FreeArt3D: Training-Free Articulated Object Generation using 3D Diffusion

CHUHAO CHEN, University of California San Diego, USA
ISABELLA LIU, University of California San Diego, USA
XINYUE WEI, University of California San Diego, USA and Hillbot Inc., USA
HAO SU, University of California San Diego, USA and Hillbot Inc., USA
MINGHUA LIU, Hillbot Inc., USA

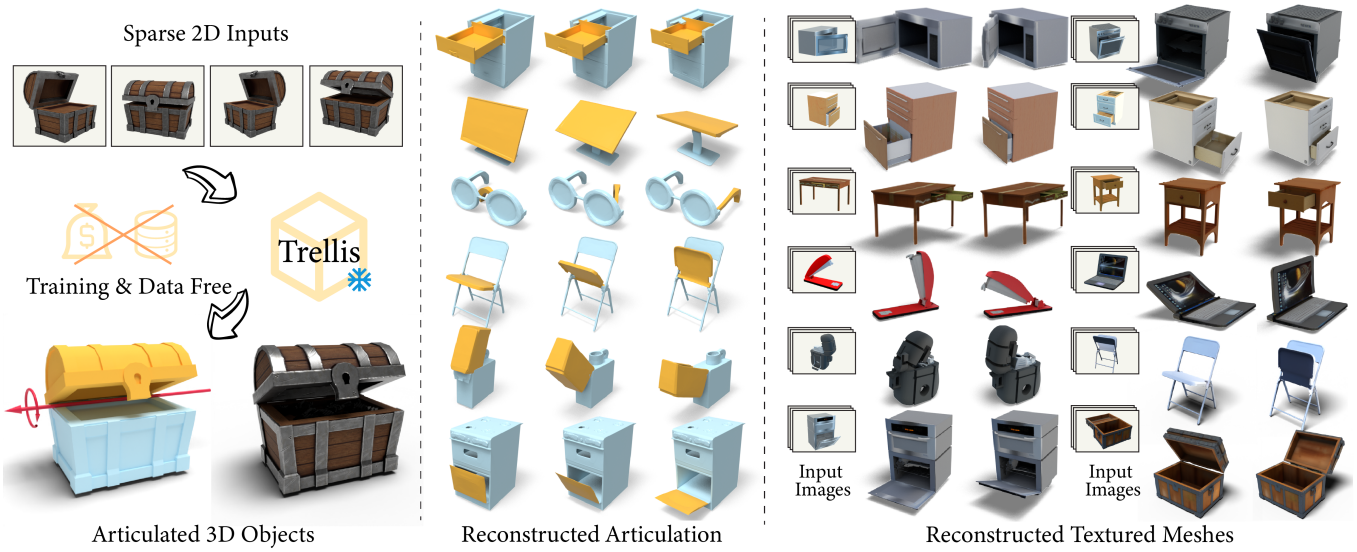

Fig. 1. Facing the scarcity of 3D articulated object datasets, we propose FreeArt3D, a novel training-free pipeline for generating 3D articulated objects without requiring any specific dataset. We employ a per-shape optimization strategy by repurposing a pretrained 3D diffusion model—originally developed for static object generation—as a 3D guidance prior. This enables the reconstruction of high-quality 3D articulated objects from sparse input views within minutes. Unlike recent methods that rely on part retrieval or template meshes and often miss fine details and overlook texture, our approach strictly follows the input prompts and produces high-quality results with faithful geometry, realistic textures, and accurate articulation structures.

Articulated 3D objects are central to many applications in robotics, AR/VR, and animation. Recent approaches to modeling such objects either rely on optimization-based reconstruction pipelines that require dense-view supervision or on feed-forward generative models that produce coarse geometric approximations and often overlook surface texture. In contrast, open-world 3D generation of static objects has achieved remarkable success, especially with the advent of native 3D diffusion models such as Trellis. However, extending these methods to articulated objects by training native 3D diffusion models poses significant challenges. In this work, we present FreeArt3D, a training-free framework for articulated 3D object generation. Instead of training a new model on limited articulated data, FreeArt3D repurposes a pre-trained static 3D diffusion model (e.g., Trellis) as a powerful shape prior. It extends Score Distillation Sampling (SDS) into the 3D-to-4D domain by treating articulation as an additional generative dimension. Given a few images captured in different articulation states, FreeArt3D jointly optimizes the object's geometry, texture, and articulation parameters—without requiring task-specific training or access to large-scale articulated datasets. Our method generates high-fidelity geometry and textures, accurately predicts underlying kinematic structures, and generalizes well across diverse object categories. Despite following a per-instance optimization paradigm, FreeArt3D completes in minutes and significantly outperforms prior state-of-the-art approaches in both quality and versatility. Code for this paper is at https://github.com/CzzzzH/FreeArt3D.

Authors' Contact Information: Chuhao Chen, University of California San Diego, USA, chc091@ucsd.edu; Isabella Liu, University of California San Diego, USA, lal005@ucsd.edu; Xinyue Wei, University of California San Diego, USA and Hillbot Inc., USA, xiwei@ucsd.edu; Hao Su, University of California San Diego, USA and Hillbot Inc., USA, haosu@ucsd.edu; Minghua Liu, Hillbot Inc., USA, m@hillbot.ai.

SA Conference Papers '25, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-2137-3/2025/12
https://doi.org/10.1145/3757377.3763845

ACM Reference Format:
Chuhao Chen, Isabella Liu, Xinyue Wei, Hao Su, and Minghua Liu. 2025. FreeArt3D: Training-Free Articulated Object Generation using 3D Diffusion. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25), December 15–18, 2025, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3757377.3763845

1 Introduction

Articulated 3D objects—ranging from everyday items like glasses and phones to larger assets such as appliances and furniture—are ubiquitous in our surroundings. Accurately capturing their geometry, appearance, and underlying kinematic structures is crucial for a wide range of applications, including robotics, digital twins, AR/VR, and animation pipelines.

Building on this need, recent efforts in articulated 3D generation and reconstruction have followed two main directions. One line of work [Liu et al. 2023d; Tseng et al. 2022; Wei et al. 2022] focuses on optimization-based methods—often leveraging NeRFs or 3D Gaussian Splatting—to recover fine-grained articulated geometry. While these techniques can produce decent results, they are computationally intensive, require dense multi-view supervision, limiting their scalability. In parallel, several feed-forward models [Chen et al. 2024b; Le et al. 2024; Liu et al. 2024a] have been proposed to directly reconstruct or generate articulated objects from more accessible inputs such as object categories, articulation graphs, or even single-view images and text prompts. Although faster, many of these models rely on coarse geometric approximations using bounding boxes, predefined templates, or small part retrieval databases. As a result, the reconstructed shapes often lack structural realism and fine detail, and many methods fail to model surface texture, further diminishing visual fidelity.

In contrast, open-world 3D generation of static objects has seen greater success. Early efforts leveraged 2D priors from pre-trained diffusion models—either through score distillation techniques [Poole et al. 2022] or by fine-tuning to enable multi-view synthesis [Liu et al. 2023e]—followed by the training of sparse-view reconstruction models [Liu et al. 2023f]. More recently, several open-world approaches [Xiang et al. 2024; Zhang et al. 2024b] have emerged that train native 3D diffusion models directly in the 3D domain. Notably, models like Trellis [Xiang et al. 2024] have demonstrated that by adopting effective 3D representations, carefully designing 3D architectures and algorithms, and scaling up training data, it is possible to generate high-quality 3D geometry with realistic textures, while maintaining strong generalization across diverse object categories.

Extending the success of static object generation to articulated 3D objects remains largely underexplored. A natural approach is to train a native 3D diffusion model tailored for articulated objects, but two key challenges arise. First, articulated objects require a representation that jointly captures multi-part geometry and kinematic structure. Second, existing datasets [Xiang et al. 2020] are limited—typically containing only a few thousand shapes—several orders of magnitude smaller than those for static objects, making it difficult to train diffusion models effectively.

To address these challenges, we introduce FreeArt3D, a training-free framework for articulated 3D object generation from sparse input views. Rather than training a new model from scratch, FreeArt3D leverages a pre-trained 3D diffusion model—such as Trellis—originally trained on static objects, as a general 3D shape prior. It extends the concept of Score Distillation Sampling (SDS), which was originally developed for guiding 3D generation using 2D diffusion models, into the 3D-to-4D domain by treating articulation as an additional generative dimension.

Given a few images of an object captured at different articulation states, FreeArt3D optimizes an articulated object representation from scratch. This representation includes separate geometries for the static body and the movable part, joint-related parameters (e.g., rotation axis and pivot point), and the joint state corresponding to each input image. By transforming the component geometries according to the estimated joint parameters and per-image joint configurations, we synthesize the 3D object in each articulation state. These synthesized 3D models, paired with their corresponding input images, are then fed into the 3D diffusion model to compute guidance signals that drive the optimization of geometry and joint representations. To ensure robust convergence and high-quality textured mesh, we introduce carefully designed strategies for shape normalization, initialization, and post-processing.

FreeArt3D enables high-fidelity articulated object generation without requiring task-specific training or access to large-scale articulated datasets. We compare FreeArt3D with recent state-of-the-art models and show that our method significantly outperforms them by a large margin in terms of geometry, appearance, and kinematic structure metrics. FreeArt3D not only generates higher-quality geometry and textures that better adhere to user input but also predicts more accurate articulated structures. Moreover, unlike previous methods that are limited to a small set of predefined categories, our approach demonstrates stronger generalizability and performs well across a wide range of object types, as shown in Figure 1. Although our method adopts a per-shape optimization paradigm, it completes within a reasonable time—typically just a few minutes. We also demonstrate that our method can be easily extended to support multiple joints and achieves robust performance on real-world captured images.

2 Related Work

2.1 Understanding, Reconstruction, and Generation of 3D Articulated Objects

Prior research has extensively investigated the understanding of articulated objects, focusing on tasks such as detecting movable parts from a single image [Jiang et al. 2022b; Sun et al. 2023] or from video sequences [Qian et al. 2022]. A substantial body of work targets joint parameter estimation—such as joint axes, pivot points, and articulation angles—given various forms of input, including point clouds [Fu et al. 2024; Jiang et al. 2022a; Li et al. 2020; Liu et al. 2023a; Wang et al. 2019; Xu et al. 2022; Yan et al. 2020], RGB-D images [Abbatematteo et al. 2019; Che et al. 2024; Hu et al. 2017; Jain et al. 2022a, 2021; Liu et al. 2022], video [Liu et al. 2020], or 4D dynamic point clouds [Liu et al. 2023b]. Beyond static analysis, active perception methods have been proposed to facilitate more accurate estimation [Yan et al. 2023; Zeng et al. 2024]. Recently, vision-language models (VLMs) have also been finetuned for joint estimation tasks [Huang et al. 2024]. Several methods address temporal modeling of articulated objects, such as tracking and pose estimation over time [Heppert et al. 2022; Weng et al. 2021].

Beyond understanding, numerous methods aim to reconstruct or generate articulated objects. A popular approach is per-instance

optimization using Neural Radiance Fields (NeRF) [Deng et al. 2024; Liu et al. 2023d; Mu et al. 2021; Song et al. 2024; Swaminathan et al. 2024; Tseng et al. 2022; Wang et al. 2024b; Weng et al. 2024; Wu et al. 2022] or 3D Gaussian Splatting [Wu et al. 2025]. While effective in capturing object-specific geometry and appearance, they are typically slow to optimize and rely on densely sampled, posed images across multiple articulation states—conditions that are challenging to satisfy in real-world scenarios, thus limiting their scalability.

To improve scalability, recent works explore feedforward approaches for articulated object reconstruction and generation from object categories, articulation graphs [Liu et al. 2024c], single images [Chen et al. 2024b; Dai et al. 2024; Kawana and Harada 2023; Liu et al. 2024a], multi-view images [Gadi Patil et al. 2023; Heppert et al. 2023; Mandi et al. 2024; Zhang et al. 2021], text [Su et al. 2024], or static 3D mesh [Qiu et al. 2025]. These models leverage diffusion models [Gao et al. 2024; Lei et al. 2023; Luo et al. 2024], vision-language models [Le et al. 2024], or large-scale reconstruction frameworks [Gao et al. 2025]. However, they often simplify object geometry using coarse approximations such as bounding boxes, template parts, or retrieval from small databases, which limits their ability to capture fine-grained and realistic shape details. Moreover, most of these methods focus solely on geometry, neglecting the reconstruction of accurate textures, which reduces the visual fidelity and realism of the results.

2.2 Optimization-Based 3D and 4D Generation

Early open-world 3D generation methods rely on per-shape optimization, iteratively refining 3D representations to match input text or images using supervision from pre-trained 2D models (e.g., CLIP [Radford et al. 2021], Stable Diffusion [Rombach et al. 2022]) via differentiable rendering. DreamFusion [Poole et al. 2022] pioneered this direction with Score Distillation Sampling (SDS), which distills denoising gradients from diffusion models. Despite their flexibility, these methods [Chen et al. 2023; Jain et al. 2022b; Lin et al. 2023; Sanghi et al. 2022; Wang et al. 2023a,b] often suffer from multi-view inconsistency (e.g., the Janus problem) and slow convergence.

Recent works extend SDS-based optimization to 4D generation [Bahmani et al. 2024; Jiang et al. 2023; Li et al. 2024; Ren et al. 2023; Singer et al. 2023; Sun et al. 2024; Zhang et al. 2024a; Zhao et al. 2023], enabling dynamic scene synthesis by jointly optimizing spatial and motion parameters (e.g., deformation fields) without explicit motion supervision. These methods leverage text-to-image or text-to-video diffusion models and apply losses across time steps or camera views to ensure both appearance and motion consistency.

While prior work focuses on general dynamic objects, such as character animations, we target articulated object reconstruction, which involves rigid part-based motion and requires disentangled geometry and articulation. Instead of relying on 2D or video diffusion models—often prone to 3D inconsistencies—we utilize static 3D diffusion priors [Xiang et al. 2024] to guide full-shape generation, significantly improving consistency across views and poses.

2.3 Feed-Forward 3D Static Object Generation

Feed-forward 3D generation methods have recently gained prominence, offering significant improvements in speed and quality by learning from large 3D datasets [Deitke et al. 2023a,b] to directly synthesize 3D representations without iterative optimization. These methods generally fall into two main paradigms. The 2D-lifting-to-3D paradigm either trains models to reconstruct 3D shapes directly from single image inputs [Hong et al. 2023; Tochilkin et al. 2024; Zou et al. 2024] or adopts a two-stage approach: first generating multi-view images using 2D priors [Liu et al. 2023e,c; Long et al. 2024; Shi et al. 2023a,b], followed by feed-forward 3D reconstruction from sparse views [Li et al. 2023; Liu et al. 2024b, 2023f, 2024d; Tang et al. 2024; Wang et al. 2024a; Wei et al. 2024; Wu et al. 2024b; Xu et al. 2024]. More recently, the 3D-native generation paradigm—pioneered by models [Wu et al. 2024a] such as CLAY [Zhang et al. 2024b], Trellis [Xiang et al. 2024], and Hunyuan3D [Zhao et al. 2025]—directly models 3D geometry using generative techniques, often through a two-step process of geometry synthesis followed by texture generation. Enabled by large-scale datasets and advances in 3D representations, VAEs, diffusion models, and autoregressive methods [Chen et al. 2024a; Siddiqui et al. 2024], this paradigm achieves more accurate geometry and higher-quality textures.

3 Method

3.1 Overview

We propose *FreeArt3D*, a training-free framework for articulated 3D object generation that formulates the problem as an optimization task per object instance, as shown in Figure 2. Instead of training a feedforward model, we leverage the prior of a pre-trained 3D diffusion model (e.g., Trellis [Xiang et al. 2024]), originally developed for static object generation, to supervise the optimization of articulated shapes. Given a sparse set of K input RGB images $\{\mathbf{I}_k\}_{k=1}^{K}$ of an articulated object captured under varying articulation states $\{\theta_k\}_{k=1}^{K}$, our goal is to reconstruct high-fidelity textured meshes of the object along with its underlying joint structure.

For simplicity, we consider objects with a single joint, and decompose the object into two components: a static body with textured mesh $\mathcal{M}_{\text{body}}$, and a movable part with textured mesh $\mathcal{M}_{\text{part}}$. FreeArt3D takes the joint type as input and jointly optimizes these geometries along with the joint parameters $\mathcal{J}$, which vary depending on the joint type. For *revolute* joints, $\mathcal{J}$ includes a unit rotation axis vector $\mathbf{a} \in \mathbb{R}^3$ and a pivot point $\mathbf{p} \in \mathbb{R}^3$ on the axis. The joint state $\theta_k \in \mathbb{R}$ then denotes the rotation angle (in radians) around $\mathbf{a}$. For *prismatic* joints, $\mathcal{J}$ includes only the translation axis $\mathbf{a} \in \mathbb{R}^3$. The joint state $\theta_k \in \mathbb{R}$ then denotes the translation magnitude along $\mathbf{a}$. If the joint states θ_k are not available, we additionally optimize them jointly with other variables as part of the reconstruction process.

During each optimization iteration, FreeArt3D samples an input image $\mathbf{I}_k$ and transforms the movable part according to the corresponding joint state θ_k, combining it with the static body to form an articulated mesh. These image-mesh pairs are then passed to a frozen 3D diffusion model, which provides gradient signals to guide the optimization of both geometry and joint parameters.

In Section 3.2, we introduce Trellis—a pre-trained 3D diffusion model originally developed for static object generation—which serves as the supervisory signal in our framework. Section 3.3 outlines our optimization pipeline for coarse geometry (in occupancy space), joint parameters $\mathcal{J}$, and articulation states θ_k. To ensure convergence, Section 3.4 presents our strategies for normalizing

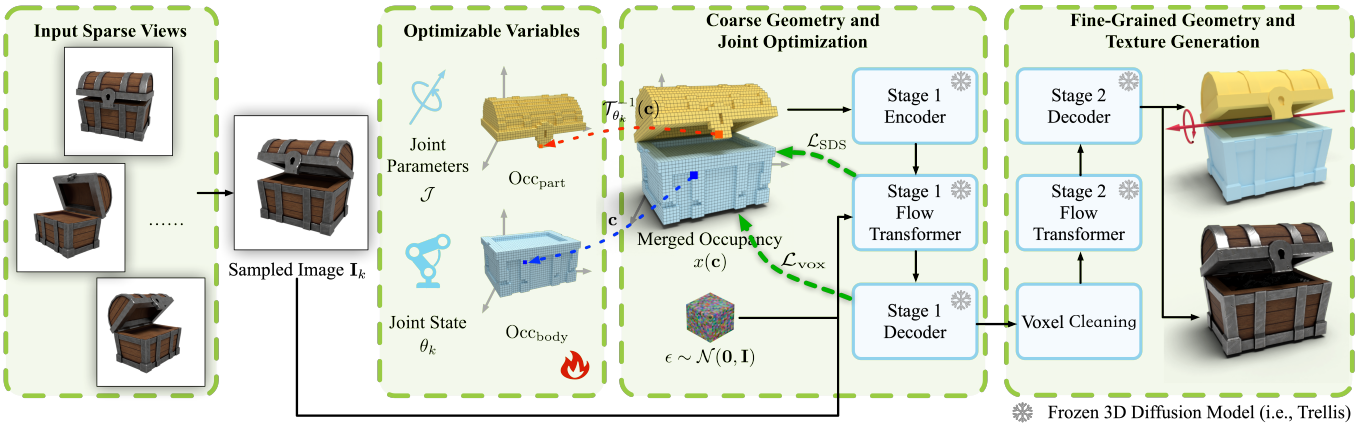

Fig. 2. FreeArt3D employs a per-shape optimization strategy. Given sparse-view images of different joint states as input, we jointly optimize two separate geometries—one for the body and one for the movable part—the joint parameters $\mathcal{J}$ (e.g., joint axis, pivot point), and optionally the joint states θ_k, if they are not provided. During the coarse geometry and joint optimization stage, we sample an image $\mathbf{I}_k$ corresponding to joint state θ_k at each iteration and aim to construct an occupancy grid of the full object under this configuration. To achieve this, we query two hash grids and transform the coordinates according to the current joint parameters $\mathcal{J}$ and joint state θ_k. The merged occupancy grid is then passed to a pretrained 3D diffusion model, Trellis [Xiang et al. 2024], along with the image $\mathbf{I}_k$ to provide gradient guidance for optimization. After completing the coarse stage, we clean the merged, optimized voxels and input them into the pretrained second-stage diffusion and VAE models to generate fine-grained geometry and realistic textures.

shape scale across joint states and initializing joint parameters and geometries. These two strategies enhance the robustness of the optimization process. Finally, Section 3.5 details the post-processing steps used to clean the optimized occupancy grid and generate high-quality meshes with fine-grained geometry and textures.

3.2 Preliminary: Trellis – 3D Diffusion for Static Objects

Trellis is a two-stage 3D diffusion framework designed to generate high-quality, textured 3D meshes of static objects from either a single image or a text prompt.

Stage 1: Coarse Geometry Generation. Trellis first models the coarse 3D structure using an occupancy grid of resolution 64^3. A variational autoencoder (VAE) is trained to encode and decode this grid through a bottleneck latent:

$$z = \text{E}_{\text{occ}}(x) \in \mathbb{R}^{16\times16\times16\times c}, \quad \hat{x} = \text{D}_{\text{occ}}(z), \tag{1}$$

where $x \in \mathbb{R}^{64\times64\times64}$ is the input occupancy grid, E_{occ} and D_{occ} are the encoder and decoder of the VAE respectively, z is the latent representation with spatial resolution 16^3 and feature dimension c, and $\hat{x}$ is the reconstructed occupancy grid. A rectified flow model $\mathcal{RF}_{\text{occ}}$ is then trained in this latent space to enable conditional generation.

Stage 2: Detailed Geometry and Texture. To capture fine-grained geometry and appearance, Trellis constructs a sparse feature volume

$$\mathcal{F} = \{(\mathbf{x}_i, \mathbf{f}_i)\}_{i=1}^{N}, \quad \mathbf{x}_i \in \{0, \dots, 63\}^3, \quad \mathbf{f}_i \in \mathbb{R}^d,$$

where $\mathcal{F}$ is a set of N feature-voxel pairs, $\mathbf{x}_i$ denotes the 3D voxel coordinate of the i-th occupied voxel, and $\mathbf{f}_i$ is the associated d-dimensional feature vector extracted from multi-view DINO embeddings. This sparse volume is encoded into a sparse latent representation using a sparse encoder–decoder pair:

$$\mathcal{Z}' = \text{E}_{\text{spa}}(\mathcal{F}) = \{(\mathbf{x}_i, \mathbf{z}'_i)\}_{i=1}^{N}, \quad \mathbf{z}'_i \in \mathbb{R}^{d'}, \quad \hat{\mathcal{F}} = \text{D}_{\text{spa}}(\mathcal{Z}'), \tag{2}$$

where E_{spa} and D_{spa} are the sparse encoder and decoder, respectively. $\mathcal{Z}'$ denotes the sparse latent feature representation, where each $\mathbf{z}'_i$ is a d'-dimensional embedding corresponding to voxel $\mathbf{x}_i$. The decoder reconstructs the sparse feature volume $\hat{\mathcal{F}}$ from $\mathcal{Z}'$.

A second rectified flow model, $\mathcal{RF}_{\text{spa}}$, is trained on this sparse latent space to enable conditional generation. The decoder, D_{spa}, then upsamples the sparse latent representation into a high-resolution sparse volume of size 256^3. Trellis employs different decoders to support multiple outputs—for example, predicting FlexiCubes coefficients for explicit mesh generation or 3D Gaussian Splatting parameters for textured rendering.

3.3 Optimization of Coarse Geometry and Joint

Formulation. This section focuses on optimizing the coarse geometry of both the static body $\mathcal{M}_{\text{body}}$ and the movable part $\mathcal{M}_{\text{part}}$ within the occupancy space, which corresponds to the first stage of Trellis. We also jointly optimize the joint parameters $\mathcal{J}$ as defined in Section 3.1. Fine-grained geometry and texture generation are discussed in Section 3.5.

To represent coarse geometry, we employ two continuous and optimizable multi-level hash grids: $\mathcal{H}_{\text{body}}$ and $\mathcal{H}_{\text{part}}$, which model the occupancy fields of the static body and movable part respectively, in the canonical (rest) frame. Each grid maps a 3D coordinate $\mathbf{c} \in \mathbb{R}^3$ to a continuous occupancy value in $[0, 1]$:

$$\text{Occ}_{\text{body}}(\mathbf{c}) = \mathcal{H}_{\text{body}}(\mathbf{c}), \quad \text{Occ}_{\text{part}}(\mathbf{c}) = \mathcal{H}_{\text{part}}(\mathbf{c}). \tag{3}$$

Occupancy Grid Construction. At each optimization iteration, we randomly sample an input image $\mathbf{I}_k$ and its associated joint state θ_k. A 64^3 occupancy grid x is then constructed by querying the values of both hash grids at the voxel coordinate $\mathbf{c}$ in the posed frame and taking the maximum occupancy value:

$$x(\mathbf{c}) = \max\left(\text{Occ}_{\text{body}}(\mathbf{c}),\ \text{Occ}_{\text{part}}(\mathcal{T}_{\theta_k}^{-1}(\mathbf{c}))\right), \tag{4}$$

where $\mathcal{T}_{\theta_k}^{-1}$ denotes the inverse articulation transform, mapping voxel coordinate $\mathbf{c}$ from the posed frame back to the canonical frame based on the joint parameters $\mathcal{J}$ and joint state θ_k. Specifically, for a

revolute joint, the transformation is: $\mathcal{T}_{\theta_k}^{-1}(\mathbf{c}) = \mathbf{R}_{\theta_k}^{-1}(\mathbf{c}-\mathbf{p})+\mathbf{p}$, where $\mathbf{p}$ is the pivot point and $\mathbf{R}_{\theta_k}$ is the rotation around axis $\mathbf{a}$ by angle θ_k. For a prismatic joint, the transformation is: $\mathcal{T}_{\theta_k}^{-1}(\mathbf{c}) = \mathbf{c} - \theta_k \cdot \mathbf{a}$, where $\mathbf{a}$ is the translation axis.

Loss Functions. The constructed occupancy grid x is then encoded into a latent representation using the VAE encoder $\mathrm{E}_{\mathrm{occ}}$, i.e., $z = \mathrm{E}_{\mathrm{occ}}(x)$. This latent vector z is paired with the input image $\mathbf{I}_k$ and passed to the frozen occupancy rectified flow model $\mathcal{RF}_{\mathrm{occ}}$ of Trellis, which provides gradient supervision indicating what a valid 3D shape corresponding to $\mathbf{I}_k$ should be. Following the Score Distillation Sampling (SDS) procedure from DreamFusion [Poole et al. 2022], we perturb the latent with noise and compute the SDS loss:

$$\nabla_{\psi}\mathcal{L}_{\mathrm{SDS}}(\mathcal{RF}_{\mathrm{occ}}, \mathbf{z}) \triangleq \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[w(t)\cdot\left(\hat{\boldsymbol{\epsilon}}_{\mathcal{RF}_{\mathrm{occ}}}(\mathbf{z}_t;\mathbf{I}_k,t)-\boldsymbol{\epsilon}\right)\cdot\frac{\partial \mathbf{z}_t}{\partial \psi}\right], \quad (5)$$

where ψ denotes all optimizable parameters, including the weights of both hash grids and the joint parameters $\mathcal{J}$, $\mathbf{z} = \mathrm{E}_{\mathrm{occ}}(x)$ is the latent encoding of the occupancy grid from Equation 4, $\mathbf{z}_t$ is the noisy latent at time t, $\boldsymbol{\epsilon} \sim \mathcal{N}(0,\mathbf{I})$ is a sampled Gaussian noise, $\hat{\boldsymbol{\epsilon}}_{\mathcal{RF}_{\mathrm{occ}}}(\mathbf{z}_t;\mathbf{I}_k,t)$ is the noise predicted by the rectified flow model conditioned on the image and timestep, $w(t)$ is a weight function balancing contributions across diffusion steps.

Additionally, inspired by [Zhu and Zhuang 2023], we introduce a voxel-space reconstruction loss to encourage consistency between the synthesized occupancy x and the denoised prediction produced by the rectified flow model $\mathcal{RF}_{\mathrm{occ}}$:

$$\mathcal{L}_{\mathrm{vox}} = \|\mathrm{D}_{\mathrm{occ}}(\hat{\mathbf{z}}_0) - x\|_2^2, \quad (6)$$

where $\hat{\mathbf{z}}_0$ is the denoised latent derived from the predicted noise $\hat{\epsilon}_{\mathcal{RF}}(\mathbf{z}_t;\mathbf{I}_k,t)$ by the occupancy rectified flow model. This voxel-based loss improves the stability of the optimization process.

The total loss for optimizing the coarse geometry and joint parameters is given by:

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{SDS}}\cdot\mathcal{L}_{\mathrm{SDS}} + \lambda_{\mathrm{vox}}\cdot\mathcal{L}_{\mathrm{vox}}, \quad (7)$$

where λ_{SDS} and λ_{vox} are scalar weights. This optimization is repeated over randomly sampled views, progressively refining the geometry and joint parameters of the object.

Discussion. While original SDS methods utilize 2D diffusion models to guide 3D generation, FreeArt3D can be viewed as a natural extension of 3D diffusion guidance to 4D generation by treating the articulation state as an additional dimension. Unlike traditional SDS-based pipelines, which suffer from multi-view inconsistencies—such as the Janus problem—due to limited and ambiguous 2D supervision, our method avoids these issues by directly providing the full 3D shape corresponding to each joint state to the 3D diffusion model. This richer supervision signal reduces ambiguity, improves consistency, and significantly accelerates convergence, while our explicit kinematic transformations ensure structural correctness and effective disentanglement of different geometries.

3.4 Normalization and Initialization

Normalization across Joint States. A key challenge in optimizing articulated objects lies in the significant scale variation across different joint states. This issue arises because Trellis normalizes all training shapes to fit within a unit bounding box. Consequently, the same object in different articulation states—such as a closed cabinet versus one with an open door—may result in drastically different scales for the static body part, depending on how much the movable part extends, as shown in Figure 3. Such scale inconsistencies can severely hinder optimization, as they disrupt the geometric correspondence across joint states and may even cause optimization failure (e.g., collapsing to zero or full occupancy).

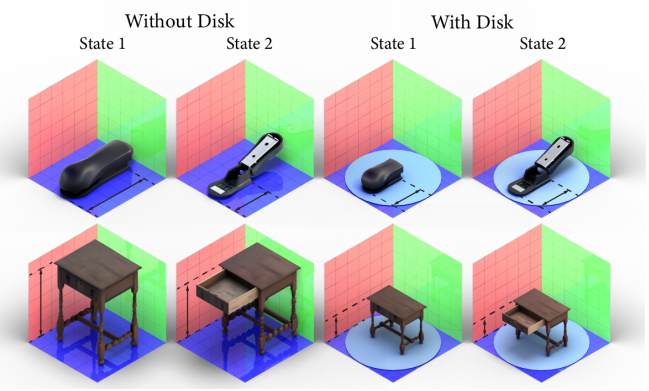

Fig. 3. **Trellis inference results across different joint states.** Since Trellis is trained on 3D data normalized to a unit cube, each articulated component (e.g., the body of a stapler or a desk) may appear at different scales across joint states, depending on how far the movable parts (e.g., the stapler handle or desk drawer) extend. This scale inconsistency hinders optimization convergence. To address this issue, we introduce a disk beneath the object that serves as a reference to support the entire cube, ensuring consistent component scales across different states.

To mitigate this, we adopt a simple yet effective strategy inspired by common practices in 3D asset creation: we add a large, fixed reference disk (resembling a carpet or floor) beneath the object in all input views. This auxiliary geometry acts as a visual anchor, providing a consistent reference for scale and spatial alignment across different joint states. By including this "carpet" in every view, the 3D diffusion model learns to interpret object scale relative to a fixed element, effectively stabilizing the predicted size of the articulated components throughout the optimization. Importantly, the carpet can be safely removed after convergence, ensuring it does not affect the final reconstruction results. This normalization strategy leads to more consistent geometric predictions across varying articulation configurations and significantly improves convergence stability.

Initialization of Joint Parameters and Geometry. The initialization of both the joint parameters and the geometry plays a crucial role in ensuring robust convergence during optimization. To obtain initial estimates of the joint parameters $\mathcal{J}$ and joint states θ_i (if not provided), we employ a simple yet effective heuristic based on cross-state correspondences. Specifically, we first run full Trellis inference independently on each input image to generate textured meshes corresponding to different joint states. These meshes are then rendered into 2D images, and we apply an off-the-shelf 2D correspondence detector, such as LoFTR [Sun et al. 2021], to identify pixel-level correspondences across views. The resulting 2D matches are subsequently lifted into 3D point pairs. We then filter out static point pairs and retain only those corresponding to the movable part. Given these 3D point pairs, we estimate $\mathcal{J}$ and θ_i by constructing transformations based on these parameters and minimizing the 3D

distances between the corresponding points. While this procedure may yield imperfect estimates—due to factors such as inconsistencies in the generated meshes or spurious correspondences—it nonetheless provides a strong initialization for subsequent optimization, as both $\mathcal{J}$ and θ_i will be further refined jointly with the geometry.

For geometry initialization, we use the occupancy grid inferred from the image corresponding to the initial joint state as a proxy for both the static body and the movable part. This provides a reasonable geometric scaffold from which the optimization process can proceed.

3.5 Fine-grained Geometry and Texture Generation.

After optimizing the coarse geometry and articulation parameters as described in Sec. 3.3, we proceed to generate high-fidelity textured meshes for both the static and movable parts using Stage 2 of Trellis.

Occupancy Denoising and Cleaning. We first construct the merged occupancy grid x_{max} corresponding to the largest joint state θ_{max}, using the optimized hash grids and Equation 4. To improve voxel quality, we inject small Gaussian noise into x_{max} and denoise it via a single forward pass through the occupancy rectified flow model $\mathcal{RF}_{\text{occ}}$, yielding a cleaner occupancy grid $\tilde{x}$. We then remove auxiliary geometry—specifically the "carpet plane"—by detecting and discarding voxels near the lowest z-values in $\tilde{x}$. Finally, we filter outlier voxels based on spatial isolation and size thresholds to produce a clean voxel grid $\bar{x}$.

Sparse Latent Construction and Decoding. From the cleaned occupancy grid $\bar{x}$, we extract the set of occupied voxel coordinates, and construct an initial sparse latent feature volume by sampling Gaussian noise. This initial sparse latent is then denoised using the second-stage flow model $\mathcal{RF}_{\text{spa}}$ of Trellis. We then decode the denoised latent using the sparse decoder D_{spa} to produce a sparse feature volume $\hat{\mathcal{F}}$, containing both FlexiCubes coefficients for mesh extraction and Gaussian Splatting parameters for texture synthesis.

Mesh Extraction and Texture Baking. To recover part-specific geometries, we partition the decoded sparse volume $\hat{\mathcal{F}}$ into two subsets corresponding to the static body and the movable part. Specifically, we examine the transformed occupancy grids of the two parts, assigning each cell to the part with higher occupancy. For each part, we extract a detailed mesh from the associated FlexiCubes representation. We then bake the surface textures onto each mesh using the decoded Gaussian Splatting parameters of the complete shape. Finally, we combine the resulting textured meshes with the optimized joint parameters $\mathcal{J}$ to form the complete articulated object with fine-grained geometry and realistic appearance. This concludes the FreeArt3D pipeline.

4 Experiment

4.1 Implementation Details and Evaluation Setup

Implementation Details. For each shape, we use $K = 6$ input images $\mathbf{I}_k$, each corresponding to a different joint state θ_k, sampled between the rest and maximum configurations. Our method only requires images captured at different articulation states, not necessarily from different camera views, although view variation is allowed, as shown in the teaser. In our experiments, however, the camera pose is fixed to a single side–upper view. We assume the joint states θ_k are unknown and jointly optimize them along with the joint parameters $\mathcal{J}$. Optimization is performed over 3000 iterations per shape using a learning rate of 0.01 and the Adam optimizer [Kingma 2014]. The loss weights λ_{SDS} and λ_{VOX} are set to 0.1 and 1.0, respectively. SDS noise is sampled from the range $[0.5, 0.8]$ before scaling. To estimate the initial joint parameters, we render 18 images—3 different views per state—using Blender, and select 36 image pairs in total to build correspondences and estimate the initialization. We adopt TinyCudaNN [Müller 2021] for hash grid implementation. To clean the occupancy voxels, we inject small noise of 0.5 before scaling. The total runtime for processing a single shape is approximately 10 minutes.

Evaluation Dataset and Metric We evaluate on the PartNet-Mobility dataset [Xiang et al. 2020], focusing on 12 categories: box, dishwasher, laptop, lighter, microwave, oven, refrigerator, safe, stapler, storage furniture, table, and washing machine. For each category, we randomly sample 12 shapes, resulting in 144 test shapes in total. Given a generated geometry pair (body and movable parts) and predicted joint parameters, we compute five metrics: Chamfer Distance (CD) [Fan et al. 2017], F-Score [Wang et al. 2018], CLIP similarity, joint axis direction error, and joint pivot error. For each shape, we sample 6 joint states and generate corresponding meshes using the estimated joint parameters. We compute the metrics for each state and report their average.

Before evaluation, we align the generated mesh and ground-truth mesh using the method from [Liu et al. 2024b]. CD and F-Score measure geometric similarity—CD is computed by sampling 100k surface points, while F-Score uses a 0.05 threshold. For appearance, we render 5 views of both predicted and ground-truth meshes and compute the average CLIP similarity [Radford et al. 2021] using ViT-L/14@336px. Following ArticulateAnything [Le et al. 2024], we report the joint axis direction error as the angle between the predicted axis $\mathbf{a}_p$ and the ground-truth axis $\mathbf{a}_g$ for both revolute and prismatic joints:

$$e_{\text{axis}} = \min\left(\arccos\left(\frac{\mathbf{a}_p \cdot \mathbf{a}_g}{\|\mathbf{a}_p\|_2 \|\mathbf{a}_g\|_2}\right), \arccos\left(\frac{-\mathbf{a}_p \cdot \mathbf{a}_g}{\|\mathbf{a}_p\|_2 \|\mathbf{a}_g\|_2}\right)\right), \tag{8}$$

where $e_{\text{axis}} \in [0, \pi]$. For revolute joints, the joint pivot error is defined as the shortest distance between the predicted and ground-truth joint axes:

$$e_{\text{pivot}} = \frac{|\mathbf{p} \cdot (\mathbf{a}_p \times \mathbf{a}_g)|}{|\mathbf{a}_p \times \mathbf{a}_g|}, \tag{9}$$

where $\mathbf{p} = \mathbf{x}_p - \mathbf{x}_g$ denotes the vector difference between the predicted and ground-truth pivot points. Note that CD and F-Score also implicitly reflect joint prediction accuracy, as they are evaluated per joint state.

4.2 Comparison with Baselines

Baselines We compare our method against four state-of-the-art approaches for modeling articulated objects: Articulate-Anything [Le et al. 2024], Singapo [Liu et al. 2024a], URDFormer [Chen et al. 2024b], and PARIS [Liu et al. 2023d]. The first three methods take a single-view image as input, whereas PARIS reconstructs from multi-view images. Both Articulate-Anything and Singapo assemble articulated objects by retrieving parts from the PartNet-Mobility dataset [Xiang et al. 2020] and applying the textures of the retrieved parts. URDFormer, on the other hand, generates articulated objects

Fig. 4. **Comparison between Singapo [Liu et al. 2024a], Articulate-Anything [Le et al. 2024], and Ours.** Unlike baseline methods that rely on part retrieval and fail to reconstruct detailed geometry and textures, our method generates meshes that closely match the input images and successfully recover fine-grained geometric details, realistic textures, and accurate articulation structures.

Table 1. **Quantitative Comparison between ParisArticulate-Anything [Le et al. 2024], Singapo [Liu et al. 2024a], URDFormer [Chen et al. 2024b], PARIS [Liu et al. 2023d] and Ours.** "–" indicates that the baseline method fails to handle the given category or the metric is not available. "Average-5" refers to the average performance over the five categories that all methods can handle. "Average-12" refers to the average performance over the twelve categories that Articulate-Anything, PARIS and our method can handle.

		Box	Dishwasher	Laptop	Lighter	Microwave	Oven	Refrigerator	Safe	Stapler	StorageFurniture	Table	WashingMachine	Average-5	Average-12
Ours	F-Score	**0.915**	**0.929**	**0.845**	**0.849**	**0.891**	**0.869**	**0.912**	**0.915**	**0.922**	**0.873**	**0.950**	0.834	**0.883**	**0.891**
	CD	**0.023**	**0.021**	**0.037**	**0.028**	**0.026**	**0.026**	**0.024**	**0.021**	**0.019**	0.026	**0.018**	0.031	**0.026**	**0.025**
	Clip-Sim	**0.883**	**0.879**	**0.871**	**0.839**	**0.885**	**0.877**	**0.885**	**0.883**	**0.881**	**0.897**	**0.922**	0.872	**0.882**	**0.881**
	Joint-Axis-Err	**0.021**	**0.028**	**0.143**	**0.040**	**0.021**	**0.170**	0.151	**0.023**	**0.039**	**0.039**	0.535	0.654	**0.208**	**0.159**
	Joint-Pivot-Err	**0.140**	**0.075**	**0.071**	**0.138**	0.150	**0.094**	**0.029**	**0.040**	**0.305**	-	-	0.171	**0.092**	**0.121**
ArtAnything	F-Score	0.642	0.761	0.693	0.760	0.781	0.808	0.824	0.719	0.754	0.847	0.748	**0.848**	0.817	0.769
	CD	0.062	0.052	0.056	0.041	0.038	0.035	0.035	0.049	0.049	0.029	0.042	**0.030**	0.036	0.043
	Clip-Sim	0.837	0.841	0.805	0.792	0.855	0.858	0.865	0.843	0.828	0.938	0.857	**0.877**	0.876	0.851
	Joint-Axis-Err	0.564	0.651	0.227	0.264	0.487	0.958	0.164	0.372	1.119	0.182	0.287	0.681	0.527	0.499
	Joint-Pivot-Err	0.265	0.179	0.227	0.141	**0.057**	0.122	0.074	0.087	0.586	-	-	0.168	0.136	0.191
Singapo	F-Score	-	0.848	-	-	0.805	0.789	0.869	-	-	0.881	0.859	0.774	0.832	-
	CD	-	0.030	-	-	0.036	0.035	0.029	-	-	**0.023**	0.026	0.036	0.031	-
	Clip-Sim	-	0.873	-	-	0.856	0.856	0.860	-	-	0.880	0.846	0.827	0.859	-
	Joint-Axis-Err	-	0.285	-	-	0.145	0.340	**0.116**	-	-	0.056	**0.081**	**0.439**	0.247	-
	Joint-Pivot-Err	-	0.095	-	-	0.077	0.147	0.034	-	-	-	-	0.101	0.094	-
URDFormer	F-Score	-	0.678	-	-	-	0.703	0.536	-	-	0.740	-	0.737	0.679	-
	CD	-	0.059	-	-	-	0.047	0.076	-	-	0.044	-	0.044	0.054	-
	Clip-Sim	-	0.846	-	-	-	0.800	0.826	-	-	0.810	-	0.805	0.817	-
	Joint-Axis-Err	-	1.301	-	-	-	1.268	1.331	-	-	1.084	-	1.306	1.258	-
	Joint-Pivot-Err	-	0.360	-	-	-	0.414	0.293	-	-	-	-	0.312	0.344	-
PARIS	F-Score	0.723	0.729	0.834	0.897	0.762	0.810	0.740	0.760	0.836	0.806	0.949	0.797	0.776	0.804
	CD	0.047	0.047	0.043	0.022	0.038	0.033	0.036	0.037	0.039	0.030	0.016	0.030	0.035	0.035
	Clip-Sim	0.755	0.790	0.762	0.785	0.776	0.771	0.807	0.775	0.779	0.751	0.778	0.798	0.783	0.777
	Joint-Axis-Err	1.090	1.334	1.014	0.490	1.413	1.283	1.212	1.464	0.421	1.334	1.537	1.072	1.247	1.139
	Joint-Pivot-Err	0.166	0.169	0.203	0.356	0.385	0.138	0.074	0.216	0.349	-	-	0.364	0.186	0.242

Table 2. **Runtime Comparison.** Runtime (seconds) of our method and baseline methods. Measured on a single NVIDIA H100 GPU.

Ours	ArtAnything	Singapo	URDFormer	PARIS
606s	200s	19s	27s	494s

by predicting bounding boxes for parts, then scaling and composing template part meshes. It directly projects the input image onto the mesh surface to serve as the texture. PARIS reconstructs the object's shape and appearance by jointly optimizing two neural fields—one for the static parts and one for the movable parts. For a fair comparison, we remove the test shapes from the retrieval database for both Articulate-Anything and Singapo. We also use a relatively sparse set of 24 views for PARIS. Among the 12 categories, Articulate-Anything and PARIS supports all, Singapo supports 7 out of 12, and URDFormer supports only 5 out of 12.

Results As shown in Figure 4, our method generates high-quality textured meshes that closely match the input images and, consequently, closely resemble the ground-truth meshes. In contrast, both Singapo and Articulate-Anything do not generate new geometry or textures for the input images; instead, they retrieve parts from a database—a process fundamentally constrained by the database's scale. Due to the scarcity of 3D articulated objects, these methods often produce results that capture only the coarse global shape while failing to reproduce fine-grained geometric details, colors, and textures. Such limitations significantly hinder their applicability in downstream tasks such as digital twins for robotics, which require accurate reconstructions of real-world environments. An interesting observation is that all retrieval-based baseline methods successfully retrieved the exact same or highly similar object from the database (Row 8), whereas our method—despite not relying on retrieval—still produces results of comparable quality. This highlights the robustness and generalization capability of our approach. Furthermore, both Singapo and Articulate-Anything frequently fail to predict accurate joint axes, as illustrated in Rows 2, 3, and 9.

Quantitative results are shown in Table 1, where our method significantly outperforms all baseline methods across all five metrics. The retrieval-based methods, Articulate-Anything and Singapo, perform considerably better than the template-mesh-based method URDFormer and optimized-based PARIS, yet still fall far short of our generative model. Notably, while baseline methods are limited to a small number of predefined categories, FreeArt3D demonstrates strong generalizability and can handle a much broader range of categories. We also report the runtimes of all methods in Table 2.

4.3 Application and Extension

Multi-Joint Extension. In our method and previous experiments, we primarily focus on articulated objects with a single prismatic or revolute joint for simplicity. However, extending our approach to more complex objects with multiple joints and mixed joint types-while still assuming a single–degree-of-freedom kinematic chain-is straightforward. To enable such an extension, several adaptations are necessary: (a) using more input images to capture the motion of each joint; (b) employing a separate hash grid to represent the occupancy of each part; and (c) jointly optimizing the geometry of all parts along with their corresponding joint parameters. As shown in Figure 6, our method supports multi-joint generation, enabling the creation of complete URDFs, which facilitates downstream applications such as robotic simulation and digital twins.

Real-World Reconstruction. Beyond standard synthetic benchmarks, we also evaluate our method on real-world captured images. For each object, we capture six images using an iPhone, with the

Fig. 5. **Real-World Demo**. For each shape, we capture six images of the object in different joint states. FreeArt3D effectively leverages these casually captured, unposed images to generate high-quality articulated objects with sharp geometry and realistic textures.

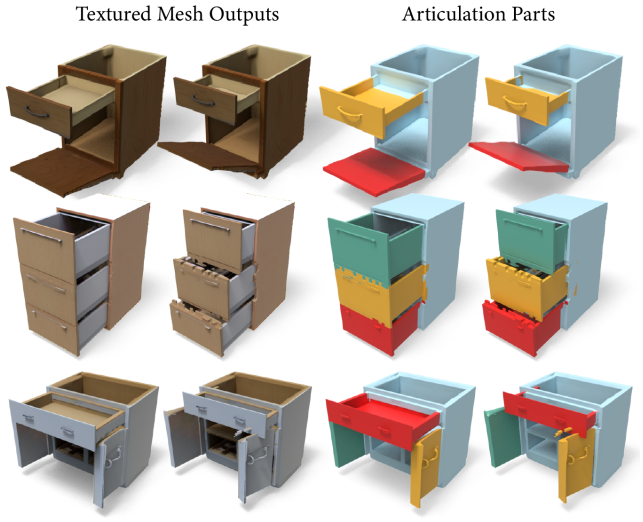

Fig. 6. **Generation Results of Multiple Parts and Joints.** Our method can be easily extended to support the generation of multiple articulated parts and joints, enabling flexible configuration of all components in the generated objects.

object in different joint states. Note that our method does not require camera poses as input, nor does it depend on dense-view images or assume full object coverage. As a result, the required sparse-view images are easy to obtain. However, we do require either a physical carpet placed beneath the object or a virtual disc added to the captured photo for normalization. Figure 5 shows example inputs and results, demonstrating that our method can potentially support a wide range of real-world applications.

Table 3. **Ablation Study.** Evaluated across all categories.

Index	Version	F-Score↑	CD ↓	CLIP-Sim ↑	Axis-Err↓	Pivot-Err↓
a	no SDS loss	0.610	0.067	0.808	0.555	0.152
b	no voxel loss	0.873	0.028	0.880	0.172	0.138
c	no disk normalization	0.749	0.048	0.854	0.730	0.244
d	rand joint initialization	0.753	0.049	0.849	1.024	0.253
e	fixed joint after initialization	0.864	0.029	0.876	0.165	0.138
f	no hashgrid	0.848	0.032	0.872	0.157	0.180
g	no voxel refinement	0.875	0.027	0.858	0.181	0.111
h	more input (6 → 21)	0.903	0.023	0.883	0.126	0.135
i	minimum input (6 → 2)	0.841	0.035	0.870	0.525	0.227
j	full	0.892	0.025	0.881	0.155	0.121

4.4 Ablation Study and Analysis

We report the quantitative results of our ablation study in Table 3.
Loss Functions. Our coarse-geometry optimization primarily relies on a combination of the SDS loss and the voxel loss. Removing the SDS loss (row (a)) and relying solely on the voxel-space L2 loss leads to a significant drop in performance, highlighting the necessity of the SDS loss. In contrast, the voxel loss (row (b)) serves as a complementary term that helps stabilize training and further improve performance.
Disk Normalization. We introduce an auxiliary disk beneath the object to promote consistent scaling of the articulated components across different joint states, as discussed in Section 3.4. The necessity of this normalization is evidenced in row (c); removing this technique results in a significant drop in performance.

Some may be concerned that adding a disk could affect Trellis's generation capability. To verify this, we evaluated Trellis on renderings of 144 3D objects with and without disks, comparing the Chamfer Distance (CD) between the generated static 3D models and the ground truth. The results show that the average CD is 0.018 with a disk and 0.024 without a disk. Interestingly, the results with disks

Table 4. **Comparison with Trellis-Enhanced Singapo. Evaluated across all categories supported by Singapo.**

Method	F-Score↑	CD↓	CLIP-Sim↑	Joint-Axis-Err↓	Joint-Pivot-Err↓
Original Singapo	0.832	0.031	0.859	0.247	0.094
Singapo w/ Trellis	0.834	0.031	**0.899**	0.255	0.109
Ours	**0.883**	**0.026**	0.882	**0.208**	**0.092**

performed slightly better, providing no evidence of capacity degradation or bias introduced by the disk. As alternative approaches for scale normalization, one could explore 3D point correspondences or fine-tuning Trellis with other normalization schemes.

Joint Initialization. In Section 3.4, we propose a preprocessing step that estimates the joint axis to serve as the initialization. In row (d), we remove this joint estimation module and instead use a random initialization, allowing the joint parameters to be optimized jointly with the occupancy volumes. We perform three trials and observe a significant performance drop, indicating that without a good initialization, relying solely on SDS and voxel losses is insufficient for accurately recovering the joint axis, as reflected in both geometric and joint metrics. In row (e), we retain the estimation module but fix the estimated joint parameters during coarse geometry optimization. This results in a slight performance drop, suggesting that our optimization process is capable of refining the initial estimated joint paramters further.

Hash Grid. In our preliminary exploration, we find that representing occupancy using a discrete 64^3 voxel grid can introduce instability during optimization. In contrast, a continuous multi-level hash grid offers a smoother and more flexible representation, leading to improved stability and effectiveness in optimization. In row (f), we compare the performance of discrete voxel grids and continuous multi-level hash grids and observe a notable performance gap.

Voxel Refinement. In Sec. 3.5, we introduce several techniques to refine the optimized occupancy voxels before passing them to the second-stage diffusion model for fine-grained geometry and texture generation. As shown in Row (g), this step further enhances performance. See Figure 7 for qualitative examples.

Number of Input Views. In our default setup, we use 6 sparse input views, which already yield robust and satisfactory results. In Row (h) and Row (i), we demonstrate that incorporating more input views (e.g., 21 views) can further improve performance slightly, while only using the minimum views still performs reasonably well.

Benefit of Trellis. Our method leverages the pre-trained 3D diffusion model Trellis [Xiang et al. 2024]. However, its success cannot be solely attributed to Trellis. In previous ablation studies, we have already demonstrated the importance of disk normalization and joint initialization. To isolate Trellis's contribution, we modified the baseline Singapo by replacing its part retrieval module with Trellis-based part generation, and the results are presented in Table 4. Note that we used ground-truth, separately rendered part images as input, whereas in practice only full-object images containing all parts are available. Because of this unrealistic input setting, Singapo's CLIP similarity is slightly better than ours. Nevertheless, our method still outperforms it on other metrics, demonstrating its overall superiority. These results indicate that directly enhancing baselines with a pre-trained 3D diffusion model is non-trivial.

Table 5. **Failure Case Analysis. The percentages indicate the proportion of failure cases, evaluated across all categories.**

Failure Type	Axis Direction↓	Pivot Point↓	Segmentation↓
Ours (w/o. disk)	45.14%	43.75%	41.67%
Ours (2-state)	32.64%	47.92%	11.81%
Ours (6-state)	**9.03%**	**11.11%**	**9.03%**

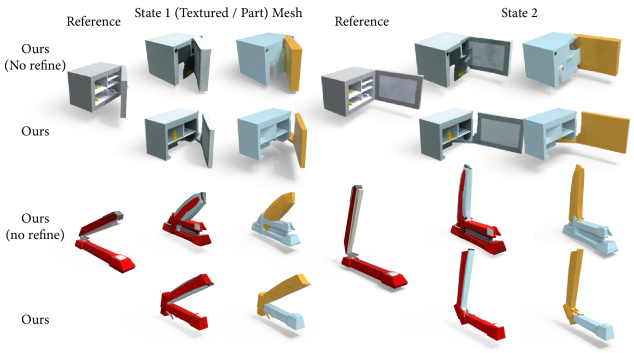

Fig. 7. **Ablation Study on Occupancy Refinement After Coarse Geometry Optimization.**

4.5 Discussion on Robustness and Failure Cases

We observe three common failure cases of our method: (1) incorrect joint axis direction (angle error > 20°), (2) inaccurate pivot point (distance error > 0.1), and (3) failed part segmentation—producing only the movable or the static part. To evaluate the robustness of our method and analyze these failures, we report the failure rates of each error type and various ablated versions in Table 5.

Overall, our method is relatively robust. With only 6 input images, the error rate for each type is around 10%, and the method achieves a 77.08% overall success rate (meeting all three criteria). We also find that robustness strongly depends on scale normalization and the quality of joint initialization. Without the proposed disk normalization, the success rate drops significantly. When using fewer input images (e.g., 2 images), the method still works in many cases, though the success rate decreases substantially, mainly due to the difficulty of initial joint estimation. Although good joint initialization is important, the joint optimization process over both geometry and joint parameters remains essential, since it can refine imperfect initializations. Remarkably, the method sometimes succeeds even with random joint initialization.

5 Conclusion and Limitation

We introduce a novel training-free pipeline for modeling articulated objects, which effectively circumvent the challenges posed by the scarcity of articulated object data. Unlike previous methods that rely on part retrieval and template meshes, our generative model enables the production of high-fidelity textured meshes that closely adhere to the input prompts. Looking ahead, it would be interesting to explore ways to further accelerate the optimization process and improve its robustness—for example, by developing more elegant solutions to normalization issues.

References

Ben Abbatematteo, Stefanie Tellex, and George Konidaris. 2019. Learning to generalize kinematic models to novel objects. In *Proceedings of the 3rd Conference on Robot Learning*.

Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B Lindell. 2024. 4d-fy: Text-to-4d generation using hybrid score distillation sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7996–8006.

Yuchen Che, Ryo Furukawa, and Asako Kanezaki. 2024. OP-Align: Object-level and Part-level Alignment for Self-supervised Category-level Articulated Object Pose Estimation. In *European Conference on Computer Vision*. Springer, 72–88.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 22246–22256.

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. 2024a. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163* (2024).

Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. 2024b. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656* (2024).

Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. 2024. Automated creation of digital cousins for robust policy learning. *arXiv preprint arXiv:2410.07408* (2024).

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

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

Jianning Deng, Kartic Subr, and Hakan Bilen. 2024. Articulate your NeRF: Unsupervised articulated object modeling via conditional view synthesis. *arXiv preprint arXiv:2406.16623* (2024).

Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 605–613.

Lian Fu, Ryoichi Ishikawa, Yoshihiro Sato, and Takeshi Oishi. 2024. CAPT: Category-level Articulation Estimation from a Single Point Cloud Using Transformer. *arXiv preprint arXiv:2402.17360* (2024).

Akshay Gadi Patil, Yiming Qian, Shan Yang, Brian Jackson, Eric Bennett, and Hao Zhang. 2023. RoSI: Recovering 3D Shape Interiors from Few Articulation Images. *arXiv e-prints* (2023), arXiv–2304.

Daoyi Gao, Yawar Siddiqui, Lei Li, and Angela Dai. 2024. MeshArt: Generating Articulated Meshes with Structure-guided Transformers. *arXiv preprint arXiv:2412.11596* (2024).

Mingju Gao, Yike Pan, Huan-ang Gao, Zongzheng Zhang, Wenyi Li, Hao Dong, Hao Tang, Li Yi, and Hao Zhao. 2025. PartRM: Modeling Part-Level Dynamics with Large Cross-State Reconstruction Model. *arXiv preprint arXiv:2503.19913* (2025).

Nick Heppert, Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Rares Andrei Ambrus, Jeannette Bohg, Abhinav Valada, and Thomas Kollar. 2023. Carto: Category and joint agnostic reconstruction of articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21201–21210.

Nick Heppert, Toki Migimatsu, Brent Yi, Claire Chen, and Jeannette Bohg. 2022. Category-independent articulated object tracking with factor graphs. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3800–3807.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2023. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400* (2023).

Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. 2017. Learning to predict part mobility from a single static snapshot. *ACM Transactions On Graphics (TOG)* 36, 6 (2017), 1–13.

Siyuan Huang, Haonan Chang, Yuhan Liu, Yimeng Zhu, Hao Dong, Peng Gao, Abdeslam Boularias, and Hongsheng Li. 2024. A3vlm: Actionable articulation-aware vision language model. *arXiv preprint arXiv:2406.07549* (2024).

Ajinkya Jain, Stephen Giguere, Rudolf Lioutikov, and Scott Niekum. 2022a. Distributional depth-based estimation of object articulation models. In *Conference on Robot Learning*. PMLR, 1611–1621.

Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. 2021. Screwnet: Category-independent articulation model estimation from depth images using screw theory. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 13670–13677.

Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. 2022b. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 867–876.

Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. 2022b. OPD: Single-view 3D Openable Part Detection. *arXiv preprint arXiv:2203.16421* (2022).

Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. 2023. Consistent4d: Consistent 360 {\deg} dynamic object generation from monocular video. *arXiv preprint arXiv:2311.02848* (2023).

Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. 2022a. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5616–5626.

Yuki Kawana and Tatsuya Harada. 2023. Detection based part-level articulated object reconstruction from single RGBD image. *Advances in Neural Information Processing Systems* 36 (2023), 18444–18473.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. 2024. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. *arXiv preprint arXiv:2410.13882* (2024).

Jiahui Lei, Congyue Deng, Bokui Shen, Leonidas Guibas, and Kostas Daniilidis. 2023. Nap: Neural 3d articulation prior. *arXiv preprint arXiv:2305.16315* (2023).

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. 2023. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214* (2023).

Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. 2020. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3706–3715.

Zhiqi Li, Yiming Chen, and Peidong Liu. 2024. Dreammesh4d: Video-to-4d generation with sparse-controlled gaussian-mesh hybrid representation. *Advances in Neural Information Processing Systems* 37 (2024), 21377–21400.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 300–309.

Jiayi Liu, Denys Iliash, Angel X Chang, Manolis Savva, and Ali Mahdavi-Amiri. 2024a. SINGAPO: Single Image Controlled Generation of Articulated Parts in Objects. *arXiv preprint arXiv:2410.16499* (2024).

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

Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. 2024c. CAGE: controllable articulation generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17880–17889.

Liu Liu, Jianming Du, Hao Wu, Xun Yang, Zhenguang Liu, Richang Hong, and Meng Wang. 2023a. Category-level articulated object 9d pose estimation via reinforcement learning. In *Proceedings of the 31st ACM International Conference on Multimedia*. 728–736.

Liu Liu, Han Xue, Wenqiang Xu, Haoyuan Fu, and Cewu Lu. 2022. Toward real-world category-level articulation pose estimation. *IEEE Transactions on Image Processing* 31 (2022), 1072–1083.

Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. 2024b. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10072–10083.

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. 2023f. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems* 36 (2023), 22226–22246.

Minghua Liu, Chong Zeng, Xinyue Wei, Ruoxi Shi, Linghao Chen, Chao Xu, Mengqi Zhang, Zhaoning Wang, Xiaoshuai Zhang, Isabella Liu, et al. 2024d. Meshformer: High-quality mesh generation with 3d-guided reconstruction model. *arXiv preprint arXiv:2408.10198* (2024).

Qihao Liu, Weichao Qiu, Weiyao Wang, Gregory D Hager, and Alan L Yuille. 2020. Nothing but geometric constraints: A model-free method for articulated object pose estimation. *arXiv preprint arXiv:2012.00088* (2020).

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023e. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9298–9309.

Shaowei Liu, Saurabh Gupta, and Shenlong Wang. 2023b. Building rearticulable models for arbitrary 3d objects from 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21138–21147.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2023c. Syncdreamer: Generating multiview-consistent images from

a single-view image. *arXiv preprint arXiv:2309.03453* (2023).
Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. 2024. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9970–9980.
Rundong Luo, Haoran Geng, Congyue Deng, Puhao Li, Zan Wang, Baoxiong Jia, Leonidas Guibas, and Siyuan Huang. 2024. Physpart: Physically plausible part completion for interactable objects. *arXiv preprint arXiv:2408.13724* (2024).
Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. 2024. Real2code: Reconstruct articulated objects via code generation. *arXiv preprint arXiv:2406.08474* (2024).
Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. 2021. A-sdf: Learning disentangled signed distance functions for articulated shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13001–13011.
Thomas Müller. 2021. *tiny-cuda-nn*. https://github.com/NVlabs/tiny-cuda-nn
Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).
Shengyi Qian, Linyi Jin, Chris Rockwell, Siyi Chen, and David F Fouhey. 2022. Understanding 3d object articulation in internet videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1599–1609.
Xiaowen Qiu, Jincheng Yang, Yian Wang, Zhehuan Chen, Yufei Wang, Tsun-Hsuan Wang, Zhou Xian, and Chuang Gan. 2025. Articulate anymesh: Open-vocabulary 3d articulated objects modeling. *arXiv preprint arXiv:2502.02590* (2025).
Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PmLR, 8748–8763.
Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. 2023. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142* (2023).
Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. 2022. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18603–18613.
Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. 2023a. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110* (2023).
Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. 2023b. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512* (2023).
Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2024. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 19615–19625.
Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. 2023. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280* (2023).
Chaoyue Song, Jiacheng Wei, Chuan Sheng Foo, Guosheng Lin, and Fayao Liu. 2024. Reacto: Reconstructing articulated objects from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5384–5395.
Jiayi Su, Youhe Feng, Zheng Li, Jinhua Song, Yangfan He, Botao Ren, and Botian Xu. 2024. Artformer: Controllable generation of diverse 3d articulated objects. *arXiv preprint arXiv:2412.07237* (2024).
Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. 2021. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8922–8931.
Qi Sun, Zhiyang Guo, Ziyu Wan, Jing Nathan Yan, Shengming Yin, Wengang Zhou, Jing Liao, and Houqiang Li. 2024. Eg4d: Explicit generation of 4d object without score distillation. *arXiv preprint arXiv:2405.18132* (2024).
Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel Xuan Chang. 2023. OPDMulti: Openable Part Detection for Multiple Objects. *arXiv preprint arXiv:2303.14087* (2023).
Archana Swaminathan, Anubhav Gupta, Kamal Gupta, Shishira R Maiya, Vatsal Agarwal, and Abhinav Shrivastava. 2024. Leia: Latent view-invariant embeddings for implicit 3d articulation. In *European Conference on Computer Vision*. Springer, 210–227.
Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. 2024. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*. Springer, 1–18.
Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. 2024. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151* (2024).
Wei-Cheng Tseng, Hung-Ju Liao, Lin Yen-Chen, and Min Sun. 2022. Cla-nerf: Category-level articulated neural radiance field. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 8454–8460.
Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. 2023a. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12619–12629.
Haowen Wang, Zhen Zhao, Zhao Jin, Zhengping Che, Liang Qiao, Yakun Huang, Zhipeng Fan, Xiuquan Qiao, and Jian Tang. 2024b. SM 3: Self-supervised Multi-task Modeling with Multi-view 2D Images for Articulated Objects. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 12492–12498.
Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. 2018. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*. 52–67.
Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. 2019. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8876–8884.
Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023b. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems* 36 (2023), 8406–8441.
Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. 2024a. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *European Conference on Computer Vision*. Springer, 57–74.
Fangyin Wei, Rohan Chabra, Lingni Ma, Christoph Lassner, Michael Zollhöfer, Szymon Rusinkiewicz, Chris Sweeney, Richard Newcombe, and Mira Slavcheva. 2022. Self-supervised neural articulated shape and appearance models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15816–15826.
Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. 2024. MeshLRM: Large Reconstruction Model for High-Quality Meshes. *arXiv preprint arXiv:2404.12385* (2024).
Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J Guibas. 2021. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13209–13218.
Yijia Weng, Bowen Wen, Jonathan Tremblay, Valts Blukis, Dieter Fox, Leonidas Guibas, and Stan Birchfield. 2024. Neural implicit representation for building digital twins of unknown articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3141–3150.
Di Wu, Liu Liu, Zhou Linli, Anran Huang, Liangtu Song, Qiaojun Yu, Qi Wu, and Cewu Lu. 2025. REArtGS: Reconstructing and Generating Articulated Objects via 3D Gaussian Splatting with Geometric and Motion Constraints. *arXiv preprint arXiv:2503.06677* (2025).
Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. 2024b. Unique3d: High-quality and efficient 3d mesh generation from a single image. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. 2024a. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *arXiv preprint arXiv:2405.14832* (2024).
Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. 2022. Dˆ 2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. *Advances in neural information processing systems* 35 (2022), 32653–32666.
Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. 2020. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11097–11107.
Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506* (2024).
Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191* (2024).
Xianghao Xu, Yifan Ruan, Srinath Sridhar, and Daniel Ritchie. 2022. Unsupervised kinematic motion detection for part-segmented 3d shape collections. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9.
Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen, Oliver Van Kaick, Hao Zhang, and Hui Huang. 2020. RPM-Net: recurrent prediction of motion and parts from point cloud. *arXiv preprint arXiv:2006.14865* (2020).
Zihao Yan, Fubao Su, Mingyang Wang, Ruizhen Hu, Hao Zhang, and Hui Huang. 2023. Interaction-Driven Active 3D Reconstruction with Object Interiors. *arXiv preprint arXiv:2310.14700* (2023).

Hongliang Zeng, Ping Zhang, Chengjiong Wu, Jiahua Wang, Tingyu Ye, and Fang Li. 2024. MARS: multimodal active robotic sensing for articulated characterization. *arXiv preprint arXiv:2407.01191* (2024).

Ge Zhang, Or Litany, Srinath Sridhar, and Leonidas Guibas. 2021. Strobenet: Category-level multiview reconstruction of articulated objects. *arXiv preprint arXiv:2105.08016* (2021).

Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. 2024a. 4diffusion: Multi-view video diffusion model for 4d generation. *Advances in Neural Information Processing Systems* 37 (2024), 15272–15295.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024b. CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

Yuyang Zhao, Zhiwen Yan, Enze Xie, Lanqing Hong, Zhenguo Li, and Gim Hee Lee. 2023. Animate124: Animating one image to 4d dynamic scene. *arXiv preprint arXiv:2311.14603* (2023).

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. 2025. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202* (2025).

Junzhe Zhu and Peiye Zhuang. 2023. HiFA: High-fidelity Text-to-3D Generation with Advanced Diffusion Guidance. arXiv:2305.18766 [cs.CV]

Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. 2024. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10324–10335.