ISART: SINGLE-IMAGE, STRUCTURE-GUIDED GENERATION OF SIM-READY ARTICULATED ASSETS

Anonymous authors
Paper under double-blind review



Figure 1: **From a Single Image to Simulation-Ready 3D Articulated Assets. Left:** Three examples show single-image generation of part-level meshes and URDF joint structures, followed by Isaac Sim simulation. We target batch generation of simulation-ready furniture and household appliances with realistic appearance and physically plausible articulation. **Right:** Diverse generated articulated assets populate an indoor scene.

ABSTRACT

Generating simulation-ready articulated assets from a single image requires detailed part geometry and coherent kinematic structures. While Coding agents built on models such as GPT6-Astra can automate programmatic modeling and joint specification, faithfully reproducing intricate, image-specific part geometry through code remains challenging. We present *ISArt*, a closed-loop framework that combines agentic structural reasoning with learned part-level geometry generation. Given an input image, a vision-language model (VLM) iteratively decomposes the object into an editable structural representation, identifying its semantic parts, spatial extents, joint types, and parent–child relationships. For objects with concealed interiors, *ISArt* additionally synthesizes open-state references to reveal these structures. Conditioned on both visual observations and the inferred structure, a diffusion transformer generates detailed geometry for each part, followed by part-level texture synthesis. An articulation agent then estimates joint origins, axes, and motion ranges from the generated geometry and evaluates geometric and motion consistency. These feedbacks guide iterative refinement, fallback selection, and candidate acceptance or rejection. By assigning detailed geometry synthesis to a learned generative model and structural planning and articulation refinement to an agentic workflow, *ISArt* produces textured part meshes with URDF-defined kinematics for robotic simulation and interactive applications.

1 INTRODUCTION

Articulated 3D assets are essential for robotic simulation, embodied AI, and interactive virtual environments. Unlike static objects, a simulation-ready articulated asset must jointly capture detailed geometry, meaningful part decomposition, and executable kinematic structures. Reconstructing such assets from a single image could substantially reduce the cost of building interactive environments.

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.

Recent approaches mainly follow two directions. Agentic and programmatic methods such as (Zhou et al., 2026) and (Wang et al., 2026b) explicitly construct object parts, connectivity, and joint structures, but often struggle to reproduce complex, image-specific geometry through code. contrast, learned 3D generation methods such as Spark (He et al., 2026) and (Wu et al., 2026) can produce high-quality part-level geometry, yet geometric decomposition alone does not guarantee valid articulation. Generated parts may not correspond to rigid motion units, and their geometry may be inconsistent with intended connectivity and kinematic constraints.

We present ISArt, a structure-guided framework for generating simulation-ready articulated assets from a single image. Its core design is to combine explicit geometric–kinematic reasoning with learned high-fidelity geometry generation, and to close the loop through simulation-based verification. ISArt first establishes a structured representation of the articulated object, uses this structure to guide part-level geometry generation, and subsequently converts the generated parts into an executable articulated asset through agentic binding and validation.

ISArt introduces two key components. First, a structure-grounded planning and part generation framework infers part semantics, connectivity, joint types, coarse motion axes, spatial extents, and cavity information, and injects these priors directly into a multi-part DiT to guide geometry synthesis. Second, an agentic articulation binding and validation framework estimates joint origins, axes, and motion ranges from the generated geometry, constructs the corresponding URDF, and iteratively refines invalid motions through Isaac Sim and video-based feedback.

Experiments on 60 articulated objects across 12 categories and 100 in-the-wild images show that ISArt improves both part-level geometry reconstruction and articulation binding over representative baselines. The generated assets can be directly deployed in Isaac Sim for downstream robotic manipulation. Our main contributions are: (1) a unified single-image framework for simulation-ready articulated asset generation; (2) structure-grounded planning and part generation with explicit geometric–kinematic conditioning; and (3) agentic articulation binding and simulation-based validation for reliable executable assets.

Our main contributions are summarized as follows:

- We introduce ISArt, a single-image framework for generating simulation-ready articulated assets by combining structure-grounded generation with agentic articulation refinement.
- We propose a structure-grounded planning and part generation framework that infers explicit geometric-kinematic priors and uses them to guide high-fidelity part-level 3D generation.
- We develop an agentic articulation binding and validation framework that estimates executable joint parameters and iteratively refines them through simulation and video-based feedback.
- We construct a comprehensive sim-ready asset generation pipeline with automated motion-binding capabilities, and release a large-scale dataset of 1,000 articulated assets to facilitate downstream physical simulation research.

2 Related work

2.1 Part-Level 3D Generation and Decomposition

Part-level 3D methods decompose existing geometry or generate complete components for independent manipulation. SAMPart3D (Yang et al., 2024), P3-SAM (Ma et al., 2025), and PartField (Liu et al., 2025b) segment geometry through feature grouping or prompt-based mask prediction. PartGen (Chen et al., 2025a) and HoloPart (Yang et al., 2026a) combine decomposition with generative completion to recover full part geometry. PartCrafter (Lin et al., 2025) jointly generates multiple part meshes; OmniPart (Yang et al., 2025) guides generation with part layouts; and AutoPartGen (Chen et al., 2025b) generates parts autoregressively. However, geometric or semantic parts need not coincide with rigid-body motion units; articulated assets additionally require kinematically consistent decomposition, connectivity, and motion constraints.

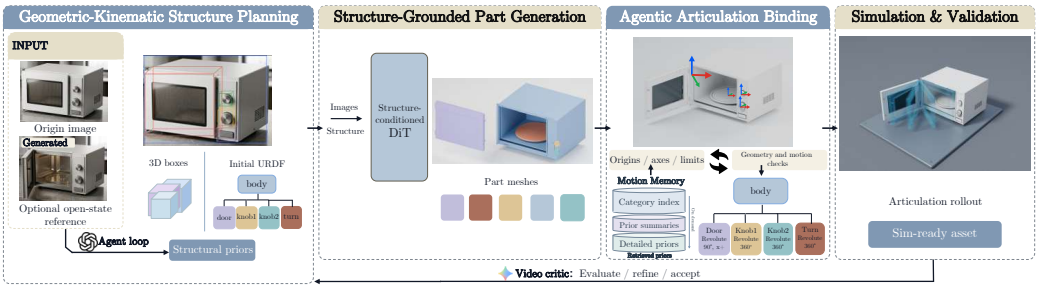


Figure 2: **Overview of our method.** ISArt predicts structural priors from a single image to guide part-level geometry generation and agentic articulation binding. Simulation and video-critic feedback drive closed-loop refinement to produce simulation-ready articulated assets.

2.2 Agentic and Programmatic Articulated Object Modeling

LLM-driven agents automate articulated modeling through code generation, tool use, and iterative feedback. The text-to-cad toolkit (earthtojake, 2026) provides CAD and URDF tools, while Articraft (Zhou et al., 2026) and ArtiCAD (Shui et al., 2026) programmatically construct geometry and joints. Real2Code (Zhao et al., 2025) predicts joint parameters from reconstructed parts, whereas Articulate-Anything (Le et al., 2025) generates assembly and joint code from retrieved meshes using actor–critic feedback. UnfoldArt (el Amine Boudjoghra et al., 2026) combines multi-agent debate with generated motion videos to refine articulation and recover interior geometry. Retrieval-based assembly is constrained by library coverage, while purely programmatic geometry may struggle to reproduce complex shapes and local image details.

2.3 Image-Conditioned Articulated Asset Generation

Image-conditioned methods infer part geometry and kinematics from a single image. URDFormer (Chen et al., 2024) and SINGAPO (Liu et al., 2025a) assemble predefined or retrieved meshes using predicted part layouts and kinematic structures. A complementary route combines image-to-3D reconstruction with mesh-based articulation: Articulate AnyMesh (Qiu et al., 2025), Particulate (Li et al., 2026), and SIMART (Zhang et al., 2026) decompose static meshes and infer joint parameters.ArtLLM (Wang et al., 2026a) predicts part layouts and kinematics from point clouds to guide articulated asset generation. PAct (Liu et al., 2026) and URDF-Anything+ (Wu et al., 2026) jointly model geometry and articulation under image conditioning; PhysX-Anything (Cao et al., 2026) and PhysForge (Yang et al., 2026b) use image-derived physical descriptions to guide geometry and kinematics; and DreamArt (Lu et al., 2025) and SPARK (He et al., 2026) exploit generated motion videos or articulated-state images as visual priors for joint refinement. Despite this progress, objects with intricate geometry or many interacting parts, such as ornate cabinets and ergonomic chairs, remain challenging to reconstruct with both image fidelity and reliable kinematics.

3 Method

3.1 Problem Formulation and Overview

Given a single input image I_0, our goal is to generate a sim-ready articulated asset

$$\mathcal{A} = F(I_0) = \big(\mathcal{M}, \mathcal{T}, \Theta\big), \qquad \mathcal{M} = \{M_i\}_{i=1}^{N}, \tag{1}$$

where $\mathcal{M}$ denotes the complete part-level geometry, $\mathcal{T}$ represents the URDF kinematic tree including part semantics, parent–child relations, and joint types, and Θ contains the continuous articulation parameters such as joint origins, axes, and motion ranges.

To achieve this, we propose **ISArt**, an agentic framework consisting of four stages: **Geometric-Kinematic Structure Planning** uses a VLM to iteratively infer the structural prior $\mathcal{S}$ from the input image I_0, and generates an open-state reference image I_{open} for objects with hidden internal geometry to provide complementary visual cues. **Structure-Grounded Part Generation** conditions on

I_0 and $\mathcal{S}$, injecting structural information into a DiT (Peebles & Xie, 2023) to generate part-level 3D geometry $\mathcal{M} = \{M_i\}_{i=1}^N$. **Agentic Articulation Binding** uses $\mathcal{S}$ to bind the part geometry $\mathcal{M}$ to the kinematic tree $\mathcal{T}$ and estimate continuous joint parameters Θ, constructing an executable articulated structure. **Simulation and validation** evaluates motion consistency through motion videos rendered in Isaac Sim, uses the feedback to guide the agent in revising Θ, rolling back, and retrying, and discards results that repeatedly fail validation, ultimately yielding a validated articulated asset $\mathcal{A} = (\mathcal{M}, \mathcal{T}, \Theta)$. An overview of the complete pipeline is shown in Fig. 2. Due to space limitation, detailed agentic workflow are deferred to appendix.

3.2 Geometric-Kinematic Structure Planning

To acquire the geometry and kinematic structure of the target object, we first use VLM to predict the global structural information

$$\mathcal{S} = (N, T_{\text{category}}), \tag{2}$$

where N denotes the number of parts and T_{category} denotes the object category. We then jointly predict structural information $\{S_i\}_{i=1}^N$ for all parts. For part i, we organize part-wise structural information into several text descriptions and vectors:

$$S_i = (T_{\text{semantic},i}, T_{\text{conn},i}, l_i, J_i, \boldsymbol{c}_{\text{box},i}, \boldsymbol{s}_{\text{box},i}, \boldsymbol{c}_{\text{cav},i}, \boldsymbol{s}_{\text{cav},i}) \tag{3}$$

where $T_{\text{semantic},i}$, denotes the semantic label of part i; $T_{\text{conn},i}$ denotes its parent and child connection in the kinematic tree; $l_i \in \{0,1\}^4$ denotes the one-hot joint-type representation of part i, corresponding to four predefined categories: fixed, revolute, prismatic, and continuous;$J_i \in \mathbb{R}^3$ denotes the coarse joint-axis direction of part i,; $\boldsymbol{c}_{\text{box},i}, \boldsymbol{s}_{\text{box},i} \in \mathbb{R}^3$ denote the center and size of the part bounding box, and $\mathbf{c}_{\text{cav},i}, \mathbf{s}_{\text{cav},i} \in \mathbb{R}^3$ denote the center and size of the associated cavity bounding box. The proposed method only handles the case of one predicted cavity (one hole). If no cavity is predicted, we set $\mathbf{c}_{\text{cav},i} = \mathbf{s}_{\text{cav},i} = \mathbf{0}$.

Optional Open-State Visual Guidance Generation. Given the structure information, the planning module optionally generates an open-state visual guidance step for objects with potentially occluded internal structures. Specifically, the VLM first determines whether such an internal concealed space exists. If so, GPT-Image-2.5 is prompted to synthesize an open-state image I_{open}, providing complementary visual guidance for the subsequent structural prediction.

Coarse URDF prediction. Inspired by recent agentic text-to-articulated-asset generation methods such as aDSL (Wang et al., 2026b) and ArtiCraft (Zhou et al., 2026), we propose an agentic URDF prediction workflow for inferring coarse kinematic structures from a single image. Given the input image, the VLM first predicts the object category T_{category}, the number of articulated parts N, and the part-level semantic labels T_{semantic}. Based on these predictions, our SDK instantiates a structured URDF template in JSON format, with one entry for each predicted part and placeholders for the corresponding URDF attributes. The VLM then completes these fields to construct a coarse URDF representation of the articulated object. After completing the JSON document, we obtain a coarse-grained URDF representation of the articulated object, from which the parent–child relationships T_{conn} can be directly extracted. The VLM is then prompted to critique and revise the predicted structure based on both the input image I_{image} and the optional open-state image I_{open}; after a number of iterative refinement loops, the refined coarse URDF structure is obtained.

Bounding box refinement. Following the same agentic paradigm, for each part i, we refine the box center $c_{\text{box},i}$, box size $s_{\text{box},i}$, and cavity box center $c_{\text{cav},i}$, cavity box size $s_{\text{cav},i}$. These four quantities are then refined through an iterative estimate-project-critique loop. Given I_{image} and the coarse URDF, the VLM first predicts an axis-aligned 3D bounding box for each part in the canonical object frame. We define the *static body* as the non-articulated base part that serves as the reference for the movable parts. Its predicted box is kept fixed and used to align the projected 3D boxes with the input image.

At each refinement step, the current 3D box b_{3d} is projected onto the image plane using the canonical camera projection Π_C. The resulting projection is compared with the corresponding 2D box b_{2d} to obtain a 2D reprojection residual r_{2d}. The VLM then predicts a correction Δb_{3d} conditioned on the residual and the current box estimates:

$$r_{2d} \leftarrow \|b_{2d} - \Pi_C(b_{3d})\|^2; \qquad \Delta b_{3d} \leftarrow \text{VLM}(r_{2d}, b_{3d}, b_{2d}). \tag{4}$$

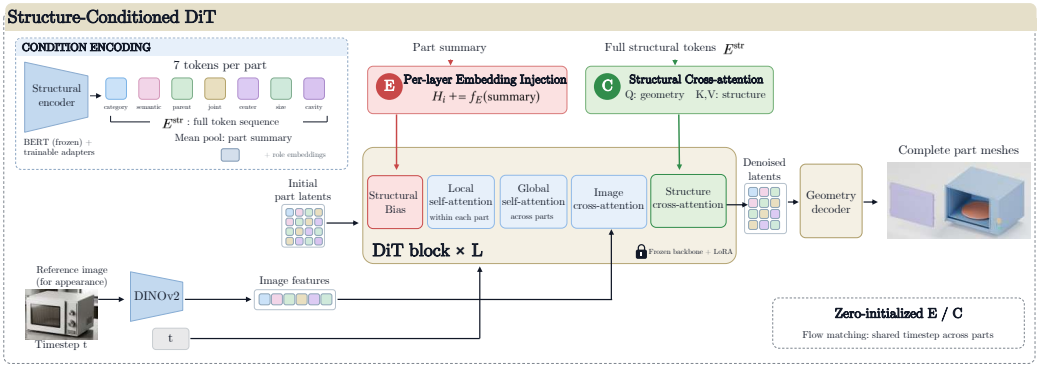


Figure 3: **Structure-conditioned DiT.** Per-layer embedding injection (E) incorporates pooled part summaries, while structural cross-attention (C) accesses each part's structural tokens. Together with image features, these conditions guide the denoising of part latents, which are decoded into complete part meshes.

The 3D box is subsequently updated as $b_{3d} \leftarrow b_{3d} + \Delta b_{3d}$, and the project–critique–update procedure is repeated while keeping the static-body box fixed. The refined boxes are finally stored as $c_{\text{box},i}$, $s_{\text{box},i}$, $c_{\text{cav},i}$, and $s_{\text{cav},i}$.

3.3 Structure-Grounded Part Generation

Given the structural priors predicted in 3.2, we incorporate them into a part-level generative model to guide geometry synthesis.Inspired by TripoSG's strong pretrained geometry prior (Li et al., 2025), which we qualitatively observe to better preserve fine geometric details and concave structures, we design a structure-guided multi-part generation network that explicitly incorporates geometric and kinematic constraints through modular structural conditioning mechanisms.The overall network architecture is illustrated in Fig. 3.

Multi-Part Geometry Generation. Given the predicted number of parts N, we perform multi-part geometry generation in a part-aligned latent space. Each part i is represented by M geometry latent tokens with feature dimension D. Generation starts from randomly initialized part latents. At sampling timestep t, the latent representations of all parts are organized as

$$\mathbf{Z}_t = [\mathbf{Z}_t^1; \mathbf{Z}_t^2; \ldots; \mathbf{Z}_t^N], \qquad \mathbf{Z}_t^i \in \mathbb{R}^{M \times D}, \tag{5}$$

where the predicted part number N directly determines the number of latent groups used for generation.

Meanwhile, we employ a shared frozen DINOv2 encoder (Oquab et al., 2024) to extract visual features from the input image I_0 and the optional open-state reference I_{open}, which provides complementary cues for occluded internal structures, yielding $\mathbf{E}^{\text{vis}} = [\mathcal{E}_{\text{img}}(I_0); \mathcal{E}_{\text{img}}(I_{\text{open}})]$.

The Multi-Part DiT iteratively updates these latents under the visual condition, progressively transforming the initial noise into part-level geometry representations. After denoising, the final latent representation of each part, $\mathbf{Z}_0^i$, is independently decoded by a pretrained geometry decoder as $M_i = \mathcal{D}_{\text{geo}}\left(\mathbf{Z}_0^i\right)$, with $\mathcal{M} = \{M_i\}_{i=1}^N$. This part-aligned latent representation provides the basis for incorporating explicit geometric and kinematic structural conditions in the following stages.The generated part meshes are subsequently textured independently using Meshy (mes), guided directly by the global input image I_0.

Part-Aligned Structural Tokenization. Given the structural attributes defined in 3.2, we encode them into $K = 7$ part-aligned structural tokens: T_{category}, $T_{\text{semantic},i}$, joint token(including l_i, J_i), $T_{\text{conn},i}$, $\boldsymbol{c}_{\text{box},i}$, $\boldsymbol{s}_{\text{box},i}$, cavity token(including $c_{\text{cav},i}$, $\boldsymbol{s}_{\text{cav},i}$.)

Specifically, object category and part semantics are treated as textual attributes; parent relation and joint type as discrete attributes; and joint axis,center, size, and cavity bounding box as continuous

attributes.Textual attributes are encoded using a frozen BERT-base encoder (Devlin et al., 2019) followed by a lightweight projection layer, discrete attributes are represented by learnable embeddings, and continuous attributes are normalized in the canonical object frame and projected with lightweight MLPs. Joint-related attributes are fused into a single joint token, while missing values are represented by learnable embeddings. A learnable role embedding is further added to distinguish different structural attributes. The resulting representation for part i is $\mathbf{E}_i^{\mathrm{str}} \in \mathbb{R}^{7\times d_s}$, which is mean-pooled into a part-level summary feature $\bar{\mathbf{e}}_i^{\mathrm{str}}$ for subsequent structural conditioning.

Modular Structural Conditioning. Relying solely on image conditioning makes it difficult for the model to capture the structural characteristics of articulated objects. As VLMs make such structural priors increasingly accessible from a single image, we explicitly inject them into the multi-part geometry generation process through two complementary conditioning mechanisms: per-layer embedding injection (*E*) and structural cross-attention (*C*). At sampling timestep t, the latent tokens of part i, $\mathbf{Z}_t^i$, are projected into the DiT hidden space as

$$\mathbf{H}_0^i = f_{\mathrm{in}}(\mathbf{Z}_t^i) + \mathbf{e}_t, \tag{6}$$

where $\mathbf{e}_t$ denotes the timestep embedding and $\mathbf{H}_l^i$ denotes the hidden geometry tokens of part i at the l-th DiT layer.

For *E*, the structural token sequence $\mathbf{E}_i^{\mathrm{str}} \in \mathbb{R}^{K\times d_s}$ is first aggregated into a part-level summary,

$$\bar{\mathbf{e}}_i^{\mathrm{str}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{E}_{i,k}^{\mathrm{str}}, \qquad K = 7, \tag{7}$$

which is projected into the DiT hidden space and injected into each layer:

$$\widetilde{\mathbf{H}}_l^i = \mathbf{H}_l^i + f_E\big(\bar{\mathbf{e}}_i^{\mathrm{str}}\big)\,. \tag{8}$$

This provides a persistent part-level structural bias throughout the generation process.

The resulting geometry tokens are then processed by the original DiT operations, including local self-attention within each part, global self-attention across parts, and image cross-attention with the visual features $\mathbf{E}^{\mathrm{vis}}$. We denote these backbone operations compactly as

$$\widehat{\mathbf{H}}_l^i = \mathcal{B}_l\left(\widetilde{\mathbf{H}}_l^i, \mathbf{E}^{\mathrm{vis}}\right), \tag{9}$$

where $\mathcal{B}_l$ denotes the original l-th DiT block.

For *C*, instead of pooling the structural information, we directly attend to the full structural token sequence. Geometry tokens serve as queries, while structural tokens provide keys and values:

$$\mathbf{Q}_i^C = \widehat{\mathbf{H}}_l^i \mathbf{W}_Q^C, \qquad \mathbf{K}_i^C = \mathbf{E}_i^{\mathrm{str}} \mathbf{W}_K^C, \qquad \mathbf{V}_i^C = \mathbf{E}_i^{\mathrm{str}} \mathbf{W}_V^C. \tag{10}$$

The structural cross-attention is then computed as

$$\mathbf{A}_i^C = \mathrm{softmax}\left(\frac{\mathbf{Q}_i^C (\mathbf{K}_i^C)^\top}{\sqrt{d}}\right), \qquad \mathbf{H}_{l+1}^i = \widehat{\mathbf{H}}_l^i + \mathbf{A}_i^C \mathbf{V}_i^C, \tag{11}$$

where d is the attention-head dimension. Since $\mathbf{E}_i^{\mathrm{str}}$ contains the seven part-aligned structural tokens, each geometry token can dynamically retrieve the semantic, spatial, and coarse URDF information most relevant to its current state.

Overall, *E* provides persistent part-level structural guidance, whereas *C* enables fine-grained token-level structural retrieval. The two mechanisms are complementary and are jointly applied throughout the DiT denoising process.

Training Objective and Adaptation. We retain the original Rectified Flow training objective without introducing additional geometry supervision. Given the interpolated latent $\mathbf{Z}_t$, the model predicts the velocity field conditioned on both visual and structural information Φ, and minimizes

$$\mathcal{L}_{\mathrm{RF}} = \mathbb{E}_t\left[\left\|V_\theta(\mathbf{Z}_t, \mathbf{\Phi}, t) - \mathbf{U}^*\right\|_2^2\right]. \tag{12}$$

Here, t denotes the sampled flow timestep, $\mathbf{Z}_t$ is the interpolated latent at timestep t, Φ denotes the combined visual and structural conditions, V_θ is the predicted velocity field, and $\mathbf{U}^*$ is the target velocity defined by the Rectified Flow interpolation path. The structural encoder, the *E*/*C* conditioning modules, and the LoRA parameters (Hu et al., 2022) inserted into the attention layers are jointly optimized under this objective. To preserve the pretrained geometry prior, the structural residual outputs of *E* and *C* are zero-initialized, allowing the model to start close to the original generator and progressively learn to exploit explicit structural conditions.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and Evaluation Protocols. We construct a unified evaluation benchmark consisting of 60 articulated objects, with 30 samples from World-SimReady-Home (Yootta, 2026) and 30 samples from GAPartNet_PhyScene (Geng et al., 2023), spanning 12 articulated-object categories. This benchmark is used to evaluate the end-to-end generation success rate, URDF binding accuracy, and part-level geometry quality. For geometry reconstruction, we report Chamfer Distance (CD) (Fan et al., 2017) and F-Score (Tatarchenko et al., 2019) at two thresholds. For URDF binding, we report AxisErr, PivotErr, and TypeErr to measure joint-axis deviation, pivot-position error, and joint-type prediction error. In addition, we collect 100 in-the-wild articulated-object images from the Internet for qualitative evaluation of end-to-end generation and articulation binding under challenging real-world conditions.

Implementation Details. We use GPT-5.6-sol for VLM-based structural reasoning, GPT-Image-2 for open-state image synthesis, and Gemini 3.1 for video-based motion evaluation. The geometry generator is optimized with a learning rate of 1×10^{-4} and an effective batch size of 8. Diffusion timesteps are sampled from a logit-normal distribution with $\mu = 0$ and $\sigma = 1$. We construct an initial training set of approximately 1,100 articulated assets by collecting articulated-object images from the Internet and using GPT-6 together with BlenderMCP to generate the corresponding programmatic 3D assets and articulation annotations. The model is first trained for 10,000 optimization steps. We then adopt a failure-driven data scaling strategy, periodically identifying poorly reconstructed categories and generating targeted samples for these failure cases using the same GPT-6 + BlenderMCP modeling pipeline. This process adds approximately 1,300 additional articulated assets, after which training is resumed from the latest checkpoint for another 15,000 steps. Overall, the training set contains approximately 2,400 articulated assets, and the complete 25,000-step training process takes about 80 hours on four NVIDIA A100 GPUs. More detailed training procedures and additional implementation details are provided in A.5.

4.2 MAIN RESULTS

End-to-End Comparison. We evaluate single-image articulated asset generation against two representative paradigms: code-driven articulated modeling, represented by ArtiCraft (Zhou et al., 2026), and geometry-first pipelines that reconstruct 3D geometry before estimating and binding articulation structures, including Articulate-Anything (Le et al., 2025), URDF-Anything+ (Wu et al., 2026), and SPARK (He et al., 2026). Figure 4 presents representative qualitative results, with more comparisons provided in A.2. Both baseline paradigms remain challenged by articulated objects with complex geometry and abundant curved surfaces, reflecting limitations in aligning textual or code representations with fine-grained geometry and in the diversity of existing URDF supervision.

URDF Binding with Fixed Geometry. To isolate articulation binding quality from geometry reconstruction, we evaluate Articulate AnyMesh (Qiu et al., 2025), Articulate-Anything (Le et al., 2025), SPARK (He et al., 2026), and our method under the same fixed-geometry setting.Additional qualitative experiments are provided in A.2. For quantitative evaluation, we instead use the ground-truth part geometry from the benchmark set introduced above, and We report AxisErr, PivotErr, and TypeErr to measure joint-axis deviation, pivot-position error, and joint-type prediction error. The quantitative results are summarized in Table 1.

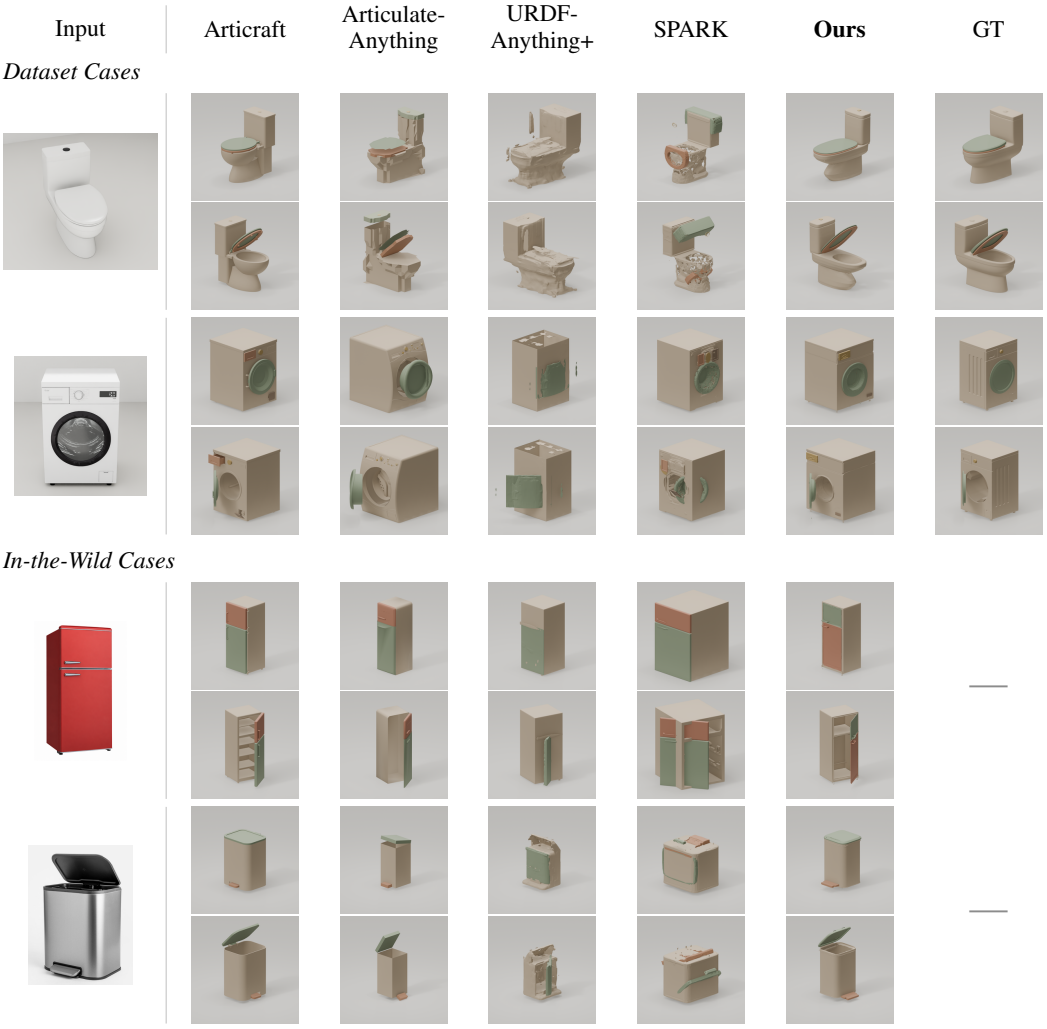


Figure 4: **Qualitative comparison of articulated asset generation.** Each case includes one input image and two articulation states (top/bottom) for each method and the available ground truth (GT). Dataset cases precede in-the-wild cases. A dash indicates unavailable 3D ground truth.

Table 1: Quantitative comparison of URDF binding using the same ground-truth part geometry. Lower values indicate better performance (except SuccessRate, where higher is better).

Method	**AxisErr** ↓	**PivotErr** ↓	**TypeErr** ↓	**SuccessRate** ↑
Articulate AnyMesh (Qiu et al., 2025)	0.72	0.02	46.67%	80.00%
Articulate-Anything (Le et al., 2025)	0.12	0.17	21.11%	90.00%
SPARK (He et al., 2026)	0.38	0.19	23.61%	100.00%
Ours	0.07	0.05	13.33%	100.00%

4.3 Part-Level Geometry Generation

To evaluate the geometric quality of the generated parts, we conduct both qualitative and quantitative comparisons with PartCrafter (Lin et al., 2025), OmniPart (Yang et al., 2025), SPARK (He et al., 2026), and a combined pipeline of TRELLIS.2 (Xiang et al., 2026), P3-SAM (Ma et al., 2025), and X-Part (Yan et al., 2025). As shown in Fig. 7, we compare the generated part meshes in both the assembled state and the exploded view to examine overall shape fidelity and part-level geometry. For quantitative evaluation, we use the unified benchmark introduced above, which contains 60 articulated objects spanning 12 categories. For each method, we reconstruct the part-level geometry

Table 2: Quantitative comparison of part-level geometry generation on the unified benchmark. Lower CD and higher F-Score indicate better performance.

Method	**CD↓**	**F-Score@0.05↑**	**F-Score@0.1↑**
PartCrafter	0.421	0.229	0.395
OmniPart	0.305	0.306	0.540
SPARK	0.278	0.314	0.505
TRELLIS.2 + P3-SAM + X-Part	0.268	0.431	0.627
Ours	0.163	0.604	0.789

from the same input images and compare the generated parts against their corresponding ground-truth meshes. We report Chamfer Distance (CD) to measure geometric discrepancy and F-Score to evaluate point-level surface alignment. Lower CD indicates better geometric fidelity, while higher F-Score indicates more accurate geometric reconstruction. We report F-Score at thresholds of 0.05 and 0.1, where the smaller threshold imposes a stricter matching criterion and therefore places greater emphasis on fine-grained geometric accuracy. The quantitative results are summarized in Table 2.

4.4 Ablation studies

As shown in Fig. 8, we conduct ablations at both inference and training time. For the inference-time ablation, we evaluate 16 held-out objects from four categories using the same trained *E+C* checkpoint and selectively disable the two conditioning branches to obtain *B0*, *E*, *C*, and *E+C*. For the training-time ablation, we train *C*-only and *E+C* variants from the same pretrained backbone for 1,000 steps on 240 training objects and evaluate them on eight held-out objects.

Qualitatively, *C* mainly improves global part layout and structural consistency, while *E* provides additional local refinements in some cases. Quantitatively, disabling *C* leads to a substantial performance drop, whereas removing *E* results in a relatively small change; the controlled training-time comparison further shows comparable overall performance between *C* and *E+C*.Table 3 reports the inference-time ablation results using F-Score@0.1 (↑) and center error (↓). F-Score@0.1 measures the geometric similarity between generated and ground-truth parts at a distance threshold of 0.1, while center error quantifies the positional discrepancy between their centers, reflecting part-placement accuracy.

5 Conclusion and limitation

We presented ISArt, a framework for generating simulation-ready articulated assets from a single image. ISArt combines agentic structural reasoning with learned part-level geometry generation, explicitly incorporating semantic, spatial, and kinematic priors into the geometry generation process, followed by agent-based articulation refinement with video feedback. Through comprehensive comparisons with representative baselines, ISArt achieves better performance on benchmark evaluations and shows strong generalization to challenging in-the-wild cases. These results demonstrate the potential of ISArt as a practical real-to-sim asset generation framework, supporting downstream applications such as robotic simulation .

Despite these promising results, ISArt still has several limitations. Its performance depends on the accuracy of the initial structural reasoning, and errors in part decomposition or kinematic structure may propagate to subsequent geometry generation and articulation binding. In addition, highly complex objects with many small, heavily occluded, or visually ambiguous movable parts remain challenging from a single image. Future work will focus on improving structural robustness and extending the framework to more diverse and complex articulated objects.

Reproducibility statement

Detailed experimental settings, training configurations, data splits, and evaluation protocols are provided in Sec.4and the AppendixA to facilitate reproduction of our experiments. We additionally provide the prompts, structured output formats, model configurations, and agent workflows used for structural prediction and articulation refinement. Sec.4,AppendixA, and our static project webpage present more than one hundred articulated assets generated by the proposed pipeline, together with their generation records and intermediate outputs for inspection. The corresponding meshes, articulation parameters, asset metadata, configuration files, and reproducible implementation will also be released.

Disclosure of AI Use

In this work, generative AI tools supported the development and refinement of research hypotheses, methodology, and experimental plans, as well as algorithm implementation and the interpretation of experimental findings. The formulation of mathematical claims, development of essential proof steps, and writing of mathematical proofs are not applicable to this study. We additionally used generative AI to synthesize and analyze prior literature, explore research directions, identify research gaps, brainstorm ideas, and improve the language and organization of the manuscript. The authors manually reviewed all AI-assisted outputs. During manuscript preparation, AI-generated drafts underwent iterative revision to ensure that they accurately conveyed the intended methodology. We also refined the presentation of pseudocode, citation formatting, and figure captions to maintain consistency between the manuscript and its figures. The authors take full responsibility for the final content of this work, including all text, claims, and assets developed with generative AI assistance.

References

Meshy. Official website, https://www.meshy.ai/.

Ziang Cao, Fangzhou Hong, Zhaoxi Chen, Liang Pan, and Ziwei Liu. Physx-anything: Simulation-ready physical 3d assets from single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5839–5848, June 2026.

Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. PartGen: Part-level 3d generation and reconstruction with multi-view diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5881–5892, June 2025a.

Minghao Chen, Jianyuan Wang, Roman Shapovalov, Tom Monnier, Hyunyoung Jung, Dilin Wang, Rakesh Ranjan, Iro Laina, and Andrea Vedaldi. Autopartgen: Autoregressive 3d part generation and discovery. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, Main Conference, pp. 153496–153521. Curran Associates, Inc., 2025b. doi: 10.52202/085713-5132. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e19b6f65791e350347bcff8a3955cb5b-Paper-Conference.pdf.

Qiuyu Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Dieter Fox, and Abhishek Gupta. URDFormer: A Pipeline for Constructing Articulated Simulation Environments from Real-World Images. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.124.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

earthtojake. text-to-cad. GitHub repository, 2026. URL https://github.com/earthtojake/text-to-cad/releases/tag/v0.5.1. Version 0.5.1. Accessed: 2026-09-12.

Mohamed el Amine Boudjoghra, Ivan Laptev, and Angela Dai. Unfoldart: Zero-shot recovery of full articulated 3d objects from text or image, 2026. URL https://arxiv.org/abs/2606.30608.

Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Sicong Gao, Maurice Pagnucco, Tomasz Bednarz, and Yang Song. Nvidia isaac sim: Enabling scalable, gpu-accelerated simulation for robotics, 2026. URL https://arxiv.org/abs/2606.03551.

Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7081–7091, June 2023.

Yumeng He, Ying Jiang, Jiayin Lu, Yin Yang, and Chenfanfu Jiang. Spark: Sim-ready part-level articulated reconstruction with vlm knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7233–7243, June 2026.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations(ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and ERIC EATON. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 17578–17602, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/2c8047bf3ed8ef6905351608d641f02f-Paper-Conference.pdf.

Ruining Li, Yuxin Yao, Chuanxia Zheng, Christian Rupprecht, Joan Lasenby, Shangzhe Wu, and Andrea Vedaldi. Particulate: Feed-forward 3d object articulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27708–27718, June 2026.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models, 2025. URL https://arxiv.org/abs/2502.06608.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Feng Yiqiang, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, Main Conference, pp. 35387–35415. Curran Associates, Inc., 2025. doi: 10.52202/085713-1189. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/32cc61322f1e2f56f989d29ccc7cfbb7-Paper-Conference.pdf.

Jiayi Liu, Denys Iliash, Angel Chang, Manolis Savva, and Ali Mahdavi Amiri. Singapo: Single image controlled generation of articulated parts in objects. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 97511–97532, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/f2ba7d0d1394609d5ce11853cfd1c9bf-Paper-Conference.pdf.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9704–9715, October 2025b.

Qingming Liu, Xinyue Yao, Shuyuan Zhang, Yueci Deng, Guiliang Liu, Zhen Liu, and Kui Jia. Pact: Part-decomposed single-view articulated object generation, 2026. URL https://arxiv.org/abs/2602.14965.

Ruijie Lu, Yu Liu, Jiaxiang Tang, Junfeng Ni, Yuxiang Wang, Diwen Wan, Gang Zeng, Yixin Chen, and Siyuan Huang. Generating objects with part-articulation from a single image. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, SA Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400721373. doi: 10.1145/3757377.3763893. URL https://doi.org/10.1145/3757377.3763893.

Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Yanwen Guo, Zhuo Chen, and Chunchao Guo. P3-sam: Native 3d part segmentation, 2025. URL https://arxiv.org/abs/2509.06784.

Alex Newman. Claude-mem: Persistent context across sessions for every agent. https://github.com/thedotmack/claude-mem, 2026. GitHub repository, accessed September 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Xiaowen Qiu, Jincheng Yang, Yian Wang, Zhehuan Chen, Yufei Wang, Tsun-Hsuan Wang, Zhou Xian, and Chuang Gan. Articulate anymesh: Open-vocabulary 3d articulated objects modeling. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4585–4606. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/qiu25b.html.

Yuan Shui, Yandong Guan, Zhanwei Zhang, Juncheng Hu, Jing Zhang, Dong Xu, and Qian Yu. Articad: Articulated cad assembly design via multi-agent code generation, 2026. URL https://arxiv.org/abs/2604.10992.

Maxim Tatarchenko, Stephan R. Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3400–3409, 2019. URL https://api.semanticscholar.org/CorpusID:115147778.

Penghao Wang, Siyuan Xie, Hongyu Yan, Xianghui Yang, Jingwei Huang, Chunchao Guo, and Jiayuan Gu. Artllm: Generating articulated assets via 3d llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34281–34291, June 2026a.

Rui-Huan Wang, Si-Tong Wei, Jia-Qi He, Heng-Yi Wei, Baoquan Chen, and Peng-Shuai Wang. adsl: Agentic 3d creation via joint agent-program design, 2026b. URL https://arxiv.org/abs/2608.17975.

Zhuangzhe Wu, Yue Xin, Chengkai Hou, Minghao Chen, Yaoxu Lyu, Jieyu Zhang, and Shanghang Zhang. Urdf-anything+: End-to-end generation for simulation-ready articulated assets, 2026. URL https://arxiv.org/abs/2603.14010.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14419–14429, June 2026.

Xinhao Yan, Jiachen Xu, Yang Li, Changfeng Ma, Yunhan Yang, Chunshi Wang, Zibo Zhao, Zeqiang Lai, Yunfei Zhao, Zhuo Chen, and Chunchao Guo. X-part: high fidelity and structure coherent shape decomposition, 2025. URL https://arxiv.org/abs/2509.08643.

Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y. Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects, 2024. URL https://arxiv.org/abs/2411.07184.

Yunhan Yang, Yufan Zhou, Yuan-Chen Guo, Zi-Xin Zou, Yukun Huang, Ying-Tian Liu, Hao Xu, Ding Liang, Yan-Pei Cao, and Xihui Liu. Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, SA Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400721373. doi: 10.1145/3757377.3763872. URL https://doi.org/10.1145/3757377.3763872.

Yunhan Yang, Yuanchen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yan-pei Cao, and Xihui Liu. Holopart: Generative 3d part amodal segmentation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 52657–52678, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/56280ad5fb53967fd55e4ba1b1cfc418-Paper-Conference.pdf.

Yunhan Yang, Chunshi Wang, Junliang Ye, Yang Li, Zanxin Chen, Zehuan Huang, Yao Mu, Zhuo Chen, Chunchao Guo, and Xihui Liu. Physforge: Generating physics-grounded 3d assets for interactive virtual world. *arXiv preprint arXiv:2605.05163*, 2026b.

Yootta. WorldSimReady-Home. https://huggingface.co/datasets/Yootta/World-SimReady-Home, 2026.

Chuanrui Zhang, Minghan Qin, Yuang Wang, Baifeng Xie, Hang Li, and Ziwei Wang. Simart: Decomposing monolithic meshes into sim-ready articulated assets via mllm. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, SIGGRAPH Conference Papers '26, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725548. doi: 10.1145/3799902.3811173. URL https://doi.org/10.1145/3799902.3811173.

Mandi Zhao, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 668–686, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/028fcbcf85435d39a40c4d61b42c99a4-Paper-Conference.pdf.

Matt Zhou, Ruining Li, Xiaoyang Lyu, Zhaomou Song, Zhening Huang, Chuanxia Zheng, Christian Rupprecht, Andrea Vedaldi, and Shangzhe Wu. Articraft: An agentic system for scalable articulated 3d asset generation, 2026. URL https://arxiv.org/abs/2605.15187.

A Appendix

A.1 Extended Asset Gallery

As shown in Fig. 5, we provide the input images and articulated-motion results for assets generated by our pipeline.These data are provided in the supplementary material.

A.2 Additional experienment

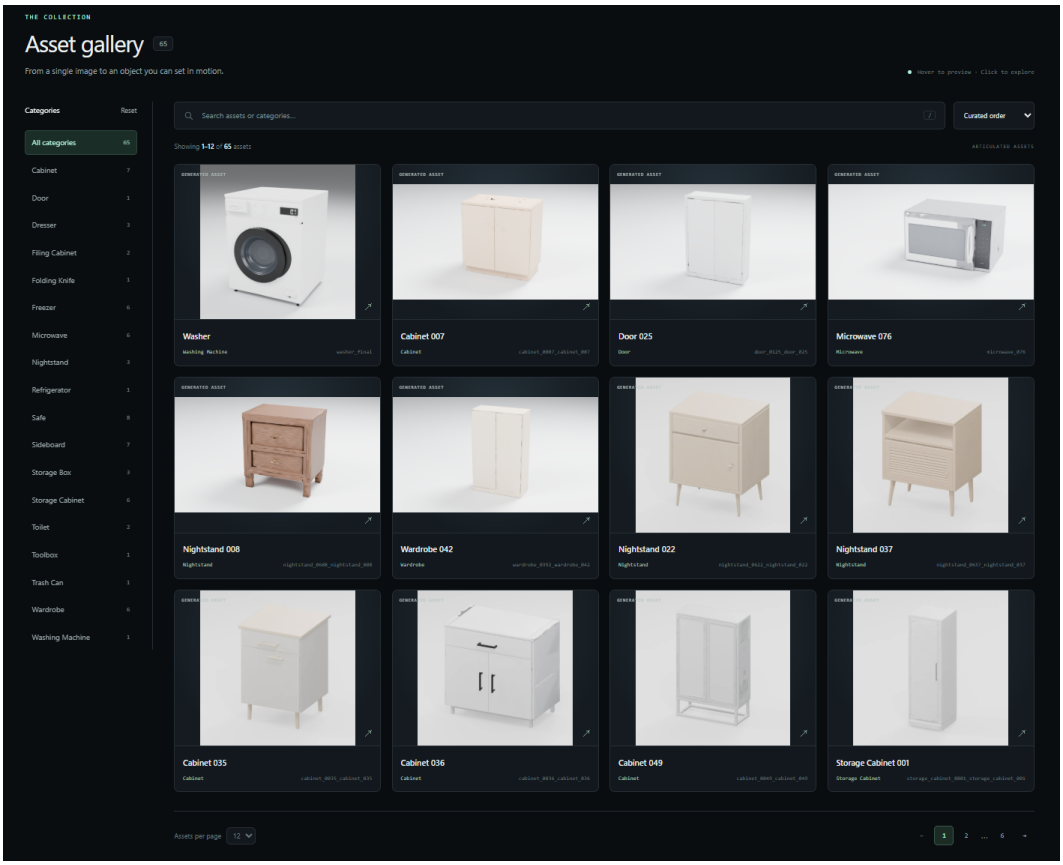


Figure 5: **Extended asset gallery.** Additional examples generated by our pipeline, including the input images and corresponding articulated-motion results.

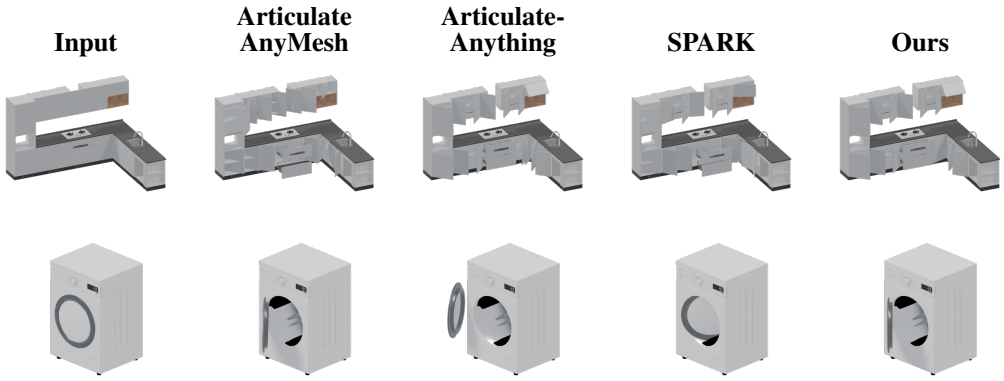


Figure 6: **Qualitative comparison of URDF binding.** We compare the articulation results produced by different methods on in-the-wild objects.

Part geometry qualitative experienment. The qualitative results of our part-level geometry generation are shown in Fig. 7. These results demonstrate that our method can generate geometrically consistent object parts while preserving the overall object structure.

Qualitative Comparison of URDF Binding. As shown in Fig. 6, we qualitatively compare URDF binding results with Articulate AnyMesh, Articulate-Anything, and SPARK. Our method produces

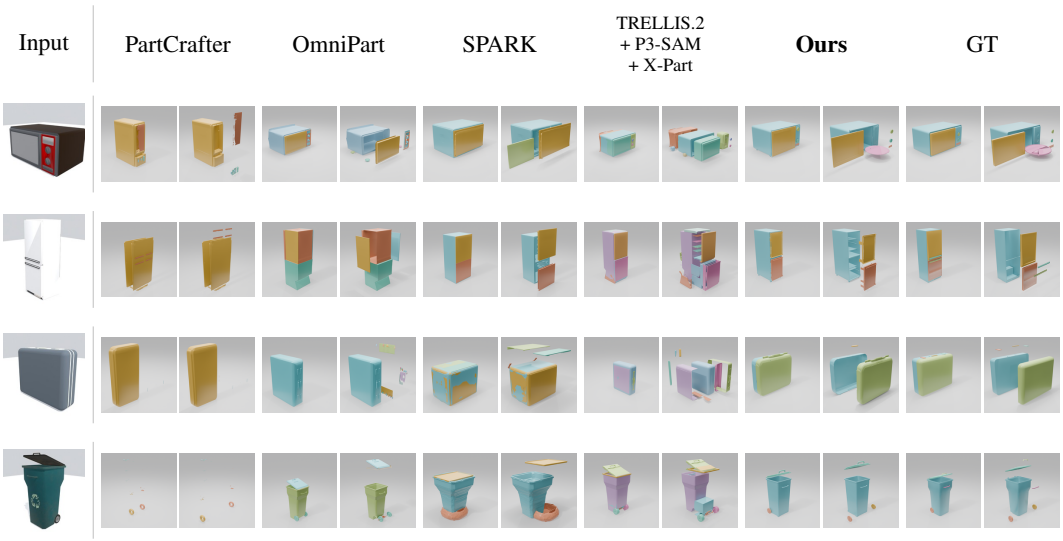


Figure 7: **Qualitative comparison of part-level geometry generation.** Each row shows one object with its input image. Two renderings are displayed side by side for each method and the ground truth (GT).

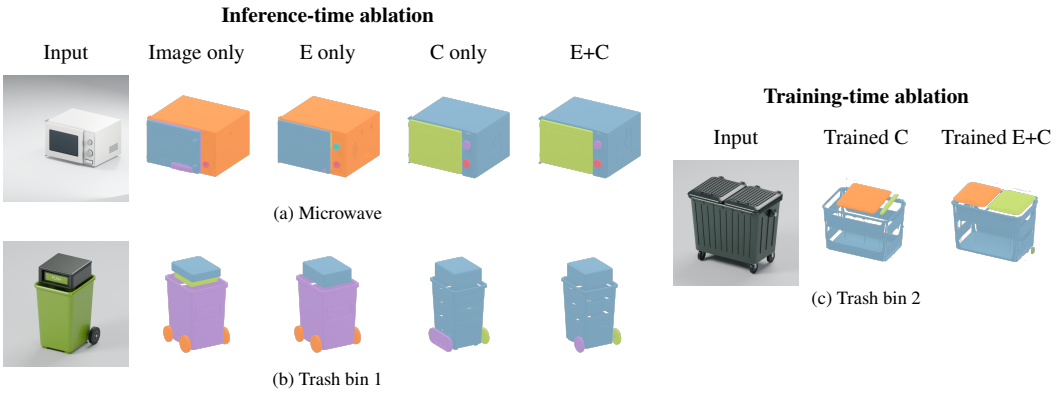


Figure 8: Ablation study results.

more consistent part-to-joint associations and more plausible articulated motions on challenging in-the-wild objects.

Abblation. The qualitative and quantitative ablation results are shown in Fig.8 and Table.3 The results confirm the effectiveness of the E and C conditioning branches.

Additional Baseline Comparisons. The rapid advancement of foundation models has led to a growing body of work on text-to-articulated asset generation. Although these methods differ from ours in input modality, they share the goal of generating articulated assets. We therefore conduct additional comparisons with *ADSL* (Wang et al., 2026b) and *Text2CAD* (earthtojake, 2026). We also compare with *PAct* (Liu et al., 2026) and *SINGAPO* (Liu et al., 2025a), as shown in Fig. 9.

A.3 Agentic Workflow

After obtaining the part-level geometry $\mathcal{M}$, we construct an agentic articulation harness that binds the geometric–kinematic structural prior S to the generated parts and estimates the continuous motion parameters required for an executable articulated structure. The harness establishes correspondences between generated parts and URDF links using part identifiers, and inherits the parent–child

Table 3: **Inference-time ablation of structural conditioning.** Results are macro-averaged across categories. B0 disables both structural branches. Center Err. measures the distance between bounding-box centers.

Variant	F@0.1 ↑	Center Err. ↓
B0	0.142	0.800
E	0.146	0.824
C	0.612	0.236
E+*C*	**0.616**	**0.220**

relations, joint types, and coarse motion axes to construct an initial kinematic tree. The agent then further infers joint axes, origins, and motion ranges by combining the current geometry with motion experience validated in previous cases. Inspired by Claude-Mem (Newman, 2026), we maintain a hierarchical articulation memory that retrieves category-level motion-prior summaries and, when needed, expands them into detailed joint types, axes, pivots, and motion ranges. Finally, these structural and continuous parameters are assembled into an initial URDF $\mathcal{U}^{(0)}$ for subsequent simulation-based validation and closed-loop refinement.

A.4 Simulation and validation

Starting from the initial URDF $\mathcal{U}^{(0)}$, we deploy the generated articulated asset in Isaac Sim (Gao et al., 2026) for physical simulation and render its joint motions for closed-loop validation. A video critic detects errors in motion axis, direction, pivot, range, collisions, and inter-part interactions, and provides feedback to guide the agent in parameter editing, rollback, and retry. If repeated parameter-level refinement still fails validation, the system further falls back to structural re-planning or geometry regeneration. Persistently invalid results are explicitly flagged and discarded, and only validated, high-confidence simulation-ready articulated assets are retained.

A.5 Additional Training Details

Rather than training with a fixed dataset throughout the entire optimization process, we adopt an iterative, failure-driven data scaling strategy. Specifically, we periodically evaluate the model on different object categories using category-wise quantitative metrics and qualitative inspection. Categories with consistently lower reconstruction quality are identified as failure cases, for which additional training data are generated and incorporated into the training set. The model is then resumed from the latest checkpoint with the augmented data.

For example, during training, we observed that the model performed noticeably worse on *toilets* than on most other categories. To address this weakness, we first collected 200 toilet images from the Internet using a web crawler. These images were then processed by the OpenAI `GPT-6` API to generate corresponding programmatic 3D assets and articulation annotations. In total, the data-generation process consumed approximately **6.82M tokens** and produced **191 valid simulation-ready assets** after automatic filtering and validation. These newly generated samples were subsequently added to the training set for targeted category-level data augmentation.

After incorporating the newly generated toilet assets, we continued training for an additional 4,000 optimization steps. As shown in Fig. 10, both the category-level quantitative metrics and qualitative reconstruction quality improved after targeted data augmentation. In particular, the updated model better preserves the overall geometry and characteristic structural details of toilets. This example illustrates how our training pipeline progressively expands the dataset according to observed model failure cases instead of relying on a one-shot, statically constructed training corpus.

B Application

Our generated assets can be directly deployed in simulation and interactive environments for downstream applications.Our generated articulated microwave is deployed in a household scene from World-SimReady-Home (Yootta, 2026), where a robotic arm performs door-opening and knob-

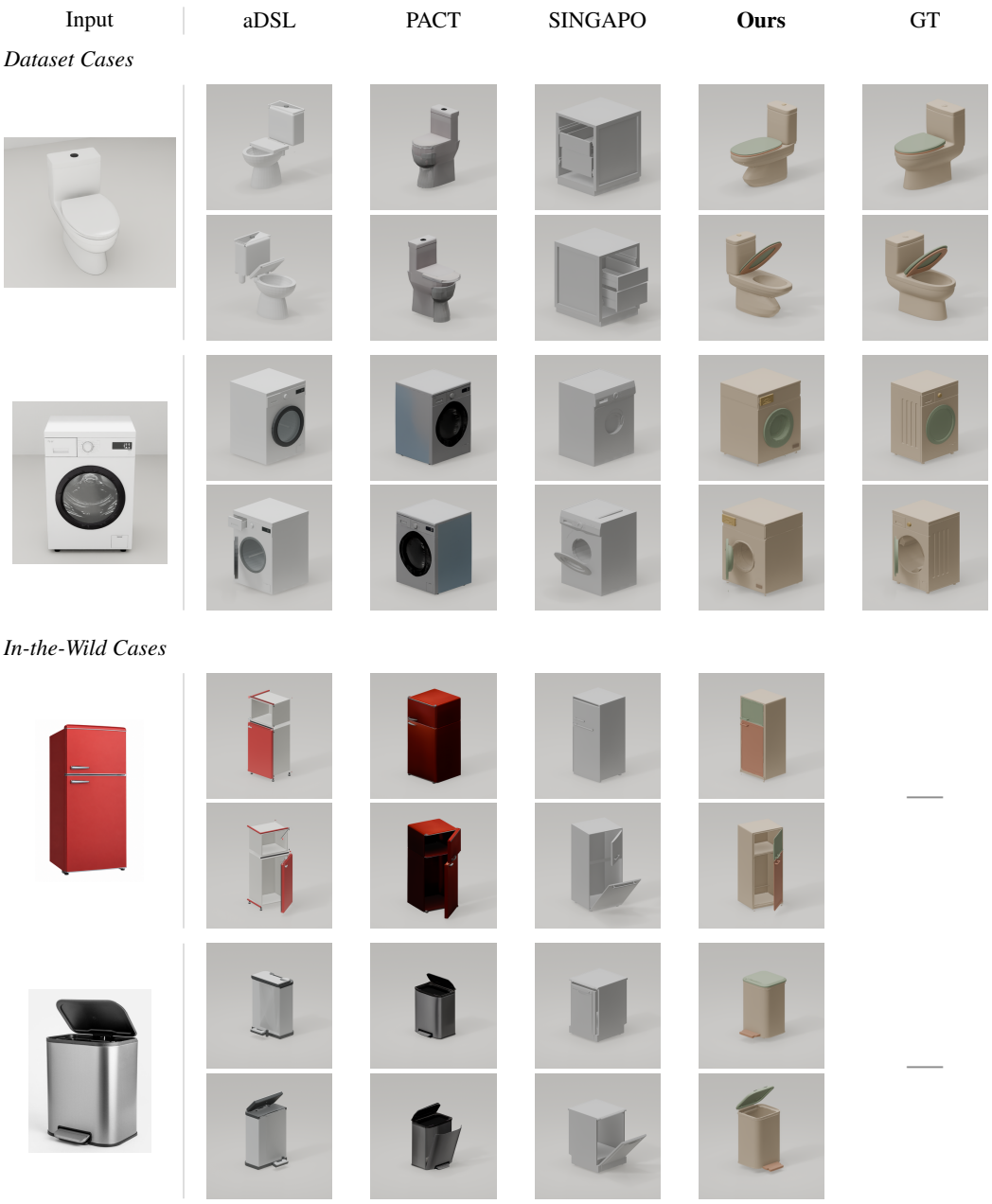
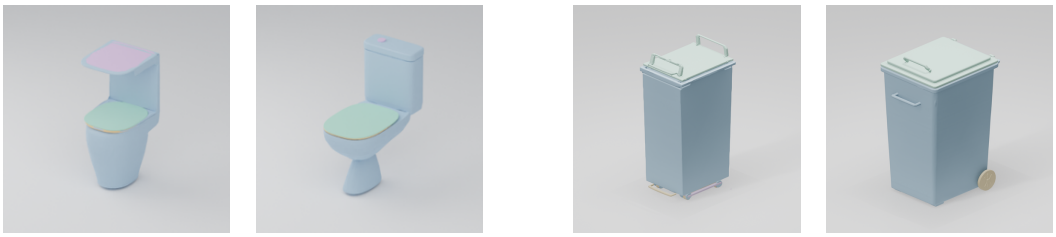


Figure 9: **Additional qualitative comparisons.** Recovering detailed geometry from a single image remains a major challenge for text-to-articulated-asset generation methods. For in-the-wild cases, the key difficulty further lies in faithfully reconstructing diverse articulated geometries across object categories.

turning tasks. As shown in Fig. 11, our pipeline converts a single image into a simulation-ready articulated asset with part-level meshes, PBR materials, and URDF-defined kinematics. These assets support robotic manipulation in Isaac Sim, such as door opening and drawer pulling.



(a) Case 1 toilet **(b) Case 2** trashcan

Figure 10: **Qualitative comparison before and after targeted training.** For each case, we show the generated geometry before (left) and after (right) additional training with category-specific data.



Figure 11: **Robotic manipulation.** The generated simulation-ready microwave is directly deployed for robotic interaction. A robot arm opens the door through multiple intermediate states and subsequently turns the control knob, demonstrating executable articulated motion.