# CRAFT: CONTROLLABLE GENERATION OF ARTICULATED OBJECTS FROM IMAGE AND TEXT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Controllable articulated-object generation is important for building interactive 3D worlds, supporting embodied intelligence and robotic simulation, and creating functional digital assets. A reference image reveals what an object looks like, but not necessarily how a user wants its parts to move. We introduce CRAFT, a framework for visually grounded, controllable articulated-object generation from a reference image and a textual articulation specification. CRAFT separates coarse structural control from detailed asset synthesis through an editable articulated 3D proxy that represents part dimensions and positions, spatial relationships, and kinematic parameters. We first construct an initial proxy using a category-informed structural initialization strategy. We then refine its spatial arrangement through adaptive structural optimization that combines instance-conditioned geometric objectives with semantic guidance from an image-derived 3D estimate. Finally, we generate detailed part assets and assemble them according to the refined proxy. Experiments demonstrate that CRAFT generates articulated objects consistent with the reference appearance while enabling text-specified control over part structure and motion.

## 1 INTRODUCTION

Generating an articulated object requires deciding not only what it looks like, but also how its parts should move. A cabinet with a familiar appearance might be configured with sliding drawers, hinged doors, or a combination of both. A reference image provides visual cues, but it does not necessarily communicate which configuration a user intends. This distinction matters when creating assets for interactive environments, embodied AI, robotics, and simulation (Shen et al., 2021; Kerr et al., 2024; Puig et al., 2024; Kim et al., 2025; Xiang et al., 2020; Li et al., 2023): a visually plausible object may still have the wrong part structure or motion mechanism.

Existing methods approach this problem through visual observations or structural and semantic specifications. Image-conditioned methods generate articulated objects consistent with a visual reference (Liu et al., 2025; Jin et al., 2025; Cao et al., 2026a), while videos and multi-state images provide additional evidence about demonstrated motion (Song et al., 2024; Liu et al., 2023; Chen et al., 2025). Graph- and text-conditioned methods instead expose design choices through part connectivity graphs (Liu et al., 2024; Lei et al., 2023; Gao et al., 2025) or language (Su et al., 2025; Gao et al., 2026). Some frameworks also support multiple input modalities (Le et al., 2025; Boudjoghra et al., 2026). Nevertheless, inferring a plausible motion mechanism from visual evidence and generating a mechanism explicitly requested by a user are different objectives. We focus on the latter while retaining a visual reference for the generated object. The central question is how to make the requested structure and articulation explicit without requiring the user to construct a complete 3D model.

We introduce CRAFT, a framework for visually grounded, controllable articulated-object generation from an image and a textual articulation specification. CRAFT, which organizes generation around an editable articulated 3D proxy. The proxy represents the intended output using part dimensions and positions, spatial relationships, and kinematic parameters. Our *key idea* is to separate coarse structural control from detailed asset synthesis: instructions are expressed through an explicit geometric and kinematic representation, while generative models supply detailed part geometry and



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

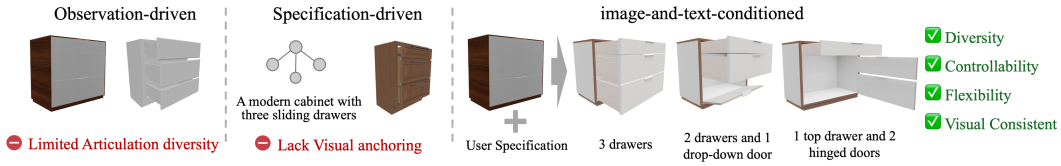



Figure 1: *Observation-driven* methods anchor object appearance but offer limited control over articulation, while *specification-driven* methods express design intent but lack object-specific visual grounding. In contrast, CRAFT generates different articulated configurations from the same visual reference by varying the textual articulation specification.

appearance. The proxy thus connects the desired articulation to the structure used for mesh assembly, rather than treating articulation as an attribute added after generating a complete object.

CRAFT first combines image-grounded part localization, text-guided articulation attributes, and category-informed depth priors to initialize the proxy. It then refines the part arrangement using semantic targets from rendered views of an image-derived 3D estimate and selectively activated geometric objectives. This instance-conditioned selection avoids imposing every layout assumption on every object. Detailed component assets are generated by semantic category and fitted to the refined proxy. Our contributions are summarized as:

- We introduce CRAFT, the first framework for visually grounded, controllable articulated-object generation that conditions on both a reference image and a textual articulation specification, preserving the object's visual appearance while enabling detailed, user-specified articulation control.
- We propose a structural initialization strategy that combines 2D part attributes with class-informed depth priors to construct a 3D proxy composed of replaceable part primitives, facilitating subsequent articulation revision through targeted updates to part dimensions and motion attributes.
- We introduce an adaptive structural optimization method that tailors geometric constraints to the structural characteristics of each object instance. Combined with semantic guidance, these constraints encourage accurate 3D structures for subsequent mesh resizing, positioning, and composition.
- Extensive experiments demonstrate that CRAFT generates high-quality articulated objects with user-specified articulation while preserving the appearance of the reference image.

## 2 Related Work

### 2.1 Specification-driven Generation

Specification-driven methods allow users to express design intent through structured representations or natural language. NAP (Lei et al., 2023) models distributions over part geometry and kinematic relations using diffusion on articulation graphs, while MeshArt (Gao et al., 2025) hierarchically generates articulation-aware structures and part meshes. CAGE (Liu et al., 2024) further enables explicit structural control by conditioning generation on an object category and a user-specified part-connectivity graph. Such specifications describe how parts are organized and connected, but leave their detailed geometry and visual appearance largely underdetermined.

Natural language offers a more accessible interface for specifying desired structures and functions. ArtFormer (Su et al., 2025) generates articulated structures and part geometry from textual descriptions, while LAM (Gao et al., 2026) formulates articulated-object generation as language-driven procedural modeling. Articulate-Anything (Le et al., 2025) also supports text-driven modeling, alongside image and video inputs, through vision-language reasoning. Although language makes functional preferences easy to communicate, text alone provides only coarse constraints on shape, proportions, materials, and textures. Thus, specification-driven generation offers convenient con-

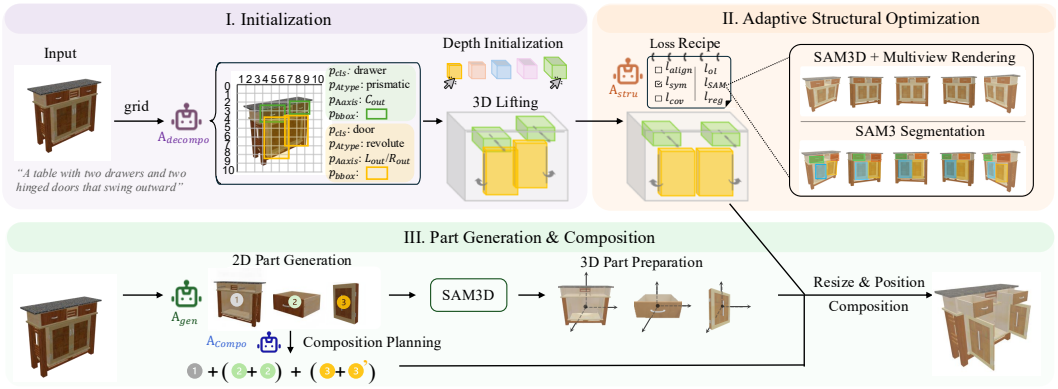



Figure 2: Pipeline Overview. Given an image and a textual articulation specification, CRAFT (I) initializes a part-level 3D proxy using visual grounding and category-informed priors, (II) optimizes it with adaptive geometric constraints and semantic guidance, and (III) generates and assembles part meshes under the optimized proxy to produce the articulated object.

trol over design intent, but relatively loose control over visual appearance, motivating the use of a complementary visual reference to anchor generation to a particular object.

### 2.2 Observation-driven Generation

Observation-driven methods use visual evidence to anchor object appearance and geometry, helping align generated assets with users' visual expectations. URDFormer (Chen et al., 2024) predicts articulated simulation assets from real-world images, and SINGAPO (Liu et al., 2025) generates articulated objects consistent with a single rest-state image. UniArt (Jin et al., 2025) jointly generates geometry, texture, part segmentation, and articulation, while PhysX-Anything (Cao et al., 2026b) extends single-image generation toward simulation-ready assets with physical and articulation attributes. PhysX-Omni (Cao et al., 2026c) further unifies the generation of rigid, deformable, and articulated physical assets. Although a reference image provides concrete visual constraints, its articulation is often ambiguous: the same visible structure may admit different joint types, axis placements, and motion directions. Consequently, appearance alone cannot determine which of these configurations a user intends, limiting its ability to support diverse articulation preferences.

Videos and multi-state images reduce this ambiguity by providing more explicit motion cues. REACTO (Song et al., 2024) reconstructs articulated objects from videos, while PARIS (Liu et al., 2023) and FreeArt3D (Chen et al., 2025) recover geometry and articulation from observations of multiple states. Such observations reveal demonstrated motions, but require additional capture effort and may be unavailable when the desired articulation has not been physically realized. UnfoldArt (Boudjoghra et al., 2026) instead leverages generative video priors for motion reasoning and hidden-geometry completion. Nevertheless, inferring plausible motion from visual evidence or generative priors does not by itself provide explicit control over a user's preferred articulation. These limitations motivate combining visual grounding with complementary specifications that allow users to prescribe part-level motions while retaining the reference appearance.

## 3 Method

### 3.1 Overview

Given an image $I$ and a text description $T$ specifying the desired articulation, our goal is to generate a set of part meshes and their associated articulation parameters that together form a 3D object geometrically consistent with $I$ and kinematically consistent with $T$. The image guides visual appearance and coarse geometry, while the text specifies desired part structure and motion. We represent

**Algorithm 1** Category-Informed Depth Initialization and 3D Lifting

**Require:** Image $I$; part descriptions $\{S_i = (\mathrm{cls}_i, b_i, p_i, t_i, h_i, u_i)\}_{i=1}^{N}$; depth priors $\alpha$
**Ensure:** Initialized articulated proxy $\mathcal{P}$

$B_{\mathrm{obj}} \leftarrow \mathrm{BBox}(\mathrm{SAM3D}(I))$
$b_{\mathrm{obj}} \leftarrow \mathrm{BBox}(\{b_i\}_{i=1}^{N})$; $D \leftarrow \mathrm{Depth}(B_{\mathrm{obj}})$
**for** $i = 1, \ldots, N$ **do**
  $d_i^{(0)} \leftarrow \alpha(\mathrm{cls}_i)D$ ▷ Default ratio: 0.5
  $B_i \leftarrow \mathrm{LiftBox}(b_i, b_{\mathrm{obj}}, B_{\mathrm{obj}}, d_i^{(0)})$
  $\mathbf{a}_i \leftarrow \Phi_{\mathrm{axis}}(B_i, t_i, h_i, u_i)$; $\mathbf{r}_i \leftarrow \mathcal{R}(t_i, h_i, u_i, B_i)$
  $P_i \leftarrow \big(\mathrm{cls}_i, B_i, (p_i, \mathbf{a}_i, \mathbf{r}_i, t_i)\big)$
**end for**
**return** $\mathcal{P} = \{P_i\}_{i=1}^{N}$

the output as

$$\mathcal{O} = \big(\{\mathcal{M}_i\}_{i=1}^{N}, \{\mathcal{K}_i\}_{i=1}^{N}\big), \quad \mathcal{M}_i = (\mathrm{cls}_i, \mathrm{M}_i), \quad \mathcal{K}_i = (p_i, a_i, r_i, t_i). \tag{1}$$

Here, $\mathcal{M}_i$ represents part $i$, comprising its semantic class label $\mathrm{cls}_i$ and mesh $\mathrm{M}_i$. The kinematic representation $\mathcal{K}_i$ specifies the parent part index $p_i$, joint axis location and direction $a_i \in \mathbb{R}^6$, motion range $r_i \in \mathbb{R}^2$, and articulation type $t_i$, including fixed, revolute, prismatic, and continuous joints. Besides, $N$ denotes the total number of parts.

To generate $\mathcal{O}$ from image and text inputs, we first initialize an articulated 3D proxy $\mathcal{P}_i$ in Sec. 3.2. This intermediate representation captures the object's part-level spatial layout and kinematic structure using simplified geometric primitives. Each part proxy is defined as $\mathcal{P}_i = (\mathrm{cls}_i, \mathcal{B}_i, \mathcal{K}_i)$ where $\mathrm{cls}_i$ and $\mathcal{K}_i$ encode the semantic class and kinematic properties of part $i$, respectively, and $\mathcal{B}_i \in \mathbb{R}^6$ is the 3D bounding box. We then refine the proxy through adaptive structural optimization in Sec. 3.3. The resulting proxy $\mathcal{P}^\star$ guides the scaling, positioning, and composition of the detailed part meshes to produce the final articulated object $\mathcal{O}$ in Sec. 3.4.

## 3.2 3D Proxy Initialization

### 3.2.1 Part Decomposition and Analysis

To obtain a part-level description grounded in visual evidence and user-specified articulation, the decomposition agent $A_{\mathrm{decomp}}$ jointly analyzes the input image $I$ and text description $T$. We overlay a $10 \times 10$ grid on $I$ to provide a shared spatial reference for localization. From the annotated image $I_{\mathrm{grid}}$ and $T$, the agent identifies the functional parts as $\mathcal{S}_i = A_{\mathrm{decomp}}(I_{\mathrm{grid}}, T) = (\mathrm{cls}_i, b_i, p_i, t_i, h_i, u_i)$. Each description contains a semantic class $\mathrm{cls}_i$, a 2D bounding box $b_i \in \mathbb{R}^4$ in grid coordinates, a parent index $p_i$, a joint type $t_i$, a qualitative axis anchor $h_i$, and an intended motion direction $u_i$. To constrain these predictions to admissible articulation configurations, we apply category-specific priors:$t_i \in \mathcal{T}(\mathrm{cls}_i)$, $h_i \in \mathcal{H}(t_i)$,$u_i \in \mathcal{U}(t_i)$ where $\mathcal{T}$, $\mathcal{H}$, and $\mathcal{U}$ denote predefined sets of admissible values based on common knowledge of object functionality and articulation. For example, we restrict drawers to prismatic joints, and allow doors to have either prismatic or revolute joints. The complete specification of $\mathcal{T}$ is provided in the appendix.

These structured descriptions connect image-grounded part localization with text-guided articulation, providing the semantic and kinematic inputs for the depth initialization and 3D lifting in the following section.

### 3.2.2 Category-informed Depth Initialization and 3D Lifting

To construct an articulated 3D proxy from the predicted part descriptions, we combine image-grounded layouts with category-informed depth priors. Since 2D bounding boxes do not determine part depths, we reconstruct an overall object mesh using SAM3D (Chen et al., 2026) and use its bounding box $B_{\mathrm{obj}}$ as a shared spatial reference. Each part receives an initial depth $d_i^{(0)} = \alpha(\mathrm{cls}_i)D$, where$\alpha(\mathrm{cls}_i) \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and $D$ is the extent of $B_{\mathrm{obj}}$ along the $Z$-axis. These ratios encode coarse functional expectations: drawers receive $0.9D$ to extend into the object interior,

whereas doors receive $0.1D$ to represent thin panels. Unspecified categories default to $0.5D$; the complete mapping is provided in the appendix.

Algorithm 1 summarizes the geometric and kinematic initialization. Given the enclosing 2D object box $b_{\text{obj}} = \text{BBox}(\{b_i\}_{i=1}^{N})$, LiftBox normalizes each part's 2D bounds relative to $b_{\text{obj}}$ and maps them to the horizontal and vertical extents of $B_{\text{obj}}$, accounting for the reversed vertical axis between image and object coordinates. It then aligns the part's front face with $Z_{\text{obj}}^{\min}$ and extends it inward by $d_i^{(0)}$, yielding a 3D box $B_i = (\mathbf{c}_i, \mathbf{s}_i)$. For articulation, we retain the predicted parent index $p_i$ and joint type $t_i$. The mapping $\Phi_{\text{axis}}$ converts the qualitative anchor $h_i$ and motion direction $u_i$ into a 3D axis point and unit direction. The range mapping $\mathcal{R}$ assigns $[0, 2\pi]$ to continuous joints, $[0, \pi/2]$ to revolute joints, and $[0, \ell_i]$ to prismatic joints, where $\ell_i$ is the box extent along the translation axis.

The resulting proxy $P_i = (\text{cls}_i, B_i, K_i)$ combines the observed part arrangement with explicit depth and articulation priors. Its independent and explicit part representations support localized updates to dimensions and motion attributes when user instructions change, while providing a coarse geometric starting point for the adaptive structural optimization in Sec. 3.3.

## 3.3 Adaptive Structural Optimization

### 3.3.1 Structural Agent

To adapt geometric constraints to each object's structure, the structural agent $A_{\text{stru}}$ analyzes the input image $I$ and selects applicable layout properties: left-right symmetry, centering alignment, and full coverage of the object's front face. These properties activate $\mathcal{L}_{\text{sym}}$, $\mathcal{L}_{\text{align}}$, and $\mathcal{L}_{\text{cov}}$, respectively, allowing optimization to exploit instance-specific regularities.

**Symmetry.** We encourage the aggregate layouts on either side of the object to have mirrored positions and matching dimensions. Specifically, let $m_x = (X_{\text{obj}}^{\min} + X_{\text{obj}}^{\max})/2$ denote the symmetry plane. We partition the parts into a left partition $\mathcal{P}_L = \{i : c_{i,x} < m_x - \tau\}$ and a right partition $\mathcal{P}_R = \{i : c_{i,x} > m_x + \tau\}$, where $\tau$ is a small tolerance used to identify parts centered on the symmetry plane. For the enclosing boxes of these two sets, with centers $\mathbf{c}_L, \mathbf{c}_R$ and dimensions $\mathbf{s}_L, \mathbf{s}_R$, we define $\mathcal{L}_{\text{sym}}$ as:

$$\mathcal{L}_{\text{sym}} = \|\rho(\mathbf{c}_L) - \mathbf{c}_R\|_2^2 + \lambda_s \|\mathbf{s}_L - \mathbf{s}_R\|_2^2, \qquad \rho(\mathbf{c}_L) = (2m_x - c_{L,x}, c_{L,y}, c_{L,z})^\top. \tag{2}$$

**Alignment.** We encourage part centers to align with the object center along the axes $\mathcal{A} \subseteq \{x, y\}$ selected by $A_{\text{stru}}$. The alignment loss is

$$\mathcal{L}_{\text{align}} = \frac{1}{|N|} \sum_{i \in N} \sum_{a \in \mathcal{A}} (c_{i,a} - c_{\text{obj},a})^2, \tag{3}$$

where $c_{i,a}$ and $c_{\text{obj},a}$ are the center coordinates of part $i$ and the object bounding box along axis $a$.

**Coverage.** For layouts expected to cover the object's front face, we penalize both uncovered regions and extension beyond the object boundary using

$$\mathcal{L}_{\text{cov}} = 1 - \frac{|F_{\mathcal{P}} \cap F_{\text{obj}}|}{|F_{\mathcal{P}} \cup F_{\text{obj}}|}. \tag{4}$$

where $F_{\mathcal{P}}$ is the union of the front-plane projections of the selected part boxes, and $F_{\text{obj}}$ is the object's front face.

### 3.3.2 Optimization Objectives

To refine the initialized proxy, CRAFT optimizes the bounding-box offsets $\Delta\mathcal{B}_i$, with $\mathcal{B}_i = \mathcal{B}_i^{(0)} + \Delta\mathcal{B}_i$. Three mandatory objectives complement the optional constraints: overlap reduction, regularization, and SAM3-guided (Carion et al., 2026) multiview consistency. Together, they discourage inter-part penetration, excessive deviations, and align the proxy with visual evidence.

**Overlap and regularization.** We penalize intersections between distinct part boxes, normalized by the object volume:

$$\mathcal{L}_{\text{ol}} = \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \frac{\text{Vol}(\mathcal{B}_i \cap \mathcal{B}_j)}{\text{Vol}(\mathcal{B}_{\text{obj}})}, \qquad \mathcal{E} = \{(i,j) : i,j \in N,\, i < j\} \tag{5}$$

where $\mathcal{E}$ is the set of unordered part pairs, and $\text{Vol}(\cdot)$ denotes volume.

To limit deviations from the initial centers and dimensions, we additionally apply

$$\mathcal{L}_{\text{reg}} = \frac{1}{|N|} \sum_{i\in N} \|\Delta\mathcal{B}_i\|_1, \tag{6}$$

where $\Delta\mathcal{B}_i \in \mathbb{R}^6$ contains the offsets to its initial center and dimensions.

**Semantic multiview consistency.** We render $M_{\text{obj}}$ from 8 viewpoints and extract part-level masks using SAM3. For each part-view pair $(i, v)$, we compare the mask's tight bounding box $\mathbf{b}^v_{i,\text{SAM}}$ with the enclosing box $\widehat{\mathbf{b}}^v_i$ of the part proxy projected using the same camera:

$$\mathcal{L}_{\text{SAM}} = \frac{1}{|\mathcal{V}|} \sum_{(i,v)\in\mathcal{V}} \left[1 - \text{IoU}\left(\widehat{\mathbf{b}}^v_i, \mathbf{b}^v_{i,\text{SAM}}\right) + \left\|\widehat{\mathbf{b}}^v_i - \mathbf{b}^v_{i,\text{SAM}}\right\|_1\right], \tag{7}$$

where $\mathcal{V}$ denotes the set of pairs. Because drawer interiors are largely occluded, visual evidence provides limited depth supervision. We therefore retain their initialized depths by fixing $\Delta d_i = 0$ and use only their front panels when computing $\mathcal{L}_{\text{SAM}}$.

In conclusion, the final objective can be represented as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ol}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{SAM}} + w_{\text{sym}}\mathcal{L}_{\text{sym}} + w_{\text{align}}\mathcal{L}_{\text{align}} + w_{\text{cov}}\mathcal{L}_{\text{cov}}, \tag{8}$$

where $w_{\text{sym}}, w_{\text{align}}, w_{\text{cov}} \in \{0, 1\}$ indicate whether the corresponding optional terms are selected, yielding the optimized articulated proxy $\mathcal{P}^\star$.

### 3.4 Part Generation and Composition

Given the input image $I$ and the optimized proxy $\mathcal{P}^\star = (\text{cls}_i, \mathcal{B}^\star_i, \mathcal{K}_i)$, we generate images of the base and part components, reconstruct their meshes, and assemble them according to $\mathcal{B}^\star$. Specifically, the base is the stationary structure remaining after removing the articulated components and serves as the fixed root of the kinematic tree.

We first use the generation agent $A_{\text{gen}}$, implemented with Gemini-2.5-flash, to generate a base image $J_{\text{base}}$ from $I$, depicting the stationary structure. For the articulated components, we generate one representative image per semantic category rather than independently generating each instance. This shared reference reduces appearance variation across repeated parts caused by stochastic generation. Let $\mathcal{C}$ denote the set of articulated part categories. For each category $c \in \mathcal{C}$, the agent receives the input image $I$, a mask-annotated view $I^{\text{mask}}_c$ highlighting the target category, and outputs the class image as $J_c = A_{\text{gen}}(I, I^{\text{mask}}_c, c)$.

The composition agent $A_{\text{compo}}$ associates $\mathcal{P}_i$ with a generated category-level component and determines how that component should be reused as $\{(\mathcal{P}_i, f_i)\}_{i\in N} = A_{\text{compo}}\left(I, \{J_c\}_{c\in\mathcal{C}}, T\right)$ where $f_i$ specifies an identity or reflection operation. For example, a generated drawer image can be reused directly for multiple drawers, whereas a door image may require flipping.

Next, the source mesh for each component is reconstructed through $\widetilde{M}_c = \text{SAM3D}(J_c)$, where $c \in \{\mathcal{C} \cup \text{base}\}$. For the base, we define its target bounding box $\mathcal{B}_{\text{base}}$ as the smallest box enclosing the region remaining after removing $\mathcal{B}^\star$ from the object box $\mathcal{B}_{\text{obj}}$. Then, apply the planned operation $f_i$ to $\widetilde{M}_c$ and fit it to its target bounding box as $M_i = \text{Fit}\left(f_i(\widetilde{M}_c), \mathcal{B}^\star_i\right)$, where $\text{Fit}(\cdot)$ scales the mesh to the target dimensions and aligns its bounding-box center with that of the proxy.

### 3.5 Controllable Articulation Revision

Our method supports articulation revision through an updated caption, using localized proxy updates and component reuse to preserve appearance. Given the original image $I$, revised caption $T'$, and

optimized proxy $\mathcal{P}^\star$, we rerun the initialization to infer the updated part classes $cls'$, kinematic parameters $\mathcal{K}'$, and category-informed depths $d'$. For each affected part, we update the corresponding $cls'$, $d'$, and $\mathcal{K}'$ in $\mathcal{P}^\star$, retaining the remaining proxy parameters.

Starting from the edited proxy, we perform 50 optimization iterations by default to obtain $\mathcal{P}^{\star\prime}$. We then reuse existing component assets, generating additional images and reconstructing meshes only for newly introduced categories. Following the updated composition plan, we fit the reused and newly generated meshes to $\mathcal{P}^{\star\prime}$ to obtain $\mathcal{M}'$. This process implements the revised articulation while preserving the appearance of unchanged components.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets** We quantitatively evaluate CRAFT on two articulated 3D object datasets: PartNet-Mobility (Xiang et al., 2020) and Objaverse-Household (Deitke et al., 2023b;a). We use Singapo's (Liu et al., 2025) PartNet-Mobility test split (77 objects, 7 categories) as an in-domain benchmark for supervised baselines. The Objaverse-Household split from UnfoldArt (Boudjoghra et al., 2026) (53 objects, 12 categories) covers both seen categories with more complex geometries and more parts, and unseen household categories with motions absent from baseline training. We also qualitatively evaluate CRAFT on product images from the official IKEA website.

**Baselines.** We compare CRAFT with five state-of-the-art image-based methods, covering both training-based and training-free approaches. Although CRAFT uses both image and text conditioning, the text descriptions in this setting are automatically generated from the input image, requiring no additional user-provided text. Singapo (Liu et al., 2025) trains a diffusion model on PartNet-Mobility and retrieves part meshes from a fixed database. PhysX-Anything (Cao et al., 2026a) fine-tunes a VLM for articulation prediction and generates parts with TRELLIS (Xiang et al., 2025). Among training-free methods, Articulate-Anything (Le et al., 2025) uses a frozen VLM in an actor–critic loop to assemble PartNet-Mobility assets; FreeArt3D (Chen et al., 2025) optimizes static 3D diffusion priors via SDS but requires multi-view observations across articulation states; and UnfoldArt (Boudjoghra et al., 2026) refines joints through agent debate over whole-object and part-level assessments grounded in generated motion videos.

**Metric.** We assess geometric accuracy, articulation accuracy, and structural plausibility using metrics adopted from Singapo (Liu et al., 2025) and FreeArt3D (Chen et al., 2025). Geometric accuracy is measured after establishing part correspondences via Hungarian matching. Following Singapo, we compute bounding-box gIoU error ($d_{\mathrm{gIoU}}$), part-centroid distance ($d_{\mathrm{cDist}}$), and per-part Chamfer distance ($d_{\mathrm{CD}}$). Each metric is evaluated in the rest state (RS-) and averaged over articulated states (AS-), with lower values indicating better performance. To assess structural plausibility, we additionally report Singapo's average overlapping ratio (AOR), which quantifies collisions between sibling parts. Articulation accuracy is evaluated following FreeArt3D using joint-axis direction error, measured in degrees, and joint-pivot error, defined as the distance between the predicted and reference axes for revolute joints.

**Implementation.** All experiments are run on one NVIDIA L40s GPU. We use Gemini-2.5-flash as the backbone of the generation agent and GPT-5.6-Luna for all other agents. By default, we perform optimization for 300 iterations using the Adam optimizer with a learning rate of $10^{-3}$.

## 4.2 Results

**Quantitative Results.** Tab. 1 compares CRAFT with existing methods on Objaverse-Household and PartNet-Mobility. Even in the image-only setting, where text conditioning is automatically derived from the input image rather than provided independently, CRAFT achieves strong performance on both benchmarks. On Objaverse-Household, it achieves the lowest errors across all three rest-state geometry metrics, alongside the lowest joint-axis and joint-pivot errors and the highest joint accuracy. On PartNet-Mobility, CRAFT achieves the lowest rest-state Chamfer distance and joint-axis error, while attaining a joint accuracy of 87.2%, outperforming the strongest baseline by 9.7

Table 1: Comparison on Objaverse-Household and PartNet-Mobility. ↓ indicates lower is better, while ↑ indicates higher is better.

| | Method | RS-dgIoU↓ | AS-dgIoU↓ | RS-dcDist↓ | AS-dcDist↓ | RS-dCD↓ | AS-dCD↓ | AOR↓ | Axis Err↓ | Pivot Err↓ | Joint Acc%↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Objaverse-Household (12 classes) | | | | | | | | | | | |
| Training | SINGAPO | 1.213 | 1.215 | 0.243 | 0.360 | 0.177 | 0.353 | 0.005 | 39.8° | 0.30 | 48.5 |
| | PhysX-Anything | 1.134 | 1.139 | 0.314 | 0.621 | 0.236 | 0.401 | 0.011 | 36.8° | 0.46 | 19.9 |
| Training-F. | Articulate-Anything | 1.178 | 1.183 | 0.216 | 0.477 | 0.136 | 0.346 | **0.001** | 23.5° | 0.33 | 57.5 |
| | FreeArt3D | 0.926 | 0.927 | 0.183 | 0.406 | 0.248 | 0.388 | 0.13 | 25.0° | 0.34 | 58.4 |
| | UnfoldArt | 0.955 | 0.960 | 0.175 | **0.244** | 0.103 | **0.188** | 0.024 | 17.3° | 0.17 | 57.5 |
| | Ours | **0.762** | **0.774** | **0.144** | 0.408 | **0.089** | 0.359 | **0.001** | **10.6°** | **0.11** | **71.0** |
| PartNet-Mobility (7 classes) | | | | | | | | | | | |
| Training | SINGAPO | **0.570** | **0.579** | **0.076** | **0.119** | 0.028 | **0.049** | **0.001** | 1.4° | **0.022** | 50.6 |
| | PhysX-Anything | 0.953 | 0.958 | 0.177 | 0.281 | 0.089 | 0.195 | 0.006 | 21.6° | 0.193 | 34.7 |
| Training-F. | Articulate-Anything | 0.949 | 0.953 | 0.145 | 0.207 | 0.057 | 0.115 | **0.001** | 7.5° | 0.36 | 72.3 |
| | FreeArt3D | 0.818 | 0.820 | 0.127 | 0.187 | 0.315 | 0.360 | 0.018 | 67.5° | 0.41 | 33.3 |
| | UnfoldArt | 0.655 | 0.662 | 0.103 | 0.154 | 0.032 | 0.078 | 0.006 | 8.0° | 0.15 | 77.5 |
| | Ours | 0.592 | 0.623 | 0.089 | 0.198 | **0.025** | 0.090 | **0.001** | **1.0°** | 0.11 | **87.2** |

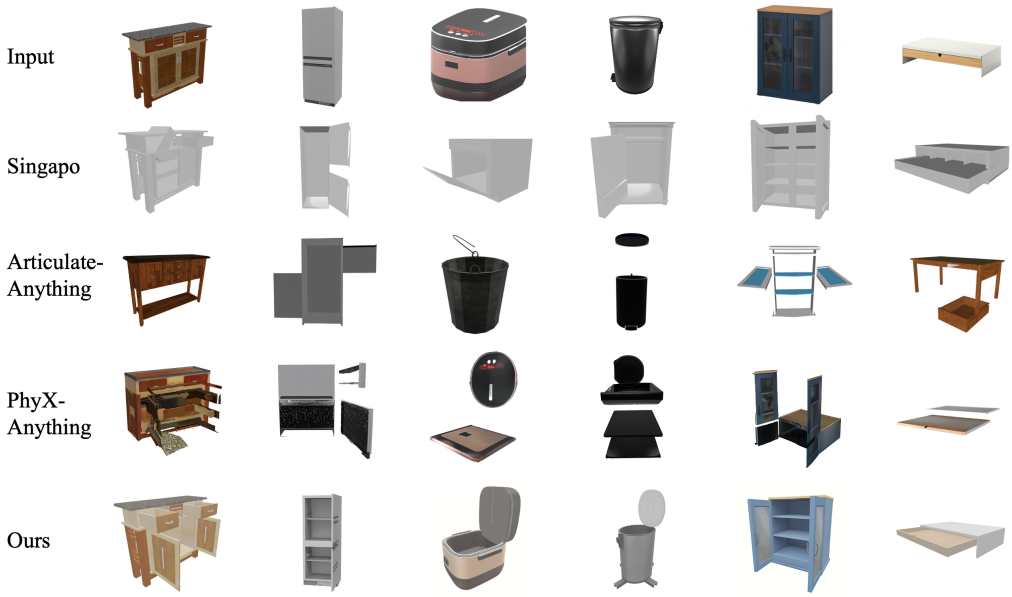



Figure 3: Qualitative Comparisons. Columns 1–2, 3–4, and 5–6 show examples from PartNet-Mobility, Objaverse, and the IKEA objects, respectively.

percentage points. It also matches the lowest AOR on both datasets. However, its articulated-state centroid and Chamfer distances remain less competitive. This limitation may stem from our use of predefined articulation-type-specific motion ranges without optimization: although the predicted geometry and joint axes are accurate, discrepancies in motion extent can lead to part misalignment in articulated states.

**Qualitative Comparisons.** Fig. 3 presents qualitative comparisons under the image-only setting. CRAFT produces coherent articulated objects that retain the overall shape and key appearance cues of the input images while exhibiting plausible part configurations. Singapo captures some coarse structures but produces category-inconsistent geometry for the rice cooker and trash bin. Articulate-Anything exhibits mismatches in object shape and part arrangement, while PhysX-Anything preserves some visual details but yields fragmented or distorted geometry in several examples. In comparison, CRAFT better maintains the spatial relationships between movable parts and their supporting structures, as illustrated by the cabinet doors, appliance lids, and sliding drawer. Although

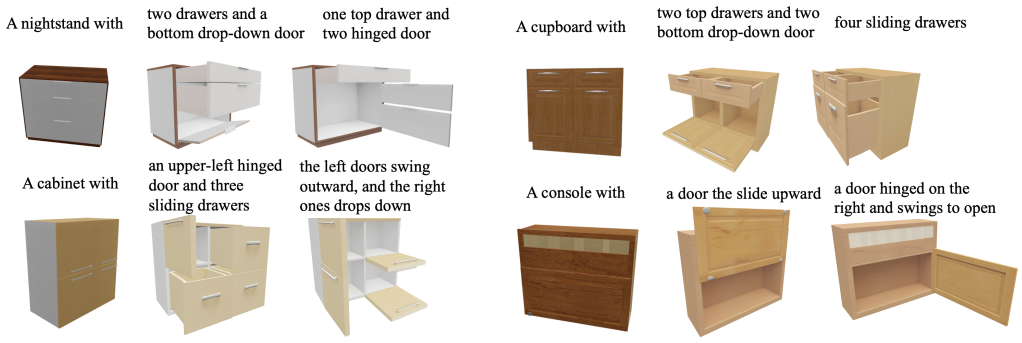



Figure 4: Controllable Generation with Image and Text Conditioning.

fine textures and colors are not fully preserved, these examples demonstrate CRAFT's ability to balance geometric coherence, appearance preservation, and plausible articulation.

**Controllable Generation with Image and Text Conditioning.** Fig. 4 illustrates how CRAFT generates different articulated structures from the same reference image through text conditioning. For the nightstand and cupboard, text instructions specify different numbers and arrangements of drawers and doors, yielding distinct part decompositions within a similar overall shape. The cabinet examples further demonstrate control over individual parts, combining sliding drawers with hinged doors or assigning different opening directions to the left and right doors. For the console, CRAFT produces either an upward-sliding door or a door hinged on the right, illustrating control over joint type and hinge placement. Across these examples, the generated objects broadly retain the appearance cues. These results demonstrate that CRAFT can use images to guide visual appearance while allowing text to specify part structure and articulation.

**Ablation Study** Tab. 2 evaluates the contributions of optimization and its adaptive mechanism on PartNet-Mobility. Removing optimization substantially degrades performance across all three geometry metrics, highlighting its importance for geometric accuracy. Retaining optimization without the adaptive mechanism improves reconstruction quality but remains less effective than the full model. The consistent gains from the adaptive mechanism further demonstrate its contribution to geometric refinement.

Table 2: Ablation studies on PartNet-Mobility.

| Variant | dgIoU↓ | dcDist↓ | dCD↓ |
|---|---|---|---|
| w/o optim. | 0.921 | 0.176 | 0.055 |
| w/o adapt. optim. | 0.712 | 0.099 | 0.037 |
| FULL | 0.592 | 0.089 | 0.025 |

## 5 Limitations and Future Work

We presented CRAFT, an image-and-text-conditioned framework for controllable articulated object generation. By combining visual observations with textual articulation specifications, CRAFT preserves reference appearance while allowing users to customize part structure and motion. Its explicit, editable 3D proxy connects prior-guided structural initialization, adaptive structural optimization, and compositional part synthesis, supporting targeted revisions through selective regeneration and component reuse. However, CRAFT uses predefined motion ranges, which may contribute to the less competitive articulated-state centroid and Chamfer distances despite strong rest-state geometry and joint-axis accuracy. Future work could jointly optimize joint limits with motion-aware constraints. Additionally, CRAFT represents articulated objects as rigid parts connected by joints and does not support articulation involving flexible or deformable components. Extending the framework to model non-rigid deformation and its coupling with joint motion is an important direction for future work.

### AI USE STATEMENT

In this work, we used generative AI tools for paper polishing. We have not used generative AI tools for generating research results, and searching for or retrieving literature are not applicable to this work. We have reviewed all AI-assisted work. All language revisions made with the assistance of generative AI were reviewed and approved by all authors to ensure that the revised text preserved the intended meaning and technical accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Mohamed el Amine Boudjoghra, Ivan Laptev, and Angela Dai. Unfoldart: Zero-shot recovery of full articulated 3d objects from text or image. *arXiv preprint arXiv:2606.30608*, 2026.

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*, pp. 5839–5848, 2026a.

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*, pp. 5839–5848, 2026b.

Ziang Cao, Yinghao Liu, Haitian Li, Runmao Yao, Fangzhou Hong, Zhaoxi Chen, Liang Pan, and Ziwei Liu. Physx-omni: Unified simulation-ready physical 3d generation for rigid, deformable, and articulated objects. *arXiv preprint arXiv:2605.21572*, 2026c.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Chuhao Chen, Isabella Liu, Xinyue Wei, Hao Su, and Minghua Liu. Freeart3d: Training-free articulated object generation using 3d diffusion. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–13, 2025.

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.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023a.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023b.

Daoyi Gao, Yawar Siddiqui, Lei Li, and Angela Dai. Meshart: Generating articulated meshes with structure-guided transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 618–627. IEEE, 2025.

Yipeng Gao, Yunhao Ge, Peilin Cai, Daniel Seita, and Laurent Itti. LAM: Language articulated object modelers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16010–16020, 2026.

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

Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. *arXiv preprint arXiv:2409.18121*, 2024.

Jeonghwan Kim, Jisoo Kim, Jeonghyeon Na, and Hanbyul Joo. Parahome: Parameterizing everyday home activities towards 3d generative modeling of human-object interactions. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1816–1828. IEEE, 2025.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In *International Conference on Learning Representations*, volume 2025, pp. 17578–17602, 2025.

Jiahui Lei, Congyue Deng, William B Shen, Leonidas Guibas, and Kostas Daniilidis. Nap: Neural 3d articulated object prior. *Advances in Neural Information Processing Systems*, 36:31878–31894, 2023.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pp. 80–93. PMLR, 2023.

Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-level reconstruction and motion analysis for articulated objects. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 352–363. IEEE, 2023.

Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. Cage: Controllable articulation generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17880–17889. IEEE, 2024.

Jiayi Liu, Denys Iliash, Angel Chang, Manolis Savva, and Ali Mahdavi Amiri. Singapo: Single image controlled generation of articulated parts in objects. In *International Conference on Learning Representations*, volume 2025, pp. 97511–97532, 2025.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations*, volume 2024, pp. 15306–15336, 2024.

Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. igibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7520–7527. IEEE, 2021.

Chaoyue Song, Jiacheng Wei, Chuan Sheng Foo, Guosheng Lin, and Fayao Liu. Reacto: Reconstructing articulated objects from a single video. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5384–5395. IEEE, 2024.

Jiayi Su, Youhe Feng, Zheng Li, Jinhua Song, Yangfan He, Botao Ren, and Botian Xu. Artformer: Controllable generation of diverse 3d articulated objects. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1894–1904. IEEE, 2025.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11094–11104. IEEE, 2020.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

# A APPENDIX

## A.1 CATEGORY-SPECIFIC PRIORS

We use category-specific admissible sets to constrain joint-type assignment. For a part with category $\mathrm{cls}_i$, its joint type is restricted to the corresponding set $\mathcal{T}(\mathrm{cls}_i)$. For example, drawers are restricted to prismatic joints, whereas doors may have either prismatic or revolute joints, accommodating both sliding and hinged configurations. The first two columns of Tab. 3 list the admissible joint types for each part category.

We associate each part category $\mathrm{cls}_i$ with a predefined depth ratio $\alpha(\mathrm{cls}_i)$. Parts belonging to the same category share the same ratio. The last column Tab. 3 specifies the mapping from part categories to depth ratios used in our method.

Table 3: Proposed category-specific joint constraints and initial depth ratios. F, P, R, and C denote fixed, prismatic, revolute, and continuous joints, respectively.

| Part category | $\mathcal{T}(\mathrm{cls}_i)$ | $\alpha(\mathrm{cls}_i)$ |
|---|---|---|
| Drawer | $\{\mathrm{P}\}$ | 0.9 |
| Door | $\{\mathrm{P}, \mathrm{R}\}$ | 0.1 |
| Lid | $\{\mathrm{P}, \mathrm{R}, \mathrm{C}\}$ | 0.9 |
| tray | $\{\mathrm{P}, \mathrm{F}\}$ | 0.7 |
| shelf | $\{\mathrm{P}, \mathrm{F}\}$ | 0.7 |
| basket | $\{\mathrm{P}, \mathrm{F}\}$ | 0.7 |
| Knob | $\{\mathrm{R}, \mathrm{C}\}$ | 0.3 |
| handle | $\{\mathrm{R}, \mathrm{F}\}$ | 0.3 |
| others | $\{\mathrm{P}, \mathrm{C}, \mathrm{R}, \mathrm{F}\}$ | 0.5 |

Broad categories may admit multiple joint types to accommodate different mechanisms. When a more specific category is available, the admissible set is narrowed accordingly; for example, hinged doors admit only revolute joints, whereas sliding doors admit only prismatic joints. Handles rigidly attached to another moving part inherit that part's motion and are not assigned an independent joint.

The depth ratios are manually specified initialization priors, rather than empirical category averages. They provide a starting configuration for common furniture and appliance parts. Categories outside Table 3 require an explicit extension of the mapping.