ARTMESH: PART-AWARE ARTICULATED MESH FIELDS WITH MOTION-CONSISTENT DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present ArtMesh, a mesh-native method for reconstructing articulated objects explicitly as connected triangle meshes with per-part rigid motion from multi-view images in start and end states. Existing 3D Gaussian Splatting pipelines for articulated reconstruction inherit the unstructured point-based geometry of their splatting base, which provides no surface topology for reasoning about part boundaries or enforcing motion consistency along the object's connectivity. ArtMesh instead builds on a mesh-based differentiable rendering backbone, enabling part-aware dynamics to act directly on the structured topology. To make the topology compatible with articulation, we introduce part-aware restricted Delaunay remeshing, producing connected submeshes whose triangles do not cross semantic part boundaries. The dynamic mesh field then optimizes articulation using bidirectional Vertex-wise Motion Consistency on transported mesh vertices and Pixel-wise Motion Consistency on rendered RGB-D observations. We introduce Articulate-100, a new benchmark of 100 articulated objects spanning 16 PartNet-Mobility categories. On this benchmark, ArtMesh outperforms prior 3DGS-based pipelines in joint-axis estimation and part-level geometric reconstruction, with the largest gains on objects with many movable parts.

1 INTRODUCTION

Reconstructing articulated objects from images is a core problem in building digital twins for robotics, embodied AI, AR/VR, and physical simulation. A useful reconstruction must deliver three things: part-level geometry, the joint motion of each movable part, and an explicit surface asset that downstream systems can consume. Existing pipelines fall short of this goal in two ways. Implicit neural representations, e.g. PARIS (Liu et al., 2023), render well but optimize a volumetric field, so a mesh must be extracted post hoc, losing sharp boundaries and joint topology. Gaussian-splatting pipelines such as ArtGS (Liu et al., 2025c) and GaussianArt (Shen et al., 2026) inherit unstructured point-based geometry without triangle connectivity or part boundaries (Fig. 1(d); full surface comparison in Fig. 8). A mesh must then be recovered by post-hoc TSDF fusion, which introduces holes, smoothing, and ambiguous connectivity near joints, where a single wrong connection across a drawer seam corrupts both the recovered geometry and the estimated motion. We discuss both families in detail in Sec. 2.

A second challenge is motion learning: image-space losses supervise 3D kinematic parameters (joint axes, pivots, translations) only indirectly, and small photometric errors can correspond to large joint-axis errors. Prior methods relate articulation states via 2D feature matching, canonical Gaussian matching, or soft mixtures of part-level motions (Liu et al., 2025c; Shen et al., 2026; Guo et al., 2025), cues that are fragile on weakly textured parts; without an explicit part-aware surface, supervision leaks across neighboring components and different rigid parts share incorrect motion.

We present **ArtMesh**, a mesh-native method for reconstructing articulated objects from multi-view RGB-D observations captured at two articulation states (Fig. 1). ArtMesh represents an object as explicit part-aware triangle meshes linked by per-part rigid motions (Fig. 1(a)). Unlike Gaussian-based pipelines that output ellipsoid primitive-based surfaces, our mesh surface optimized during training is the same surface exported at test time (Fig. 1(b,d)).

The key idea of ArtMesh is to make geometry, topology, and articulation mutually constrained (Fig. 2). Building on mesh-native differentiable rasterization (Held et al., 2026a), we attach geom-

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.

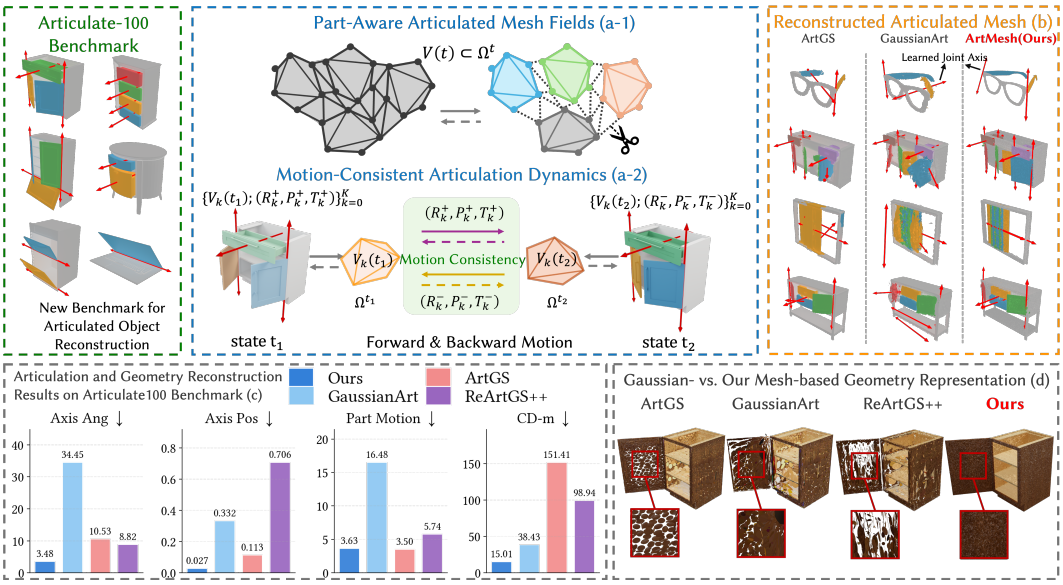

Figure 1: **ArtMesh reconstructs articulated objects as part-aware connected triangle meshes with per-part rigid motion.** **(a)** From multi-view observations at two states, we jointly recover a part-aware mesh field via per-part restricted Delaunay remeshing and a motion-consistent articulation field. **(b)** Start-state meshes transformed to the end state by the learned articulation, with predicted joint axes (red arrows; lengths not meaningful); ArtMesh produces these meshes directly, without post-hoc TSDF fusion. **(c)** Quantitative comparison on *Articulate-100*. **(d)** Baselines output unstructured Gaussians; ArtMesh outputs a connected, opaque triangle mesh directly usable in simulators.

etry, appearance, opacity, and part information to mesh vertices, harden the part assignments, and run part-aware restricted Delaunay remeshing (Fig. 2(a)) within each semantic part, producing connected per-part submeshes with no cross-part triangles, so each movable component transforms as a rigid mesh. An articulation dynamic field (Fig. 2(b)) assigns each part a rigid motion between the two states, with the backward motion the analytic inverse of the forward one. The articulation is optimized with motion consistency learning (Sec. 3.2): vertex-wise consistency transports each vertex through its part motion and compares it with the surface of the same part in the other state, while pixel-wise consistency renders the transported mesh from target-state cameras and compares with the target images.

Existing benchmarks are largely two-part (e.g., PARIS) or target perception and manipulation rather than reconstruction (e.g., GAPartNet (Geng et al., 2023)), leaving multi-part reconstruction underserved. To evaluate articulated reconstruction beyond simple two-part examples, we introduce Articulate-100 (Fig. 1 and Sec. 4.1), a benchmark of 100 articulated objects sampled from PartNet-Mobility (Xiang et al., 2020) across 16 categories and a wide range of part counts. This benchmark stresses multi-part objects where topology, part separation, and motion consistency become increasingly important. On Articulate-100, ArtMesh achieves the best aggregate performance on joint-axis estimation and part-level geometry reconstruction, and shows the strongest gains on objects with three or more parts (Fig. 1(c)).

Our contributions are:

- **Part-aware articulated mesh representation (Sec. 3.1)**: explicit per-part triangle submeshes linked by rigid joints, yielding a directly usable mesh asset without post-hoc conversion; part-aware restricted Delaunay remeshing removes cross-part triangles that would deform incorrectly under rigid motion.
- **Motion-consistency articulation learning (Sec. 3.2)**: we optimize per-part rigid motion by proposed bidirectional vertex-wise and pixel-wise motion consistency, with the backward motion implemented as the analytic inverse of the forward motion.
- **Articulate-100 benchmark (Sec. 4.1)**: a 100-object benchmark spanning 16 PartNet-Mobility (Xiang et al., 2020) categories and diverse part counts, on which ArtMesh achieves the strongest joint estimation and part-level geometry (Fig. 1(c)), with larger gains on objects with more parts.

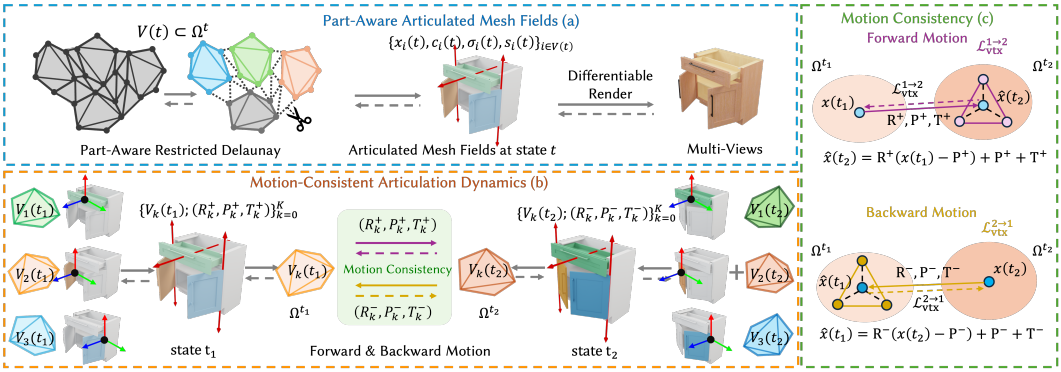

Figure 2: **Overview of our framework.** From multi-view RGB-D observations at two states $t_1, t_2 \in \{0, 1\}$, we reconstruct a pair of part-aware triangle meshes in correspondence and the per-part rigid articulation relating them. The *Part-Aware Mesh Field* (Sec. 3.1) partitions each state's mesh into per-part clusters via part-aware restricted Delaunay triangulation (Fig. 3(a)). The *Articulation Dynamic Field* assigns each part a forward rigid motion (R_k^+, P_k^+, T_k^+) and its analytic inverse. The *Articulation-aware Motion Consistency Loss* (Sec. 3.2) transports each vertex through the articulation and compares its color and opacity against the closest same-part triangle in the other state, in both directions.

2 Related Work

Geometric representations for articulated objects. Implicit methods built on NeRF (Mildenhall et al., 2020) or SDFs (Wang et al., 2021) – A-SDF (Mu et al., 2021), CLA-NeRF (Tseng et al., 2022), NARF (Noguchi et al., 2021), PARIS (Liu et al., 2023), DigitalTwinArt (Weng et al., 2024), Ditto (Jiang et al., 2022b), LEIA (Swaminathan et al., 2024), Articulate-Your-NeRF (Deng et al., 2024) – output volumetric fields needing lossy post-hoc isosurfacing decoupled from optimization. Following 3D Gaussian Splatting (Kerbl et al., 2023), a line of work (Liu et al., 2025c; Shen et al., 2026; Lin et al., 2025; Yu et al., 2026; Guo et al., 2025; Wu et al., 2025; 2026; Kim et al., 2025; Lee et al., 2026) replaces this field with explicit Gaussians; some add surface or structural constraints (e.g., ReArtGS (Wu et al., 2025) regularizes the Gaussian opacity field with an unbiased SDF, its follow-up ReArtGS++ (Wu et al., 2026) instead uses planar Gaussians with a temporal geometry constraint across articulation states, and StructureGS (Lee et al., 2026) enforces part-boundary structure), but all inherit unstructured point-based geometry – gaps and ill-defined surfaces at part boundaries, no connectivity topology – so mesh recovery again needs lossy post-processing. StructureGS is the closest in spirit, adding part-boundary structure to the Gaussians, but like these others it stays point-based and still needs post-hoc meshing. ArtMesh departs from both by reconstructing a connected triangle mesh end-to-end.

Mesh-native differentiable reconstruction. A separate line renders explicit surfaces: mesh rasterizers (Kato et al., 2018; Liu et al., 2019) established mesh-level gradients, and later work converts splatting to meshes (Huang et al., 2024a; Yu et al., 2024; Zhang et al., 2026; Guédon & Lepetit, 2024; Guédon et al., 2025) or renders mesh-like primitives – Triangle Splatting (Held et al., 2026b) (unconnected soup) and MeshSplatting (Held et al., 2026a), our predecessor, which yields connected manifold meshes via a restricted Delaunay step. ArtMesh adopts its backbone and runs the Delaunay step *per part* to respect piecewise-rigid structure.

Feedforward and language-driven articulation estimation. Orthogonal approaches treat articulation as a prediction problem: regressing motion parameters, masks, or URDF assets from single-image or category-level input (Wang et al., 2019; Weng et al., 2021; Mo et al., 2021; Geng et al., 2023; 2024; Zhao et al., 2025; Chen et al., 2024; Yuan et al., 2025; Jiang et al., 2022a; Sun et al., 2024; Liu et al., 2025a; Lu et al., 2025; Gao et al., 2025), or leveraging vision-language models to generate URDFs and affordances (Le et al., 2025; Huang et al., 2024b; Li et al., 2024; 2025). These are fast at inference but bounded by training categories, with geometry often coarse or retrieved from a mesh library. ArtMesh targets the per-object regime, where fidelity matters more than generalization and the surface is recovered from observations.

3 Method

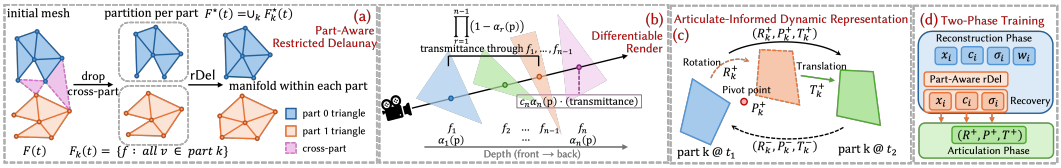

Figure 3: **Method sub-components. (a) Part-Aware Restricted Delaunay:** cross-part triangles (purple, dashed) are dropped and restricted Delaunay runs per cluster, yielding $F^\star(t)$ — manifold within each part. **(b) Differentiable Render:** the mesh faces $f_1, \ldots, f_n$ overlapping a pixel $\mathbf{p}$ are ordered front-to-back by depth. Each face contributes its color-opacity product $c_n\alpha_n(\mathbf{p})$, attenuated by the transmittance $\prod_{r<n}(1-\alpha_r(\mathbf{p}))$ of the faces in front. These contributions are alpha-composited into the pixel color (Eqs. (21), (22)). **(c) Articulate-Informed Dynamic Representation:** the forward rigid transform (R_k^+, P_k^+, T_k^+) carries part k from state t_1 (blue) to state t_2 (green): a rotation R_k^+ about pivot P_k^+, then a translation T_k^+. The backward transform (R_k^-, P_k^-, T_k^-) (dashed) is its analytic inverse, mapping t_2 back to t_1 (Eqs. (3), (18)). **(d) Two-Phase Training:** the *reconstruction phase* optimizes mesh attributes, then runs Part-Aware rDel; the *articulation phase* freezes geometry and optimizes only (R^+, P^+, T^+).

Framework overview. As illustrated in Fig. 2, ArtMesh reconstructs an articulated object from multi-view observations captured at two states $t_1, t_2 \in \{0, 1\}$. Given multi-view RGB-D images at two articulation states, we reconstruct a connected part-aware triangle mesh together with one rigid transform per movable part (joint axis, pivot, and translation) linking the two states, exportable as a URDF asset (Appendix C).

ArtMesh main components. 1) The *Part-Aware Articulated Mesh Field* (Fig. 2(a)) represents each state-t surface as a triangle mesh with per-vertex position, radiance, opacity, and part weights, hardened and remeshed independently within each semantic part so no triangle crosses a part boundary; **2)** The *Vertex-wise Dynamic Articulation Representation* (Fig. 2(b)) assigns each part a forward rigid transform (R_k^+, P_k^+, T_k^+) from t_1 to t_2 and its analytic inverse, linking the part meshes across states with a single consistent motion; **3)** *Articulation-Aware Motion Consistency Learning* (Fig. 2(c)) optimizes these motions bidirectionally: a vertex-wise loss transports vertices through the part motion and matches them to the same-part target surface, and a pixel-wise loss renders the transported mesh from target-state cameras against the observed images.

3.1 Part-Aware Articulated Mesh Field

Part-Aware Articulated Mesh Field. We reconstruct a part-aware mesh field $\mathcal{M}(t) = (V(t), F(t))$ for each state $t \in \{t_1, t_2\}$, where the face set $F(t) \subset V(t) \times V(t) \times V(t)$ specifies the triangle connectivity over the vertices. Each vertex $v_i(t) = (x_i(t), c_i(t), \sigma_i(t), s_i(t)) \in V(t)$ stores a 3D position $x_i(t) \in \mathbb{R}^3$, spherical-harmonic color coefficients $c_i(t)$, opacity $\sigma_i(t) \in [0, 1]$, and part logits $s_i(t) \in \mathbb{R}^{K+1}$ over $K+1$ rigid components, including one static base part and K movable parts. The soft part weights $w_i(t) = \mathrm{softmax}(s_i(t))$ are hardened after the reconstruction stage to labels $p_i(t) = \arg\max_{k\in\{0,\ldots,K\}} w_{i,k}(t)$, which partition the vertices into part-specific sets $V_k(t) = \{v_i(t) \in V(t) : p_i(t) = k\}$ with $V(t) = \bigcup_{k=0}^K V_k(t)$. The opacity $\sigma_i(t)$ and spherical-harmonic color $c_i(t)$ are the vertex attributes alpha-composited by the differentiable mesh renderer used for the image losses (Fig. 3(b); Eqs. (21), (22)). The articulation field (Fig.3 (c)) assigns each part $k \in \{0, \ldots, K\}$ a forward rigid motion consisting of a rotation $R_k^+ \in \mathrm{SO}(3)$, a pivot $P_k^+ \in \mathbb{R}^3$, and a translation $T_k^+ \in \mathbb{R}^3$. Given the hardened part assignment $p(v_i) = \arg\max_k w_{i,k}$, each vertex is mapped from state t_1 to state t_2 by the rigid transform of its assigned part (Eq. (3)). For revolute joints, a non-identity R_k^+ with $T_k^+ = 0$ encodes a pure hinge; for prismatic joints, $R_k^+ = I$ and T_k^+ encodes a slider.

Part-aware Restricted Delaunay Remeshing. Global remeshing is suitable for static scenes but unsafe for articulated objects. Triangles may connect vertices from different parts and shear when those parts move independently. To make the topology compatible with articulation, we adapt restricted Delaunay remeshing (Cheng et al., 2013) to the articulated setting by running it independently within each articulated part, eliminating cross-part triangles that deform incorrectly under rigid motion.

Given the hardened part labels, we first split the face set by part:

$$F_k(t) = \{f = (i, j, l) \in F(t) : p_i(t) = p_j(t) = p_l(t) = k\}\,. \tag{1}$$

Faces whose vertices belong to different parts are discarded. We then run restricted Delaunay remeshing separately on each part:

$$F_k^\star(t) = \mathrm{rDel}\left(V_k(t), F_k(t)\right), \qquad k = 0, \ldots, K. \tag{2}$$

The final topology is the union of all remeshed part topologies, $F^\star(t) = \bigcup_{k=0}^{K} F_k^\star(t)$, giving $\mathcal{M}^\star(t) = (V(t), F^\star(t))$. This operation changes only connectivity. Vertex positions, colors, opacities, and part labels remain attached to their original vertices. As a result, $\mathcal{M}^\star(t)$ is composed of connected per-part submeshes.

3.2 Articulation-aware Motion Consistency Learning

The attributes attached to vertices in $\mathcal{M}(t_1)$ and $\mathcal{M}(t_2)$ are supervised by their own state-specific observations, but moving-part geometry and appearance may drift across states. Therefore, optimizing the articulation only along one direction does not guarantee that the implied inverse motion also explains the opposite-state observations. To tackle it, we introduce a bidirectional motion-consistency learning (Fig. 2(c)) with: **vertex-wise motion consistency**, directly constrains the transported mesh vertices in 3D; and **pixel-wise motion consistency**, constrains the same part-wise motion through differentiable rendering. Both share the same forward rigid transforms and analytic inverse transforms, preventing the forward and backward directions from drifting.

3.2.1 Vertex-wise Motion Consistency.

We transport each vertex by the rigid motion of its assigned part and require it to be consistent with the corresponding surface in the other articulation state.

Vertex-wise Dynamic Articulation Representation. Let $p_i \in \{0, \ldots, K\}$ denote the hardened part label of vertex v_i. For each part k, we write the forward motion from state t_1 to state t_2 as an affine rigid transform (Fig. 3(c))

$$g_k^+(x) \;=\; R_k^+\left(x - P_k^+\right) + P_k^+ + T_k^+ = R_k^+ x + b_k^+. \tag{3}$$

where $b_k^+ = -R_k^+ P_k^+ + P_k^+ + T_k^+$. Thus, the forward transported position of vertex $v_i \in V_{p_i}(t_1)$ is

$$\hat{x}_i^{1\to 2} = g_{p_i}^+(x_i(t_1)) = R_{p_i}^+ x_i(t_1) + b_{p_i}^+. \tag{4}$$

The backward motion (R_k^-, T_k^-, P_k^-) that maps state t_2 back to state t_1 is not independently learned: it is the analytic inverse of the forward affine transform, $g_k^-(y) = (R_k^+)^\top(y - b_k^+) = R_k^- y + b_k^-$, with $R_k^- = (R_k^+)^\top$, $b_k^- = -(R_k^+)^\top b_k^+$, and, in pivot form with $P_k^- = P_k^+$, $T_k^- = -(R_k^+)^\top T_k^+$ (full homogeneous-coordinate derivation in Appendix H).

Bidirectional Vertex Consistency. For each transported vertex $\hat{x}_i^{1\to 2}$, we search only within the same part p_i. Let

$$f_i^{1\to 2} = \arg\min_{f \in F_{p_i}^\star(t_2)} d\big(\hat{x}_i^{1\to 2}, f\big) \tag{5}$$

be the closest triangle in the target-state submesh of the same part, and let $\beta_{ij}^{1\to 2}$ be the barycentric weights of the closest point on this triangle. We interpolate the target vertex attributes (Fig. 2(c)) as

$$\tilde{a}_i^{1\to 2} = \sum_{j \in f_i^{1\to 2}} \beta_{ij}^{1\to 2} a_j(t_2), \qquad a_j(t) = \left(x_j(t), c_j(t), \sigma_j(t)\right), \quad \tilde{a}_i^{1\to 2} = \left(\tilde{x}_i^{1\to 2}, \tilde{c}_i^{1\to 2}, \tilde{\sigma}_i^{1\to 2}\right). \tag{6}$$

The forward vertex-wise motion-consistency loss is then

$$\mathcal{L}_{\mathrm{vtx}}^{1\to 2} = \frac{1}{|V(t_1)|} \sum_{v_i \in V(t_1)} \eta_i^{1\to 2} \Big[\lambda_x \left\| \hat{x}_i^{1\to 2} - \tilde{x}_i^{1\to 2} \right\|_2^2 + \lambda_c \left\| c_i(t_1) - \tilde{c}_i^{1\to 2} \right\|_2^2 + \lambda_\sigma \left| \sigma_i(t_1) - \tilde{\sigma}_i^{1\to 2} \right|^2 \Big], \tag{7}$$

where $\eta_i^{1\to 2}$ is a validity indicator that masks vertices without a valid same-part target correspondence, the geometric term pulls each transported vertex $\hat{x}_i^{1\to 2}$ toward its interpolated target position $\tilde{x}_i^{1\to 2}$, and the color and opacity terms enforce appearance consistency under the rigid part motion. The backward direction is defined symmetrically: each vertex $v_j \in V(t_2)$ is transported back to state t_1 using the inverse motion, $\hat{x}_j^{2\to 1} = g_{p_j}^-(x_j(t_2))$. We then find the closest same-part triangle in $F_{p_j}^\star(t_1)$, interpolate the corresponding target attributes, and compute $\mathcal{L}_{\mathrm{vtx}}^{2\to 1}$ similarly to Eq. 7. This term directly supervises the articulated motion in 3D triangle vertex space.

3.2.2 PIXEL-WISE MOTION CONSISTENCY.

The pixel-wise motion-consistency supervises the same part-wise affine motion with differentiable rendering. With the renderer $\mathcal{R}$, applying the forward part motions to the state t_1 mesh gives the articulated source mesh

$$\hat{\mathcal{M}}^{1\to 2} = G^{+}(\mathcal{M}(t_1)) = \left(\left\{\hat{x}_i^{1\to 2}, c_i(t_1), \sigma_i(t_1)\right\}_{v_i\in V(t_1)}, F^{\star}(t_1)\right), \tag{8}$$

where each vertex position is transformed by the affine motion of its assigned part as in Eq. (4). For a camera $\Pi_m^{t_2}$ observing state t_2, we render the RGB image $\hat{I}_m^{1\to 2} = \mathcal{R}_I\left(\hat{\mathcal{M}}^{1\to 2}, \Pi_m^{t_2}\right)$, where $\mathcal{R}_I$ denotes the RGB channel of the renderer $\mathcal{R}$ in Eq. (12). The forward pixel-wise motion-consistency loss compares the rendered observations with the true ones at state t_2:

$$\mathcal{L}_{\text{pix}}^{1\to 2} = \frac{1}{|\mathcal{C}_{t_2}|}\sum_{m\in\mathcal{C}_{t_2}} \ell_{\text{pix}}\left(\hat{I}_m^{1\to 2}, I_m(t_2)\right), \tag{9}$$

where the per-view loss ℓ_{pix} combines an L_1 RGB term and an SSIM term,

$$\ell_{\text{pix}}(\hat{I}, I) = \lambda_{\text{rgb}}\left\|\hat{I} - I\right\|_1 + \lambda_{\text{ssim}}\left(1 - \text{SSIM}(\hat{I}, I)\right). \tag{10}$$

The backward pixel-wise motion-consistency term is defined analogously. The inverse motion G^{-} is applied to the state-t_2 mesh to produce $\hat{\mathcal{M}}^{2\to 1}$, which is then rendered from state-t_1 cameras $\Pi_m^{t_1}$ and compared against the true state-t_1 observations $I_m(t_1)$ to form $\mathcal{L}_{\text{pix}}^{2\to 1}$, with the same per-view loss ℓ_{pix} as in Eq. (9).

This loss provides dense pixel-level supervision for the same part-wise affine motion used in the vertex-wise term. The final motion-consistency objective sums the two vertex-wise and two pixel-wise terms,

$$\mathcal{L}_{\text{motion}} = \lambda_{\text{VMC}}\underbrace{\left(\mathcal{L}_{\text{vtx}}^{1\to 2} + \mathcal{L}_{\text{vtx}}^{2\to 1}\right)}_{\mathcal{L}_{\text{VMC}}} + \lambda_{\text{PMC}}\underbrace{\left(\mathcal{L}_{\text{pix}}^{1\to 2} + \mathcal{L}_{\text{pix}}^{2\to 1}\right)}_{\mathcal{L}_{\text{PMC}}}. \tag{11}$$

The two phases (Fig. 3(d)) form one end-to-end training run, with the articulation-phase freeze preventing articulation signals from corrupting canonical geometry that has already converged; exact iteration budgets are in Appendix G.

3.3 DIFFERENTIABLE MESH RENDERING AND OPTIMIZATION

For a camera Π_m^t at state t, mesh renderer $\mathcal{R}$ produces RGB, depth, and part-mask predictions:

$$\left(\hat{I}_m(t), \hat{D}_m(t), \hat{S}_m(t)\right) = \mathcal{R}\left(\mathcal{M}^{\star}(t), \Pi_m^t\right). \tag{12}$$

For rendering, the faces overlapping a pixel are ordered front-to-back and alpha-composited with opacity-weighted smooth triangle window functions (Fig. 3(b)), detailed in Appendix H. We optimize each state's mesh with a differentiable mesh renderer $\mathcal{R}$ by minimizing a reconstruction loss between its rendered outputs and their observations (defined in Appendix G). During training, for each camera $m \in \mathcal{C}_t$ at state t, we have the RGB image $I_m(t)$, depth map $D_m(t)$ observations, and semantic part map $S_m(t)$ (ground-truth or predicted by our PArtSAM).

PArtSAM. When part masks are not provided, inspired by PartSAM(Zhu et al., 2026), we propose a **P**art-level 3D-consistent **Art**iculation segmentation, **PArt**SAM, to predict them: it back-projects the posed RGB-D views of each state into a colored point cloud, uses the motion cues between the two states to separate movable from static surfaces and to split the movable region into individual parts, and renders the resulting point labels back into per-view part maps in the same format as $S_m(t)$. Full details are in Appendix D.

In the reconstruction stage, we optimize vertex positions, radiance, opacity, and part logits for each state. We then harden part assignments, apply the per-part restricted Delaunay remeshing in Eq. (2), and run a short reconstruction stage with fixed topology to refine vertex positions and appearance after remeshing. In the articulation stage, the part-aware meshes $\mathcal{M}^{\star}(t_1)$ and $\mathcal{M}^{\star}(t_2)$ are fixed, and only the per-part rigid motion parameters (Fig. 3 (c)) are optimized using the vertex-wise and pixel-wise motion-consistency losses defined in Sec. 3.2. Geometry is frozen here, so we supervise with RGB and the vertex-wise consistency only. The reconstruction objective, mesh initialization, and hyperparameters are in Appendix G.

4 EXPERIMENTS

4.1 BENCHMARKS AND IMPLEMENTATION DETAILS

Articulate-100. We construct **Articulate-100** (Fig. 10 in the appendix), a benchmark of 100 articulated objects randomly sampled from PartNet-Mobility (Xiang et al., 2020) across 16 categories, from single-joint objects (*Laptop*, *Knife*, *Pliers*) to multi-part furniture. The distribution is dominated by *Table* (34) and *StorageFurniture* (42), reflecting PartNet-Mobility itself, to expose how each method scales with articulation complexity; Fig. 10 shows sample objects and the category and part-count distributions.

PARIS. For comparability with prior work and real-world evaluation, we additionally use the PARIS benchmark (Liu et al., 2023), containing 10 synthetic and 2 real articulated objects with two-part articulation (one revolute or prismatic joint).

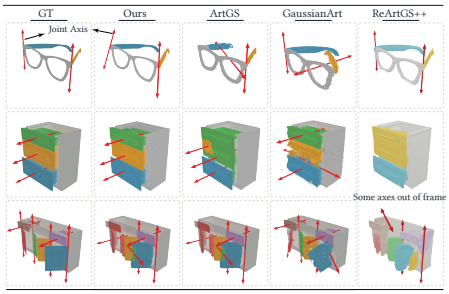

Figure 4: Qualitative comparisons on multi-part Articulate-100 objects (state-1 meshes); both states and more samples in Fig. 11. Red arrows mark predicted joint axes (lengths not meaningful).

Baselines. On Articulate-100, we compare ArtMesh against three articulated 3DGS pipelines: **ArtGS** (Liu et al., 2025c), **GaussianArt** (Shen et al., 2026), and **ReArtGS++** (Wu et al., 2026). On PARIS we include the original **PARIS** (Liu et al., 2023) method. The multi-view semantic information ArtMesh and the segmentation-consuming baselines rely on can come from two sources: ground-truth part masks, or masks predicted by a segmentation frontend (our finetuned PArtSAM; Appendix D).

Metrics. We report five metrics: **Axis Ang** (°), angular deviation between predicted and ground-truth joint axes; **Axis Pos** (0.1 m), Euclidean distance between predicted and ground-truth axis origins (revolute joints only); **Part Motion** (° for revolute, m for prismatic), joint-state error; **CD-s** (mm), Chamfer Distance on static parts; and **CD-m** (mm), Chamfer Distance on movable parts. Chamfer Distances are computed on 10,000 points sampled from predicted and ground-truth meshes.

4.2 RESULTS ON ARTICULATE-100

Qualitative results. Fig. 4 and Fig. 11 show qualitative comparisons on representative multi-part objects. ArtGS produces ambiguous part groupings and misaligned joints once the part count exceeds three, which manifests as fractured movable meshes and incorrect learned motion. GaussianArt recovers reasonable static geometry but its predicted axes drift on objects with many similarly-shaped parts (e.g., multi-drawer storage units). ArtMesh produces clean part separations and correctly aligned axes.

Quantitative results. Table 1 reports per-bucket and aggregate results. On the 2-part bucket, ArtGS attains lower Part Motion and Axis Pos errors (it derives segmentation from motion, and a one-static/one-movable split is much easier), while ArtMesh achieves lower Axis Ang error and substantially lower Chamfer Distances. The gap widens with part count: ArtMesh is best on every metric for objects with three or more parts, and in aggregate is best on Axis Ang, Axis Pos, and both Chamfer Distances, with Part Motion close behind ArtGS. ReArtGS++ (Wu et al., 2026) lags on the remaining articulation metrics and on movable-part geometry, so its planar Gaussian articulation does not scale to many parts. The failure patterns of ArtGS and GaussianArt match the limitations each paper identifies, amplified by our broader category mix; see Appendix E. Ours-PArtSAM replaces the ground-truth masks with predictions from our finetuned PArtSAM segmenter (Appendix D); the row measures how much the *reconstruction* degrades when segmentation is predicted rather than given. On 2-part objects the parts are large and well separated, so predicted masks are essentially as accurate as ground truth and the reconstruction is on par. On multi-part objects, segmentation noise at part boundaries propagates into the geometry and the Chamfer Distances grow accordingly. Even so, the predicted-mask reconstruction stays competitive with the baselines, matching ArtGS on aggregate Axis Ang and Part Motion and staying far ahead of GaussianArt (Sec. 4.1, Appendix D). Where a predicted-mask value edges out its ground-truth counterpart, this is a scoring artifact: over-segmenting a part into a larger region averages the Part Motion error over more (partly static) surface and reads lower.

Table 1: Quantitative results on Articulate-100 benchmark. For each metric and part-count column, the **best** value is in bold. Ours-PArtSAM uses predicted PArtSAM part masks instead of the ground-truth masks all other rows receive (Sec. 4.2).

		2 Parts (34)	3 Parts (20)	4-5 Parts (31)	6+ Parts (15)	All (100)
Axis Ang (°) ↓	ArtGS	2.72	11.32	13.42	21.19	10.53
	ReArtGS++	**0.67**	5.76	15.62	17.33	8.82
	GaussianArt-GTMask	53.25	32.49	19.73	24.87	34.45
	Ours-PArtSAM	1.74	9.95	17.38	16.31	10.42
	Ours-GTMask	2.02	**4.30**	**2.93**	**6.83**	**3.48**
Axis Pos (0.1 m) ↓	ArtGS	**0.00**	0.05	0.19	0.30	0.11
	ReArtGS++	0.12	0.75	0.83	1.72	0.71
	GaussianArt-GTMask	**0.00**	0.15	0.83	0.29	0.33
	Ours-PArtSAM	0.05	0.36	0.14	0.08	0.15
	Ours-GTMask	0.04	**0.03**	**0.01**	**0.03**	**0.03**
Part Motion (°/m) ↓	ArtGS	**0.32**	4.96	3.71	8.36	**3.50**
	ReArtGS++	4.28	7.89	4.79	8.14	5.74
	GaussianArt-GTMask	25.52	14.76	9.42	12.86	16.48
	Ours-PArtSAM	1.27	3.44	4.69	6.68	3.58
	Ours-GTMask	5.82	**2.12**	**2.32**	**3.42**	3.63
CD-s (mm) ↓	ArtGS	22.37	22.39	24.92	25.03	23.56
	ReArtGS++	20.37	22.58	28.37	39.30	26.13
	GaussianArt-GTMask	22.91	24.23	26.27	25.64	24.63
	Ours-PArtSAM	18.54	19.59	28.42	47.40	26.14
	Ours-GTMask	**17.95**	**18.86**	**19.66**	**20.11**	**18.98**
CD-m (mm) ↓	ArtGS	34.17	157.51	195.35	318.18	151.41
	ReArtGS++	37.95	114.88	136.17	139.01	98.94
	GaussianArt-GTMask	57.48	31.73	24.94	32.02	38.43
	Ours-PArtSAM	15.23	79.71	82.57	78.71	58.52
	Ours-GTMask	**11.09**	**15.74**	**15.27**	**22.40**	**15.01**

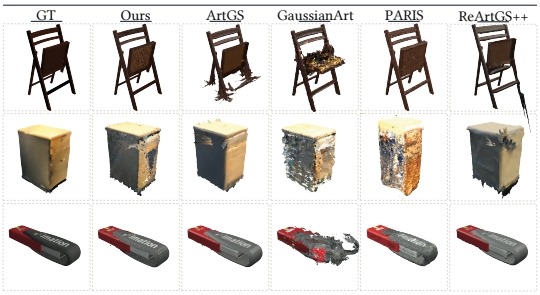

Figure 5: Results on PARIS (Liu et al., 2023), where ArtMesh's high-part-count advantages are least exercised. Minor color differences reflect Blender-rendered meshes rather than rasterizer output.

Table 2: Paris benchmark results, averaged over all objects of each split with joint types weighted by object count (Syn: 8 revolute, 2 prismatic; Real: 1 each). **Best**, Second.

	Axis Ang ↓	Axis Pos ↓	Part Motion ↓	CD-s ↓	CD-m ↓
Syn (10)					
PARIS	5.76	1.82	51.48	28.19	85.79
ArtGS	**0.03**	**0.00**	**0.02**	25.50	19.14
GaussianArt	72.01	**0.00**	36.50	22.70	58.96
ReArtGS++	13.14	**0.00**	**0.02**	25.61	58.26
Ours	0.68	0.02	0.54	**19.59**	**12.44**
Real (2)					
PARIS	7.76	0.05	2.54	114.11	259.99
ArtGS	2.95	0.25	0.97	37.36	**47.86**
GaussianArt	46.14	**0.00**	21.52	48.22	140.28
ReArtGS++	40.81	0.20	1.31	32.24	59.18
Ours	**2.37**	0.07	**0.66**	**27.25**	50.84

4.3 Results on PARIS

We additionally evaluate on PARIS (Liu et al., 2023), a small two-part benchmark where ArtMesh's scaling and per-part-isolation advantages are least exercised. Results are in Table 2 and Fig. 5, with additional textured comparisons in Fig. 9 in the appendix. On the synthetic split, ArtGS and ArtMesh are competitive on motion parameters; GaussianArt fails on revolute joints with near-orthogonal axis predictions (90° Axis Ang on the revolute subset), the same cross-part blending failure observed on Articulate-100. ArtMesh attains the best Chamfer Distances on both static and movable parts, reflecting its structured mesh representation. On the challenging real split, ArtMesh achieves the best Axis Ang, Part Motion, and CD-s, and is second to ArtGS on CD-m.

4.4 Ablation Studies

We ablate four components (Table 3, Fig. 6). (1) **Vertex-wise appearance (color) consistency:** the strongest cross-state cue; removal raises Axis Ang and CD-m on weakly textured parts. (2) **Vertex-wise geometry (opacity) consistency:** removal degrades motion comparably with smaller

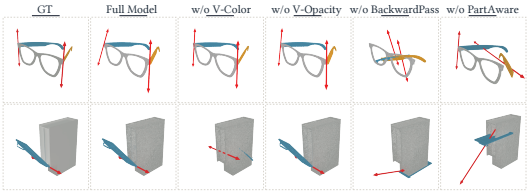

Figure 6: Ablation of four ArtMesh components on Articulate-100: (i) *color consistency*, (ii) *opacity consistency*, (iii) the *backward pass*, and (iv) *part-aware restricted Delaunay*. See Table 3.

Table 3: Results on Articulate-100 benchmark. Best are in **bold**.

	Full Model	w/o V-Color	w/o V-Opacity	w/o Backward	w/o Part-Aware
Axis Ang (°) ↓	**3.48**	5.87	6.49	10.25	7.02
Axis Pos (0.1 m) ↓	**0.03**	0.04	0.04	0.06	**0.03**
Part Motion (°/m) ↓	**3.63**	5.39	5.49	5.81	4.95
CD-s (mm) ↓	18.98	18.96	18.99	18.94	**18.58**
CD-m (mm) ↓	**15.01**	21.91	19.86	29.06	19.30

geometry effects, mainly disambiguating which triangles move together. (3) **Backward pass:** the most damaging ablation (Axis Ang, Part Motion, CD-m); bidirectional supervision keeps the motion field self-consistent. (4) **Part-aware restricted Delaunay:** replacing it with global Delaunay degrades Axis Ang and CD-m while slightly improving CD-s; the part-aware constraint thus trades a negligible amount of static-geometry accuracy for better articulation, its main role being to prevent consistency losses from pulling vertices across joints.

4.5 Mesh Quality

Optimizing the mesh directly helps twice. Per-part connectivity gives the motion-consistency losses a rigid structure, so joints are estimated on the exported geometry, whereas ArtGS, ReArtGS++, and GaussianArt produce no mesh until post-hoc TSDF extraction. Our exported mesh is directly usable in simulators without any TSDF step (Fig. 1(d), Appendix C); for a like-for-like comparison we apply the same TSDF pass the baselines require, under which our mesh has the lowest Self-Penetration Depth (Table 4).

Failure cases and limitations are analyzed in Appendix F.

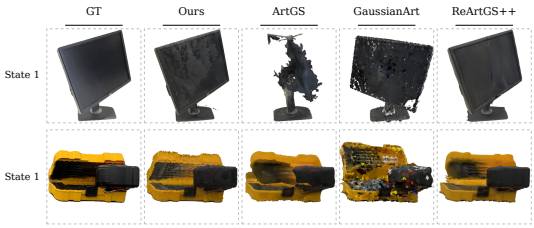

Figure 7: Real-world reconstructions on SplArt (Lin et al., 2025) (commanded state; both states in Fig. 13). ArtMesh produces clean meshes from real captures, while ArtGS collapses on the texture-poor monitor, GaussianArt yields fragmented surfaces, and ReArtGS++ reconstructs some objects but does not cleanly recover the more complex textures.

Table 4: Mesh quality via Self-Penetration Depth (SPD ↓). ArtGS, ReArtGS++, and GaussianArt have no direct mesh (N/A) and require TSDF extraction.

	ArtGS	GaussianArt	ReArtGS++	Ours
Mesh ↓	N/A	N/A	N/A	0.0494
TSDF ↓	0.0247	0.0128	0.0100	0.0043

5 Conclusion

We presented ArtMesh, a method for reconstructing articulated objects as connected triangle meshes with per-part rigid motion from multi-view images at two states. Three design choices drive it: per-part restricted Delaunay remeshing that keeps triangles from crossing part boundaries, color and opacity consistency losses that supervise articulation through 3D vertex correspondences rather than 2D feature matches, and a bidirectional cycle. On Articulate-100, ArtMesh outperforms the evaluated 3DGS baselines (ArtGS, ReArtGS++, and GaussianArt) on joint-axis estimation and part-level geometry, with the largest gains on many-part objects. Limitations: accuracy drops under predicted rather than provided masks, and matching the baselines' lowest self-penetration still needs the same TSDF pass. Improving surface quality, reducing the segmentation reliance, and extending cycle training to multi-state sequences remain future work; the simulator-ready meshes already narrow the gap between reconstruction and the geometry pipelines of robotics and simulation.

Ethics Statement

ArtMesh reconstructs articulated 3D geometry from multi-view captures and involves no human subjects, personal data, or sensitive attributes; our experiments use synthetic PartNet-Mobility (Xiang et al., 2020) objects, the PARIS (Liu et al., 2023) benchmark, and publicly released SplArt (Lin et al., 2025) captures. Part-level digital twins of articulated objects are intended for simulation, robotics, and interactive graphics; as with any reconstruction pipeline, downstream physical deployment should include standard validation and safety checks, since the recovered meshes and estimated joints are approximate. ArtMesh also inherits the object and category distribution of PartNet-Mobility, which is skewed toward common furniture; we report results broken down by part count so this bias is visible rather than hidden.

Large language models were used for surface-level writing assistance (grammar and phrasing), and we used AI code assistants to help write portions of the implementation. ArtMesh itself contains no language or foundation-model component: its geometry and per-part articulation are optimized purely from multi-view image observations.

Reproducibility Statement

Full training and implementation details, hyperparameters, and the optimization schedule are given in Appendix G, with additional method details in Appendix H. The construction of the Articulate-100 benchmark is described in Section 4.1, the evaluation protocol and the five metrics in Section 4.1, and the predicted-mask (PArtSAM) and further experiments in Appendix D. The Articulate-100 benchmark is publicly available, and we will release the code upon publication.

REFERENCES

Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. URDFormer: A Pipeline for Constructing Articulated Simulation Environments from Real-World Images. In *RSS*, 2024.

Siu-Wing Cheng, Tamal Krishna Dey, Jonathan Shewchuk, and Sartaj Sahni. *Delaunay mesh generation*. CRC Press Boca Raton, 2013.

Jianning Deng, Kartic Subr, and Hakan Bilen. Articulate your NeRF: Unsupervised articulated object modeling via conditional view synthesis. In *NeurIPS*, 2024.

Mingju Gao, Yike Pan, Huan-ang Gao, Zongzheng Zhang, Wenyi Li, Hao Dong, Hao Tang, Li Yi, and Hao Zhao. PartRM: Modeling Part-Level Dynamics with Large Cross-State Reconstruction Model. In *CVPR*, pp. 7004–7014, 2025.

Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. GAPartNet: Cross-Category Domain-Generalizable Object Perception and Manipulation via Generalizable and Actionable Parts. *CVPR*, 2023.

Haoran Geng, Songlin Wei, Congyue Deng, Bokui Shen, He Wang, and Leonidas Guibas. SAGE: Bridging Semantic and Actionable Parts for GEneralizable Articulated-Object Manipulation under Language Instructions. In *RSS*, 2024.

Antoine Guédon and Vincent Lepetit. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. In *CVPR*, 2024.

Antoine Guédon, Diego Gomez, Nissim Maruani, Bingchen Gong, George Drettakis, and Maks Ovsjanikov. MILo: Mesh-In-the-Loop Gaussian Splatting for Detailed and Efficient Surface Reconstruction. *ACM Transactions on Graphics*, 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 *CVPR*, pp. 27144–27153, 2025.

Jan Held, Sanghyun Son, Renaud Vandeghen, Daniel Rebain, Matheus Gadelha, Yi Zhou, Anthony Cioppa, Ming C. Lin, Marc Van Droogenbroeck, and Andrea Tagliasacchi. MeshSplatting: Differentiable Rendering with Opaque Meshes. In *CVPR*, 2026a.

Jan Held, Renaud Vandeghen, Adrien Deliege, Abdullah Hamdi, Silvio Giancola, Anthony Cioppa, Andrea Vedaldi, Bernard Ghanem, Andrea Tagliasacchi, and Marc Van Droogenbroeck. Triangle Splatting for Real-Time Radiance Field Rendering. In *3DV*, 2026b.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *SIGGRAPH*. Association for Computing Machinery, 2024a.

Siyuan Huang, Haonan Chang, Yuhan Liu, Yimeng Zhu, Hao Dong, Peng Gao, Abdeslam Boularias, and Hongsheng Li. A3VLM: Actionable Articulation-Aware Vision Language Model. *CoRL*, 2024b.

Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. OPD: Single-view 3D openable part detection. In *ECCV*, pp. 410–426. Springer, 2022a.

Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building Digital Twins of Articulated Objects from Interaction. In *CVPR*, 2022b.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D Mesh Renderer. In *CVPR*, 2018.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Seungyeon Kim, Junsu Ha, Young Hun Kim, Yonghyeon Lee, and Frank C. Park. ScrewSplat: An End-to-End Method for Articulated Object Recognition. In *CoRL*, 2025.

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. *ICLR*, 2025.

Gahye Lee, Gyoonseo Kim, Wonjong Jang, Jooeun Son, and Seungyong Lee. StructureGS: Structure-aware Gaussian Splatting for Articulated Object Reconstruction. In *ECCV*, 2026.

Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. ManipLLM: Embodied Multimodal Large Language Model for Object-Centric Robotic Manipulation. In *CVPR*, pp. 18061–18070, 2024.

Zhe Li, Xiang Bai, Jieyu Zhang, Zhuangzhe Wu, Che Xu, Ying Li, Chengkai Hou, and Shanghang Zhang. URDF-Anything: Constructing Articulated Objects with 3D Multimodal Language Model. In *NeurIPS*, 2025.

Shengjie Lin, Jiading Fang, Muhammad Zubair Irshad, Vitor Campagnolo Guizilini, Rares Andrei Ambrus, Greg Shakhnarovich, and Matthew R. Walter. SplArt: Articulation Estimation and Part-Level Reconstruction with 3D Gaussian Splatting. In *ICCV*, 2025.

Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. PARIS: Part-level Reconstruction and Motion Analysis for Articulated Objects. In *ICCV*, pp. 352–363, 2023.

Jiayi Liu, Denys Iliash, Angel X Chang, Manolis Savva, and Ali Mahdavi-Amiri. SINGAPO: Single Image Controlled Generation of Articulated Parts in Object. *ICLR*, 2025a.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. PartField: Learning 3D Feature Fields for Part Segmentation and Beyond. In *ICCV*, 2025b.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning. In *ICCV*, 2019.

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 *ICLR*, 2025c.

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. In *SIGGRAPH Asia*, 2025.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020.

Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2Act: From Pixels to Actions for Articulated 3D Objects. In *ICCV*, 2021.

Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan L. Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-SDF: Learning Disentangled Signed Distance Functions for Articulated Shape Representation. In *ICCV*, pp. 12981–12991, 2021.

Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural Articulated Radiance Field. In *ICCV*, 2021.

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. In *3DV*, pp. 384–395, 2026.

Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel Xuan Chang. OPDMulti: Openable Part Detection for Multiple Objects. *3DV*, 2024.

Archana Swaminathan, Anubhav Gupta, Kamal Gupta, Shishira R. Maiya, Vatsal Agarwal, and Abhinav Shrivastava. LEIA: Latent View-invariant Embeddings for Implicit 3D Articulation. In *ECCV*, 2024.

Wei-Cheng Tseng, Hung-Ju Liao, Yen-Chen Lin, and Min Sun. CLA-NeRF: Category-Level Articulated Neural Radiance Field. In *ICRA*, 2022.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *NeurIPS*, 2021.

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 *CVPR*, pp. 8876–8884, 2019.

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 *ICCV*, pp. 13209–13218, October 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 *CVPR*, 2024.

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. In *NeurIPS*, 2025.

Di Wu, Liu Liu, Anran Huang, Yuyan Liu, Qiaojun Yu, Shaofan Liu, Liangtu Song, and Cewu Lu. REArtGS++: Generalizable Articulation Reconstruction with Temporal Geometry Constraint via Planar Gaussian Splatting. In *CVPR*, 2026.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A SimulAted Part-based Interactive ENvironment. In *CVPR*, June 2020.

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. In *CVPR*, 2026.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes. *ACM Transactions on Graphics*, 43(6):1–13, 2024.

Sylvia Yuan, Ruoxi Shi, Xinyue Wei, Xiaoshuai Zhang, Hao Su, and Minghua Liu. LARM: A Large Articulated-Object Reconstruction Model. In *SIGGRAPH Asia*, 2025.

Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. RaDe-GS: Rasterizing Depth in Gaussian Splatting. *ACM Transactions on Graphics*, 45(2):1–14, 2026.

Mandi Zhao, Yijia Weng, Dominik Bauer, and Shuran Song. Real2Code: Reconstruct Articulated Objects via Code Generation. In *ICLR*, 2025.

Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. PartSAM: A Scalable Promptable Part Segmentation Model Trained on Native 3D Data. In *ICLR*, 2026.