Articulated Object Reconstruction from Rest-State Observation

Daeun Lee, Jaeah Lee*, Woosung Kim*, Haebeom Jung, and Jaesik Park

Seoul National University, Republic of Korea

Fig. 1: We propose `Rest2Art`, an approach for reconstructing articulated objects with part-level geometry and joint parameters from rest-state observations alone.

Abstract. Building interactive digital twins requires recovering both 3D geometry and the kinematic structures that govern how objects articulate. Yet existing methods for articulated object reconstruction require explicitly observable motion from multiple articulation states. We introduce a *rest-state* formulation that reconstructs articulated objects from a single closed configuration, an inherently ill-posed setting where geometry, semantics, and motion priors compensate for the absence of motion cues. Our framework adopts an explicit mesh as an intermediate representation for cross-model verification and fusion, reconciling noisy outputs from vision-language and segmentation models into spatially consistent part structures. To estimate joint parameters without observed motion, we use a video diffusion model to synthesize articulation hypotheses and validate them through geometric consistency. Our approach achieves accurate part decomposition and physically plausible articulation, performing competitively with motion-observing reconstruction-based, generation-based, and modular pretrained-model baselines.

Keywords: Articulated Object · Mesh Segmentation · Digital Twin

* Indicates equal contribution

Table 1: Comparison of articulated object reconstruction methods. If *# Parts* is given as a number, it indicates that methods require prior knowledge of the part count up to that value, while *any* denotes that methods discover parts without such priors.

Framework	Task	Input			Intermediate Representation
		Type	# States	# Parts	
Ditto [16]	Recon.	PC	2	2	Neural Field
PARIS [27]	Recon.	RGBs	2	2	Neural Field
DTA [48]	Recon.	RGB-Ds	2	3	Neural Field
ArtGS [30]	Recon.	RGB(-D)s	2	any	3DGS
Real2Code [59]	Recon.	RGB(-D)s	1 (articulated)	any	Point Clouds
LARM [57]	Recon.	RGBs	2	2	Latent
REArtGS [52]	Recon.	RGBs	2	2	3DGS
REArtGS++ [51]	Recon.	RGBs	2	any	3DGS
Articulation in Motion [1]	Recon.	RGBs+video	video	any	3DGS
Articulate AnyMesh [38]	Recon.	Mesh	1 (rest)	any	Mesh
URDFormer [5]	Gen.	RGB	1 (rest)	any	-
Singapo [26]	Gen.	RGB	1 (rest)	any	-
Ours	Recon.	RGBs	1 (rest)	any	Mesh

1 Introduction

Humans exhibit a remarkable ability to infer how unfamiliar objects articulate from their appearance alone. Upon encountering a closed cabinet, we can anticipate how its doors swing or drawers slide before any interaction occurs. Yet this inference is not always reliable: we sometimes push a door that should slide or misjudge how far a drawer extends. But once we observe an object in an open state, or watch it being opened, such ambiguity largely disappears.

Existing work on articulated object reconstruction focuses precisely on this well-constrained scenario. Table 1 summarizes methods across input conditions and intermediate representations. Some methods [16,27,30,48,51,52] require observing objects in carefully selected articulation states and assume prior knowledge of the number of movable parts. Others relax some of these constraints through feed-forward prediction [57] or by employing foundation models to estimate joint parameters from a single articulated-state observation, where part displacements remain visible [59]. In these cases, however, the input must contain explicitly visible motion.

In our work, we aim to replicate "what humans routinely do" by inferring how objects articulate from their closed configurations alone. We call this the *rest-state* setting and focus on openable objects such as furniture, which exhibit diverse multi-part articulation patterns. In practice, objects are most often encountered in their closed configuration. It is also the state that can be reliably captured at scale, as it is consistently present in online product photos, internet imagery, and scene-level datasets such as ScanNet [8, 56] and Replica [42]. Our key observation is that pretrained models can supply rich semantic and motion priors for this task. A video diffusion model, for instance, can imagine how a closed drawer might open. However, individual model outputs are often noisy, and each may disagree on which parts move and how they articulate.

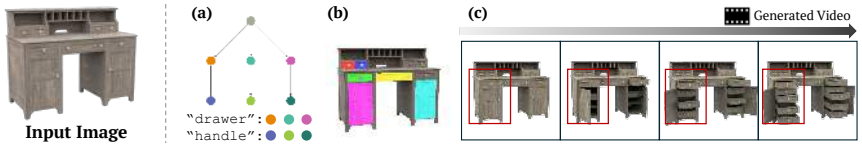

Fig. 2: Naïvely applying pretrained models to rest-state articulated objects. Given a closed cabinet with 2 doors and 7 drawers: (a) a VLM [37] predicts an incomplete part hierarchy; (b) segmentation masks [3] miss parts in certain views; (c) a video diffusion model [46] hallucinates nonexistent drawers when opening a door.

A Vision-Language Model (VLM) may predict an incomplete part hierarchy, a segmentation model may miss parts in certain views, and a video diffusion model may hallucinate nonexistent components (Fig. 2). We address this by grounding all predictions in an explicit *mesh*, whose surface connectivity enables spatially consistent part boundaries, reliable occlusion reasoning, and confidence-weighted aggregation of noisy, view-dependent predictions into a coherent 3D result. This design also makes the framework input-agnostic, as any source that produces a mesh can serve as input, including multi-view reconstruction [14, 18, 43], single-image 3D generation [45, 54], and existing 3D assets.

Based on this insight, we present Rest-to-Articulation (`Rest2Art`), a framework that reconstructs multi-part articulated objects from rest-state observations alone. We extract a surface mesh from diverse inputs and iteratively co-refine part hierarchies and segmentations using a VLM [2,37] and SAM3 [3], using their disagreements as a signal to correct each other. Validated masks are lifted onto the mesh through confidence-weighted evidence accumulation and refined via graph-based label propagation. Since no motion is observed, we synthesize plausible articulation sequences using a video diffusion model [46] to generate motion hypotheses, extract 2D trajectories, and fit rigid-body joint models by minimizing reprojection errors. The mesh geometry further constrains the joint estimation. We penalize axis configurations that cause inter-part penetration and refine the joint alignment against the mesh. Segmented patches are then completed into volumetric meshes suitable for physical simulation.

To summarize, our contributions are as follows:

- We formulate rest-state articulated object reconstruction as an ill-posed recovery problem and present a complete framework that operates from a single closed configuration without any observed motion.
- We design an iterative co-refinement loop between a Vision-Language Model and a segmentation model, where their disagreements drive mutual correction, and lift the validated evidence onto the mesh through confidence-weighted per-part aggregation.
- We propose using video diffusion models as a source of motion hypotheses and constrain the resulting joint parameters against the mesh geometry.
- We validate across diverse inputs including multi-view captures, single-image 3D generation, and existing 3D assets, demonstrating competitive joint estimation accuracy against both reconstruction and generation baselines.

2 Related Work

Articulated Object Recovery. Recovering articulated objects requires jointly estimating 3D geometry and kinematic structure across movable parts. Early approaches relied on category-specific priors or predefined kinematic templates [13, 25, 55], limiting their ability to estimate articulations of unseen objects. More recent methods adopt per-scene optimization [12, 27, 30, 48, 51, 52, 58] or incorporate foundation models for part-level understanding [22, 58]. Others leverage implicit object-centric representations [13] and interaction-based cues such as drag points [24] or interaction videos [1]. Despite this progress, most methods require multi-state observations, part annotations, or partially open configurations [59], making rest-state reconstruction particularly challenging. For single rest-state inputs, URDFormer [5] and Singapo [26] leverage 3D generative models to predict articulated geometry from a single image. Articulate AnyMesh [38] instead brings generative priors into a reconstruction framework, sequentially chaining pretrained models for open-vocabulary articulation modeling. In contrast, our approach recovers both accurate geometry and joint parameters from a single closed configuration, without prior knowledge of the number of parts, by iteratively cross-validating foundation model outputs against the reconstructed mesh rather than chaining them sequentially, and constraining joint estimation through mesh-based geometric verification.

Part Segmentation for Articulated Objects. Articulating a rest-state object requires first identifying its movable parts, which is challenging in closed configurations where uniform colors, box-like geometry, and flush boundaries leave little discriminative signal. Recent zero-shot 3D part segmentation methods lift 2D detections onto point clouds via multi-view voting [29, 33, 44, 60], train feedforward point embeddings on web-scale assets [31], or learn continuous 3D feature fields via contrastive distillation [28]. In the articulated object domain, Real2Code [59] fine-tunes SAM [21] with point prompts, but assumes an articulated-state input where displaced parts are visually distinguishable. Our method addresses this by first co-refining part hierarchies and segmentation masks through iterative cross-model verification, then fusing the validated outputs onto the mesh surface through confidence-weighted multi-view accumulation and graph-based label propagation, without part annotations or training.

3 Method

Given a rest-state mesh of an articulated object (Sec. 3.1), our pipeline proceeds in three stages. We first identify movable parts through co-refinement of hierarchy predictions and segmentation masks (Sec. 3.2). We then estimate joint parameters by synthesizing articulation videos and fitting rigid-body models to extracted trajectories (Sec. 3.3). Finally, we complete each part into a volumetric mesh for physics simulation (Sec. 3.4). An overview is shown in Fig. 3.

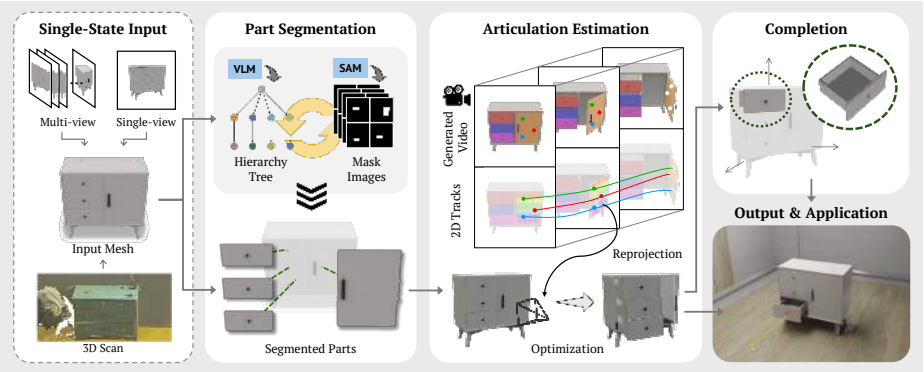

Fig. 3: Method overview. `Rest2Art` reconstructs articulated objects from rest-state inputs by co-refining part hierarchies and masks, lifting evidence onto the mesh, synthesizing videos for joint estimation, and completing interior geometry.

3.1 Problem Statement

We focus on multi-part openable objects, such as furniture, whose closed configuration can be clearly defined. Formally, given a rest-state object mesh, we output part-wise meshes $\{\mathcal{M}_k\}_{k=1}^{K}$ and joint parameters $\{\Psi_k\}_{k=1}^{K-1}$, where K is the number of parts, automatically determined without prior annotation.

We model the articulated object as a collection of rigid body parts connected by joints, where each part's motion can be represented by an SE(3) transformation. We consider *prismatic* and *revolute* joints, which cover most furniture articulation mechanisms. For a prismatic joint, we define $\Psi^p = \{\mathbf{v} \in \mathbb{R}^3, d \in \mathbb{R}\}$, where $\mathbf{v}$ represents the translation direction and d is the scalar displacement along that direction. For a revolute joint, we define $\Psi^r = \{\boldsymbol{\omega} \in \mathbb{R}^3, \mathbf{q} \in \mathbb{R}^3, \theta \in \mathbb{R}\}$, where $\boldsymbol{\omega}$ denotes the direction of the rotation axis, $\mathbf{q}$ is a point on the axis, and θ indicates the rotation angle.

3.2 Part Segmentation

Without motion cues, articulation can only be inferred from static geometry and semantics, making part segmentation the central challenge of rest-state recovery. We decouple segmentation from reconstruction by adopting a *mesh* as the intermediate 3D representation. Unlike point- or volume-based primitives, the mesh explicitly encodes surface connectivity, which enables precise occlusion reasoning and spatially consistent part boundaries, and serves as the anchor for verifying and fusing outputs from multiple pretrained models throughout this stage.

We obtain an initial mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, where $\mathcal{V}$ and $\mathcal{F}$ denote vertices and faces, respectively. Our pipeline is reconstruction-agnostic and accepts any mesh-based output, including multi-view reconstruction [14, 18, 43], single-image 3D generation [45, 54], or scene scans. The mesh is canonically aligned and re-

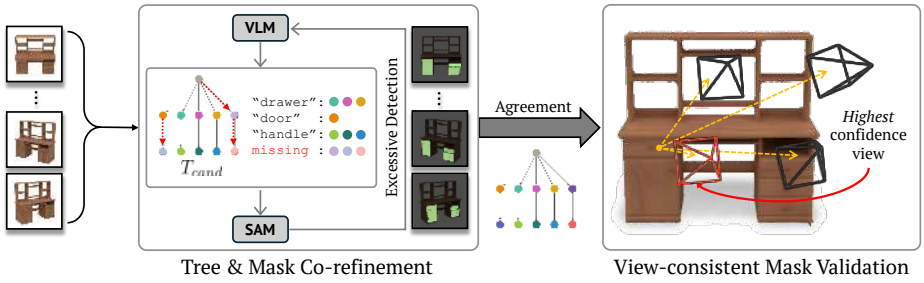

Fig. 4: Co-refinement of hierarchy and part masks. Disagreements between the VLM hierarchy tree and SAM3 masks drive mutual correction. Validated masks are then back-projected onto the mesh to enforce cross-view consistency.

rendered from N' viewpoints chosen to maximize visibility of articulated components. The view selection strategy is detailed in Sec. A.2.

Co-Refinement of Hierarchy and Part Masks. We leverage a VLM [2,37] to propose a candidate hierarchy tree T and SAM3 [3] to produce per-view part masks M_v. However, naïvely combining their outputs often fails due to missing parts, spurious detections, and view-dependent segmentation noise. We therefore introduce an iterative cross-model verification loop that uses disagreements between the two models as a signal for mutual correction.

From T, we extract C semantic prompts $\mathcal{S}(T) = \{s_1, s_2, \ldots, s_C\}$ (*e.g.*, `drawer`, `handle`, `door`), where C is the number of distinct part categories in T, along with their expected instance counts $n(s, T)$ for each $s \in \mathcal{S}(T)$. We query the segmentation model with each prompt s on every view v to obtain a set of masks $M_{v,s}$, and compare the detected count $|M_{v,s}|$ against $n(s, T)$ (Fig. 4):

- **Agreement** ($|M_{v,s}| = n(s, T)$ for all $s \in \mathcal{S}(T)$): the corresponding masks are cached as validated evidence for this view.
- **Excess detections** ($\exists\, s \in \mathcal{S}(T)$ s.t. $|M_{v,s}| > n(s, T)$): T is incomplete. We overlay the detected masks in vivid colors onto the input view, making part boundaries visually distinguishable even on textureless objects, and feed this annotated image back to the VLM to revise T. Verification then restarts under the updated tree.
- **Missing detections** ($\exists\, s \in \mathcal{S}(T)$ s.t. $|M_{v,s}| < n(s, T)$): rather than pruning T, we defer the decision and aggregate evidence across all views before concluding the part is missing.

We fix the maximum number of refinement rounds to two, which suffices in practice. To further enforce cross-view consistency, we exploit the mesh geometry to back-project high-confidence masks into 3D surface points and reproject them onto other views as point prompts for the segmentation model. A prompted mask is accepted only when it agrees with the cached mask for the same part.

Confidence-Weighted Evidence Lifting. Given the verified hierarchy T and per-view masks $\{M_v\}_{v=1}^{N'}$, we lift 2D mask evidence onto the mesh to obtain a globally consistent 3D labeling. We perform *per-part* view selection and weighting, so each part's labeling is driven by its most reliable observations.

For each part p, we rank views by a quality score that jointly considers detection confidence and mask coverage, and select the top-K views $\mathbf{V}_p$. We rasterize $\mathcal{M}$ from each selected view v using z-buffered rendering [40] with back-face culling, producing a per-pixel face index map ϕ_v. The *face support set* of mask M_p^v is defined as:

$$\mathrm{supp}(M_p^v) = \{f \in \mathcal{F} \mid \exists\, \mathbf{x} \in M_p^v \text{ s.t. } \phi_v(\mathbf{x}) = f\}. \tag{1}$$

We accumulate evidence across selected views to obtain a per-face, per-part score:

$$E(f,p) = \sum_{v \in \mathbf{V}_p} w_p^v \cdot c_p^v \cdot \mathbb{1}\big[f \in \mathrm{supp}(M_p^v)\big], \tag{2}$$

where w_p^v is the normalized view weight and c_p^v is the detection confidence. Each face is assigned the part with the maximal accumulated evidence,

$$\ell(f) = \arg\max_p \; E(f,p), \tag{3}$$

while faces receiving no support default to the base part.

Graph-Based Refinement. Lifting 2D masks onto the mesh yields a coarse part assignment that still contains three types of errors: boundary inconsistencies from conflicting multi-view evidence, spurious fragments from noisy detections, and under-segmented regions missed by all selected views. We address these sequentially by exploiting the mesh adjacency graph, which provides the structural connectivity.

We first identify reliable seed regions by computing face-level connected components for each part and retaining only the dominant component per label. The remaining disconnected fragments, which typically arise from noisy or inconsistent detections, form an unassigned pool $\mathcal{U}$.

To recover under-segmented regions and resolve fragments in $\mathcal{U}$, we propagate labels from seed regions via multi-source shortest-path computation on the vertex adjacency graph. Each unassigned vertex is assigned the label of the nearest seed part:

$$\ell(u) = \arg\min_p \; d_p(u), \quad u \in \mathcal{U}, \tag{4}$$

where $d_p(u)$ is the shortest-path distance from u to the nearest seed of part p, with ties broken by view quality score. This propagation respects surface topology, expanding parts along the mesh rather than across geometric discontinuities.

Finally, we verify each propagated label by projecting the relabeled vertex back to its part's best-scoring view, accepting the label only if the projection falls within the corresponding 2D mask. Vertices that fail this cross-modal check are reassigned to the next nearest part, ensuring that the final labeling is supported by both geometric proximity and 2D evidence.

3.3 Articulation Estimation

Given the segmented parts and their hierarchy tree T, we estimate joint parameters for each articulated part. In the rest-state setting, no motion is directly observed. We address this by synthesizing articulation videos through a video diffusion model and recovering joint parameters from the observed 2D trajectories. Crucially, the synthesized motion serves only as a *source of hypotheses*; the final joint parameters are determined by geometric consistency, not by the fidelity of the generated video.

Articulation Video Synthesis. We overlay the resulting part masks with transparency onto the view with the highest average quality score across parts, explicitly highlighting all movable parts in a single conditioning image for the video model. We employ Wan2.2 [46] with a LoRA weight from VBVR [47], which encourages spatiotemporal reasoning over physical properties such as continuity and causality. A VLM [2] generates a text prompt describing the expected articulation. Prompt construction details are provided in Sec. A.4.

From the synthesized video, we extract dense 2D point trajectories using CoTracker3 [17], initializing tracking points on each part's mask region in the first frame. The mesh provides metric depth at each tracked pixel via rendering, allowing us to unproject the frame-0 positions into 3D camera-frame coordinates as rest-pose anchor points $\{\mathbf{p}_0^j\}_{j=1}^{N_{\text{tr}}}$, where N_{tr} is the number of tracked points.

Although the generated video may not be geometrically exact, we only require that the *type* and *direction* of motion are approximately correct, since the final joint parameters are fitted against the 3D mesh rather than the video itself. Extracting point tracks and fitting rigid-body joint models is well established for real motion observations [10, 49]; our setting differs in that no real motion exists, so the video diffusion model must *imagine* plausible articulation while the mesh provides geometric grounding to validate it.

We identify the reliable portion of the synthesized video by truncating tracks at the first frame where the fraction of visible tracks falls below a threshold. This addresses two issues that both manifest as falling visibility: temporal degradation in generated video, where errors compound over frames and motion becomes implausible [46], and accumulated tracker drift.

Motion Type Classification. Before fitting parametric models, we classify each part's motion as revolute, prismatic, or static using only the 2D tracks. We compute two cues from the truncated track sequence. First, the coefficient of variation (CV) of per-point displacement magnitudes: revolute motion produces high CV because points at different radii from the rotation axis travel different distances, whereas prismatic motion yields low CV as all points undergo the same translation. Second, a curvature score that measures the directional change of displacement vectors over time: revolute tracks curve while prismatic tracks remain straight. Parts with negligible displacement are classified as static. When both cues agree, the classification is used directly. In ambiguous cases, we fit both models and select the one with lower reprojection error.

Joint Parameter Fitting. We formulate joint estimation as minimizing the 2D reprojection error between the motion-model prediction and the observed tracks. For a revolute joint, each rest-pose anchor point $\mathbf{p}_0$ is rotated about an axis $(\hat{\boldsymbol{\omega}}, \mathbf{q})$ by a per-frame angle θ_t:

$$\mathbf{p}_t = R(\hat{\boldsymbol{\omega}},\, \theta_t)\,(\mathbf{p}_0 - \mathbf{q}) + \mathbf{q}, \tag{5}$$

where $R(\hat{\boldsymbol{\omega}}, \theta_t)$ denotes the Rodrigues rotation matrix. For a prismatic joint, the model simplifies to a per-frame scalar displacement d_t along a unit direction $\hat{\mathbf{v}}$:

$$\mathbf{p}_t = \mathbf{p}_0 + d_t\,\hat{\mathbf{v}}. \tag{6}$$

In both cases, the predicted 3D positions are projected back to 2D using known camera intrinsics, and the parameters are optimized with a robust loss that down-weights outlier correspondences, followed by a trimmed refit.

The fitted revolute axis, however, inherits ambiguity from monocular depth: multiple 3D rotation axes produce identical 2D projections, making the true rotation direction underdetermined from tracks alone. We resolve this using the segmented mesh by penalizing axes whose rotation drives the child part into the parent. We further refine $\hat{\boldsymbol{\omega}}$ by softly blending it with the principal direction of the adjacent part boundary, which encodes the physical articulation structure.

To improve robustness against video generation artifacts, we synthesize two articulation videos with different random seeds and fit joint parameters from each independently. When both fits agree on the motion type, we combine their trajectory evidence and refit jointly, yielding more stable estimates.

3.4 Mesh Completion

The segmented parts are shell-like patches, but physics simulators and URDF export require volumetric solids. We convert each part into a closed mesh by offsetting vertices inward along their normals. Since vertex normals near cut edges are unreliable, we apply boundary-aware normal smoothing to avoid artifacts.

While this solidification suffices for exterior-dominant parts such as doors and knobs, components like drawers require explicit interior geometry, as rest-state inputs limit visibility of interior structures. For these, we generate a parametric inner tray and blend it with the solidified exterior. Details of both procedures are provided in Sec. A.6.

4 Experiments

We conduct experiments to validate our rest-state modeling approach from multiple perspectives. We evaluate part segmentation quality against existing baselines (Sec. 4.2), compare joint estimation accuracy with single-state and two-state methods (Sec. 4.3), validate our design choices through ablations (Sec. 4.4), and demonstrate generalization to diverse inputs (Sec. 4.5).

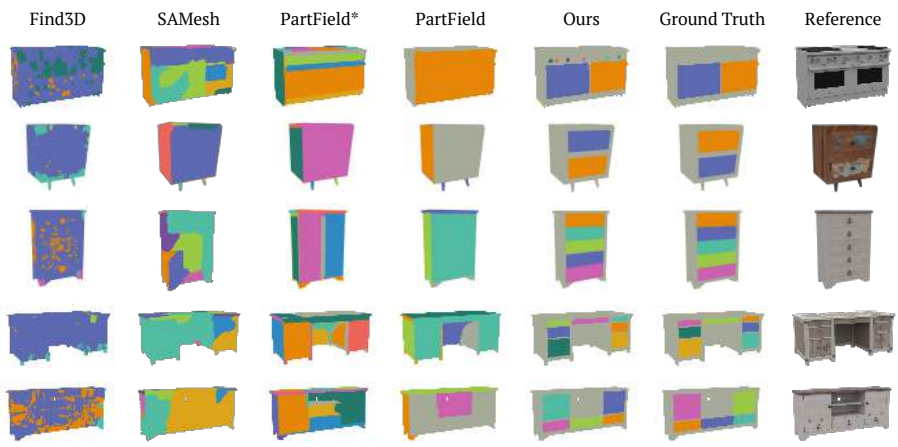

Fig. 5: Qualitative comparison of part segmentation on ACD dataset [15]. PartField* uses a fixed 20-cluster setting, while PartField uses the ground-truth part count. Our method produces more accurate part boundaries and segments all interactable parts, including knobs (top row) missing from ground-truth annotations.

4.1 Experimental Details

Dataset. We evaluate on Articulated Containers Dataset (ACD) [15], a benchmark of articulated furniture objects from Habitat Synthetic Scenes Dataset (HSSD) [19] and Amazon Berkeley Objects (ABO) [6]. All objects are rendered in closed configuration. To assess real-world robustness, we additionally evaluate joint estimation on the MultiScan Dataset [32]. We also validate on web images and scanned meshes to evaluate generalization across input modalities.

Implementation Details. For evaluation, we reconstruct meshes from multi-view renderings using 2DGS [14], which preserves metric scale consistent with ground truth. Results with alternative mesh sources are presented in Tab. 5. We use GPT-5.2 [37] for hierarchy prediction and SAM3 [3] for mask generation. For articulation estimation, we generate videos using Wan2.2 [46] with VBVR LoRA [47] and extract 2D trajectories using CoTracker3 [17].

4.2 Evaluation of Part Segmentation

Baselines. We compare with feed-forward 3D segmentation methods PartField [28], Find3D [31], and the zero-shot mesh segmentation method SAMesh [44] on the ACD-HSSD dataset [15, 19]. PartField requires the number of parts as input. We evaluate in two settings, one using the ground-truth part count and another with a fixed count of 20 to examine whether over-segmentation can capture functionally meaningful boundaries. For Find3D, which requires text queries, we use the same prompts as our method for fair comparison. Since Find3D operates

Table 2: Joint estimation comparison on the ACD dataset [15]. Ang., Pos., and Type denote axis angular error (°), position error (dm), and type accuracy (%), respectively. ⟳ denotes our co-refinement strategy, where the co-refined hierarchy tree is used as input. †REArtGS requires a single ground-truth joint type per object as input. Cell highlights denote ranks: best, second, third.

	Setting		ACD-HSSD [19]			ACD-ABO [6]		
Method	Task	State	Ang.↓	Pos.↓	Type↑	Ang.↓	Pos.↓	Type↑
URDFormer [5]	Gen.	rest-state	14.59	6.52	16.23	13.64	6.11	19.12
Singapo [26]	Gen.	rest-state	1.97	3.21	59.42	0.13	2.63	41.47
Singapo [26] + ⟳	Gen.	rest-state	2.24	2.85	61.21	0.14	2.33	67.94
ArtGS [30]	Recon.	two states	53.12	1.37	57.71	46.29	0.96	60.76
REArtGS† [52]	Recon.	two states	58.02	6.95	33.30	59.20	6.77	35.61
REArtGS++ [51]	Recon.	two states	41.82	2.87	59.38	45.92	1.23	81.28
Articulation in Motion [1]	Recon.	video	30.29	N/A	25.09	19.74	0.81	41.90
Articulate AnyMesh [38]	Recon.	rest-state	11.35	1.31	67.12	25.65	0.08	38.94
Ours	Recon.	rest-state	11.35	0.85	74.49	4.78	0.20	73.38

on point clouds, we sample 5k points from each mesh and transfer predicted labels back to mesh faces following [28]. Our method uses the same configuration across all categories and datasets without any training or object-specific tuning.

Qualitative Results. As shown in Fig. 5, baseline methods struggle on rest-state furniture, where uniform colors, repetitive patterns, and flush boundaries provide minimal discriminative signal for approaches that rely on geometric or visual features alone. Our method avoids this failure mode by design. The co-refinement loop ensures part counts and masks are cross-validated between the VLM and segmentation model before any evidence is committed to the mesh. The confidence-weighted lifting then discards unreliable views rather than averaging them. Quantitative results are provided in Sec. C.3.

4.3 Evaluation of Joint Estimation

Baselines. We compare against reconstruction-based baselines. Among them, only Articulate AnyMesh [38] shares our rest-state setup, operating directly on our 2DGS-reconstructed mesh. ArtGS [30], REArtGS [52], and REArtGS++ [51] are two-state methods that leverage explicit motion observations from paired multi-view images of distinct articulation states. We synthesize such images by selecting the last physically plausible frame from a generated video, producing multi-view articulated-state images via Qwen-Image-Edit [50], estimating camera parameters with MASt3R-SfM [11,23], and filtering out low-confidence views. Articulation in Motion [1] additionally requires an interaction video, for which we use a video generated by the same VDM. Details are provided in Sec. B.1.

We additionally compare with URDFormer [5] and Singapo [26], generation-based approaches that can be directly applied in the rest-state setting without

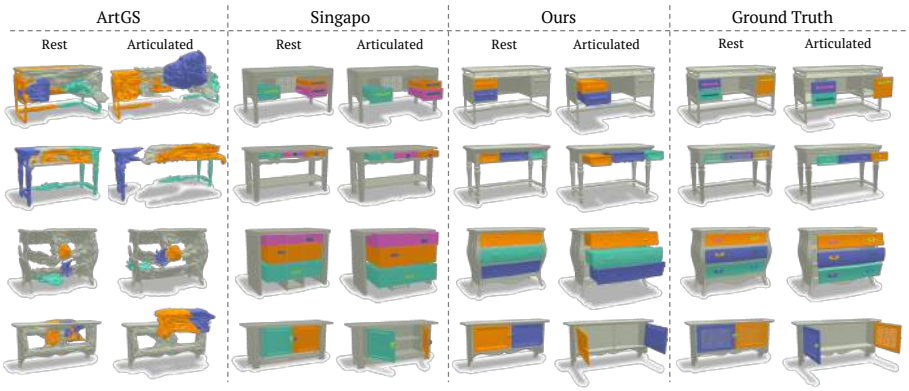

Fig. 6: Qualitative comparison of joint estimation on ACD dataset [15]. Each example shows the predicted rest and articulated states.

requiring motion observations. Since both methods produce articulated structures at a normalized scale, we estimate a scale factor for each predicted axis to align it with the ground-truth geometry before evaluation. Following prior work [27, 30, 48], we report axis angular error (Axis Ang.), axis positional error (Axis Pos.), and joint type accuracy (Type Acc.). Axis angular and positional errors are computed only for parts where the predicted joint type is correct.

Results on Synthetic Data. We report the quantitative (Table 2) and qualitative (Fig. 6) comparison. Articulate AnyMesh [38], which shares our rest-state setup, applies pretrained models sequentially per module and is therefore sensitive to noisy individual predictions, causing it to struggle on objects with subtle articulation boundaries. ArtGS [30], REArtGS [52], and REArtGS++ [51] struggle with the synthesized multi-state input, as reconstruction-based two-state methods require precise geometric consistency between states. In particular, even the static base part is often detected as moving due to slight misalignment between the synthesized states, causing the entire reconstruction to collapse. Articulation in Motion [1], designed for real interaction videos, exhibits a similar mismatch with our synthesized video input. This highlights that rest-state recovery requires robustness to noisy intermediate outputs, whether from pretrained models or motion estimates, which our framework achieves by grounding all evidence in the mesh geometry.

The generative methods benefit from producing structures in the canonical space with axis-aligned joint parameters, yielding low angular error after alignment. This advantage, however, comes at the cost of metric fidelity, as their outputs require post-hoc scale alignment for meaningful evaluation.

Our method achieves the highest type accuracy among methods directly applicable to rest-state inputs on both datasets, and competitive axis estimation overall, despite recovering joint parameters directly from optimization on generated trajectories without axis-aligned priors. This demonstrates that the combi-

Table 3: Joint estimation on the MultiScan dataset [32]. Cov. (%) denotes the percentage of evaluable scenes, over which Ang., Pos., and Type are computed. Remaining metrics and cell highlights follow Tab. 2.

Method	Cov.↑	Ang.↓	Pos.↓	Type↑
REArtGS† [52]	31.91	51.93	29.60	64.40
REArtGS++ [51]	36.17	62.72	6.85	48.74
Articulate AnyMesh [38]	61.70	30.51	3.45	12.63
Ours	48.94	17.23	0.26	65.78

Table 4: Ablation on model selection for co-refinement(↻) on ACD dataset. $\mathcal{T}$ Acc. and # Parts Acc. denote hierarchy tree and per-part count accuracy(%). † denotes the use of ground-truth part labels for text grounding.

Method	Co-Refine	$\mathcal{T}$ Acc.↑	# Parts Acc.↑
GPT-5.2 [37]	–	62.3%	59.4%
Qwen3-VL [2]	–	52.2%	47.8%
SAM3† [3]	–	–	82.6%
Grounded-SAM2† [39, 41]	–	–	43.5%
GPT-5.2 + SAM3	✗	62.3	59.4
Qwen3-VL + SAM3	✗	52.2	47.8
Qwen3-VL + Grounded-SAM2	✗	53.6	46.4
GPT-5.2 ↻ SAM3 (Ours)	✓	72.5 (+10.2)	59.4 (+0.0)
Qwen3-VL ↻ SAM3 (Ours)	✓	72.5 (+20.3)	63.8 (+16.0)
Qwen3-VL ↻ Grounded-SAM2 (Ours)	✓	60.9 (+7.3)	47.8 (+1.4)

nation of video-based hypothesis sampling and mesh-grounded geometric fitting provides a viable path for rest-state articulation estimation at metric scale.

Results on Real-World Data. We further evaluate joint estimation on MultiScan [32], which provides real-world RGB-D scans of indoor scenes with articulated furniture. We preprocess scene-level annotations into per-instance articulated objects with their RGB views; details are provided in Sec. B.2. Since MultiScan naturally provides two-state captures per object, we use them directly as input to two-state reconstruction baselines [51, 52]. For single-state methods, including Articulate AnyMesh [38] and ours, we use both available states as input and report the result with the lower reprojection error.

Despite noisy real-world inputs, our method recovers relatively accurate joints (Table 3), outperforming reconstruction baselines on type accuracy and angular, and thereby demonstrating the robustness of our mesh-grounded formulation under real-world noise.

4.4 Ablation Study

Co-Refinement Model Selection. Table 4 validates our co-refinement loop on the ACD dataset [6, 15, 19], reporting hierarchy tree accuracy and per-part

Table 5: Ablation on mesh reconstruction methods.

	Segmentation		Joint Estimation	
	IoU ↑	mAP ↑	Axis Dir ↓	Axis Pos ↓
3DGS [18]	0.30	0.83	1.69	0.154
SVRaster [43]	0.83	0.96	1.86	0.146
2DGS [14]	0.97	0.99	3.67	0.154

Fig. 7: Qualitative ablation on mesh reconstruction methods.

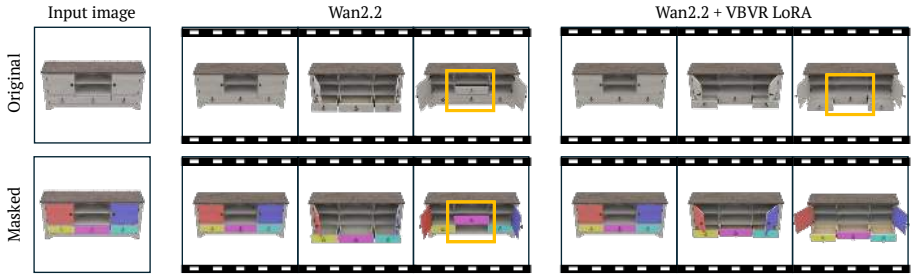

Fig. 8: Qualitative ablation on video generation. We compare Wan2.2 [46] with and without VBVR LoRA [47], conditioned on the original image (top) and our mask overlay (bottom). Yellow boxes highlight implausible motion.

count accuracy. A VLM [2,37] alone predicts plausible hierarchies but frequently miscounts parts, as rest-state objects offer minimal visual cues for distinguishing individual components. A segmentation model [3,39,41] alone discovers parts but cannot infer hierarchical relationships without semantic grounding. Combining both in the iterative verification loop yields the best performance on both metrics, confirming that the two models provide complementary signals that are only fully realized through mutual correction. Additional co-refinement analysis is provided in Sec. C.1.

Mesh Reconstruction Backend. Table 5 compares different reconstruction backends, measuring both segmentation quality and joint estimation accuracy. For IoU computation between meshes with different topologies, we sample 100k surface points per part and measure overlap based on nearest-neighbor distances.

Video Generation Design. We ablate two design choices in our articulation video synthesis (Fig. 8). Wan2.2 [46] alone frequently generates implausible motion such as deforming or hallucinating parts. VBVR LoRA [47] improves plausibility, but mask overlay is necessary to guide the model toward parts with subtle boundaries, reducing hallucination from 23.1% to 11.5% and raising the accurate articulation rate from 69.2% to 80.8% on ACD-HSSD [19]. Additional analysis of video generation behavior is provided in Sec. C.2.

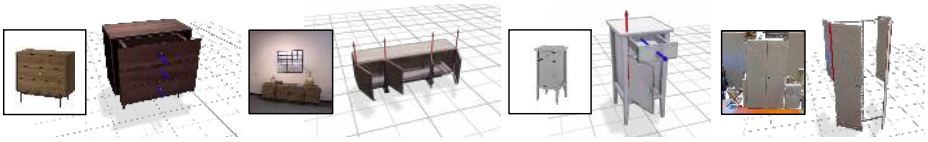

Fig. 9: Generalization to Diverse Inputs. Our method applies to single-image inputs and scanned meshes, in addition to multi-view captures.

4.5 Generalization to Diverse Inputs

Since our framework is input-agnostic, it extends beyond multi-view captures. For single-image inputs, we replace the multi-view reconstruction stage with an image-to-3D generative model [45, 54]; for scanned meshes, the pipeline applies directly without modification. Figure 9 presents results on online product photos, an indoor scene from Replica [42], and scanned meshes from ScanNet++ [56], demonstrating that our formulation generalizes across diverse inputs and real-world imagery. Additional results are provided in Secs. D.1 and D.2.

5 Conclusion

We present `Rest2Art`, a framework that reconstructs articulated objects from rest-state observations alone by cross-validating pretrained model outputs against explicit mesh geometry. This demonstrates that models with inconsistent outputs can yield robust results through structured geometric verification. Our reconstruction-based formulation preserves metric scale and geometric fidelity, producing digital replicas with physically meaningful joint parameters for applications such as interactive simulation and spatial planning. Extending to more complex kinematic structures and improving the physical plausibility of video generation models are promising future directions.

Acknowledgements

This work was supported by IITP grants (RS-2025-25442338: AI star Fellowship Support Program at SNU (50%); RS-2025-02303703: Realworld multi-space fusion and 6DoF free-viewpoint immersive visualization for extended reality (45%); RS-2021-II211343: AI Graduate School Program at SNU (5%)) funded by the Korea government (MSIT). We thank Joonghyuk Shin, Minkyun Seo, Jinmo Kim, Youngjoong Kim, Minji Seo, Seunguk Do, and Eun Sun Lee for helpful advice and discussions during the paper.

References

1. Ai, H., Chang, W., Jiao, J., Leonardis, A., Ofek, E.: Articulation in motion: Prior-free part mobility analysis for articulated objects by dynamic-static disentanglement. In: Int. Conf. Learn. Represent. (2026)
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
3. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Dollár, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: SAM 3: Segment anything with concepts. In: Int. Conf. Learn. Represent. (2026), https://openreview.net/forum?id=r35clVtGzw
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)
5. Chen, Z., Walsman, A., Memmel, M., Mo, K., Fang, A., Vemuri, K., Wu, A., Fox, D., Gupta, A.: Urdformer: A pipeline for constructing articulated simulation environments from real-world images. In: Proceedings of Robotics: Science and Systems (2024)
6. Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Vicente, T.F.Y., Dideriksen, T., Arora, H., et al.: ABO: Dataset and benchmarks for real-world 3D object understanding. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 21126–21136 (2022)
7. Contributors, L.: Lightx2v: Light video generation inference framework. https://github.com/ModelTC/lightx2v (2025)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5828–5839 (2017)
9. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3D objects. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 13142–13153 (2023)
10. Dharmarajan, K., Huang, W., Wu, J., Fei-Fei, L., Zhang, R.: Dream2flow: Bridging video generation and open-world manipulation with 3d object flow. In: Int. Conf. on Robotics and Automation (ICRA) (2026)
11. Duisterhof, B.P., Zust, L., Weinzaepfel, P., Leroy, V., Cabon, Y., Revaud, J.: Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In: International Conference on 3D Vision (3DV). pp. 1–10. IEEE (2025)
12. Guo, J., Xin, Y., Liu, G., Xu, K., Liu, L., Hu, R.: Articulatedgs: Self-supervised digital twin modeling of articulated objects using 3d gaussian splatting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 27144–27153 (2025)

13. Heppert, N., Irshad, M.Z., Zakharov, S., Liu, K., Ambrus, R.A., Bohg, J., Valada, A., Kollar, T.: Carto: Category and joint agnostic reconstruction of articulated objects. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 21201–21210 (2023)
14. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: SIGGRAPH Conference Papers. Association for Computing Machinery (2024). https://doi.org/10.1145/3641519.3657428
15. Iliash, D., Jiang, H., Zhang, Y., Savva, M., Chang, A.X.: S2o: Static to openable enhancement for articulated 3d objects. In: IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6785–6795 (2026)
16. Jiang, Z., Hsu, C.C., Zhu, Y.: Ditto: Building digital twins of articulated objects from interaction. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5616–5626 (2022)
17. Karaev, N., Makarov, Y., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In: Int. Conf. Comput. Vis. pp. 6013–6022 (2025)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
19. Khanna*, M., Mao*, Y., Jiang, H., Haresh, S., Shacklett, B., Batra, D., Clegg, A., Undersander, E., Chang, A.X., Savva, M.: Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16384–16393 (June 2024)
20. Kijai: Vbvr lora for wan2.2 i2v. https://huggingface.co/Kijai/WanVideo_comfy (2025), accessed: 2026-03-08
21. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. In: Int. Conf. Comput. Vis. pp. 4015–4026 (2023)
22. Le, L., Xie, J., Liang, W., Wang, H.J., Yang, Y., Ma, Y.J., Vedder, K., Krishna, A., Jayaraman, D., Eaton, E.: Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In: Int. Conf. Learn. Represent. vol. 2025, pp. 17578–17602 (2025)
23. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: Eur. Conf. Comput. Vis. pp. 71–91. Springer (2024)
24. Li, R., Zheng, C., Rupprecht, C., Vedaldi, A.: Dragapart: Learning a part-level motion prior for articulated objects. In: Eur. Conf. Comput. Vis. pp. 165–183. Springer (2024)
25. Li, X., Wang, H., Yi, L., Guibas, L.J., Abbott, A.L., Song, S.: Category-level articulated object pose estimation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3706–3715 (2020)
26. Liu, J., Iliash, D., Chang, A., Savva, M., Mahdavi Amiri, A.: Singapo: Single image controlled generation of articulated parts in objects. In: Int. Conf. Learn. Represent. vol. 2025, pp. 97511–97532 (2025)
27. Liu, J., Mahdavi-Amiri, A., Savva, M.: PARIS: Part-level reconstruction and motion analysis for articulated objects. In: Int. Conf. Comput. Vis. pp. 352–363 (2023)
28. Liu, M., Uy, M.A., Xiang, D., Su, H., Fidler, S., Sharp, N., Gao, J.: Partfield: Learning 3d feature fields for part segmentation and beyond. In: Int. Conf. Comput. Vis. pp. 9704–9715 (2025)
29. Liu, M., Zhu, Y., Cai, H., Han, S., Ling, Z., Porikli, F., Su, H.: Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 21736–21746 (2023)

30. Liu, Y., Jia, B., Lu, R., Ni, J., Zhu, S.C., Huang, S.: Building interactable replicas of complex articulated objects via gaussian splatting. In: Int. Conf. Learn. Represent. (2025)
31. Ma, Z., Yue, Y., Gkioxari, G.: Find any part in 3d. In: Int. Conf. Comput. Vis. pp. 7818–7827 (2025)
32. Mao, Y., Zhang, Y., Jiang, H., Chang, A., Savva, M.: MultiScan: Scalable RGBD scanning for 3D environments with articulated objects. In: Adv. Neural Inform. Process. Syst. vol. 35, pp. 9058–9071 (2022)
33. Michele, B., Boulch, A., Puy, G., Bucher, M., Marlet, R.: Generative zero-shot learning for semantic segmentation of 3d point clouds. In: Int. Conf. on 3D Vision (3DV). pp. 992–1002. IEEE (2021)
34. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 909–918 (2019)
35. NVIDIA: Isaac Sim, `https://github.com/isaac-sim/IsaacSim`
36. Odin, L.: Qwen-image-edit-2511 multiple angles lora. `https://huggingface.co/fal/Qwen-Image-Edit-2511-Multiple-Angles-LoRA` (2025), accessed: 2026-03-08
37. OpenAI: GPT-5 system card. `https://cdn.openai.com/gpt-5-system-card.pdf` (2025), accessed: 2026-05-20
38. Qiu, X., Yang, J., Wang, Y., Chen, Z., Wang, Y., Wang, T.H., Xian, Z., Gan, C.: Articulate AnyMesh: Open-vocabulary 3D articulated objects modeling. In: Conference on Robot Learning (CoRL) (2025)
39. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: SAM 2: Segment anything in images and videos. In: Int. Conf. Learn. Represent. vol. 2025, pp. 28085–28128 (2025)
40. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv preprint arXiv:2007.08501 (2020)
41. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded sam: Assembling open-world models for diverse visual tasks (2024)
42. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
43. Sun, C., Choe, J., Loop, C., Ma, W.C., Wang, Y.C.F.: Sparse voxels rasterization: Real-time high-fidelity radiance field rendering. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16187–16196 (2025)
44. Tang, G., Zhao, W., Ford, L., Benhaim, D., Zhang, P.: Segment any mesh. arXiv preprint arXiv:2408.13679 (2024)
45. Team, S.D., Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., Lin, A., Liu, J., Ma, Z., Sagar, A., Song, B., Wang, X., Yang, J., Zhang, B., Dollár, P., Gkioxari, G., Feiszli, M., Malik, J.: Sam 3d: 3dfy anything in images. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7220–7232 (2026)

46. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
47. Wang, M., Wang, R., Lin, J., Ji, R., Wiedemer, T., Gao, Q., Luo, D., Qian, Y., Huang, L., Hong, Z., et al.: A very big video reasoning suite. arXiv preprint arXiv:2602.20159 (2026)
48. Weng, Y., Wen, B., Tremblay, J., Blukis, V., Fox, D., Guibas, L., Birchfield, S.: Neural implicit representation for building digital twins of unknown articulated objects. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3141–3150 (2024)
49. Werby, A., Büchner, M., Röfer, A., Huang, C., Burgard, W., Valada, A.: Articulated object estimation in the wild. In: Conference on Robot Learning (CoRL). vol. 2 (2025)
50. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), https://arxiv.org/abs/2508.02324
51. Wu, D., Liu, L., Huang, A., Liu, Y., Yu, Q., Liu, S., Song, L., Lu, C.: ReArtGS++: Generalizable articulation reconstruction with temporal geometry constraint via planar gaussian splatting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1177–1186 (2026)
52. Wu, D., Liu, L., Linli, Z., Huang, A., Song, L., Yu, Q., Wu, Q., Lu, C.: ReArtGS: Reconstructing and generating articulated objects via 3D gaussian splatting with geometric and motion constraints. In: Adv. Neural Inform. Process. Syst. vol. 38, pp. 102889–102915 (2026)
53. Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., Yi, L., Chang, A.X., Guibas, L.J., Su, H.: SAPIEN: A simulated part-based interactive environment. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11097–11107 (June 2020)
54. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 21469–21480 (2025)
55. Xue, H., Liu, L., Xu, W., Fu, H., Lu, C.: Omad: Object model with articulated deformations for pose estimation and retrieval. In: Brit. Mach. Vis. Conf. (2021)
56. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Int. Conf. Comput. Vis. pp. 12–22 (2023)
57. Yuan, S., Shi, R., Wei, X., Zhang, X., Su, H., Liu, M.: Larm: A large articulated object reconstruction model. In: SIGGRAPH Asia Conference Papers. pp. 1–12 (2025)
58. Zhang, C., Lee, G.H.: Iaao: Interactive affordance learning for articulated objects in 3d environments. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 12132–12142 (2025)

59. Zhao, M., Weng, Y., Bauer, D., Song, S.: Real2code: Reconstruct articulated objects via code generation. In: Int. Conf. Learn. Represent. vol. 2025, pp. 668–686 (2025)
60. Zhou, Y., Gu, J., Li, X., Liu, M., Fang, Y., Su, H.: Partslip++: Enhancing low-shot 3d part segmentation via multi-view instance segmentation and maximum likelihood estimation. Int. Conf. Comput. Vis. Worksh. (2025)

A Implementation Details

This section provides additional implementation details for each stage of `Rest2Art`.

- Section A.1 lists the key implementation parameters.
- Section A.2 describes the view selection and canonical alignment strategy.
- Section A.3 details the hierarchy tree construction and co-refinement procedure.
- Section A.4 describes the video generation model and prompting strategy.
- Section A.5 details track extraction and visibility-gated filtering.
- Section A.6 describes the mesh completion and solidification procedure.
- Section A.7 reports runtime and memory analysis.

A.1 Key Parameters

Table A lists the key implementation parameters.

Table A: Implementation parameters of `Rest2Art`. Camera parameters apply only when the input is a standalone mesh; for NVS-based inputs, intrinsics are inherited from training cameras and only extrinsics follow the placement rules in Sec. A.2.

Param.	Description	Value
View Selection		
N'	Total re-rendered views	12
$R_{\max}$	Max co-refinement rounds	2
Camera (mesh input only)		
–	Rendering resolution	800×800
–	Field of view	35.49
–	Camera radius	4.5
–	Azimuth range from frontal ($-y$)	±30
–	Elevation range	15–45
Articulation Estimation		
R	Number of video generation seeds	2
τ_{vis}	Visibility threshold for track truncation	0.5

A.2 View Selection Strategy

We first align the mesh's oriented bounding box (OBB) to a Blender-convention coordinate system ($+z$ up, $-y$ frontal), then re-render it from N' viewpoints centered around the frontal direction to maximize visibility of articulated components. When the mesh is produced by a Novel View Synthesis (NVS)-based pipeline (*e.g.*, 2DGS [14], 3DGS [18], SVRaster [43]), we reuse the same renderer. For standalone mesh inputs, we use Pytorch3D renderer [40] and apply the fixed camera configuration in Tab. A.

After obtaining per-view masks $\{M_{v,s}\}$, each view v is scored per part p by detection confidence weighted by log mask area, preventing views with disproportionately large masks from dominating. For parent parts, the score is further boosted by the fraction of child parts visible in that view, encouraging selection of views that expose the full part hierarchy. The normalized view weight w_p^v in Eq. (2) is the ratio of each view's score to the best-scoring view for part p.

A.3 Hierarchy Tree Construction

We query GPT-5.2 [37] with all N' re-rendered views and select the prediction with the highest part count as the initial hierarchy tree T, following the hierarchy graph prediction approach of Singapo [26]. We use temperature 0.4 and 6-shot in-context examples. During co-refinement, the same prompt template is reused with the mask overlay image, so the model can visually observe the segmented parts and revise its count accordingly. Parts absent from the final segmentation are pruned from T at the end of co-refinement. The system prompt is as follows:

> You are an expert in the recognition of articulated parts of an object in an image. You will be provided with an image of an articulated object.
>
> You should follow the following steps to achieve the task:
> (1) Recognize all the articulated parts of the object from the set {base, door, knob, handle, drawer, tray}. There must always be exactly one base. Trays can only exist in microwaves. Each handle must be attached to either a door or a drawer. Each door can have at most two handles.
>
> To distinguish between a door and a drawer, reason carefully about handle placement, panel proportion, and likely motion affordance: doors provide access to internal compartments and may be hinged or sliding; drawer fronts are box-like units that slide outward. Do not rely on a single rigid rule; use overall geometry, proportion, and functional affordance.
>
> (2) Describe how the parts are connected and organize them into a part connectivity graph. The base must always be the root. All other parts must be attached hierarchically under the base or another articulated part.
>
> Output format: Include a valid JSON object within "'json ... "' fences with exactly one root key "base". Output only the textual description followed by the JSON block.

A.4 Video Generation Details

Articulation Video Synthesis. We generate articulation videos using Wan2.2-I2V-A14B [46] with VBVR LoRA [20,47] in an image-to-video (I2V) setting. The conditioning image is a composite of the front-view rendering with part masks

blended in distinct colors ($\alpha = 0.35$). We produce 81 frames at 960×960 resolution, which are subsequently downsampled to match the rendering resolution. Inference is accelerated via LightX2V [7] using a 4-step distilled schedule with guidance scale 3.5, with three LoRA adapters applied at strength 1.0 across high- and low-noise phases. We synthesize $R = 2$ videos with different random seeds.

Prompt Generation. We use Qwen3-VL [2] to generate a text prompt from the overlay image using the following instruction template. A fixed suffix `"No camera movement. Camera locked."` is appended to suppress camera drift.

> You are an expert in articulated object reasoning and motion-aware prompt generation. Given a single image of an object, write ONE WAN2.2 video-generation prompt that forces ALL articulated parts to move simultaneously while the camera remains completely fixed.
>
> *Requirements:*
> (1) The camera is strictly locked for the entire clip: fixed intrinsics and extrinsics, no pan/tilt/roll/zoom/dolly, no stabilization drift, no parallax, no reframing.
> (2) All articulated parts move at the same time: drawers slide outward together, doors/panels swing open together, handles move only with their attached parts. Motions are synchronized, smooth, mechanically plausible, moderate range, gravity-consistent, and collision-free.
> (3) Preserve exact object identity and geometry; do not add, remove, or duplicate parts. Keep textures, patterns, background, and lighting consistent across frames.
> (4) Photorealistic, soft studio lighting, stable geometry, cinematic clarity.
>
> Do NOT mention segmentation masks, colors, or annotations.
> Output only the final WAN2.2 prompt.

Multi-Seed Consensus. Each seed is processed independently through tracking and joint fitting. For each part, the best seed is selected by a score combining median displacement and reprojection error. If a part is classified as static across all R seeds, two additional videos are generated as a fallback to recover missed articulations. A detailed breakdown of video generation outcomes and failure modes is provided in Sec. C.2 and Sec. C.4.

A.5 Track Extraction and Filtering

Tracking points are initialized on a dense grid within the eroded part mask of the first frame, where mask erosion (2% of the bounding box short side) removes unreliable boundary points. CoTracker3 [17] performs bidirectional tracking from frame 0. Tracks are truncated at the first frame where the visible fraction drops below $\tau_{\text{vis}} = 0.5$ relative to frame 0, jointly filtering degraded video frames and accumulated tracker drift.

A.6 Mesh Completion Details

The interior geometry of articulated objects is inherently unobservable from rest-state images, making true reconstruction ill-posed. Rather than inferring hidden geometry, we focus on volumetrizing each part into a watertight solid suitable for physical simulation and robotic manipulation. We complete each open part surface through two steps: solidification and, for drawer parts, interior tray generation.

Solidification. We offset the part surface inward by a base thickness of $\delta = 0.015\,\mathrm{m}$ to generate an inner surface, then connect boundary edges with side walls to produce a closed solid. To handle unreliable normals near cut edges, we apply boundary-aware normal smoothing that progressively blends normals within a few hops of the boundary before offsetting. To avoid interpenetration between adjacent parts, the per-vertex thickness is adaptively clamped to $\min(\delta,\ 0.45 \cdot d_i)$, where d_i is the distance to the nearest vertex of a neighboring part, with a minimum of 0.5 mm. Parts that are already watertight are left unchanged.

Drawer Interior Generation. For drawer parts, we additionally generate an interior tray to produce a geometrically plausible hollow structure. The tray is extruded inward (opposite to the front face normal) with each dimension scaled to $0.75\times$ the corresponding drawer extent, where depth is measured as the available space d_{avail} between the drawer back face and the base part boundary. Wall thickness is set equal to δ.

A.7 Runtime and Memory Analysis

We report per-stage runtimes and GPU memory usage measured on the ACD dataset [15] using a single NVIDIA H200 GPU. Table B summarizes the mean runtime per stage, and Fig. A reports the per-stage GPU memory profile. Each entry includes the model loading time. Mesh reconstruction is performed by an external NVS pipeline (*e.g.*, 2DGS [14]) and is excluded from the `Rest2Art` runtime. The dominant bottleneck is video generation, which accounts for approximately half of the total runtime and dominates peak GPU memory. Mesh completion runtime scales with part count and mesh complexity. Co-refinement reaches up to 72 s when VLM re-inference is triggered. Tracking and joint fitting are lightweight, each completing within 25 s across all tested scenes. When static recovery is triggered, two additional videos are generated, adding approximately 110 s of inference time to the total.

Table B: Per-stage runtime on ACD dataset.

Stage	Mean (s)
Co-refinement	30
Segmentation	3
Prompt Gen.	18
Video Gen.	172
Tracking	6
Joint Fitting	7
Mesh Completion	40
Total	**∼277**

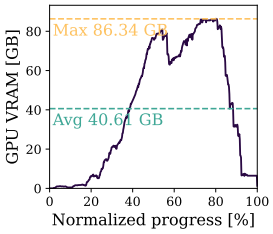

Fig. A: Per-stage VRAM usage on ACD dataset.

B Experimental Details

This section provides additional experimental details for `Rest2Art`.

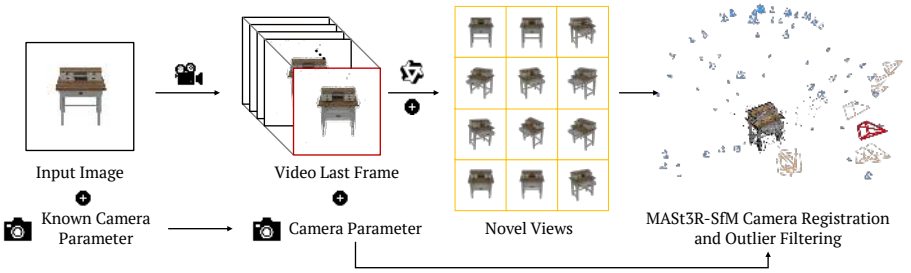

Fig. B: Input synthesis pipeline for two-state reconstruction baselines. Yellow frustums denote novel views registered via MASt3R-SfM [11, 23], and the red frustum denotes the reference view shared with the rest-state input.

- Section B.1 details the two-state input construction procedure for reconstruction methods requiring two-state input [1, 30, 51, 52].
- Section B.2 details the preprocessing procedure for the MultiScan [32] real-world evaluation.

B.1 Two-State Input Construction

Why Direct Comparison Is Infeasible. Two-state reconstruction methods [30,51,52] require paired multi-view images of two distinct articulation states as input. Articulation in Motion [1] requires both multi-state observations and an interaction video showing the articulation motion. In the rest-state setting, only a single closed-configuration observation is available, making direct application infeasible. Rather than excluding the comparison entirely, we construct the best achievable inputs using the same video diffusion pipeline as our method, establishing an upper-bound comparison that reflects the inherent difficulty of the rest-state setting rather than any limitation of these baselines themselves.

Two-State Input Synthesis. We select the last physically plausible frame from the generated articulation video as the articulated-state reference. Since our prompt enforces a strictly fixed camera throughout the video, the last frame shares the same camera parameters as the input rest-state image. We therefore reuse the input camera as the reference pose for the articulated state. This camera-sharing assumption also applies to our method, but multi-state methods are more sensitive to camera parameter accuracy given their reliance on precise multi-view geometry. We therefore manually verify the absence of camera drift and hallucination artifacts for every generated video on the ACD dataset [6, 15, 19], retaining 46 of 69 videos before running these multi-state baselines. Multi-view articulated-state images are then synthesized from the selected frame using Qwen-Image-Edit [36, 50] with a multi-angle LoRA, which produces view-consistent images that faithfully preserve object identity and geometry across viewpoints, yielding up to 13 images per object, including the

input view (Fig. B). The mask overlay is not applied to video generation in this preprocessing to maintain a consistent appearance between the rest and articulated states; this leads to higher hallucination rates compared to our method, as shown in Fig. 8.

Camera Estimation and View Filtering. Camera parameters are estimated via MASt3R-SfM [11, 23], anchored to the input view as the coordinate reference. We apply a two-pass filtering strategy to remove geometrically inconsistent views. In the first pass, all views are registered jointly, and outlier cameras are identified using a Median Absolute Deviation (MAD)-based robust test, where a view is rejected if its z-score exceeds $\tau = 3.0$. Filtering is constrained to remove at most 30% of views while retaining a minimum of 5. In the second pass, SfM is re-run on the remaining views to improve camera accuracy. On average, 12.54 of 13 generated views are retained per object across the 46 scenes, with outlier removal occurring in 9 scenes (20%).

B.2 MultiScan Preprocessing Details

Since MultiScan [32] annotations are at the scene level, we extract object-centric articulated instances as follows. Using the mesh-level `objectId` and `partId` labels, we crop each annotated object from the scene mesh and export its part-segmented mesh together with joint origins and axes. We retain only openable objects whose annotations are valid. For each instance, we then select visible RGB views by projecting its 3D points into the subsampled image frames with the provided intrinsics and poses, filter the views by visibility and scene-level occlusion, and transform the retained images and poses into the object's local frame. Note that the two states are not always well-curated pairs of rest and articulated configurations; we retain such instances as part of the real-world capture noise rather than filtering them out.

C Additional Experiments and Analyses

This section provides supplementary experimental results for `Rest2Art`.
- Section C.1 analyzes the convergence behavior of the co-refinement loop.
- Section C.2 analyzes the choice of video diffusion as the motion source and the robustness of joint estimation to its artifacts.
- Section C.3 evaluates quantitative part segmentation results against 3D segmentation baselines [28, 31].
- Section C.4 analyzes failure modes and directions for improvement.

C.1 Convergence of Co-Refinement

We say that the co-refinement loop *converges* when the VLM's predicted part counts agree with the segmentation model's detections across all part categories,

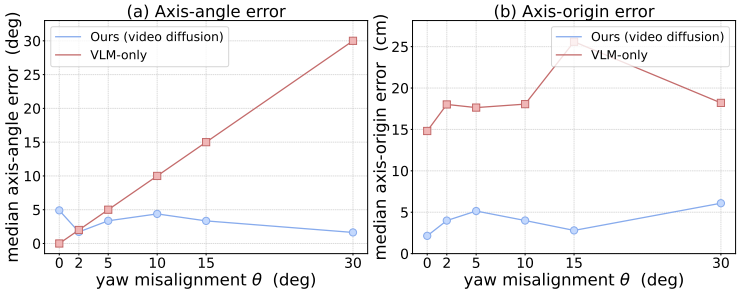

Fig. C: Joint axis accuracy under yaw misalignment on the ACD dataset [15]. As the object is rotated about the vertical axis away from canonical alignment, VLM-based direct regression degrades sharply, exposing its reliance on viewpoints. Our mesh-grounded fitting remains robust.

requiring no further refinement rounds. We cap the maximum number of refinement rounds at $R_{\max} = 2$ (Tab. A), under which 84.1% of ACD scenes reach agreement. This bound keeps inference cost low while resolving most disagreements: raising $R_{\max}$ to 20 yields only a modest gain to 92.8%.

The remaining 7.2% of cases do not diverge; instead, they oscillate between two stable configurations across iterations. This behavior typically arises on objects with visual ambiguity, such as low-texture surfaces or weak part boundaries, where the VLM hierarchy and the segmentation outputs fail to settle on a single consistent decomposition.

C.2 VDM Output as Motion Hypothesis

Why Video Diffusion, Not a VLM. A natural alternative to a video diffusion model (VDM) is to directly regress joint parameters with a vision-language model (VLM). However, VLMs are trained predominantly on 2D images and lack reliable 3D spatial reasoning, making direct regression of continuous joint parameters from a single rest-state image fundamentally unreliable. Reported successes typically depend on canonical, axis-aligned viewpoints, where joint directions reduce to a small set of memorized orientations (*e.g.*, $\pm x$, $\pm y$, $\pm z$) rather than reflecting genuine geometric inference. We confirm this empirically by perturbing the object pose with yaw misalignment (Fig. C): VLM-based regression degrades sharply under even modest deviation, while our pipeline remains stable. VDMs, by contrast, are trained on motion data and produce temporally coherent trajectories that our mesh-grounded rigid-motion fitting converts into joint parameters robust to pose noise.

Robustness to VDM Artifacts. Video diffusion models produce plausible articulation trajectories from a single image, motivating their use in our pipeline. At the same time, video generation remains an active research area, particularly

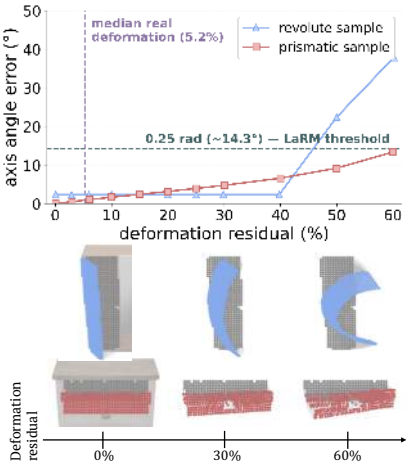

Fig. D: Axis error under injected deformation. The dashed marker indicates the empirical median deformation on the ACD dataset [15].

Table C: Part segmentation results on ACD.

	Find3D [31]	PartField [28]	**Ours**
IoU	0.20	0.21	**0.88**
mAP	0.70	0.46	**0.89**

in 3D-aware reasoning and physically grounded world models. Our pipeline accordingly treats VDM output as a motion-hypothesis source and the 3D mesh as the geometric ground truth.

We use VBVR LoRA [20,47] to suppress hallucination in generation (Fig. 8), and mesh-grounded rigid-body fitting to handle unreliable trajectories and non-rigid drift. For deformation specifically, we synthetically inject deformation into the part geometry and measure the resulting axis error (Fig. D). At the empirical median deformation of 5.2% on the ACD dataset [15], the axis error remains negligible, confirming that our pipeline operates effectively within the regime of realistic generation noise.

C.3 Segmentation Quantitative Results

We evaluate part segmentation quality on the ACD dataset [6, 15, 19] using IoU and mAP, comparing against 3D segmentation methods PartField [28] and Find3D [31]. `Rest2Art` substantially outperforms both baselines across all metrics, demonstrating the benefit of grounding segmentation in the reconstructed mesh geometry combined with the VLM-predicted part hierarchy. Find3D achieves relatively high mAP despite low IoU, as text-prompted parts are often under-segmented, leaving a large unsegmented base region that dominates the evaluation and inflates mAP.

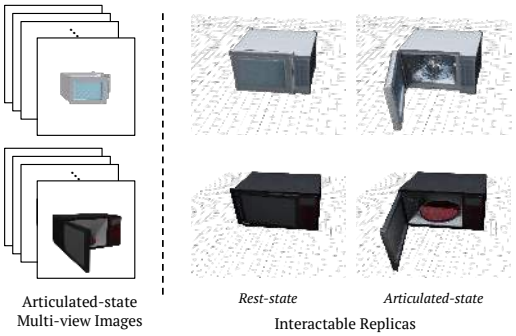

Fig. E: Results on articulated-state inputs from the PartNet-Mobility *microwave* category.

C.4 Failure Case Analysis

Although our co-refinement pipeline with mask overlay suppresses hallucination artifacts, 2 out of 69 videos (2.9%) still exhibit corrupted geometry that prevents meaningful track extraction. Degraded joint accuracy primarily occurs on objects with subtle or short-range articulation, where track displacement is too small to reliably estimate motion parameters. We see two directions for improvement. First, increasing the number of generated videos enlarges the hypothesis pool, improving the chance of capturing reliable motion for difficult cases. Second, since our pipeline leverages 3D geometric cues such as collision and spatial constraints for joint refinement, the quality of the underlying mesh directly affects estimation accuracy. Cleaner part boundaries from improved segmentation, potentially assisted by generative remeshing models, would reduce ambiguity in the geometric reasoning stage.

D Applications

This section demonstrates `Rest2Art` beyond the standard evaluation setting.
- Section D.1 shows that `Rest2Art` generalizes to articulated-state inputs without any modification.
- Section D.2 validates `Rest2Art` in real-world captures across diverse input modalities and articulation states.
- Section D.3 extends `Rest2Art` to broader object categories through a meta-prompt mechanism.
- Section D.4 demonstrates downstream applications of `Rest2Art`.

D.1 Single Articulated-State Input

Although `Rest2Art` is designed for rest-state observation, the same pipeline generalizes to objects observed in a partially or fully open configuration, as shown in

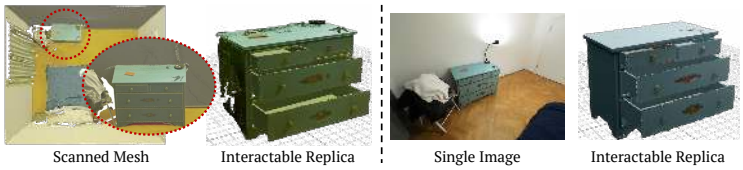

Fig. F: Real-world results on a ScanNet++ scene [56]. Left: a raw scanned mesh. **Right:** a single captured image reconstructed via SAM3D [45].

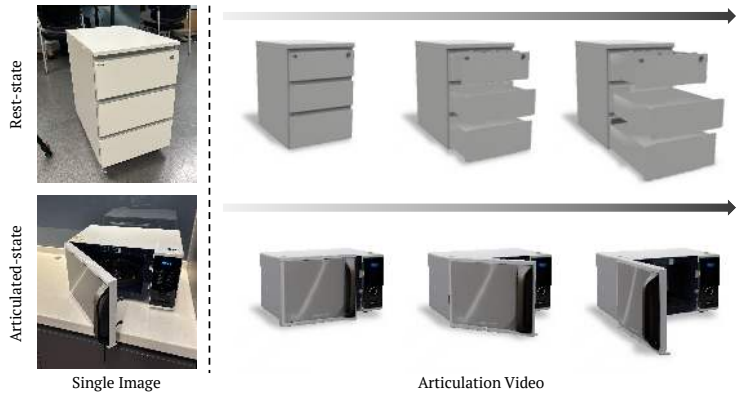

Fig. G: Real-world single-image results. Given a single casually captured image, `Rest2Art` reconstructs the object via SAM3D [45] and estimates articulation. **Top:** A rest-state cabinet. **Bottom:** A microwave captured in a partially open configuration.

Fig. E. This is because rest-state reconstruction is strictly harder: the closed configuration provides no direct motion cues, requiring the full pipeline, including motion hypothesis generation and mesh-grounded refinement, to recover joint parameters from appearance alone. An articulated-state observation, by contrast, already encodes part displacement in the input geometry, so the same pipeline succeeds with less reliance on the hypothesis generator. As a result, any input that `Rest2Art` handles in the rest-state setting is also handled in the articulated-state setting, but not vice versa. We demonstrate this on *microwave* instances from the PartNet-Mobility dataset [4, 34, 53], omitting mesh completion as the articulated state directly exposes interior geometry.

D.2 Real-World Demonstrations

We demonstrate `Rest2Art` on a diverse set of real-world captures spanning multiple input modalities. Figure F presents results on the same ScanNet++ scene [56] under two modalities: a raw scanned mesh and a single captured image reconstructed via SAM3D [45], showing consistent articulation estimates across reconstruction pipelines. Figure G shows results on casually captured in-the-wild images, covering both rest-state and partially open inputs. In all cases, `Rest2Art`

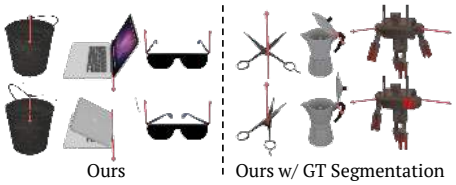

Fig. H: Generalization to broader object categories via meta-prompting.

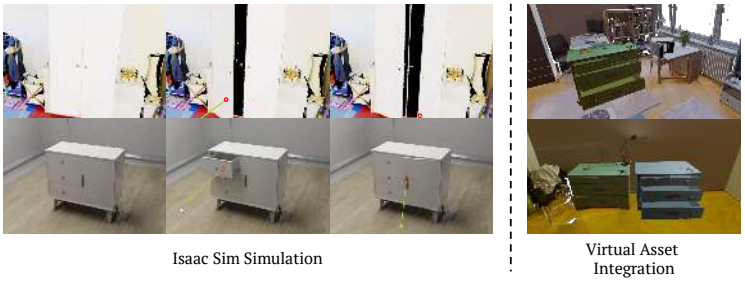

Fig. I: Downstream applications of `Rest2Art`. Left: Physics-based manipulation in Isaac Sim [35], showing a reconstructed cabinet with operable drawers. **Right:** Virtual asset integration, placing a reconstructed interactable replica into a real-world scene.

exports simulation-ready URDF assets suitable for embodied AI and robotics applications.

D.3 Generalization to General Object Categories

While our main evaluation targets openable furniture with prismatic and revolute joints, our framework generalizes to broader categories through a *meta-prompt* mechanism. Articulated objects span a wide range of part vocabularies (*e.g.*, a laptop's display lid, a robot's arm) that no single fixed prompt covers well. Our pipeline therefore issues two VLM calls: the first sends a category-agnostic system prompt and asks the VLM to rewrite it for the input object, adapting the part vocabulary, the per-object definition of "base," and the example response. The second call applies the rewritten prompt to the same image and produces the part-connectivity tree, which feeds the unchanged mask, segmentation, video, and joint-fitting modules. Figure H shows representative results on PartNet-Mobility [4,34,53] categories (eyeglasses, laptops, scissors, buckets) and Objaverse [9] assets (moka pot, mechanical robots). Extending to non-planar joints and more general articulation models remains an open direction.

D.4 Downstream Applications

The simulation-ready URDF assets produced by `Rest2Art` can be directly deployed in downstream applications including physics simulation and virtual asset integration, as shown in Fig. I.