SCULPT: SUBTRACTIVE COMPOSITION FOR 3D PART GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Part-aware 3D generation aims to create digital assets that are coherent as complete objects while exposing structural parts for editing, material assignment, animation, and reuse. Existing methods impose this structure outside the native generation loop: segmentation-based methods partition an already generated shape, while additive methods synthesize parts from predefined layouts, boxes, or tokens and then reconcile them into a whole. The former fixes the object before part boundaries are determined; the latter leaves shared boundaries vulnerable to gaps, interpenetrations, and material discontinuities. In this paper, we propose **SCULPT**, a framework that addresses these challenges through *subtractive composition*. Given a complete object represented in a structured 3D latent space, SCULPT iteratively applies a joint split predictor to generate one extracted part together with the remaining object. The predictor performs a coupled denoising process conditioned on both the image and the current 3D state, so the extracted part and updated remainder are generated together rather than reconciled after generation. The joint split predictor processes both outputs on the union of their native sparse 3D supports, allowing neighboring supports to overlap rather than imposing a disjoint voxel partition. The rollout ends when the remainder support becomes empty or reaches a fixed safety cap, allowing the number of generated parts to adapt to each object within that bound. Experiments show state-of-the-art geometry on PartObjaverse with strong complete-object reconstruction after part assembly, and fine-grained textured decompositions for dataset, text-to-image, and real-photograph inputs beyond the benchmark.

1 INTRODUCTION

In modern graphics pipelines, a 3D asset is rarely a monolithic shape; artists rig limbs, assign materials to individual components, and edit sub-parts independently, yet expect the assembled whole to remain visually coherent. While recent single-image 3D generative models such as TRELLIS (Xiang et al., 2025), TRELLIS. 2 (Xiang et al., 2026), and Hunyuan3D 2.5 (Lai et al., 2025) excel at synthesizing detailed geometry and appearance from visual cues, they typically output a single, indivisible asset without considering the part structure. *Part-aware 3D generation* asks these models to produce both: a coherent whole and an object-dependent number of semantic components that can be manipulated independently and reassembled. This variable output cardinality is central to the task.

Existing approaches handle part structure in two main ways, neither of which fully resolves this challenge. Direct part synthesis (Liu et al., 2024; Chen et al., 2025a; Yang et al., 2025) represents the complete part set through image-space decisions, fixed slots, explicit layouts, or variable-length sequences, so the object-dependent cardinality becomes part of a single global output prediction. Segmentation-based methods (Yang et al., 2024d; Liu et al., 2025; Ma et al., 2025) instead label components on an already generated mesh or volume. They preserve the generated exterior, but a segmentation only partitions the existing surface: it never asks the generator to determine and complete both sides of a part boundary, including newly exposed contact surfaces and their materials. Both limitations motivate retaining the generator throughout decomposition without predicting the entire part set at once.

Modern holistic 3D generators themselves provide the missing prior: through whole-object generation, they learn recurring substructures such as legs, wheels, and handles, and how they form coherent

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

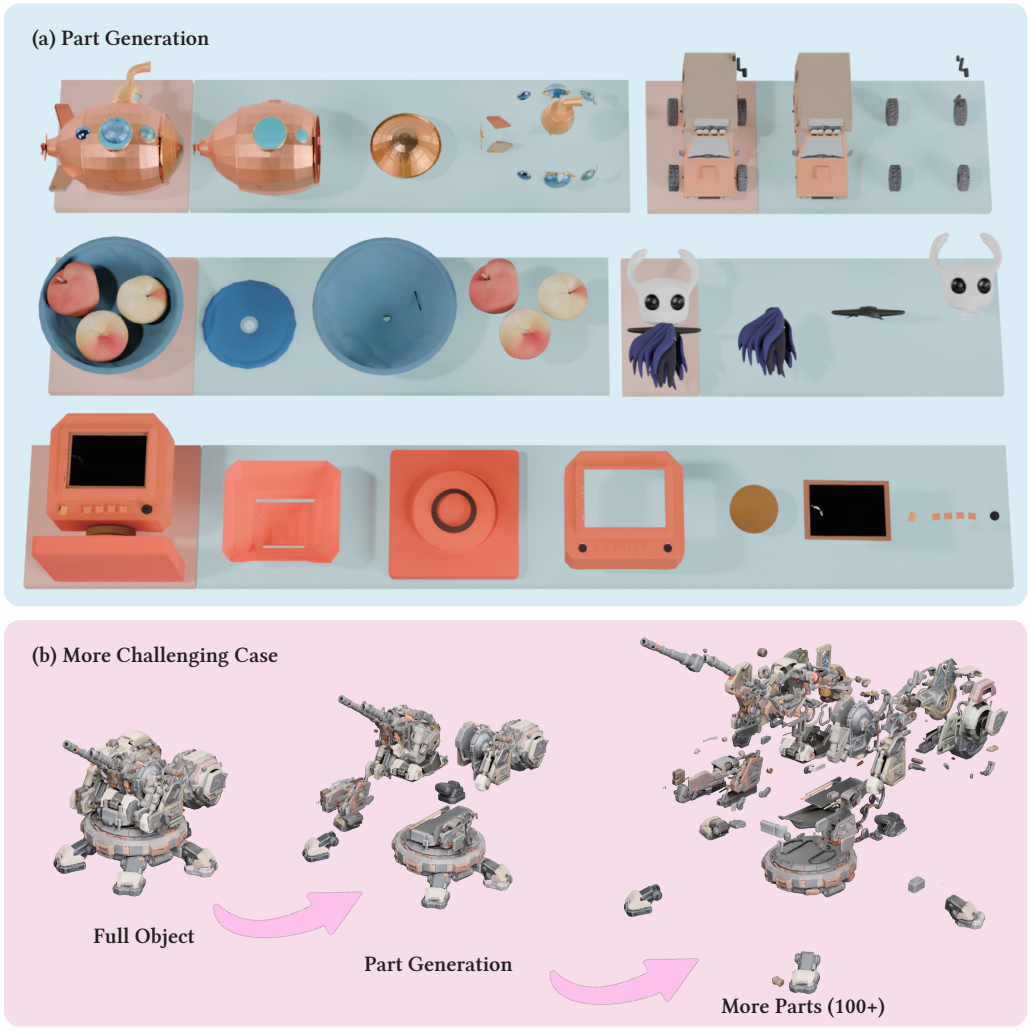

Figure 1: **SCULPT generates complete 3D objects together with coherent part structure.** Starting from an image-conditioned complete object, SCULPT subtractively extracts semantically meaningful components while retaining their alignment in the original object frame. (a) Part generation: complete objects and their extracted parts across diverse object categories, with coherent geometry, boundaries, and appearance. (b) More challenging case: recursive subtractive decomposition—by applying the split operation to previously extracted parts, SCULPT decomposes a complex 3D asset into more than 100 fine-grained components.

geometry and appearance. This suggests treating decomposition as a generation problem rather than recognition over a fixed asset. We therefore introduce **SCULPT**[1] and formulate part-aware generation as *subtractive composition*. Instead of emitting the complete part set in one prediction, SCULPT repeatedly applies a fixed-signature operation: given the current object state, it jointly generates one extracted part and the remaining object. The remainder is then passed to the next split, and the same operation is reused until its predicted support becomes empty or the rollout reaches a fixed safety cap. Cardinality is thus expressed through the rollout length rather than the dimensionality of a single prediction, and generating each part together with its remainder allows their boundary to be determined during generation rather than imposed afterward.

[1]Named after sculpture as a subtractive process: each step separates one component from the remaining whole.

To realize this generative split, we train a *joint split predictor* that maps the conditioning image and the current 3D state to the next extracted part and the updated remainder. It is built from decomposition flow transformer blocks, in which a joint denoising branch initialized from the holistic generator produces the coupled part–remainder target and a remainder-control branch injects features from the current 3D state. Unlike 2D separation methods constrained by projected pixels, or voxelized partitions that snap boundaries to grid faces, SCULPT jointly processes the part and remainder on the union of their native sparse supports and allows them to share a non-empty interface shell, while all outputs remain in the coordinate frame of the complete object. Figure 1 shows representative results.

In summary, our main contributions are:

- We formulate part-aware 3D generation as *subtractive composition*, turning a variable-cardinality output into recurrent, fixed-signature part–remainder splits whose number follows the rollout rather than a fixed set of output slots.
- We instantiate this formulation with a *joint split predictor* that adapts a holistic image-to-3D prior through image-conditioned joint denoising and remainder-conditioned control, so each part and its remainder are generated in a shared trajectory.
- We introduce native-support constraints for subtractive generation—a sparse-support composition loss, inference-time support clipping, and empty-remainder termination—that keep the recurrent state within the object being decomposed.
- SCULPT achieves state-of-the-art part-, semantic-group-, and object-level geometry on a part-annotated Objaverse benchmark and produces coherent textured decompositions for dataset images, a text-to-image-generated input, and a real photograph.

2 Related Work

2.1 Holistic 3D Object Generation

Complete 3D assets are generated from text or images by optimization against 2D diffusion priors (Poole et al., 2023; Lin et al., 2023; Shi et al., 2024), by direct 3D diffusion (Nichol et al., 2022; Jun & Nichol, 2023), or by feed-forward reconstruction (Hong et al., 2024; Tang et al., 2024; Xu et al., 2024). Recent systems such as TRELLIS (Xiang et al., 2025), TRELLIS. 2 (Xiang et al., 2026), Hunyuan3D 2.5 (Lai et al., 2025), UniLat3D (Wu et al., 2025), and TIGON (Cen et al., 2026) build on native 3D latent representations to support high-quality asset generation, and SCULPT adapts such a holistic generator into a recurrent split predictor. An extended discussion is given in Appendix C.

2.2 3D Part Segmentation

3D part segmentation methods decompose an object into semantically meaningful components. Early approaches segment shapes into primitives with learned point-wise features (Yan et al., 2021) trained on datasets such as PartNet (Mo et al., 2019). Recent methods obtain part cues from 2D foundation models (Oquab et al., 2024; Siméoni et al., 2025; Kirillov et al., 2023) or operate with native 3D backbones (Qi et al., 2017; Wu et al., 2024), including PartSLIP, Point-SAM, PartSAM, SAMPart3D, and P3-SAM (Liu et al., 2023; Zhou et al., 2025; Zhu et al., 2026; Yang et al., 2024d; Ma et al., 2025), and segmentation has also been lifted to reconstructed scene representations (Ye et al., 2024). Because these methods operate on a fixed asset, they preserve the generated geometry and provide a strong reference for part-level decomposition. Their output, however, is a partition of an existing shape rather than a generative process, so the decomposition does not directly update the object state as parts are extracted.

2.3 Part-Structured 3D Generation and Reconstruction

Part-structured 3D generation and reconstruction make part structure native to the synthesis or recovery process. Beyond early part-hierarchy models (Mo et al., 2020), some methods use multi-view generation or image-space part cues to recover complete semantic assemblies, including PartGen (Chen et al., 2025a), HoloPart (Yang et al., 2026), and Part123 (Liu et al., 2024). Other systems model part-structured objects through dual volume packing (Tang et al., 2025), compositional latent diffusion in PartCrafter (Lin et al., 2025), semantic decoupling and structural cohesion in OmniPart (Yang

et al., 2025), contextual part latents in CoPart (Dong et al., 2025), bounding-box-prompted shape decomposition in X-Part (Yan et al., 2026), or full-resolution per-part synthesis in FullPart (Ding et al., 2026). BANG (Zhang et al., 2025) and Part-X-MLLM (Wang et al., 2026) further expose part structure through exploded-view dynamics and language-planned boxes. Across these methods, maintaining coherence among generated parts remains an explicit design problem. Closest to our formulation, UniPart (He et al., 2026) likewise taps the part-aware priors that emerge in whole-object generative learning, yet it realizes decomposition as a latent segmentation stage followed by per-part diffusion. AutoPartGen (Chen et al., 2025b) generates parts autoregressively with automatic termination, but each part is added by conditioning on the whole object and previously generated parts, so no explicit remainder state constrains what remains to be generated, and its geometry-only latents do not model appearance. In contrast, SCULPT carries an explicit remaining-object state through every split and denoises each part jointly with that remainder on the union of their native sparse supports, keeping boundaries, geometry, and materials consistent as the decomposition proceeds.

2.4 Conditional Control and Joint Generation

ControlNet (Zhang et al., 2023) and adapter-based methods (Mou et al., 2024; Ye et al., 2023) steer pretrained diffusion models with additional spatial or image conditions while retaining their learned priors, and JointNet (Zhang et al., 2024) and JointDiT (Kwon et al., 2025) extend diffusion backbones to coupled outputs such as RGB-depth generation. SCULPT combines both ideas—a zero-initialized control branch and a packed joint target—but applies them to recurrent 3D part decomposition conditioned on an evolving remaining-object state, which these works do not address (Appendix C).

3 Method

3.1 Overview

Given a conditioning image $\mathcal{I}$, SCULPT generates an ordered, variable-length collection of 3D parts. Each part latent decodes to a textured mesh with geometry and material in the coordinate frame of the complete object, so the decoded parts can be combined directly into one asset. We call the process *subtractive composition*: starting from a complete object, the model repeatedly separates one part and carries the remaining object to the next step.

SCULPT uses the pretrained image-conditioned TRELLIS. 2 generator G_ϕ (Xiang et al., 2026) to obtain the complete-object latent $\boldsymbol{z}_0$. Its core module is a *joint split predictor* P_θ. Let $\boldsymbol{z}_{i-1}$ be the current remaining-object latent before split i. The predictor takes this latent together with the image and returns the next part $\hat{\boldsymbol{z}}_i$ and the updated remainder $\boldsymbol{z}_i$:

$$\boldsymbol{z}_0 = G_\phi(\mathcal{I}), \tag{1}$$

$$(\hat{\boldsymbol{z}}_i, \boldsymbol{z}_i) = P_\theta(\boldsymbol{z}_{i-1}, \mathcal{I}). \tag{2}$$

Here and below, a hatted quantity denotes the extracted part and the corresponding unhatted quantity denotes the remainder. The extracted latent is stored, while the remainder is used as the input to the next split. Repeating the same two-output prediction allows different objects to produce different numbers of parts, up to the fixed rollout cap. Figure 2 summarizes this process.

Structured latent backbone. Every complete object, part, and remainder is a TRELLIS. 2 structured latent (Xiang et al., 2026) $\boldsymbol{z} = (\boldsymbol{z}^v, \boldsymbol{z}^g, \boldsymbol{z}^m)$, whose sparse-structure, geometry, and material components ($s \in \{v, g, m\}$) are generated by three successive rectified-flow stages (Appendix A.1). The sparse-structure latent determines an active O-Voxel support $\mathcal{O} \in \{0,1\}^{N^3}$ at grid resolution N, on which the geometry and material stages generate aligned features; we use $\cup$, $\cap$, and $\varnothing$ for elementwise operations on supports. A decoder $\mathcal{D}$ maps the latent to a textured mesh in a shared $[-1,1]^3$ object frame, so parts and remainders need no normalization or registration before composition. SCULPT retains this stage order and flow parameterization, but changes the clean target to a coupled part–remainder pair and conditions the velocity network on the current remainder.

3.2 Subtractive Training Sequences

We construct supervision for both outputs of every step from part-structured 3D assets. The complete mesh and all of its parts are jointly normalized to the shared object frame without per-part rescaling

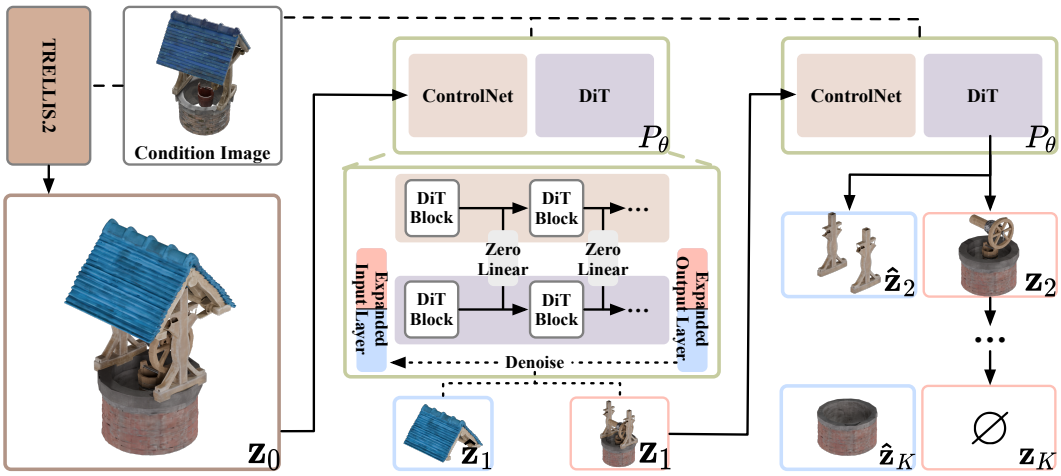

Figure 2: **SCULPT turns whole-object generation into recurrent part–remainder prediction.** TRELLIS. 2 maps the conditioning image $\mathcal{I}$ to a complete structured latent $\boldsymbol{z}_0$. At split i, the image- and remainder-conditioned predictor P_θ jointly denoises an extracted part $\hat{\boldsymbol{z}}_i$ (blue) and the updated remainder $\boldsymbol{z}_i$ (red); only the remainder is carried forward until its support becomes empty or the rollout reaches its cap. Inset: a ControlNet branch encodes the current remainder and injects zero-initialized residuals into the DiT blocks.

and placed in one deterministic extraction order per asset (Appendix A.2). Let $\mathcal{M}_0$ be a complete mesh with K ordered parts, let $\hat{\mathcal{M}}_i$ be the part extracted at step i, and let $\mathcal{M}_i$ be the mesh remaining after the first i parts have been extracted, so that

$$\mathcal{M}_{i-1} = \hat{\mathcal{M}}_i \oplus \mathcal{M}_i, \quad i = 1, \ldots, K, \qquad \mathcal{M}_K = \varnothing, \tag{3}$$

where $\oplus$ denotes mesh union in the shared frame. One annotated asset thus provides K split examples, the last of which has an empty target remainder that supervises termination.

We render the conditioning image $\mathcal{I}$ from the complete mesh. Each extracted part $\hat{\mathcal{M}}_i$ and remainder $\mathcal{M}_i$ is then converted independently to an O-Voxel in the common frame and encoded as $\hat{\boldsymbol{z}}_i$ and $\boldsymbol{z}_i$. Independent conversion allows the part support $\hat{\mathcal{O}}_i$ and remainder support $\mathcal{O}_i$ to overlap at their contact region instead of forcing a disjoint voxel partition. The resulting latent training tuple is

$$\mathcal{T}_i = (\mathcal{I}, \boldsymbol{z}_{i-1}; \hat{\boldsymbol{z}}_i, \boldsymbol{z}_i), \qquad i = 1, \ldots, K. \tag{4}$$

This matches the interface of Eq. 2 for nonempty and terminal remainders alike; the terminal target is an all-empty occupancy grid with zero-length geometry and material tensors. During training, $\boldsymbol{z}_{i-1}$ is encoded from the annotated remainder mesh; during inference, it is the remainder predicted by the preceding split.

3.3 Decomposition Flow Transformer

The split predictor uses two complementary conditions: the image describes the complete asset, while the current remainder specifies which 3D content is still present. We implement P_θ as three stage-wise *decomposition flow transformers*, $P_\theta = \{P_\theta^v, P_\theta^g, P_\theta^m\}$, following the sparse-structure, geometry, and material order of the backbone. For stage s, we pack the extracted-part and remainder targets into $\boldsymbol{y}_i^s = (\hat{\boldsymbol{z}}_i^s, \boldsymbol{z}_i^s)$ and denoise them jointly. At the sparse-structure stage, $\boldsymbol{y}_i^v$ is a two-channel dense tensor on the N^3 grid. At the geometry and material stages, the part and remainder can occupy different sparse coordinates, so we place their features on the union support $\hat{\mathcal{O}}_i \cup \mathcal{O}_i$, jointly denoise the packed features, and route the two outputs back to their respective supports (Appendix A.3). The two outputs therefore share the same denoising computation and image context at every stage.

Remainder-conditioned blocks. Each stage predictor contains a joint denoising branch and a parallel remainder-control branch. Let $\boldsymbol{h}_{\ell,i}^s$ be the packed feature in the joint branch at block ℓ, let $\boldsymbol{c}_{\ell,i}^s$ be the control feature, and let $\boldsymbol{e}_\mathcal{I}$ and $\boldsymbol{e}_t$ be the image and flow-time embeddings. We denote

the control and joint transformer blocks by C_ℓ^s and B_ℓ^s, and their residual projection by Z_ℓ^s. A block updates the two branches as

$$\boldsymbol{c}^s_{\ell+1,i} = C^s_\ell\big(\boldsymbol{c}^s_{\ell,i}, \boldsymbol{z}^s_{i-1}; \boldsymbol{e}_\mathcal{I}, \boldsymbol{e}_t\big)\,, \tag{5}$$

$$\boldsymbol{h}^s_{\ell+1,i} = B^s_\ell\big(\boldsymbol{h}^s_{\ell,i}; \boldsymbol{e}_\mathcal{I}, \boldsymbol{e}_t\big) + Z^s_\ell\big(\boldsymbol{c}^s_{\ell+1,i}\big)\,. \tag{6}$$

The joint blocks B_ℓ^s are initialized from the corresponding TRELLIS. 2 checkpoint. The residual projections Z_ℓ^s are initialized to zero, and both branches are then trained for the split task. This design starts from the pretrained image-to-3D generator while adding the current 3D remainder as a block-wise condition.

Training objectives. For each stage s, the packed target follows the rectified-flow path

$$\boldsymbol{y}^s_i(t) = (1-t)\boldsymbol{y}^s_i + t\boldsymbol{\epsilon}. \tag{7}$$

We train the stage-wise velocity predictor with

$$\mathcal{L}^s_{\text{flow}} = \mathbb{E}\left[\left\|\boldsymbol{v}^s_\theta(\boldsymbol{y}^s_i(t), t, \mathcal{I}, \boldsymbol{z}^s_{i-1}) - (\boldsymbol{\epsilon} - \boldsymbol{y}^s_i)\right\|_2^2\right], \tag{8}$$

where the expectation is over training tuples, flow times, and noise. The packed target fixes the roles of the two outputs, while joint denoising lets their predictions interact throughout the trajectory. Because this loss never compares the combined sparse support with the input remainder, we add a composition loss at the sparse-structure stage. From the predicted velocity, we first estimate the clean packed state,

$$\tilde{\boldsymbol{y}}^v_{i,0}(t) = \boldsymbol{y}^v_i(t) - t\,\boldsymbol{v}^v_\theta(\boldsymbol{y}^v_i(t), t, \mathcal{I}, \boldsymbol{z}^v_{i-1}). \tag{9}$$

After unpacking its two occupancy-logit channels and applying a sigmoid, we obtain part and remainder occupancy probabilities $(\hat{\boldsymbol{p}}_i, \boldsymbol{p}_i)$. Their differentiable union is

$$\boldsymbol{u}_i = \mathbf{1} - (\mathbf{1} - \hat{\boldsymbol{p}}_i) \odot (\mathbf{1} - \boldsymbol{p}_i). \tag{10}$$

where $\odot$ denotes elementwise multiplication. We compare this union with the support $\mathcal{O}_{i-1}$ of the input remainder using voxel-wise binary cross-entropy, where q indexes grid voxels. The sparse-stage objective adds this term with weight λ_{comp}:

$$\mathcal{L}_{\text{comp}} = -\frac{1}{N^3}\sum_q \left[\mathcal{O}_{i-1,q}\log u_{i,q} + (1-\mathcal{O}_{i-1,q})\log(1-u_{i,q})\right], \tag{11}$$

$$\mathcal{L}^v = \mathcal{L}^v_{\text{flow}} + \lambda_{\text{comp}}\mathcal{L}_{\text{comp}}, \qquad \mathcal{L}^g = \mathcal{L}^g_{\text{flow}}, \qquad \mathcal{L}^m = \mathcal{L}^m_{\text{flow}}. \tag{12}$$

The three stages are trained separately. The composition term encourages the predicted part and remainder to jointly cover the current input support; since it constrains their union rather than their intersection, voxels may be active in both outputs. Such overlapping boundary voxels form an *interface shell* that avoids a hard voxel-face partition between two parts meeting in 3D.

3.4 Subtractive Composition

At inference time, predictions from one split become the input to the next. To keep this recurrent state within the current object, we convert the two sparse-stage probability maps into hard supports with a fixed occupancy threshold τ_{occ} and clip both supports to $\mathcal{O}_{i-1}$:

$$\hat{\mathcal{O}}_i = \mathbf{1}[\hat{\boldsymbol{p}}_i > \tau_{\text{occ}}] \cap \mathcal{O}_{i-1}, \qquad \mathcal{O}_i = \mathbf{1}[\boldsymbol{p}_i > \tau_{\text{occ}}] \cap \mathcal{O}_{i-1}. \tag{13}$$

Here $\mathbf{1}[\cdot]$ is the indicator function. We mask the predicted sparse latents to these clipped supports before running the geometry and material stages. The updated remainder latent $\boldsymbol{z}_i$ is then carried to the next split, while $\hat{\boldsymbol{z}}_i$ is stored as an output part.

Clipping yields a non-expanding sequence of supports, $\hat{\mathcal{O}}_i \subseteq \mathcal{O}_{i-1}$ and $\mathcal{O}_i \subseteq \mathcal{O}_{i-1} \subseteq \cdots \subseteq \mathcal{O}_0$, so every predicted state stays inside the initial object. The rollout stops when the remainder support is empty or at the fixed cap $K_{\max} = 24$; a nonempty capped remainder is retained as one additional output rather than silently discarded. Each output latent is decoded independently by $\mathcal{D}$, and the meshes are united in the shared object frame without per-part rescaling, snapping, or registration, so a fixed two-output interface yields a number of parts that follows the rollout length. Algorithm 1 in Appendix A.5 gives the complete procedure, and Appendix Fig. 4 shows an example rollout.

Table 1: **Comparison on PartObjaverse under matched single-image conditioning.** Geometry (CD, F1) at the part and object levels, and rendered appearance of the extracted parts (CLIP score and FID with Inception and DINOv2 features); semantic-group-level geometry is in Appendix Table 4. Bold and underlined entries mark the best and second-best values in each column; "–" denotes an unavailable measurement. PF and HP abbreviate PartField and HoloPart.

	Part geometry			Part appearance			Object geometry		
Method	CD↓	F1@.1↑	F1@.05↑	CLIP↑	FD_{Incep} ↓	FD_{DINO} ↓	CD↓	F1@.1↑	F1@.05↑
Part123	0.0188	0.8133	0.6279	0.8431	64.26	587.01	0.0126	0.8229	0.5739
OmniPart	0.0136	0.8612	0.7025	0.8242	83.52	344.05	0.0032	0.9690	0.8732
TRELLIS+PF	0.0127	0.8750	0.7169	0.8630	27.94	238.14	0.0037	0.9634	0.8563
TRELLIS+SAM3D	0.0278	0.7410	0.5520	0.8175	117.65	725.24	–	–	–
TRELLIS+PF+HP	0.0137	0.8521	0.6862	0.8501	33.19	232.20	0.0034	0.9654	0.8639
TRELLIS.2+PF	<u>0.0115</u>	**0.8897**	<u>0.7554</u>	<u>0.8631</u>	<u>27.88</u>	237.97	<u>0.0021</u>	<u>0.9793</u>	<u>0.9119</u>
TRELLIS.2+SAM3D	0.0531	0.5323	0.3283	0.8176	117.86	725.93	–	–	–
TRELLIS.2+PF+HP	0.0453	0.6016	0.3966	0.8504	33.15	<u>231.66</u>	0.0332	0.6474	0.4345
SCULPT (ours)	**0.0107**	<u>0.8858</u>	**0.7599**	**0.9075**	**20.73**	**168.03**	**0.0020**	**0.9839**	**0.9212**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Training data. PartVerse-XL (Ding et al., 2026) provides the supervision required by our subtractive formulation: each asset contains human-refined part meshes aligned with its complete mesh. The dataset is curated from Objaverse-XL (Deitke et al., 2023). After quality filtering and removal of every asset that matches an evaluation mesh (Appendix B.2), the training set contains 37,425 objects and 330,455 supervised part–remainder splits.

Evaluation data. We evaluate geometry on PartObjaverse (Yang et al., 2024d), whose 200 meshes carry instance-level and semantic part annotations created independently of PartVerse-XL, enabling evaluation of parts, semantic groups, and the reassembled object. Figure 3 adds two dataset images, a text-to-image-generated input, and a real photograph beyond the benchmark.

Implementation details. The three stage-specific split predictors are initialized from the public TRELLIS. 2 checkpoints (Xiang et al., 2026) and trained for 800K steps in total on 32 GPUs with $\lambda_{\text{comp}} = 0.1$; optimizer settings and token-count filtering are given in Appendix B.3.

Metrics. To separate the quality of generated components from the fidelity of their assembled whole, we evaluate in the shared $[-1, 1]^3$ frame at the *part level*, where Hungarian-matched predicted and reference parts are scored individually, and at the *object level*, where the assembled parts are compared with the complete reference mesh; the *semantic-group level*, which unions matched predictions sharing a semantic label, is reported in Appendix B.4. Geometry is measured by Chamfer distance (CD) and F1 at thresholds 0.1 and 0.05, the latter capturing finer agreement. Appearance is measured at the part level on rendered views of the textured extracted parts: CLIP score (Radford et al., 2021) measures semantic similarity to the conditioning image, and FID (Heusel et al., 2017) compares the distribution of rendered parts with that of the rendered reference parts in Inception and DINOv2 (Oquab et al., 2024) feature spaces.

Compared methods. The baselines cover both alternatives to subtractive generation: direct part generation with Part123 (Liu et al., 2024) and OmniPart (Yang et al., 2025), and post-hoc decomposition of a TRELLIS (Xiang et al., 2025) or TRELLIS. 2 (Xiang et al., 2026) object with PartField (Liu et al., 2025) or SAM3D (Yang et al., 2023), optionally followed by HoloPart completion (Yang et al., 2026). All methods receive the same conditioning image when supported and use official checkpoints with the authors' recommended inference settings.

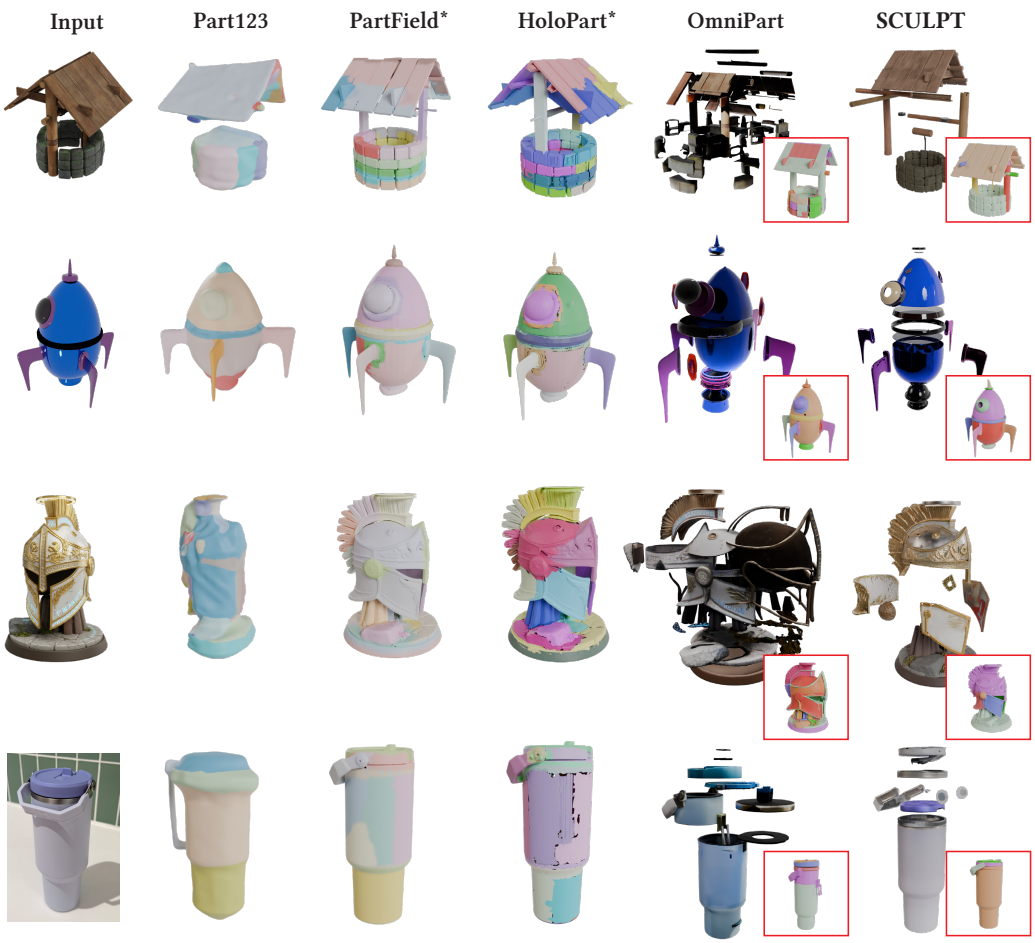

Figure 3: **Qualitative Comparison.** Each row uses the same conditioning image: two dataset images, one text-to-image example, and one real photograph (bottom). An asterisk marks a baseline rendered with semantic part colors rather than generated textures. Red-framed insets show the OmniPart and SCULPT parts reassembled in their common object frame.

4.2 Quantitative Comparison

Table 1 asks whether a method can improve individual parts without degrading the assembled object. SCULPT achieves the lowest CD and the highest F1@.05 at both the part and object levels, together with the best object-level F1@.1, so its advantage is not confined to either isolated components or the final union; the semantic-group level in Appendix Table 4 follows the same pattern.

Direct part generation. Relative to OmniPart, the strongest baseline in this group, SCULPT improves every part-level and object-level geometry metric, reducing object-level CD from 0.0032 to 0.0020. Starting from a complete-object latent therefore does not merely preserve the initial whole; it also yields more accurate individual components than direct part generation.

Post-hoc decomposition. TRELLIS. 2+PartField is the most competitive post-hoc baseline and shares the same complete-object generator. SCULPT reduces its CD by 7.0% and 4.8% at the part and object levels and improves F1@.05 at both. The baseline remains slightly higher on the coarser part-level F1@.1, but at the stricter 0.05 threshold SCULPT is better, consistent with its lower CD and stronger fine-scale agreement, and these more accurate splits reassemble into a more faithful complete shape.

Table 2: **Component ablations on PartObjaverse at the part level.** "Adapt g, m" denotes adaptation of the geometry and material stages, $\mathcal{L}_{\text{comp}}$ is composition supervision, and "Clip" is inference-time clipping to the current remainder. A checkmark enables a component and "–" disables it. Lower CD and higher F1 are better; bold marks the best value.

Variant	Adapt g, m	$\mathcal{L}_{\text{comp}}$	Clip	CD↓	F1@.1↑	F1@.05↑
Voxel only	–	✓	✓	0.0439	0.6801	0.5405
w/o comp. loss	✓	–	✓	0.0279	0.7379	0.5636
w/o clipping	✓	✓	–	0.0260	0.7517	0.5728
Full	✓	✓	✓	**0.0107**	**0.8858**	**0.7599**

Appearance. The rendered parts show the largest margins. SCULPT raises the part-level CLIP score from 0.8631 (TRELLIS. 2+PartField) to 0.9075 and lowers Inception and DINOv2 FID from the best baseline values of 27.88 and 231.66 to 20.73 and 168.03, reductions of 26% and 27%. Post-hoc decomposition inherits the whole-object texture but leaves cut surfaces untextured, and HoloPart completion recovers only part of this gap, whereas SCULPT generates geometry and material for both sides of every split, so its extracted parts render as complete textured objects and achieve the best appearance fidelity.

4.3 Qualitative Comparison

Figure 3 tests whether the predicted components remain visually meaningful under more varied image conditions. The examples cover structures that are difficult to recover as useful parts: a well with a broad roof, thin supports, and a recessed bucket; a rocket with a central body, nose, and narrow appendages; a text-to-image ornament with many small protrusions; and a real tumbler that must separate into lid, handle, and body. Across these cases, SCULPT separates recognizable components without moving them out of the coordinate frame of the complete object. The red-framed insets expose a second requirement hidden when parts are viewed separately: their union should recover the complete asset. SCULPT's reassemblies retain the silhouette and relative placement of the conditioning object across coarse structures and small accessories, unlike components positioned independently after synthesis, which explains why the part-level gains in Table 1 survive assembly. SCULPT also decodes texture for every part, retaining cues such as wood, metal, and ceramic.

4.4 Ablation Studies

Table 2 isolates the design choices that turn a whole-object generator into a recurrent split predictor by removing one capability at a time from the full model. The *Voxel only* variant adapts only the sparse-structure predictor; its CD quadruples (0.0107 to 0.0439), so sparse support alone is insufficient and the downstream stages must also adapt. Removing composition supervision (CD 0.0279) or inference-time support clipping (CD 0.0260) is also clearly harmful: the predicted part and remainder must be trained to cover the current object together, and a split that extends beyond the current remainder weakens the recurrent decomposition. Together, these ablations connect the end-to-end gains to the formulation: full-stage adaptation carries the split through the structured latent hierarchy, composition supervision trains the two predicted supports to cover the current object together, and clipping keeps successive predictions within the current subtractive state.

5 Conclusion

We introduced **SCULPT**, a subtractive formulation of part-aware 3D generation that turns a complete-object latent into the initial state of a recurrent generative split, each step jointly producing one part and the object that remains. This transition provides a fixed prediction signature, an explicit state that conditions every later decision, and variable cardinality through the trajectory length. On PartObjaverse, SCULPT achieves the best CD and F1@.05 at the part, semantic-group, and object levels, showing that fine-grained decomposition does not come at the expense of the assembled shape, and ablations establish the value of downstream-stage adaptation, composition supervision, and inference-time clipping. Calibrated stopping, granularity control, and interface-aware objectives are natural next steps.

REFERENCES

Jiazhong Cen, Jiemin Fang, Sikuang Li, Guanjun Wu, Chen Yang, Taoran Yi, Zanwei Zhou, Zhikuan Bao, Lingxi Xie, Wei Shen, and Qi Tian. Text-image conditioned 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 604–614, June 2026.

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

Minghao Chen, Jianyuan Wang, Roman Shapovalov, Tom Monnier, Hyunyoung Jung, Dilin Wang, Rakesh Ranjan, Iro Laina, and Andrea Vedaldi. AutoPartGen: Autoregressive 3D part generation and discovery. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=ljJGBcpn7q.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Lihe Ding, Shaocong Dong, Yaokun Li, Chenjian Gao, Xiao Chen, Rui Han, Yihao Kuang, Hong Zhang, Bo Huang, Zhanpeng Huang, Zibin Wang, Dan Xu, and Tianfan Xue. Fullpart: Generating each 3d part at full resolution. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QlRlE7a1p4.

Shaocong Dong, Lihe Ding, Xiao Chen, Yaokun Li, Yuxin Wang, Yucheng Wang, Qi Wang, Jaehyeok Kim, Chenjian Gao, Zhanpeng Huang, Zibin Wang, Tianfan Xue, and Dan Xu. From one to more: Contextual part latents for 3d generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8230–8240, October 2025.

Xufan He, Yushuang Wu, Xiaoyang Guo, Chongjie Ye, Jiaqing Zhou, Tianlei Hu, Xiaoguang Han, and Dong Du. UniPart: Part-level 3D generation with unified 3D geom-seg latents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34227–34236, June 2026.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sllU8vvsFF.

Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions, 2023. URL https://arxiv.org/abs/2305.02463.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October 2023.

Byung-Ki Kwon, Qi Dai, Hyoseok Lee, Chong Luo, and Tae-Hyun Oh. Jointdit: Enhancing rgb-depth joint modeling with diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25261–25271, October 2025.

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, Sheng Zhang, Xin Huang, Di Luo, Fan Yang, Fang Yang, Lifu Wang, Sicong Liu, Yixuan Tang, Yulin Cai, Zebin He, Tian Liu, Yuhong Liu, Jie Jiang, Linus, Jingwei Huang, and Chunchao Guo. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 300–309, June 2023.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=0e1KaHtgvU.

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*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705250. doi: 10.1145/3641519.3657482. URL https://doi.org/10.1145/3641519.3657482.

Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21736–21746, June 2023.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9704–9715, October 2025.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9970–9980, June 2024.

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

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J. Mitra, and Leonidas J. Guibas. StructEdit: Learning structural shape variations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8859–8868, June 2020.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i5.28226. URL https://doi.org/10.1609/aaai.v38i5.28226.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts, 2022. URL https://arxiv.org/abs/2212.08751.

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

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FjNys5c7VyY.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5105–5114, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FUgrjq2pbB.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

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 *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part IV*, pp. 1–18, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73234-8. doi: 10.1007/978-3-031-73235-5_1. URL https://doi.org/10.1007/978-3-031-73235-5_1.

Jiaxiang Tang, Ruijie Lu, Max Li, Zekun Hao, Xuan Li, Fangyin Wei, Shuran Song, Gang Zeng, Ming-Yu Liu, and Tsung-Yi Lin. Efficient part-level 3d object generation via dual volume packing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=qbVbZWxUib.

Chunshi Wang, Junliang Ye, Yunhan Yang, Yang Li, Zizhuo Lin, Jun Zhu, Zhuo Chen, Yawei Luo, and Chunchao Guo. Part-X-MLLM: Part-aware 3D multimodal large language model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=WffiETiSeU.

Guanjun Wu, Jiemin Fang, Chen Yang, Sikuang Li, Taoran Yi, Jia Lu, Zanwei Zhou, Jiazhong Cen, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Xinggang Wang, and Qi Tian. Unilat3d: Geometry-appearance unified latents for single-stage 3d generation, 2025. URL https://arxiv.org/abs/2509.25079.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851, June 2024.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21469–21480, 2025.

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

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

Siming Yan, Zhenpei Yang, Chongyang Ma, Haibin Huang, Etienne Vouga, and Qixing Huang. HPNet: Deep primitive segmentation using hybrid representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2753–2762, October 2021.

Xinhao Yan, Jiachen Xu, Yang Li, Changfeng Ma, Yunhan Yang, Chunshi Wang, Zibo Zhao, Zeqiang Lai, Yunfei Zhao, Zhuo Chen, and Chunchao Guo. X-part: High fidelity and structure coherent shape decomposition and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27062–27071, June 2026.

Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Gaussianobject: High-quality 3d object reconstruction from four views with gaussian splatting. *ACM Trans. Graph.*, 43(6), November 2024a. ISSN 0730-0301. doi: 10.1145/3687759. URL https://doi.org/10.1145/3687759.

Xiuyu Yang, Yunze Man, Jun-Kun Chen, and Yu-Xiong Wang. SceneCraft: Layout-guided 3D scene generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language guided generation of 3D embodied AI environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16227–16237, June 2024c.

Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes, 2023. URL https://arxiv.org/abs/2306.03908.

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, 2024d. URL https://arxiv.org/abs/2411.07184.

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

Yunhan Yang, Yuan-Chen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yan-Pei Cao, and Xihui Liu. Holopart: Generative 3d part amodal segmentation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2VsBJwefDC.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2308.06721.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. In *Computer Vision – ECCV 2024*, Cham, 2024. Springer Nature Switzerland. doi: 10.1007/978-3-031-73397-0_10.

Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6796–6807, June 2024.

Jingyang Zhang, Shiwei Li, Yuanxun Lu, Tian Fang, David Neil McKinnon, Yanghai Tsin, Long Quan, and Yao Yao. Jointnet: Extending text-to-image diffusion for dense distribution modeling. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kv5xE1p3jz.

Longwen Zhang, Qixuan Zhang, Haoran Jiang, Yinuo Bai, Wei Yang, Lan Xu, and Jingyi Yu. BANG: Dividing 3D assets via generative exploded dynamics. *ACM Transactions on Graphics*, 44(4), 2025. doi: 10.1145/3730840. URL https://doi.org/10.1145/3730840.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847, October 2023.

Yuchen Zhou, Jiayuan Gu, Tung Yen Chiang, Fanbo Xiang, and Hao Su. Point-SAM: Promptable 3d segmentation model for point clouds. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yXCTDhZDh6.

Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. PartSAM: A scalable promptable part segmentation model trained on native 3d data. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=y8sZUQPYXC.

A Method Details

A.1 Background: Structured Latent Backbone

Jointly predicting a part and a remainder requires both outputs to encode geometry and material in the coordinate system of the complete object. We therefore use the structured latent representation of TRELLIS. 2 (Xiang et al., 2026) for every complete object, part, and remainder:

$$\boldsymbol{z} = (\boldsymbol{z}^v, \boldsymbol{z}^g, \boldsymbol{z}^m), \qquad s \in \{v, g, m\}, \tag{14}$$

where v, g, and m index the sparse-structure, geometry, and material stages. The sparse-structure latent determines an active O-Voxel support $\mathcal{O} \in \{0,1\}^{N^3}$ on an O-Voxel grid, the backbone's sparse voxel representation, at resolution N. The geometry stage generates features on this support, and the material stage generates features aligned with the geometry. The decoder $\mathcal{D}$ converts the three latent components into a textured mesh in a shared $[-1,1]^3$ object frame. Because parts and remainders use this same frame, they do not require independent normalization or registration before composition.

Each of the three TRELLIS. 2 stages is a rectified-flow generator. For a clean target $\boldsymbol{x}_0$, noise $\boldsymbol{\epsilon}$, and time $t \in [0,1]$, the forward path is

$$\boldsymbol{x}(t) = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}, \qquad \dot{\boldsymbol{x}}(t) = \boldsymbol{\epsilon} - \boldsymbol{x}_0. \tag{15}$$

A velocity network $\boldsymbol{v}_\theta$ is trained by conditional flow matching to predict this vector field,

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}} \left\| \boldsymbol{v}_\theta(\boldsymbol{x}(t), t) - (\boldsymbol{\epsilon} - \boldsymbol{x}_0) \right\|_2^2 . \tag{16}$$

SCULPT retains this stage order and flow parameterization, but changes the clean target from one latent to a coupled part–remainder pair and conditions the velocity network on the current remainder (Sec. 3).

A.2 Training-Sequence Construction

Part ordering. After jointly normalizing the complete mesh and all of its parts to the shared object frame, we sort the parts lexicographically by their centroids along the z–x–y axes, giving one deterministic extraction order per asset.

Mesh-level tuples. With the notation of Sec. 3.2, the ordered parts define one mesh-level training tuple per step and one sequence per asset,

$$\mathcal{T}_i^{\mathcal{M}} = (\mathcal{M}_{i-1}; \hat{\mathcal{M}}_i, \mathcal{M}_i), \qquad i = 1, \ldots, K, \tag{17}$$

$$\mathcal{S}(\mathcal{M}_0) = \{\mathcal{T}_i^{\mathcal{M}}\}_{i=1}^{K}, \tag{18}$$

where the final tuple has the empty remainder $\mathcal{M}_K = \varnothing$. Figure 4 visualizes such a sequence as it is reproduced at inference time.

Terminal-state representation. For the terminal empty mesh, the sparse-structure target is an all-empty occupancy grid, while the geometry and material targets are zero-length sparse tensors. The terminal state therefore uses the same stage-wise predictor interfaces as nonempty remainders, without a separate terminal output type. Terminal tuples are included during training so that these representations are observed by the predictors.

A.3 Stage-wise Prediction and Sparse Packing

One split runs the three stage-wise predictors in the sparse-structure, geometry, and material order of the backbone:

$$(\hat{\boldsymbol{z}}_i^v, \boldsymbol{z}_i^v) = P_\theta^v(\boldsymbol{z}_{i-1}^v, \mathcal{I}), \tag{19}$$

$$(\hat{\boldsymbol{z}}_i^g, \boldsymbol{z}_i^g) = P_\theta^g(\boldsymbol{z}_{i-1}^g, \mathcal{I}; \hat{\mathcal{O}}_i, \mathcal{O}_i), \tag{20}$$

$$(\hat{\boldsymbol{z}}_i^m, \boldsymbol{z}_i^m) = P_\theta^m(\boldsymbol{z}_{i-1}^m, \mathcal{I}; \hat{\boldsymbol{z}}_i^g, \boldsymbol{z}_i^g). \tag{21}$$

At the sparse-structure stage, the packed part–remainder target is represented as a two-channel dense N^3 tensor. At the geometry and material stages, the two outputs generally occupy different sparse

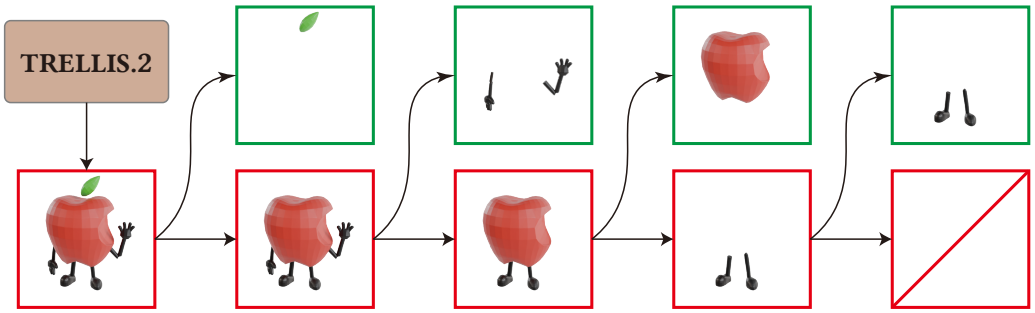

Figure 4: **A decomposition unfolds as a sequence of part–remainder transitions.** Starting from the TRELLIS. 2 whole-object state (red), each split emits one stored part (green) and carries only the updated remainder forward. In this example, the leaf, arms, body, and legs are removed successively; the rollout terminates when the remainder becomes empty.

supports. We therefore place their features on the union denoising grid, retain masks for the part and remainder destinations, and route the jointly denoised features back to their respective supports. This packing lets the two outputs share the denoising computation without requiring identical sparse coordinates.

A.4 Checkpoint and Control Initialization

The joint denoising branch is initialized from the corresponding public TRELLIS. 2 checkpoint. The residual projections that inject features from the remainder-control branch are zero-initialized, so the added control path makes no residual contribution at initialization. The joint and control branches are then optimized for the part–remainder prediction task.

A.5 Recurrent Inference

Algorithm 1 lists the complete recurrent inference procedure of Sec. 3.4.

Support invariant and stopping index. Clipping in Eq. 13 gives a non-expanding sequence of supports,

$$\hat{\mathcal{O}}_i \subseteq \mathcal{O}_{i-1}, \qquad \mathcal{O}_i \subseteq \mathcal{O}_{i-1} \subseteq \cdots \subseteq \mathcal{O}_0. \tag{22}$$

The rollout stops when the remainder support is empty or when it reaches the fixed cap $K_{\max}$. Equivalently, its final split index is

$$K = \min(\{i \le K_{\max} : |\mathcal{O}_i| = 0\} \cup \{K_{\max}\}). \tag{23}$$

Pipelining. The stage factorization also permits pipelined execution across recurrent splits. Once step i produces its remainder sparse latent, sparse-structure prediction for step $i+1$ can begin while step i continues through geometry and material generation. This schedule changes execution overlap but not the predicted states or their decoding order.

Algorithm 1 SCULPT recurrent inference. Each split predicts and clips an extracted part and updated remainder. The rollout stops at an empty remainder or after $K_{\max} = 24$ splits, retaining a nonempty capped remainder before decoding and assembly.

Require: image $\mathcal{I}$; whole-object generator G_ϕ; trained split predictors $P_\theta^v, P_\theta^g, P_\theta^m$; decoder $\mathcal{D}$; threshold τ_{occ}; cap $K_{\max} = 24$
Ensure: ordered output sequence $\mathcal{Z}_{\text{out}}$ and assembled asset

$\boldsymbol{z}_0 \leftarrow G_\phi(\mathcal{I})$ and extract $\mathcal{O}_0$ from $\boldsymbol{z}_0^v$
$\mathcal{Z}_{\text{out}} \leftarrow [\,]$
for $i = 1, \ldots, K_{\max}$ **do**
    $(\hat{\boldsymbol{z}}_i^v, \boldsymbol{z}_i^v) \leftarrow P_\theta^v(\boldsymbol{z}_{i-1}^v, \mathcal{I})$
    Threshold and mask both sparse latents, yielding the clipped supports in Eq. 13
    $(\hat{\boldsymbol{z}}_i^g, \boldsymbol{z}_i^g) \leftarrow P_\theta^g(\boldsymbol{z}_{i-1}^g, \mathcal{I}; \hat{\mathcal{O}}_i, \mathcal{O}_i)$
    $(\hat{\boldsymbol{z}}_i^m, \boldsymbol{z}_i^m) \leftarrow P_\theta^m(\boldsymbol{z}_{i-1}^m, \mathcal{I}; \hat{\boldsymbol{z}}_i^g, \boldsymbol{z}_i^g)$
    Assemble $\hat{\boldsymbol{z}}_i$ and $\boldsymbol{z}_i$ from their v, g, m components
    $\text{append}(\mathcal{Z}_{\text{out}}, \hat{\boldsymbol{z}}_i)$; $K \leftarrow i$
    if $\mathcal{O}_i = \varnothing$ **then**
        break
    end if
end for
if $\mathcal{O}_K \neq \varnothing$ **then**
    $\text{append}(\mathcal{Z}_{\text{out}}, \boldsymbol{z}_K)$ ▷ retain capped remainder
end if
return $\mathcal{Z}_{\text{out}}$ and $\bigcup_{\boldsymbol{z} \in \mathcal{Z}_{\text{out}}} \mathcal{D}(\boldsymbol{z})$

B Additional Details and Results

B.1 Notation

Table 3 summarizes the main symbols used in SCULPT. We use subscripts to index subtractive decomposition steps and superscripts $s \in \{v, g, m\}$ to index the sparse-structure, geometry, and material stages of the structured latent.

B.2 Dataset Details

Training-set filtering. From PartVerse-XL we remove assets with excessive component counts, extremely small components, missing PBR materials, or low-quality geometry and texture.

Evaluation-set exclusion. To prevent geometry leakage, we remove any PartVerse-XL training asset whose SHA-256 identifier matches a PartObjaverse evaluation mesh before filtering and tuple construction. SAMPart3D and PartVerse-XL are treated as independent annotation sources: their part vocabularies, boundary placements, and decomposition granularities may differ, so no PartVerse-XL labels are transferred to the evaluation benchmark and no SAMPart3D labels are used during training.

B.3 Implementation Details

We initialize the three stage-specific split predictors from the corresponding public TRELLIS. 2 checkpoints (Xiang et al., 2026); each predictor contains 30 transformer blocks. The stages are trained for 800K optimization steps in total with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-2}), a constant learning rate of 10^{-4}, and a per-GPU batch size of one on 32 GPUs. We set the composition-loss weight to $\lambda_{\text{comp}} = 0.1$. For the geometry and material stages, we omit splits whose extracted part contains fewer than 8 or more than 8192 tokens. This removes 1.5% of splits and 2.3% of assets from those two stages; the sparse-structure stage uses the full training set.

Table 3: **Notation used in SCULPT.** Hats denote extracted-part quantities; for mesh, latent, support, and occupancy pairs, the corresponding unhatted quantity denotes the remainder. Initial whole-object states are $\mathcal{M}_0$, $\boldsymbol{z}_0$, and $\mathcal{O}_0$.

Symbol	Meaning
$\mathcal{I}$	Conditioning image.
$\mathcal{M}_0$	Complete normalized training mesh in the shared object frame.
$\hat{\mathcal{M}}_i$	Mesh of the i-th extracted part under the deterministic part order.
$\mathcal{M}_i$	Remaining mesh after extracting the first i ordered parts; $\mathcal{M}_K = \varnothing$.
$\oplus$	Mesh-level union in the shared object frame.
$\boldsymbol{z}$	Generic structured 3D latent.
$\boldsymbol{z} = (\boldsymbol{z}^v, \boldsymbol{z}^g, \boldsymbol{z}^m)$	Sparse-structure, geometry, and material latent components.
$\boldsymbol{z}_0$	Initial complete-object latent before any part is extracted.
$\boldsymbol{z}_{i-1}$	Current remaining-object latent before the i-th split.
$\hat{\boldsymbol{z}}_i$	Extracted part latent predicted at step i.
$\boldsymbol{z}_i$	Updated remaining-object latent after extracting $\hat{\boldsymbol{z}}_i$.
K	Final split index, determined by empty support or the safety cap.
$K_{\max}$	Maximum number of recurrent splits; $K_{\max} = 24$.
$\mathcal{D}$	TRELLIS. 2 decoder used to decode each structured latent into a 3D asset.
P_θ	Joint split predictor, $(\hat{\boldsymbol{z}}_i, \boldsymbol{z}_i) = P_\theta(\boldsymbol{z}_{i-1}, \mathcal{I})$.
P_θ^s	Stage-wise split predictor for $s \in \{v, g, m\}$.
$\boldsymbol{y}_i^s$	Packed joint target at stage s: $\boldsymbol{y}_i^s = (\hat{\boldsymbol{z}}_i^s, \boldsymbol{z}_i^s)$.
$\boldsymbol{y}_i^s(t)$	Noisy rectified-flow sample, $(1-t)\boldsymbol{y}_i^s + t\boldsymbol{\epsilon}$.
$\boldsymbol{v}_\theta^s$	Stage-wise velocity predictor for the packed joint target.
$\mathcal{O}$	Generic hard sparse-structure support map on the O-Voxel grid.
$\mathcal{O}_0$	Complete-object support associated with $\boldsymbol{z}_0^v$.
$\hat{\mathcal{O}}_i$	Extracted part support associated with $\hat{\boldsymbol{z}}_i^v$.
$\mathcal{O}_i$	Remaining support associated with $\boldsymbol{z}_i^v$.
$\cup, \cap, \varnothing$	Elementwise support union, support intersection, and empty support.
$(\hat{\boldsymbol{p}}_i, \boldsymbol{p}_i)$	Predicted clean occupancy probabilities for part and remainder.
τ_{occ}	Fixed inference-time occupancy threshold.
N	O-Voxel grid resolution, with $\mathcal{O} \in \{0,1\}^{N^3}$.
$\boldsymbol{x}_0$	Clean target sample used only in the generic flow-matching path.
$\boldsymbol{x}(t)$	Interpolated flow sample, $(1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}$.
t	Flow timestep in $[0, 1]$.
$\boldsymbol{\epsilon}$	Random noise sample for rectified-flow training or sampling.
$\mathcal{L}_{\text{flow}}^s$	Stage-wise rectified-flow loss on $\boldsymbol{y}_i^s$.
$\mathcal{L}_{\text{comp}}$	BCE between the previous hard support and the differentiable probabilistic union.
λ_{comp}	Weight for the composition loss.

B.4 Semantic-Group-Level Geometry

At the semantic-group level, each Hungarian-matched prediction inherits the semantic label of its reference part, and predictions sharing a label are unioned before scoring. Table 4 reports the results for all methods of Table 1; SCULPT again obtains the lowest CD and the highest F1@.05, reducing CD by 8.5% relative to TRELLIS. 2+PartField.

C Extended Related Work

This section expands the summaries of Secs. 2.1 and 2.4.

Holistic 3D object generation. 3D object generation has seen rapid progress in recent years, with a variety of approaches for generating complete assets from text or images. Optimization-based methods (Poole et al., 2023; Yi et al., 2024; Lin et al., 2023; Shi et al., 2024) use 2D or multi-view diffusion priors (Ho et al., 2020; Song et al., 2021) to optimize 3D representations. Direct 3D diffusion models generate point clouds or implicit functions (Nichol et al., 2022; Jun & Nichol,

Table 4: **Semantic-group-level geometry on PartObjaverse.** Lower CD and higher F1 are better. Bold and underlined entries mark the best and second-best values. PF and HP abbreviate PartField and HoloPart.

Method	CD↓	F1@.1↑	F1@.05↑
Part123	0.0194	0.8105	0.6299
OmniPart	0.0135	0.8623	0.7060
TRELLIS+PF	0.0130	0.8734	0.7197
TRELLIS+SAM3D	0.0261	0.7483	0.5583
TRELLIS+PF+HP	0.0139	0.8510	0.6879
TRELLIS.2+PF	<u>0.0117</u>	**0.8903**	<u>0.7595</u>
TRELLIS.2+SAM3D	0.0505	0.5528	0.3486
TRELLIS.2+PF+HP	0.0441	0.6156	0.4106
SCULPT (ours)	**0.0107**	<u>0.8851</u>	**0.7614**

2023), while feed-forward and sparse-view reconstruction pipelines improve efficiency and view consistency (Long et al., 2024; Yang et al., 2024a; Hong et al., 2024; Tang et al., 2024; Xu et al., 2024). More recent systems, including TRELLIS (Xiang et al., 2025), TRELLIS. 2 (Xiang et al., 2026), Hunyuan3D 2.5 (Lai et al., 2025), UniLat3D (Wu et al., 2025), and TIGON (Cen et al., 2026), build on native 3D latent representations to support high-quality asset generation. Beyond single objects, generative pipelines have also been scaled to full scenes, where layouts or language arrange assets into coherent environments (Yang et al., 2024b;c).

Conditional control and joint generation. Conditional control studies how to steer pretrained generative models with additional signals while retaining their learned priors. ControlNet (Zhang et al., 2023) adds trainable branches to text-to-image diffusion models for spatial conditions such as edges, depth, or pose, and adapter-based methods such as T2I-Adapter (Mou et al., 2024) and IP-Adapter (Ye et al., 2023) inject spatial controls or image prompts with lightweight modules. JointNet (Zhang et al., 2024) and JointDiT (Kwon et al., 2025) further show that diffusion backbones can be extended to coupled outputs such as RGB-depth generation, while text-image conditioned 3D generation (Cen et al., 2026) explores multimodal control by combining visual exemplars with textual specifications. These works demonstrate the potential of conditioning and joint generation to steer pretrained models toward more complex outputs, but they do not directly address recurrent 3D part decomposition conditioned on an evolving remaining-object state.

D USE OF AI ASSISTANTS

In accordance with the ICLR 2027 policy on the use of AI in research, we disclose that large language model (LLM) based assistants were used solely to aid and polish the writing of this paper. Specifically, LLMs were used to improve grammar and wording, tighten and condense paragraphs to meet the page limit, suggest alternative phrasings of sentences drafted by the authors, and check the consistency of cross-references, notation, and LaTeX formatting.

LLMs were not used to conceive the research idea or the SCULPT formulation, to design the method or the experiments, to implement or run any code, to generate, clean, or annotate data, to produce any figure or quantitative result, or to analyze or interpret the results. All technical content, claims, and conclusions originate from the authors.

Every LLM-assisted edit was reviewed by the authors for factual accuracy, faithfulness to the intended meaning, and consistency with the reported experiments before being incorporated. The authors take full responsibility for the final content of this work.