PartCrafter: Structured 3D Mesh Generation via Compositional Latent Diffusion Transformers

Yuchen Lin[1,3*], **Chenguo Lin**[1*], **Panwang Pan**[2†],
Honglei Yan[2], **Yiqiang Feng**[2], **Yadong Mu**[1], **Katerina Fragkiadaki**[3]
[*] Equal contribution [†] Project lead
[1]Peking University, [2]ByteDance, [3]Carnegie Mellon University
https://wgsxm.github.io/projects/partcrafter

Figure 1: **PARTCRAFTER** is a structured 3D generative model that **jointly generates multiple parts and objects from a single RGB image in one shot**, without the need for segmented image inputs.

Abstract

We introduce PARTCRAFTER, the first structured 3D generative model that jointly synthesizes multiple semantically meaningful and geometrically distinct 3D meshes from a single RGB image. Unlike existing methods that either produce monolithic 3D shapes or follow two-stage pipelines, *i.e.* first segmenting an image and then reconstructing each segment, PARTCRAFTER adopts a unified, compositional generation architecture that does not rely on pre-segmented inputs. Conditioned on a single image, it simultaneously denoises multiple 3D parts, enabling end-to-end part-aware generation of both individual objects and complex multi-object scenes. PARTCRAFTER builds upon a pretrained 3D mesh diffusion transformer (DiT) trained on whole objects, inheriting the pretrained weights, encoder, and decoder, and introduces two key innovations: (1) **A compositional latent space**, where each 3D part is represented by a set of disentangled latent tokens; (2) **A hierarchical attention mechanism** that enables structured information flow both within individual parts and across all parts, ensuring global coherence while preserving part-level detail during generation. To support part-level supervision, we curate a new dataset by mining part-level annotations from large-scale 3D object datasets. Experiments show that PARTCRAFTER outperforms existing approaches in generating decomposable 3D meshes, including parts that are not directly visible in input images, demonstrating the strength of part-aware generative priors for 3D understanding and synthesis. Code and training data are released.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

A central organizing principle in perception is the role of objects and parts—semantically coherent units that serve as compositional building blocks for higher-level cognitive tasks, including language, planning, and reasoning. This part-based structure facilitates generalization, as parts can be independently interpreted, recombined, and reused across different contexts. In contrast, most contemporary neural networks lack the ability to form and manipulate such structured, symbol-like entities.

In 3D generation, diffusion-based generative models have shown strong capabilities in synthesizing entire 3D object meshes from scratch or from images [1, 2, 3]. However, these models typically operate at the whole-object level and do not support part-level decomposition. This limitation restricts their applicability in downstream tasks such as texture mapping, animation, physical simulation, and scene editing. Some recent efforts have attempted to address this by first segmenting images into semantic parts and then reconstructing each part in 3D [4, 5, 6, 7]. However, this two-stage pipeline suffers from errors in segmentation, extensive computational costs for extra segmentation models, and difficulties in scaling up, thereby limiting both robustness and fidelity.

We introduce PARTCRAFTER, a structured generative model for 3D scenes that enables part-level generation from a single RGB image through a compositional latent space. PARTCRAFTER jointly generates multiple distinct 3D parts by binding each to a dedicated set of latent variables. This disentanglement allows parts to be independently edited, removed, or added without disrupting the rest of the scene. PARTCRAFTER builds upon large-scale pretrained 3D object latent diffusion models, which represent object meshes as sets of latent tokens [8] aligned either explicitly [3] or implicitly [1, 2] to regions in 3D space. PARTCRAFTER restructures these pretrained models into a compositional architecture equipped with a varying number of latent token sets. It guides each latent token set during the denoising process to associate with a particular 3D part entity, through a novel local-global attention mechanism that facilitates both intra-part and inter-part information flow, in an identity-aware and permutation-invariant way. A shared decoder then maps each latent set into a coherent 3D mesh. We initialize the model's encoder, decoder, and denoising transformer using weights pretrained on whole-object mesh generation tasks [1]. When conditioned on a single RGB image, the model produces structured 3D outputs with coherent part-level decomposition, eliminating the need for brittle segmentation-then-reconstruction pipelines.

To support training, we construct a large-scale dataset by mining part-level annotations from existing 3D object repositories. Many assets in datasets like Objaverse [9] contain part information in their GLTF metadata, as they are often authored using modular components. Rather than flattening these into single meshes, as done in prior work, we retain their part annotations. Our curated dataset merges Objaverse [9], ShapeNet [10], and the Amazon Berkeley Objects (ABO) dataset [11], resulting in a rich collection of part-annotated 3D models suitable for learning compositional generation. As for scene-level generation, we leverage the existing 3D scene dataset 3D-Front [12] for training.

We evaluate PARTCRAFTER on both 3D part-level object generation and 3D scene reconstruction, and compare it against existing two-stage methods [6, 7] that first segment the input image and then reconstruct each segment. Our results show that PARTCRAFTER achieves higher generation quality and better efficiency. As shown in our experimental section, PARTCRAFTER can automatically infer invisible 3D structures from a single image. It can be equally well used at the object or scene level, which makes it a universal model for 3D scene reconstruction. Notably, PARTCRAFTER surpasses its underlying 3D object generative model on object reconstruction fidelity, showing that understanding the compositional structure of objects enhances the quality of 3D generation.

In summary, our contributions are as follows:

- We propose PARTCRAFTER, a structured 3D generation model with explicit part-level binding, capable of generating semantically meaningful 3D components to compose an object or a scene from image prompts, without any segmentation input.
- We design a novel compositional DiT architecture with structured latent spaces, identity-aware part-global attention, and shared decoders.
- A new dataset with part annotations is curated from existing 3D assets.
- PARTCRAFTER achieves state-of-the-art performance on structured 3D generation tasks, demonstrating strong results on both individual objects and complex multi-object scenes. Comprehensive ablation studies validate the contribution of each component in our approach.

2 Related Work

3D Object Generation Previous works on 3D object generation have adopted various representations, including voxels [13, 14, 15], point clouds [16, 17, 18], signed distance fields (SDFs) [19, 20], neural radiance fields (NeRFs) [21, 22, 23, 24, 25], and 3D Gaussian splitting (3DGS) [26, 27, 28]. We focus on 3D meshes for their compatibility with real-world 3D content creation pipelines. One line of mesh generation works produces explicit mesh structures autoregressively [29, 30, 31, 32, 33, 34]. The other line of 3D mesh generation methods uses latent diffusion models [3, 8, 35, 36, 37], and has demonstrated strong performance, particularly when training high-capacity diffusion models across large-scale datasets [1, 2, 38, 39, 40]. PARTCRAFTER builds upon the latter line of 3D mesh generative models. However, these methods typically generate 3D objects as holistic entities, neglecting the natural part-based decomposition that exists in 3D object datasets and is commonly employed by human artists during creation. PARTCRAFTER addresses this limitation.

3D Part-level Object Generation 3D part-level object generation is a long-standing problem in computer vision. One line of work [41, 42] has focused on assembling existing parts by inferring the right translations and orientations. Another line of work, which is more related to our work, generates 3D geometry of parts and their structure to form a complete object. Different primitives have been used as representation, such as point clouds [43], voxels [44], implicit neural fields [45, 46, 47, 48], and other parametric representations [49, 50, 51]. These works are limited by either the scale of datasets [10, 52] or the expressiveness of the representation. To improve the visual quality and utilize prior knowledge of 2D pretrained models, several recent works [4, 5] propose to leverage multi-view diffusion and 2D segmentation models [53] to generate NeRF [21] or NeuS [54] with part labels. Concurrent work HoloPart [6] proposes to first segment the 3D object mesh into parts and then refine the parts' geometry with a pretrained 3D DiT [1]. Although effective, these methods are heavily dependent on the segmentation quality and are difficult to scale up. In contrast, our method can generate structured 3D part-level objects in one shot without any 2D or 3D segmentation.

3D Scene Generation Prior work on 3D object-composed scene generation typically extracts abstract representations such as layouts [55, 56], graphs [57, 58], or segmentations [59, 60, 61, 62, 63, 64] to model object relationships, and then generates [65, 66, 67] or retrieves [68, 68] objects based on these representations and input conditions. The closest work to ours is MIDI [7], which generates compositional 3D scenes by segmenting the input image [69] and prompting object-level 3D diffusion models with region segments. However, MIDI requires all components to be visible in the image, hindering its ability to segment and reconstruct occluded parts, which is a shared limitation across all two-stage segment-and-reconstruct methods. PARTCRAFTER is inspired by MIDI but overcomes this limitation. Without any additional segmentation model, PARTCRAFTER is capable of generating parts even when they are not visible in the conditioning image prompt. PARTCRAFTER demonstrates that explicit segmentation is not a necessary prerequisite for structured 3D scene generation.

3 PARTCRAFTER: A Compositional 3D Diffusion Model

The architecture of PARTCRAFTER is illustrated in Figure 2. It is a compositional generative model that generates structured 3D assets by simultaneously denoising several part-specified latent token sets. Specifically, given a prompt RGB image $\mathbf{c}$ and a user-specified number of parts N, PARTCRAFTER generates a structured 3D asset $\mathcal{O} \coloneqq \{\mathbf{p}_i\}_{i=1}^{N}$, *e.g.,* a part-decomposable object or a 3D scene composed of object parts, where each part $\mathbf{p}_i$ is a separate, semantically meaningful component of the asset. 3D meshes are utilized as the output representation. At inference time, all part meshes $\mathbf{p}_i \coloneqq \{\mathcal{V}_i, \mathcal{F}_i\}$ are generated and decoded simultaneously, where $\mathcal{V}_i \in \mathbb{R}^{V_i \times 3}$ is the vertex set and $\mathcal{F}_i \in \mathbb{N}^{F_i \times 3}$ is the face set of the i-th part. Notably, coordinates of each part's vertices are in the same global canonical space $[-1, 1]^3$, allowing for easy assembly of the parts into a complete object or scene, without any need for additional transformations.

PARTCRAFTER builds upon a pretrained 3D object mesh generation model [1] by utilizing and reassembling its encoders, decoders, and DiT blocks into a compositional multi-entity generation architecture. We provide a description of object-level 3D generation models in Section 3.1, followed by a detailed presentation of our proposed structured multi-entity model in Section 3.2.

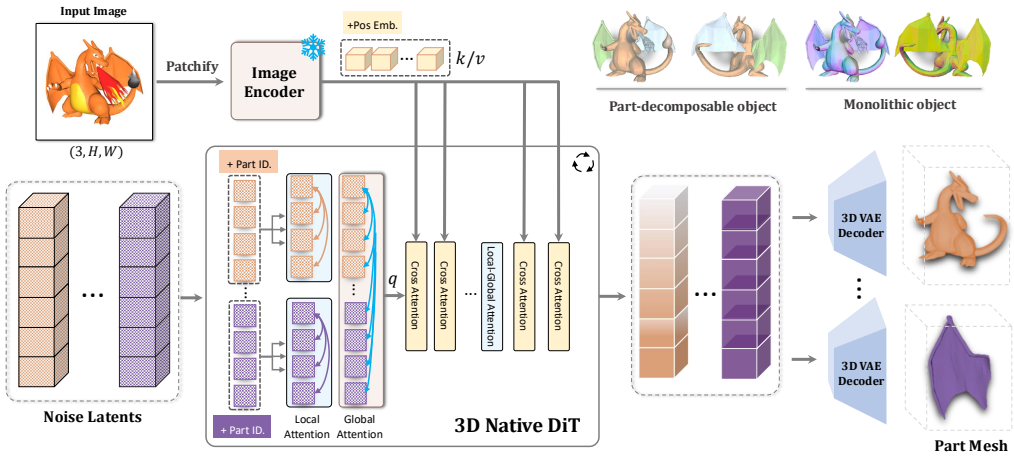

Figure 2: **Architecture of PARTCRAFTER.** Our model utilizes local-global attention to capture both part-level and global features. Part ID embeddings and incorporation of the image condition into both local and global features ensure the independence and semantic coherence of the generated parts.

3.1 Preliminary: Diffusion Transformer for 3D Object Mesh Generation

Pretrained 3D object generation models have shown impressive performance in generating high-quality 3D objects. Our model specifically builds on top of TripoSG [1], a state-of-the-art 3D image-to-mesh generative model. TripoSG first encodes 3D shapes into a set of latent vectors with a transformer-based Variational Autoencoder (VAE) using 3DShape2VecSet [8]. The decoder of the VAE uses a Signed Distance Function (SDF) representation, which enables sharper geometry and avoids aliasing. A rectified flow model is trained on the VAE latents to generate new 3D shapes from noise. The model is conditioned on a single input image using DINOv2 [70] features, injected via cross-attention in every transformer block. It is trained on 2 million high-quality 3D shapes curated from Objaverse [9, 71] and ShapeNet [10] through a rigorous four-stage pipeline involving quality scoring, filtering out noisy data, mesh repair, and conversion to a watertight SDF-compatible format.

3.2 PARTCRAFTER

The design of PARTCRAFTER is motivated by our observation that foundation 3D latent diffusion models such as TripoSG [1], although trained on object-level training examples, can be applied effectively out-of-the-box to auto-encode 3D parts as well, without normalizing the input 3D mesh to the center of the canonical space (as done for 3D object meshes). We thus adapt the neural components of TripoSG to build a part-decomposable 3D asset reconstruction model.

Compositional Latent Space The key insight of 3D object mesh generative models with a set of latents is that each token in the latent space of the pretrained 3D VAE [1, 2, 8] is implicitly associated with an area of the canonical 3D space. PARTCRAFTER expands the latent space of monolithic object generative models with multiple sets of latents, each taking care of a separate 3D part. Specifically, an object or a scene is represented as a set of N parts $\mathcal{O} = \{\mathbf{p}_i\}_{i=1}^{N}$ and each part $\mathbf{p}_i$ is represented by a set of latent tokens $\boldsymbol{z}_i = \{z_{ij}\}_{j=1}^{K} \in \mathbb{R}^{K \times C}$, where K is the number of tokens in the component's set and C is the dimension of each token. To distinguish across parts, we add a learnable part identity embedding $\mathbf{e}_i \in \mathbb{R}^C$ to the tokens of each part $\mathbf{p}_i$. These embeddings are randomly initialized and optimized during training. To support the inherent permutation invariance of the parts, we shuffle the order of the parts in training. Since $\boldsymbol{z}_i$ represents the features of the i-th part, we construct the global 3D asset tokens simply by concatenating the tokens of all parts, i.e., $\boldsymbol{\mathcal{Z}} = \{\boldsymbol{z}_i\}_{i=1}^{N} \in \mathbb{R}^{NK \times C}$.

Local-Global Denoising Transformer We fuse information across sets of latent tokens to enable both local-level and global-level reasoning. We apply **local attention** independently to the tokens $\boldsymbol{z}_i$ of each part. This captures localized features within each part, ensuring that their internal structure remains distinct. After capturing part-level features, we apply **global attention** over the entire set of

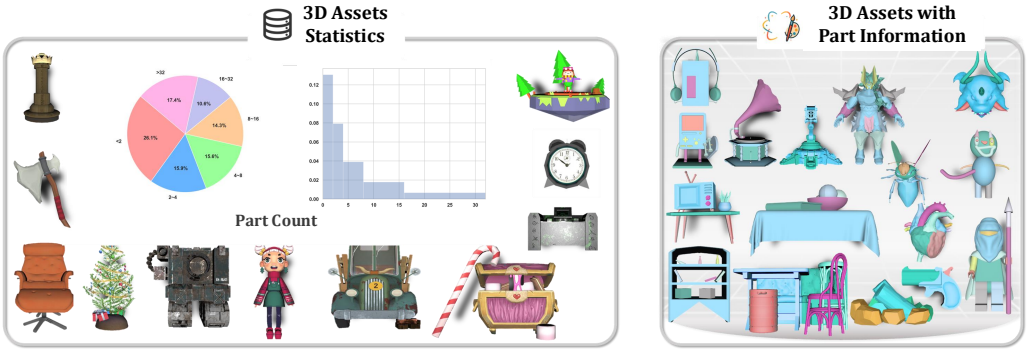

Figure 3: **Dataset Overview**. Large-scale 3D object datasets [9, 10, 11] often contain rich part annotations. The pie chart and bin chart visualize the distribution of an object's part count.

tokens $\boldsymbol{\mathcal{Z}}$ to model global interactions across parts. The attention operations are defined as:

$$\mathbf{A}_i^{\text{local}} = \text{softmax}\left(\frac{\boldsymbol{z}_i \boldsymbol{z}_i^T}{\sqrt{C}}\right) \in \mathbb{R}^{K\times K}, \quad \mathbf{A}^{\text{global}} = \text{softmax}\left(\frac{\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{Z}}^T}{\sqrt{C}}\right) \in \mathbb{R}^{NK\times NK}, \tag{1}$$

where $\mathbf{A}$ denotes the resulting attention maps. This hierarchical attention structure captures both fine-grained part details and global 3D context, enabling effective information exchange between local and holistic representations. We adopt the DiT architecture with long skip connections [35, 72, 73], as in TripoSG [1], and replace its original attention module with our part-global attention mechanism.

We inject DINOv2 [70] features of the condition image $\mathbf{c}$ into both levels of attention. Specifically, we use cross-attention within both the local and global attention. This dual-conditioning design enables the model to align the overall part-based composition with the input image while ensuring that each part remains semantically meaningful. Our design choices are validated in Section 4.3.

Training Objective We train PARTCRAFTER by rectified flow matching [74, 75, 76, 77], which maps the noisy Gaussian distribution to the data distribution in a linear trajectory. Specifically, given an object latent $\boldsymbol{\mathcal{Z}}_0 = \{\boldsymbol{z}_i\}_{i=1}^N$ and a condition image $\mathbf{c}$, we perturb the latents by adding Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ at a noise level t, yielding a noisy latent representation $\boldsymbol{\mathcal{Z}}_t = t\boldsymbol{\mathcal{Z}}_0 + (1-t)\,\boldsymbol{\epsilon}$. The model is then trained to predict the velocity term $\boldsymbol{\epsilon} - \boldsymbol{\mathcal{Z}}_0$ given the noisy latent $\boldsymbol{\mathcal{Z}}_t$ at the noise level t, conditioned on the condition image $\mathbf{c}$, by minimizing the following objective:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{\boldsymbol{\mathcal{Z}},\boldsymbol{\epsilon},t}\left[\left\|(\boldsymbol{\epsilon} - \boldsymbol{\mathcal{Z}}_0) - \mathbf{v}_\theta\left(\boldsymbol{\mathcal{Z}}_t, t, \mathbf{c}\right)\right\|^2\right], \tag{2}$$

where $\mathbf{v}_\theta$ is the velocity prediction. Importantly, the noise level t is shared across all parts of the 3D scene or object to ensure consistent trajectory sampling.

3.3 Dataset Curation

3D object datasets such as Objaverse [9] and Objaverse-XL [71] have made millions of 3D assets available to the research community. While previous works [1, 3, 38] have primarily focused on directly utilizing these datasets [9, 10, 11, 52, 71, 78] for 3D whole-object generation, we observe that the rich part-level metadata included in many of these 3D models offers an opportunity to enable more structured 3D generation. As shown in Figure 3, over half of the objects in a subset [79] of Objaverse [9] contain explicit part annotations. These part labels often originate from artists' workflows, where objects are intentionally decomposed into semantically meaningful components to facilitate modular design, such as modeling a pair of scissors using two distinct blades. To train our model, we curate a dataset by combining multiple sources [9, 10, 11, 79], yielding 130,000 3D objects, of which 100,000 contain multiple parts. We further refine this dataset by filtering based on texture quality, part count, and average part-level Intersection over Union (IoU) to ensure high-quality supervision. The resulting dataset comprises approximately 50,000 part-labeled objects and 300,000 individual parts. For 3D scenes, we utilize the existing 3D object-composed scene dataset 3D-Front [12]. Additional dataset statistics and filtering criteria are detailed in Appendix A.

3.4 Implementation Details

We modify the 21 DiT blocks in TripoSG [1] by alternating their original attention processors with our proposed local-global attention mechanism. Specifically, global-level attention is applied to DiT blocks with even indices, while local-level attention is used in the odd-indexed blocks, following the long-cut strategy. We validate this architectural design in Section 4.3. PARTCRAFTER is trained on 8 H20 GPUs with a batch size of 256 by fully finetuning the pretrained TripoSG [1]. We first train a base model for up to 8 parts on our curated part-level object dataset at a learning rate of 1e-4 for 5K iterations. For part-decomposable objects, we then finetune the base model to support up to 16 parts. For object-composed scenes, we further adapt the base model to the 3D-Front [12] dataset for up to 8 objects. Both finetuning processes last for 5K iterations at a reduced learning rate of 5e-5. We include 30% monolithic objects in training for regularization. This curriculum training strategy avoids loss spikes and catastrophic forgetting during the training process. The whole training process takes about 2 days. We use 512 tokens for each part, which we find is sufficient to represent part geometry and semantics. We evaluate PARTCRAFTER on a test set of about 2K data samples.

4 Experiments

Our experiments aim to answer the following questions: **(1)** How does PARTCRAFTER perform in part-level reconstruction of objects and scenes compared to existing state-of-the-art models that first segment and then reconstruct parts at the object and scene level? **(2)** Can PARTCRAFTER reconstruct parts that are not visible in the image prompt? **(3)** How do results vary with different numbers of parts? **(4)** What are the contributions of design choices in our local-global denoising transformer?

Baselines To the best of our knowledge, PARTCRAFTER is the first work to generate 3D part-level object meshes from a single image. Recent works Part123 [4] and PartGen [5] reconstruct 3D neural fields [21, 54] from images, which are not directly comparable to our work that focuses on meshes. We consider the following baselines: **(1)** HoloPart [6] on object level, which is a concurrent work that first segments a given 3D object mesh and then completes the coarse-segmented parts into fine-grained meshes. We adapt HoloPart to our setting by utilizing TripoSG [1] to generate a mesh from an image, and apply HoloPart to get the part meshes. For fair comparison, we align the number of tokens in TripoSG with our method, that is, $N \times 512$ tokens for N parts. **(2)** MIDI [7] on scene level, which reconstructs multi-instance 3D scenes using object segmentation prompts. We provide **ground truth segmentation masks** for MIDI, while PARTCRAFTER does not need any segmentation.

Evaluation Metrics We measure the generation result of structured 3D assets in both the global (object or scene) and part level. **(1)** Fidelity of the generated mesh. Since we do not have the correspondence between the generated and ground truth parts, we simply concatenate the parts to form a single mesh. We evaluate the fidelity of generated 3D meshes by L2 Chamfer Distance (CD) and F-Score with a threshold of 0.1. Lower Chamfer Distance and higher F-Score indicate higher similarity between the generated and ground truth meshes. **(2)** Geometry Independence of Generated Part Meshes. We use the Average Intersection over Union (IoU) to evaluate the geometry independence of generated part meshes. We compute the average IoU between each generated part by voxelizing the canonical space into $64 \times 64 \times 64$ grids. Lower IoU indicates less overlap between generated parts, thus demonstrating better part independence. The optimal metrics are reached when generated parts are non-intersecting and can be composed into an object similar to the ground truth. We report the average generation time of objects or scenes with 4 parts in the test set on an H20 GPU.

4.1 3D Part-level Object Generation

We evaluate the performance of PARTCRAFTER on the 3D part-level object generation task. As shown in Table 1, PARTCRAFTER outperforms HoloPart by a large margin in both object-level and part-level metrics. Given an image, PARTCRAFTER is able to generate a 3D part-decomposable mesh with high fidelity and geometry independence in seconds. HoloPart requires substantially more time to segment the object mesh, and its segmentation process suffers due to the lower geometry quality of the generated mesh compared to real artistic meshes, which hinders its performance. Notably, PARTCRAFTER surpasses our backbone model TripoSG [1] in object-level metrics, even when we align the number of tokens in TripoSG with our method. This suggests that our local-global attention mechanism enables better object-level representation learning, thereby **enhancing**

Table 1: **Evaluation on 3D Part-level Object Generation**. We report evaluation result on Objaverse [9], ShapeNet [10], and ABO [11]. We denote TripoSG* [1] as the number-of-token-aligned backbone model. Higher F-Score and lower CD, IoU indicate better results. Best results are bolded.

3D Part-level Generation	Objaverse [9]			ShapeNet [10]			ABO [11]			↓ Time
	↓ CD	↑ F-Score	↓ IoU	↓ CD	↑ F-Score	↓ IoU	↓ CD	↑ F-Score	↓ IoU	
Dataset	/	/	0.0796	/	/	0.1827	/	/	0.0137	/
TripoSG [1]	0.3104	0.5940	/	0.3751	0.5050	/	0.2017	0.7096	/	/
TripoSG* [1]	0.1821	0.7115	/	0.3301	0.5589	/	0.1503	0.7723	/	
HoloPart [6]	0.1916	0.6916	0.0443	0.3511	0.5498	0.1107	0.1338	0.8093	0.0449	18min
Ours	**0.1726**	**0.7472**	**0.0359**	**0.3205**	**0.5668**	**0.0293**	**0.1047**	**0.8617**	**0.0243**	**34s**

Table 2: **Evaluation on 3D Object-Composed Scene Generation**. We report evaluation results on 3D-Front [12], as well as on a challenging subset of 3D-Front [12] characterized by severe occlusions.

3D Scene Generation	3D-Front [12]			3D-Front (Occluded) [12]			↓ Run Time
	↓ CD	↑ F-Score	↓ IoU	↓ CD	↑ F-Score	↓ IoU	
MIDI [7]	0.1602	0.7931	**0.0013**	0.2591	0.6618	**0.0020**	80s
Ours	**0.1491**	**0.8148**	0.0034	**0.1508**	**0.7800**	0.0035	**34s**

the generation process through improved structural understanding. We also observe that PARTCRAFTER performs worse on ShapeNet [10] than on Objaverse [9] and ABO [11], primarily due to the backbone's degraded performance on ShapeNet. We present qualitative results in Figure 4 to show that PARTCRAFTER **can infer parts invisible in the conditioning image** and generate 3D part-level objects. We further provide **3D object texture generation** results in Appendix D.

4.2 3D Object-Composed Scene Generation

We conduct experiments on 3D scene generation using the 3D-Front [12] dataset. As shown in Table 2, PARTCRAFTER outperforms MIDI [7] in reconstruction fidelity metrics. While MIDI [7] uses **ground truth segmentation masks** for evaluation, PARTCRAFTER does not require any segmentation. To further validate our method, we select a subset of 3D scenes in 3D-Front [12] with severe occlusion, where the ground truth segmentation masks cannot segment all objects. We observe that the performance of MIDI [7] degrades significantly in these cases, while PARTCRAFTER still maintains a high level of generation quality. MIDI [7] slightly surpasses PARTCRAFTER in IoU metrics, primarily due to the fact that ground truth 2D segmentation masks are used in their pipeline. Qualitative results are provided in Figure 5 to demonstrate that PARTCRAFTER can **recover complex 3D structures** and generate high-quality meshes for 3D scenes, all from a single image.

For additional qualitative results of our method, please check Appendix E and the supplementary file.

4.3 Ablations

We present quantitative results in Table 3 to validate the effectiveness of each component in our method. All experimental settings are consistent with our base model, using a maximum of 8 parts and 512 tokens per part, trained for 5K iterations with a learning rate of 1e-4 and a batch size of 256.

Necessity of Local-Global Attention To assess the necessity of local-global attention, we conduct ablation experiments by removing either the local attention or the global attention. We observe that the model completely collapses without local attention and fails to generate any meaningful 3D mesh (Exp. 2). In contrast, removing global attention still allows the model to produce 3D meshes, but the outputs lack part decomposition, exhibit significant geometry overlap, and result in notably higher IoU values (Exp. 3). These findings demonstrate that local-global attention is essential for learning meaningful 3D representations and capturing the 3D structural relationships globally.

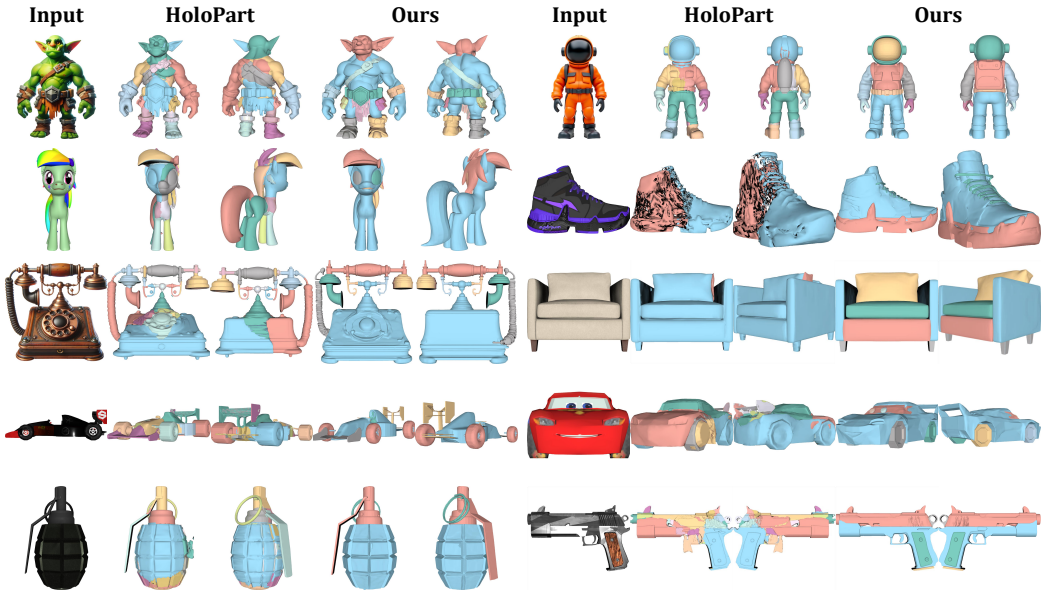

Figure 4: **Qualitative Results on 3D Part-Level Object Generation.** We present visualization results of HoloPart [6] and our method. Different colors indicate different parts of generated objects.

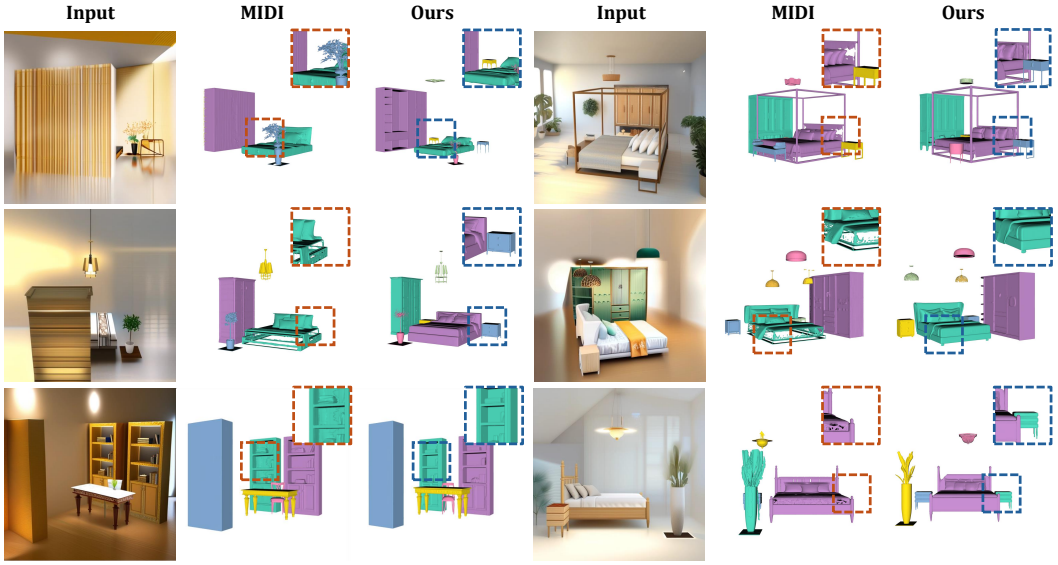

Figure 5: **Qualitative Results on 3D Scene Generation.** We present visualization results of MIDI [7] and our method. Different colors indicate different objects in the generated 3D scene.

Necessity of Part Identity Control We remove part identity embeddings from attentions, and the model collapses due to ambiguity in distinguishing between different parts (Exp. 1). We further explore how to incorporate control signals into the local-global attention (Exp. 4). Enabling cross-attention in the local attention improves mesh fidelity but reduces geometry independence, while enabling it in the global attention enhances geometry independence at the cost of fidelity. Based on these observations, we adopt cross-attention in both local and global attention modules to strike a balance between fidelity and independence, achieving the best overall performance.

Order of Local-Global Attention Since our DiT follows a U-Net-like architecture with long skip connections, we investigate how the ordering of local-global attention affects performance (Exp. 5). Across the 21 DiT blocks, we approximately balance the number of local and global attention modules. We explore three different configurations for placing global-level attention: **(1)** in the middle, **(2)** at the beginning and end, and **(3)** in an alternating pattern. Among these, the alternating configuration

Table 3: **Ablation Study for PARTCRAFTER**. We report evaluation results on Objaverse [9] dataset.

Exp-ID	Part Emb	Self Local-Attn	Self Global-Attn	Cross Local-Attn	Cross Global-Attn	Global-Attn Order	↓ CD	↑ F-Score	↓ IoU
#1	✗	✓	✓	✓	✓	Middle	0.3143	0.4978	0.1401
	✓	✓	✓	✓	✓	Middle	0.1711	0.7374	0.0814
#2	✓	✗	✓	✗	✓	Middle	0.4632	0.2327	0.0541
	✓	✓	✓	✓	✓	Middle	0.1711	0.7374	0.0814
#3	✓	✓	✗	✓	✗	Middle	0.2606	0.5978	0.1602
	✓	✓	✓	✓	✓	Middle	0.1711	0.7374	0.0814
#4	✓	✓	✓	✓	✗	Middle	0.1744	0.7334	0.0869
	✓	✓	✓	✗	✓	Middle	0.1847	0.7188	0.0824
	✓	✓	✓	✓	✓	Middle	0.1711	0.7374	0.0814
#5	✓	✓	✓	✓	✓	Middle	0.1711	0.7374	0.0814
	✓	✓	✓	✓	✓	Sides	0.1901	0.7114	0.0336
	✓	✓	✓	✓	✓	Alternating	0.1781	0.7212	0.0518
Ours	✓	✓	✓	✓	✓	Alternating	0.1781	0.7212	0.0518

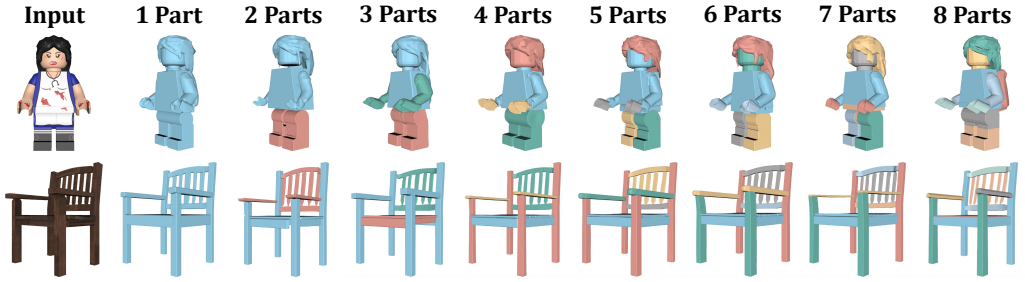

Figure 6: **Qualitative Results on Different Number of Parts.** We show that PARTCRAFTER can generate reasonable results with different granularities of part decomposition given the same image.

yields the best trade-off between local-level and global-level metrics, suggesting that this arrangement enables more effective information exchange and integration across both levels.

Specified Number of Parts We present qualitative results in Figure 6 to show that PARTCRAFTER can generate reasonable results with a varied number of parts given the same image. It validates that PARTCRAFTER captures 3D features hierarchically and handles different granularities of part decomposition, which can be a useful feature for downstream applications and commercial use.

5 Conclusion

In this work, we propose PARTCRAFTER, a novel 3D native structured generative model. Trained on our curated part-level 3D mesh dataset, PARTCRAFTER reconstructs part-level objects and scenes without relying on any 2D or 3D segmentation information that existing models need. PARTCRAFTER validates the feasibility of integrating 3D structural understanding into the generative process.

Limitations and Future Works PARTCRAFTER is trained on 50K part-level data, which is relatively small compared to that used to train 3D object generation models (typically millions). Future works can consider scaling up DiT training with more data of higher quality.

Broader Impact We conducted a foundational study in 3D computer vision. While our large-scale training may have environmental implications, we believe the benefits of advancing 3D vision justify further exploration. The importance of the problem studied is discussed in the Introduction.

Acknowledgment

This research is supported by a grant from Bytedance PICO (No. CT20240607105793).

References

[1] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.

[2] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 2024.

[3] 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. *arXiv preprint arXiv:2412.01506*, 2024.

[4] Anran Liu, Cheng Lin, Yuan Liu, Xiaoxiao Long, Zhiyang Dou, Hao-Xiang Guo, Ping Luo, and Wenping Wang. Part123: part-aware 3d reconstruction from a single-view image. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

[5] Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models. *arXiv preprint arXiv:2412.18608*, 2024.

[6] Yunhan Yang, Yuan-Chen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yan-Pei Cao, and Xihui Liu. Holopart: Generative 3d part amodal segmentation. *arXiv preprint arXiv:2504.07943*, 2025.

[7] Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. *arXiv preprint arXiv:2412.03558*, 2024.

[8] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 2023.

[9] 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 (CVPR)*, 2023.

[10] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[11] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[12] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[13] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016.

[14] Aditya Sanghi, Rao Fu, Vivian Liu, Karl DD Willis, Hooman Shayani, Amir H Khasahmadi, Srinath Sridhar, and Daniel Ritchie. Clip-sculptor: Zero-shot generation of high-fidelity and diverse shapes from natural language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[15] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[16] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[17] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[18] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[19] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[20] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[21] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

[22] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] Ziang Cao, Fangzhou Hong, Tong Wu, Liang Pan, and Ziwei Liu. Large-vocabulary 3d diffusion model with transformer. In *International Conference on Learning Representations (ICLR)*, 2024.

[24] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *International Conference on Learning Representations (ICLR)*, 2024.

[25] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. In *International Conference on Learning Representations (ICLR)*, 2024.

[26] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: A structured and explicit radiance representation for 3d generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[27] Xianglong He, Junyi Chen, Sida Peng, Di Huang, Yangguang Li, Xiaoshui Huang, Chun Yuan, Wanli Ouyang, and Tong He. Gvgen: Text-to-3d generation with volumetric representation. In *European Conference on Computer Vision (ECCV)*, 2024.

[28] Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation. *arXiv preprint arXiv:2501.16764*, 2025.

[29] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

[31] Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. *arXiv preprint arXiv:2409.18114*, 2024.

[32] Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. Llama-mesh: Unifying 3d mesh generation with language models. *arXiv preprint arXiv:2411.09595*, 2024.

[33] Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. *arXiv preprint arXiv:2503.15265*, 2025.

[34] Xianglong He, Junyi Chen, Di Huang, Zexiang Liu, Xiaoshui Huang, Wanli Ouyang, Chun Yuan, and Yangguang Li. Meshcraft: Exploring efficient and controllable mesh generation with flow-based dits. *arXiv preprint arXiv:2503.23022*, 2025.

[35] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems*, 2023.

[36] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[37] Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. *arXiv preprint arXiv:2412.17808*, 2024.

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

[39] Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. *arXiv preprint arXiv:2503.21732*, 2025.

[40] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*, 2025.

[41] Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, Hao Dong, et al. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems*, 2020.

[42] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[43] George Kiyohiro Nakayama, Mikaela Angelina Uy, Jiahui Huang, Shi-Min Hu, Ke Li, and Leonidas Guibas. Difffacto: Controllable part-based 3d point cloud generation with cross diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[44] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (TOG)*, 2019.

[45] Dmitry Petrov, Matheus Gadelha, Radomír Měch, and Evangelos Kalogerakis. Anise: Assembly-based neural implicit surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[46] Connor Lin, Niloy Mitra, Gordon Wetzstein, Leonidas J Guibas, and Paul Guerrero. Neuform: Adaptive overfitting for neural shape editing. *Advances in Neural Information Processing Systems*, 35:15217–15229, 2022.

[47] Juil Koo, Seungwoo Yoo, Minh Hieu Nguyen, and Minhyuk Sung. Salad: Part-level latent diffusion for 3d shape generation and manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[48] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019.

[49] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7154–7164, 2019.

[50] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[51] Jingwen Ye, Yuze He, Yanning Zhou, Yiqin Zhu, Kaiwen Xiao, Yong-Jin Liu, Wei Yang, and Xiao Han. Primitiveanything: Human-crafted 3d primitive assembly generation with auto-regressive transformer. *arXiv preprint arXiv:2505.04622*, 2025.

[52] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019.

[53] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.

[54] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[55] Chenguo Lin and Yadong Mu. Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. *arXiv preprint arXiv:2402.04717*, 2024.

[56] Chenguo Lin, Yuchen Lin, Panwang Pan, Xuanyang Zhang, and Yadong Mu. Instructlayout: Instruction-driven 2d and 3d layout synthesis with semantic graph prior. *arXiv preprint arXiv:2407.07580*, 2024.

[57] Gege Gao, Weiyang Liu, Anpei Chen, Andreas Geiger, and Bernhard Schölkopf. Graphdreamer: Compositional 3d scene synthesis from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[58] Haotian Bai, Yuanhuiyi Lyu, Lutao Jiang, Sijia Li, Haonan Lu, Xiaodong Lin, and Lin Wang. Componerf: Text-guided multi-object compositional nerf with editable 3d scene layout. *arXiv preprint arXiv:2303.13843*, 2023.

[59] Tao Chu, Pan Zhang, Qiong Liu, and Jiaqi Wang. Buol: A bottom-up framework with occupancy-aware lifting for panoptic 3d scene reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[60] Daoyi Gao, Dávid Rozenberszki, Stefan Leutenegger, and Angela Dai. Diffcad: Weakly-supervised probabilistic cad model retrieval and alignment from an rgb image. *ACM Transactions on Graphics (TOG)*, 2024.

[61] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: Robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[62] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2cad: 3d shape prediction by learning to segment and retrieve. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 2020.

[63] Haonan Han, Rui Yang, Huan Liao, Jiankai Xing, Zunnan Xu, Xiaoming Yu, Junwei Zha, Xiu Li, and Wanhua Li. Reparo: Compositional 3d assets generation with differentiable 3d layout alignment. *arXiv preprint arXiv:2405.18525*, 2024.

[64] Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Zero-shot scene reconstruction from single images with deep prior assembly. *arXiv preprint arXiv:2410.15971*, 2024.

[65] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 2021.

[66] Yongwei Chen, Tengfei Wang, Tong Wu, Xingang Pan, Kui Jia, and Ziwei Liu. Comboverse: Compositional 3d assets creation using spatially-aware diffusion guidance. In *European Conference on Computer Vision*, 2024.

[67] Andreea Dogaru, Mert Özer, and Bernhard Egger. Generalizable 3d scene reconstruction via divide and conquer from a single view. *arXiv preprint arXiv:2404.03421*, 2024.

[68] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[69] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[70] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[71] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[72] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023.

[73] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[74] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

[75] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[76] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[77] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[78] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 2016.

[79] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision (ECCV)*, 2024.

[80] Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024.

[81] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work performed by the authors in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code and data will be made available upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all the training and test details necessary to understand the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not report information about the statistical significance of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release data or models that have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets used in the paper are properly credited and the license and terms of use are explicitly mentioned and properly respected in the supplementary material.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Dataset Details

We collect our part-level dataset from Objaverse [9] (ODC-By v1.0 License), ShapeNet-Core[10] (Custom License), and Amazon Berkeley Objects [11] (CC-BY 4.0 License). We use a high-quality subset of Objaverse provided by LGM [79]. We filtered out objects without textures and selected those with fewer than 16 parts and a maximum IoU of less than 0.1. The resulting dataset comprises approximately 50,000 part-labeled objects and 300,000 individual parts. We adopt an additional 30,000 monolithic objects as regularization. We use the 3D-Front (CC-BY-NC 4.0 License) dataset processed by MIDI [7]. We will release our dataset under a CC-BY 4.0 license.

B Implementation Details

Our model builds on TripoSG [1] (MIT License). As for part-level object generation, we adapt HoloPart [6] (MIT License) into a generative pipeline. Specifically, we first generate a 3D mesh from the input image using TripoSG and then use HoloPart to generate a part-level object. We use the same 3D segmentation model, SAMPart3D [80] (MIT License), as Holopart. As for 3D scene generation, we adopt MIDI [7] (MIT License) as a baseline. We will release our code under an MIT license.

C Real-world Results

We train PARTCRAFTER on synthetic rendered images, which allows us to have a large and diverse dataset. To test the boundary of our model's generalization ability, we test PARTCRAFTER on real-world images from the CO3D [81] dataset. As expected, the performance was lower compared to synthetic images due to the domain gap. To address this, we explored transferring the style of real-world images to make them look more like images rendered from a graphics engine using recent image editing models, such as GPT-4o. Specifically, we use the prompt: *"Preserve all details and perform image-to-image style transfer to convert the image into the style of a 3D rendering (Objaverse-style rendering)."* We find that our model performs significantly better in the style-transferred images. Therefore, we propose a two-stage pipeline for real-world image inference: first, use an image editing model to transfer the style of the input image, and then use PARTCRAFTER to generate the 3D part-level object. As shown in Figure 7, this simple pipeline works surprisingly well.

D Texture Generation

As shown in Figure 8, we further generate textures for the generated 3D part-level objects using an off-the-shelf texture generation model Hunyuan3D-2 [38] (Hunyuan3D-2 License). Although Hunyuan3D-2 is trained on monolithic objects, it performs well on part-level objects. Since we know which part each vertex belongs to, we can manipulate the generated UV map accordingly, assigning textures to their corresponding parts. Therefore, our method is capable of **generating 3D objects composed of multiple distinct parts with respective textures, all from a single image**. Compared to previous methods that generate shape and texture for monolithic objects, our method produces part-aware 3D objects with distinct textures for each component. Because prior approaches treat the object as a single whole, textures often suffer from artifacts such as color bleeding between parts, and the resulting shapes may lack physically plausible part-level structure. In contrast, our method ensures more accurate texture assignment and generates 3D structures that better reflect the compositional nature of real-world objects. Thanks to these features, we believe PARTCRAFTER holds great potential for downstream applications such as real-to-simulation transfer and robotics training, where accurate and part-aware 3D representations with textures are crucial.

E More Results

We provide more generation results on 3D part-level object in Figure 9, 10, 11, and 12 and 3D object-composed scene in Figure 13. For better visualization results, please see the supplementary files and our project website: https://wgsxm.github.io/projects/partcrafter/.

Figure 7: **Qualitative Results on 3D Part-Level Object Generation from Real-World Images.** We use GPT-4o to transfer the style of real-world images to make them look like images rendered from a graphics engine. Then, we use PARTCRAFTER to generate 3D part-level objects.

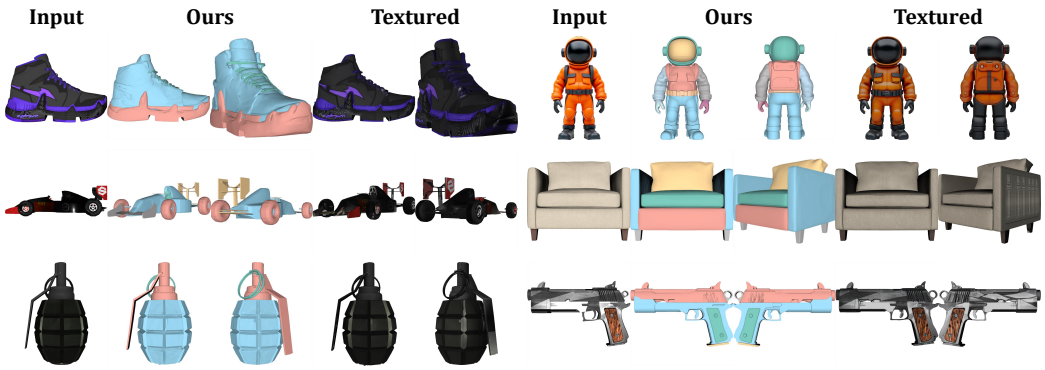

Figure 8: **Qualitative Results on 3D Textured Part-Level Object Generation.**

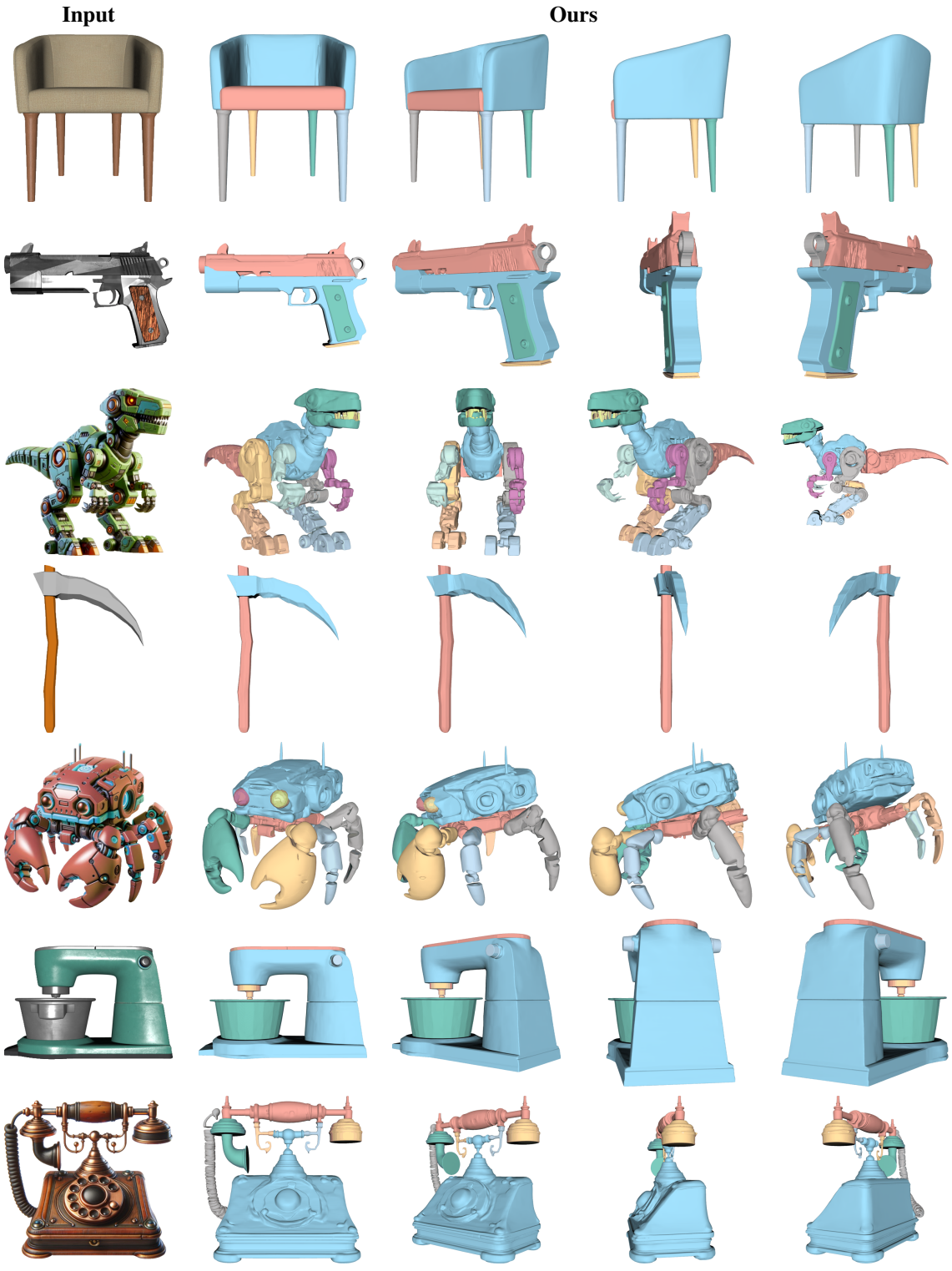

Figure 9: More results of image-conditioned 3D part-level object generation of PARTCRAFTER.

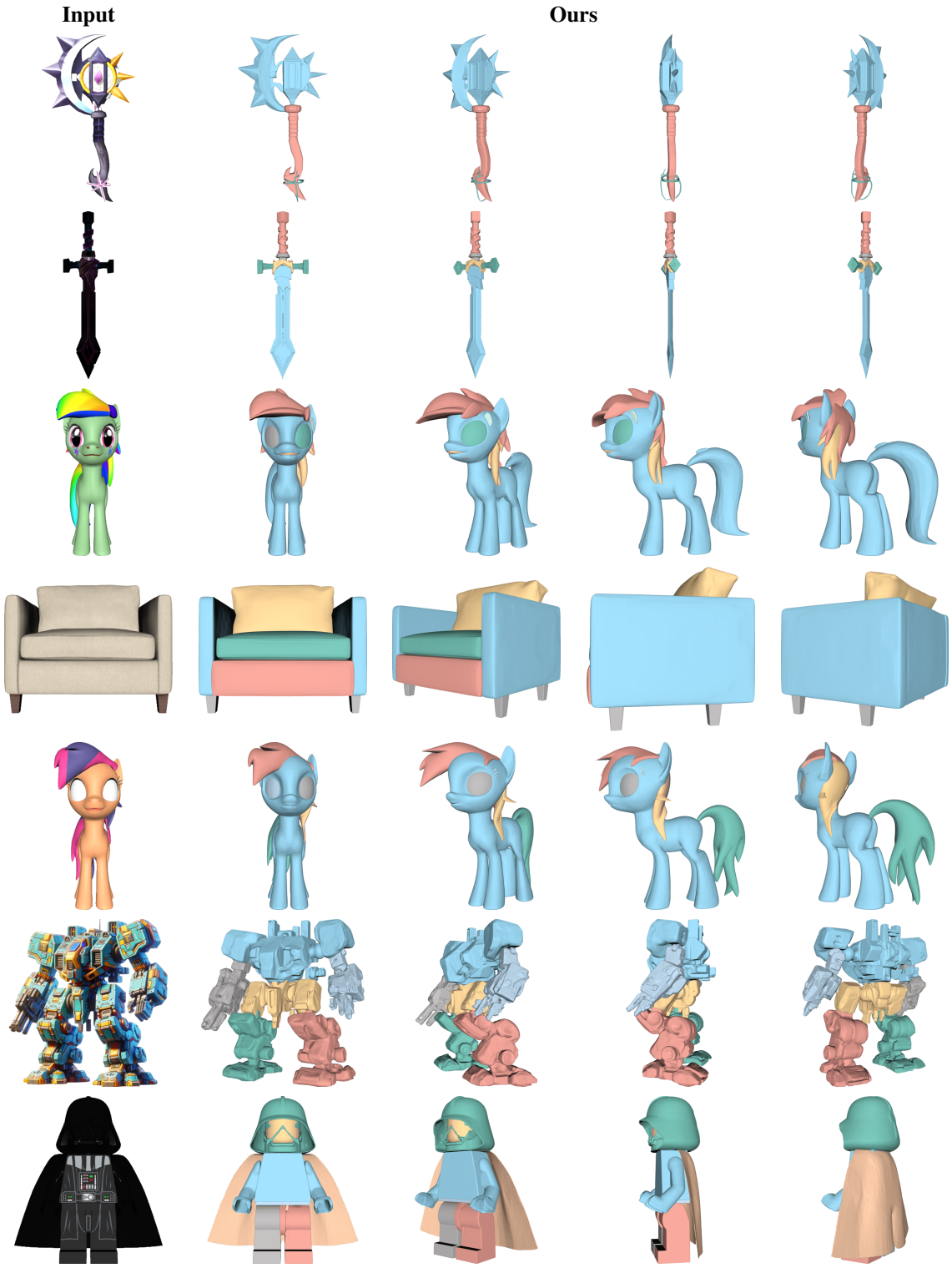

Figure 10: More results of image-conditioned 3D part-level object generation of PARTCRAFTER.

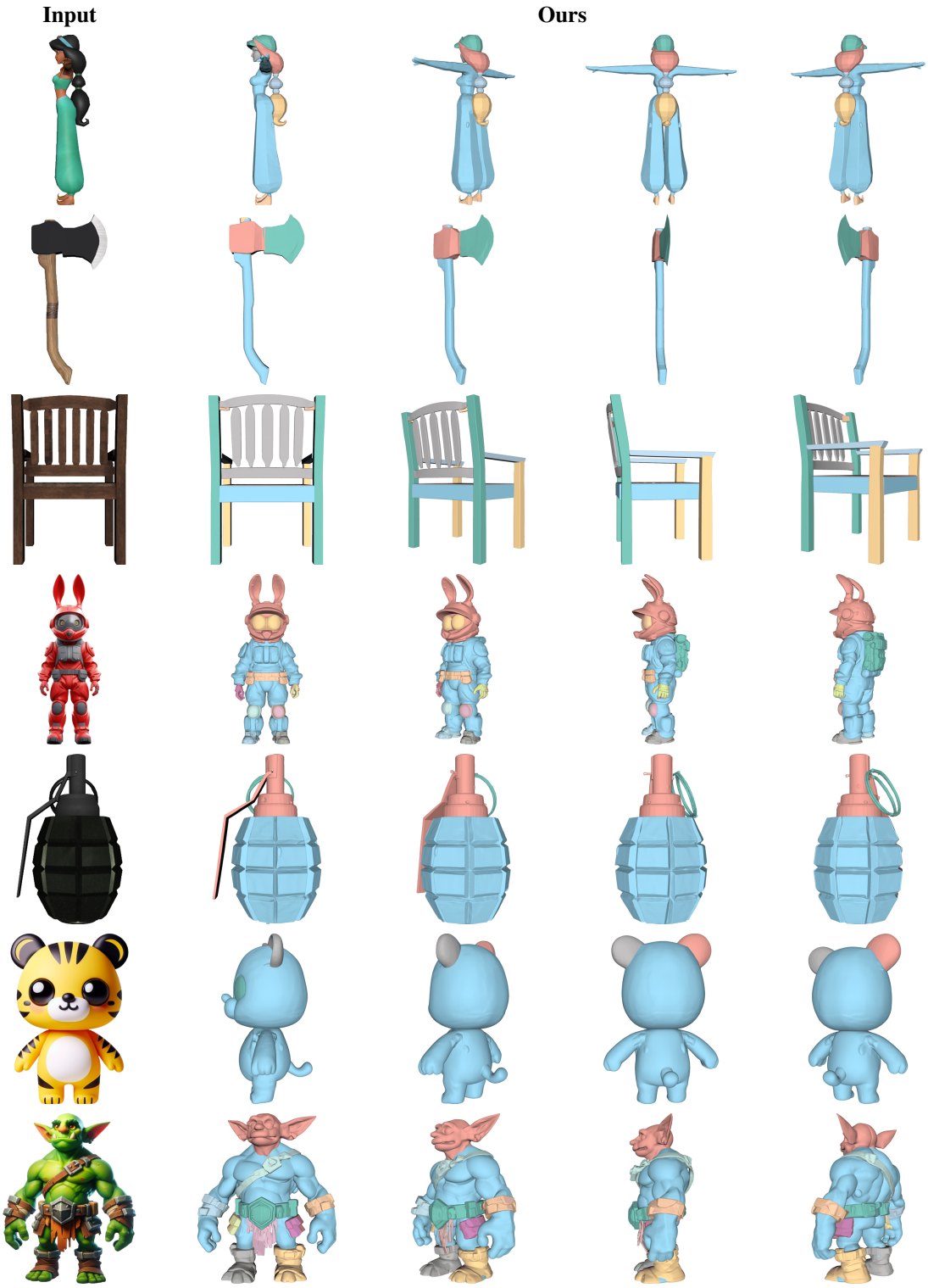

Figure 11: More results of image-conditioned 3D part-level object generation of PARTCRAFTER.

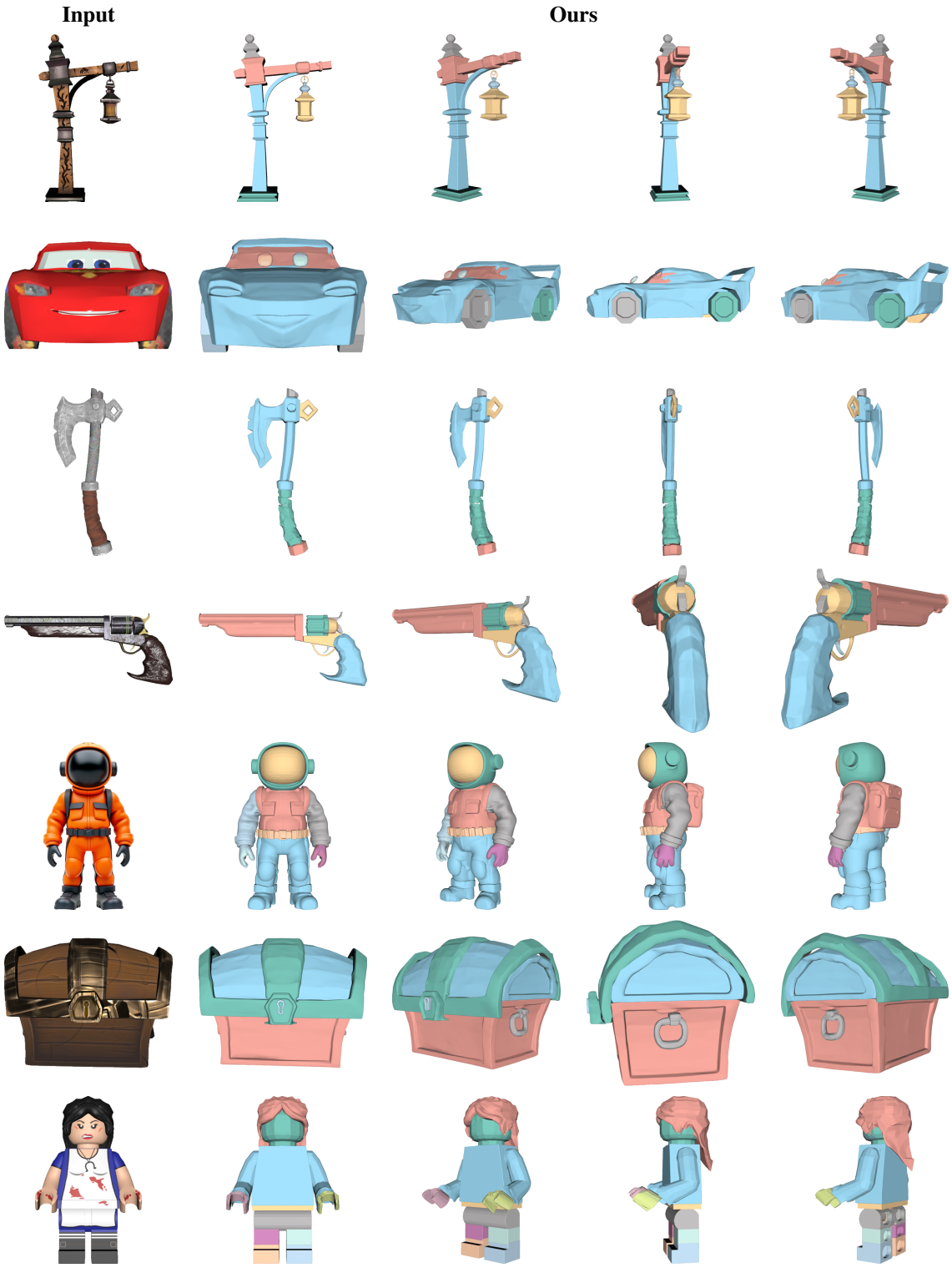

Figure 12: More results of image-conditioned 3D part-level object generation of PARTCRAFTER.

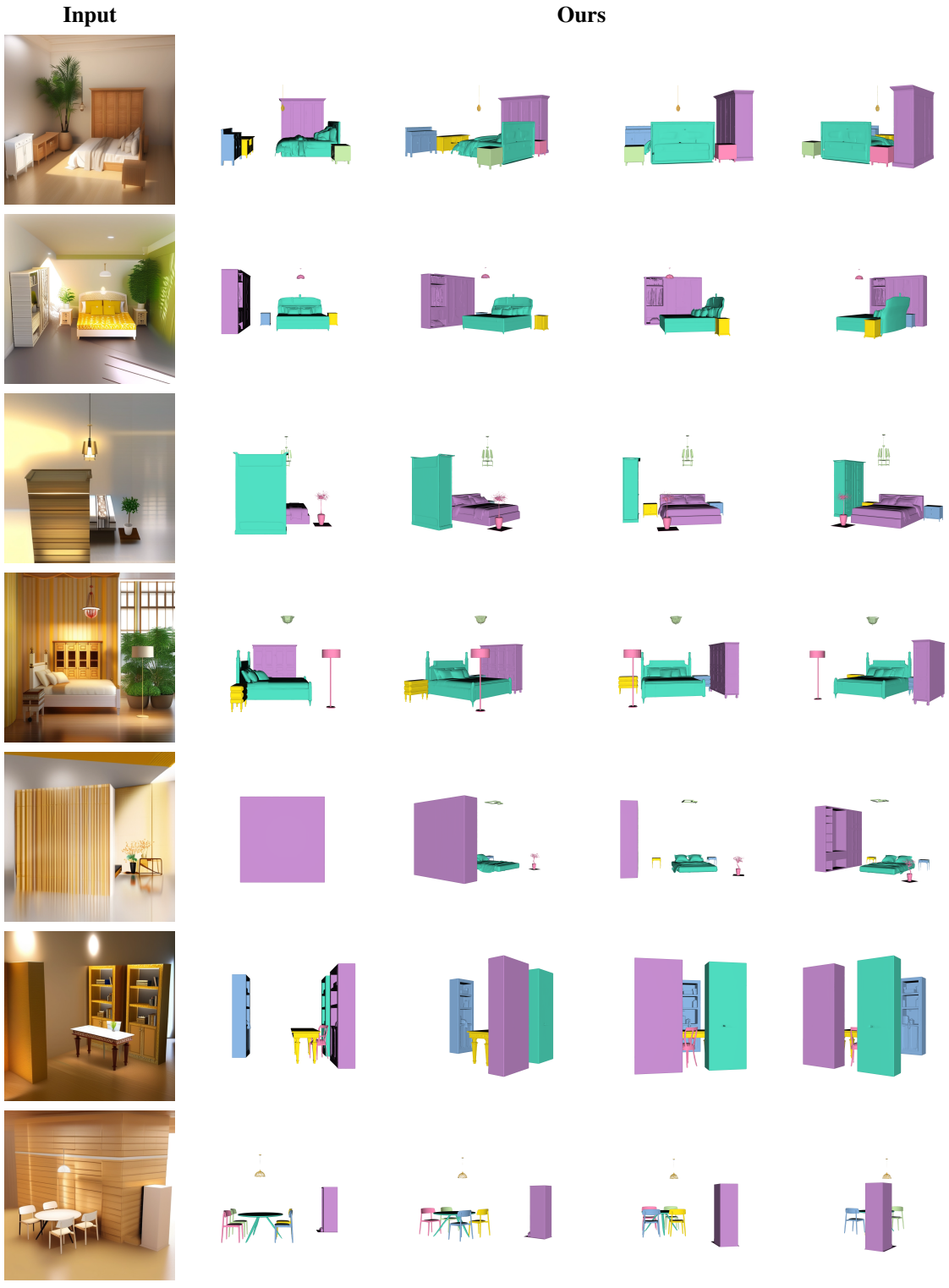

Figure 13: More results of image-conditioned 3D scene generation of PARTCRAFTER.