MotionAnymesh: Physics-Grounded Articulation for Simulation-Ready Digital Twins

WenBo Xu[1], Liu Liu[1]✉, Li Zhang[2], Dan Guo[1], and RuoNan Liu[3]

[1] Hefei University of Technology
[2] The Hong Kong Polytechnic University
[3] Shanghai Jiao Tong University

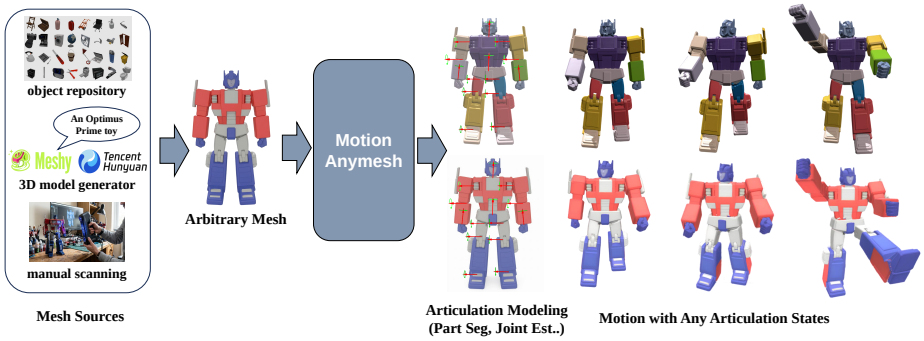

Fig. 1: MotionAnymesh: Physics-Grounded Articulation of Static 3D Assets. While current SOTAs (e.g., Articulate-AnyMesh) rely on ungrounded semantics and often suffer from severe inter-penetration, MotionAnymesh ensures physically plausible articulation. Through *kinematic-aware perception* and *physics-constrained optimization*, our zero-shot framework transforms static meshes into collision-free, simulation-ready URDF digital twins for direct deployment in embodied AI tasks.

Abstract. Converting static 3D meshes into interactable articulated assets is crucial for embodied AI and robotic simulation. However, existing zero-shot pipelines struggle with complex assets due to a critical lack of physical grounding. Specifically, ungrounded Vision-Language Models (VLMs) frequently suffer from kinematic hallucinations, while unconstrained joint estimation inevitably leads to catastrophic mesh inter-penetration during physical simulation. To bridge this gap, we propose **MotionAnymesh**, an automated zero-shot framework that seamlessly transforms unstructured static meshes into simulation-ready digital twins. Our method features a kinematic-aware part segmentation module that grounds VLM reasoning with explicit SP4D physical priors, effectively eradicating kinematic hallucinations. Furthermore, we introduce a geometry-physics joint estimation pipeline that combines robust

✉ Corresponding author.

type-aware initialization with physics-constrained trajectory optimization to rigorously guarantee collision-free articulation. Extensive experiments demonstrate that MotionAnymesh significantly outperforms state-of-the-art baselines in both geometric precision and dynamic physical executability, providing highly reliable assets for downstream applications. Our project page is available at `https://xwb0117.github.io/Motionanymesh/`

Keywords: Articulated Object Modeling · Kinematic Segmentation · 3D Assets

1 Introduction

Articulated objects are ubiquitous in human daily environments and interactions. Constructing high-fidelity digital twins of these objects is crucial for applications such as embodied AI [1, 5–7, 14, 15, 18], robotic manipulation simulation [10, 21, 42], and virtual/augmented reality (VR/AR) [17]. However, a significant gap exists between the richness of real-world interaction data and the scarcity of assets in simulation environments. In current large-scale open-source 3D asset libraries (such as Objaverse [4]), the vast majority of models are purely static meshes, lacking the kinematic structures necessary for interaction, part-level physical boundary segmentation, and precise joint parameters. Traditionally, converting this vast amount of static assets into URDF models that support physically simulated manipulation requires extremely time-consuming and costly manual modeling. Therefore, how to automatically achieve the "static-to-articulated" transformation of 3D assets has become a critical challenge that urgently needs to be addressed in the fields of computer vision and graphics.

Existing pipelines [20,30,35] for articulated object parsing are predominantly bottlenecked by two fundamental flaws. First, they commonly rely on 2D-to-3D mask lifting strategies (e.g., performing 2D segmentation on rendered views and projecting back to 3D). This view-dependent mechanism inherently fragments the geometric continuity of the 3D shape, inevitably yielding jagged part boundaries and failing completely on self-occluded internal structures. Second, when directly employing Vision-Language Models (VLMs) for open-vocabulary part decomposition, the models rely heavily on semantic priors rather than physical constraints. When faced with complex or irregular mechanical components lacking explicit semantic names, the VLM frequently suffers from severe kinematic hallucinations, either erroneously merging distinct active parts or over-segmenting monolithic structures. Furthermore, existing generative or regression-based approaches [3, 9, 13, 20, 22, 24, 35, 39] for joint parameter estimation typically lack strict spatial and physical constraints. Although the predicted joint axes or origins may appear plausible in isolation, even a marginal geometric deviation will violently accumulate during long-range actuation. Consequently, when imported into rigorous physics engines like SAPIEN [42], these assets frequently violate non-interpenetration constraints, resulting in catastrophic non-physical behaviors such as severe mesh collision, structural detachment, or kinematic freez-

ing. These limitations highlight a critical gap between current visual perception pipelines and genuinely physically executable digital twins.

To address these fundamental limitations, we propose **MotionAnymesh**, a unified zero-shot framework that reconstructs simulation-ready articulated structures from static meshes (as illustrated in Fig. 1). To eliminate geometric fragmentation while avoiding VLM hallucinations, we strategically decouple boundary extraction from semantic reasoning. We first extract fine-grained geometric primitives directly in the 3D-native space, rigorously preserving sharp and physically valid boundaries. Subsequently, rather than relying on pure semantics, we introduce explicit multi-view kinematic priors derived from SP4D [46]. By grounding the VLM with these physical cues and multi-view observations, we guide it to assemble the 3D-native primitives like following a physical "assembly manual", effectively eradicating kinematic hallucinations. Building upon the recovered part hierarchy, we introduce a rigorous two-stage joint estimation pipeline to guarantee physical executability. Initially, a *Type-Aware Kinematic Initialization* deduces robust initial parameters from localized contact interfaces via PCA and robust 2D RANSAC fitting. Subsequently, a *Physics-Constrained Trajectory Optimization* refines these axes by minimizing unified surface distances. During virtual articulation, any geometric inter-penetration induces strong asymmetric penalties, forcing the joint parameters to systematically correct micro-misalignments and converge toward absolutely valid motion manifolds.

We conduct extensive experiments across diverse benchmarks. Quantitative and qualitative results demonstrate that MotionAnymesh significantly outperforms existing zero-shot baselines, producing geometrically precise and strictly collision-free URDF assets ready for embodied AI downstream tasks. The contributions of this work can be summarized as follows:

1. We introduce **MotionAnymesh**, a novel zero-shot pipeline that directly converts static 3D assets into simulation-ready articulated objects, bridging the gap between geometric perception and interactive physical simulation.
2. We propose a robust perception-to-actuation methodology: a Kinematic-Aware Part Segmentation that integrates SP4D priors with VLM reasoning to eliminate hallucinations, and a Geometry-Physics Joint Estimation pipeline, combining robust type-aware initialization with physics-constrained optimization to strictly guarantee collision-free kinematics.
3. We establish a comprehensive evaluation protocol, demonstrating through extensive experiments that our method achieves superior performance.

2 Related Work

2.1 Articulated object modeling

Existing methods for articulated object modeling broadly fall into three categories. First, visual reconstruction methods [11,16,25,29,36,40] precisely capture kinematics but require dense spatio-temporal observations (e.g., multi-view or

multi-state inputs), limiting their scalability to massive single-state static 3D assets. Second, generative and procedural approaches [3,24,26,32,37] rely on predefined mesh libraries or templates, which hinders their generalization to diverse, open-world shapes. Recently, foundation-model-driven pipelines have enabled open-vocabulary articulation but struggle with complex assets. Methods heavily reliant on Vision-Language Models (VLMs) or 3D LLMs [20, 23, 35, 44] lack physical grounding, making them highly susceptible to kinematic hallucinations when parsing irregular mechanical components without explicit semantic names. Alternatively, approaches leveraging synthesized generative priors (e.g., diffusion or motion videos) [2, 12, 30] are prone to geometric inter-penetration and motion distortion under strict 3D topological constraints. Consequently, they fail to provide the high-precision physical guidance required for robust parameter optimization on objects with deeply nested internal structures.

2.2 Part aware 3D generation

Instead of generating a 3D object as a whole, part aware 3D generation methods generate objects with part-level information. Recent part-aware 3D generation methods aim to synthesize objects with explicit compositional structures. Early approaches [8, 33, 41, 43] predominantly rely on Variational Autoencoders (VAEs) or graph-based networks to explicitly disentangle global structural topology from detailed part geometries. More recently, diffusion-based paradigms [19, 34] have gained traction, leveraging powerful generative priors to synthesize high-fidelity part latents. Additionally, recent zero-shot 3D segmentation networks [27, 28, 31, 38, 45, 48] excel at identifying geometric boundaries. However, lacking kinematic priors, their operations based purely on local features inevitably lead to the over-segmentation of meshes. To address this lack of kinematic awareness, another line of research specifically targets the kinematic segmentation of articulated objects, recent foundation-model-driven pipelines [20, 30, 35] employ Vision-Language Models (VLMs) to jointly infer semantic parts and kinematic structures. Yet, these paradigms introduce their own bottlenecks. Pipelines relying on 2D-to-3D segmentation lifting inevitably destroy geometric purity, resulting in jagged physical boundaries and failing on heavily occluded interiors. Furthermore, purely semantic-driven segmentation frequently suffers from severe kinematic hallucinations when confronted with fine-grained, irregular mechanical components that lack explicit semantic labels.

3 Method

3.1 Overview

As illustrated in Fig. 2, MotionAnymesh transforms static 3D meshes into simulation-ready articulated digital twins through three integrated stages: (1) *Kinematic-Aware Part Segmentation* (Sec. 3.2), which extracts geometrically pure primitives and clusters them using SP4D-guided VLM reasoning; (2) *Joint Estimation*

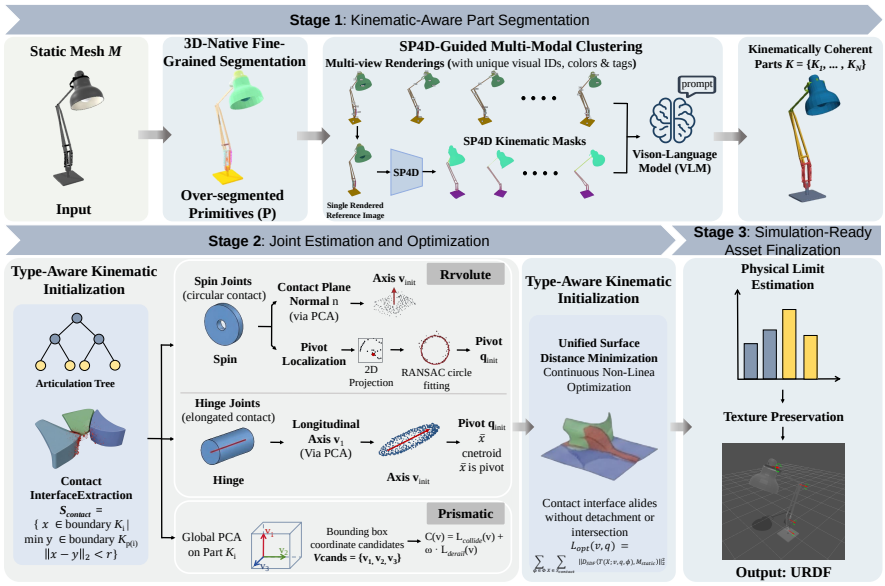

Fig. 2: Overview of the **MotionAnymesh** framework. Our pipeline consists of three integrated stages: (1) Kinematic-Aware Part Segmentation, which extracts 3D-native primitives and clusters them using SP4D kinematic priors and VLM reasoning; (2) Joint Estimation and Optimization, featuring type-aware geometric initialization and physics-constrained trajectory refinement to ensure collision-free articulation ; and (3) Simulation-Ready Asset Finalization, providing simulation-ready URDF models with preserved textures.

and Optimization (Sec. 3.3), which deduces robust joint parameters and strictly guarantees collision-free kinematics via unified trajectory optimization; and (3) *Simulation-Ready Asset Finalization* (Sec. 3.4), which determines physical limits and preserves textures to output high-fidelity URDF models.

3.2 Kinematic-Aware Part Segmentation

3D-Native Fine-Grained Segmentation. To preserve geometric purity, we operate natively in the 3D space. Given the input static mesh $\mathcal{M}$, we employ a 3D-native foundation model, P3-SAM [31], to extract low-level geometric boundaries based on spatial concavities and structural connectivity. This process over-segments the mesh into a set of geometrically pure, disjoint primitives $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$. While these primitives preserve perfect physical boundaries, they lack high-level kinematic semantics (e.g., a single drawer might be fractured into a handle, a front panel, and sliding rails).

SP4D-Guided Multi-Modal Clustering. To assemble the fragmented primitives $\mathcal{P}$ into kinematically coherent movable parts $\mathcal{K} = \{K_1, \ldots, K_N\}$, we formulate a multi-modal clustering process. While large Vision-Language Models

(VLMs) demonstrate strong semantic reasoning capabilities, relying exclusively on them often leads to severe kinematic hallucinations, especially when encountering complex mechanical components devoid of explicit semantic labels. Therefore, we introduce explicit kinematic priors derived from SP4D [46] to ground the VLM's reasoning in physical reality.

Specifically, we render the over-segmented mesh from multiple viewpoints, projecting the 3D primitives $\mathcal{P}$ into 2D images and assigning a distinct visual ID (unique colors and numerical tags) to each primitive. Simultaneously, we leverage SP4D to extract explicit kinematic priors. By simply feeding a single rendered reference image of the object into SP4D, it infers and synthesizes consistent multi-view kinematic segmentation masks. These masks explicitly highlight the coarse functional regions of movable parts across different viewing angles. By feeding both the multi-view primitive images (with visual IDs) and the corresponding SP4D kinematic masks into the VLM, we prompt the model to visually correlate the fine-grained geometric IDs with the coarse kinematic regions. We leverage the VLM as a a constrained spatial reasoning agent: it groups 3D-native primitive IDs under SP4D motion priors, but does not directly regress joint parameters. It cross-references the purely geometric primitives against the explicit physical priors, effectively grouping the fractured pieces into coherent, functional kinematic sets (detailed prompts are provided in the Supplementary Material). The final aggregated parts $K_i = \bigcup_{j \in \mathcal{I}_i} p_j$ (where $\mathcal{I}_i$ is the index set of primitives belonging to the i-th kinematic part) possess both pristine 3D geometric boundaries and accurate kinematic hierarchies.

3.3 Joint Estimation and Optimization

With the mesh successfully decomposed into kinematically coherent parts $\mathcal{K} = \{K_0, K_1, \ldots, K_N\}$, these isolated components still lack the structural connectivity and motion semantics required for physical simulation. To achieve this, we harness the common-sense reasoning capabilities of the Vision-Language Model (VLM) to construct a comprehensive *articulation tree*. Building upon the visual identifiers assigned during the clustering stage, we prompt the VLM to infer the parent-child dependencies among the parts and to predict the corresponding semantic joint type for each movable component. Based on visual cues and mechanical priors, the VLM broadly classifies the joints into two primary categories: Revolute (further sub-categorized into Spin and Hinge) and Prismatic. Guided by this VLM-derived articulation tree, we deduce the precise physical joint parameters. To ensure physically plausible articulation, we propose a two-stage, type-aware geometric approach: *type-aware kinematic initialization* followed by *physics-constrained trajectory optimization.*

Contact Interface Extraction. Regardless of the joint type, the physical mechanism is intrinsically anchored at the spatial intersection between an articulated part K_i and its parent $K_{P(i)}$. We first extract the contact point cloud

$S_{contact}$ by identifying vertices on K_i that are in close proximity to $K_{P(i)}$:

$$S_{contact} = \left\{ \mathbf{x} \in \partial K_i \;\middle|\; \min_{\mathbf{y} \in \partial K_{P(i)}} \|\mathbf{x} - \mathbf{y}\|_2 < \tau \right\} \tag{1}$$

where ∂K denotes the surface vertices, and τ is a predefined distance threshold, empirically set to 0.01 m.

Type-Aware Kinematic Initialization. With the contact interface extracted, we employ distinct geometric strategies to deduce the initial joint parameters tailored to their specific mechanical behaviors:

- **Revolute Joints:** For rotational mechanisms, the kinematics are fundamentally defined by a rotational axis $\mathbf{v}_{init}$ and a pivot point q_{init}. Depending on the geometric nature of the contact interface, we categorize revolute joints into two subtypes and adopt tailored initialization strategies:
  - *Spin Joints:* For spin mechanisms (e.g., wheels, knobs), the visible mechanical connection is typically characterized by an axially symmetric intersection boundary. In most surface meshes, the extracted contact point cloud $S_{contact}$ between the spin part and its base forms a disc-like or shallow annular band. By applying Principal Component Analysis (PCA) to $S_{contact}$, we evaluate its spatial variance. Since these localized intersection points distribute predominantly along the radial cross-section rather than the axial depth, the eigenvector corresponding to the smallest eigenvalue $\mathbf{n}$ robustly captures the orthogonal normal of this distribution. We assign this normal directly as the initial rotational axis ($\mathbf{v}_{init} = \mathbf{n}$).

    To accurately localize the rotational pivot q_{init} against surface noise, we propose a robust 2D-projection and RANSAC-based fitting pipeline. Let $\bar{\mathbf{x}}$ be the geometric centroid of $S_{contact}$. We establish a local 2D orthogonal coordinate system using two unit vectors $\mathbf{b}_1$ and $\mathbf{b}_2$, both perpendicular to $\mathbf{n}$. Every 3D contact point $\mathbf{x} \in S_{contact}$ is projected into this 2D local space to obtain its 2D coordinates $\mathbf{p}$:

    $$\mathbf{p} = \begin{bmatrix} (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{b}_1 \\ (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{b}_2 \end{bmatrix} \tag{2}$$

    Geometrically, the contact boundary of a spin joint must exhibit rotational symmetry around its axis. Consequently, when projected onto the local 2D plane perpendicular to $\mathbf{n}$, the valid contact points naturally distribute along a circular locus. To rigorously deduce this center while rejecting outliers and handling partial observations, we employ the RANSAC algorithm to iteratively fit a 2D circle to the projected points $\mathbf{p} = \{\mathbf{p}_i\}_{i=1}^N$. We evaluate the hypothesized circle (center $\mathbf{c}_{2D}$, radius r) by maximizing the consensus set (inliers):

    $$\mathbf{c}_{2D}^*, r^* = \arg\max_{\mathbf{c}_{2D}, r} \sum_{i=1}^{N} \mathbb{I}\left(\left| \|\mathbf{p}_i - \mathbf{c}_{2D}\|_2 - r \right| < \delta \right) \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function and δ is the inlier distance threshold that accommodates surface noise and quantization errors. In our implementation, δ is empirically set to 0.005 m. Finally, the optimal 2D center $\mathbf{c}_{2D}^* = [x_c, y_c]^T$ is mapped back into the global 3D space to serve as the highly accurate joint pivot q_{init}:

$$q_{init} = \bar{\mathbf{x}} + x_c \mathbf{b}_1 + y_c \mathbf{b}_2 \tag{4}$$

- *Hinge Joints:* Unlike spin mechanisms, the contact regions of hinge joints (e.g., door hinges, laptop screens) are characteristically distributed along the axis of rotation, exhibiting a dominant longitudinal span. This holds true whether the physical connection is a continuous strip or comprises multiple discrete anchor points. Based on this mechanical prior, we propose a robust PCA-based initialization. We compute the covariance matrix of the zero-centered contact point cloud $S_{contact}$. Regardless of the local patch shapes, the primary eigenvector $\mathbf{v}_1$ (corresponding to the largest eigenvalue) intrinsically captures the direction of maximum spatial variance of the entire point set, perfectly representing the global longitudinal hinge line. Thus, we assign the initial rotational axis as $\mathbf{v}_{init} = \mathbf{v}_1$. Furthermore, because such contact distributions are generally symmetric along their rotational axis, we directly assign the geometric centroid $\bar{\mathbf{x}}$ of the point cloud as the optimal mechanical pivot ($q_{init} = \bar{\mathbf{x}}$).

– **Prismatic Joints:** Unlike revolute joints, the sliding axis of a prismatic joint is dictated by the global geometry of the part rather than the localized contact patch. We first perform global PCA on the entire point cloud of K_i to boldly extract its three local bounding-box coordinate axes $\mathcal{V}_{cands} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ as candidates.
To determine the true sliding direction among these candidates, we propose a *Normalized Dual-Penalty Trajectory Verification* mechanism. The physical intuition is that a correctly estimated prismatic axis must allow the part to slide without penetrating the parent body (collision-free) while strictly adhering to its mechanical rails (derailment-free). We define a set of discrete translation steps $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ covering both forward and backward movements. For each candidate axis $\mathbf{v} \in \mathcal{V}_{cands}$ and step $d \in \mathcal{D}$, the virtually translated part is denoted as $K_i(d, \mathbf{v}) = \{\mathbf{x} + d\mathbf{v} \mid \mathbf{x} \in K_i\}$.
We evaluate a comprehensive scoring function $\mathcal{C}(\mathbf{v})$ that penalizes both geometric collision ($\mathcal{L}_{collide}$) and mechanical derailment ($\mathcal{L}_{derail}$):

$$\mathcal{C}(\mathbf{v}) = \mathcal{L}_{collide}(\mathbf{v}) + \omega \cdot \mathcal{L}_{derail}(\mathbf{v}) \tag{5}$$

Specifically, the collision penalty $\mathcal{L}_{collide}$ computes the normalized ratio of points on the moved part that geometrically inter-penetrate the static parent $K_{P(i)}$. It is defined via an indicator function $\mathbb{I}(\cdot)$ to count penetration events:

$$\mathcal{L}_{collide}(\mathbf{v}) = \frac{1}{|\mathcal{D}||K_i|} \sum_{d \in \mathcal{D}} \sum_{\mathbf{x}' \in K_i(d,\mathbf{v})} \mathbb{I}\left(\min_{\mathbf{y} \in K_{P(i)}} \|\mathbf{x}' - \mathbf{y}\|_2 < \epsilon_c \right) \tag{6}$$

where ϵ_c is a strict distance threshold to robustly define a penetration event. In our implementation, ϵ_c is empirically set to 0.005 m. Conversely, the derailment penalty $\mathcal{L}_{derail}$ ensures the moving part does not detach from its original sliding track. It measures the average absolute divergence distance from the original stationary contact points $\mathcal{S}_{contact}$ to the newly translated surface:

$$\mathcal{L}_{derail}(\mathbf{v}) = \frac{1}{|\mathcal{D}||\mathcal{S}_{contact}|} \sum_{d \in \mathcal{D}} \sum_{\mathbf{p} \in \mathcal{S}_{contact}} \min_{\mathbf{x}' \in K_i(d,\mathbf{v})} \|\mathbf{p} - \mathbf{x}'\|_2 \tag{7}$$

To equalize the magnitude differences between the dimensionless probability $\mathcal{L}_{collide}$ and the absolute spatial deviation $\mathcal{L}_{derail}$, ω serves as a crucial balancing hyperparameter. We empirically set $\omega = 20$ to align their scales. The axis $\mathbf{v}^* = \arg\min_{\mathbf{v} \in \mathcal{V}_{cands}} \mathcal{C}(\mathbf{v})$ that yields the minimum dual-penalty cost is selected as the definitive sliding direction. Pivot points are mathematically undefined and unnecessary for prismatic joints.

Physics-Constrained Trajectory Optimization. Although the initialized parameters provide a plausible mechanical layout, subtle surface misalignments (e.g., PCA-derived axes slightly tilted relative to true rails) may induce friction or penetration during long-range motion. To guarantee simulation-ready assets, we formulate a continuous non-linear optimization to refine joint parameters.

To achieve this, we enforce a *Unified Surface Distance Minimization* constraint across all joint types. The core physical intuition is that during articulation, the contact interface $S_{contact}$ should act as a perfect mechanical bearing or sliding rail, seamlessly gliding against the parent surface without detachment or intersection. Let $\mathcal{T}(\mathbf{x}; \mathbf{v}, \mathbf{q}, \phi)$ denote the 3D rigid transformation applied to a contact point $\mathbf{x} \in S_{contact}$ during motion. For Revolute joints, $\phi = \theta$ represents a rotation angle around $\mathbf{v}$ anchored at $\mathbf{q}$. For Prismatic joints, $\phi = d$ represents a translation distance along $\mathbf{v}$, reducing the transformation to $\mathcal{T}(\mathbf{x}; \mathbf{v}, d) = \mathbf{x} + d \cdot \mathbf{v}$.

To prevent geometric inter-penetration and maintain tight kinematic constraints, we simulate the part's motion across a set of discrete virtual states Φ (representing angular steps Θ for revolute joints, or spatial steps D for prismatic joints). We minimize the unified trajectory deviation loss $\mathcal{L}_{opt}$:

$$\mathcal{L}_{opt}(\mathbf{v}, \mathbf{q}) = \sum_{\phi \in \Phi} \sum_{\mathbf{x} \in S_{contact}} \left\| \mathcal{D}_{SDF}\big(\mathcal{T}(\mathbf{x}; \mathbf{v}, \mathbf{q}, \phi), \mathcal{M}_{static}\big) \right\|_2^2 \tag{8}$$

where $\mathcal{D}_{SDF}(\cdot, \cdot)$ computes the Signed Distance Field (SDF) relative to the static environment $\mathcal{M}_{static}$. Optimized via the Levenberg-Marquardt algorithm, this unified term rigorously ensures that the moving part maintains minimal, uniform proximity to the static base throughout its entire range of motion, guaranteeing physically valid, collision-free kinematics for both rotational and translational mechanisms.

3.4 Simulation-Ready Asset Finalization

To render the generated digital twin fully simulation-ready, two critical final steps are required: determining the valid Range of Motion (RoM) for each joint and finalizing the asset's visual fidelity for standard URDF assembly.

Physical Limit Estimation. Unconstrained joints can lead to unrealistic behaviors in physics simulators, such as doors rotating into walls or drawers flying out of cabinets. We employ a forward-simulation collision detection mechanism to automatically deduce the kinematic limits. For *Revolute joints*, starting from the rest state (0°), we incrementally rotate the movable part in both positive and negative directions up to a maximum of ±180°. The joint limits $[\theta_{min}, \theta_{max}]$ are strictly defined by the exact angles where a geometric collision (evaluated via mesh intersection) occurs with the static base. For *Prismatic joints*, the physical constraints are two-fold. In the inward pushing direction, the limit is typically bounded by collision (e.g., a drawer hitting the cabinet's backplate). In the outward pulling direction, we propose a novel *contact-loss* criterion. We incrementally translate the part along its optimized axis $\mathbf{v}^*$ and continuously monitor the mutual contact area with the parent part. The maximum extension limit d_{max} is determined at the point where this contact area drops to zero, physically simulating the moment a sliding component detaches from its rails.

Texture Preservation and Generative Re-texturing. To ensure visual fidelity, our pipeline inherently preserves the original UV mapping and texture information of the input mesh. If the source asset is textured, the decomposed parts naturally retain their original appearance in the final output. However, because true 3D segmentation inevitably exposes texture-less internal cross-sections, and to support customized visual editing, we introduce an optional generative re-texturing module. Leveraging state-of-the-art 3D generative models like Hunyuan3D [47], we can independently generate entirely new, high-fidelity textures for each isolated part based on user-defined text prompts. Ultimately, these textured components, alongside their optimized kinematic parameters, are automatically assembled into a standard URDF format, delivering collision-free, photorealistic articulated assets ready for immediate deployment in interactive simulation environments.

4 Experiments

Implementation Details. We employ P3-SAM [31] for 3D-native geometric primitive extraction and SP4D [46] to generate multi-view kinematic masks, with GPT-4o serving as our core VLM. For trajectory optimization, we utilize Trimesh to compute the SDF and solve the non-linear objective via SciPy's Nelder-Mead algorithm. Hunyuan3D [47] is integrated for optional generative re-texturing. Finally, the optimized parameters and processed meshes are compiled into simulation-ready URDF assets. All experiments are executed on three NVIDIA RTX 4090 GPUs.

Datasets. To evaluate our zero-shot framework, we construct a test suite from three sources: (1) **PartNet-Mobility** [42], providing standard ground-truth

URDF annotations for exact metric calculations; (2) **Objaverse** [4], featuring diverse, open-vocabulary static meshes without predefined templates; and (3) **Generative 3D Assets** from recent Text/Image-to-3D models to stress-test robustness. To enable quantitative comparisons on the latter two unstructured domains, we manually authored their ground-truth URDFs, including part masks and joint parameters.

Baselines. We compare against five state-of-the-art 3D articulation approaches: Articulate-Anything [20], Articulate-AnyMesh [35], URDFormer [3], SINGAPO [24], and PARIS [25].

Metrics. To evaluate the generated articulated digital twins, we employ a combination of static geometric metrics and dynamic physical simulation tests: **1) Part Segmentation Quality:** We measure the **mean Intersection over Union (mIoU)** between the predicted movable parts and ground-truth meshes to assess geometric precision. Additionally, we report **Count Accuracy (Count Acc)**, defined as the percentage of test instances where the inferred number of active kinematic links perfectly matches the ground truth, which validates the topological reasoning capability of our VLM-guided clustering. **2) Joint Parameter Estimation:** We benchmark the estimated kinematic parameters against ground-truth URDF annotations using three metrics: (i) **Joint Type Error**: a binary metric indicating the misclassification of the joint type; (ii) **Joint Axis Error**: the angular deviation between predicted and ground-truth joint axes, normalized within $[0, \pi]$; and (iii) **Joint Pivot Error**: the L_2 Euclidean distance between predicted and actual joint pivot locations. **3) Physical Executability:** To guarantee dynamic stability beyond static parameters, we load the generated URDF assets into the SAPIEN physics simulator [42]. Physical executability is defined as the success rate of URDFs that can be fully actuated along their valid ranges without exhibiting catastrophic non-physical behaviors, such as severe mesh inter-penetration, structural detachment, or kinematic freezing. This critically verifies the asset's suitability for downstream sim-to-real applications.

4.1 Comparison with Baselines

We comprehensively evaluate MotionAnymesh against state-of-the-art articulation frameworks, including Articulate-Anything [20], Articulate-AnyMesh [35], SINGAPO [24], URDFormer [3], and PARIS [25]. The quantitative results across diverse 3D assets are summarized in Table 1. Our qualitative comparisons in Figure 3 focus on the most competitive zero-shot foundation-model-driven baselines (Articulate-Anything and Articulate-AnyMesh).

Part Segmentation. MotionAnymesh significantly outperforms all baselines in geometric decomposition, achieving an mIoU of **0.86** and a Count Accuracy of **0.92**, surpassing the strongest baseline (Articulate-AnyMesh) by massive absolute margins of $+$**0.27** and $+$**0.18**, respectively. Existing retrieval-based methods (URDFormer, Articulate-Anything, SINGAPO) are fundamentally bottlenecked by predefined CAD libraries. As illustrated by Articulate-Anything in Fig. 3, while such paradigms succeed on in-domain objects (e.g., the *storage*),

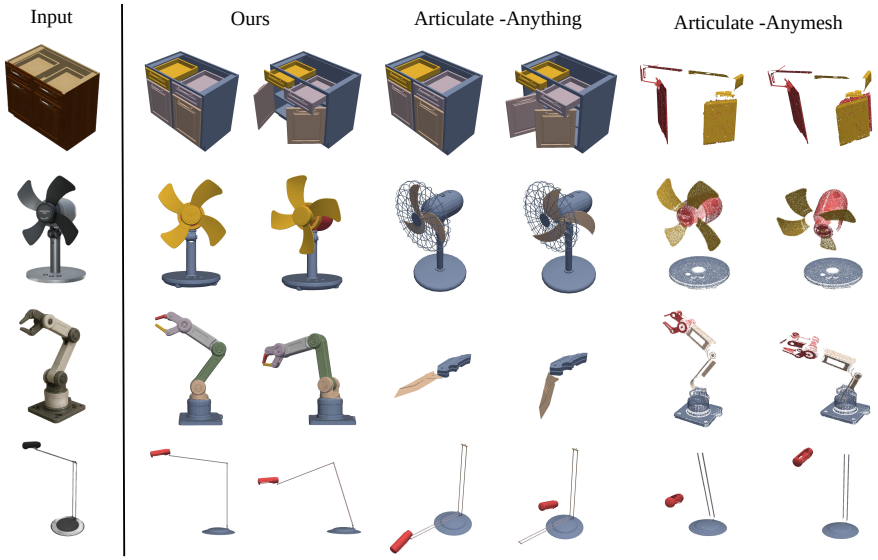

Fig. 3: Qualitative comparison of articulated object modeling. Compared with Articulate-Anything and Articulate-AnyMesh, MotionAnymesh produces cleaner geometric boundaries and more accurate kinematic structures, especially for complex mechanical objects like the robot arm and multi-part lamps.

they catastrophically fail on novel geometries like the *robot_ arm*, yielding severe boundary mismatches or entirely irrelevant structures. Conversely, the zero-shot Articulate-AnyMesh relies heavily on open-vocabulary VLM reasoning, suffering from severe over-segmentation on irregular internal mechanisms lacking explicit semantic names. In contrast, our 3D-native approach, grounded by explicit SP4D kinematic priors, rigorously ensures topologically sound and geometrically intact segmentation across open-world assets.

Joint Estimation and Physical Executability. Regarding kinematic alignment, MotionAnymesh drastically reduces geometric deviations, slashing Axis Error down to **0.12** and Pivot Error to **0.10**. Retrieval-based baselines frequently fail at spatial composition even when correct parts are retrieved (e.g., the *lamp* in Fig. 3), leading to completely erroneous joint parameters. Meanwhile, Articulate-AnyMesh attempts to estimate joints via 2D-to-3D VLM projection heuristics, making it highly susceptible to 3D spatial hallucinations and severe axis deviations. MotionAnymesh elegantly circumvents these bottlenecks by anchoring kinematics in robust 3D-native geometry. Our Type-Aware Kinematic Initialization deduces reliable parameters directly from localized contact interfaces, while our Physics-Constrained Trajectory Optimization strictly penalizes geometric inter-penetration. Consequently, our pipeline not only minimizes joint errors but achieves an unprecedented **87%** Physical Executability, nearly doubling the success rate of the optimal method (Articulate-Anything at 46%).

Table 1: Quantitative comparison against state-of-the-art baselines across diverse 3D assets. We evaluate our method against others on three core dimensions: Part Segmentation, Joint Parameter Estimation, and Physical Executability. ↑ indicates higher is better, and ↓ indicates lower is better. Best results are highlighted in **bold**.

Method	Part Segmentation		Joint Estimation			Simulation
	mIoU ↑	Count Acc ↑	Type Err ↓	Axis Err ↓	Pivot Err ↓	Executability ↑
PARIS [25]	0.17	0.23	0.67	1.56	1.14	11%
URDFormer [3]	0.21	0.33	0.72	1.31	1.53	21%
SINGAPO [24]	0.52	0.66	0.24	0.73	0.57	43%
Articulate-Anything [20]	0.47	0.61	0.21	0.86	0.64	46%
Articulate-AnyMesh [35]	0.59	0.74	0.35	0.64	0.44	35%
MotionAnymesh (Ours)	**0.86**	**0.92**	**0.08**	**0.12**	**0.10**	**87%**

4.2 Ablation Study

To rigorously validate the individual contributions of our proposed core modules, we systematically disable specific components of MotionAnymesh and evaluate the degraded variants. The decoupled ablation results for part segmentation and joint estimation are presented in Table 2 and Table 3, respectively.

Table 2: Ablation on SP4D kinematic priors. Removing SP4D guidance ("w/o SP4D") causes the VLM to rely solely on semantic logic, leading to severe kinematic hallucinations and over-segmentation.

Clustering Strategy	mIoU ↑	Count Acc ↑
Pure VLM Semantics (w/o SP4D)	0.68	0.81
SP4D-Guided Clustering (Ours)	**0.86**	**0.92**

Effectiveness of SP4D Kinematic Priors. We introduce a "w/o SP4D" variant by removing multi-view kinematic masks during part clustering, forcing the VLM to aggregate 3D-native primitives based purely on visual semantics. As shown in Table 2, this purely semantic approach leads to a drastic drop in Count Accuracy and mIoU. Semantic understanding alone is insufficient for articulation modeling; without explicit kinematic priors, the VLM frequently suffers from severe kinematic hallucinations, either erroneously merging distinct but visually similar components, or over-segmenting monolithic irregular structures lacking explicit semantic labels. In contrast, our SP4D-guided clustering strictly bounds the VLM's reasoning within physical reality, ensuring geometrically intact and topologically accurate part extraction.

Necessity of Physics-Constrained Optimization. The "w/o Opt." variant bypasses the non-linear trajectory refinement, directly outputting the initialized joint parameters. As shown in Table 3, while achieving seemingly reasonable static metrics, it suffers a catastrophic collapse in dynamic Physical

Table 3: Impact of physics-constrained optimization. While geometric initialization provides a reasonable starting point, our non-linear optimization is essential to eliminate micro-misalignments and guarantee high physical executability in simulators.

Joint Estimation Method	Axis Err ↓	Pivot Err ↓	Executability ↑
Kinematic Initialization (w/o Opt.)	0.23	0.27	65%
Physics-Constrained Opt. (Ours)	**0.12**	**0.10**	**87%**

Executability. Purely geometric initialization only provides a macro-level orthogonal guess; even a marginal angular deviation violently accumulates during long-range articulation, inevitably causing severe mesh inter-penetration or structural freezing in physics simulators. By actively penalizing geometric collisions and minimizing surface distances, our optimization systematically corrects these micro-misalignments, guaranteeing collision-free and seamlessly executable digital twins.

4.3 Applications

To demonstrate the versatility and physical reliability of MotionAnymesh, we showcase its integration into two downstream workflows: open-vocabulary asset generation and embodied robotic manipulation.

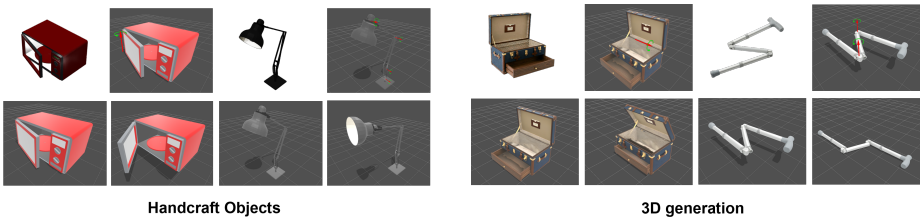

Fig. 4: Versatile articulation across diverse domains. MotionAnymesh successfully processes handcrafted assets and AI-generated surface meshes, transforming unstructured geometry into interactive digital twins without manual intervention.

Versatile Articulation for Diverse 3D Assets. MotionAnymesh serves as a universal articulation engine across diverse 3D domains (Fig. 4). For existing artist-designed datasets (e.g., PartNet-Mobility), it automatically extracts part segmentations and deduces precise joint structures without manual annotations. Furthermore, by processing static surface meshes synthesized by state-of-the-art 3D generative models, MotionAnymesh effortlessly transforms unstructured, open-vocabulary AI assets into high-fidelity, interactable URDF models ready for interactive simulation.

Real-to-Sim-to-Real Robotic Manipulation. To validate the physical executability of our digital twins, we deploy MotionAnymesh in a complete Real-

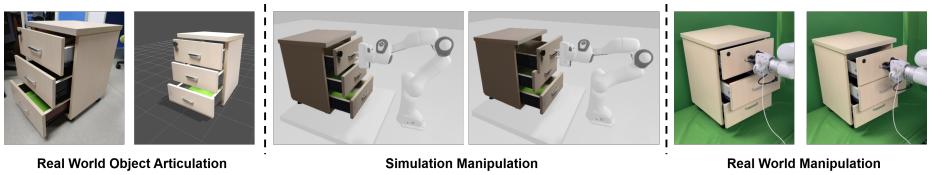

Fig. 5: Real-to-Sim-to-Real application. (Left) A static mesh is reconstructed from a single image. (Middle) MotionAnymesh generates a collision-free URDF for policy learning in simulation. (Right) The learned manipulation policy is deployed onto a physical robot, validating the sim-to-real fidelity of our estimated parameters.

to-Sim-to-Real pipeline (Fig. 5). Starting from a single real-world photograph, Hunyuan3D [47] reconstructs a static 3D surface mesh. MotionAnymesh then processes this raw geometry into a strictly collision-free, articulated URDF. After a robotic arm learns manipulation policies on this interactive asset in simulation, the learned trajectory is successfully deployed onto a physical robot. This end-to-end execution strongly validates that our physics-constrained pipeline produces highly accurate, sim-to-real-ready joint parameters and segmentations.

5 Conclusion

In this paper, we presented **MotionAnymesh**, an automated zero-shot framework that directly transforms unstructured static 3D meshes into simulation-ready articulated digital twins. Addressing the critical lack of physical grounding in existing perception pipelines, we introduced a robust perception-to-actuation methodology. Our kinematic-aware part segmentation grounds Vision-Language Models with explicit SP4D physical priors, effectively reducing kinematic hallucinations, while our geometry-physics joint estimation pipeline enforces strict trajectory optimization to guarantee collision-free kinematics. Extensive evaluations demonstrate that MotionAnymesh significantly outperforms state-of-the-art baselines in both geometric precision and dynamic physical executability. As a mesh-first pipeline, MotionAnymesh is most reliable when functional parts remain geometrically separable; severely fused scans, missing surfaces, or highly ambiguous contact regions may still require cleaner geometry or limited manual correction. This limitation also points to future integration with stronger 3D-native perception and repair models.

Acknowledgements

This work was supported by the Deep Earth Probe and Mineral Resources Exploration—National Science and Technology Major Project under Grant 2025ZD1010704, and the National Natural Science Foundation of China under Grant 62302143.

References

1. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2022)
2. Chen, C., Liu, I., Wei, X., Su, H., Liu, M.: Freeart3d: Training-free articulated object generation using 3d diffusion. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–13 (2025)
3. 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. arXiv preprint arXiv:2405.11656 (2024)
4. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13142–13153 (2023)
5. Deng, J., Subr, K., Bilen, H.: Articulate your nerf: Unsupervised articulated object modeling via conditional view synthesis. Advances in Neural Information Processing Systems **37**, 119717–119741 (2024)
6. Deng, Y., Zhang, Y., Geng, C., Wu, S., Wu, J.: Anymate: A dataset and baselines for learning 3d object rigging. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–10 (2025)
7. Duan, J., Yu, S., Tan, H.L., Zhu, H., Tan, C.: A survey of embodied ai: From simulators to research tasks. IEEE Transactions on Emerging Topics in Computational Intelligence **6**(2), 230–244 (2022)
8. Gao, L., Yang, J., Wu, T., Yuan, Y.J., Fu, H., Lai, Y.K., Zhang, H.: Sdm-net: Deep generative network for structured deformable mesh. ACM Transactions on Graphics (ToG) **38**(6), 1–15 (2019)
9. Geng, H., Xu, H., Zhao, C., Xu, C., Yi, L., Huang, S., Wang, H.: 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. pp. 7081–7091 (2023)
10. Gu, J., Xiang, F., Li, X., Ling, Z., Liu, X., Mu, T., Tang, Y., Tao, S., Wei, X., Yao, Y., et al.: Maniskill2: A unified benchmark for generalizable manipulation skills. arXiv preprint arXiv:2302.04659 (2023)
11. 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: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27144–27153 (2025)
12. He, Y., Jiang, Y., Lu, J., Yang, Y., Jiang, C.: Spark: Sim-ready part-level articulated reconstruction with vlm knowledge. arXiv preprint arXiv:2512.01629 (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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21201–21210 (2023)
14. Huang, Z., Guo, Y.C., Wang, H., Yi, R., Ma, L., Cao, Y.P., Sheng, L.: Mv-adapter: Multi-view consistent image generation made easy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16377–16387 (2025)

15. Jiang, N., He, Z., Wang, Z., Li, H., Chen, Y., Huang, S., Zhu, Y.: Autonomous character-scene interaction synthesis from text instruction. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
16. Jiang, Z., Hsu, C.C., Zhu, Y.: Ditto: Building digital twins of articulated objects from interaction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5616–5626 (2022)
17. Kim, D., Ha, T., Hong, J., Kim, S., Choi, S., Ko, H., Woo, W.: Meta-objects: Interactive and multisensory virtual objects learned from the real world for use in augmented reality. IEEE Computer Graphics and Applications **45**(3), 134–143 (2025)
18. Kim, J., Kim, J., Na, J., Joo, H.: Parahome: Parameterizing everyday home activities towards 3d generative modeling of human-object interactions. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1816–1828 (2025)
19. Koo, J., Yoo, S., Nguyen, M.H., Sung, M.: Salad: Part-level latent diffusion for 3d shape generation and manipulation. In: Proceedings of the ieee/cvf international conference on computer vision. pp. 14441–14451 (2023)
20. 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. arXiv preprint arXiv:2410.13882 (2024)
21. Li, C., Xia, F., Martín-Martín, R., Lingelbach, M., Srivastava, S., Shen, B., Vainio, K., Gokmen, C., Dharan, G., Jain, T., et al.: igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. arXiv preprint arXiv:2108.03272 (2021)
22. Li, X., Wang, H., Yi, L., Guibas, L.J., Abbott, A.L., Song, S.: Category-level articulated object pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3706–3715 (2020)
23. Li, Z., Bai, X., Zhang, J., Wu, Z., Xu, C., Li, Y., Hou, C., Zhang, S.: Urdf-anything: Constructing articulated objects with 3d multimodal language model. arXiv preprint arXiv:2511.00940 (2025)
24. Liu, J., Iliash, D., Chang, A.X., Savva, M., Mahdavi-Amiri, A.: Singapo: Single image controlled generation of articulated parts in objects. arXiv preprint arXiv:2410.16499 (2024)
25. Liu, J., Mahdavi-Amiri, A., Savva, M.: Paris: Part-level reconstruction and motion analysis for articulated objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 352–363 (2023)
26. Liu, J., Tam, H.I.I., Mahdavi-Amiri, A., Savva, M.: Cage: Controllable articulation generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17880–17889 (2024)
27. 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: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9704–9715 (2025)
28. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21736–21746 (2023)
29. Liu, Y., Jia, B., Lu, R., Ni, J., Zhu, S.C., Huang, S.: Building interactable replicas of complex articulated objects via gaussian splatting. In: ICLR (2025)
30. Lu, R., Liu, Y., Tang, J., Ni, J., Wang, Y., Wan, D., Zeng, G., Chen, Y., Huang, S.: Dreamart: Generating interactable articulated objects from a single image. arXiv preprint arXiv:2507.05763 (2025)

31. Ma, C., Li, Y., Yan, X., Xu, J., Yang, Y., Wang, C., Zhao, Z., Guo, Y., Chen, Z., Guo, C.: P3-sam: Native 3d part segmentation. arXiv preprint arXiv:2509.06784 (2025)
32. Mandi, Z., Weng, Y., Bauer, D., Song, S.: Real2code: Reconstruct articulated objects via code generation. arXiv preprint arXiv:2406.08474 (2024)
33. Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L.J.: Structurenet: Hierarchical graph networks for 3d shape generation. arXiv preprint arXiv:1908.00575 (2019)
34. Nakayama, G.K., Uy, M.A., Huang, J., Hu, S.M., Li, K., Guibas, L.: Difffacto: Controllable part-based 3d point cloud generation with cross diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14257–14267 (2023)
35. 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. arXiv preprint arXiv:2502.02590 (2025)
36. Shen, L., Zhang, S., Li, H., Yang, P., Huang, Z., Zhang, Z., Zhao, H.: Gaussianart: Unified modeling of geometry and motion for articulated objects. arXiv preprint arXiv:2508.14891 (2025)
37. Su, J., Feng, Y., Li, Z., Song, J., He, Y., Ren, B., Xu, B.: Artformer: Controllable generation of diverse 3d articulated objects. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1894–1904 (2025)
38. Tang, G., Zhao, W., Ford, L., Benhaim, D., Zhang, P.: Segment any mesh. arXiv preprint arXiv:2408.13679 (2024)
39. Wang, X., Zhou, B., Shi, Y., Chen, X., Zhao, Q., Xu, K.: Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8876–8884 (2019)
40. 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. arXiv preprint arXiv:2503.06677 (2025)
41. Wu, R., Zhuang, Y., Xu, K., Zhang, H., Chen, B.: Pq-net: A generative part seq2seq network for 3d shapes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 829–838 (2020)
42. Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., et al.: Sapien: A simulated part-based interactive environment. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11097–11107 (2020)
43. Yang, J., Mo, K., Lai, Y.K., Guibas, L.J., Gao, L.: Dsg-net: Learning disentangled structure and geometry for 3d shape generation. ACM Transactions on Graphics (TOG) **42**(1), 1–17 (2022)
44. Yang, Y., Xie, L., Luo, Z., Zhao, Z., Ding, T., Gao, M., Zheng, F.: Artiworld: Llm-driven articulation of 3d objects in scenes. arXiv preprint arXiv:2511.12977 (2025)
45. Yang, Y., Huang, Y., Guo, Y.C., Lu, L., Wu, X., Lam, E.Y., Cao, Y.P., Liu, X.: Sampart3d: Segment any part in 3d objects. arXiv preprint arXiv:2411.07184 (2024)
46. Zhang, H., Yao, C.H., Donné, S., Ahuja, N., Jampani, V.: Stable part diffusion 4d: Multi-view rgb and kinematic parts video generation. arXiv preprint arXiv:2509.10687 (2025)

47. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. arXiv preprint arXiv:2501.12202 (2025)
48. Zhu, Z., Wan, L., Xu, R., Zhang, Y., Chen, H., Dou, Z., Lin, C., Liu, Y., Wei, M.: Partsam: A scalable promptable part segmentation model trained on native 3d data. arXiv preprint arXiv:2509.21965 (2025)