Pygmalion: Bridging Reconstruction and Generation in Sparse Voxel-based 3D Modeling

Guan Luo[1,2]* **Jing Lin**[2]* **Xuanyu Yi**[2]* **Jiahang Liu**[2] **Song-Hai Zhang**[1]† **Jianfeng Zhang**[2]†
[1] Tsinghua University [2] ByteDance Seed



Figure 1: All 3D assets presented here are generated by Pygmalion from a single image.

Abstract

Recent advances in sparse voxel-based VAEs have demonstrated remarkable capabilities in high-fidelity 3D autoencoding, yet their generative counterparts consistently lag behind their reconstruction performance. A fundamental cause of this gap lies in the *brittle output representation* in sparse voxel decoders, which parameterize geometry through discrete topological decisions, such as intersection flags or occupancy signs. Small perturbations in these parameters, which are unavoidable under diffusion sampling, can be amplified into abrupt topological ruptures and grid-like artifacts. We argue that a generation-friendly representation should ensure that such perturbations induce only smooth geometric transitions. To this end, we propose Pygmalion, a generation-friendly autoencoding framework that reformulates sparse voxel decoding as a coupled SDF parameterization. To retain explicit mesh-level supervision within this parameterization, which is non-trivial as the SDF jointly governs both topology and geometry, we introduce hinge-based sign correction, case-aware geometry supervision, and rendering-based refinement, achieving high-fidelity reconstruction while preserving generative robustness. Built upon this generation-friendly representation, Pygmalion scales consistently across DiT model sizes and voxel resolutions. Extensive experiments demonstrate state-of-the-art reconstruction fidelity and high-quality 3D generation with complete surfaces, sharp features, and fine geometric details. Our project page is available at https://pygmalionanonymous.github.io.

*Equal Contribution.
†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Comparison of representative sparse voxel-based methods across output representation, sharpness, smoothness, reconstruction, and geometric fidelity. ↑, –, and ↓ mean high, moderate, and low levels, respectively.

Method	Representation	Sharpness	Smoothness	Reconstruction	Generation
Trellis [59]	FlexiCubes	↓	✔	↓	Low-res, holes
SparseFlex [10]	FlexiCubes	↓	✔	–	Over-smooth
Direct3D-S2 [57]	MC	↓	✔	↓	Over-smooth
Sparc3D [22]	Def MC + Occ/UDF	–	✖	–	Grid-like artifacts
FaithC v1.5 [31]	DC + Intersection Flag	↑	✖	↑	N/A
Trellis2 [58]	DC + Intersection Flag	↑	✖	↑	Holes
TopoMesh [30]	FlexiCubes + Occ/UDF	↑	✖	↑	Grid-like artifacts
Ours	FlexiCubes	↑	✔	↑	High-quality 😊

1 Introduction

Deep generative models for 3D content creation have converged on a two-stage paradigm in which a VAE [15] compresses 3D shapes into a latent space and a DiT [36] generates within it. In this paradigm, the VAE's reconstruction fidelity sets a firm upper bound on generation quality. Among various designs, sparse voxel-based VAEs [10, 22, 30, 31, 58, 59] have emerged as the state of the art for high-fidelity reconstruction, leveraging the local inductive bias of voxel grids to faithfully capture fine geometric details that VecSet-based alternatives [3, 21, 49, 63, 64] tend to over-smooth. However, a notable gap persists between this reconstruction strength and DiT-driven generation: DiT-generated outputs are plagued by holes, surface noise, and grid-like artifacts (Fig. 5), falling far short of the reconstruction quality achieved by the same VAE. In fact, they even underperform well-designed VecSet-based DiTs [1, 50] in generation tasks.

A fundamental yet often overlooked cause of this gap lies in the *brittle output representation* of sparse voxel decoders. Recent sparse voxel-based VAEs [22, 30, 31, 58] rely on discrete signals for topology construction during decoding, such as intersection flags or occupancy signs, introducing hard decision boundaries into the parameter-to-geometry process. In the autoencoding regime, encoder-provided latents are precise enough to keep these discrete predictions on the safe side of such boundaries. During generation, however, DiT-sampled latents are inherently noisier and more ambiguous, pushing predictions toward or across decision boundaries and causing holes and grid-like artifacts. The straightforward remedy is to adopt a continuous representation, but doing so removes the explicit topology signal that explicit supervision relies on, making it harder to preserve sharp features and fine geometric details [30].

This tension gives rise to a smoothness–sharpness trade-off across prior methods. Trellis2 [58] and FaithC [31] determine topology via discrete intersection flags, where a single incorrect prediction removes an entire edge and creates holes. Sparc3D [22] and TopoMesh [30] adopt a decoupled formulation, combining occupancy signs with UDFs for vertex interpolation, where sign inconsistencies lead to unstable vertex positions and grid-like artifacts. In contrast, Trellis [59], SparseFlex [10], and Direct3D-S2 [57] adopt smoother parameterizations, improving generation robustness but at the cost of reconstruction sharpness due to the lack of explicit mesh-level supervision. Tab. 1 summarizes this trade-off, motivating a representation that achieves both.

We argue that such a representation must satisfy two key properties: **(i) Sharpness-aware supervised:** the representation must support explicit mesh-level supervision capable of preserving sharp features and fine geometric details. **(ii) Smoothly parameterized:** the mapping from predicted parameters to extracted geometry contains no hard thresholds or sign-based switches, so that small perturbations in the decoded parameters induce smooth geometric changes rather than abrupt topological failures. This property is crucial under diffusion sampling, where imprecise or stochastic latent predictions should lead to graceful geometric variation instead of structural collapse.

To meet both requirements simultaneously, we propose Pygmalion[3], a generation-friendly, sharpness-preserving autoencoding framework for sparse voxel-based 3D modeling. The key idea is to adopt FlexiCubes [42] with a coupled SDF parameterization, where SDF values directly drive vertex placement without hard thresholding, ensuring small perturbations induce only smooth geometric transitions. However, unlike decoupled formulations, the coupled SDF jointly governs topology and

[3]Named after the mythological sculptor whose ivory statue was brought to life, mirroring our goal of closing the gap between reconstruction fidelity and generation quality.

vertex placement, making explicit mesh-level supervision non-trivial. To resolve this, we introduce a **hinge-based sign correction loss** that decouples topology and geometry optimization at the loss level. Since incorrect topologies may persist in some voxels despite sign guidance, we further employ a **case-aware supervision scheme** that prevents misleading gradients from mismatched vertex correspondences, and **rendering-based refinement** as a complementary signal for residual incorrect regions. Together, they enable explicit mesh-level supervision within a fully smooth parameterization, preserving sharp features without sacrificing generative robustness. Built on this, Pygmalion scales consistently across DiT model sizes (0.6B to 4.5B) and resolutions (up to 1536^3).

In summary, our contributions are listed below:

- We identify the *brittle output representation* of existing sparse voxel-based VAEs as a fundamental cause of the reconstruction-generation gap and propose two principles for generation-friendly decoding: smooth parameterization and sharpness-aware supervision.
- We reformulate sparse voxel decoding as a coupled SDF in FlexiCubes, and introduce hinge-based sign correction, case-aware supervision, and rendering-based refinement to enable explicit mesh-level supervision without sacrificing smoothness.
- Pygmalion achieves state-of-the-art reconstruction fidelity and high-quality generation with complete surfaces, sharp features, and fine geometric details, with consistent quality gains across DiT model sizes and resolutions.

2 Related Work

Scalable 3D Generation. 3D generative modeling has evolved through multiple paradigms, each with distinct trade-offs among generation quality, efficiency, and scalability. Optimization-based methods [2, 23, 37, 38, 46, 48, 54] distill 2D diffusion priors [40] into implicit representations [14, 33] via Score Distillation Sampling (SDS), producing detailed assets but requiring costly per-shape optimization. Multi-view generation methods [25, 27, 43, 44, 52, 53], coupled with sparse-view reconstruction [11, 19, 24, 47, 55, 60], enable fast feed-forward inference but suffer from multi-view inconsistencies, leading to degraded geometric quality and generation stability. Native 3D generative methods [10, 20, 39, 49, 57, 58, 59, 63, 64, 65] have emerged as the dominant paradigm for scalable 3D generation by learning directly in the 3D latent space, offering a strong balance between generation quality and scalability. With advances in DiT distillation [41, 45] and decoding acceleration [17, 58], they further support fast inference for large-scale 3D content creation.

3D Shape VAEs. A key component of native 3D generation is the shape VAE that maps complex geometry into a structured latent space. Early shape autoencoders explore various representations, including point clouds [34, 62], triplanes [9, 56], and implicit fields [13], but generally struggle with scalability, generalization, and geometric fidelity. VecSet-based VAEs [3, 16, 20, 21, 49, 63, 64, 65] represent shapes as unordered sets of latent vectors, offering a flexible and scalable latent space for diffusion-based generation. However, their global tokenization makes it difficult to preserve fine local geometry, especially sharp features and high-frequency details. Sparse voxel-based VAEs [10, 22, 30, 31, 57, 58, 59, 61], which our work builds upon, have emerged as the state-of-the-art for high-resolution shape modeling. By leveraging sparse voxel grids, these methods achieve strong locality and efficient resolution scaling. Recent advances further improve decoding quality through geometric [22], feature-level [31, 58], or mesh-level [30] explicit supervision, establishing high-fidelity VAE backbones. However, these reconstruction-oriented designs do not translate into robust generation, a gap that motivates re-examination of the decoder output representation.

Isosurface Extraction. Extracting explicit mesh surfaces from volumetric representations is crucial in sparse voxel-based generation. Marching Cubes (MC) [28] interpolates vertices along cell edges based on SDF values, and is therefore limited in geometric expressiveness. Dual Contouring (DC) [12] places vertices freely within grid cells, enabling sharp feature reconstruction, but requires accurate gradients and lacks manifold guarantees. Recent methods [5, 31, 58] build upon DC representation but typically introduce discrete decisions (e.g., intersection flags or active dual masks) to determine local topology. Dual Marching Cubes (DMC) [35] operates on dual graph structures, placing vertices within cells, improving expressiveness over MC. FlexiCubes [42] further introduces grid deformation [8] and learnable interpolation weights, enabling differentiable extraction with a smooth

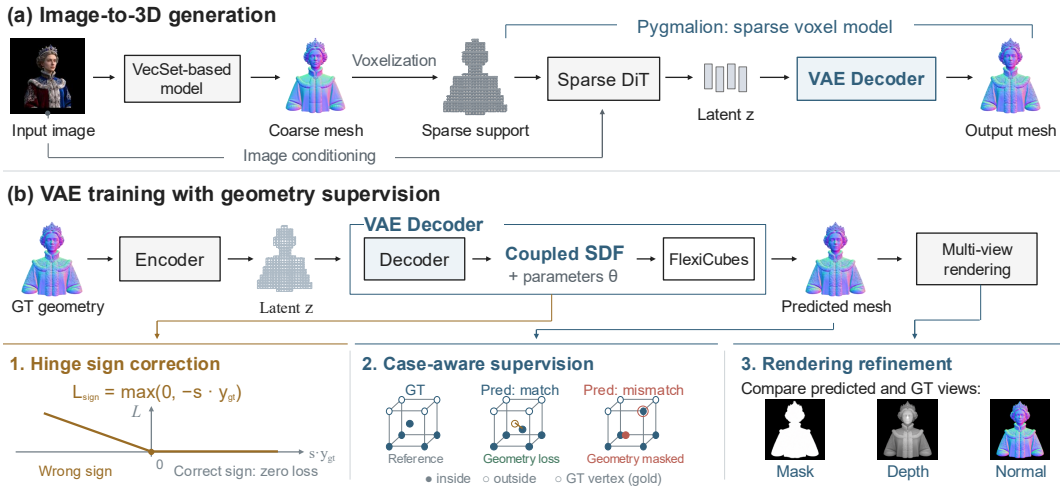

Figure 2: **Pipeline overview.** (a) Image-to-3D generation: a VecSet-based model generates a coarse mesh defining sparse support; a sparse voxel-based model refines geometry on this support to produce a high-quality mesh. (b) End-to-end mesh autoencoding: the VAE encodes and reconstructs meshes with sign, case-aware geometry, and late-stage rendering supervision.

output parameterization. However, FlexiCubes is originally designed for differentiable rendering [18]: its deformations and weights are not directly specified by ground-truth geometry, making explicit geometric supervision non-trivial. Prior methods thus either rely solely on rendering supervision [10, 59], sacrificing high-frequency detail, or decouple the SDF into occupancy and unsigned distance [30] to enable explicit supervision, reintroducing discontinuous decision boundaries.

3 Method

Overview. Figure 2 summarizes our two-stage image-to-3D pipeline. A coarse VecSet model generates a mesh that defines sparse voxel support; a fine-stage DiT samples image-conditioned latents on this support, which our VAE then decodes via coupled SDF and FlexiCubes. We train the VAE with sign correction and case-aware mesh losses, add rendering refinement later, and then freeze the VAE to train the DiT.

3.1 Preliminary: Explicit Mesh-Level Supervision via Voxel Alignment

TopoMesh [30] represents both ground-truth and predicted geometry under a shared DMC framework. Both meshes are extracted from sparse voxel grids at the same resolution, where voxel-level alignment induces local vertex- and face-level correspondences between predictions and targets. This enables explicit mesh-level supervision without auxiliary global matching.

To supervise the sign configuration at each voxel, which determines local mesh topology, TopoMesh decouples the SDF s into an occupancy logit o and an unsigned distance u. The occupancy is supervised against ground-truth o_{gt} using Binary Cross-Entropy:

$$\mathcal{L}_{\text{topo}} = \text{BCEWithLogits}(o, o_{\text{gt}}) \tag{1}$$

During mesh extraction, the signed distance is recovered as:

$$s = u \cdot (1 - 2 \cdot \mathbf{1}[o \geq 0]) \tag{2}$$

where $u \geq 0$ is the predicted unsigned distance. Vertex positions are then obtained by the differentiable FlexiCubes extraction procedure, which interpolates the recovered SDF values together with learnable interpolation weights to place vertices within each cell. This decoupling enables stable training for reconstruction, but the thresholded sign recovery introduces a hard decision boundary that becomes problematic during generation, as analyzed in Sec. 3.2.

3.2 What is a Smooth Parameterization for Generation?

We argue that the decoupled Occ/UDF design in TopoMesh enables explicit mesh-level supervision and high-fidelity reconstruction, but its hard sign thresholding introduces brittleness that becomes

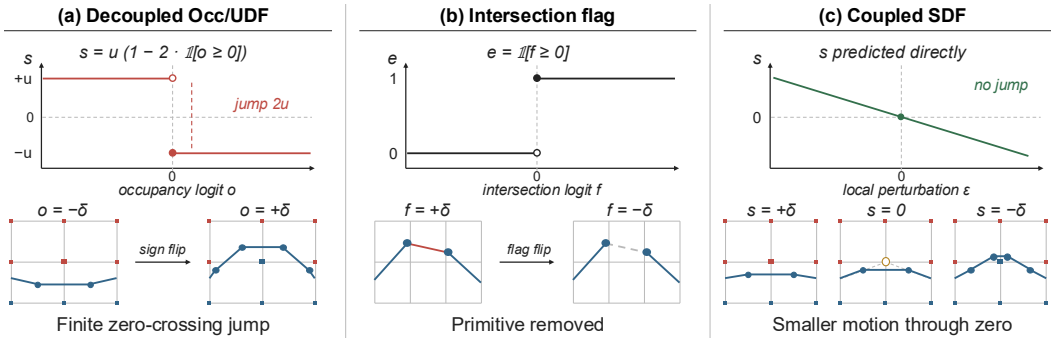

Figure 3: Decoder parameterizations under small perturbations. Top: parameter mappings. Bottom: dual geometry. Decoupled Occ/UDF causes finite zero-crossing jumps; intersection-flag flips can remove primitives and create holes; coupled SDF enables smooth local geometric transitions.

critical during generation. In autoencoding, the encoder provides relatively clean latents, allowing the decoder to make discrete predictions with high confidence. During generation, however, latents sampled from the diffusion model are inherently noisier and more ambiguous, making decoded parameters more likely to approach or cross decision boundaries. We therefore analyze three representative parameterizations under such perturbations, as illustrated in Fig. 3.

Decoupled Occ/UDF. When the occupancy logit o is perturbed across zero, the recovered SDF changes abruptly from $+u$ to $-u$, or vice versa. Since vertex positions are determined by interpolating SDF values at neighboring grid corners, this sudden sign reversal causes the zero-crossing to jump discontinuously, producing visible grid-like artifacts.

Intersection Flag. Some methods [31, 58] use binary flags to directly control the existence of edges or dual primitives in the extracted mesh. A flag predicted slightly below its threshold can therefore remove an entire primitive, creating holes in the output surface.

Coupled SDF. In contrast, a coupled SDF involves no hard thresholding, as the predicted values directly participate in vertex interpolation. Small perturbations in SDF values therefore lead to continuous changes in the interpolated zero crossing, producing smooth geometric transitions.

Based on this analysis, we adopt coupled SDF as a continuous output field, eliminating hard thresholding from the decoder representation. This provides a continuous, noise-robust parameterization for diffusion-based 3D generation. The remaining challenge is how to apply explicit mesh-level supervision within this coupled framework, which is addressed next.

3.3 Explicit Mesh-Level Supervision within Coupled Parameterization

Directly applying explicit supervision on a coupled SDF parameterization is non-trivial. In FlexiCubes, the SDF jointly determines local topology through its sign and vertex placement through its magnitude, making topology correction and geometry refinement difficult to supervise independently. The BCE occupancy supervision used in decoupled formulations cannot be directly applied without reintroducing a hard decision boundary. To retain a smooth coupled representation while enabling effective explicit supervision, we introduce three complementary mechanisms.

Hinge-Based Sign Correction. To provide topology guidance without constraining the SDF magnitude once the sign is correct, we introduce a hinge-based loss on SDF:

$$\mathcal{L}_{\text{sign}} = \max(0, -s \cdot y_{\text{gt}}), \quad y_{\text{gt}} = 1 - 2 \cdot o_{\text{gt}} \tag{3}$$

where s is the predicted SDF, $o_{\text{gt}} \in \{0, 1\}$ is the ground-truth occupancy ($o_{\text{gt}} = 1$ denotes inside with negative SDF, $o_{\text{gt}} = 0$ denotes outside with positive SDF), and $y_{\text{gt}} \in \{-1, 1\}$ denotes the corresponding target sign. The loss activates only when the predicted sign disagrees with the target. Once the sign is correct, the loss becomes zero, leaving the SDF magnitude to be optimized solely by mesh-level geometric losses. This decouples topology correction from geometry refinement at the loss level, without introducing discontinuities into the output representation.

Case-Aware Supervision Scheme. In FlexiCubes, the correspondence between predicted and ground-truth vertices is reliable only when their local topological configurations match. For a voxel v with

eight corners, we encode the sign pattern and define its validity mask as

$$b_i = \mathbf{1}[s_i < 0], \quad c(v) = \sum_{i=0}^{7} 2^i b_i, \quad m_v = \mathbf{1}[c(v) = c_{\text{gt}}(v)] \tag{4}$$

where s_i is the predicted SDF at corner i, b_i is its binary inside indicator, m_v is a validity mask that equals one only when all eight signs match the ground-truth code c_{gt}(v) on the same voxel grid. Under the shared extraction convention, we apply vertex losses only to vertices from valid cells, and normal losses only to faces whose associated vertices all come from valid cells. This avoids misleading gradients from mismatched mesh elements. Sign correction and masked geometry losses are optimized jointly: the sign loss corrects mismatches while geometry losses refine regions with valid correspondences.

Rendering-Based Refinement. For regions where the local configurations remain incorrect, explicit mesh-level losses cannot provide reliable supervision. We complement them with multi-view rendering losses following [59], introduced only in the later training stage to prevent it from overriding the dominant mesh-level signal. Unlike methods such as Trellis [59] and SparseFlex [10] that primarily rely on rendering and inevitably oversmooth sharp features, rendering loss here serves only as a complementary refinement. We ablate its contribution in the experiments.

Together, these designs enable high-fidelity mesh autoencoding within the coupled SDF parameterization while maintaining the smoothness required for diffusion-based generation.

3.4 Scalable and High-Quality 3D Generation

Building on the noise robustness of our coupled SDF representation (Sec. 3.2), we integrate Pygmalion into a scalable image-to-3D pipeline following the high-level two-stage paradigm of Ultra3D [4]. The first stage uses a VecSet-based model to generate a coarse mesh capturing global shape structure, which is voxelized to define sparse voxel support for the second stage. The sparse voxel VAE encodes fine-grained geometry via Voxel-Point Cross Attention [30] and stacked ConvNeXT blocks [26, 58]. For the DiT, we adopt the standard architecture [36] but remove the patchify operation following [58], scaling model sizes to 0.6B, 1.5B, and 4.5B parameters. Built on the generation-friendly VAE, all variants scale consistently along two axes: model capacity during training and output resolution during inference, directly generating meshes at 1536^3 without cascaded refinement.

4 Experiment

4.1 Reconstruction

Baselines and Evaluation. We compare Pygmalion with VecSet-based VAEs, including TripoSG [21], Dora [3], Hunyuan3D-2.1 [49], as well as sparse voxel-based VAEs, including Trellis [59], Direct3D-S2 [57], SparseFlex [10], Trellis2 [58], and TopoMesh [30]. All baselines are evaluated using official implementations, except for TopoMesh, for which we report results provided by the authors. We evaluate on Topo-Bench [30] and Dora-Bench (L3-L4) [3], which contain 1K and 1.4K objects, respectively, with complex structures and high-frequency geometric details. All methods produce meshes at 1024^3 resolution except Trellis, which operates at 256^3 resolution. We report 3D metrics including Chamfer Distance (CD), F-Score (F1), and F-Score on points near sharp features (F1-S) [30], together with 2D metrics including PSNR and LPIPS computed from normal maps rendered from 12 surrounding views.

High-fidelity Reconstruction. We present quantitative results in Tab. 2 and qualitative results in Fig. 4. Our method achieves the best reconstruction quality in both high-frequency detail recovery and overall visual fidelity. As highlighted by red boxes, Direct3D-S2 and SparseFlex over-smooth fine details such as the panel seams on the telescope, Trellis2 produces visible holes on the telescope barrel, and TopoMesh exhibits grid-like artifacts on the flat walls of the attic. In contrast, our method faithfully reconstructs fine geometric details with complete surfaces. Moreover, our method is significantly more compact, reconstructing meshes at 1024^3 resolution with only an average of 13K tokens, significantly fewer than Direct3D-S2 (76K), SparseFlex (245K), and TopoMesh (56K), while still matching or outperforming them on most metrics.

Table 2: Quantitative result of VAE reconstruction. All methods except Trellis produce meshes at 1024^3 resolution. CD and LPIPS are multiplied by 10^5 and 10^2, respectively.

Method	#Token	#Dim	Topo-Bench					Dora-Bench				
			CD ↓	F1 ↑	F1-S ↑	PSNR ↑	LPIPS ↓	CD ↓	F1 ↑	F1-S ↑	PSNR ↑	LPIPS ↓
TripoSG	4K	64	2.658	0.893	0.715	24.91	4.561	1.697	0.959	0.717	27.45	3.305
Dora	4K	64	2.167	0.905	0.754	25.19	5.127	1.814	0.964	0.768	27.58	3.739
HY3D-2.1	4K	64	2.538	0.888	0.767	24.28	5.976	1.606	0.954	0.770	27.12	3.782
Trellis	13K	8	18.616	0.583	0.308	20.38	7.529	14.716	0.715	0.279	20.73	5.866
Direct3D-S2	76K	16	2.713	0.813	0.694	24.78	5.512	2.313	0.881	0.819	25.14	5.576
SparseFlex	245K	8	**1.840**	**0.920**	0.873	30.81	1.281	1.625	0.973	0.844	33.30	0.812
Trellis2	13K	32	1.893	0.915	0.868	32.62	1.290	1.124	0.971	0.840	33.19	1.135
TopoMesh	56K	8	1.882	0.917	**0.932**	31.99	1.127	1.126	0.973	0.915	34.30	0.652
Ours	13K	32	1.856	0.918	0.928	**33.47**	**0.818**	**1.089**	**0.974**	**0.915**	**35.85**	**0.452**

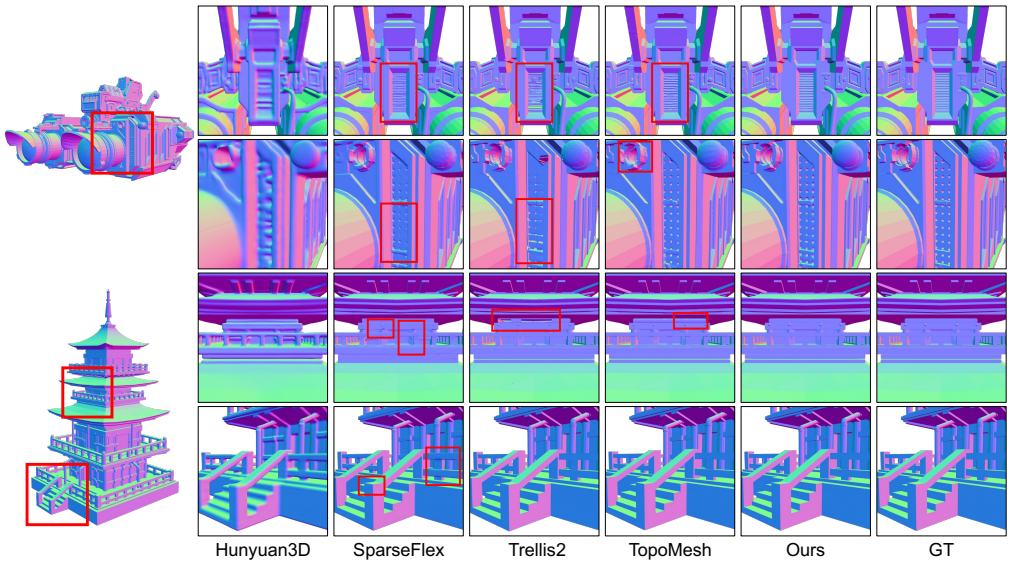

Figure 4: Qualitative comparison of VAE reconstruction. Trellis2 produces holes due to incorrect intersection flag predictions, and TopoMesh exhibits grid-like artifacts from discontinuous sign decisions. In contrast, Pygmalion reconstructs clean, high-fidelity meshes.

4.2 Generation

Baselines and Evaluation. We compare Pygmalion against leading closed-source models, including Tripo H3.1 [51], Hunyuan3D 3.1 [50], Hitem3D 2.1 [32], as well as the open-source state-of-the-art Trellis2 [58]. All methods generate meshes at 1536^3 resolution; for closed-source models, results are obtained through their official web interfaces under the highest-precision setting (e.g., maximum polygon budget). We collect over 40 evaluation images primarily from the official galleries of the compared closed-source models, ensuring the evaluation covers cases where existing methods perform competitively. The images span diverse categories including characters, buildings, mechs, and everyday objects, as shown in Fig. 6 (left). We report the Uni3D score [66], a 3D-aware embedding similarity metric, for all 0.6B, 1.5B, and 4.5B model variants to measure image-shape alignment (Tab. 4). For perceptual quality and geometric fidelity, we conduct a user study comparing our 4.5B model against those methods. We recruit 15 raters with professional 3D modeling backgrounds, and for each pair of generated meshes, raters are asked to select the better result or mark them as comparable. Fig. 6 (right) reports human preference results.

High-quality Generation. We present qualitative results in Fig. 5 and quantitative comparisons in Fig. 6. Pygmalion generates high-quality 3D assets with better geometric details and image-mesh alignment. For instance, Pygmalion more faithfully recovers the facial expression of the female bust and the hairstyle of the male figure, whereas Hitem3D 2.1 and Trellis2 produce visible holes on the generated surfaces, and Hunyuan3D 3.1 blurs fine geometric details such as the ornaments on the female dress. Quantitatively, Pygmalion achieves the highest Uni3D score [66] (*0.4589*)

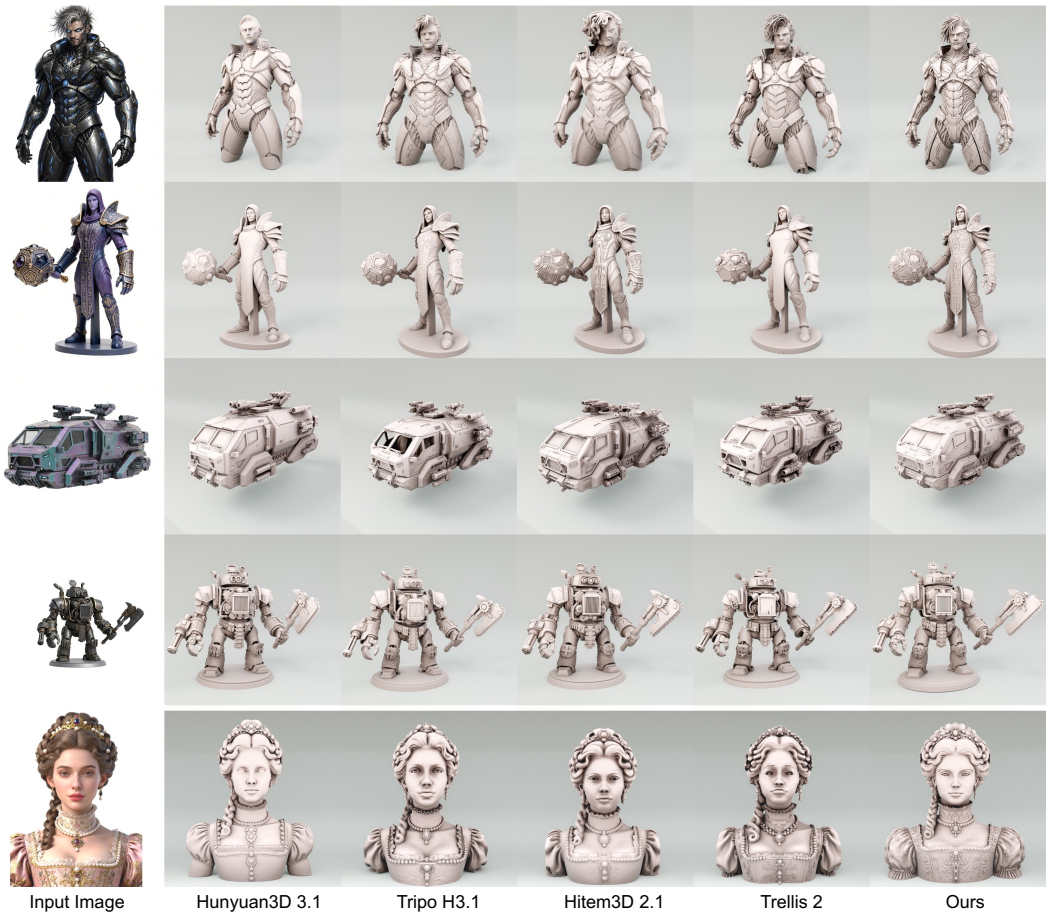

Figure 5: Qualitative comparison of image-to-3D generation against state-of-the-art methods. Our method produces meshes with sharp details, cleaner surfaces, and faithful geometry to input images.

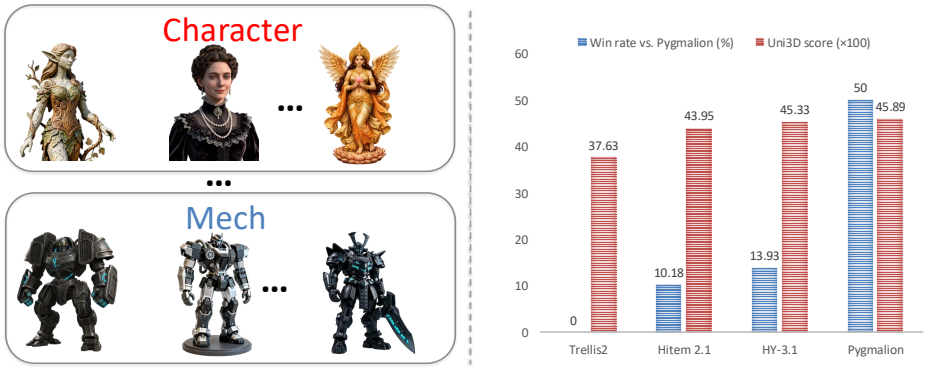

Figure 6: Evaluation set and quantitative results. **Left:** representative test images spanning diverse categories with high geometric precision requirements. **Right:** Win rate of baselines against Pygmalion (blue, %, user study with 15 modelers on 40 test images) and Uni3D score (red, $\times 100$). Pygmalion's self win rate is set to 50% for visualization.

among all compared methods, demonstrating faithful alignment to the input images. In the user study, Pygmalion is preferred in over *86%* of pairwise comparisons against every baseline.

Mesh Integrity. We present topology results in Tab. 3 to evaluate mesh integrity in both reconstruction and generation, including watertight ratio (WT), non-manifold edge percentage (NME), and the boundary loop (BL), as detailed in Appendix C. Higher WT and lower NME/BL indicate better mesh topology. Our reconstruction reaches WT 0.999 and BL 0.12. In generation, both our method and Hunyuan3D 3.1 have WT 1.000 and zero reported NME/BL. Our method achieves the best mesh integrity in both reconstruction and generation, with the highest WT and lowest NME and BL.

Table 3: Quantitative results of mesh integrity in reconstruction and generation.

Reconstruction				Generation			
Method	WT ↑	NME ↓	BL ↓	Method	WT ↑	NME ↓	BL ↓
Direct3D-S2	0.164	5.1×10^{-2}	140.74	Hunyuan3D 3.1	**1.000**	**0.0**	**0.0**
SparseFlex	0.217	9.5×10^{-3}	32.02	Tripo 3.1	0.271	3.0×10^{-4}	0.02
Trellis2	0.001	8.2×10^{-1}	3399.74	Hitem3D 2.1	0.979	1.0×10^{-6}	0.0
TopoMesh	0.963	9.4×10^{-5}	1.43	Trellis2	0.000	4.1	104.7
Ours	**0.999**	3.8×10^{-5}	**0.12**	Ours	**1.000**	**0.0**	**0.0**

Table 4: Quantitative results of ablation. CD, LPIPS, and NME are scaled by 10^5, 10^2, and 10^5.

	Reconstruction								Generation: Uni3D ↑		
Setting	CD ↓	F1 ↑	F1-S ↑	PSNR ↑	SSIM ↑	LPIPS ↓	WT ↑	NME ↓	0.6B	1.5B	4.5B
Decoupled Occ/UDF	1.517	0.970	0.896	33.97	0.952	1.001	0.986	285.7	0.3987	0.3985	0.4179
Coupled SDF + BCE	1.287	0.960	0.863	33.60	0.946	1.345	0.996	284.3	–	–	–
Coupled SDF + Hinge	1.279	0.963	0.874	34.21	0.954	1.006	0.998	**3.7**	–	–	–
+ Case-Aware	1.125	0.972	0.913	35.24	0.965	0.512	0.997	4.1	–	–	–
+ Rendering (Full)	**1.089**	**0.974**	**0.915**	**35.85**	**0.968**	**0.452**	**0.999**	3.8	**0.4407**	**0.4520**	**0.4589**

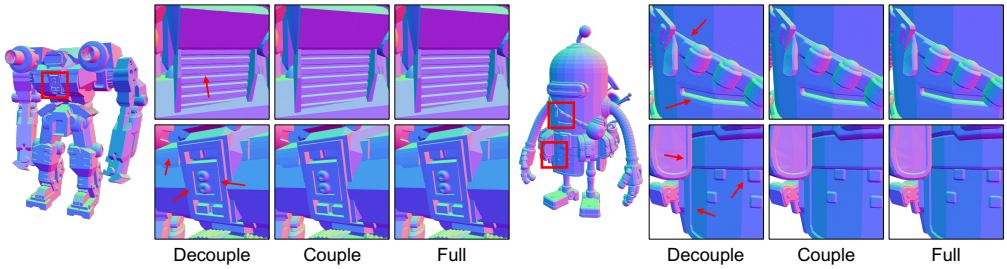

Figure 7: Ablation on output representation for reconstruction. The decoupled Occ/UDF variant exhibits grid-like artifacts, such as noticeable noise on flat surfaces (red arrows), which the coupled SDF eliminates. Rendering refinement further improves overall surface quality.

4.3 Ablation

Coupled SDF. To isolate the effect of output representation, we compare the coupled SDF representation with the decoupled Occ/UDF formulation in both reconstruction and generation. For reconstruction, as shown in Tab. 4 and Fig. 7, coupled SDF with explicit mesh-level supervision already improves geometric detail and visual quality without rendering-based refinement, removing grid-like artifacts at surface seams and producing clean, sharp transitions. For generation, Tab. 4 and Fig. 8 further show consistent improvements across all DiT model sizes, especially in detail-rich regions where coupled SDF suppresses grid-like artifacts and yields cleaner surfaces.

Rendering-based Refinement. As shown in Tab. 4 and Fig. 7, rendering-based refinement further improves geometric fidelity and visual quality, complementing explicit mesh-level supervision in regions where local topological configurations remain incorrect.

Robustness to Latent Perturbations. To evaluate noise robustness, we perturb latent codes with Gaussian noise scaled by each channel's standard deviation, using $\sigma \in \{0, 0.10, 0.20\}$. Tab. 5 compares our coupled and decoupled variants with Trellis2. At $\sigma = 0.20$, coupled SDF achieves substantially lower CD (1.679 vs. 12.591) and higher WT (0.973 vs. 0.868) than decoupled Occ/UDF, demonstrating greater robustness in both geometric fidelity and mesh integrity.

DiT Scalability. We train 0.6B, 1.5B, 4.5B DiTs on top of Pygmalion's VAE. Fig. 9 shows two scaling axes: larger models improve image alignment and details such as roof eaves, lion statues, and column reliefs; higher inference resolution (512^3 to 1536^3) recovers finer surface carvings. Despite growing detail richness, all variants maintain clean surfaces free of grid-like artifacts or topology failures, demonstrating that the continuous SDF representation preserves geometric integrity under aggressive scaling. Quantitatively, the Uni3D [66] score improves with model size (Tab. 4).

Table 5: Robustness to latent perturbations. CD and LPIPS are multiplied by 10^5 and 10^2.

Representation	σ	CD ↓	F1 ↑	F1-S ↑	PSNR ↑	SSIM ↑	LPIPS ↓	WT ↑	NME ↓
Intersection flag	0	1.124	0.971	0.840	33.19	0.948	1.135	0.001	8.2×10^{-1}
Decoupled Occ/UDF	0	1.517	0.970	0.896	33.97	0.952	1.001	0.986	2.9×10^{-3}
Coupled SDF	0	**1.089**	**0.974**	**0.915**	**35.85**	**0.968**	**0.452**	**0.999**	$\mathbf{3.8 \times 10^{-5}}$
Intersection flag	0.1	1.829	0.969	0.823	30.06	0.912	2.192	0.000	1.2
Decoupled Occ/UDF	0.1	8.532	0.963	0.864	31.44	0.920	2.586	0.978	8.4×10^{-3}
Coupled SDF	0.1	**1.414**	**0.971**	**0.894**	**32.78**	**0.937**	**1.145**	**0.992**	$\mathbf{3.4 \times 10^{-3}}$
Intersection flag	0.2	2.481	0.963	0.804	26.81	0.846	3.892	0.000	1.8
Decoupled Occ/UDF	0.2	12.591	0.954	0.843	28.02	0.845	4.191	0.868	1.2×10^{-2}
Coupled SDF	0.2	**1.679**	**0.970**	**0.878**	**30.04**	**0.872**	**1.243**	**0.973**	$\mathbf{5.3 \times 10^{-3}}$

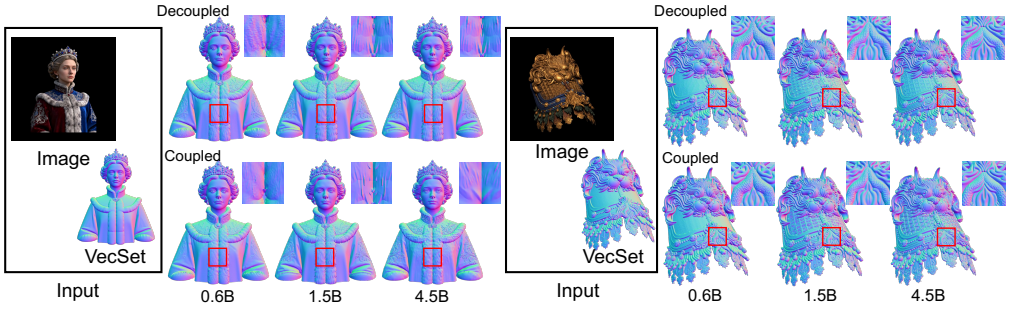

Figure 8: Ablation on output representation for generation. Decoupled exhibits grid-like noise across all DiT sizes, while coupled produces clean surfaces even at 0.6B.

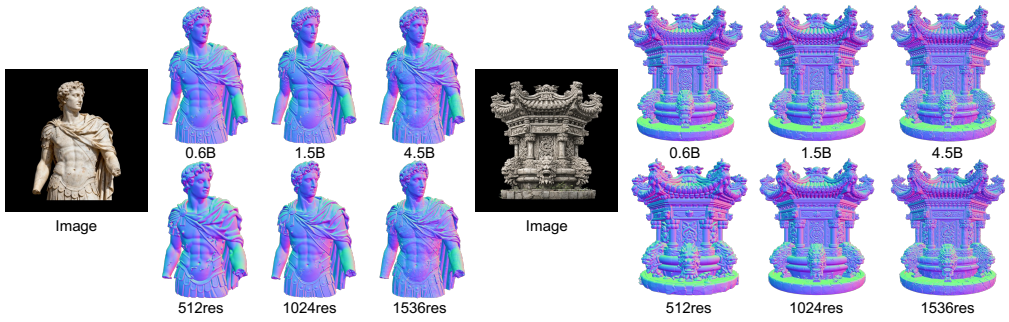

Figure 9: Ablation on DiT scalability. Our pipeline scales along two axes: increasing model capacity improves image alignment and geometric detail, while increasing inference resolution reveals finer surface features—both preserving complete, clean surfaces.

5 Limitation

Our method has two main limitations. Sparse voxel token counts grow with resolution and geometric complexity, leading to a higher inference cost than compact VecSet-based representations. In addition, the two-stage generation pipeline may suffer from error accumulation: inaccurate first-stage coarse structures can limit subsequent sparse voxel refinement, which operates on local voxel neighborhoods and cannot reliably correct large-scale structural errors.

6 Conclusion

We present Pygmalion, a sparse voxel VAE that combines coupled SDF with hinge-based sign correction, case-aware supervision, and rendering refinement to improve reconstruction fidelity and generation robustness. Built on this VAE, our DiT models scale consistently across model sizes and inference resolutions, generating high-quality meshes at up to 1536^3 resolution. Experiments demonstrate state-of-the-art reconstruction and generation performance with fine geometric details and high mesh integrity, highlighting smooth output parameterization as a key factor for robust sparse voxel-based 3D generation.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key Research and Development Program of China (No. 2023YFF0905100), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China(No. JYB2025XDXM101), and the Natural Science Foundation of China (No. 62132012, 62361146854).

References

[1] ByteDance Seed. Seed3D 2.0: Advancing high-fidelity simulation-ready 3d content generation. `https://lf3-static.bytednsdoc.com/obj/eden-cn/lapzild-tss/ljhwZthlaukjlkulzlp/pdf/Seed3D_v2.pdf`.

[2] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[3] Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[4] Yiwen Chen, Zhihao Li, Yikai Wang, Hu Zhang, Qin Li, Chi Zhang, and Guosheng Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention. *arXiv preprint arXiv:2507.17745*, 2025.

[5] Zhiqin Chen, Andrea Tagliasacchi, Thomas A. Funkhouser, and Hao Zhang. Neural dual contouring. *ACM Trans. Graph.*, 2022.

[6] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[7] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 2023.

[8] Jun Gao, Wenzheng Chen, Tommy Xiang, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Learning deformable tetrahedral meshes for 3d reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[9] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oguz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023.

[10] Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[11] 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 *International Conference on Learning Representations (ICLR)*, 2024.

[12] Tao Ju, Frank Losasso, Scott Schaefer, and Joe D. Warren. Dual contouring of hermite data. *ACM Trans. Graph.*, 2002.

[13] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023.

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[16] Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. LATTICE: democratize high-fidelity 3d generation at scale. *arXiv preprint arXiv:2512.03052*, 2025.

[17] Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qingxiang Lin, Jingwei Huang, Yuhong Liu, Jie Jiang, Chunchao Guo, and Xiangyu Yue. Unleashing vecset diffusion model for fast shape generation. *arXiv preprint arXiv:2503.16302*, 2025.

[18] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.*, 2020.

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

[20] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native diffusion and interactive geometry refiner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[21] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[22] Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *arXiv preprint arXiv:2505.14521*, 2025.

[23] 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)*, 2023.

[24] Minghua Liu, Chong Zeng, Xinyue Wei, Ruoxi Shi, Linghao Chen, Chao Xu, Mengqi Zhang, Zhaoning Wang, Xiaoshuai Zhang, Isabella Liu, Hongzhi Wu, and Hao Su. Meshformer : High-quality mesh generation with 3d-guided reconstruction model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[25] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *International Conference on Learning Representations (ICLR)*, 2024.

[26] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[27] 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)*, 2024.

[28] William E. Lorensen and Harvey 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 1987, Anaheim, California, USA, July 27-31, 1987*, 1987.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[30] Guan Luo, Xiu Li, Rui Chen, Xuanyu Yi, Jing Lin, Chia-Hao Chen, Jiahang Liu, Song-Hai Zhang, and Jianfeng Zhang. Topomesh: High-fidelity mesh autoencoding via topological unification. *arXiv preprint arXiv:2603.24278*, 2026.

[31] Yihao Luo, Xianglong He, Chuanyu Pan, Yiwen Chen, Jiaqi Wu, Yangguang Li, Wanli Ouyang, Yuanming Hu, Guang Yang, and Choon Hwai Yap. Faithful contouring: Near-lossless 3d voxel representation free from iso-surface. *arXiv preprint arXiv:2511.04029*, 2025.

[32] MathMagic. Hitem3D: Ai-powered 3d model generator. `https://www.hitem3d.ai/`.

[33] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

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

[35] Gregory M. Nielson. Dual marching cubes. In *IEEE Visualization*, 2004.

[36] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[37] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations (ICLR)*, 2023.

[38] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

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

[41] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

[42] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Trans. Graph.*, 2023.

[43] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[44] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International Conference on Learning Representations (ICLR)*, 2024.

[45] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023.

[46] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. In *International Conference on Learning Representations (ICLR)*, 2024.

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

[48] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations (ICLR)*, 2024.

[49] Tencent Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025.

[50] Tencent Hunyuan3D Team. Tencent Hunyuan 3D. `https://3d.hunyuan.tencent.com/`.

[51] Tripo AI. Tripo: Ai 3d model generator. `https://www.tripo3d.ai/`.

[52] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. SV3D: novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision (ECCV)*, 2024.

[53] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

[54] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[55] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. CRM: single image to 3d textured mesh with convolutional reconstruction model. In *European Conference on Computer Vision (ECCV)*, 2024.

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

[57] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[58] 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. *arXiv preprint arXiv:2512.14692*, 2025.

[59] 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 (CVPR)*, 2025.

[60] 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.

[61] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[62] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: latent point diffusion models for 3d shape generation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[63] Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 2023.

[64] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. CLAY: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Trans. Graph.*, 2024.

[65] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[66] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations (ICLR)*, 2024.

A Continuity Analysis

We analyze the brittleness of intersection-flag and decoupled Occ/UDF representations, and show that the coupled SDF representation provides a continuous parameter-to-geometry mapping. In decoupled representations, small perturbations can be amplified by hard thresholding into discontinuous changes in local topology or vertex positions. In contrast, coupled SDF allows decoder parameters to directly and continuously determine the extracted geometry, so parameter perturbations induce only smooth geometric variations.

For clarity, we analyze the extraction at a local level. In MC-based methods, a vertex is placed at the zero crossing along a grid edge connecting corners i and j at positions $\mathbf{x}_i$ and $\mathbf{x}_j$. In FlexiCubes, vertex placement further involves a weighted interpolation over neighboring corners. Despite the difference, the continuity analysis follows the same logic: we examine whether the extracted vertex position varies continuously under small perturbations of the decoded parameters.

A.1 Intersection Flag

In methods like Trellis2 [58] and FaithC [31], the existence of each local mesh primitive is determined by thresholding a predicted intersection flag f:

$$\text{primitive exists} \iff f \geq 0. \tag{5}$$

This hard threshold induces a discontinuous mapping from decoder outputs to mesh connectivity. When f is perturbed from $+\epsilon$ to $-\epsilon$, the primitive switches from present to absent. As a result, an infinitesimal perturbation in f can remove an entire edge or face, producing a visible change in the extracted mesh, leading to holes. Therefore, the extraction mapping is discontinuous at the decision boundary $f = 0$, making intersection-flag parameterizations brittle under small perturbations.

A.2 Decoupled Occ/UDF

In methods such as TopoMesh [30] and Sparc3D [22], the signed distance is recovered from a separate occupancy logit o and unsigned distance $u > 0$:

$$s = u \cdot (1 - 2 \cdot \mathbf{1}[o \geq 0]). \tag{6}$$

This introduces a hard threshold at $o = 0$. When o is perturbed from $+\epsilon$ to $-\epsilon$ with arbitrarily small $\epsilon > 0$, the recovered SDF changes from $-u$ to $+u$, producing a jump of magnitude $2u$. Since extracted vertices are obtained by interpolating SDF values along grid edges, such a discontinuous change in s_i can abruptly alter the local zero-crossing position. Therefore, the extraction mapping is discontinuous at the occupancy decision boundary $o = 0$, leading to vertex jumps and grid-aligned artifacts under small perturbations.

A.3 Coupled SDF

In the coupled SDF parameterization, the decoder directly predicts signed distance values $s_i \in \mathbb{R}$ at grid corners, together with the continuous parameters used by FlexiCubes for vertex placement. Unlike decoupled Occ/UDF, the sign is not recovered from a separate thresholded occupancy variable; instead, the same continuous SDF values determine both the local sign configuration and the interpolated vertex positions. For an active grid edge (i, j) with opposite signs, the zero crossing is obtained by linear interpolation:

$$\mathbf{v}_{ij} = \frac{s_j \cdot \mathbf{x}_i - s_i \cdot \mathbf{x}_j}{s_j - s_i}. \tag{7}$$

This mapping is continuous in (s_i, s_j) whenever the denominator is non-zero, which holds for any non-degenerate zero-crossing edge. Therefore, small perturbations of the predicted SDF values induce continuous changes in the extracted vertex position. Since the learnable interpolation and deformation parameters in FlexiCubes also affect vertex placement through continuous operations, the extracted geometry varies smoothly under small perturbations of the decoded parameters.

Unlike decoupled Occ/UDF, coupled SDF avoids thresholded sign recovery. In the decoupled case, an infinitesimal perturbation of the occupancy logit can flip the recovered SDF from $+u$ to $-u$, causing a finite jump in the values used for interpolation. In the coupled case, the sign changes only when

the SDF value itself crosses zero continuously. As this value approaches zero, the corresponding zero crossing moves smoothly toward the grid corner; after the sign changes, it moves away along the adjacent edge. Therefore, no positional discontinuity is introduced. This yields continuous local changes in geometry and avoids the vertex jumps that lead to grid-like artifacts.

B Implementation Details

B.1 Architecture

VAE. Our encoder aggregates point cloud features into sparse voxel features using Voxel-Point Cross Attention [30], followed by stacked ConvNeXT [26, 58] blocks to produce the sparse voxel latent representation. The decoder processes sparse voxel latents with ConvNeXT blocks and progressively upsamples them to the output resolution for mesh extraction. The VAE contains approximately 666M parameters.

DiT. We adopt a standard DiT [36] architecture composed of self-attention, cross-attention, and MLP layers. We train sparse voxel-based DiT model variants at three different scales with configurations summarized in Tab. 6.

Table 6: DiT model configurations at three scales.

Model Size	Layers	Hidden Size	MLP Size	Num Heads
0.6B	24	1024	4096	16
1.5B	28	1536	6144	24
4.5B	48	2048	8192	32

B.2 Training

Training Losses. We adopt the explicit mesh-level losses from TopoMesh [30], but replace the L1 losses on vertex positions and face orientations with MSE losses to provide smoother gradients as training converges. In the early training stage, we apply only explicit mesh-level supervision. In the later training stage, we introduce rendering-based refinement following Trellis [59], rendering multi-view masks, depth maps, and normal maps, and computing MSE losses against the ground truth. The final objective is:

$$\mathcal{L} = \omega_{\text{sign}}\mathcal{L}_{\text{sign}} + \omega_{\text{vert}}\mathcal{L}_{\text{vert}} + \omega_{\text{normal}}\mathcal{L}_{\text{normal}} + \omega_{\text{render}}\mathcal{L}_{\text{render}} \tag{8}$$

To balance different loss terms, we monitor the Frobenius norm of their gradients at the VAE output layer, i.e., the linear projection to the output representation. The weights ω_{sign}, ω_{vert}, and ω_{normal} are chosen so that the three explicit supervision terms yield comparable gradient magnitudes. The rendering weight ω_{render} is set to contribute approximately one-tenth of the explicit supervision gradient, preventing the over-smoothing as observed in SparseFlex [10]. The final weights are $\omega_{\text{sign}} = 10$, $\omega_{\text{vert}} = 4$, $\omega_{\text{normal}} = 1.0$, and $\omega_{\text{render}} = 0.2$.

Strategy. We use the progressive training strategy from [30]. For the VAE, we train from mesh resolution 512^3 to 1024^3, with corresponding latent resolutions of 32^3 and 64^3. The DiT is trained similarly at latent resolutions 32^3 and 64^3. All experiments are conducted on 64 NVIDIA A800 GPUs, with the VAE trained for 4 days and the DiT for 2 weeks, with the AdamW [29] optimizer and a fixed learning rate of 10^{-4}.

Datasets. We curate 500K objects from Objaverse [6], filtering out samples with excessively large or small voxel counts. For each object, we render 20 views using BlenderProc [7] as conditioning images for DiT training.

C Evaluation Details

3D metrics. We normalize each mesh to $[-1, 1]^3$ and uniformly sample 500K points on the surface for computing Chamfer Distance and F-Score, and an additional 500K points sampled from sharp regions with dihedral angles sharper than 30° for computing F1-S.

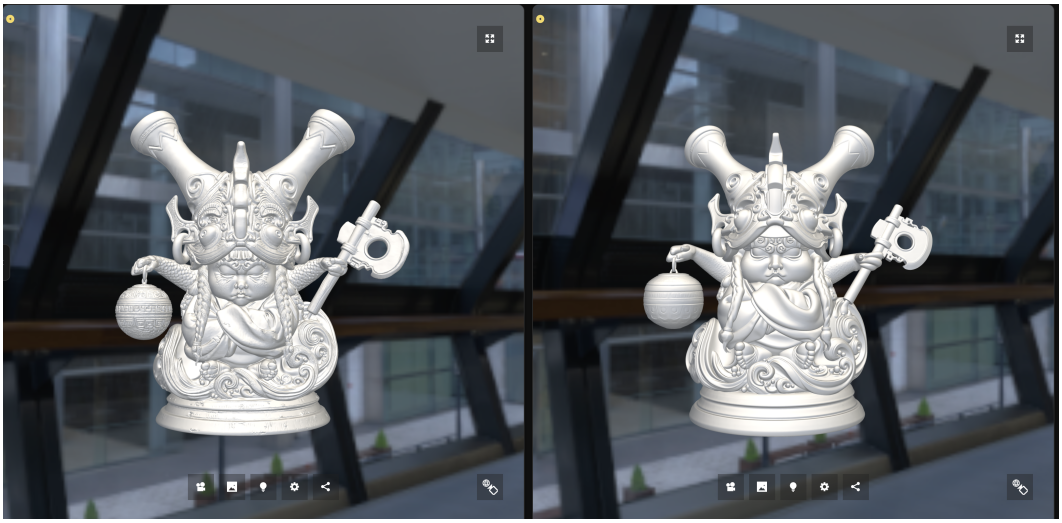

Figure 10: **User study interface.** Side-by-side viewer used in our blind paired comparison: Model A (left) and Model B (right), with order randomized per test image. Each panel supports independent rotation, zoom, and shading control, allowing raters to inspect geometric details from arbitrary viewpoints before judging which result is better, or whether the two are comparable.

2D metrics. We render normal maps at 1024×1024 resolution with camera radius 2 and FoV $40°$. The 12 views are arranged in three elevation rings: elevations $\{30°, 0°, -30°\}$ with azimuths $\{0°, 90°, 180°, 270°\}$, $\{30°, 120°, 210°, 300°\}$, and $\{60°, 150°, 240°, 330°\}$, respectively.

Mesh Integrity. We evaluate mesh integrity using three metrics. The watertight ratio (WT) is the fraction of output meshes that form closed surfaces. The non-manifold edge percentage (NME) is the percentage of edges that violate manifoldness, and the boundary loop count (BL) is the mean number of open boundary loops per mesh.

Latent-Perturbation Robustness. We inject Gaussian noise into the normalized latent space and rescale it by the per-channel standard deviation. Equivalently, for channel c,

$$\widetilde{z}_c = z_c + \sigma a_c \epsilon_c, \qquad \epsilon_c \sim \mathcal{N}(0, I), \tag{9}$$

where a_c is the standard deviation of channel c and $\sigma \in \{0, 0.10, 0.20\}$ controls the perturbation magnitude. For each model, we keep the decoder settings fixed across noise levels. We compare coupled SDF with decoupled Occ/UDF and use the Trellis2 implementation as the intersection-flag reference, including its representation-specific preprocessing.

User Study. We conduct a comprehensive and blind paired comparison study to assess geometric fidelity (Fig. 10). For each pair, two meshes (our 4.5B model vs. a baseline) are presented side-by-side as Model A and Model B in an interactive 3D viewer supporting free rotation, zoom, and shading control, with order randomized per test image to prevent positional bias. We recruit 15 raters with professional 3D modeling backgrounds, who together evaluate over 40 test images across all comparisons. Raters are asked to judge which result is better, or whether the two are comparable.

D Computational Cost

Tab. 7 reports end-to-end inference on a single A800 GPU at 1536^3 output resolution, averaged over multiple runs. We report latency, peak GPU memory, fine-stage token counts, and timings for coarse generation, diffusion, and decoding. The cost comparison uses the open-source Hunyuan3D-2.1; Hunyuan3D 3.1 is evaluated for generation quality through its web interface.

Our 1.5B model takes 82.0 s and 42.64 GB, compared with 77.6 s and 42.29 GB for Trellis2. Scaling from 0.6B to 4.5B increases latency from 58.3 to 154.6 s, while peak memory remains around 43 GB.

Table 7: End-to-end image-to-3D inference cost on a single A800 GPU at 1536^3 output resolution, averaged over multiple runs. Times are in seconds and memory in GB. Diff. and Dec. denote fine-stage diffusion and decoding. NFE denotes the number of function evaluations. Pygmalion model names indicate the fine-stage DiT size; timings include the coarse stage.

Method	Coarse	Fine tokens	NFE	Diff.	Dec.	Total	Peak mem.
Hunyuan3D-2.1	—	4,096	100	11.5	63.9	75.4	60.49
Trellis2	3.2	45,266	21	70.0	4.4	77.6	42.29
Pygmalion-0.6B	19.4	41,647	20	29.1	9.8	58.3	43.10
Pygmalion-1.5B	19.4	41,647	20	53.4	9.2	82.0	42.64
Pygmalion-4.5B	19.4	41,647	20	125.7	9.5	154.6	43.34

The coarse stage takes 19.4 s across all three variants. The approximately 41.6K fine-stage tokens correspond to 1536^3 generation, whereas the 13K tokens in Tab. 2 refer to 1024^3 reconstruction.

E Visualization

We provide more visualizations of VAE reconstruction in Fig. 11 and image-to-3D generation in Fig. 12.

F Ethics

Fake information by generative models can have negative consequences for society. Therefore, we do not allow the use of our model for any activities that could potentially harm society. We hope that users will comply with this requirement.

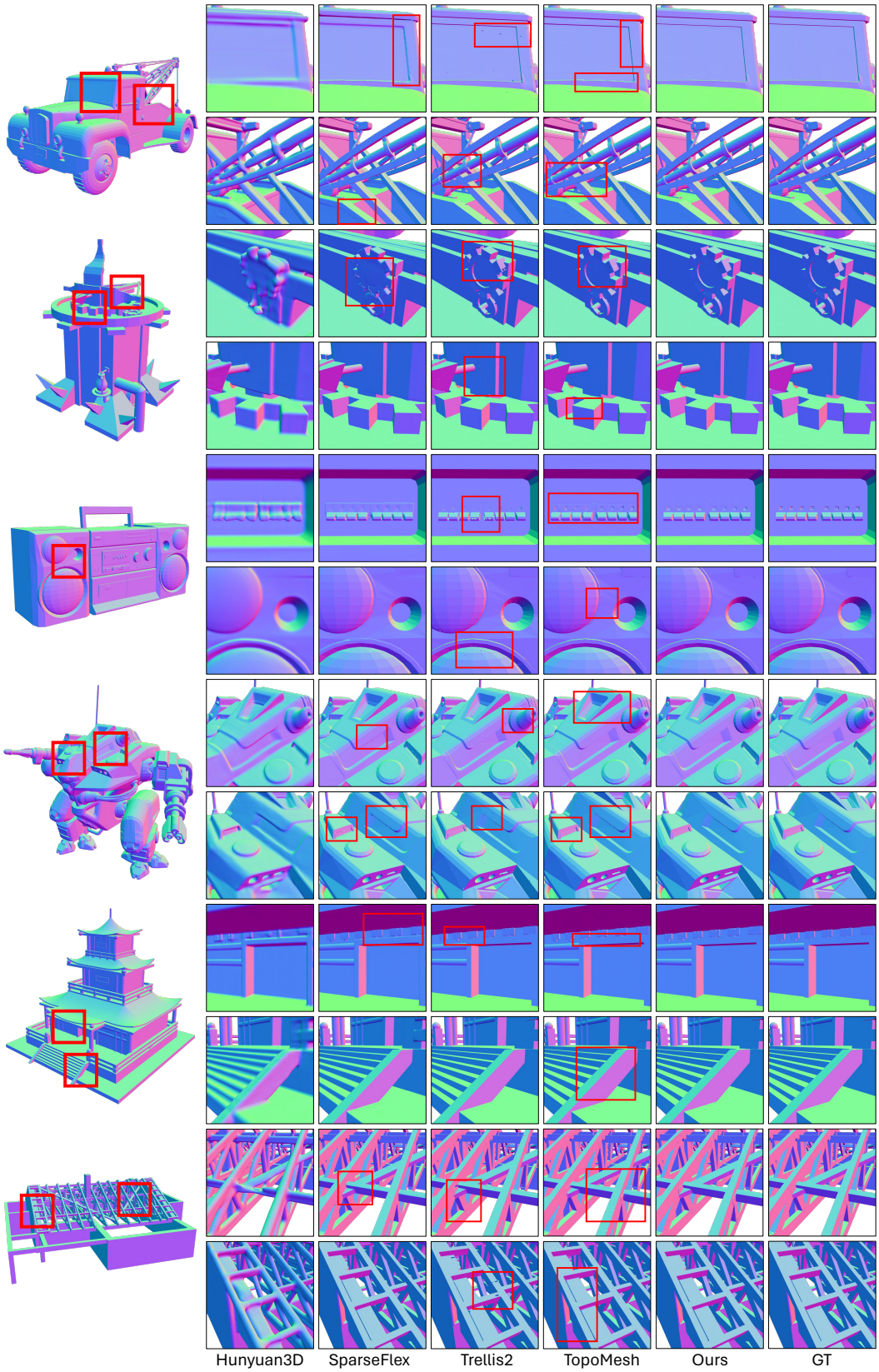

Figure 11: Qualitative comparison of VAE reconstruction, our method better preserves fine geometric details, as highlighted by red boxes.

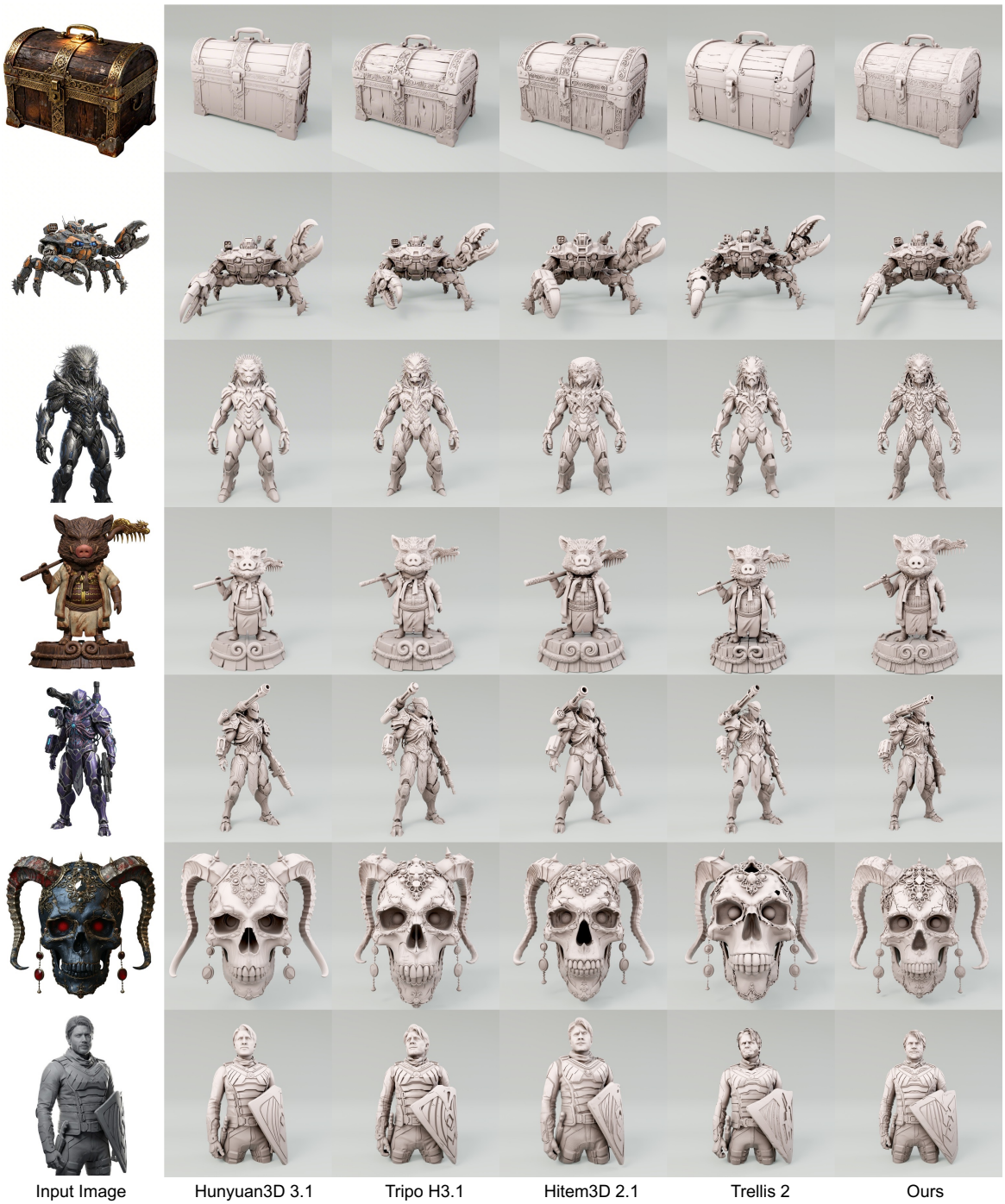

Figure 12: Qualitative comparison of DiT generation, our method better produces high-quality 3D assets with complete surfaces and fine geometric details.

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: We state the scope of our work in the abstract and summarize our contributions at the end of the introduction.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 discuss the limitations of our method in Sec. 5.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: We provide a complete analysis of the continuous property of coupled representation in Sec. A.

Guidelines:

- The answer [N/A] 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: We provide comprehensive details for reproducing both training and evaluation results, including dataset, network architecture, and training configuration in Sec. B.

   Guidelines:

   - The answer [N/A] 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 provide sufficient details for reproducibility, including dataset preparation, model architecture, and training configuration in Sec. B.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

Answer: [Yes]

Justification: We provide sufficient details for reproducibility, including dataset preparation, model architecture, and training configuration in Sec. B.

Guidelines:

- The answer [N/A] 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: [Yes]

Justification: We report standard metrics on fixed benchmarks, following common practice in the field.

Guidelines:

- The answer [N/A] 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 provide sufficient details for the training process and the required resources in Sec. B.

   Guidelines:

   - The answer [N/A] 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: We commit to upholding the code of ethics by NeurIPS. Sec. F.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Generative models may pose threats to society, we do not allow the use of our model in any such activities. Sec. F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We require the user to comply with the code of ethics. Sec. F.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use the Objaverse dataset, which is under ODC-By v1.0 license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: We provide screenshots in Fig. 10.

    - The answer [N/A] 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: [Yes]

    Justification: The user study involves adult professional raters who voluntarily participated and gave consent before the study. The study only asks participants to compare anonymized 3D assets and does not collect sensitive personal information.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We only use LLM for editing and formatting.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.