtGS bridges this gap by integrating an explicit articulation model

directly into the deformable Gaussian framework, enabling high-fidelity reconstruction for articulated objects.

2.2 Articulated Object Reconstruction

Reconstructing articulated objects presents a dual challenge: one must solve for both the part-level geometry and the underlying articulation parameters. One family of methods employs end-to-end models to predict both part segmentation and joint parameters (Heppert et al., 2023; Wei et al., 2022; Kawana et al., 2021; Zhao et al., 2024a; Jiang et al., 2022; Ma et al., 2023; Nie et al., 2022; Hsu et al., 2023; Goyal et al., 2025; Xia et al., 2025; Li et al., 2026a; Lee et al., 2026c; Wang et al., 2026a). The generalization of learned predictors depends on the coverage of their training data and priors.

The dominant paradigm relies on multi-view observations in multiple discrete articulation states (Liu et al., 2025; Tseng et al., 2022; Mu et al., 2021; Lewis et al., 2022; Liu et al., 2023a; Deng et al., 2024; Swaminathan et al., 2024; Noguchi et al., 2022; Zhang et al., 2021; Pillai et al., 2015; Liu et al., 2023c; Wang et al., 2025a; Lewis et al., 2025; Zhang & Lee, 2025; Li et al., 2026b; Lee et al., 2026b; Deng et al., 2025; Wu et al., 2025a). These methods exploit geometric constraints from multi-view observations at fixed articulation states, but require capturing the object in multiple configurations. Knowledge-guided part-level reconstruction offers another source of articulation priors (He et al., 2025). Single-state approaches instead rely on additional priors to infer unobserved articulation (Wu et al., 2025b; Lee et al., 2026a).

Video-based articulated reconstruction has been explored by earlier methods (Song et al., 2024; Kerr et al., 2024; Peng et al., 2025), alongside monocular mobility analysis (Zhou et al., 2025) and subsequent extensions to additional motion and interaction settings (Dai et al., 2026; Delitzas et al., 2026; Wang et al., 2026b). VideoArtGS builds on this established direction. Its contribution is a motion prior guidance pipeline that converts noisy 3D tracks into part and articulation initialization, together with hybrid assignment for refining part geometry and motion. Our experiments evaluate the resulting accuracy on objects with one or multiple movable parts.

3 Method

Given a monocular video sequence $\{I_t\}_{t=1}^{T}$, VideoArtGS reconstructs articulated objects with part meshes $\mathcal{M}$ and articulation parameters Ψ. We first use a geometry estimator (e.g., SpatialTrackerV2 (Xiao et al., 2025) or Depth-Anything-3 (Lin et al., 2025a)) to estimate the depths and camera poses, and then reconstruct the object with 3D Gaussians $\mathcal{G} = \{G_i\}_{i=1}^{N}$ and an articulation-based deformation field $\mathcal{F}$. This field contains a part segmentation module S_ϕ and articulation parameters Ψ (including axis directions $\boldsymbol{d}$, axis origins $\boldsymbol{o}$, and time-variant joint states θ_t) that control the dynamics of each part. We also introduce motion prior from a pre-trained tracking model to guide the initialization and optimization of the deformation field. An overview of VideoArtGS is presented in Fig. 1, with details on key designs provided in the following sections.

3.1 Articulation-based Deformation Field

To model the temporal dynamics of an articulated object, we formulate an articulation-based deformation field $\mathcal{F}$ that transforms a set of canonical Gaussians $G_i^c = \{\boldsymbol{\mu}_i^c, \boldsymbol{r}_i^c, \boldsymbol{s}_i, \sigma_i, \boldsymbol{h}_i\}$ into the observation state $G_i^t = \{\boldsymbol{\mu}_i^t, \boldsymbol{r}_i^t, \boldsymbol{s}_i, \sigma_i, \boldsymbol{h}_i\}$ for any given time t. Since articulation is a rigid process, the intrinsic properties of each Gaussian—its scale ($\boldsymbol{s}_i^c$), opacity (σ_i^c), and appearance ($\boldsymbol{h}_i^c$)—are treated as time-invariant, while its position ($\boldsymbol{\mu}_i^t$) and rotation ($\boldsymbol{r}_i^t$) are time-variant. Following ArtGS (Liu et al., 2025), VideoArtGS first assigns each Gaussian to object parts through a segmentation module $S_\phi(\cdot)$ and then applies the corresponding rigid transformation for each part:

$$\boldsymbol{m}_i = S_\phi(G_i^c), \quad G_i^t = \sum_{k=1}^{K} m_{ik} \cdot \mathcal{T}_k^t(G_i^c) \tag{1}$$

where $\boldsymbol{m}_i = [m_{i1}, \ldots, m_{iK}]$ represents the assignment probabilities of i-th Gaussian to K parts, and $\mathcal{T}_k^t$ denotes the rigid transformation for k-th part at time t. The number of movable parts and joint types (revolute or prismatic) are discrete inputs obtained by Vision Language Models (Appendix A.5). Continuous geometry and joint parameters are estimated by the reconstruction model.

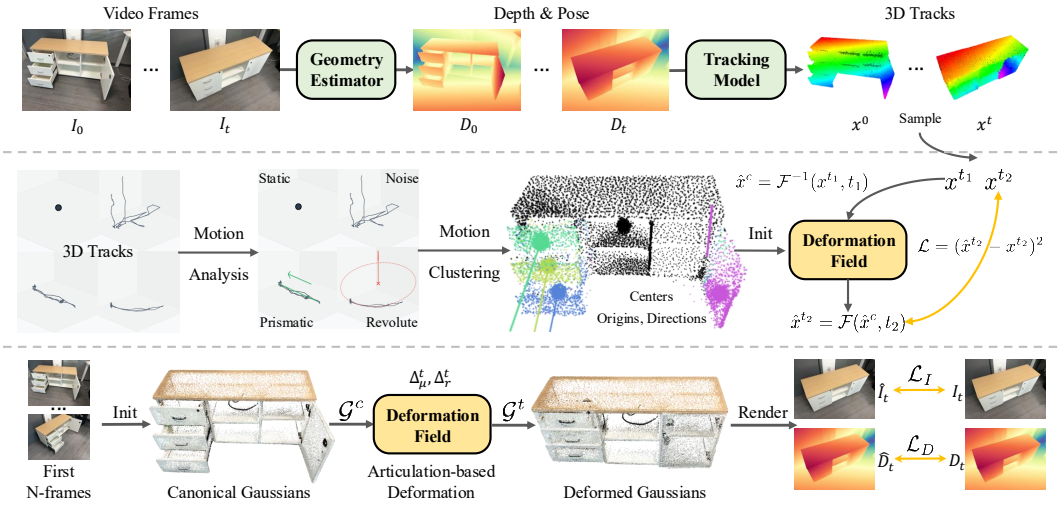


Figure 1: **The overview of VideoArtGS.** Given video frames, we first estimate the depths along with camera poses and then use a tracking model to obtain 3D tracks. We design a motion prior guidance pipeline to analyze and group these tracks, initializing our articulation-based deformation field with motion information and optimizing it with tracking loss. Finally, we reconstruct the object with 3D Gaussians and the deformation field, jointly optimizing both modules by rendering and tracking loss.

Hybrid Center-grid Part Assignment To effectively assign Gaussians to articulable parts, ArtGS (Liu et al., 2025) proposes a center-based part assignment module that segments parts using the Mahalanobis distance between Gaussians and learnable centers. However, this approach faces limitations when the static base part has complex geometries. A simple but key observation is that static regions remain fixed in space. Unlike movable parts that naturally form distinct motion clusters, the static base part is better characterized by its fixed spatial volume rather than a movable center. We therefore propose a hybrid center-grid part assignment module that combines two strategies. For the $K-1$ movable parts, we define learnable part centers $C_k = (\boldsymbol{p}_k, \boldsymbol{V}_k, \boldsymbol{\lambda}_k)$ with center location $\boldsymbol{p}_k \in \mathbb{R}^3$, rotation matrix $\boldsymbol{V}_k \in \mathbb{R}^{3\times3}$, and scale vector $\boldsymbol{\lambda}_k \in \mathbb{R}^3$. For the static base part, we use a learnable hash grid H to model its spatial region directly. Given the canonical position $\boldsymbol{\mu}_i^c$ of a Gaussian G_i^c, we compute its assignment probabilities $\boldsymbol{m}_i$ by fusing scores from both models. First, we compute the squared Mahalanobis distance $\boldsymbol{D}_{i,k}$ to each of the $K-1$ movable part centers:

$$\boldsymbol{D}_{i,k} = \left(\frac{\boldsymbol{V}_k(\boldsymbol{\mu}_i^c - \boldsymbol{p}_k)}{\boldsymbol{\lambda}_k}\right)^\top \left(\frac{\boldsymbol{V}_k(\boldsymbol{\mu}_i^c - \boldsymbol{p}_k)}{\boldsymbol{\lambda}_k}\right) + \Delta_{i,k}, \tag{2}$$

where $\Delta_{i,k}$ is a residual term for improving boundary identification that is introduced by ArtGS (Liu et al., 2025). Simultaneously, we query the hash grid at the Gaussian's position to get a feature vector, which is processed by a small MLP to produce a single logit, $l_i = \text{MLP}(H(\boldsymbol{\mu}_i^c))$, representing the "staticness" score. The final assignment probabilities $\boldsymbol{m}_i \in \mathbb{R}^K$ are obtained by concatenating the static logit with the negative distances of the movable parts and applying a softmax function:

$$\boldsymbol{m}_i = \text{Softmax}\left(\text{concat}\left([l_i, -\boldsymbol{D}_{i,1}, \ldots, -\boldsymbol{D}_{i,K-1}]\right)\right). \tag{3}$$

This hybrid formulation enables robust segmentation by leveraging both structured geometric relationships for movable parts and flexible spatial modeling for complex static regions.

3.2 Motion Prior Guidance

We use a pre-trained tracking model TAPIP3D (Zhang et al., 2025a) to obtain 3D tracking trajectories, providing a motion prior to guide the initialization and optimization of the deformation field.

Motion Pattern Analysis. To identify noise (tracking errors or severe jitter inconsistent with the modeled kinematics) and extract motion information from tracking trajectories, we first analyze the

motion pattern of each trajectory and divide all trajectories into 4 classes: static, prismatic, revolute, and noise. If the maximum displacement distance of the i-th trajectory $\{\boldsymbol{x}_i^t\}_{t=1}^T$ is below a threshold ϵ_s, it is classified as a static trajectory. For the remaining dynamic trajectories, we use line fitting and circle fitting to identify the motion type and motion parameters. However, a trajectory may contain long stationary intervals and only a short period of motion, particularly when different parts move at different times. Repeated samples near stationary positions can dominate trajectory fitting. To deal with this problem, we design an adaptive spatial downsampling approach. Specifically, we first voxelize each trajectory, and then retain only one point in each voxel. To handle different ranges of trajectories, we dynamically adjust the voxel size based on the range of the trajectory. After downsampling, we use the remaining points to fit the trajectory.

For prismatic motion, we use Principal Component Analysis (PCA) for line fitting, combined with the RANSAC algorithm to improve robustness. For revolute motion, we first fit the best plane, then fit a circle on that plane. We use Singular Value Decomposition (SVD) to find the normal vector and verify whether the trajectory conforms to rigid rotation. If the line/circle fitting error of a trajectory is less than pre-defined thresholds ϵ_l/ϵ_c, it is considered a valid track; otherwise, it is treated as a noise track. For a valid track, we prioritize models with smaller fitting errors. The above process also provides the direction of prismatic trajectories and the direction and origin of revolute trajectories.

Motion Clustering. Given valid trajectories with their motion type and motion parameters, we construct feature vectors and then use K-means clustering to group trajectories into different parts. For prismatic motion, the feature vector contains starting position, average position, motion direction, and normalized velocity. For revolute motion, the feature vector contains starting position, average position, axis direction, axis origin, and angular velocity. To improve clustering quality, we adopt an iterative filtering strategy, combining directional angle filtering and Euclidean distance filtering to remove outliers. Finally, we generate articulation information for core parameters initialization of the deformation field $\mathcal{F}$, including the axis direction $\boldsymbol{d}$, axis origin $\boldsymbol{o}$, and part centers $\boldsymbol{p}$.

Deformation Field Initialization. We randomly initialize the remaining parameters of $\mathcal{F}$ and then use the tracking trajectories to optimize them. We design two different losses $\mathcal{L}_{c2o}$ and $\mathcal{L}_{o2o}$ to optimize the deformation field. $\mathcal{L}_{c2o}$ is the canonical-to-observation loss, which provides direct supervision for the deformation from canonical state to the observation state:

$$\hat{\boldsymbol{x}}_i^t = \mathcal{F}(\boldsymbol{x}_i^c, t), \quad \mathcal{L}_{c2o} = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{x}_i^t - \hat{\boldsymbol{x}}_i^t||_2^2, \tag{4}$$

where $\boldsymbol{x}_i^c, \boldsymbol{x}_i^t$ are point positions sampled from trajectory $\{\boldsymbol{x}_i^t\}_{t=1}^T$. $\mathcal{L}_{o2o}$ is the observation-to-observation loss, which enhances temporal consistency between two observation states t_0 and t_1:

$$\hat{\boldsymbol{x}}_i^c = \mathcal{F}^{-1}(\boldsymbol{x}_i^{t_0}, t_0), \quad \hat{\boldsymbol{x}}_i^{t_1} = \mathcal{F}(\hat{\boldsymbol{x}}_i^c, t_1), \quad \mathcal{L}_{o2o} = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{x}_i^{t_1} - \hat{\boldsymbol{x}}_i^{t_1}||_2^2, \tag{5}$$

where $\mathcal{F}^{-1}$ is the inverse deformation field of $\mathcal{F}$ and $\mathcal{F}^{-1}$ shares the same parameters with $\mathcal{F}$. See the *supplementary* for details of the inverse deformation field. We randomly sample pairs of tracking trajectories at time (c, t) for $\mathcal{L}_{c2o}$ and (t_0, t_1) within a 30-frame window for $\mathcal{L}_{o2o}$ to robustly optimize $\mathcal{F}$. The final tracking loss could be calculated by:

$$\mathcal{L}_{\text{track}} = \mathcal{L}_{c2o} + \mathcal{L}_{o2o}. \tag{6}$$

3.3 Optimization

To reconstruct high-quality geometry of objects, we assume the object remains static in the first N frames and initialize canonical Gaussians $\mathcal{G}^c$ with these frames. We train $\mathcal{G}^c$ with the rendering loss $\mathcal{L}_{\text{render}} = (1 - \lambda_{\text{SSIM}})\mathcal{L}_1 + \lambda_{\text{SSIM}}\mathcal{L}_{\text{D-SSIM}} + \mathcal{L}_D$ used in ArtGS (Liu et al., 2025), where $\mathcal{L}_D = \log\left(1 + ||\boldsymbol{D} - \bar{\boldsymbol{D}}||_1\right)$ is a depth loss. After initializing the deformation field $\mathcal{F}$ and canonical Gaussians $\mathcal{G}^c$, we jointly optimize $\mathcal{F}$ and $\mathcal{G}^c$ across all video frames and tracking trajectories with rendering loss $\mathcal{L}_{\text{render}}$ and canonical-to-observation tracking loss $\mathcal{L}_{c2o}$:

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \lambda_{c2o}\mathcal{L}_{c2o}. \tag{7}$$

We provide more implementation and model training details in the *supplementary*.

4 EXPERIMENTS

Datasets We conduct a comprehensive evaluation on three distinct datasets to assess the performance of existing methods on objects with varying articulation complexity. (1) iTACO-S, a dataset proposed by (Peng et al., 2025), which serves as our benchmark for simple articulated objects. It consists of 73 test videos across 11 categories of synthetic objects from the PartNet-Mobility dataset (Xiang et al., 2020), where each object has only a single movable part. (2) VideoArtGS-20, a newly curated dataset that evaluates the performance on more complex scenarios. It contains 20 videos of complex articulated objects of 10 categories from PartNet-Mobility, featuring more challenging kinematics with 2 to 9 movable parts per object. (3) iTACO-L, a large scale version of iTACO-S proposed by (Peng et al., 2025), which contains 711 videos.

Metrics Our evaluation protocol includes metrics for both articulation estimation and mesh reconstruction quality. For articulation estimation, we measure axis direction error (°), axis position error (cm), and joint state error (° for revolute joints, cm for prismatic joints) between the predicted and ground-truth joint parameters. For mesh reconstruction, we assess geometric accuracy using the bi-directional Chamfer Distance (CD). This is computed between the reconstructed mesh and the ground-truth mesh, using 10,000 points uniformly sampled from each surface. We report the CD (in cm) for the whole object (CD-w), the static part (CD-s), and the movable parts (CD-m).

Training Configuration We initialize the deformation field and canonical Gaussians separately, then jointly optimize them. See Appendix A.4 for stage-wise settings.

4.1 RESULTS ON SIMPLE ARTICULATED OBJECTS

Experimental Setup For this benchmark, we use the iTACO-S dataset. We perform a quantitative comparison against three state-of-the-art methods: ArticulateAnything (Le et al., 2025), RSRD (Kerr et al., 2024), and iTACO (Peng et al., 2025). Following the standard evaluation protocol established by iTACO (Peng et al., 2025), all metrics are reported as the mean and standard deviation (mean ± std) across all test videos, and articulation estimation metrics are divided into revolute and prismatic. For a fair and direct comparison, our experimental setup utilizes ground-truth depth and camera poses, and the results for all baseline methods are taken directly from iTACO (Peng et al., 2025). To ensure consistency with our evaluation, we have converted their reported metrics from meters (m) to centimeters (cm) and from radians to degrees. We also retrain and evaluate iTACO on this dataset. Additional results on 711 iTACO-L videos are reported in Appendix C.2.

Results and Analysis The quantitative results, presented in Tab. 1, demonstrate that our method substantially outperforms all baseline approaches across all metrics. The most significant gains are in joint parameter estimation, where VideoArtGS achieves an order-of-magnitude reduction in error compared to the second-best method, iTACO. This dramatic increase in accuracy is primarily attributable to our motion prior guidance, which provides an accurate starting point for optimization that prior methods lack. Our method also achieves a new state of the art in reconstruction quality. The exceptional improvements on both movable parts and the static part validate the effectiveness of VideoArtGS. As illustrated in Fig. 2, our method consistently produces high-fidelity mesh reconstructions with clean part boundaries and precise articulation. This demonstrates the robustness of our approach across the diverse object categories. The trend also holds on the 711-video iTACO-L benchmark: VideoArtGS achieves lower errors on all reported metrics, reducing revolute-axis error from $16.62°$ for iTACO to $1.98°$ and prismatic-axis error from $15.47°$ to $1.27°$. Appendix C.2 provides the complete comparison.

4.2 RESULTS ON COMPLEX ARTICULATED OBJECTS

Experimental Setup We evaluate the performance on our newly curated VideoArtGS-20 dataset, which contains complex, multi-part objects. We compare our method against current state-of-the-art methods ArticulateAnything (Le et al., 2025) and iTACO (Peng et al., 2025). As RSRD (Kerr et al., 2024) failed to correctly segment parts, we don't use it as a baseline. All metrics are averaged across all parts and reported as mean ± std over all objects. A critical limitation of prior work is that iTACO (Peng et al., 2025) is designed only for a single movable part. To establish a baseline, we

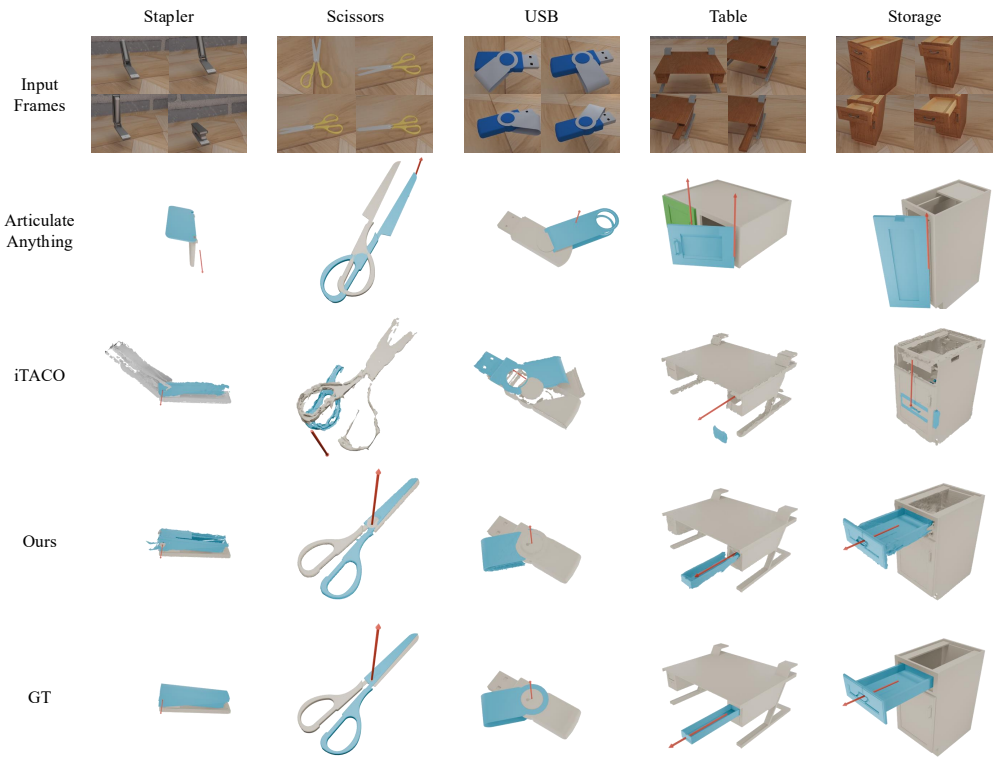


Figure 2: **Qualitative results on iTACO-S dataset.** We present reconstruction comparisons between baselines and our model on the iTACO-S dataset.

Table 1: **Quantitative evaluation on iTACO-S dataset.** Metrics are reported as mean ± std over all test videos. Lower (↓) is better on all metrics, and the **best results** are highlighted in bold. † means the results are taken from iTACO (Peng et al., 2025).

Method	Revolute Joint Estimation			Prismatic Joint Estimation		Reconstruction		
	Axis (°)	Position (cm)	State (°)	Axis (°)	State (cm)	CD-w (cm)	CD-m (cm)	CD-s (cm)
ArticulateAnything† (Le et al., 2025)	46.98±45.27	81.00±40.00	N/A	52.71±44.69	N/A	11.00±22.00	59.00±73.00	7.00±18.00
RSRD† (Kerr et al., 2024)	67.06±29.22	203.00±748.00	59.02±34.38	69.91±24.07	70.00±48.00	339.00±2147.00	82.00±117.00	14.00±41.00
iTACO † (Peng et al., 2025)	18.34±32.09	13.00±25.00	14.32±26.35	13.75±18.91	8.00±22.00	1.00±1.00	13.00±26.00	6.00±19.00
iTACO (Peng et al., 2025)	13.83±28.15	11.55±22.39	10.25±21.27	14.37±19.08	3.44±6.25	3.45±16.46	12.21±24.44	5.39±17.09
Ours	**0.32±0.44**	**0.42±0.75**	**1.15±2.29**	**0.35±0.45**	**1.03±2.46**	**0.29±0.24**	**1.11±2.11**	**0.40±0.32**

extend it to multi-part objects: we manually isolate video segments where only a single part is in motion and then merge the moving map to extract multiple part meshes.

Results and Analysis On the complex, multi-part VideoArtGS-20 dataset, our method's advantages become even more pronounced. Compared to iTACO-S, VideoArtGS-20 has larger camera motion and includes more moving parts, posing a greater challenge to existing baselines. As shown in Fig. 3, iTACO struggles to accurately segment moving parts, while ArticulateAnything often retrieves incorrect parts. As demonstrated in Tab. 2, VideoArtGS achieves state-of-the-art performance, drastically outperforming baselines in this complex multi-part setting. Mean joint-axis error decreases from $43.65°$ for ArticulateAnything and $48.88°$ for the multi-part iTACO baseline to $0.34°$. CD-m decreases from 17.66 and 30.63, respectively, to 0.26. These results confirm that VideoArtGS's advantages generalize from simple to complex scenarios, providing a robust and scalable solution for reconstructing articulated objects from monocular video.

Table 2: **Quantitative evaluation on VideoArtGS-20 dataset.** Metrics are reported as mean ± std over all test videos. Lower (↓) is better on all metrics, and the **best results** are highlighted in bold.

Method	Axis (°)	Position(cm)	CD-w (cm)	CD-m (cm)	CD-s (cm)
ArticulateAnything (Le et al., 2025)	43.65 ± 44.72	15.66 ± 36.20	16.10 ± 37.34	17.66 ± 36.74	16.04 ± 37.36
iTACO (Peng et al., 2025)	48.88 ± 24.18	37.04 ± 31.82	5.07 ± 21.78	30.63 ± 25.64	10.22 ± 22.23
Ours	**0.34±0.80**	**0.10±0.10**	**0.09±0.09**	**0.26±0.61**	**0.24±0.58**

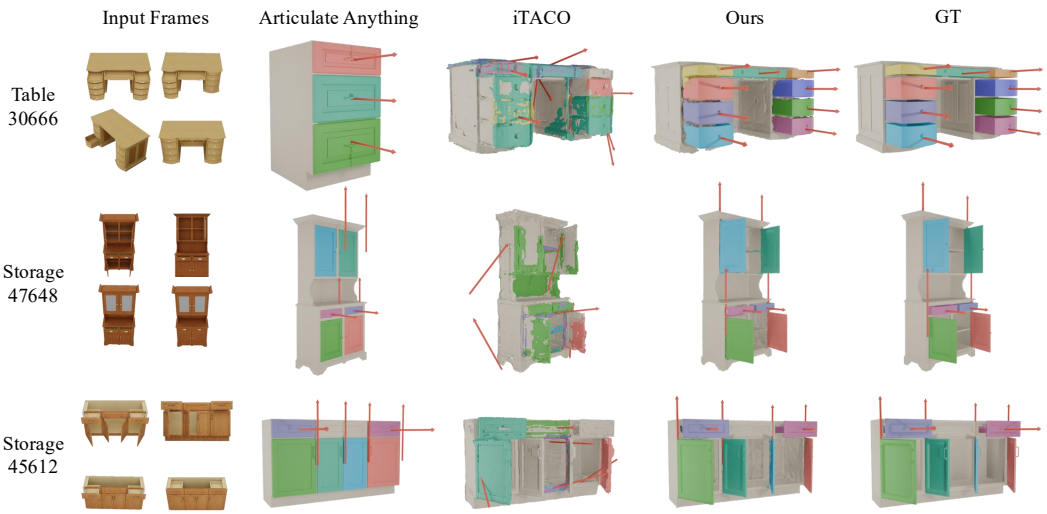


Figure 3: **Qualitative results on VideoArtGS-20 dataset.** We present reconstruction comparisons between baselines and our model on the VideoArtGS-20 dataset.

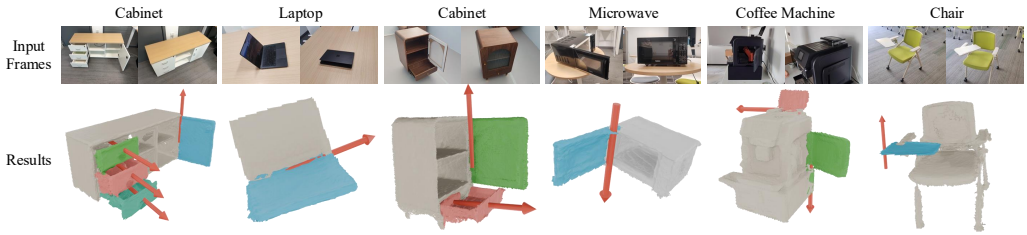


Figure 4: **Qualitative results on real-world data.** We present results of our model on real-world data, including both simple two-part and complex multi-part objects.

4.3 Results on Real-World Data

Experimental Setup We also validate the effectiveness of VideoArtGS on real-world data. We capture monocular videos using a mobile phone camera without LiDAR. We use articulated objects of different categories with different numbers of joints to verify the generalization ability of our method. The input to our model is solely the monocular RGB video.

Results and Analysis As shown in Fig. 4, our VideoArtGS successfully reconstructs a diverse set of articulated objects from self-captured, real-world monocular videos. VideoArtGS effectively decouples the object's geometry from motion, fulfilling the promise of creating interactable digital twins. A qualitative comparison with iTACO on its real-world data is provided in Appendix C.3.

Table 3: **Ablation studies on VideoArtGS-20 dataset.** Lower (↓) is better on all metrics, and the **best results** are highlighted in bold.

Method	Axis (°)	Position (cm)	CD-w (cm)	CD-m (cm)	CD-s (cm)
Ours	**0.34±0.80**	**0.10±0.10**	**0.09±0.09**	0.26±0.61	**0.24±0.58**
w/o motion prior	55.28±15.49	23.74±17.49	10.18±29.94	87.77±17.02	14.37±29.69
w/o center init	20.64±21.64	22.42±25.03	10.33±29.90	83.32±14.06	14.07±29.80
w/o deform init	3.96±3.73	2.45±3.07	0.11±0.12	1.50±2.71	0.72±2.05
w/o axis init	0.60±1.26	0.86±2.58	**0.09±0.09**	**0.25±0.56**	0.27±0.74
w/o hybrid	1.21±1.77	2.51±10.47	0.15±0.29	10.35±23.84	0.50±1.08
w/o $\mathcal{L}_{\text{o2o}}$	0.68±1.55	0.57±1.85	0.11±0.12	0.58±1.03	0.27±0.65
w/o $\mathcal{L}_{\text{c2o}}$	0.40±0.79	0.13±0.11	0.09±0.10	0.35±0.86	0.26±0.70

4.4 Ablation Studies

To validate each component's effectiveness, we systematically remove components on the VideoArtGS-20 dataset and analyze their impact. Results are reported as mean ± std in Tab. 3.

Motion Prior Guidance. Removing motion prior guidance, including center, joint-axis, and deformation-field initialization, results in a catastrophic failure of the model. This large degradation identifies motion-derived initialization as a major contributor to accurate multi-part reconstruction.

Initialization of components. Removing the part centers initialization (w/o center init) leads to a complete failure, highlighting the importance of spatial anchors. Removing the deformation field initialization (w/o deform init) mainly degrades movable part reconstruction. Removing the axis initialization (w/o axis init) has a smaller impact, suggesting that well-initialized part centers and correspondences help recover the joint axes during optimization.

Hybrid Center-Grid Assignment. Replacing the hybrid center-grid assignment module with the center-based assignment module (w/o hybrid) leads to performance drops, especially for movable parts. This result highlights that our hybrid assignment is essential for articulation dynamics.

Tracking Losses. Disabling the observation-to-observation loss (w/o $\mathcal{L}_{o2o}$) degrades performance more than the canonical-to-observation loss (w/o $\mathcal{L}_{c2o}$). This indicates that enforcing temporal consistency on the observation space is more critical for achieving precise joint estimation.

Additional Analyses. Under the tracking protocol in Appendix C.4, trajectories generated by the optimized deformation field reduce mean EPE from 0.16 to 0.09 m on VideoArtGS-20. Thus, the reconstructed motion improves on the supplied tracks rather than simply reproducing them. Appendix C.5 analyzes the trade-off between rejecting noisy tracks and retaining motion evidence. The initial static-segment requirement, upstream dependencies, and observed failure cases are discussed in Appendices B and C.7.

5 Conclusion

We introduce VideoArtGS, a motion-prior-guided framework for articulated object reconstruction from monocular videos. Its central design is to turn noisy 3D tracks into part and articulation initialization, then jointly refine geometry and motion with a hybrid center-grid assignment model. Experiments on iTACO-S, iTACO-L, and VideoArtGS-20 demonstrate substantial improvements in joint estimation and part-level reconstruction. The ablations highlight the importance of coherent part-center initialization and assignment design, while real-world examples demonstrate reconstruction from monocular RGB recordings. These results establish our approach as an effective basis for accurate articulated reconstruction.

Reproducibility Statement We re-executed all experiments before submission to ensure reproducibility and consistency of our results. Detailed implementation and training procedures are provided in the appendix. Upon paper acceptance, we will release all code, data, and model weights publicly. The planned code release will include training scripts, evaluation protocols, and detailed documentation to facilitate easy reproduction of all experimental results.

AI USE STATEMENT

Our pipeline uses GPT-4o to obtain the number of movable parts and joint types, as described in the method and appendix. We additionally evaluated GPT-5.6 Sol and GPT-6 Sol for movable-part count and joint-type prediction in the Appendix. Codex assisted with migrating the LaTeX source to the ICLR 2027 template, resolving compilation issues and polishing writing.

REFERENCES

Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. *arXiv preprint arXiv:2404.03613*, 2024.

Martin Büchner, Adrian Röfer, Tim Engelbracht, Tim Welschehold, Zuria Bauer, Hermann Blum, Marc Pollefeys, and Abhinav Valada. Articulated 3d scene graphs for open-world mobile manipulation. *arXiv preprint arXiv:2602.16356*, 2026.

Honghua Chen, Yushi Lan, Yongwei Chen, and Xingang Pan. Artilatent: Realistic articulated 3d object generation via structured latents. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, 2025.

Ruihang Chu, Zhengzhe Liu, Xiaoqing Ye, Xiao Tan, Xiaojuan Qi, Chi-Wing Fu, and Jiaya Jia. Command-driven articulated object understanding and manipulation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Hang Dai, Hongwei Fan, Han Zhang, Duojin Wu, Jiyao Zhang, and Hao Dong. FreeArtGS: Articulated gaussian splatting under free-moving scenario. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Alexandros Delitzas, Chenyangguang Zhang, Alexey Gavryushin, Tommaso Di Mario, Boyang Sun, Rishabh Dabral, Leonidas Guibas, Christian Theobalt, Marc Pollefeys, Francis Engelmann, and Daniel Barath. FunREC: Reconstructing functional 3D scenes from egocentric interaction videos. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jianning Deng, Kartic Subr, and Hakan Bilen. Articulate your nerf: Unsupervised articulated object modeling via conditional view synthesis. *arXiv preprint arXiv:2406.16623*, 2024.

Jianning Deng, Kartic Subr, and Hakan Bilen. Paoli: Pose-free articulated object learning from sparse-view images. *arXiv preprint arXiv:2509.04276*, 2025.

Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, et al. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning. *arXiv preprint arXiv:2504.18904*, 2025.

Pradyumn Goyal, Dmitry Petrov, Sheldon Andrews, Yizhak Ben-Shabat, Hsueh-Ti Derek Liu, and Evangelos Kalogerakis. Geopard: Geometric pretraining for articulation prediction in 3d shapes. *arXiv preprint arXiv:2504.02747*, 2025.

Qiuyi Gu, Yuze Sheng, Jincheng Yu, Jiahao Tang, Xiaolong Shan, Zhaoyang Shen, Tinghao Yi, Xiaodan Liang, Xinlei Chen, and Yu Wang. Artisg: Functional 3d scene graph construction via human-demonstrated articulated objects manipulation. *arXiv preprint arXiv:2512.24845*, 2025.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Zhiyang Guo, Wengang Zhou, Li Li, Min Wang, and Houqiang Li. Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. *arXiv preprint arXiv:2403.11447*, 2024.

Yumeng He, Ying Jiang, Jiayin Lu, Yin Yang, and Chenfanfu Jiang. Spark: Sim-ready part-level articulated reconstruction with vlm knowledge. *arXiv preprint arXiv:2512.01629*, 2025.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2023.

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.

Jiangyong Huang, Baoxiong Jia, Yan Wang, Ziyu Zhu, Xiongkun Linghu, Qing Li, Song-Chun Zhu, and Siyuan Huang. Unveiling the mist over 3d vision-language understanding: Object-centric evaluation with chain-of-analysis. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Jiangyong Huang, Xiaojian Ma, Xiongkun Linghu, Yue Fan, Junchao He, Wenxin Tan, Qing Li, Song-Chun Zhu, Yixin Chen, Baoxiong Jia, and Siyuan Huang. Leo-vl: Efficient scene representation for scalable 3d vision-language learning. *arXiv preprint arXiv:2506.09935*, 2025b.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zhao Huang, Boyang Sun, Alexandros Delitzas, Jiaqi Chen, and Marc Pollefeys. React3d: Recovering articulations for interactive physical 3d scenes. *arXiv preprint arXiv:2510.11340*, 2025c.

Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2021.

Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023.

Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: a physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

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

Bu Jin, Weize Li, Songen Gu, Yupeng Zheng, Yuhang Zheng, Zhengyi Zhou, and Yao Yao. Uniart: Unified 3d representation for generating 3d articulated objects with open-set articulation. *arXiv preprint arXiv:2511.21887*, 2025.

HyunJun Jung, Nikolas Brasch, Jifei Song, Eduardo Perez-Pellitero, Yiren Zhou, Zhihao Li, Nassir Navab, and Benjamin Busam. Deformable 3d gaussian splatting for animatable human avatars. *arXiv preprint arXiv:2312.15059*, 2023.

Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. An efficient 3d gaussian representation for monocular/multi-view dynamic scenes. *arXiv preprint arXiv:2311.12897*, 2023.

Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Unsupervised pose-aware part decomposition for 3d articulated objects. *arXiv preprint arXiv:2110.04411*, 2021.

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.

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 (CoRL)*, 2024.

Seungyeon Kim, Junsu Ha, Young Hun Kim, Yonghyeon Lee, and Frank C Park. Screwsplat: An end-to-end method for articulated object recognition. *arXiv preprint arXiv:2508.02146*, 2025.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023.

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 *Proceedings of International Conference on Learning Representations (ICLR)*, 2025.

Daeun Lee, Jaeah Lee, Woosung Kim, Haebeom Jung, and Jaesik Park. Articulated object reconstruction from rest-state observation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2026a.

Gahye Lee, Gyoonseo Kim, Wonjong Jang, Jooeun Son, and Seungyong Lee. StructureGS: Structure-aware gaussian splatting for articulated object reconstruction. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2026b.

Inseo Lee, Yoonji Kim, Eugene Sohn, Jiwoong Lee, Jungmin You, Joonseok Lee, and Jin-Hwa Kim. ArtSplat: Feed-forward articulated 3D gaussian splatting from sparse multi-state uncalibrated views. *arXiv preprint arXiv:2605.24304*, 2026c.

Jiahui Lei, Congyue Deng, William B Shen, Leonidas J Guibas, and Kostas Daniilidis. Nap: Neural 3d articulated object prior. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic gaussian fusion from casual videos via 4D motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024b.

Stanley Lewis, Jana Pavlasek, and Odest Chadwicke Jenkins. Narf22: Neural articulated radiance fields for configuration-aware rendering. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2022.

Stanley Lewis, Vishal Chandra, Tom Gao, and Odest Chadwicke Jenkins. Splatart: Articulated gaussian splatting with estimated object structure. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

Haitian Li, Haozhe Xie, Junxiang Xu, Beichen Wen, Fangzhou Hong, and Ziwei Liu. MonoArt: Progressive structural reasoning for monocular articulated 3D reconstruction. *arXiv preprint arXiv:2603.19231*, 2026a.

Ruining Li, Yuxin Yao, Chuanxia Zheng, Christian Rupprecht, Joan Lasenby, Shangzhe Wu, and Andrea Vedaldi. Particulate: Feed-forward 3d object articulation. *arXiv preprint arXiv:2512.11798*, 2025a.

Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Xuelu Li, Zhaonan Wang, Xiaogang Wang, Lei Wu, Manyi Li, and Changhe Tu. ArtPro: Self-supervised articulated object reconstruction with adaptive integration of mobility proposals. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *CVPR*, 2025b.

Zizhang Li, Cheng Zhang, Zhengqin Li, Henry Howard-Jenkins, Zhaoyang Lv, Chen Geng, Jiajun Wu, Richard Newcombe, Jakob Engel, and Zhao Dong. Art: Articulated reconstruction transformer. *arXiv preprint arXiv:2512.14671*, 2025c.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025a.

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. *arXiv preprint arXiv:2506.03594*, 2025b.

Xiongkun Linghu, Jiangyong Huang, Ziyu Zhu, Baoxiong Jia, and Siyuan Huang. Scenecot: Eliciting grounded chain-of-thought reasoning in 3d scenes. *arXiv preprint arXiv:2510.16714*, 2025.

Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-level reconstruction and motion analysis for articulated objects. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023a.

Jiayi Liu, Manolis Savva, and Ali Mahdavi-Amiri. Survey on modeling of articulated objects. *arXiv preprint arXiv:2403.14937*, 2024a.

Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. Cage: Controllable articulation generation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Jiayi Liu, Denys Iliash, Angel X Chang, Manolis Savva, and Ali Mahdavi-Amiri. Singapo: Single image controlled generation of articulated parts in objects. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2025.

Liu Liu, Han Xue, Wenqiang Xu, Haoyuan Fu, and Cewu Lu. Toward real-world category-level articulation pose estimation. *Proceedings of Transactions on Image Processing (TIP)*, 31:1072–1083, 2022.

Qingming LIU, Yuan Liu, Jiepeng Wang, Xianqiang Lyu, Peng Wang, Wenping Wang, and Junhui Hou. MoDGS: Dynamic gaussian splatting from casually-captured monocular videos with depth priors. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2prShxdLkX.

Qingming Liu, Xinyue Yao, Shuyuan Zhang, Yueci Deng, Guiliang Liu, Zhen Liu, and Kui Jia. Pact: Part-decomposed single-view articulated object generation. *arXiv preprint arXiv:2602.14965*, 2026.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9298–9309, 2023b.

Shaowei Liu, Saurabh Gupta, and Shenlong Wang. Building rearticulable models for arbitrary 3d objects from 4d point clouds. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023c.

Xueyi Liu, Ji Zhang, Ruizhen Hu, Haibin Huang, He Wang, and Li Yi. Self-supervised category-level articulated object pose estimation with part-level se (3) equivariance. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023d.

Yu Liu, Baoxiong Jia, Yixin Chen, and Siyuan Huang. Slotlifter: Slot-guided feature lifting for learning object-centric radiance fields. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2024c.

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 International Conference on Learning Representations (ICLR)*, 2025.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9263–9274, 2025a.

Ruijie Lu, Yixin Chen, Junfeng Ni, Baoxiong Jia, Yu Liu, Diwen Wan, Gang Zeng, and Siyuan Huang. Movis: Enhancing multi-object novel view synthesis for indoor scenes. *arXiv preprint arXiv:2412.11457*, 2024a.

Ruijie Lu, Yixin Chen, Yu Liu, Jiaxiang Tang, Junfeng Ni, Diwen Wan, Gang Zeng, and Siyuan Huang. Taco: Taming diffusion for in-the-wild video amodal completion. *arXiv preprint arXiv:2503.12049*, 2025b.

Ruijie Lu, Yu Liu, Jiaxiang Tang, Junfeng Ni, Yuxiang Wang, Diwen Wan, Gang Zeng, Yixin Chen, and Siyuan Huang. Dreamart: Generating interactable articulated objects from a single image. *arXiv preprint arXiv:2507.05763*, 2025c.

Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *Proceedings of International Conference on 3D Vision (3DV)*, 2024.

Rundong Luo, Haoran Geng, Congyue Deng, Puhao Li, Zan Wang, Baoxiong Jia, Leonidas Guibas, and Siyuan Huang. Physpart: Physically plausible part completion for interactable objects. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2025.

Liqian Ma, Jiaojiao Meng, Shuntao Liu, Weihang Chen, Jing Xu, and Rui Chen. Sim2real 2: Actively building explicit physics model for precise articulated object manipulation. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2023.

Roberto Martín-Martín, Sebastian Höfer, and Oliver Brock. An integrated approach to visual perception of articulated objects. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2016.

Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

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 International Conference on Computer Vision (ICCV)*, 2021.

Junfeng Ni, Yixin Chen, Zhifei Yang, Yu Liu, Ruijie Lu, Song-Chun Zhu, and Siyuan Huang. G4splat: Geometry-guided gaussian splatting with generative prior. *arXiv preprint arXiv:2510.12099*, 2025a.

Junfeng Ni, Yu Liu, Ruijie Lu, Zirui Zhou, Song-Chun Zhu, Yixin Chen, and Siyuan Huang. Decompositional neural scene reconstruction with generative diffusion prior. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

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.

Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch it move: Unsupervised discovery of 3d joints for re-posing of articulated objects. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Weikun Peng, Jun Lv, Cewu Lu, and Manolis Savva. iTACO: Interactable Digital Twins of Articulated Objects from Casually Captured RGBD Videos. In *3DV 2026*, 2025.

Sudeep Pillai, Matthew R Walter, and Seth Teller. Learning articulated motions from visual demonstration. *arXiv preprint arXiv:1502.01659*, 2015.

Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Hongyu Shen, Junfeng Ni, Yixin Chen, Weishuo Li, Mingtao Pei, and Siyuan Huang. Trace3d: Consistent segmentation lifting via gaussian instance tracing. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2025a.

Licheng Shen, Saining Zhang, Honghan Li, Peilin Yang, Zihao Huang, Zongzheng Zhang, and Hao Zhao. Gaussianart: Unified modeling of geometry and motion for articulated objects. *arXiv preprint arXiv:2508.14891*, 2025b.

Chaoyue Song, Jiacheng Wei, Chuan Sheng Foo, Guosheng Lin, and Fayao Liu. Reacto: Reconstructing articulated objects from a single video. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41, 2011.

Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel Xuan Chang. Opdmulti: Openable part detection for multiple objects. *arXiv preprint arXiv:2303.14087*, 2023.

Archana Swaminathan, Anubhav Gupta, Kamal Gupta, Shishira R Maiya, Vatsal Agarwal, and Abhinav Shrivastava. Leia: Latent view-invariant embeddings for implicit 3d articulation. *arXiv preprint arXiv:2409.06703*, 2024.

Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021.

Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

Wei-Cheng Tseng, Hung-Ju Liao, Lin Yen-Chen, and Min Sun. Cla-nerf: Category-level articulated neural radiance field. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2022.

Haowen Wang, Xiaoping Yuan, Zhao Jin, Zhen Zhao, Zhengping Che, Yousong Xue, Jin Tian, Yakun Huang, and Jian Tang. Self-supervised multi-part articulated objects modeling via deformable gaussian splatting and progressive primitive segmentation. *arXiv preprint arXiv:2506.09663*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025b.

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of Motion: 4D reconstruction from a single video. In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 9660–9672, 2025c.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025d.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Xuyang Wang, Zhenyu Li, Jian Ding, Habib Slim, Peter Wonka, Hongdong Li, and Mohamed Elhoseiny. Artic-O: End-to-end articulated object reconstruction via latent geometry learning. In *Proceedings of the SIGGRAPH Asia 2026 Conference Papers*, 2026a.

Yan Wang, Baoxiong Jia, Ziyu Zhu, and Siyuan Huang. Masked point-entity contrast for open-vocabulary 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14125–14136, 2025e.

Zikai Wang, Zhilu Zhang, Yiqing Wang, Hui Li, and Wangmeng Zuo. ArtHOI: Taming foundation models for monocular 4D reconstruction of hand-articulated-object interactions. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Fangyin Wei, Rohan Chabra, 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J Guibas. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

Yijia Weng, Bowen Wen, Jonathan Tremblay, Valts Blukis, Dieter Fox, Leonidas Guibas, and Stan Birchfield. Neural implicit representation for building digital twins of unknown articulated objects. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Abdelrhman Werby, Martin Büchner, Adrian Röfer, Chenguang Huang, Wolfram Burgard, and Abhinav Valada. Articulated object estimation in the wild. In *Conference on Robot Learning (CoRL)*, 2025.

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. *arXiv preprint arXiv:2503.06677*, 2025a.

Di Wu, Liu Liu, Xueyu Yuan, Qiaojun Yu, Wenxiao Chen, Ruilong Yan, Yiming Tang, and Liangtu Song. Spags: Sparse-view articulated object reconstruction from single state via planar gaussian splatting. *arXiv preprint arXiv:2511.17092*, 2025b.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, Yue Qian, Xiaohang Zhan, and Yueqi Duan. 4d-fly: Fast 4d reconstruction from a single monocular video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025c.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Iurii Makarov, Bingyi Kang, Xin Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *ICCV*, 2025.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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.

Gengshan Yang, Chaoyang Wang, N Dinesh Reddy, and Deva Ramanan. Reconstructing animatable categories from videos. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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.

Huangyue Yu, Baoxiong Jia, Yixin Chen, Yandan Yang, Puhao Li, Rongpeng Su, Jiaxin Li, Qing Li, Wei Liang, Song-Chun Zhu, et al. Metascenes: Towards automated replica creation for real-world 3d scans. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1667–1679, 2025a.

Tianjiao Yu, Vedant Shah, Muntasir Wahed, Ying Shen, Kiet A Nguyen, and Ismini Lourentzou. Part^{2}gs: Part-aware modeling of articulated objects using 3d gaussian splatting. *arXiv preprint arXiv:2506.17212*, 2025b.

Sylvia Yuan, Ruoxi Shi, Xinyue Wei, Xiaoshuai Zhang, Hao Su, and Minghua Liu. Larm: A large articulated object reconstruction model. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, 2025.

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

Can Zhang and Gim Hee Lee. Iaao: Interactive affordance learning for articulated objects in 3d environments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12132–12142, 2025.

Ge Zhang, Or Litany, Srinath Sridhar, and Leonidas Guibas. Strobenet: Category-level multiview reconstruction of articulated objects. *arXiv preprint arXiv:2105.08016*, 2021.

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.

Tianshan Zhang, Zeyu Zhang, and Hao Tang. Dragmesh: Interactive 3d generation made easy. *arXiv preprint arXiv:2512.06424*, 2025b.

Zhoutong Zhang, Forrester Cole, Zhengqi Li, Noah Snavely, Michael Rubinstein, and William T. Freeman. Structure and motion from casual videos. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022.

Mandi Zhao, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. *arXiv preprint arXiv:2406.08474*, 2024a.

Zihang Zhao, Yuyang Li, Wanlin Li, Zhenghao Qi, Lecheng Ruan, Yixin Zhu, and Kaspar Althoefer. Tac-man: Tactile-informed prior-free manipulation of articulated objects. *Transactions on Robotics (T-RO)*, 2024b.

Zihang Zhao, Zhenghao Qi, Yuyang Li, Leiyao Cui, Zhi Han, Lecheng Ruan, and Yixin Zhu. Tacman-turbo: Proactive tactile control for robust and efficient articulated object manipulation. *arXiv preprint arXiv:2508.02204*, 2025.

Hongyi Zhou, Xiaogang Wang, Yulan Guo, and Kai Xu. Monomobility: Zero-shot 3d mobility analysis from monocular videos. *arXiv preprint arXiv:2505.11868*, 2025.

VideoArtGS: Building Digital Twins of Articulated Objects from Monocular Video
Supplementary Material

A Implementation and Training Details

A.1 VideoArtGS-20 Dataset

We introduce and evaluate our method on VideoArtGS-20, a newly curated dataset featuring 10 object categories: Faucet, Door, Refrigerator, Table, Storage Furniture, Bucket, Eyeglasses, Oven, Window, and Printer. For each object, we render a monocular video with 150 static frames in different viewpoints and 60 dynamic frames for each movable part. The dataset provides a challenging benchmark with objects containing up to 10 parts and 9 movable joints. Further details and visualizations are available in Tab. A.1 and Fig. A.1.

A.2 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) represents a 3D scene using a collection of 3D Gaussians (Kerbl et al., 2023). Each Gaussian G_i is parameterized by its center $\boldsymbol{\mu}_i \in \mathbb{R}^3$, covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$, opacity $\sigma_i \in [0,1]$, and spherical harmonics coefficients $\boldsymbol{h}_i$ for view-dependent color. The opacity of a 3D Gaussian at spatial point $\boldsymbol{x}$ is computed as:

$$\alpha_i(\boldsymbol{x}) = \sigma_i \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right), \quad \text{where} \quad \boldsymbol{\Sigma}_i = \boldsymbol{R}_i\boldsymbol{S}_i\boldsymbol{S}_i^\top\boldsymbol{R}_i^\top. \tag{A.1}$$

To ensure $\boldsymbol{\Sigma}_i$ remains positive semi-definite, it is decomposed into a rotation matrix $\boldsymbol{R}_i$ (parameterized by quaternion $\boldsymbol{r}_i$) and a scaling diagonal matrix $\boldsymbol{S}_i$ (parameterized by scale vector $\boldsymbol{s}_i$). To render an image, 3D Gaussians are projected onto the 2D image plane and aggregated using α-blending:

$$\boldsymbol{I} = \sum_{i=1}^{N} T_i \alpha_i^{\text{2D}} \mathcal{SH}(\boldsymbol{h}_i, \boldsymbol{v}_i), \quad \text{where} \quad T_i = \prod_{j=1}^{i-1}(1-\alpha_j^{\text{2D}}). \tag{A.2}$$

Here, α_i^{2D} is the 2D version of Eq. (A.1), $\mathcal{SH}(\cdot)$ calculates spherical harmonics for view direction $\boldsymbol{v}_i$. Given multi-view images $\{\bar{\boldsymbol{I}}_i\}_{i=1}^N$, 3DGS optimizes the parameters using L1 loss and D-SSIM loss (Kerbl et al., 2023) with a loss weight λ_{SSIM}:

$$\mathcal{L}_I = (1-\lambda_{\text{SSIM}})\mathcal{L}_1 + \lambda_{\text{SSIM}}\mathcal{L}_{\text{D-SSIM}}, \tag{A.3}$$

A.3 Articulation Modeling

Articulation Modeling Building upon the part assignments, we model articulation through learnable joint parameters, including axis direction $\boldsymbol{d}$, axis origin $\boldsymbol{o}$, and time-variant joint state θ^t. To learn a smooth trajectory of joint states, we model it with Fourier embedding $E(\cdot)$ followed by a learnable MLP: $\theta^t = \text{MLP}(E(t))$. We represent the rigid transformation as dual-quaternion $\boldsymbol{q}^t = (\boldsymbol{q}_r^t, \boldsymbol{q}_d^t)$ for smooth skinning, where $\boldsymbol{q}_r^t, \boldsymbol{q}_d^t$ represent the rotation and translation components respectively. The dual-quaternion of each joint could be calculated as:

$$\begin{aligned}
&\text{prismatic}: \boldsymbol{q}_r^t = (1,0,0,0), \quad \bar{\boldsymbol{o}}^t = (0, \theta^t \cdot \boldsymbol{d}), \quad \boldsymbol{q}_d^t = 0.5 \cdot \bar{\boldsymbol{o}}^t \otimes \boldsymbol{q}_r^t, \\
&\text{revolute}: \boldsymbol{q}_r^t = (\cos\frac{\theta^t}{2}, \sin\frac{\theta^t}{2} \cdot \boldsymbol{d}), \quad \bar{\boldsymbol{o}}^t = (0, \boldsymbol{o}), \quad \boldsymbol{q}_d^t = 0.5 \cdot (\bar{\boldsymbol{o}}^t \otimes \boldsymbol{q}_r^t - \boldsymbol{q}_r^t \otimes \bar{\boldsymbol{o}}^t).
\end{aligned} \tag{A.4}$$

Then we calculate the per-gaussian dual-quaternion $\boldsymbol{q}_i^t$ with part assignment probabilities $\boldsymbol{m}_i$ by:

$$\boldsymbol{q}_i^t = \sum_{k=1}^{K} m_{ik} \cdot \boldsymbol{q}_k^t = (\sum_{k=1}^{K} m_{ik} \cdot \boldsymbol{q}_{k,r}^t, \sum_{k=1}^{K} m_{ik} \cdot \boldsymbol{q}_{k,d}^t). \tag{A.5}$$

where $\boldsymbol{q}_k^t$ is the dual-quaternion of k-th part. The position and rotation of Gaussian G_i^t are obtained by:

$$\boldsymbol{\mu}_i^t = \boldsymbol{R}_i^t \cdot \boldsymbol{\mu}_i^c + \boldsymbol{t}_i^t, \quad \boldsymbol{r}_i^t = \boldsymbol{q}_{i,r}^t \otimes \boldsymbol{r}_i^c, \tag{A.6}$$

where $\boldsymbol{R}_i^t$ and $\boldsymbol{t}_i^t$ is rotation matrix and translation vector derived from $\boldsymbol{q}_i^t$, and $\otimes$ denotes quaternion multiplication operation. We derive the dual-quaternion formulation below.

Table A.1: Dataset configuration.

Object ID	Category	#Part	#Joint	#Revolute	#Prismatic
168	Faucet	3	2	2	0
1280	Faucet	3	2	2	0
8961	Door	3	2	2	0
9016	Door	3	2	2	0
10489	Refrigerator	3	2	2	0
10655	Refrigerator	3	2	2	0
25493	Table	4	3	0	3
30666	Table	10	9	0	9
31249	Table	5	4	2	2
45194	Storage Furniture	5	4	2	2
45503	Storage Furniture	4	3	3	0
45612	Storage Furniture	7	6	4	2
47648	Storage Furniture	7	6	4	2
100481	Bucket	3	2	2	0
101284	Eyeglasses	3	2	2	0
101287	Eyeglasses	3	2	2	0
101808	Oven	3	2	2	0
101908	Oven	4	3	3	0
103015	Window	4	3	3	0
103811	Printer	7	6	0	6
Average	—	4.35	3.35	2.05	1.3

Dual Quaternions for SE(3) Transformation A general rigid SE(3) transformation in 3D space consists of a rotation followed by a translation. A dual quaternion represents this combined operation within a single algebraic entity. Let the rotation be represented by a unit quaternion $\boldsymbol{q}_r$ and the translation by a vector $\boldsymbol{t}$. A point $\boldsymbol{p}$ in space, represented as a pure quaternion $\bar{\boldsymbol{p}} = (0, \boldsymbol{p})$, is transformed to a new point $\boldsymbol{p}'$ by first applying the rotation and then the translation: $\boldsymbol{p}' = \boldsymbol{q}_r \otimes \boldsymbol{p} \otimes \boldsymbol{q}_r^* + \boldsymbol{t}$, where $\boldsymbol{q}_r^*$ is the conjugate of $\boldsymbol{q}_r$ and $\otimes$ denotes the quaternion multiplication operation.

A dual quaternion $\boldsymbol{q}$ is defined as $\boldsymbol{q} = \boldsymbol{q}_r + \varepsilon \boldsymbol{q}_d$, where $\boldsymbol{q}_r$ is the real part, $\boldsymbol{q}_d$ is the dual part, and ε is the dual unit with the property $\varepsilon^2 = 0$. Given the rotation quaternion $\boldsymbol{q}_r$ and translation $\boldsymbol{t}$, the dual part $\boldsymbol{q}_d$ could be calculated by: $\boldsymbol{q}_d = \frac{1}{2}(0, \boldsymbol{t}) \otimes \boldsymbol{q}_r$.

Dual Quaternions for Articulated Transformation We apply the above principles to derive the specific formulas for prismatic and revolute joints at time t.

Prismatic: A prismatic joint executes a pure translation with no rotation, so that the real part is the unit quaternion $\boldsymbol{q}_r^t = (1, 0, 0, 0)$. Given the axis direction $\boldsymbol{d}$ and joint state θ^t, its translation component is $\boldsymbol{t} = \theta^t \cdot \boldsymbol{d}$. Let $\bar{\boldsymbol{o}}^t = (0, \theta^t \cdot \boldsymbol{d})$, the dual part could be calculated by: $\boldsymbol{q}_d = \frac{1}{2}\bar{\boldsymbol{o}} \otimes \boldsymbol{q}_r^t$.

Revolute: A revolute joint executes a pure rotation, not about the world origin, but about the joint's origin point $\boldsymbol{o}$. Given the axis direction $\boldsymbol{d}$, axis origin $\boldsymbol{o}$ and joint state θ^t this "off-center" rotation is equivalent to a sequence of three operations: (1) translate the system so the pivot point $\boldsymbol{o}$ moves to the origin: $\bar{\boldsymbol{o}}^t = (0, \boldsymbol{o}),\ \boldsymbol{q}_{T_1} = 1 - \frac{\varepsilon}{2}\bar{\boldsymbol{o}}$); (2) perform the rotation around the origin: $\boldsymbol{q}_R = \boldsymbol{q}_r^t = (\cos\frac{\theta^t}{2}, \sin\frac{\theta^t}{2} \cdot \boldsymbol{d})$. (3) translate the system back: $\boldsymbol{q}_{T_2} = 1 + \frac{\varepsilon}{2}\bar{\boldsymbol{o}}$. The total transformation $\boldsymbol{q}^t$ is the product:

$$\boldsymbol{q}^t = \boldsymbol{q}_{T_2}\boldsymbol{q}_R\boldsymbol{q}_{T_1} = (1 + \frac{\varepsilon}{2}\bar{\boldsymbol{o}})\boldsymbol{q}_r^t(1 - \frac{\varepsilon}{2}\bar{\boldsymbol{o}}) = (1 + \frac{\varepsilon}{2}\bar{\boldsymbol{o}})(\boldsymbol{q}_r^t - \frac{\varepsilon}{2}\boldsymbol{q}_r^t\bar{\boldsymbol{o}}) = \boldsymbol{q}_r^t + \frac{\varepsilon}{2}(\bar{\boldsymbol{o}}\boldsymbol{q}_r^t - \boldsymbol{q}_r^t\bar{\boldsymbol{o}}),$$

where $\otimes$ is omitted for brevity. As a result, the real part and dual part are calculated as: $\boldsymbol{q}_r^t = (\cos\frac{\theta^t}{2}, \sin\frac{\theta^t}{2} \cdot \boldsymbol{d}),\ \boldsymbol{q}_d^t = \frac{1}{2}(\bar{\boldsymbol{o}} \otimes \boldsymbol{q}_r^t - \boldsymbol{q}_r^t \otimes \bar{\boldsymbol{o}})$.



Figure A.1: Visualization of VideoArtGS-20 dataset.

A.4 Initialization and Optimization

Inverse Deformation Field Given a point position $\boldsymbol{x}_i^{t_0}$ sampled from the trajectory $\{\boldsymbol{x}_i^t\}_{t=1}^T$, we extend our part assignment module from canonical space to observation space to obtain the part assignment probabilities $\boldsymbol{m}_i^{t_0}$ of $\boldsymbol{x}_i^{t_0}$ at time t_0. Specifically, we deform the learnable centers $C_k = (\boldsymbol{p}_k, \boldsymbol{V}_k, \lambda_k)$ from canonical space to observation space by:

$$\boldsymbol{p}_k^{t_0} = \boldsymbol{R}_k^{t_0} \cdot \boldsymbol{p}_k + \boldsymbol{t}_k^{t_0}, \quad \boldsymbol{V}_k^{t_0} = \boldsymbol{V}_k \cdot (\boldsymbol{R}_k^{t_0})^T, \quad C_k^{t_0} = (\boldsymbol{p}_k^{t_0}, \boldsymbol{V}_k^{t_0}, \lambda_k), \tag{A.7}$$

where $\boldsymbol{R}_k^{t_0}$ and $\boldsymbol{t}_k^{t_0}$ is rotation matrix and translation vector derived from $\boldsymbol{q}_k^{t_0}$. We replace C_k with $C_k^{t_0}$ in Eqs. (2) and (3) to calculate $\boldsymbol{m}_i^{t_0}$, then the canonical position $\hat{\boldsymbol{x}}_i^c$ is calculated by:

$$\boldsymbol{q}_i^{t_0} = \sum_{k=1}^{K} m_{ik}^{t_0} \cdot \boldsymbol{q}_k^{t_0}, \quad \hat{\boldsymbol{x}}_i^c = (\boldsymbol{R}_i^{t_0})^{-1} \cdot (\boldsymbol{x}_i^{t_0} - \boldsymbol{t}_i^{t_0}), \tag{A.8}$$

where $\boldsymbol{R}_i^{t_0}$ and $\boldsymbol{t}_i^{t_0}$ is rotation matrix and translation vector derived from $\boldsymbol{q}_i^{t_0}$.

Training Configuration We train deformation field $\mathcal{F}$ for 10K steps with loss $\mathcal{L}_{\text{track}} = \mathcal{L}_{o2o} + \mathcal{L}_{c2o}$ described in Eq. (4) and Eq. (5), taking 5-10 minutes per object. We train canonical Gaussians $\mathcal{G}^c$ for 20K steps with loss $\mathcal{L}_{\text{render}} = (1 - \lambda_{\text{SSIM}})\mathcal{L}_1 + \lambda_{\text{SSIM}}\mathcal{L}_{\text{D-SSIM}} + \mathcal{L}_D$, where $\lambda_{\text{SSIM}} = 0.2$ is used in experiments. This stage takes about 4 minutes per object. We jointly optimize the canonical Gaussians and deformation field for 20K steps with $\mathcal{L} = \mathcal{L}_I + \mathcal{L}_D + \lambda_{c2o}\mathcal{L}_{c2o}$, where $\lambda_{c2o} = 0.5$. This stage takes 10-20 minutes per object.

A.5 JOINT TYPE PREDICTION USING GPT-4O

Inspired by SINGAPO (Liu et al., 2025), we use GPT-4o to predict the number of joints and joint types. We input the video and a step-by-step instruction to make GPT-4o understand the articulated objects. The version of GPT-4o used in our experiments is gpt-4o-2024-11-20. GPT-4o provides only discrete part-count and joint-type information; it does not predict the continuous geometry or joint parameters optimized by our reconstruction model. These discrete inputs can also be specified manually. The instruction is:

System:

The user will provide a sequence of pictures of articulated objects, some of which shows how human interact with it. The system will analyze:

1. Identify the articulated object in each picture.
2. Identify the static part of this articulated object.
3. Identify all movable parts of the object and output the composed information in the form of one json file.

Example answer:
The articulated object is a carbinet.
The static part is the cabinet base.
The movable parts are the top-drawer, bottom-drawer, and middle-drawer.
The joint type is a slider.
The output json file is:

```
[
  {
    "id": 0,
    "name": "cabinet_base",
    "joint": "heavy",
    "parent": -1
  },
  {
    "id": 1,
    "name": "top_drawer",
    "joint": "slider",
    "parent": 0
  },
  {
    "id": 2,
    "name": "bottom_drawer",
    "joint": "slider",
    "parent": 1
  },
  {
    "id": 3,
    "name": "middle_drawer",
    "joint": "slider",
    "parent": 1
  }
]
```

User:

Analyze the picture of an articulated object and output the information in the form of a json file.

Figure A.2: Prompt for GPT-4o to predict the number of joints and joint types.

B LIMITATIONS

Our method, while effective, has limitations that open avenues for future research.

Dependency on Upstream Perception Models. The final quality of our reconstruction is inherently dependent on the accuracy of the upstream models used for perception. Our pipeline first relies on a monocular depth and camera pose estimator (e.g., VGGT). Subsequently, a pre-trained tracking model (e.g., TAPIP3D) generates 3D motion tracks. If the depth or camera pose estimates contain significant errors, the resulting 3D tracks will be noisy and fail to capture the object's true rigid-body motion. This can lead to failures in our downstream fitting and clustering steps, resulting in distorted geometry or incorrect joint estimation. However, as this is a rapidly advancing field, we anticipate that progress in visual foundation models and tracking models will continue to mitigate this dependency.

Canonical Gaussian Initialization. Our current framework assumes that the input video begins with a short sequence (N frames) where the scene is static. This segment is crucial for initializing the canonical Gaussian representation of the object's geometry. While this assumption is practical for data captured by a user (self-shot), it restricts the method's applicability to in-the-wild videos from the internet, which often begin with immediate motion. Relaxing this condition is non-trivial, as it makes the ill-posed problem of disentangling geometry from motion even more challenging. A promising direction for future work is to incorporate powerful generative priors (Liu et al., 2023b; Lu

Table A.2: **Detailed results on iTACO-S dataset**. Following iTACO(Peng et al., 2025), failed cases are penalized with a 90° axis error and 100cm for other metrics. "Imp." denotes VideoArtGS's improvement over iTACO. The **best results** are highlighted in bold.

Metric	CD-s (cm) ↓			CD-w (cm) ↓			CD-m (cm) ↓			Axis (°) ↓			Position (cm) ↓		
Method	iTACO	Ours	Imp.	iTACO	Ours	Imp.	iTACO	Ours	Imp.	iTACO	Ours	Imp.	iTACO	Ours	Imp.
Box	1.56	**0.03**	1.53	1.04	**0.07**	0.97	0.34	**0.10**	0.24	0.06	**0.06**	0.00	2.31	**0.17**	2.14
Dishwasher	0.74	**0.37**	0.37	0.58	**0.30**	0.28	3.34	**0.14**	3.20	3.27	**0.11**	3.16	11.93	**0.39**	11.54
Laptop	7.30	**0.15**	7.15	0.09	**0.09**	0.00	0.13	**0.10**	0.03	8.40	**0.06**	8.35	3.48	**0.22**	3.26
Microwave	2.84	**0.35**	2.49	2.06	**0.31**	1.74	0.20	**0.07**	0.13	0.40	**0.26**	0.14	9.49	**0.10**	9.39
Refrigerator	22.91	**0.28**	22.63	22.66	**0.16**	22.50	22.05	**0.49**	21.56	18.91	**0.67**	18.24	23.08	**0.30**	22.78
Scissors	10.05	**0.16**	9.88	0.02	**0.02**	0.00	8.52	**0.04**	8.47	30.21	**0.32**	29.90	5.86	**0.23**	5.63
Stapler	64.07	**0.37**	63.70	50.13	**0.27**	49.85	51.26	**0.19**	51.07	46.07	**0.05**	46.01	50.19	**1.83**	48.36
StorageFurniture	**0.61**	0.68	-0.07	0.48	**0.48**	0.00	10.39	**3.02**	7.37	4.47	**0.14**	4.33	0.61	**0.01**	0.60
Table	0.40	**0.35**	0.04	0.43	**0.32**	0.11	19.21	**1.56**	17.65	17.22	**0.44**	16.78	3.01	**0.07**	2.94
USB	1.75	**0.43**	1.32	0.24	**0.19**	0.05	4.27	**0.95**	3.33	11.29	**0.18**	11.12	3.95	**0.05**	3.90
WashingMachine	2.20	**0.82**	1.37	1.93	**0.76**	1.17	4.32	**0.09**	4.23	13.03	**1.29**	11.74	5.28	**1.89**	3.39

Table A.3: **Detailed results on VideoArtGS-20 dataset**. Following iTACO(Peng et al., 2025), failed cases are penalized with a 90° axis error and 100cm for others. "Imp." denotes VideoArtGS's improvement over iTACO. The **best results** are highlighted in bold.

Metric	CD-s (cm) ↓			CD-w (cm) ↓			CD-m (cm) ↓			Axis (°) ↓			Position (cm) ↓		
Method	iTACO	Ours	Imp.	iTACO	Ours	Imp.	iTACO	Ours	Imp.	iTACO	Ours	Imp.	iTACO	Ours	Imp.
100481	2.63	**0.11**	2.52	**0.05**	0.11	-0.07	1.43	**0.03**	1.40	43.93	**0.06**	43.88	35.92	**0.00**	35.91
101284	7.28	**0.01**	7.28	0.02	**0.01**	0.01	76.58	**0.00**	76.58	14.91	**0.18**	14.73	29.31	**0.13**	29.18
101287	10.16	**0.01**	10.15	0.01	**0.01**	0.00	24.70	**0.00**	24.70	9.07	**0.20**	8.87	28.55	**0.03**	28.52
101808	0.89	**0.10**	0.79	**0.09**	0.10	-0.01	13.78	**0.01**	13.76	44.95	**3.79**	41.16	35.94	**0.02**	35.92
101908	2.02	**0.10**	1.93	**0.05**	0.09	-0.04	19.53	**0.01**	19.52	78.44	**0.14**	78.29	65.14	**0.11**	65.03
103015	100.00	**0.27**	99.73	100.00	**0.24**	99.76	100.00	**0.01**	99.99	54.74	**0.07**	54.66	49.06	**0.12**	48.93
103811	4.51	**0.53**	3.98	**0.30**	0.43	-0.13	18.74	**2.65**	16.09	80.81	**0.22**	80.59	0.00	**0.00**	0.00
10489	3.60	**0.05**	3.55	0.07	**0.06**	0.01	60.79	**0.01**	60.78	68.98	**0.08**	68.89	133.32	**0.18**	133.14
10655	3.05	**0.06**	2.99	0.10	**0.08**	0.02	0.08	**0.01**	0.07	7.71	**0.03**	7.68	35.16	**0.20**	34.96
1280	0.56	**0.11**	0.45	**0.02**	0.03	-0.01	6.07	**0.03**	6.04	74.84	**0.68**	74.15	48.53	**0.32**	48.21
168	39.67	**2.68**	36.99	**0.02**	0.04	-0.02	17.23	**0.43**	16.80	53.59	**0.32**	53.27	10.59	**0.22**	10.36
25493	0.75	**0.11**	0.63	0.09	**0.06**	0.03	65.98	**0.18**	65.79	53.67	**0.14**	53.53	0.00	**0.00**	0.00
30666	1.19	**0.22**	0.97	**0.12**	0.17	-0.05	37.51	**1.17**	36.34	66.94	**0.14**	66.80	0.00	**0.00**	0.00
31249	1.51	**0.11**	1.41	0.09	**0.09**	0.00	11.98	**0.22**	11.76	40.68	**0.08**	40.59	3.69	**0.02**	3.68
45194	4.60	**0.09**	4.51	**0.09**	0.10	-0.01	41.50	**0.01**	41.49	47.82	**0.11**	47.71	43.39	**0.03**	43.36
45503	3.11	**0.07**	3.04	**0.07**	0.08	-0.01	21.32	**0.01**	21.31	62.82	**0.03**	62.79	34.59	**0.11**	34.48
45612	1.87	**0.06**	1.81	**0.06**	0.07	-0.02	28.81	**0.02**	28.79	48.19	**0.13**	48.06	16.74	**0.05**	16.70
47648	0.69	**0.06**	0.64	**0.05**	0.06	0.00	24.90	**0.33**	24.57	48.90	**0.18**	48.72	37.10	**0.06**	37.04
8961	4.92	**0.02**	4.90	0.03	**0.03**	0.00	27.25	**0.02**	27.23	1.05	**0.02**	1.03	84.21	**0.05**	84.16
9016	11.38	**0.02**	11.36	0.03	**0.03**	0.00	14.42	**0.02**	14.40	75.53	**0.10**	75.43	49.52	**0.35**	49.17
Average	10.22	**0.24**	9.98	5.07	**0.09**	4.97	30.63	**0.26**	30.37	48.88	**0.34**	48.54	37.04	**0.10**	36.93

et al., 2024a; 2025b). Such models could help infer a plausible canonical shape even from a video with continuous motion, thereby enabling reconstruction from arbitrary monocular inputs.

Reliance on Motion for Part Segmentation. Our approach infers part segmentation exclusively from motion cues by clustering the derived 3D tracks. This reliance on dynamics places high demands on the tracking quality and can be fragile for objects with many parts or for parts that exhibit very subtle relative motion. In such challenging cases, the segmentation quality can degrade, leading to incorrectly merged or split components. A valuable future direction is to augment our motion-based clustering with appearance-based priors from pre-trained foundation models. For instance, integrating semantic features from DINOv2 (Oquab et al., 2023) or segmentation masks from models like SAM (Kirillov et al., 2023) could provide a powerful, independent signal for identifying object parts, making the segmentation process significantly more robust.

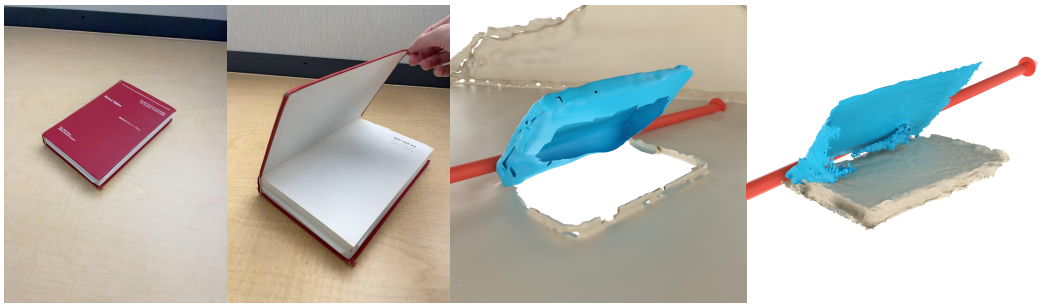
C Additional Experiment Results and Analysis

C.1 Detailed Results on iTACO-S and VideoArtGS-20

We provide category-level results on iTACO-S and object-level results on VideoArtGS-20 in Tabs. A.2 and A.3. The improvements are particularly consistent for movable-part geometry: VideoArtGS achieves lower CD-m in every iTACO-S category and on all 20 objects in VideoArtGS-20.

Table A.4: Quantitative evaluation on iTACO-L (711 videos). Results are mean ± standard deviation. Lower is better for all metrics.

Method	Revolute Joint Estimation			Prismatic Joint Estimation		Reconstruction		
	Axis (°)	Position (cm)	State (°)	Axis (°)	State (cm)	CD-w (cm)	CD-m (cm)	CD-s (cm)
ArticulateAnything	41.83 ± 44.12	73.00 ± 40.00	N/A	47.56 ± 44.69	N/A	8.00 ± 21.00	42.00 ± 64.00	4.00 ± 15.00
iTACO	16.62 ± 29.22	12.00 ± 22.00	15.47 ± 26.36	15.47 ± 22.92	6.00 ± 15.00	1.00 ± 3.00	10.00 ± 21.00	6.00 ± 19.00
Ours	**1.98±10.60**	**1.78±12.00**	**2.57±7.30**	**1.27±6.15**	**0.87±3.08**	**0.30±0.35**	**1.78±7.20**	**0.43±1.23**



Input Views iTACO Ours

Figure A.3: Qualitative comparison with iTACO on real-world data from iTACO.

C.2 Results on iTACO-L

We additionally evaluate on iTACO-L (Peng et al., 2025), which contains 711 videos. Table A.4 reports the comparison with ArticulateAnything and iTACO. Our method achieves lower errors across all reported metrics. For example, revolute-axis error decreases from 16.62° for iTACO to 1.98°, while CD-m decreases from 10.00 to 1.78 cm. These results extend our evaluation beyond iTACO-S.

C.3 Qualitative Comparison on Real-World Videos

We additionally compare with iTACO on real-world data from iTACO (Peng et al., 2025). Figure A.3 presents the available qualitative comparison of movable and static geometry. This comparison complements the main-paper examples but provides no quantitative reconstruction errors.

C.4 Tracking Improvement

Tracks from TAPIP3D (Zhang et al., 2025a) maintain noisy and inaccurate trajectories. Our pipeline filters noise and refines the tracks, enabling more accurate learning of dynamic and articulation parameters. We add track-quality metrics to demonstrate our method's track-correction capability.

Table A.5: **Tracking performance on VideoArtGS-20.** Best results are in bold.

Method	EPE (m) ↓	$\delta_{0.05}$ ↑	$\delta_{0.10}$ ↑
TAPIP3D	0.16	55.57	62.07
Ours	**0.09**	**73.02**	**76.98**
Improvement	+0.07	+17.45	+14.91

Evaluation protocol. Following Shape of Motion (Wang et al., 2025c), we use the 3D end-point-error (EPE), which measures the Euclidean distance between ground truth and predicted 3D tracks. We also report $\delta_{0.05}$ and $\delta_{0.10}$, the percentages of evaluated points whose 3D errors are within 0.05 m and 0.10 m, respectively. Given noisy tracks from TAPIP3D (Zhang et al., 2025a), we use our motion prior guidance pipeline to filter noise and then input the filtered tracks as query points to our optimized model, calculating new tracks using our learned deformation field. Ground truth tracks are obtained by deforming the query points with ground truth joint parameters and states at each time step. Each query point inherits the part label of its nearest vertex on the ground truth meshes.

Results. As shown in Tab. A.6, our method improves all three average tracking metrics. EPE decreases on 19 of 20 objects, while both threshold-based accuracy metrics improve on all 20 objects.

Table A.6: **Tracking performance on VideoArtGS-20 dataset**. "Imp." denotes VideoArtGS's improvement over TAPIP3D. The **best results** are highlighted in bold.

Metric	EPE (m) ↓			$\delta_{0.05}$ ↑			$\delta_{0.10}$ ↑		
Method	TAPIP3D	Ours	Imp.	TAPIP3D	Ours	Imp.	TAPIP3D	Ours	Imp.
100481	0.16	**0.12**	0.04	59.04	**66.95**	7.92	65.40	**72.76**	7.37
101284	0.03	**0.01**	0.02	88.99	**95.96**	6.98	90.46	**96.75**	6.29
101287	0.08	**0.02**	0.06	70.11	**91.73**	21.62	74.16	**93.16**	19.00
101808	0.13	**0.06**	0.07	57.83	**76.65**	18.82	64.66	**81.31**	16.65
101908	0.14	**0.06**	0.08	58.73	**81.16**	22.43	64.64	**84.38**	19.74
103015	0.02	**0.01**	0.02	89.70	**93.38**	3.68	93.35	**96.22**	2.87
103811	0.16	**0.09**	0.08	65.16	**80.87**	15.70	67.86	**81.15**	13.29
10489	0.32	**0.14**	0.18	45.49	**75.72**	30.23	50.88	**78.84**	27.95
10655	0.30	**0.14**	0.15	35.31	**66.00**	30.69	43.81	**69.36**	25.55
1280	0.04	**0.02**	0.02	83.63	**91.12**	7.49	88.27	**94.07**	5.80
168	0.43	**0.32**	0.11	43.91	**52.82**	8.91	51.29	**58.39**	7.10
25493	0.08	**0.04**	0.04	72.33	**78.55**	6.21	76.59	**83.49**	6.90
30666	0.12	**0.06**	0.05	59.75	**80.04**	20.29	64.98	**80.28**	15.30
31249	0.12	**0.07**	0.05	60.21	**76.67**	16.47	63.36	**78.54**	15.17
45194	0.23	**0.12**	0.10	42.23	**70.65**	28.42	48.28	**71.34**	23.06
45503	0.19	**0.10**	0.09	42.32	**69.75**	27.43	48.46	**72.88**	24.41
45612	0.16	**0.08**	0.09	35.50	**66.47**	30.98	43.17	**69.60**	26.43
47648	0.09	**0.03**	0.06	58.05	**81.46**	23.42	66.89	**82.98**	16.10
8961	0.18	**0.15**	0.02	22.74	**29.21**	6.47	37.46	**46.51**	9.05
9016	**0.16**	0.26	-0.10	20.36	**35.27**	14.91	37.47	**47.54**	10.06
Average	0.16	**0.09**	0.07	55.57	**73.02**	17.45	62.07	**76.98**	14.91

Table A.7: Initialization joint parameter error with different thresholds ϵ_l/ϵ_c. The **best results** are highlighted in bold.

ϵ_l/ϵ_c	0.100	0.040	0.020	0.010	0.005
Axis (°)	2.52	2.46	2.32	**2.26**	4.18
Position (cm)	0.51	**0.50**	0.57	0.63	2.39

Table A.8: Optimized results with different thresholds ϵ_l/ϵ_c. The **best results** are highlighted in bold.

ϵ_l/ϵ_c	iTACO	0.100	0.010	0.005
CD-s (cm)	10.22	0.26	**0.24**	0.41
CD-m (cm)	30.63	1.17	**0.26**	14.57
Axis (°)	48.88	2.32	**0.34**	5.13
Position (cm)	37.04	1.06	**0.10**	2.05

These improvements are particularly significant for challenging objects (e.g. 47648, 30666) with complex articulated motion.

C.5 Sensitivity Analysis of Fitting Threshold ϵ_l/ϵ_c

We use equal line/circle fitting thresholds, with $\epsilon_l = \epsilon_c = 0.01$ in the main experiments. Tables A.7 and A.8 summarize the results, and Tables A.9 and A.10 provide object-level results.

The method does not require removing all tracking noise: the learnable hybrid center-grid assignment can refine imperfect initialization during optimization. Nevertheless, the threshold affects reconstruction accuracy. Increasing it from 0.01 to 0.1 changes CD-m from 0.26 to 1.17 cm and axis error from $0.34°$ to $2.32°$. Reducing it to 0.005 rejects too many trajectories and increases CD-m to 14.57 cm and axis error to $5.13°$, with severe degradation on some objects. Although all tested settings outperform the reported iTACO baseline in Table A.8, their accuracy varies.

C.6 Efficiency Comparison

We provide runtime and GPU memory comparison with a single NVIDIA RTX 3090 GPU in Tab. A.11. Our method achieves a favorable balance between efficiency and reconstruction quality. While ArticulateAnything is faster, it uses GPT to predict joint parameters rather than reconstructing from video, making direct comparison less meaningful. Compared to iTACO, our approach reduces

Table A.9: Initialization joint parameter error with different fitting threshold ϵ_l/ϵ_c on VideoArtGS-20 dataset. The **best results** are highlighted in bold.

Metric	Axis (°)					Position (cm)				
ϵ_l/ϵ_c	0.100	0.040	0.020	0.010	0.005	0.100	0.040	0.020	0.010	0.005
100481	5.45	5.45	5.70	5.80	**4.45**	0.16	0.16	**0.15**	1.52	6.92
101284	**0.86**	0.86	0.93	0.93	2.14	**1.34**	**1.34**	1.42	1.70	2.95
101287	**1.07**	1.07	1.14	1.36	3.07	0.35	0.35	0.22	**0.21**	1.21
101808	7.76	7.76	7.78	7.70	**7.04**	1.00	1.00	1.00	1.02	**0.92**
101908	1.21	1.21	1.15	1.42	**0.89**	1.20	1.20	1.60	1.14	**0.32**
103015	**5.89**	**5.89**	5.90	6.14	6.13	**1.25**	**1.25**	**1.25**	1.26	1.36
103811	0.69	0.54	0.65	0.57	**0.38**	0.00	0.00	0.00	0.00	0.00
10489	0.99	0.99	0.94	1.02	**0.50**	**0.22**	**0.22**	0.32	0.40	0.39
10655	**1.48**	1.50	1.56	1.68	2.18	0.54	**0.51**	0.52	0.54	25.51
1280	4.88	4.88	4.88	5.58	**2.48**	1.23	1.23	1.23	1.58	**0.72**
168	**1.54**	**1.54**	1.55	1.65	44.25	1.41	1.41	1.40	**1.32**	3.38
25493	0.34	0.34	0.34	0.37	**0.31**	0.00	0.00	0.00	0.00	0.00
30666	1.10	1.37	0.96	0.91	**0.80**	0.00	0.00	0.00	0.00	0.00
31249	2.50	2.49	2.16	1.08	**0.50**	**0.12**	**0.12**	**0.12**	0.16	0.25
45194	1.75	1.66	1.37	0.66	**0.25**	0.22	0.22	**0.21**	0.26	0.34
45503	1.24	1.23	1.23	**1.22**	2.95	0.10	0.10	0.10	0.11	**0.08**
45612	3.14	3.15	2.77	1.91	**1.72**	0.21	0.21	0.20	0.25	**0.16**
47648	6.16	4.95	2.83	2.89	**2.09**	0.19	0.19	0.19	**0.17**	0.24
8961	0.96	0.96	0.97	0.98	**0.74**	0.42	0.42	0.42	0.41	**0.33**
9016	1.38	1.38	1.51	1.44	**0.80**	0.18	**0.16**	0.98	0.60	2.84
Average	2.52	2.46	2.32	**2.26**	4.18	0.51	**0.50**	0.57	0.63	2.39

Table A.10: Optimized results with different fitting threshold ϵ_l/ϵ_c on VideoArtGS-20 dataset. We ignore the CD-w metric because it changes almost imperceptibly with different ϵ_l/ϵ_c. The **best results** are highlighted in bold.

Metric	CD_s (cm)				CD_m (cm)				Axis (°)				Position (cm)			
ϵ_l/ϵ_c	iTACO	0.100	0.010	0.005	iTACO	0.100	0.010	0.005	iTACO	0.100	0.010	0.005	iTACO	0.100	0.010	0.005
100481	2.63	**0.11**	**0.11**	0.37	1.43	**0.02**	0.03	14.07	43.93	**0.01**	0.06	12.37	35.92	0.02	**0.00**	2.53
101284	7.28	**0.01**	**0.01**	**0.01**	76.58	**0.00**	**0.00**	**0.00**	14.91	**0.14**	0.18	0.35	29.31	**0.10**	0.13	0.15
101287	10.16	**0.01**	**0.01**	**0.01**	24.70	**0.00**	**0.00**	**0.00**	9.07	0.25	**0.20**	0.28	28.55	0.06	**0.03**	0.07
101808	0.89	**0.10**	**0.10**	**0.10**	13.78	**0.01**	**0.01**	**0.01**	44.95	**3.79**	**3.79**	3.80	35.94	0.09	**0.02**	0.09
101908	2.02	**0.10**	**0.10**	**0.10**	19.53	**0.01**	**0.01**	**0.01**	78.44	0.12	0.14	**0.10**	65.14	**0.08**	0.11	0.10
103015	100.00	0.20	0.27	**0.18**	100.00	**0.01**	**0.01**	**0.01**	54.74	0.13	**0.07**	0.13	49.06	0.24	**0.12**	0.23
103811	4.51	0.49	0.53	**0.43**	18.74	4.33	**2.65**	21.53	80.81	**0.17**	0.22	1.24	0.00	**0.00**	**0.00**	**0.00**
10489	3.60	**0.05**	**0.05**	**0.05**	60.79	**0.01**	**0.01**	**0.01**	68.98	**0.08**	**0.08**	0.09	133.32	0.29	**0.18**	0.27
10655	3.05	**0.06**	**0.06**	2.40	0.08	**0.01**	**0.01**	132.42	7.71	**0.01**	0.03	0.04	35.16	**0.16**	0.20	17.51
1280	0.56	0.09	0.11	**0.08**	6.07	0.08	**0.03**	0.12	74.84	**0.67**	0.68	1.05	48.53	0.31	0.32	**0.30**
168	39.67	3.13	**2.68**	3.74	17.23	**0.25**	0.43	92.35	53.59	0.35	**0.32**	45.11	10.59	0.25	**0.22**	0.52
25493	0.75	**0.11**	**0.11**	**0.11**	65.98	**0.15**	0.18	0.16	53.67	**0.11**	0.14	**0.11**	0.00	**0.00**	**0.00**	**0.00**
30666	1.19	**0.22**	**0.22**	**0.22**	37.51	1.20	**1.17**	13.36	66.94	**0.13**	0.14	0.66	0.00	**0.00**	**0.00**	**0.00**
31249	1.51	**0.10**	0.11	0.11	11.98	**0.17**	0.22	0.24	40.68	**0.07**	0.08	**0.07**	3.69	**0.00**	0.02	0.01
45194	4.60	0.15	**0.09**	0.12	41.50	0.14	**0.01**	0.09	47.82	0.25	**0.11**	0.23	43.39	15.82	**0.03**	15.59
45503	3.11	**0.07**	**0.07**	**0.07**	21.32	**0.01**	**0.01**	**0.01**	62.82	**0.02**	0.03	**0.02**	34.59	**0.10**	0.11	**0.10**
45612	1.87	**0.06**	**0.06**	**0.06**	28.81	**0.01**	0.02	**0.01**	48.19	0.12	0.13	**0.11**	16.74	**0.04**	0.05	0.05
47648	0.69	0.07	**0.06**	0.07	24.90	16.92	**0.33**	16.94	48.90	39.79	**0.18**	36.81	37.10	3.21	**0.06**	3.17
8961	4.92	**0.02**	**0.02**	**0.02**	27.25	**0.02**	**0.02**	**0.02**	1.05	**0.02**	**0.02**	**0.02**	84.21	**0.03**	0.05	0.05
9016	11.38	**0.02**	**0.02**	**0.02**	14.42	0.02	0.02	**0.01**	75.53	0.10	0.10	**0.09**	49.52	0.34	0.35	**0.32**
Average	10.22	0.26	**0.24**	0.41	30.63	1.17	**0.26**	14.57	48.88	2.32	**0.34**	5.13	37.04	1.06	**0.10**	2.05

GPU memory requirements and significantly improves efficiency. iTACO requires 20-30 minutes per joint, meaning multi-joint objects can take hours to process, whereas our method completes reconstruction for objects with multiple parts in 15-40 minutes total. This efficiency gain stems from jointly reconstructing all articulated parts rather than processing joints sequentially.

C.7 Failure Case Analysis

GPT-4o Prediction We provide failure cases of GPT-4o prediction in Tab. A.12. GPT-4o made 2 incorrect predictions on the 73 videos in iTACO-S (wrong joint types) and 3 incorrect predictions on VideoArtGS-20 (wrong number of parts or joint types). These counts reflect the raw GPT-4o predictions before manual correction. We manually corrected erroneous predictions of the number of movable parts and joint types before running the reconstruction experiments. For fair comparison, all compared methods were provided with the corrected number of movable parts; thus, errors in GPT-4o

Table A.11: Efficiency Comparison.

Method	GPU Memory	Runtime
ArticulateAnything	N/A	2-5 minutes
iTACO	24GB	20-30 minutes per joint
Ours	12GB	15-40 minutes

Table A.12: Failure cases of GPT-4o. '1r1p' means 1 revolute and 1 prismatic joint.

Dataset	Object ID	Category	GT Joint	Pred. Joint
iTACO-S	19898	Table	1r	1p
iTACO-S	22433	Table	1r	1p
VideoArtGS-20	30666	Table	9p	7p
VideoArtGS-20	103811	Printer	6p	1r1p
VideoArtGS-20	1280	Faucet	2r	3r



(a) (b)

Figure A.4: **Failure cases**. We illustrate failure cases of our VideoArtGS.

part-count predictions did not give our method an advantage over the baselines. The model tends to misclassify joint types when the range of motion is limited and underestimates the number of parts in objects with multiple joints. Notably, we tried alternative methods for automatically detecting the number of parts in the early stages of our experiments, including clustering and detection-based approaches. However, these methods exhibited weaker generalization than GPT-4o. Their reliance on additional assumptions makes them more prone to failure on novel data. We further examine whether these structural prediction errors can be mitigated by using more capable vision-language models. Among the five failure cases of GPT-4o, GPT-5.6 Sol correctly resolves three, leaving only objects 19898 and 103811 incorrectly predicted. GPT-6 Sol correctly predicts the number of movable parts and joint types for all five cases. These results suggest that advances in upstream vision-language models can reduce the need for manual correction of the discrete structural inputs used by our reconstruction framework.

Imperfect Part Segmentation As shown in Fig. A.4 (a), when multiple components are spatially close and share identical motion patterns (e.g., prismatic joints moving in the same direction), our method may fail to correctly segment them. As discussed in Appendix B, this limitation arises from the lack of semantic information in our current approach. Integrating semantic segmentation models such as SAM (Kirillov et al., 2023) could potentially address this issue.

Clustering Error As shown in Fig. A.4 (b), when objects have numerous parts and tracks contain substantial noise, our method may produce incorrect initial centers. Large initialization errors are difficult to correct during optimization, as the model tends to converge to local minima. Addressing this limitation requires more robust methods for discovering part centers, such as incorporating semantic information to help with clustering.

D Additional Related Work

Related scene representations and applications. Scene replica creation (Yu et al., 2025a), reconstruction with generative priors (Ni et al., 2025b;a), and interactive 3D generation (Zhang et al., 2025b) address complementary aspects of building 3D assets. Segmentation lifting (Shen et al., 2025a) and object-centric or vision-language representations (Jia et al., 2023; Huang et al., 2025b;a; Wang et al., 2025e) support their organization and understanding. For VideoArtGS, generative priors could aid canonical-shape initialization, while semantic representations could help distinguish parts with similar motion. Reconstructed part geometry and kinematics could, in turn, provide useful object-level representations for articulated scene modeling and interaction (Huang et al., 2025c; Gu et al., 2025; Zhao et al., 2025).

Camera pose and geometry estimation. DROID-SLAM (Teed & Deng, 2021), Casual-SAM (Zhang et al., 2022), and Mega-SaM (Li et al., 2025b) estimate camera motion and scene geometry. Geometric perception models such as DUSt3R (Wang et al., 2024) and VGGT (Wang et al., 2025b), alongside dynamic perception methods such as MonST3R (Zhang et al., 2024), CUT3R (Wang et al., 2025d), and SpatialTrackerV2 (Xiao et al., 2025), provide foundations for video-based reconstruction. However, recovering geometry and correspondences does not by itself establish part decomposition or joint structure. VideoArtGS addresses this additional step by organizing motion trajectories into articulated parts and using them to initialize and optimize an explicit articulation model.

Articulated generation and animatable models. Articulated priors and generation (Lei et al., 2023; Liu et al., 2024b; Lu et al., 2025c; Chen et al., 2025; Liu et al., 2026; Jin et al., 2025), together with physically plausible part completion (Luo et al., 2025), offer complementary ways to construct articulated assets. Such priors could help complete unobserved geometry, whereas our motion-guided formulation emphasizes consistency with the observed object and its trajectories. Joint geometry-and-motion modeling (Shen et al., 2025b) and category-level animatable reconstruction (Yang et al., 2023) also pursue reposable representations. Within this broader goal, VideoArtGS focuses on converting noisy 3D tracks into structured initialization for joint geometry-and-motion reconstruction.

Articulation understanding and interaction. Related research spans part mobility and structure estimation (Hu et al., 2017; Yi et al., 2018; Wang et al., 2019; Yan et al., 2020; Sun et al., 2023), kinematic modeling (Sturm et al., 2011; Martín-Martín et al., 2016; Jain et al., 2021), and articulated pose estimation and tracking (Li et al., 2020; Liu et al., 2022; Weng et al., 2021; Liu et al., 2023d). Interaction-oriented methods further study part discovery and manipulation (Gadre et al., 2021; Mo et al., 2021; Chu et al., 2023). These directions investigate how object structure determines possible motion and interaction. VideoArtGS connects this structural understanding with surface reconstruction: part assignments and joint parameters are optimized together with object geometry, with temporal correspondences linking the two.

E ADDITIONAL QUALITATIVE RESULTS

We provide additional qualitative results below.

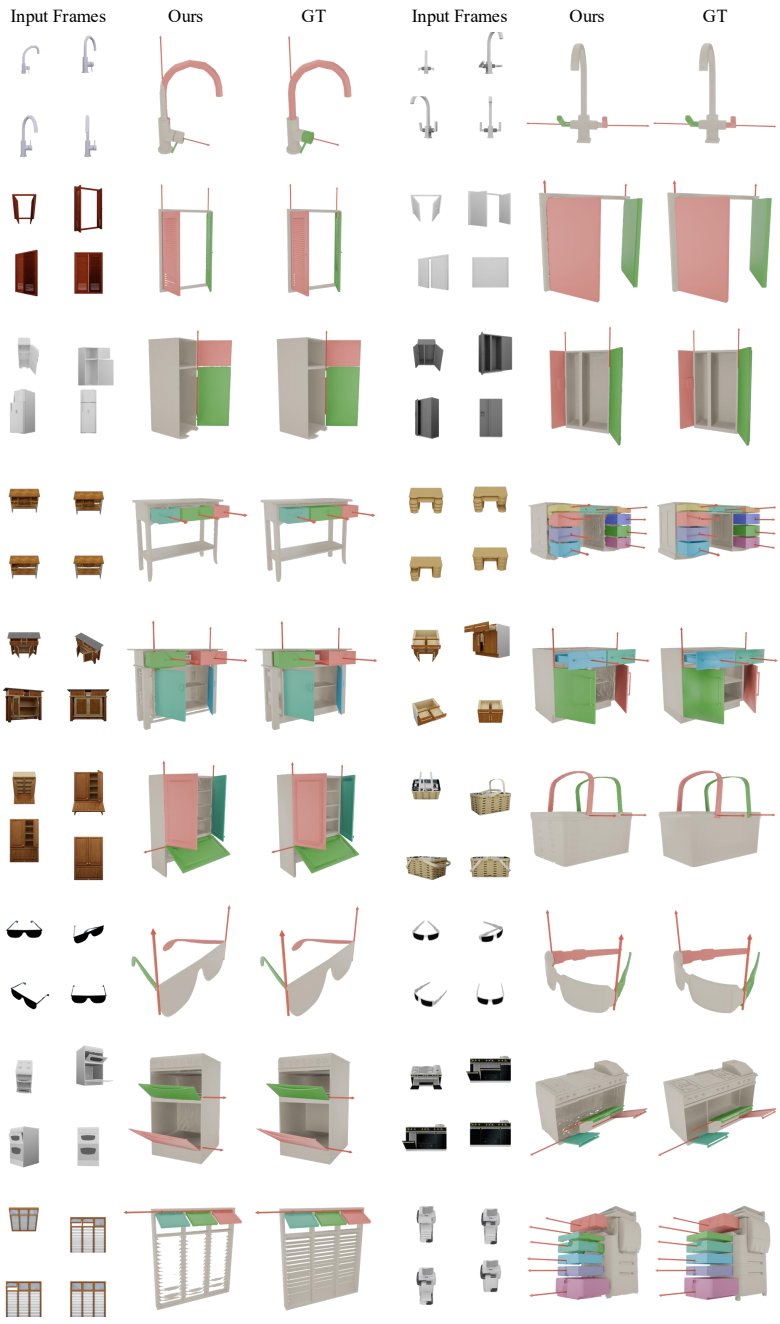


Figure A.5: **Additional qualitative results on VideoArtGS-20.**