

---

# Sparc3D: Sparse Representation and Construction for High-Resolution 3D Shapes Modeling

---

Zhihao Li<sup>1,2\*</sup>, Yufei Wang<sup>1</sup>, Heliang Zheng<sup>2,‡</sup>, Yihao Luo<sup>2,3,†</sup>, Bihan Wen<sup>1,†</sup>

<sup>1</sup>Department of EEE, Nanyang Technological University, Singapore

<sup>2</sup>Math Magic <sup>3</sup>Imperial-X, Imperial College London, UK

zhihao005@e.ntu.edu.sg, yufei001@e.ntu.edu.sg, zhengh11j@gmail.com

y.luo23@imperial.ac.uk, bihan.wen@ntu.edu.sg

<https://lizhihao6.github.io/Sparc3D>

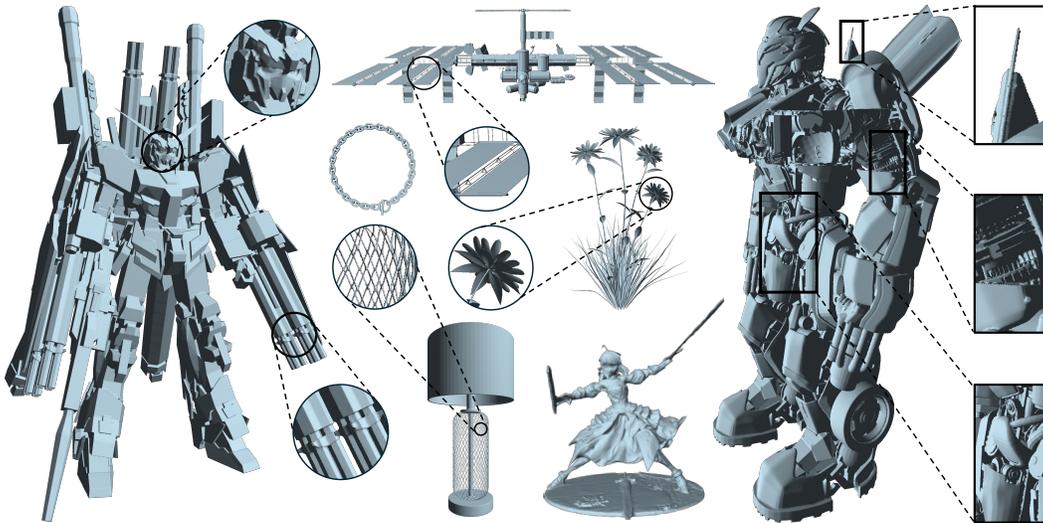


Figure 1: **Sparc3D Reconstruction Results.** Leveraging our sparse deformable marching cubes (**Sparcubes**) representation and sparse convolutional VAE (**Sparconv-VAE**), our method achieves state-of-the-art reconstruction quality on challenging 3D inputs. It robustly handles open surfaces (automatically closed into watertight meshes), recovers hidden interior structures, and faithfully reconstructs highly complex geometries (see zoom-in views, top to bottom). All outputs are fully watertight and 3D-printable, demonstrating the potential of our framework for high-resolution 3D mesh generation. *Best viewed with zoom-in.*

## Abstract

High-fidelity 3D object synthesis remains significantly more challenging than 2D image generation due to the unstructured nature of mesh data and the cubic complexity of dense volumetric grids. Existing two-stage pipelines—compressing meshes with a VAE (using either 2D or 3D supervision), followed by latent diffusion sampling—often suffer from severe detail loss caused by inefficient representations and modality mismatches introduced in VAE. We introduce **Sparc3D**, a unified framework that combines a sparse deformable marching cubes representation **Sparcubes**

---

\*This work was conducted during Zhihao Li’s research internship at Math Magic.

†Corresponding authors; ‡project lead.

with a novel encoder **Sparconv-VAE**. Sparcubes converts raw meshes into high-resolution ( $1024^3$ ) surfaces with arbitrary topology by scattering signed distance and deformation fields onto a sparse cube, allowing differentiable optimization. Sparconv-VAE is the first modality-consistent variational autoencoder built entirely upon sparse convolutional networks, enabling efficient and near-lossless 3D reconstruction suitable for high-resolution generative modeling through latent diffusion. Sparc3D achieves state-of-the-art reconstruction fidelity on challenging inputs, including open surfaces, disconnected components, and intricate geometry. It preserves fine-grained shape details, reduces training and inference cost, and integrates naturally with latent diffusion models for scalable, high-resolution 3D generation.

## 1 Introduction

Recent breakthroughs in 3D object generation [2, 15, 17, 24, 29, 32, 39] have enabled applications in virtual domains, such as AR/VR [3, 14, 16, 20] and robotics simulation [34, 35]—as well as in physical contexts like 3D printing [29]. Despite this progress, synthesizing high-fidelity 3D assets remains far more challenging than generating 2D imagery or text stemming from the inherently unstructured nature of 3D data and the cubic scaling of dense volumetric representations.

Drawing on the success of text-to-image diffusion models [25], many 3D generation pipelines [2, 15, 17, 32, 39] employ a two-stage process: a variational autoencoder (VAE) followed by latent diffusion. Most current VAEs employ either 3D supervision [2, 15, 17, 39]—typically via a global vector set representation—or 2D supervision [32]—commonly via a sparse-voxel representation—but both suffer from limited resolution and a modality mismatch between inputs and outputs.

3D supervised VAEs [2, 15, 17, 39] require watertight meshes for sampling supervision signals, yet most raw meshes are not watertight and must be remeshed. The common pipeline [2, 15, 39] shown in Fig. 2, first samples an Unsigned Distance Function (UDF) on a voxel grid, then approximates a Signed Distance Function (SDF) by subtracting two voxel sizes—halving the effective resolution and introducing errors that propagate through both VAE reconstruction and diffusion generation. After applying Marching Cubes [19] or Dual Marching Cubes [26], the mesh becomes double-layered, necessitating an additional step to retain only the largest connected component. This step inadvertently discards smaller yet crucial features, misaligning the conditioned input image from raw mesh with the reconstructed 3D shape during diffusion training.

Most recently, TRELLIS [32] demonstrated the possibility of training a 3D VAE using only 2D supervision. While it avoids degradation from mesh conversion, it still depends on dense volumetric grids for 2D projections, which limit resolution. Moreover, lacking any 3D topological constraints, the generated object’s interior geometry may be incorrect and its surfaces can remain open—a critical flaw for applications such as 3D printing.

All of these VAEs also contend with a fundamental modality gap: 3D supervised methods [2, 15, 17, 39] ingest surface points and normals but decode SDF values, while TRELLIS [32] encodes voxelized DINOv2 [23] features into SDF. Bridging this gap requires heavy attention mechanisms, which increase model complexity and risk of amplifying underlying inconsistencies.

In this work, we introduce **Sparcubes** (Sparse Deformable Marching Cubes), a fast, near-lossless pipeline for converting raw meshes into watertight surfaces. Our method begins by identifying a sparse set of activated voxels from the input mesh and performing a flood-fill to assign coarse signed labels. We then optimize grid-vertex deformations via gradient descent and refine them using a view-dependent 2D rendering loss. Sparcubes converts a raw mesh into a watertight  $1024^3$  grid in under 30 seconds—achieving a threefold speedup over prior watertight conversion methods [2, 15] without sacrificing fine details or small components.

Building on Sparcubes, we introduce **Sparconv-VAE**, a modality-consistent VAE composed of a sparse encoder and a self-pruning decoder. By eliminating the modality gap, Sparconv-VAE can employ a lightweight architecture without relying on heavy global attention mechanisms. Experimental results show that our VAE achieves state-of-the-art reconstruction performance and minimal training cost. Furthermore, it seamlessly integrates with existing latent diffusion pipelines such as TRELLIS [32], further enhancing the resolution of the generated 3D objects.

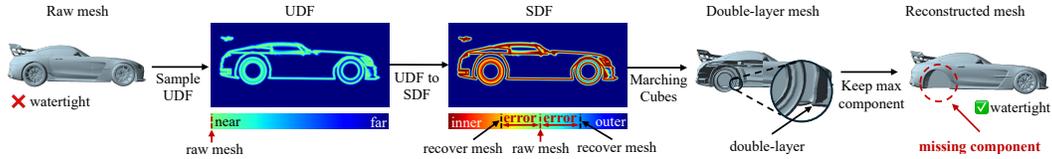


Figure 2: **Problems of the previous SDFs extraction pipeline.** The widely used SDFs extraction workflow [2, 15, 39] suffers from two critical failures: resolution degradation (show as **error**) and missing geometry (**circled** on the right). Converting UDF to SDF by subtracting two voxel sizes effectively halves the spatial resolution. Moreover, the SDF extraction yields a double-layer mesh, from which only the largest connected component is retained, inadvertently discarding smaller but important component. Together, these two deficiencies substantially limit the upper-bound performance of downstream VAEs and generation models. *Best viewed with zoom-in.*

Our main contributions can be summarized as:

- We propose **Sparcubes**, a fast, near-lossless remeshing algorithm that converts raw meshes into watertight surfaces at  $1024^3$  resolution in approximately 30 s, achieving a  $3\times$  speedup over prior methods without sacrificing any components.
- We introduce **Sparconv-VAE**, a modality-consistent variational autoencoder that employs a sparse convolutional encoder and a self-pruning decoder. By eliminating the input–output modality gap, our architecture achieves high computational efficiency and near-lossless reconstruction, without global attention.
- Experimental results show that, our **Sparc3D** framework, comprising Sparcubes and Sparconv-VAE, achieves state-of-the-art reconstruction fidelity, reduces training cost, and seamlessly integrates with current latent diffusion frameworks to enhance the resolution of generated 3D objects.

## 2 Related Work

### 2.1 3D Shape Representation and Generation

**Mesh and point cloud.** Triangle meshes and point clouds are the most common representations of 3D data. Triangle meshes, composed of vertices and triangular faces, offer precise modeling of surface detail and arbitrary topology. However, their irregular graph structure complicates learning, requiring neural networks to handle non-uniform neighborhoods, varying vertex counts, and the absence of a canonical ordering. To address this, recent methods [4, 5, 11, 28, 30] adopt autoregressive models to jointly generate geometry and connectivity, though these suffer from limited context length and slow sampling. In contrast, point clouds represent surfaces as unordered sets of 3D points, making them easy to sample from distributions [21, 31, 33], but difficult to convert directly into watertight surfaces due to the lack of explicit connectivity [9, 36].

**Isosurface.** Isosurface is a continuous surface representing a mesh boundary via a signed distance field (SDF). Most methods [13, 19, 26, 27] subdivide space into voxels, polygonize each cell, then stitch them into a mesh. Marching Cubes uses a fixed lookup table but can suffer topological ambiguities [19]. Dual Marching Cubes (DMC) fixes this by placing vertices on edges where the isosurface crosses and linking them via Dual Contouring, yielding watertight meshes [13, 26]. Both rely on a uniform cube size, limiting detail; FlexiCubes [27] deforms the grid and applies isosurface slope weights to adapt voxel sizes to local geometry, improving accuracy.

Many 3D generation methods adopt SDF-based supervision [2, 15, 17, 24, 32, 38, 39]. Techniques relying solely on 2D supervision [32] may implicitly learn an SDF, but often produce open surfaces or incorrect interiors due to the lack of volumetric constraints. In contrast, fully 3D-supervised approaches [2, 15, 17, 24, 38, 39] extract explicit SDFs from meshes with arbitrary topology, making accurate and adaptive SDF extraction a critical but challenging component for high-fidelity reconstruction.

## 2.2 3D Shape VAEs

**VecSet-based VAEs.** VecSet-based methods represent 3D shapes as sets of global latent vectors constructed from local surface features. 3DShape2VecSet [37] embeds sampled points and normals into a VecSet using a transformer and supervises decoding via surrounding SDF values. CLAY [38] scales the architecture to larger datasets and model sizes, while TripoSG [18] enhances expressiveness through mixture-of-experts (MoE) modules. Dora [2] and Hunyuan2 [39] improve sampling by prioritizing high-curvature regions. Despite these advances, all approaches face a modality mismatch: local point features are compressed into global latent vectors and decoded back to local fields, forcing the VAE to perform both feature abstraction and modality conversion, which increases reliance on attention and model complexity.

**Sparse voxel-based VAEs.** In contrast, sparse voxel-based VAEs preserve spatial structure by converting meshes into sparse voxel grids with feature vectors. XCube [24] replaces the global VecSet in 3DShape2VecSet [37] with voxel-aligned SDF and normal features, improving detail preservation. TRELLIS [32] enriches this representation with aggregated DINOv2 [23] features, enabling joint modeling of 3D geometry and texture. TripoSF [12] further scales the framework to high-resolution reconstructions (see Supplementary Material for details). Nonetheless, these methods still face the challenge of modality conversion—mapping point normals or DINOv2 descriptors to continuous SDF fields remains a key bottleneck.

## 3 Method

### 3.1 Preliminaries

**Distance fields.** A distance field is a scalar function  $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}$  that measures the distance to a surface. The unsigned distance function (UDF) encodes only magnitude, while the signed distance function (SDF) adds sign to distinguish interior and exterior:

$$\text{UDF}(\mathbf{x}, \mathcal{M}) = \min_{\mathbf{y} \in \mathcal{M}} \|\mathbf{x} - \mathbf{y}\|_2, \quad \text{SDF}(\mathbf{x}, \mathcal{M}) = \text{sign}(\mathbf{x}, \mathcal{M}) \cdot \text{UDF}(\mathbf{x}, \mathcal{M}), \quad (1)$$

where  $\text{sign}(\mathbf{x}, \mathcal{M}) \in \{-1, +1\}$  indicates inside/outside status. For non-watertight or non-manifold meshes, computing sign is non-trivial [1].

**Marching cubes and sparse variants.** The marching cubes algorithm [19] extracts an isosurface mesh from a volumetric field  $\Phi$  by interpolating surface positions across a voxel grid:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 \mid \Phi(\mathbf{x}) = 0\}. \quad (2)$$

Sparse variants [10] operate on narrow bands  $|\Phi(\mathbf{x})| < \epsilon$  to reduce memory usage. We further introduce a sparse cube grid  $(V, C)$ , where  $V$  is a set of sampled vertices and  $C$  contains 8-vertex cubes. Deformable and weighted dual variants, exemplified by FlexiCubes [27], extend this process by modeling the surface as a deformed version of the sparse cube grid. Specifically, each grid node  $n_i$  in the initial grid is displaced to a new position  $n_i + \Delta n_i$ , forming a refined grid  $(N + \Delta N, C, \Phi, W)$  that better conforms to the implicit surface, where the displacements  $\Delta n_i$  and per-node weights  $w_i \in W$  are learnable during optimization.

### 3.2 Sparcubes

Our method, **Sparcubes** (Sparse Deformable Marching Cubes), reconstructs watertight and geometrically accurate surfaces from arbitrary input meshes through sparse volumetric sampling, coarse-to-fine sign estimation, and deformable refinement. Unlike dense voxel methods, Sparcubes represents geometry using a sparse set of voxel cubes, where each cube vertex carries a signed distance value. This representation enables efficient computation, memory scalability, and supports downstream surface extraction or direct use in learning-based pipelines.

As shown in Fig. 3, the core pipeline consists of the following steps:

**Step 1: Active voxel extraction and UDF computation.** We begin by identifying a sparse set of *active voxels* within a narrow band around the input surface. These are voxels whose corner vertices

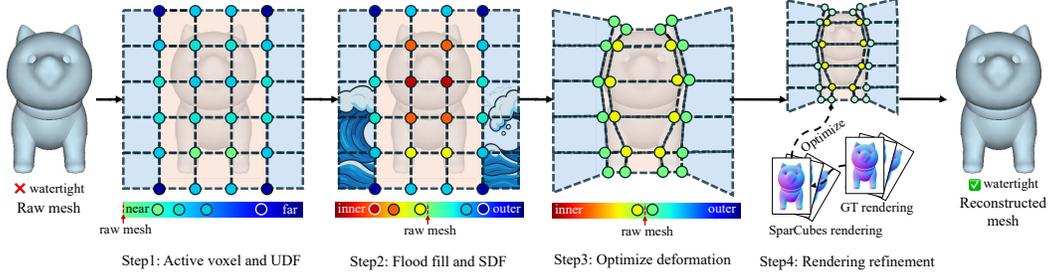


Figure 3: Illustration of our Sparcubes reconstruction pipeline for converting a raw mesh into a watertight mesh.

lie within a threshold distance  $\epsilon$  from the mesh  $\mathcal{M}$ . For each corner vertex  $\mathbf{x} \in \mathbb{R}^3$ , we compute the unsigned distance to the surface:

$$\text{UDF}(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{M}} \|\mathbf{x} - \mathbf{y}\|_2. \quad (3)$$

This yields a sparse volumetric grid  $\Phi$ , with distance values concentrated near the surface geometry, suitable for efficient storage and processing.

**Step 2: Flood fill for coarse sign labeling.** To convert the unsigned field into a signed distance function (SDF), we apply a volumetric flood fill algorithm [22] starting from known exterior regions (e.g., corners of the bounding box). This produces a binary occupancy label  $T(\mathbf{x}) \in \{0, 1\}$ , indicating whether point  $\mathbf{x}$  is inside or outside the shape. We then construct the coarse signed distance field as:

$$\text{SDF}(\mathbf{x}) = (1 - 2T(\mathbf{x})) \cdot \text{UDF}(\mathbf{x}), \quad (4)$$

which gives a consistent sign assignment under simple labeling and forms the basis for further refinement.

**Step 3: Gradient-based deformation optimization.** Instead of explicitly refining a globally accurate SDF, we directly optimize the geometry of the sparse cube structure to better conform to the underlying surface. Given an initial volumetric representation  $(V, C, \Phi_v)$ —where  $V$  denotes the set of sparse cube corner vertices,  $C$  is the set of active cubes, and  $\Phi$  is the signed distance field defined at each vertex—we perform a geometric deformation to obtain  $(V + \Delta V, C, \Phi_v)$ . This results in a geometry-aware sparse SDF volume that more accurately approximates the zero level set of the implicit surface. Notably, for points where  $\Phi(x) > 0$ , the SDF values are often only coarse approximations, particularly in regions far from the observed surface or near topological ambiguities. These regions may exhibit significant errors due to poor connectivity, occlusions, or non-watertight input geometry. As such, rather than refining  $\Phi$  globally, we optimize the vertex positions  $\Delta V$  to implicitly correct the spatial alignment of the zero level set. To improve the accuracy of sign estimation and geometric alignment, we displace each vertex slightly along the unsigned distance field gradient:

$$\mathbf{x}' = \mathbf{x} - \eta \cdot \nabla \text{UDF}(\mathbf{x}). \quad (5)$$

This heuristic captures local curvature and topological cues that are not easily recovered through purely topological methods such as flood fill. It also allows us to estimate sign information in regions with ambiguous connectivity, such as thin shells or open surfaces. The final data structure is a sparse cube grid with SDF values on each corner  $(V, C, \Phi_v, \Delta V)$ , denoted as **Sparcubes**.

**Step 4: Rendering-based refinement.** Sparcubes supports differentiable mesh extraction, enabling further end-to-end refinement with perceptual signals. When multi-view images, silhouettes, or depth maps are available, we optionally introduce a differentiable rendering loss to further enhance visual realism and geometric alignment. Given a reconstructed mesh  $\mathcal{M}_r$  extracted from the deformed Sparcubes, we compute a multi-term rendering loss:

$$\mathcal{L}_{\text{render}} = \|\mathcal{R}^D(\mathcal{M}_r) - \mathcal{I}_{\text{obs}}^D\|_2^2 + \|\mathcal{R}^N(\mathcal{M}_r) - \mathcal{I}_{\text{obs}}^N\|_2^2, \quad (6)$$

where  $\mathcal{R}^D(\mathcal{M}_r)$  denotes the rendered depth image of the mesh under known camera parameters, and  $\mathcal{R}^N(\mathcal{M}_r)$  is the corresponding rendered normal map. The terms  $\mathcal{I}_{\text{obs}}^D$  and  $\mathcal{I}_{\text{obs}}^N$  are the observed depth image and the ground truth normal map derived from the input or canonical mesh. Leveraging our voxel-based data structure, we can easily identify visible voxels and render exclusively within those regions, greatly reducing computational cost.

### 3.3 Sparconv-VAE

Building on our Sparcubes representation, we develop **Sparconv-VAE**, a sparse convolution-based variational autoencoder without high-consuming global attentions, which directly compresses the Sparcubes parameters  $\{\phi \in \Phi_v, \delta \in \Delta V\}$  into a sparse latent feature  $\mathbf{z}$  and decodes back to the same format without any modality conversion.

**Architecture and Loss Function.** Our encoder is a cascade of sparse residual convolutional blocks that progressively downsample the input features. At the coarsest resolution, a lightweight local attention module aggregates neighborhood information. The decoder mirrors this process, interleaving sparse residual convolutions with self-pruning upsample blocks to restore the original resolution and predict the Sparcubes parameters  $\{\hat{\phi}, \hat{\delta}\}$ . Each self-pruning block first predicts the occupancy of the subdivided voxel occupancy mask  $\mathbf{o}$ , supervised by  $\mathcal{L}_{\text{occ}} = \text{BCE}(\hat{\mathbf{o}}, \mathbf{o})$ , then applies a learned upsampling to refine the voxel features. Because  $\phi$  is sign-sensitive (inside vs. outside), we split its prediction into a sign branch and a magnitude branch. The sign branch predicts  $\text{sign}(\hat{\phi})$  under  $\mathcal{L}_{\phi_{\text{sign}}} = \text{BCE}(\text{sign}(\hat{\phi}), \text{sign}(\phi))$ , while the magnitude branch regresses  $\hat{\phi}$  with  $\mathcal{L}_{\phi_{\text{mag}}} = \|\hat{\phi}, \phi\|_2$ . The deformation vectors are optimized via  $\mathcal{L}_{\delta} = \|\hat{\delta}, \delta\|_2$ . Finally, we regularize the latent distribution using the VAE’s Kullback–Leibler divergence  $\mathcal{L}_{\text{KL}} = \text{KL}(q(\mathbf{z}|\delta, \phi)||p(\mathbf{z}))$ , yielding a single cohesive training objective that jointly minimizes the occupancy, sign, magnitude, deformation and KL divergence losses:

$$\mathcal{L} = \lambda_{\text{occ}} \mathcal{L}_{\text{occ}} + \lambda_{\text{sign}} \mathcal{L}_{\phi_{\text{sign}}} + \lambda_{\text{mag}} \mathcal{L}_{\phi_{\text{mag}}} + \lambda_{\delta} \mathcal{L}_{\delta} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}. \tag{7}$$

A detailed description of the module design and the choice of  $\lambda$  are in the Supplementary Material.

**Hole filling.** Although predicted occupancy may be imperfect and introduce small holes, our inherently watertight Sparcubes representation allows for straightforward hole detection and filling. We first identify boundary half-edges. For each face  $\mathbf{f} = \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2\}$ , we emit the directed edges  $(\mathbf{v}_0 \rightarrow \mathbf{v}_1), (\mathbf{v}_1 \rightarrow \mathbf{v}_2)$ , and  $(\mathbf{v}_2 \rightarrow \mathbf{v}_0)$ . By sorting each pair of vertices into undirected edges and counting occurrences, edges whose undirected counterpart appears only once are marked as boundary edges. We build an outgoing-edge map keyed by source vertex, and then recover closed boundary loops by walking each edge until returning to its start. To triangulate each boundary loop  $\mathcal{C} = \{\mathbf{v}_i\}_{i=1}^n$ , we follow a classic ear-filling pipeline: compute a geometric score at every vertex, fill the “best” ear, and repeat until all open small boundaries vanish. Specifically, the score on each pending filled angle  $A_i$  is defined by

$$A_i = \text{atan2}(\|\mathbf{d}_{i-1 \rightarrow i} \times \mathbf{d}_{i \rightarrow i+1}\|_2, -\mathbf{d}_{i-1 \rightarrow i} \cdot \mathbf{d}_{i \rightarrow i+1}). \tag{8}$$

In each iteration, we select the vertex with the smallest  $A_i$  (*i.e.*, the sharpest convex ear), form the triangle  $(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$ , and update the boundary. Merging all new triangles with the original face set closes every small hole.

## 4 Experiments

### 4.1 Experiment Settings

**Implementation details.** We implement all Sparcubes as custom CUDA kernels. Following TRELLIS [32], we train both the Sparconv-VAE and its latent flow model on 500 K high-quality assets from Objaverse [8] and Objaverse-XL [7]. The VAE runs on 32 A100 GPUs (batch size 32) with AdamW (initial LR  $1 \times 10^{-4}$ ) for two days. We then fine-tune the TRELLIS latent flow model on our VAE latents using 64 A100 GPUs (batch size 64) for ten days. At inference, we sample with a classifier-free guidance scale of 3.5 over 25 steps, matching TRELLIS settings.

**Dataset.** Following Dora [2], we curated a VAE test set by selecting the most challenging examples from the ABO [6] and Objaverse [8] datasets—specifically those exhibiting occluded components, intricate geometric details, and open surfaces. To avoid any overlap with training data used in prior work, we additionally assembled a “Wild” dataset with multiple components from online sources that is disjoint from both ABO and Objaverse. For generation, we also benchmarked our method against TRELLIS [32] using the wild dataset.

**Compared methods.** We compare our Sparconv-VAE with the previous state-of-the-art vaes, including TRELLIS [32], Craftsman [15], Dora [2] and XCubces [24]. Because our diffusion architecture and

Table 1: **Quantitative comparison of watertight remeshing across the ABO [6], Objaverse [8], and In-the-Wild datasets.** Chamfer Distance (CD,  $\times 10^4$ ), Absolute Normal Consistency (ANC,  $\times 10^2$ ) and F1 score (F1,  $\times 10^2$ ) are reported.

Method	ABO [6]			Objaverse [8]			Wild		
	CD ↓	ANC ↑	F1 ↑	CD ↓	ANC ↑	F1 ↑	CD ↓	ANC ↑	F1 ↑
Dora-wt-512 [2]	1.16	76.94	83.18	4.25	75.77	61.35	67.2	78.51	64.99
Dora-wt-1024 [2]	1.07	76.94	84.56	4.35	75.04	63.84	63.7	78.77	65.90
Ours-wt-512	1.01	77.75	85.21	3.09	75.35	64.81	0.47	88.58	96.95
Ours-wt-1024	1.00	77.66	85.39	3.01	74.98	65.65	0.46	88.55	97.06

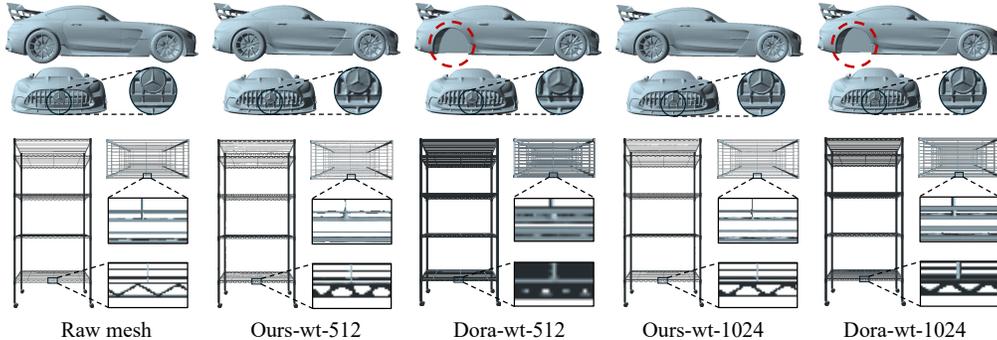


Figure 4: **Qualitative comparison of watertight remeshing pipelines.** We evaluate our Sparcubes remeshing pipeline against previous widely used one [2, 15, 39], *i.e.*, Dora-wt [2], at voxel resolutions of 512 and 1024. Compared with the previous method, our Sparcubes preserves crucial components (e.g., the car wheel) and recovers finer geometric details (e.g., the shelving frame). Our wt-512 result even outperforms the wt-1024 remeshed by Dora-wt [2]. *Best viewed with zoom-in.*

model size match those of TRELIS [32], we evaluate our generation results against it to ensure a fair comparison.

## 4.2 Comparison Results

**Watertight remeshing results.** We evaluate both our watertight remeshing (serving as VAE ground truth) on the ABO [6], Objaverse [8], and Wild datasets, using Chamfer distance (CD), Absolute Normal Consistency (ANC), and F1 score (F1) as metrics. As Table 1 shows, our Sparcubes consistently outperforms prior pipelines [2, 15, 39] (reported under “Dora-wt” [2] for brevity) across all datasets and metrics. Remarkably, our wt-512 remeshed outputs even exceed the quality of the wt-1024 results produced by previous methods. Fig. 4 presents qualitative comparisons: our approach faithfully preserves critical components (*e.g.*, the car wheel) and recovers fine geometric details (*e.g.*, shelving frames).

**VAE reconstruction results.** We further assess our Sparconv-VAE reconstruction against TRELIS [32], Craftsman [15], Dora [2], and XCubes [24] in Table 2. Across the majority of datasets and metrics, our Sparconv-VAE outperforms these prior methods. Qualitative results in Fig. 5 illustrate that our VAE faithfully reconstructs complex shapes with fine details (all columns), converts open surfaces into double-layered watertight meshes (columns 1, 4, and 6), and reveals invisible hidden internal structures (column 6).

**Generation results.** We also validate the effectiveness of our Sparconv-VAE for generation by fine-tuning the TRELIS [32] pretrained model. Under the same diffusion architecture and model size (see Fig. 6), our approach synthesizes watertight 3D shapes with exceptional fidelity and rich detail—capturing, for example, the sharp ridges of pavilion eaves, the subtle facial features of human figures, and the intricate structural elements of robots.

Table 2: **Quantitative comparison of VAE reconstruction across the ABO [6], Objaverse [8], and In-the-Wild datasets.** Chamfer Distance (CD,  $\times 10^4$ ), Absolute Normal Consistency (ANC,  $\times 10^2$ ) and F1 score (F1,  $\times 10^2$ ) are reported.

Method	ABO [6]			Objaverse [8]			Wild		
	CD ↓	ANC ↑	F1 ↑	CD ↓	ANC ↑	F1 ↑	CD ↓	ANC ↑	F1 ↑
TRELLIS [32]	1.32	75.48	80.59	4.29	74.34	59.27	0.70	85.60	94.04
Craftsman [15]	1.51	77.46	77.47	2.53	77.37	55.28	0.89	87.81	92.28
Dora [2]	1.45	77.21	78.54	4.85	77.19	54.37	68.2	78.79	62.07
XCubes [24]	1.42	65.45	77.57	3.67	61.81	51.65	2.02	62.21	73.74
Ours-512	1.01	78.09	85.33	3.09	75.59	64.92	0.47	88.74	96.97
Ours-1024	1.00	77.69	85.41	3.00	75.10	65.75	0.46	88.70	97.12

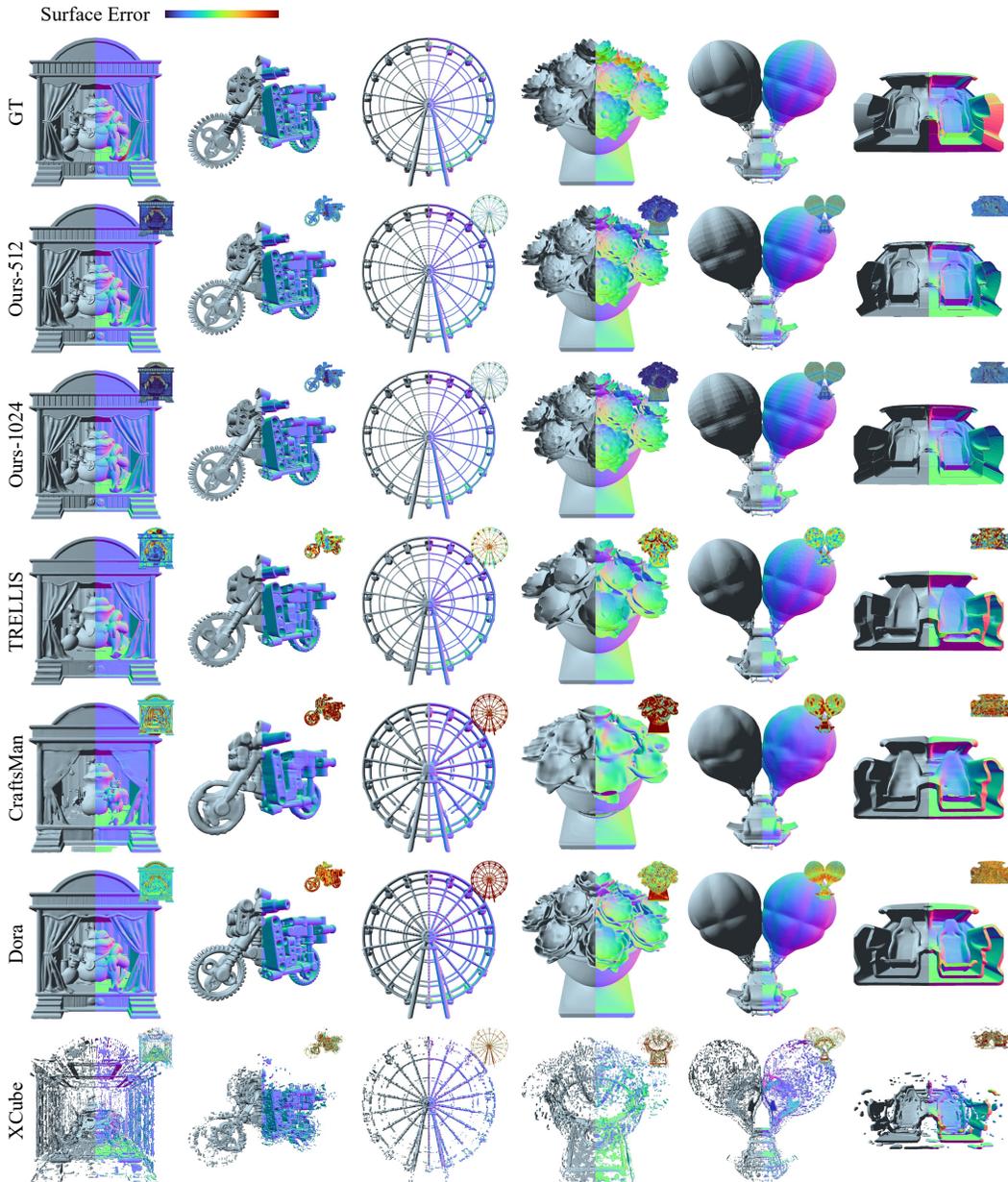


Figure 5: **Qualitative comparison of VAE reconstructions.** Our Sparconv-VAE demonstrates superior performance in reconstructing complex geometries, converting open surfaces into double-layered watertight meshes, and revealing invisible internal structures. *Best viewed with zoom-in.*



Figure 6: **Qualitative comparison of single-image-to-3D generation.** Under the same architecture and model size [32], the generator trained with our Sparconv-VAE yields more detailed reconstructions than TRELLIS [32]. *Best viewed with zoom-in.*

Table 3: **Ablation study on Steps 3 and 4 in the preprocessing.** Chamfer Distance (CD,  $\times 10^4$ ), Absolute Normal Consistency (ANC,  $\times 10^2$ ), and F1 score (F1,  $\times 10^2$ ) are reported.

w/ Step 3	w/ Step 4	CD ↓	ANC ↑	F1 ↑
✓		0.97	77.34	88.13
	✓	0.95	77.81	88.73
✓	✓	0.92	78.08	89.52

### 4.3 Ablation Studies

**Conversion cost.** Compared to existing remeshing methods [2, 15, 39], our Sparcubes achieves substantial speedups: at a 512-voxel resolution, conversion takes only around 15 s—half than [2, 15, 39]—and at 1024-voxel resolution, it completes in around 30 s versus around 90 s by [2, 15, 39]. Moreover, by eliminating modality conversion in our VAE design, we avoid the additional SDF resampling step, which in earlier pipelines adds roughly 20 s at 512 resolution and about 70 s at 1024 resolution [2, 15, 39]. Detailed performance comparisons can be found in the Supplementary Material.

**Training cost.** Thanks to our modality-consistent design, Sparconv-VAE converges less than two days—about four times faster than previous methods, *i.e.*, sparse voxel-based TRELLIS [32] and vecset-based approaches [2, 15] each require roughly seven days to train.

**VAE with 2D rendering supervision.** We also investigate the effect of incorporating 2D rendering losses into our VAE by using the mask, depth, and normal rendering objectives. We find that adding 2D rendering supervision yields negligible improvement for our Sparconv-VAE. This observation concurs with Dora [2], where extra 2D rendering losses were likewise deemed unnecessary for a 3D-supervised VAEs. We attribute this to the fact that sufficiently dense 2D renderings encode essentially the same information as the underlying 3D geometry—each view being a projection of the same 3D shape.

**Ablation of each in the preprocessing.** We perform an ablation study on Steps 3 and 4 to isolate the contributions of deformation optimization and rendering-based refinement:

In addition to consistent quantitative gains across all metrics, enabling both steps leads to substantial visual improvements—often beyond what the metrics alone can capture in terms of perceptual quality.

## 5 Conclusion

We introduce Sparc3D, a unified framework that tackles two longstanding bottlenecks in 3D generation pipelines: topology-preserving remeshing and modality-consistent latent encoding. At its heart, Sparcubes transforms raw, non-watertight meshes into fully watertight surfaces at high resolution—retaining fine details and small components. Building on this, Sparconv-VAE, a sparse-convolutional variational autoencoder with a self-pruning decoder, directly compresses and reconstructs our sparse representation without resorting to heavyweight attention, achieving state-of-the-art reconstruction fidelity and faster convergence. When coupled with latent diffusion (e.g., TRELIS), Sparc3D elevates generation resolution for downstream 3D asset synthesis. Together, these contributions establish a robust, scalable foundation for high-fidelity 3D generation in both virtual (AR/VR, robotics simulation) and physical (3D printing) domains.

**Limitations.** While our Sparcubes remeshing algorithm excels at preserving fine geometry and exterior components, it shares several drawbacks common to prior methods. First, it does not retain any original texture information. Second, when applied to fully closed meshes with internal structures, hidden elements will be discarded during the remeshing process.

## References

- [1] M. Atzmon and Y. Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2565–2574, 2020. 4
- [2] R. Chen, J. Zhang, Y. Liang, G. Luo, W. Li, J. Liu, X. Li, X. Long, J. Feng, and P. Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. *arXiv preprint arXiv:2412.17808*, 2024. 2, 3, 4, 6, 7, 8, 9
- [3] T. Chen, C. Ding, S. Zhang, C. Yu, Y. Zang, Z. Li, S. Peng, and L. Sun. Rapid 3d model generation with intuitive 3d input. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12554–12564, 2024. 2
- [4] Y. Chen, T. He, D. Huang, W. Ye, S. Chen, J. Tang, X. Chen, Z. Cai, L. Yang, G. Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 3
- [5] Y. Chen, Y. Wang, Y. Luo, Z. Wang, Z. Chen, J. Zhu, C. Zhang, and G. Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555*, 2024. 3
- [6] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Y. Vicente, T. Dideriksen, H. 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*, pages 21126–21136, 2022. 6, 7, 8
- [7] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023. 6
- [8] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023. 6, 7, 8
- [9] E. Gómez-Déniz. Another generalization of the geometric distribution. *Test*, 19:399–415, 2010. 3
- [10] R. Hanocka, A. Hertz, N. Fish, R. Giryas, S. Fleishman, and D. Cohen-Or. Point2mesh: A self-prior for deformable meshes. In *ACM Transactions on Graphics (TOG)*, volume 39, 2020. 4
- [11] Z. Hao, D. W. Romero, T.-Y. Lin, and M.-Y. Liu. Meshtron: High-fidelity, artist-like 3d mesh generation at scale. *arXiv preprint arXiv:2412.09548*, 2024. 3
- [12] X. He, Z.-X. Zou, C.-H. Chen, Y.-C. Guo, D. Liang, C. Yuan, W. Ouyang, Y.-P. Cao, and Y. Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. *arXiv preprint arXiv:2503.21732*, 2025. 4

- [13] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 339–346, 2002. 3
- [14] W. Li. Synthesizing 3d vr sketch using generative adversarial neural network. In *Proceedings of the 2023 7th International Conference on Big Data and Internet of Things*, pages 122–128, 2023. 2
- [15] W. Li, J. Liu, R. Chen, Y. Liang, X. Chen, P. Tan, and X. Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024. 2, 3, 6, 7, 8, 9
- [16] X. Li, Q. Zhang, D. Kang, W. Cheng, Y. Gao, J. Zhang, Z. Liang, J. Liao, Y.-P. Cao, and Y. Shan. Advances in 3d generation: A survey. *arXiv preprint arXiv:2401.17807*, 2024. 2
- [17] Y. Li, Z.-X. Zou, Z. Liu, D. Wang, Y. Liang, Z. Yu, X. Liu, Y.-C. Guo, D. Liang, W. Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025. 2, 3
- [18] Y. Li, Z.-X. Zou, Z. Liu, D. Wang, Y. Liang, Z. Yu, X. Liu, Y.-C. Guo, D. Liang, W. Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025. 4
- [19] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87*, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. 2, 3, 4
- [20] L. Luo, P. N. Chowdhury, T. Xiang, Y.-Z. Song, and Y. Gryaditskaya. 3d vr sketch guided 3d shape prototyping and exploration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9267–9276, 2023. 2
- [21] S. Luo and W. Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2837–2845, 2021. 3
- [22] S. Mauch. *Efficient algorithms for solving static Hamilton-Jacobi equations*. PhD thesis, California Institute of Technology, 2000. 5
- [23] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 4
- [24] X. Ren, J. Huang, X. Zeng, K. Museth, S. Fidler, and F. Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4209–4219, 2024. 2, 3, 4, 6, 7, 8
- [25] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2
- [26] S. Schaefer and J. Warren. Dual marching cubes: Primal contouring of dual grids. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, pages 70–76. IEEE, 2004. 2, 3
- [27] T. Shen, J. Munkberg, J. Hasselgren, K. Yin, Z. Wang, W. Chen, Z. Gojcic, S. Fidler, N. Sharp, and J. Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023. 3, 4
- [28] Y. Siddiqui, A. Alliegro, A. Artemov, T. Tommasi, D. Sirigatti, V. Rosov, A. Dai, and M. Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19615–19625, 2024. 3
- [29] T. Wang, B. Zhang, T. Zhang, S. Gu, J. Bao, T. Baltrusaitis, J. Shen, D. Chen, F. Wen, Q. Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4563–4573, 2023. 2
- [30] Z. Wang, J. Lorraine, Y. Wang, H. Su, J. Zhu, S. Fidler, and X. Zeng. Llama-mesh: Unifying 3d mesh generation with language models. *arXiv preprint arXiv:2411.09595*, 2024. 3
- [31] L. Wu, D. Wang, C. Gong, X. Liu, Y. Xiong, R. Ranjan, R. Krishnamoorthi, V. Chandra, and Q. Liu. Fast point cloud generation with straight flows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9445–9454, 2023. 3

- [32] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 2, 3, 4, 6, 7, 8, 9
- [33] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 3
- [34] Y. Yang, B. Jia, P. Zhi, and S. Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16262–16272, 2024. 2
- [35] Y. Yang, F.-Y. Sun, L. Weihs, E. VanderBilt, A. Herrasti, W. Han, J. Wu, N. Haber, R. Krishna, L. Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237, 2024. 2
- [36] L. Yushi, S. Zhou, Z. Lyu, F. Hong, S. Yang, B. Dai, X. Pan, and C. C. Loy. Gaussiananything: Interactive point cloud flow matching for 3d generation. In *The Thirteenth International Conference on Learning Representations*, 2025. 3
- [37] B. Zhang, J. Tang, M. Niessner, and P. Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023. 4
- [38] L. Zhang, Z. Wang, Q. Zhang, Q. Qiu, A. Pang, H. Jiang, W. Yang, L. Xu, and J. Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024. 3, 4
- [39] Z. Zhao, Z. Lai, Q. Lin, Y. Zhao, H. Liu, S. Yang, Y. Feng, M. Yang, S. Zhang, X. Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 2, 3, 4, 7, 9

## 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 abstract and Introduction (Sec. 1) clearly state our three main contributions, and integration with latent diffusion—and these match the methods and results shown in Sections 3 and 4.

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: We include a "Limitations" paragraph on last section discussing issues such as loss of original texture information and the inability to recover hidden internal structures in fully closed meshes.

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: [No]

Justification: This work presents algorithmic methods and empirical validation without formal theorems or proofs, so full assumptions and proofs are not applicable.

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: Section 4.1 (“Experiment Settings”) provides full details on datasets (Objaverse, ABO, Wild), hardware, training schedules, batch sizes, optimizer settings, and convergence times.

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: We will release the code once the paper has been accepted and has successfully passed internal review.

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: We specify data splits, hyperparameters, batch sizes, GPU counts, and guidance scales in Section 4.1.

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: We omit error bars, confidence intervals, and statistical tests due to the prohibitive cost of training.

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: We detail compute setup, number of GPUs used, batch sizes—and report wall-clock training times in Section 4.1.

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 uses only synthetic 3D assets and standard algorithms, involves no human subjects or sensitive data, and adheres to NeurIPS ethical guidelines.

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: [No]

Justification: We focus on foundational 3D generation methods and do not include a discussion of societal impacts or potential misuse.

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 poses no high-risk misuse scenarios requiring special release safeguards for data or models.

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: [No]

Justification: We cite all datasets used in the paper.

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](https://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: No new datasets or pretrained models are released alongside this work.

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 work does not involve any human subjects or crowdsourced data.

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: There is no human-subjects research, so IRB approval is not applicable.

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: Large language models are not used as part of the core methods; any language tools used were for writing only.

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.