Direct3D-V3: Latent-Free Diffusion Transformers in the Native Voxel Domain

SHUANG WU*, Nanjing University, China and DreamTech, China
JIACHEN QIAN, The University of Hong Kong, Hong Kong and DreamTech, China
YOUTIAN LIN, Nanjing University, China
FEIHU ZHANG, DreamTech, China
PHILIP TORR, University of Oxford, United Kingdom
YAO YAO†, Nanjing University, China
XUN CAO†, Nanjing University, China



Fig. 1. Direct3D-V3 generates diverse, high-fidelity 3D meshes by modeling raw SDF voxel grids directly with latent-free diffusion transformers, removing the VAE bottleneck used by latent diffusion pipelines.

*This work was conducted during an internship at DreamTech.
†Corresponding authors.

Authors' Contact Information: Shuang Wu, Nanjing University, Nanjing, China and DreamTech, Shenzhen, China, wushuang@smail.nju.edu.cn; Jiachen Qian, The University of Hong Kong, Hong Kong, Hong Kong and DreamTech, Shenzhen, China, qianjiachen@dreamtech.ai; Youtian Lin, Nanjing University, Suzhou, China, linyoutian.loyot@gmail.com; Feihu Zhang, DreamTech, Shenzhen, China, zhangfeihu@dreamtech.ai; Philip Torr, University of Oxford, Oxford, United Kingdom, philip.torr@eng.ox.ac.uk; Yao Yao, Nanjing University, Suzhou, China, yaoyao@nju.edu.cn; Xun Cao, Nanjing University, Nanjing, China, caoxun@nju.edu.cn.

This work is licensed under a Creative Commons Attribution 4.0 International License.

Native 3D generation has largely been built around latent diffusion, with recent advances concentrating on stronger variational autoencoders (VAE) and increasingly elaborate latent representations, from vecset and triplanes to structured sparse voxels. Despite their architectural differences, these methods all share the same premise: geometry is first compressed into a latent space and only then modeled by the generator. This paradigm carries two intrinsic limitations. First, VAE reconstruction removes fine geometric detail before the diffusion model ever sees the data. Second, the two-stage pipeline creates an objective mismatch, since the VAE is trained for reconstruction whereas the diffusion model is trained for distribution matching. We take a different route and propose Direct3D-V3, a latent-free diffusion transformer

© 2026 Copyright held by the owner/author(s).
ACM 1557-7368/2026/12-ART244
https://doi.org/10.1145/3842524

(DiT) framework that discards the VAE altogether and operates in the native voxel domain, directly modeling raw SDF voxel grids during both training and sampling. To make this direct voxel formulation geometrically reliable, we introduce an implicit manifold regularization over neighboring voxels to suppress non-watertight surfaces, and convert predicted SDF grids into meshes via Marching Cubes so that rendered normal losses can supervise the model directly from its predicted SDF. To improve alignment between the conditioning image and generated geometry, Direct3D-V3 also predicts the camera parameters of the conditioning image as an auxiliary task, encouraging the DiT to learn the mapping from image observations to 3D shape. This latent-free design removes the reconstruction bottleneck, brings training under an objective in the native voxel domain, and yields meshes with substantially richer local detail. The predicted camera and native SDF voxel output further make Direct3D-V3 extensible to generative reconstruction, where diffusion is interleaved with normal-guided refinement to improve alignment with the observed view while the generative prior completes unseen regions. Extensive experiments show that Direct3D-V3 achieves competitive results to state-of-the-art latent-based methods, while producing finer geometry.

CCS Concepts: • **Computing methodologies → Artificial intelligence**.

Additional Key Words and Phrases: 3D generation, diffusion transformer, latent-free, native voxel domain

ACM Reference Format:
Shuang Wu, Jiachen Qian, Youtian Lin, Feihu Zhang, Philip Torr, Yao Yao, and Xun Cao. 2026. Direct3D-V3: Latent-Free Diffusion Transformers in the Native Voxel Domain. *ACM Trans. Graph.* 45, 6, Article 244 (December 2026), 18 pages. https://doi.org/10.1145/3842524

1 Introduction

Recent years have seen rapid progress in 3D shape generation, driven by advances in generative modeling and large-scale 3D datasets. This progress has opened up a wide range of applications, from game and film asset creation to AR/VR content authoring and robotic simulation. In many of these settings, the practical goal is to recover a high-fidelity 3D mesh from reference images while preserving faithful geometry, fine surface details, and alignment to the observed view.

The dominant recipe for this problem is latent diffusion [Rombach et al. 2022]. A variational autoencoder (VAE) is first trained to compress 3D shapes into a latent space, and a diffusion model is then trained to generate within that space. Following this recipe, recent work has largely focused on how to parameterize the VAE latent space. Existing designs can be broadly organized into vecset representations [Zhang et al. 2023, 2024], triplane [Wu et al. 2024], and structured sparse-voxel latents [Li et al. 2026a; Wu et al. 2026; Xiang et al. 2025]. Although these designs encode geometry differently, they share the same premise: geometry is first compressed into a learned latent representation and only then modeled by the generator. This latent paradigm carries two limitations for high-fidelity geometry generation. First, VAE reconstruction acts as an information bottleneck: fine details lost during encoding cannot be recovered by the downstream diffusion model. Second, the two-stage pipeline introduces an objective mismatch, because the VAE is optimized for reconstruction whereas the diffusion model is optimized to match the learned latent distribution. The generator is therefore constrained by both the fidelity of the learned representation and the gap between independently trained objectives. These observations suggest that further refinement of latent design is unlikely to remove either limitation, and motivate us to revisit the premise of latent compression itself.

We therefore take a different route and propose Direct3D-V3, a latent-free diffusion transformer (DiT) framework that operates in the native voxel domain by directly modeling raw SDF voxel grids during both training and sampling. This places the generator in the same signal space where geometry is ultimately evaluated, but also makes scalability the central challenge: a dense SDF volume grows cubically with resolution. To address this, we adopt a dual-granularity architecture: a patch-level DiT reasons about coarse semantics and global layout on a sparse sequence of patch tokens, while a lightweight voxel-level DiT restores fine geometric detail at voxel granularity. This coarse-to-fine organization reflects a broader principle in high-resolution pixel-space diffusion [Wang et al. 2025; Yu et al. 2025]: global structure and high-frequency detail are easier to model when they are handled at compatible granularities. For 3D geometry, surface sparsity makes this principle particularly effective, since the most informative signed-distance variation lies near the zero-level set. Concretely, the patch-level model operates on surface patches, while the voxel-level model predicts high-frequency details within selected patches. Together, these modules make native SDF diffusion tractable at 1536^3.

Operating directly on SDF voxels introduces a new optimization challenge, but also opens up a new form of geometric supervision. On the one hand, even small perturbations in predicted SDF values can cause Marching Cubes to extract non-watertight surfaces with open boundaries, a failure mode that VAE decoders tend to suppress implicitly by learning the manifold structure of meshes. We address this issue with an implicit manifold regularization loss that penalizes abrupt SDF discontinuities between neighboring voxels, substantially reducing non-watertight artifacts in practice. On the other hand, once the DiT predicts raw SDFs, each grid can be converted directly into a mesh via Marching Cubes, allowing rendered normal losses to supervise the SDF voxel output through the extracted mesh. A related idea was explored by Hi3DGen [Ye et al. 2025], but its reliance on VAE decoding for normal rendering incurs substantial memory overhead, limiting its applicable resolution to 256^3. By contrast, our approach operates without such a bottleneck, scaling to resolutions as high as 1536^3, and the resulting normal supervision yields measurable improvements in both the visual quality and geometric fidelity of the generated meshes. To improve alignment between the conditioning image and generated geometry, Direct3D-V3 also predicts the camera parameters of the conditioning image as an auxiliary task, encouraging the DiT to learn the mapping from image observations to 3D shape.

With the predicted camera and native SDF voxel output, Direct3D-V3 can be extended to generative reconstruction. The camera supplies the input view for test-time refinement, while the raw SDF grid provides geometry that can be optimized without repeatedly passing through a heavy VAE decoder. Building on this interface, we first use a pre-trained normal prediction model to estimate the normal map of the input image. We then extract a mesh from the predicted SDF via Marching Cubes, render the mesh normal map under the predicted camera, and optimize the discrepancy between the rendered normal and the predicted normal. We interleave this

normal-based refinement with diffusion sampling, allowing the reconstruction signal to tighten alignment with the input image while the generative prior continues to complete unseen regions and regularize the surface.

In summary, the main contributions of this paper are:

- We propose Direct3D-V3, a latent-free DiT framework that trains and samples directly on raw SDF voxel grids in the native voxel domain and scales to a resolution of 1536^3 through a dual-granularity architecture.
- We equip the native voxel formulation with three complementary signals: implicit manifold regularization for SDF consistency, rendered normal supervision for surface detail, and an auxiliary camera branch for alignment between the conditioning image and generated geometry.
- We further show that the latent-free design extends naturally to generative reconstruction, enabling effective normal-guided test-time refinement while preserving the geometric prior learned by the generator.
- Extensive experiments on image-to-3D generation show that Direct3D-V3 achieves competitive or superior performance to state-of-the-art latent-based methods, while producing meshes with finer geometric detail and better alignment with the input image.

2 Related Work

2.1 Native 3D Generation

Recent progress in 3D generation has been dominated by latent diffusion pipelines, in which a variational autoencoder first compresses geometry into a compact latent space and a diffusion model is then trained to generate in that space [Rombach et al. 2022]. Within this paradigm, one important line of work builds on vecset [Zhang et al. 2023] representations, which provide an implicit latent description of 3D shape through a fixed-length set of tokens that can be decoded into neural fields. CLAY [Zhang et al. 2024] scales this representation to controllable large-scale 3D asset generation. TripoSG [Li et al. 2025] further combines vecset latents with large-scale rectified flow training to improve synthesis fidelity. Dora [Chen et al. 2025a] and Hunyuan3D 2.0 [Zhao et al. 2025] study this line from the VAE perspective, showing that sharp edge sampling and stronger cross-attention improve the preservation of fine geometric structures during shape autoencoding.

Another representative family uses explicit triplane features. Direct3D [Wu et al. 2024] factorizes the latent into three orthogonal feature planes, injecting explicit positional structure into the latent representation and making diffusion modeling easier to scale than purely implicit token sets. Hyper3D [Guo et al. 2025] further augments the triplane latent with a low-resolution 3D grid latent, which supplements explicit 3D information and supports higher resolution representation with more efficient training.

A third and rapidly growing family uses sparse voxel structures as the latent representation. TRELLIS [Xiang et al. 2025] proposes the unified structured latent representation, which can be decoded into multiple downstream 3D formats, including NeRF [Mildenhall et al. 2021], FlexiCubes [Shen et al. 2023], and 3D Gaussian Splatting [Kerbl et al. 2023]. Direct3D-S2 [Wu et al. 2026] introduces spatial sparse attention, which significantly improves both training and inference efficiency for high resolution generation. SparseFlex [He et al. 2025b] focuses on high resolution shape modeling with arbitrary topology, extending sparse structured generation to more flexible geometric outputs. Sparc3D [Li et al. 2026a] introduces a sparse deformable marching cubes representation that can efficiently convert raw meshes to high resolution watertight ones. TRELLIS2 [Xiang et al. 2026] introduces a new sparse voxel structure, O-Voxel, that can jointly encode geometry and appearance in a compact native representation. Despite their differences in architecture and engineering, all of these methods still rely on compressing geometry into a learned latent space before generation. Our work departs from this common assumption and asks whether the latent stage can be removed entirely.

2.2 Latent-Free Diffusion Models

The idea of training diffusion models directly in data space has deep roots in image generation. Early models such as DDPM [Ho et al. 2020], Improved DDPM [Nichol and Dhariwal 2021], ADM [Dhariwal and Nichol 2021], and EDM [Karras et al. 2022] all operate directly in pixel space and establish the effectiveness of end-to-end denoising in the original signal domain. As model scale increased, however, many works shifted to latent space for efficiency. LDM [Rombach et al. 2022] made latent compression the dominant recipe for high-resolution diffusion, and transformer-based generators such as DiT [Peebles and Xie 2023] and U-ViT [Bao et al. 2023] further strengthened this trend by showing that large transformer backbones remain highly effective when operating on compact latent tokens.

Recent work has returned to the question of whether latent compression is necessary at all. JiT [Li and He 2025] argues that diffusion models should directly predict clean data rather than noise-like quantities, and shows that simple large patch transformers can already become strong pixel-space generators without tokenizers or auxiliary losses. DiP [Chen et al. 2025b] studies how to make pixel-space diffusion practical at higher resolution by combining a DiT backbone with a lightweight patch detailer head, balancing global structure modeling with local detail synthesis. DeCo [Ma et al. 2025] observes that modeling low-frequency semantics and high-frequency details with a single transformer is inefficient, and therefore decouples them using a semantic DiT, a lightweight pixel decoder, and a frequency-aware flow matching objective. PixelDiT [Yu et al. 2025] instead adopts a fully transformer-based dual-level design, where a patch-level DiT captures global semantics and a pixel-level DiT restores texture details directly in pixel space. These methods show that latent-free diffusion can be competitive for image generation, but they cannot be directly transferred to 3D generation, where SDF voxel grids are sparse, memory-intensive, and coupled to mesh extraction. SDF-Diffusion [Shim et al. 2023] provides an early 3D counterpart by avoiding VAE compression, yet its generation is limited to 128^3 resolution, which restricts the geometric precision of the resulting shapes. Our Direct3D-V3 extends the latent-free principle to high resolution native 3D generation by operating on surface SDF voxel grids and scaling to resolutions such as 1536^3.

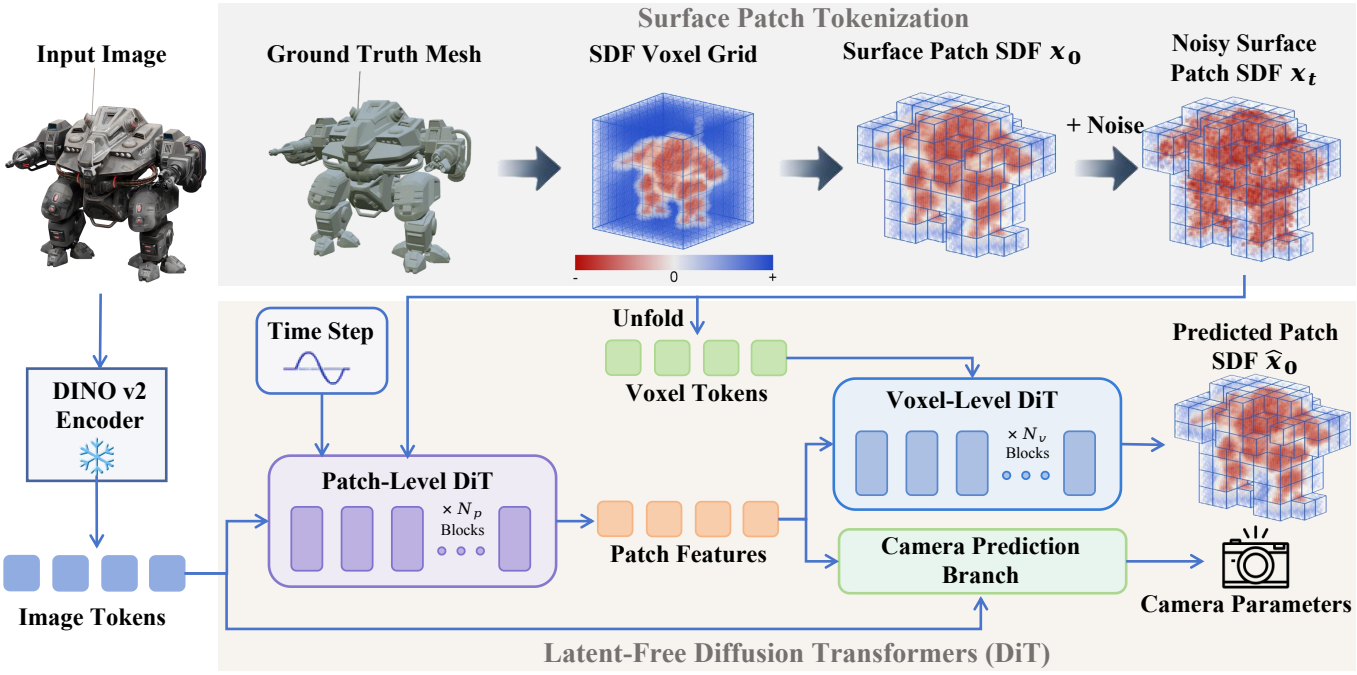


Fig. 2. Framework of Direct3D-V3. Rather than compressing geometry into a learned latent representation, Direct3D-V3 trains and samples directly on raw SDF voxel grids. During training, we extract the SDF voxel grid and surface patch SDF values from the ground-truth mesh, and add Gaussian noise to the surface patch SDF values to form the DiT input $\mathbf{x}_t$. During inference, the first-stage model is used only to provide the surface patch indices; sampling then starts from Gaussian noise on these sparse patches, and the latent-free DiT predicts the clean SDF values directly in the native voxel domain. The patch-level DiT uses the noisy SDF signal, DINOv2 image tokens, and timestep embedding to reason efficiently over global shape semantics, while a lightweight voxel-level DiT refines voxel-level SDF values within each selected patch. In parallel, a camera prediction branch combines patch-level features with image tokens to regress normalized intrinsic and extrinsic parameters, encouraging the generated shape to remain spatially aligned with the conditioning image.

2.3 Generative Reconstruction

One important line of work uses image diffusion priors to improve image-to-3D reconstruction. Zero-1-to-3 [Liu et al. 2023] learns viewpoint-conditioned novel view synthesis from a single image and shows that a diffusion prior with explicit relative camera control can already support zero-shot 3D reconstruction. SyncDreamer [Liu et al. 2024] improves this direction with a synchronized multi-view diffusion model that generates all target views jointly and correlates them through 3D-aware feature attention, substantially improving cross-view consistency. Wonder3D [Long et al. 2024] predicts both multi-view color images and multi-view normal maps through cross-domain diffusion, then fuses these geometric cues into textured mesh reconstruction. InstantMesh [Xu et al. 2024] combines an off-the-shelf multi-view diffusion model with a sparse-view large reconstruction model and a differentiable iso-surface extraction module, enabling high-quality mesh generation in only a few seconds. These methods significantly improve coarse structure recovery, novel view consistency, and surface appearance, but their optimization is still driven primarily by image-space consistency rather than direct alignment of fine geometry with the observed input view.

Another related line exploits video diffusion priors for view synthesis and reconstruction. Stable Virtual Camera [Zhou et al. 2025] formulates generative view synthesis as a general diffusion model conditioned on arbitrary input views and target camera trajectories, producing temporally smooth novel views under large viewpoint changes. GEN3C [Ren et al. 2025] further introduces a 3D cache built from depth predicted on seed images or previous frames, and conditions video generation on renderings of this cache under user-specified camera trajectories. This design yields more precise camera control and stronger temporal 3D consistency than prior video diffusion models. These methods offer richer camera-aware priors than single-image diffusion, but their supervision still lives mainly in image or video space and is not directly tied to mesh-level geometric detail.

Closest to our setting are methods that place reconstruction constraints inside generation itself. ReconViaGen [Chang et al. 2025] designs global geometry conditions for coarse structure, local per-view conditions for fine geometry and texture, and rendering-aware velocity compensation to correct the denoising trajectory toward the input views. CUPID [Huang et al. 2025] uses a two-stage flow pipeline: it first predicts an occupancy cube and a UV cube to estimate the object-centric camera, then uses a pose-aligned conditioner to inject pixel-aligned image features for detailed geometry and appearance. Pixal3D [Li et al. 2026b] generates 3D assets in a view-aligned space to better preserve image-to-3D correspondence and input details. Our method follows this direction, but benefits from a simpler native interface: the latent-free DiT directly outputs

SDF voxel grids rather than an intermediate latent representation. This allows the generated geometry to be converted into a mesh by Marching Cubes, rendered under the predicted camera, and directly aligned to normal maps estimated from the input view by a pre-trained normal predictor. As a result, our framework can push fine geometric detail to match the input image more directly, rather than focusing mainly on overall structure or texture.

3 Latent-Free Diffusion Transformers for 3D Generation

Existing latent-based methods compress 3D shapes into an intermediate latent space, introducing an unavoidable fidelity gap between the original geometry and what the diffusion model ultimately learns to generate. To circumvent this fundamental limitation, Direct3D-V3 performs training and sampling directly on raw SDF voxel grids, eliminating the VAE entirely, as detailed in Section 3.1. To balance geometric expressiveness with computational tractability at high resolutions, Direct3D-V3 employs two complementary DiT modules operating at patch and voxel granularity respectively, decoupling coarse shape semantics from fine-grained surface detail, as described in Section 3.2. To ensure the geometric validity of generated shapes, we introduce an implicit manifold regularization that enforces smooth SDF transitions between neighboring voxels, detailed in Section 3.3. Since the model predicts SDF voxel grids directly, we can also extract meshes with Marching Cubes and apply rendered normal supervision in the geometric space, as detailed in Section 3.4. Finally, the framework additionally predicts the camera parameters of the conditioning image, which pushes the generated mesh to align more closely with the input view, as elaborated in Section 3.5. An overview of the framework is shown in Figure 2.

3.1 Latent-Free Diffusion on SDF Voxel Grids

Latent diffusion has become a practical way to reduce the cost of high-dimensional generation, but in 3D shape generation the compression stage also constrains the geometry that the diffusion model can learn. A 3D VAE must encode detailed surface structures into a compact representation and later decode them back to geometry, which can discard fine local detail and introduce the reconstruction bottleneck discussed above. The latent space is also optimized for reconstruction fidelity rather than for the distributional objective used by diffusion, leading to a mismatch between the two training stages. We therefore discard the latent space and train the diffusion model directly in the voxel domain, using raw SDF voxel grids as the generative state.

Geometric Formulation on SDF Voxel Grids. We represent a 3D shape as a uniform SDF voxel grid $\mathbf{X} \in \mathbb{R}^{R\times R\times R\times 1}$, where each voxel stores a SDF value, and R denotes the grid resolution. While our latent-free formulation is in principle compatible with other volumetric representations such as FlexiCubes [Shen et al. 2023], Sparc3D [Li et al. 2026a], and O-Voxel [Xiang et al. 2026], we deliberately adopt this minimal representation to validate our core hypothesis that latent-free diffusion can match or surpass latent-based methods without relying on any sophisticated representation design.

Surface Patch Tokenization. Directly operating on a dense voxel grid of resolution R^3 is computationally prohibitive for the attention mechanisms in DiT. To reduce the token sequence length, we partition the voxel grid into non-overlapping patches of size p^3, yielding $(\frac{R}{p})^3$ patch tokens, each of dimensionality p^3. However, even with patch tokenization, the number of tokens at high resolutions such as 1536^3 remains large. We further exploit the observation that SDF values are only meaningful in the vicinity of the mesh surface, and adopt a sparse surface voxel representation that restricts computation to surface-adjacent patches. Concretely, we follow the two-stage generation paradigm that has become standard practice in high-resolution 3D generation methods such as TRELLIS [Xiang et al. 2025], Direct3D-S2 [Wu et al. 2026], and LATTICE [Lai et al. 2025]. In the first stage, we employ a model to rapidly generate a coarse mesh at low resolution, from which we identify the set of occupied surface patches $\mathcal{S}$. In the second stage, our latent-free DiT performs diffusion exclusively over these sparse patches, operating on the token sequence $\left\{\mathbf{x}_i \in \mathbb{R}^{p^3} \mid i \in \mathcal{S}\right\}$, where $|\mathcal{S}| \ll (R/p)^3$. The first-stage model is only used to localize surface patches and is not a focus of this paper. In practice, this stage can be implemented in the same spirit as existing high-resolution multi-stage pipelines: one may generate a coarse mesh with a vecset-based model and extract the corresponding patch indices, or predict the occupied patch set

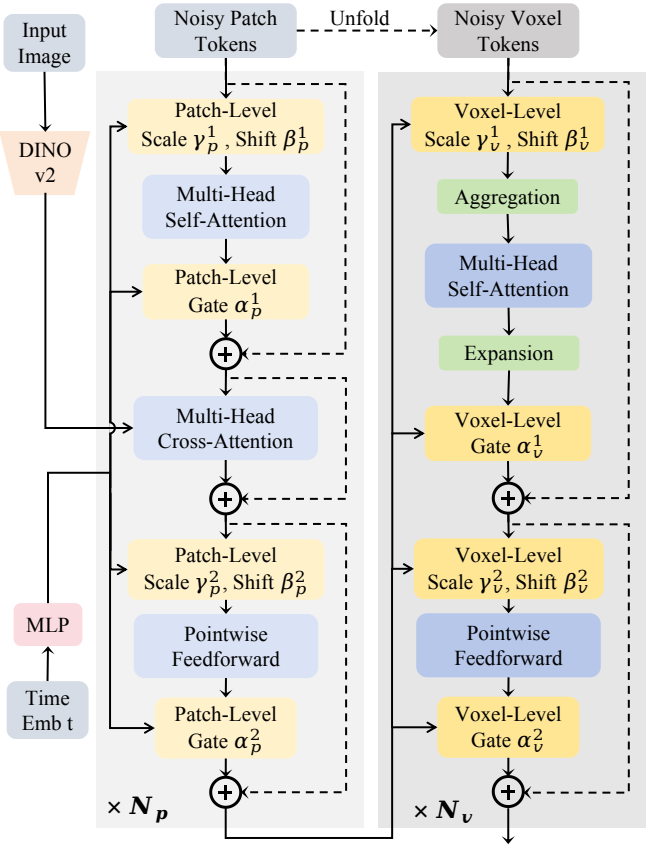


Fig. 3. The architecture of the dual-granularity DiT.

directly with a dense voxel DiT as in TRELLIS. Our method is agnostic to this choice, as long as it provides the surface patch index set $\mathcal{S}$ for the high-resolution latent-free DiT.

Native Flow Matching in the Geometric Domain. Unlike latent diffusion models that perform denoising in an abstract latent space, our DiT operates natively in the geometric domain, learning to denoise raw SDF voxel grids directly. We train our DiT using the flow matching [Albergo and Vanden-Eijnden 2022; Lipman et al. 2022] objective, which constructs a probability path between Gaussian noise and the data distribution via the linear interpolation schedule:

$$\mathbf{x}_t = t \cdot \mathbf{x}_0 + (1-t) \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad t \in [0,1], \tag{1}$$

with ground-truth velocity $\mathbf{v}_t = \mathbf{x}_0 - \boldsymbol{\epsilon}$. Following the x_0-prediction paradigm of JiT [Li and He 2025], our network f_θ directly predicts the clean voxel grid $\hat{\mathbf{x}}_0$ from the noisy observation $\mathbf{x}_t$ at timestep t:

$$\hat{\mathbf{x}}_0 = f_\theta(\mathbf{x}_t,\, t,\, \mathbf{c}), \tag{2}$$

where $\mathbf{c}$ denotes the conditioning signal. The predicted $\hat{\mathbf{x}}_0$ is converted to the estimated velocity via:

$$\hat{\mathbf{v}}_t = \frac{\hat{\mathbf{x}}_0 - \mathbf{x}_t}{1-t}, \tag{3}$$

and the flow matching loss is defined as:

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{t,\,\mathbf{x}_0,\,\boldsymbol{\epsilon}} \left[\|\hat{\mathbf{v}}_t - \mathbf{v}_t\|^2 \right]. \tag{4}$$

3.2 Dual-Granularity DiT Architecture

While sparse patch tokenization substantially reduces the sequence length for attention computation, it introduces a tension between efficiency and geometric fidelity. A large patch size keeps the number of tokens manageable, but results in patch embeddings of high dimensionality p^3, making it difficult for a single DiT to simultaneously model global shape semantics and recover fine-grained surface detail. Conversely, operating directly at voxel granularity preserves geometric detail but renders attention computation intractable. Inspired by the frequency decoupling strategy of DDT [Wang et al. 2025] and PixelDiT [Yu et al. 2025], which separate low-frequency semantic content from high-frequency detail, we propose a Dual-Granularity DiT architecture that decomposes this challenge across two complementary modules: a patch-level DiT that models coarse shape semantics and global layout at patch granularity, and a lightweight Voxel-level DiT that restores fine-grained geometric detail at voxel granularity, conditioned on the output of the patch-level DiT. The architecture is illustrated in Figure 3.

Patch-Level DiT. The Patch-level DiT, which comprises N_p DiT blocks, operates on the patch token sequence $\left\{\mathbf{x}_i \in \mathbb{R}^{p^3} \mid i \in \mathcal{S}\right\}$ and is responsible for modeling the semantic structure and global layout of the generated shape. Within each block, AdaLN injects the diffusion timestep embedding t by predicting global modulation parameters $(\gamma_p^1, \beta_p^1, \alpha_p^1, \gamma_p^2, \beta_p^2, \alpha_p^2)$, which jointly normalize and modulate the patch-token features during denoising. A cross-attention layer incorporates the image conditioning $\mathbf{c}$, extracted by a DINOv2-Large [Oquab et al. 2023] encoder, and a self-attention layer models the dependencies among patch tokens, capturing the global structural relationships of the 3D shape. The output of the patch-level DiT is a sequence of patch features $\mathbf{s}_p \in \mathbb{R}^{L \times D_p}$, where $L = |\mathcal{S}|$ denotes the number of sparse patches and D_p is the hidden size.

Voxel-Level DiT. While the patch-level DiT captures global structure efficiently, its large patch granularity makes it difficult to preserve fine geometric variation inside each patch. We therefore introduce a lightweight Voxel-Level DiT that refines local surface detail under the guidance of the patch-level output. For the L sparse patches selected by the first stage, the input SDF values to the voxel branch are arranged as $\mathbf{X}_v \in \mathbb{R}^{L \times p^3 \times 1}$. We reshape them into a voxel token sequence of length $L \cdot p^3$ and map each scalar SDF value to a voxel-level embedding $\mathbf{H}^{(0)} \in \mathbb{R}^{(L \cdot p^3) \times D_v}$ through a linear projection, where D_v denotes the hidden dimension of the voxel branch. The patch-level output $\mathbf{s}_p \in \mathbb{R}^{L \times D_p}$ serves as the conditioning signal for local refinement. At the l-th Voxel-Level DiT block, a linear layer takes $\mathbf{s}_p$ as input and predicts voxel-level AdaLN parameters

$$\mathbf{m}^{(l)} = g^{(l)}(\mathbf{s}_p) \in \mathbb{R}^{L \times (p^3 \cdot 6 D_v)}. \tag{5}$$

After reshaping, these parameters provide the modulation terms for each voxel token, written analogously as $[\gamma_v^1, \beta_v^1, \alpha_v^1, \gamma_v^2, \beta_v^2, \alpha_v^2]$ and used together in AdaLN to normalize and modulate each voxel-token feature. In this way, the patch-level feature does not merely provide a single coarse conditioning vector, but modulates all voxel tokens inside the patch at every block. Direct self-attention over all $L \cdot p^3$ voxel tokens is still unnecessarily expensive. We therefore reshape the voxel features back to patch form, aggregate the p^3 voxel tokens within each patch into one compact patch token before self-attention, and expand them back afterward:

$$\begin{aligned} \mathbf{U}^{(l)} &= \mathcal{A}_{\text{agg}}^{(l)}\left(\mathbf{H}_{\text{p}}^{(l)}\right), \\ \overline{\mathbf{U}}^{(l)} &= \text{SelfAttn}^{(l)}\left(\mathbf{U}^{(l)}\right), \\ \widehat{\mathbf{H}}_{\text{p}}^{(l)} &= \mathcal{A}_{\text{exp}}^{(l)}\left(\overline{\mathbf{U}}^{(l)}\right). \end{aligned} \tag{6}$$

where $\mathbf{H}_{\text{p}}^{(l)} \in \mathbb{R}^{L \times p^3 D_v}$ is obtained by regrouping voxel tokens patch by patch. The aggregation module $\mathcal{A}_{\text{agg}}^{(l)}$ maps $\mathbb{R}^{p^3 D_v}$ to $\mathbb{R}^{D_p}$, and the expansion module $\mathcal{A}_{\text{exp}}^{(l)}$ maps $\mathbb{R}^{D_p}$ back to $\mathbb{R}^{p^3 D_v}$; both are lightweight projection modules implemented with linear layers, bias terms, and activation functions. The compact tokens $\mathbf{U}^{(l)}, \overline{\mathbf{U}}^{(l)} \in \mathbb{R}^{L \times D_p}$ attend across sparse patches, and the expanded features $\widehat{\mathbf{H}}_{\text{patch}}^{(l)} \in \mathbb{R}^{L \times p^3 D_v}$ are finally reshaped to $\mathbb{R}^{(L \cdot p^3) \times D_v}$. This aggregation and expansion design keeps the voxel branch lightweight while still allowing the model to restore fine surface geometry under the guidance of patch-level semantics.

Overall, the patch-level DiT provides global semantic structure over sparse surface patches, while the voxel-level DiT restores local SDF variation within each patch. This dual-granularity design avoids applying expensive attention to all voxel tokens directly, yet still preserves the fine geometric detail needed for high-resolution shape generation.

3.3 Implicit Manifold Regularization

Removing the VAE changes not only the representation space, but also the geometric prior available to the generator. In latent-based

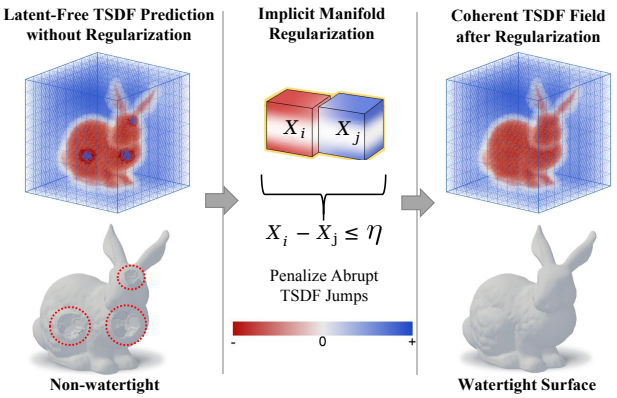


Fig. 4. Illustration of the Implicit Manifold Regularization (IMR). Without the decoder-induced geometric prior used in latent-based pipelines, a latent-free DiT can predict locally inconsistent TSDF fields, leading to abrupt zero-level crossings and non-watertight meshes after Marching Cubes. Since the surface manifold is implicitly defined by the zero level set of the TSDF, we regularize this manifold by penalizing neighboring voxels whose predicted SDF values differ by more than one voxel length η. This local constraint encourages coherent TSDF transitions around the surface, reducing holes and producing a more watertight extracted mesh while preserving the generated shape.

methods, the VAE decoder is trained to reconstruct valid 3D shapes and therefore implicitly learns the local structure of plausible SDF fields around mesh surfaces. Since the mesh surface is extracted as the zero level set of the SDF, this local distance-field structure acts as an implicit prior on the extracted surface manifold. As a result, even noisy or inaccurate compressed features can often be decoded into a watertight surface without large holes. A latent-free DiT lacks this decoder-induced prior: small local errors in predicted SDF values can create inconsistent zero crossings after Marching Cubes, producing meshes with holes or non-watertight regions. We address this issue with implicit manifold regularization, which does not directly constrain mesh topology, but regularizes the local TSDF field that implicitly defines the surface manifold.

Eikonal regularization is commonly used in 3D reconstruction to constrain learned implicit fields. In our setting, however, the DiT is directly supervised by ground-truth TSDF values, while the Eikonal loss enforces a unit 2-norm gradient at arbitrary spatial points. This constraint can conflict with the empirical TSDF distribution, making DiT optimization unstable and in severe cases causing training collapse. We therefore impose a discrete Lipschitz constraint on neighboring voxels, as illustrated in Figure 4. Let $\hat{\mathbf{X}}$ denote the predicted TSDF grid and let η be the physical length of one voxel. For each pair of neighboring voxels $(i, j) \in \mathcal{N}$, a valid TSDF field should not change by more than one voxel length. We therefore define the implicit manifold regularization loss as:

$$\mathcal{L}_{\text{imr}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} \left[\left|\hat{\mathbf{X}}_i - \hat{\mathbf{X}}_j\right| - \eta\right]_+^2, \tag{7}$$

where $[\cdot]_+ = \max(\cdot, 0)$ and $\mathcal{N}$ contains adjacent voxel pairs along the three grid axes. This loss penalizes only abrupt TSDF jumps that violate the local distance-field constraint, while leaving valid local variation unconstrained.

By injecting this implicit manifold prior directly into the latent-free DiT, the model learns to produce smoother and more coherent TSDF fields, which in turn improves the watertightness of meshes extracted by Marching Cubes. We include this term in the final training objective introduced at the end of the method section.

3.4 Native Geometric Optimization

The latent-free formulation enables supervision to be applied directly in the native geometric space. Since the DiT predicts the clean SDF voxel grid $\hat{\mathbf{x}}_0$ rather than an intermediate latent representation, each prediction can be immediately converted into a mesh through Marching Cubes. Given randomly sampled camera parameters π, we render a normal map from the extracted mesh and compare it with the normal map rendered from the ground-truth mesh under the same camera. This creates a direct geometric supervision signal on the generated surface, rather than supervising the model through a compressed proxy.

Formally, let

$$\hat{\mathbf{M}} = \text{MC}(\hat{\mathbf{x}}_0), \qquad \hat{\mathbf{N}}_\pi = \mathcal{R}_{\text{n}}\left(\hat{\mathbf{M}}, \pi\right), \tag{8}$$

where $\text{MC}(\cdot)$ denotes Marching Cubes and $\mathcal{R}_{\text{n}}(\cdot, \pi)$ denotes normal rendering under camera π. In practice, we use differentiable sparse Marching Cubes to extract meshes from the predicted sparse SDF grid, and render normal maps with Nvdiffrast under the specified camera parameters. This pipeline evaluates the supervision on the extracted surface while keeping gradients connected to the native SDF prediction. Let $\mathbf{N}_\pi^{\text{gt}}$ be the normal map rendered from the ground-truth mesh with the same camera. We define the native geometric optimization loss as

$$\mathcal{L}_{\text{geo}} = \left\|\hat{\mathbf{N}}_\pi - \mathbf{N}_\pi^{\text{gt}}\right\|_1. \tag{9}$$

This loss encourages the predicted SDF grid to produce meshes whose visible surface normals match the target geometry, improving the visual fidelity of generated meshes.

Normal-based supervision has also been explored by Hi3DGen [Ye et al. 2025], but it is applied in a latent-based pipeline. The DiT prediction in Hi3DGen must first be decoded by a VAE decoder before a mesh can be extracted and rendered for normal supervision. This additional decoding step is feasible at relatively low resolutions such as 256^3, but becomes memory-intensive at high resolutions such as 1536^3. In contrast, our latent-free DiT outputs the SDF voxel grid directly, so Marching Cubes and normal rendering can be applied without any VAE decoding. This makes native geometric optimization practical even at high resolution.

3.5 Camera Parameter Prediction

To strengthen the spatial correspondence between the input image and the generated geometry, we further add a camera prediction branch to the latent-free DiT. Since Direct3D-V3 targets object-level generation rather than scene reconstruction, we adopt an object-centric look-at camera model. In this setting, the object is assumed to be centered in a canonical coordinate frame, and the camera is

described by its viewing direction, distance, field of view, and image-plane offset. This avoids unnecessary free variables for object-level conditioning while still capturing the view variation needed to align the generated geometry with the conditioning image.

The branch starts from a learnable camera token $\mathbf{q}_{\text{cam}}$, which first attends to the patch-level DiT output $\mathbf{s}_p$ and then attends to the DINOv2 image tokens $\mathbf{z}$:

$$\bar{\mathbf{q}}_{\text{cam}} = \text{CrossAttn}\big(\text{CrossAttn}\big(\mathbf{q}_{\text{cam}}, \mathbf{s}_p\big), \mathbf{z}\big). \tag{10}$$

The updated camera token is then passed through an MLP to predict the following camera vector:

$$\hat{\boldsymbol{\kappa}} = \text{MLP}(\bar{\mathbf{q}}_{\text{cam}}), \qquad \boldsymbol{\kappa} = (sin(\theta), cos(\theta), \phi, \rho, \psi, c_x, c_y). \tag{11}$$

Here θ and ϕ denote azimuth and elevation, $\rho = \log d$ denotes log camera distance, $\psi = \log \tan fov$ denotes log field of view, and (c_x, c_y) are the principal point offsets. All the target parameters are normalized to $[-1, 1]$. We supervise this branch with

$$\mathcal{L}_{\text{cam}} = \left\| \hat{\boldsymbol{\kappa}} - \boldsymbol{\kappa}^{\text{gt}} \right\|_1. \tag{12}$$

This compact parameterization is well suited to single-object generation. The azimuth and elevation provide a normalized representation of view direction, while $\log d$ and $\log \tan fov$ stabilize the dynamic range of distance and field of view. The principal point offsets (c_x, c_y) capture image-plane shifts with only two normalized scalars. As an auxiliary task, camera prediction encourages the DiT to learn the mapping between image observations and 3D space, which improves the alignment between the generated mesh and the conditioning image.

The final training objective combines flow matching, implicit manifold regularization, native geometric optimization, and camera prediction:

$$\mathcal{L} = \mathcal{L}_{\text{fm}} + \lambda_{\text{imr}}\mathcal{L}_{\text{imr}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}} + \lambda_{\text{cam}}\mathcal{L}_{\text{cam}}, \tag{13}$$

where λ_{imr}, λ_{geo}, and λ_{cam} control the strengths of the corresponding auxiliary losses.

4 Application: Generative Reconstruction

The latent-free DiT can be used not only as a generator, but also as a native 3D prior for reconstruction. Given a single input image, standard image-to-3D generation samples a plausible complete shape conditioned on the image, but the generated surface may still deviate from the observed view in fine geometric detail. Our framework exposes two quantities that make test-time reconstruction natural: the DiT predicts the camera parameters of the conditioning image, and its clean prediction is the SDF voxel grid itself. We therefore refine the sampling trajectory directly in the SDF domain by comparing normals rendered from the predicted mesh with normals estimated from the input image. This turns the generator into a reconstruction prior: the diffusion trajectory maintains plausible unseen geometry, while the visible surface is corrected by an explicit geometric signal from the input view. The generative reconstruction pipeline is illustrated in Figure 5.

4.1 Native Geometry as a Reconstruction Prior

The central advantage of our latent-free formulation is that every clean DiT prediction can be immediately treated as geometry. Let $\mathbf{I}$ be the input image, $\mathbf{M}_\mathbf{I}$ be its foreground mask, and Φ_{n} be a pretrained normal prediction model such as Lotus-2 [He et al. 2025a]. We estimate an input view normal map $\mathbf{N}_\mathbf{I} = \Phi_{\text{n}}(\mathbf{I})$, which provides a dense geometric target over the visible object region. During sampling, the DiT predicts a clean SDF voxel grid $\hat{\mathbf{x}}_0$ together with an initial camera estimate $\hat{\boldsymbol{\kappa}}$:

$$(\hat{\mathbf{x}}_0, \hat{\boldsymbol{\kappa}}) = f_\theta(\mathbf{x}_t, t, \mathbf{I}). \tag{14}$$

Because $\hat{\mathbf{x}}_0$ is already a raw SDF grid, we can extract a mesh $\hat{\mathbf{M}} = \text{MC}(\hat{\mathbf{x}}_0)$ and render it under $\hat{\boldsymbol{\kappa}}$ without invoking a VAE decoder.

This native geometry path is what makes the reconstruction procedure practical. In a latent-based pipeline, each inner optimization step must decode an intermediate latent representation into geometry before mesh extraction and rendering, which makes iterative test-time refinement expensive at high resolution. In our framework, Marching Cubes and differentiable rendering are applied directly to the DiT output, and gradients can be propagated back to the SDF grid itself. The normal predictor supplies a view-aligned geometric observation, the camera branch supplies the approximate image-to-shape pose, and the latent-free SDF output supplies the optimizable geometry. Together, these components form a closed reconstruction loop in the native geometry domain.

4.2 Observation Alignment in the Predicted View

Before applying normal guidance, we align the predicted view with the input observation. Although the camera branch is trained to predict the relative camera parameters of the conditioning image, small errors in azimuth, elevation, distance, field of view, or image center can shift the rendered normal map relative to the input. Optimizing SDF values under such a misaligned view would push the geometry in the wrong direction. We therefore first refine the predicted camera using the foreground mask, which is robust to local normal ambiguity.

Starting from the DiT prediction $\hat{\boldsymbol{\kappa}}$, we render the geometry mask of the predicted mesh and compare it with the input foreground mask:

$$\mathcal{L}_{\text{mask}}(\boldsymbol{\kappa}) = \|\mathcal{R}_{\text{m}}(\text{MC}(\hat{\mathbf{x}}_0), \boldsymbol{\kappa}) - \mathbf{M}_\mathbf{I}\|_2^2, \tag{15}$$

where $\mathcal{R}_{\text{m}}$ denotes differentiable mask rendering. The corrected camera $\boldsymbol{\kappa}^\star$ is obtained by gradient descent on the camera vector, initialized from $\hat{\boldsymbol{\kappa}}$. In practice, the DiT already predicts a reliable pose, so this step is a local calibration rather than a full pose search; about 50 iterations are sufficient for convergence in our setting. Importantly, this calibration modifies only the camera parameters and leaves the generated geometry unchanged.

With the corrected camera $\boldsymbol{\kappa}^\star$, the input normal map becomes an aligned geometric target for SDF refinement. Given a current SDF grid $\mathbf{x}$, we extract a mesh by Marching Cubes, render its normal map under $\boldsymbol{\kappa}^\star$, and compare it with the predicted input normals inside the foreground mask:

$$\mathcal{L}_{\text{rec}}(\mathbf{x}) = \left\| \mathbf{M}_\mathbf{I} \odot \left(\mathcal{R}_{\text{n}}\big(\text{MC}(\mathbf{x}), \boldsymbol{\kappa}^\star\big) - \mathbf{N}_\mathbf{I}\right) \right\|_1, \tag{16}$$

where $\mathcal{R}_{\text{n}}$ denotes normal rendering. The foreground mask prevents background pixels from affecting the update, while the normal loss focuses the reconstruction signal on visible surface orientation rather than color or texture. This design directly pushes geometric details in the observed view to agree with the input image.

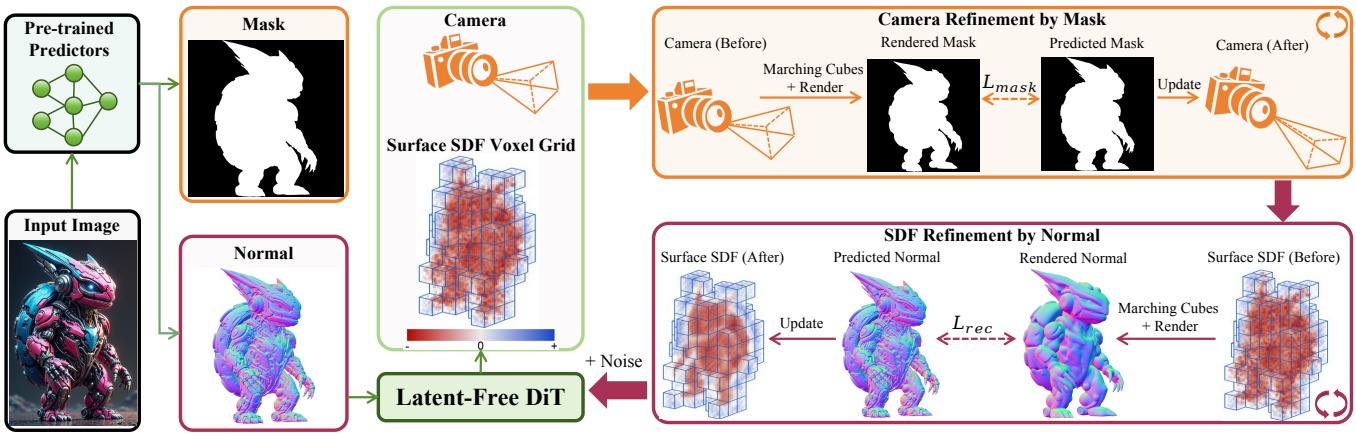


Fig. 5. Generative reconstruction pipeline. Given the input images, we use pre-trained predictors to estimate the corresponding foreground mask and normal map as auxiliary observations. The key reconstruction mechanism lies in how these cues are coupled with the latent-free DiT output: the mask calibrates the predicted camera, while the normal map drives direct refinement of the predicted SDF voxel grid. At each guided sampling step, the DiT proposes a clean SDF grid together with an image-view camera. We first align the rendered mask with the estimated one to correct the camera, and then compare rendered surface normals with the estimated normal map to update the SDF values. The refined SDF is re-noised and passed back to the sampler, coupling observation-driven correction with the learned generative prior for the final reconstruction.

The SDF grid is updated by backpropagation through mesh extraction and rendering:

$$\mathbf{x}^{r+1} = \mathbf{x}^{r} - \alpha_x \nabla_{\mathbf{x}^r} \mathcal{L}_{\text{rec}}(\mathbf{x}^r). \tag{17}$$

Because the optimized variable is the raw SDF voxel grid, the update directly changes the zero-level set that will be extracted as the final mesh. This differs from latent refinement, where a gradient step updates an abstract latent code and the geometric effect depends on the decoder. Here, the normal loss, rendered mesh, and optimized SDF all live in the same native geometry space, making the update direct and efficient.

4.3 Generative Refinement along the Sampling Trajectory

We interleave SDF refinement with diffusion sampling rather than applying it only after generation finishes. We apply reconstruction guidance only when $t < t_1$: although the current noisy state $\mathbf{x}_t$ is not yet a final geometry, the DiT already predicts a clean SDF estimate $\hat{\mathbf{x}}_0$, which can be rendered and corrected to steer the subsequent generative trajectory. Once $t \geq t_1$, we stop reconstruction guidance and let the DiT continue standard denoising without additional test-time optimization.

At each timestep, we first determine the clean SDF estimate used for the next sampling update. We set $\mathbf{x}_0^\star$ to the refined SDF grid when reconstruction guidance is applied, and to the DiT prediction $\hat{\mathbf{x}}_0$ otherwise. We then re-noise $\mathbf{x}_0^\star$ using the same linear schedule as flow matching:

$$\mathbf{x}_{t_{\text{next}}} = t_{\text{next}} \cdot \mathbf{x}_0^\star + (1 - t_{\text{next}}) \cdot \boldsymbol{\epsilon}. \tag{18}$$

This update keeps reconstruction and generation coupled during the guidance window. The normal loss pulls the visible surface toward the input image in the early denoising stage, while the subsequent DiT steps continue to project the sample toward the learned shape distribution. As a result, the method improves alignment with the observed view without turning inference into unconstrained per-example fitting, and the generative prior continues to complete invisible regions and regularize the full shape.

Algorithm 1 summarizes the full procedure. The camera is calibrated once when the sampler first enters the early reconstruction window, because the DiT camera estimate is already close and the mask objective quickly removes the remaining alignment error. SDF refinement is then applied only for steps with $t < t_1$, each time using the corrected camera and the same input normal map. The final output of the algorithm is the last clean SDF voxel grid.

5 Experiments

5.1 Datasets and Evaluation Metrics

We train Direct3D-V3 on 320K meshes from a high-quality subset of Objaverse-XL [Deitke et al. 2023]. To examine data scalability, we also conduct an additional training experiment by adding approximately 200K private meshes to this public subset. For each mesh, we render 16 RGB images and 16 normal maps as conditioning images, with randomized viewpoints, illumination, and focal lengths.

For image-to-3D generation, we evaluate on 102 input images from real photographs, web-collected images, and AI-generated images. Since these inputs do not have ground-truth 3D shapes, we measure the consistency between each input image and the generated geometry using two pretrained 3D representation models: Uni3D [Zhou et al. 2024] and ULIP-2 [Xue et al. 2024]. These metrics compare the input-image feature with the generated-shape feature, providing complementary estimates of alignment between input images and generated geometry.

For module ablations, we use a held-out Objaverse-XL subset that has no overlap with the training data. To evaluate geometric quality more directly, we report Normal-FID, a normal-rendering variant of the Fréchet Inception Distance [Heusel et al. 2017]. Specifically, for

Algorithm 1: Generative Reconstruction with Native SDF Refinement

Input: Input image $\mathbf{I}$, foreground mask $\mathbf{M}_\mathbf{I}$, normal predictor Φ_n, latent-free DiT $f_{\boldsymbol{\theta}}$, reconstruction threshold t_1
Output: SDF voxel grid $\mathbf{x}_0^\star$
Prepare observation cues. Estimate the visible normal map from the input image: $\mathbf{N}_\mathbf{I} \leftarrow \Phi_\mathrm{n}(\mathbf{I})$;
Initialize generation. Draw the initial noisy SDF sampling state $\mathbf{x}_t$ from Gaussian noise;
Initialize reconstruction states. Set the calibrated camera and current clean SDF to empty: $\boldsymbol{\kappa}^\star \leftarrow \varnothing$, $\mathbf{x}_0^\star \leftarrow \varnothing$;
for *each sampling step* $t \rightarrow t_\mathrm{next}$ **do**
  Use the latent-free DiT to estimate the clean SDF grid and the camera of the conditioning image: $(\hat{\mathbf{x}}_0, \hat{\boldsymbol{\kappa}}) \leftarrow f_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \mathbf{I})$;
  if $t < t_1$ *and* $\boldsymbol{\kappa}^\star = \varnothing$ **then**
    Initialize the camera with the DiT pose prediction: $\boldsymbol{\kappa} \leftarrow \hat{\boldsymbol{\kappa}}$;
    for $r = 1$ **to** K_cam **do**
      $\mathcal{L}_\mathrm{mask} \leftarrow \|\mathcal{R}_\mathrm{m}(\mathrm{MC}(\hat{\mathbf{x}}_0), \boldsymbol{\kappa}) - \mathbf{M}_\mathbf{I}\|_2^2$;
      Update the camera parameters using $\nabla_{\boldsymbol{\kappa}} \mathcal{L}_\mathrm{mask}$;
    end for
    Store the calibrated camera: $\boldsymbol{\kappa}^\star \leftarrow \boldsymbol{\kappa}$;
  end if
  if $t < t_1$ **then**
    Start from the current clean SDF prediction: $\mathbf{x}_0^\star \leftarrow \hat{\mathbf{x}}_0$;
    for $r = 1$ **to** K_sdf **do**
      Render surface normals under the calibrated camera and compare them with the input normal target: $\mathcal{L}_\mathrm{rec} \leftarrow \|\mathbf{M}_\mathbf{I} \odot (\mathcal{R}_\mathrm{n}(\mathrm{MC}(\mathbf{x}_0^\star), \boldsymbol{\kappa}^\star) - \mathbf{N}_\mathbf{I})\|_1$;
      Update the SDF voxel values using $\nabla_{\mathbf{x}_0^\star} \mathcal{L}_\mathrm{rec}$ to move the visible zero-level set toward the observation;
    end for
  else
    Keep the DiT clean prediction unchanged: $\mathbf{x}_0^\star \leftarrow \hat{\mathbf{x}}_0$;
  end if
  Re-noise the selected clean SDF estimate for the next DiT step: $\mathbf{x}_{t_\mathrm{next}} \leftarrow t_\mathrm{next} \cdot \mathbf{x}_0^\star + (1 - t_\mathrm{next}) \cdot \boldsymbol{\epsilon}$;
end for
return $\mathbf{x}_0^\star$;

each generated mesh and its corresponding ground-truth mesh, we render normal maps from identical camera views and compute FID between the generated and ground-truth normal renderings. This protocol focuses the ablation study on shape fidelity and surface quality.

For generative reconstruction, we report quantitative results on Google Scanned Objects (GSO) [Downs et al. 2022]. We employ Chamfer Distance (CD), F1-score at 0.95 and 0.99, and Volume IoU to evaluate reconstruction quality. Since GSO contains relatively simple scanned objects and does not fully reflect reconstruction quality on complex geometry, we additionally use images generated by AI and images collected from online communities for qualitative comparison.

5.2 Implementation Details

Our latent-free DiT trains and samples directly on raw SDF voxel grids. We use a truncated SDF with a truncation threshold of $1/64$ and a patch size of $p = 8$. To obtain the surface patch indices, we follow the setting of Hunyuan3D 2.1 [Hunyuan3D et al. 2025] and train a vecset DiT as the first-stage generator, which quickly produces a coarse mesh for surface patch extraction. The patch-level DiT contains 17 blocks with hidden size 1536 and 16 attention heads, while the voxel-level DiT contains 2 blocks with voxel hidden size $D_v = 8$. All self-attention layers in the DiT use 3D shifted-window attention for efficiency. Blocks alternate between regular window attention and the same partition shifted by half a window on the 3D patch grid. We set each window to half of the patch grid resolution along every axis; the shifted partition lets consecutive layers cover offset neighborhoods and exchange information across window boundaries without global attention. Image conditioning features are extracted by DINOv2-Large from input images at resolution 1022. We train the model on 16 NVIDIA A800 GPUs with AdamW and bfloat16 mixed precision, and reduce memory use with gradient checkpointing and FSDP2. The loss weights are set to $\lambda_\mathrm{imr} = 1 \times 10^{-5}$, $\lambda_\mathrm{geo} = 1.0$, and $\lambda_\mathrm{cam} = 5 \times 10^{-2}$. Normal maps for geometric supervision are rendered at resolution 1024. During sampling, we use 15 flow-matching steps and classifier-free guidance scale 4.0. To stabilize high-resolution latent-free training, we adopt a progressive schedule from 256^3 to 1536^3 resolution, as summarized in Table 1.

Table 1. Progressive training schedule for Direct3D-V3.

Resolution	Learning rate	Batch size	Training Steps
256^3	1e-4	8×16	400K
512^3	1e-4	4×16	50K
1024^3	5e-5	1×16	50K
1536^3	5e-5	1×16	10K

For generative reconstruction, we use the same DiT checkpoint as a shape prior and refine the clean SDF prediction during sampling. The mask-based camera calibration uses $K_\mathrm{cam} = 50$ optimization steps. SDF refinement is applied when $t < t_1$, where $t_1 = 0.8$, using $K_\mathrm{sdf} = 100$ gradient steps. We use Lotus-2 as the normal predictor and render normals at resolution 1024 for the reconstruction loss.

5.3 Comparison with Image-to-3D Generation Methods

We evaluate Direct3D-V3 on the test images described in Section 5.1, comparing against recent open-source image-to-3D methods including Michelangelo [Zhao et al. 2023], Hunyuan3D 2.1 [Hunyuan3D et al. 2025], Direct3D-S2 [Wu et al. 2026], UltraShape [Jia et al. 2025], and TRELLIS2 [Xiang et al. 2026]. Table 2 reports quantitative results using Uni3D and ULIP-2, which measure consistency between the input image and generated shape in pretrained 3D feature spaces. Direct3D-V3 outperforms all competing methods on both metrics, and adding private training data further improves the scores. However, Uni3D and ULIP-2 mainly reflect semantic alignment with the conditioning image and cannot directly measure geometric precision, such as surface regularity, thin structures, and local details. We therefore conduct a user study with 1–5 quality ratings. As shown in Figure 6, Direct3D-V3 receives the highest user preference, indicating that users favor the geometric quality of our generated meshes.

Table 2. Quantitative comparison with recent image-to-3D methods. Higher is better for all metrics, and the best result is shown in bold. Ours* denotes Direct3D-V3 trained with the public training data plus approximately 200K additional private meshes.

Method	Uni3D ↑	ULIP-2 ↑
Michelangelo	0.3404	0.3605
Hunyuan3D 2.1	0.4187	0.4424
Direct3D-S2	0.4190	0.4485
UltraShape	0.4185	0.4510
TRELLIS2	0.4172	0.4405
Ours	0.4198	0.4535
Ours*	**0.4224**	**0.4547**

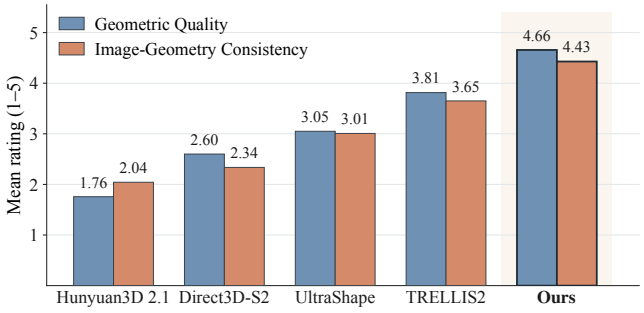


Fig. 6. User study results for image-to-3D generation. Scores range from 1 to 5, and higher is better.

Figure 7 provides a qualitative comparison of generated meshes. Hunyuan3D 2.1 usually captures the coarse object shape and produces stable overall geometry, but it tends to miss or smooth out fine structures, especially on hair, beard, ornaments, and other high-frequency regions. Direct3D-S2 generates richer local details, yet these details are sometimes less organized: surface patterns can become cluttered, and the generated geometry may drift from the conditioning image when the input contains dense structures or ambiguous contours. UltraShape follows the input silhouette well, but its geometric precision still has a visible ceiling on thin parts and high-frequency surface patterns. TRELLIS2 is the closest high-resolution competitor, since both TRELLIS2 and Direct3D-V3 scale to 1536^3 generation and can recover substantially more detail than lower-resolution methods. Even so, Direct3D-V3 produces more faithful and regular local geometry in several representative cases: it better preserves hair and beard structures in the first and fifth examples, sharper tire tread patterns in the second example, more complete facial details in the third example, and more regular ropes on the ship in the fourth example. Moreover, unlike TRELLIS2 and Direct3D-S2, our approach is implemented as a pure transformer architecture and does not require specialized sparse 3D convolution infrastructure, making the high-resolution generator simpler while remaining competitive with the strongest sparse 3D frameworks.

Table 3. Quantitative comparison on GSO generative reconstruction. The table reports Chamfer Distance (CD), F1-score at 0.95 and 0.99, and Volume IoU. The best result is shown in bold.

Method	CD ↓	$F1_{0.95}$ ↑	$F1_{0.99}$ ↑	Vol. IoU ↑
TRELLIS	0.0153	0.9586	0.5123	0.3315
CUPID	0.0141	0.9550	0.5728	0.3744
ReconViaGen	0.0194	0.9239	0.4585	0.2830
TRELLIS2	0.0147	0.9510	0.5934	**0.3884**
Ours	**0.0131**	**0.9600**	**0.6043**	0.3815

5.4 Generative Reconstruction Results

We further evaluate our Direct3D-V3 in the single-view generative reconstruction setting. We compare against TRELLIS [Xiang et al. 2025], TRELLIS2 [Xiang et al. 2026], CUPID [Huang et al. 2025], and ReconViaGen [Chang et al. 2025]. Given a single image, Direct3D-V3 first samples a complete shape from the learned prior, uses the predicted camera parameters to establish the input view, and refines the native SDF prediction with normal-guided test-time optimization. Since the refinement is applied directly to the SDF grid, the observed surface can be corrected in the geometry domain while the diffusion prior continues to regularize unobserved regions.

Table 3 reports reconstruction accuracy on GSO using CD, F1-score at 0.95 and 0.99, and Volume IoU. Direct3D-V3 achieves the best CD and F1 scores among all compared methods, while obtaining a competitive Volume IoU slightly below TRELLIS2. Although these results validate our reconstruction framework on a standard benchmark, GSO contains relatively simple scanned objects and does not fully reveal how different methods handle complex geometry. We therefore further compare on AI images and community images, which contain denser local structures, richer visual styles, and more challenging fine detail. As shown in Figure 8, CUPID, ReconViaGen, TRELLIS, and TRELLIS2 recover plausible complete shapes, but their surfaces are often over-smoothed or locally inconsistent with the input image on thin structures, concave parts, and fine details. Direct3D-V3 reconstructs more faithful visible surface geometry, with stronger alignment to the input image, sharper local details, and coherent global completion.

We also extend our framework to multi-view inputs with minimal changes. We concatenate the DINOv2 image tokens extracted from all views and inject them into the latent-free DiT through the same cross-attention layers, fine-tuning the model for 20K steps. At inference time, we use the mesh generated by the TRELLIS MultiDiffusion pipeline to obtain the surface patch indices. We evaluate this setting on scenes from the standard Tanks and Temples dataset [Knapitsch et al. 2017]; the qualitative results are shown in Figure 9, where our method reconstructs meshes with sharper and more complete surface details than competing approaches.

5.5 Ablation Study

All ablations are conducted on the held-out Objaverse-XL subset described in Section 5.1. By default, we extract surface patch indices directly from the ground-truth meshes to isolate the ablation results from errors introduced by the first-stage generator. We report

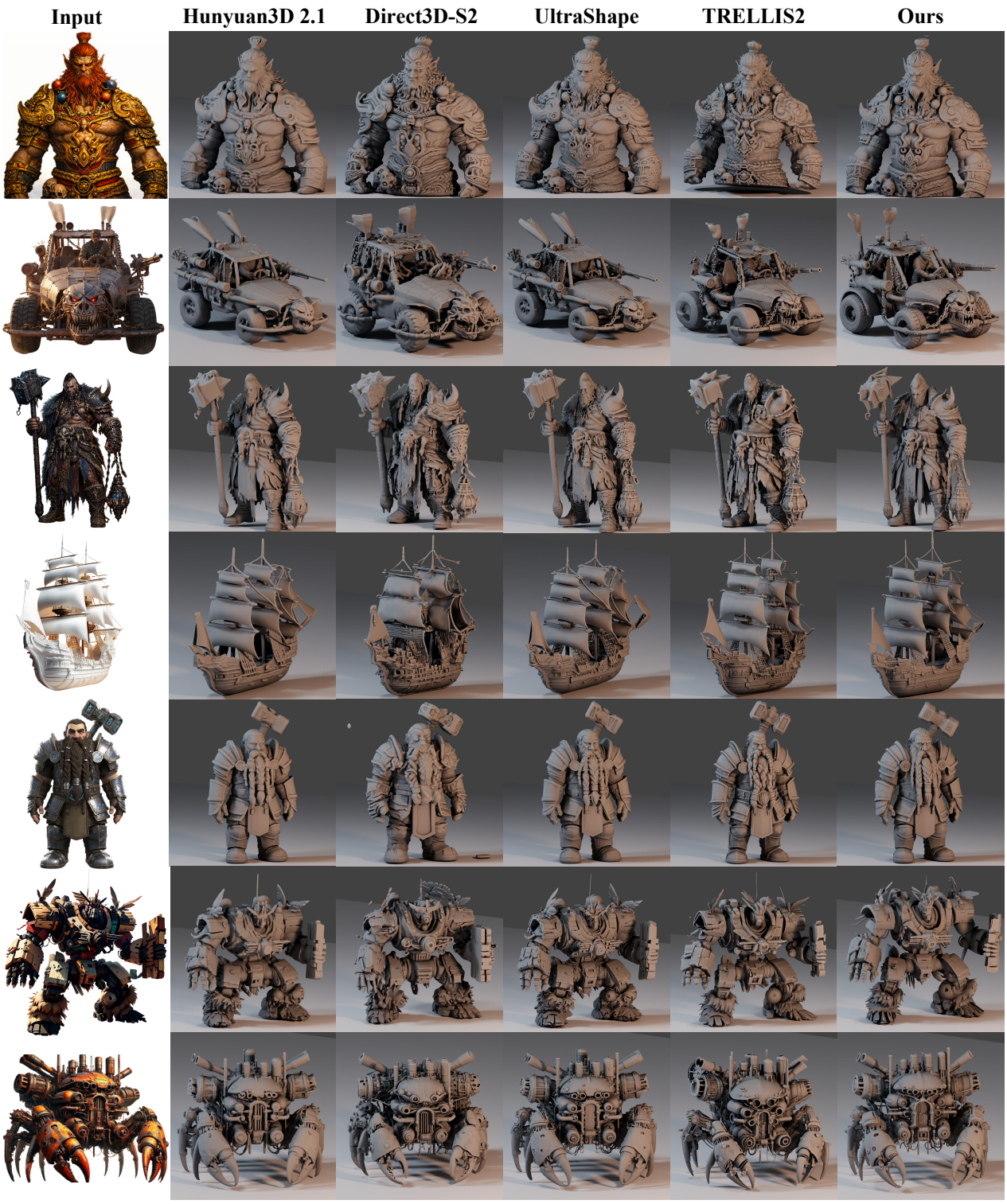


Fig. 7. Qualitative comparison with recent open-source image-to-3D generation methods.

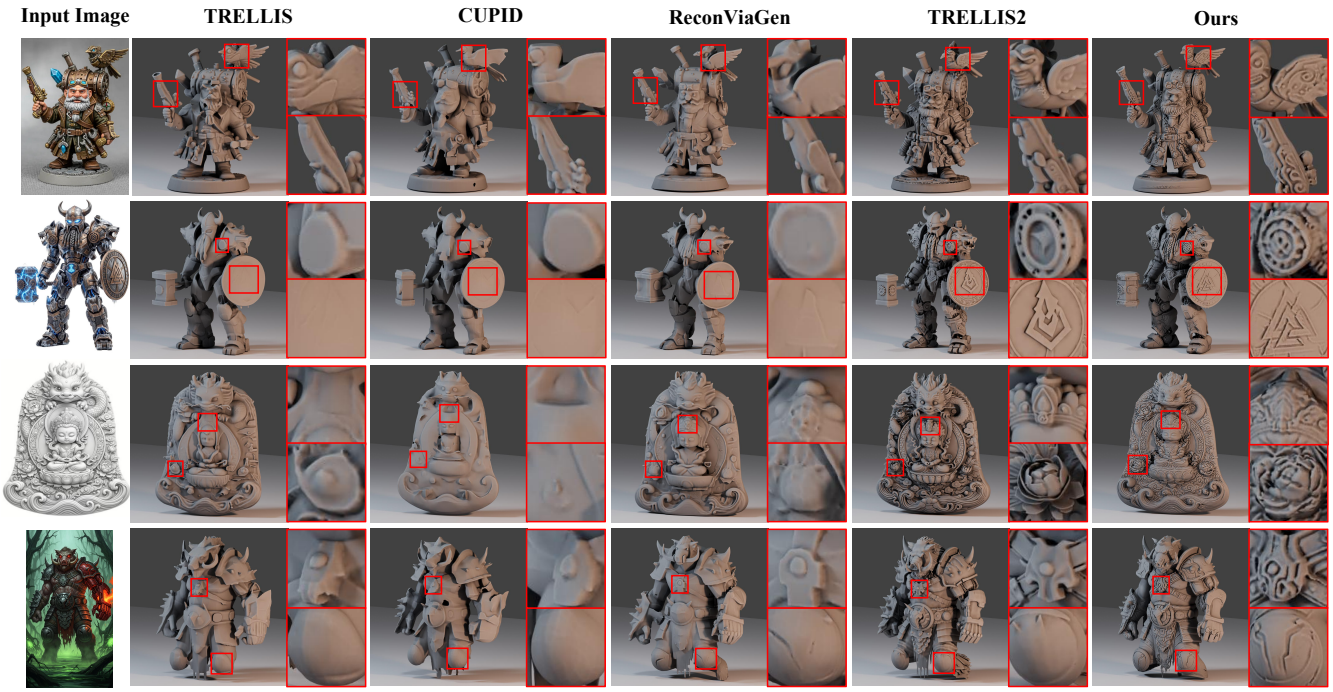


Fig. 8. Qualitative comparison for single-view generative reconstruction. The figure compares reconstructed meshes from the same challenging input images.

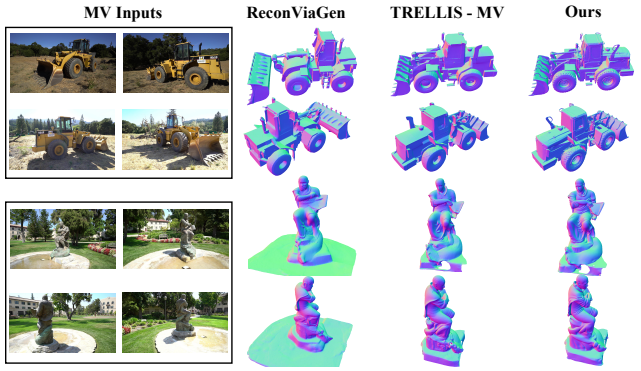


Fig. 9. Qualitative comparison for multi-view generative reconstruction on scenes from the Tanks and Temples dataset.

Table 4. Core component ablations on the held-out Objaverse-XL subset. Each checkmark indicates that the corresponding module is enabled. Lower is better for Normal-FID, and the best result is shown in bold.

Patch DiT	Voxel DiT	IMR	Normal Sup.	Normal-FID ↓
✓	-	-	-	92.96
✓	✓	-	-	75.80
✓	✓	✓	-	73.24
✓	✓	✓	✓	**68.97**

Table 5. Effect of IMR on boundary artifacts. Lower boundary-edge count indicates fewer open mesh boundaries.

Resolution	Method	Avg. boundary-edge count ↓
256^3	w/o IMR	2,462
	w/ IMR	**196**
1024^3	w/o IMR	25,353
	w/ IMR	**898**

Normal-FID computed from rendered normal maps, following the evaluation protocol in Section 5.1. Unless otherwise specified, we run ablation studies at 256^3 resolution for efficiency.

Effect of each core component. Table 4 and Figure 10 isolate the contribution of each component in the latent-free geometry pipeline. With only the patch-level DiT, the SDF values within each selected patch are encoded as channels of a patch token, so local intra-patch variation is not modeled explicitly at voxel granularity. Adding the voxel-level DiT introduces an intra-patch refinement stage that models the local SDF field directly over voxel tokens, improving surface detail and reducing Normal-FID from 92.96 to 75.80. However, the generated meshes still contain many holes, reflecting the missing manifold prior of a VAE decoder. IMR addresses this issue by regularizing local SDF transitions around the zero level set, greatly reducing holes and improving mesh validity. Table 5 further shows that IMR sharply reduces boundary edges at both 256^3 and 1024^3 resolutions. After this boundary reduction, standard post-processing hole filling can close the remaining small openings and produce watertight meshes. With normal-rendering supervision,

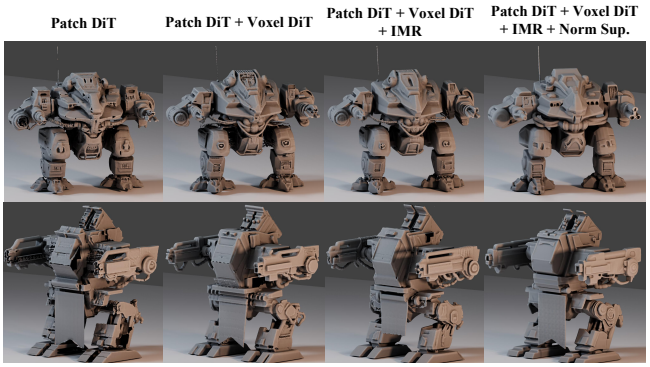


Fig. 10. Qualitative ablation of the core components. The patch-level DiT alone produces coarse and incomplete geometry. Adding the voxel-level DiT restores finer details but still leaves many holes; IMR greatly reduces these artifacts, and normal-rendering supervision further improves the quality.

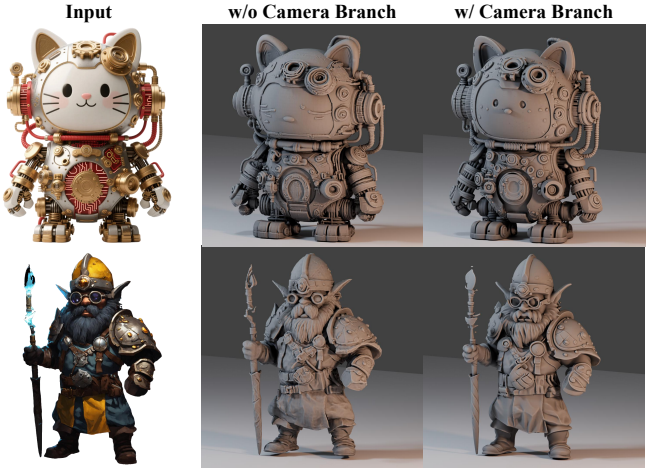


Fig. 11. Qualitative ablation of the camera prediction branch. Adding the branch improves alignment between the generated mesh and the conditioning image, especially for structures visible from the conditioning view.

the prediction is optimized entirely in the geometry domain after mesh extraction, further improving surface quality and achieving the best Normal-FID of 68.97.

Effect of camera prediction branch. Figure 11 evaluates the auxiliary camera prediction branch. Without this branch, the generated mesh can still be plausible in overall shape, but the geometry visible from the conditioning view may drift from the input image, leading to mismatched local structures. Adding camera prediction provides an explicit alignment signal during training, encouraging the DiT to associate image observations with the corresponding 3D viewpoint. As a result, the generated mesh aligns better with the input image while preserving the same overall geometry quality.

Effect of patch size. Table 6 reports the generation quality under different patch sizes. Smaller patches assign each token to a finer SDF region, which preserves more geometric detail and improves Normal-FID. However, reducing the patch size also increases the patch sequence length, leading to higher memory and attention cost. Larger patches are cheaper but blur local structures because each token must represent a broader surface neighborhood. We therefore choose $p = 8$ as the default setting, which provides a practical trade-off between geometric fidelity and computational efficiency.

Table 6. Patch-size ablation for the latent-free DiT. Smaller patches improve Normal-FID but increase the number of patch tokens; we use $p = 8$ as the default patch size.

Patch size	Normal-FID ↓
$p = 4$	**57.63**
$p = 8$ (default)	68.97
$p = 12$	76.21

Table 7. First-stage choices evaluated on the Objaverse-XL validation set at 512^3 resolution. Lower is better for Normal-FID.

First stage	Normal-FID ↓
Dense latent-free DiT	80.61
Hunyuan3D 2.1	80.43
Ours (VecSet DiT)	**76.62**

Effect of first-stage surface patch indices. Figure 12 compares three sources of surface patch indices at 512^3 resolution. These indices define the sparse support used by the latent-free DiT. In our default configuration, a pre-trained VecSet DiT generates a coarse mesh, from which we extract the indices. We also train a dense DiT with the same architecture as the latent-free DiT to directly predict a 64^3 dense voxel grid, where each voxel indicates whether the corresponding region belongs to the surface patch set. This direct predictor performs similarly to our default mesh-based source, showing that the second-stage latent-free DiT is not sensitive to a specific index generator as long as the predicted surface support is reasonable. The VecSet DiT remains preferable for its flexibility: since it outputs a mesh before patch extraction, surface patch indices can be produced at arbitrary resolutions. We further replace the first-stage generator with Hunyuan3D 2.1 [Hunyuan3D et al. 2025] and obtain similar results, which further supports the robustness of the latent-free DiT to different sources of surface patch indices. Table 7 reports Normal-FID on the Objaverse-XL validation set when the surface patch indices are extracted from different first-stage generators instead of the ground-truth meshes. Among the tested choices, using our VecSet DiT as the first-stage generator achieves the best result.

6 Conclusion and Limitations

We presented Direct3D-V3, a latent-free DiT framework that revisits the need for VAE-based compression in high-fidelity 3D generation. Instead of modeling compressed 3D latent representations, Direct3D-V3 performs diffusion in the native voxel domain by directly denoising raw SDF voxel grids. To make this formulation

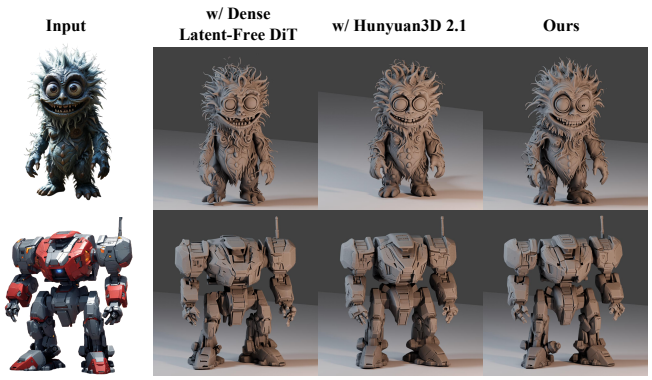


Fig. 12. Ablation of first-stage surface patch indices at 512^3 resolution. Comparable results across a pre-trained VecSet DiT, a dense 64^3 index predictor, and Hunyuan3D 2.1 indicate that the latent-free DiT is robust to different sources of surface patch indices.

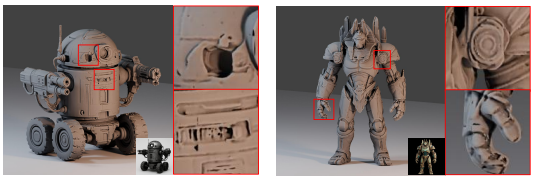


Fig. 13. Failure case of single-view generative reconstruction. When the first-stage surface indices are poorly aligned with the conditioning image, normal-guided SDF refinement operates on inaccurate sparse support and can distort the mesh.

scalable, the model uses sparse surface patch modeling and a dual-granularity DiT, where a patch-level transformer captures global shape structure and a lightweight voxel-level transformer restores local SDF variation. Beyond scalability, Direct3D-V3 makes the native SDF output geometrically reliable by constraining neighboring SDF values with implicit manifold regularization, supervising the extracted surface through rendered normals, and estimating the conditioning camera with an auxiliary branch for better view alignment. The same native SDF voxel output also enables generative reconstruction, where test-time normal guidance refines the observed surface while the learned prior completes unseen regions. Experiments show competitive or superior performance against recent image-to-3D methods, together with finer local geometry.

Limitations. The current framework has two main limitations. First, generative reconstruction depends on accurate first-stage surface indices, which define the sparse SDF support used by the latent-free DiT. If this support misses image-observed structures or places patches in inconsistent regions, test-time normal guidance may distort the mesh, especially around thin structures or ambiguous silhouettes. Figure 13 visualizes this failure mode, where inaccurate first-stage surface indices cause normal-guided refinement to over-update the native SDF and produce distorted geometry. Improving robustness to inaccurate sparse support remains an important direction, potentially through stronger camera estimation, guidance with uncertainty estimates, or more constrained geometry refinement. Second, Direct3D-V3 currently focuses on geometry only. In principle, the latent-free formulation can be extended beyond SDF values by adding channels for texture, material, or PBR attributes to the voxel grid. However, jointly modeling geometry and appearance in this native space will require additional design to keep training stable and to preserve consistency between surface structure and material signals. We leave this unified latent-free geometry and appearance generation setting for future work.

Acknowledgments

This work was supported by National Key R&D Program of China (2023YFB3209702), National Natural Science Foundation of China (62472213), Gusu Innovation & Entrepreneurship Leading Talents Program (ZXL2024361), and computing resources from DreamTech.

References

Michael S Albergo and Eric Vanden-Eijnden. 2022. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571* (2022).

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. 2023. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 22669–22679.

Jiahao Chang, Chongjie Ye, Yushuang Wu, Yuantao Chen, Yidan Zhang, Zhongjin Luo, Chenghong Li, Yihao Zhi, and Xiaoguang Han. 2025. ReconViaGen: Towards Accurate Multi-view 3D Object Reconstruction via Generation. *arXiv preprint arXiv:2510.23306* (2025).

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. 2025a. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16251–16261.

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. 2025b. Dip: Taming diffusion models in pixel space. *arXiv preprint arXiv:2511.18822* (2025).

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

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. 2022. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*. Ieee, 2553–2560.

Jingyu Guo, Sensen Gao, Jia-Wang Bian, Wanhu Sun, Heliang Zheng, Rongfei Jia, and Mingming Gong. 2025. Hyper3d: Efficient 3d representation via hybrid triplane and octree feature for enhanced 3d shape variational auto-encoders. *arXiv preprint arXiv:2503.10403* (2025).

Jing He, Haodong Li, Mingzhi Sheng, and Ying-Cong Chen. 2025a. Lotus-2: Advancing Geometric Dense Prediction with Powerful Image Generative Model. *arXiv preprint arXiv:2512.01030* (2025).

Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. 2025b. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14822–14833.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

Binbin Huang, Haobin Duan, Yiqun Zhao, Zibo Zhao, Yi Ma, and Shenghua Gao. 2025. CUPID: Generative 3D Reconstruction via Joint Object and Pose Modeling. *arXiv preprint arXiv:2510.20776* (2025).

Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. 2025. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442* (2025).

Tanghui Jia, Dongyu Yan, Dehao Hao, Yang Li, Kaiyi Zhang, Xianyi He, Lanjiong Li, Yuhan Wang, Jinnan Chen, Lutao Jiang, et al. 2025. UltraShape 1.0: High-Fidelity 3D Shape Generation via Scalable Geometric Refinement. *arXiv preprint arXiv:2512.21185* (2025).

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems* 35 (2022), 26565–26577.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 42, 4 (2023), 139–1.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1–13.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. 2025. LATTICE: Democratize High-Fidelity 3D Generation at Scale. *arXiv preprint arXiv:2512.03052* (2025).

Dong-Yang Li, Wang Zhao, Yuxin Chen, Wenbo Hu, Meng-Hao Guo, Fang-Lue Zhang, Ying Shan, and Shi-Min Hu. 2026b. Pixal3D: Pixel-Aligned 3D Generation from Images. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–12.

Tianhong Li and Kaiming He. 2025. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720* (2025).

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

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. 2026a. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *Advances in Neural Information Processing Systems* 38 (2026), 118582–118600.

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

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

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2024. Syncdreamer: Generating multiview-consistent images from a single-view image. In *International conference on learning representations*, Vol. 2024. 27676–27697.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. 2024. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9970–9980.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. 2025. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. *arXiv preprint arXiv:2511.19365* (2025).

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*. PMLR, 8162–8171.

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

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4195–4205.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. 2025. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6121–6132.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. 2023. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (ToG)* 42, 4 (2023), 1–16.

Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. 2023. Diffusion-based signed distance fields for 3d shape generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 20887–20897.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. 2025. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741* (2025).

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. 2024. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *Advances in Neural Information Processing Systems* 37 (2024), 121859–121881.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, Yao Yao, et al. 2026. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *Advances in Neural Information Processing Systems* 38 (2026), 170778–170804.

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

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

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

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, et al. 2024. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 27091–27101.

Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. 2025. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 25050–25061.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. 2025. Pixeldit: Pixel diffusion transformers for image generation. *arXiv preprint arXiv:2511.20645* (2025).

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 2023. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)* 42, 4 (2023), 1–16.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

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

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. 2023. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems* 36 (2023), 73969–73982.

Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. 2025. Stable virtual camera: Generative view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12405–12414.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. 2024. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations*, Vol. 2024. 46766–46782.

A Computational Cost and Scaling

Table 8. Computational cost and scaling across resolutions and patch sizes. Memory and per-step inference time are measured on one A800 GPU.

Res.	Patch	Avg. Patch Tokens	Memory (MB)	Time/step (s)
256^3	4	7,980	4,386	0.31
	8	1,629	3,716	0.12
	12	721	3,798	0.12
512^3	4	35,576	8,082	1.55
	8	7,340	5,716	0.37
	12	2,986	5,080	0.25
1024^3	4	153,606	21,490	12.72
	8	32,536	13,312	1.82
	12	13,422	10,620	1.09
1536^3	4	353,637	45,128	53.76
	8	76,435	24,174	4.64
	12	31,473	19,450	2.59



Fig. 14. Additional visual results generated by Direct3D-V3.

Table 8 reports the average number of active patch tokens, average GPU memory usage, and per-step inference time across resolutions and patch sizes. The cost increases with the number of active patch tokens, making smaller patches more expensive at high resolution. We use patch size 8 as a practical balance between geometric fidelity and efficiency, which supports native SDF generation up to 1536^3.

B Additional Visual Results

Figure 14 shows more visual results generated by Direct3D-V3.

C Additional Qualitative Comparisons

Figure 15 provides additional qualitative comparisons for image-to-3D generation. These examples complement the main results in Figure 7 and include more diverse input images. Across these cases, Direct3D-V3 produces coherent complete shapes while preserving sharper local geometry and more regular surface details than competing methods.

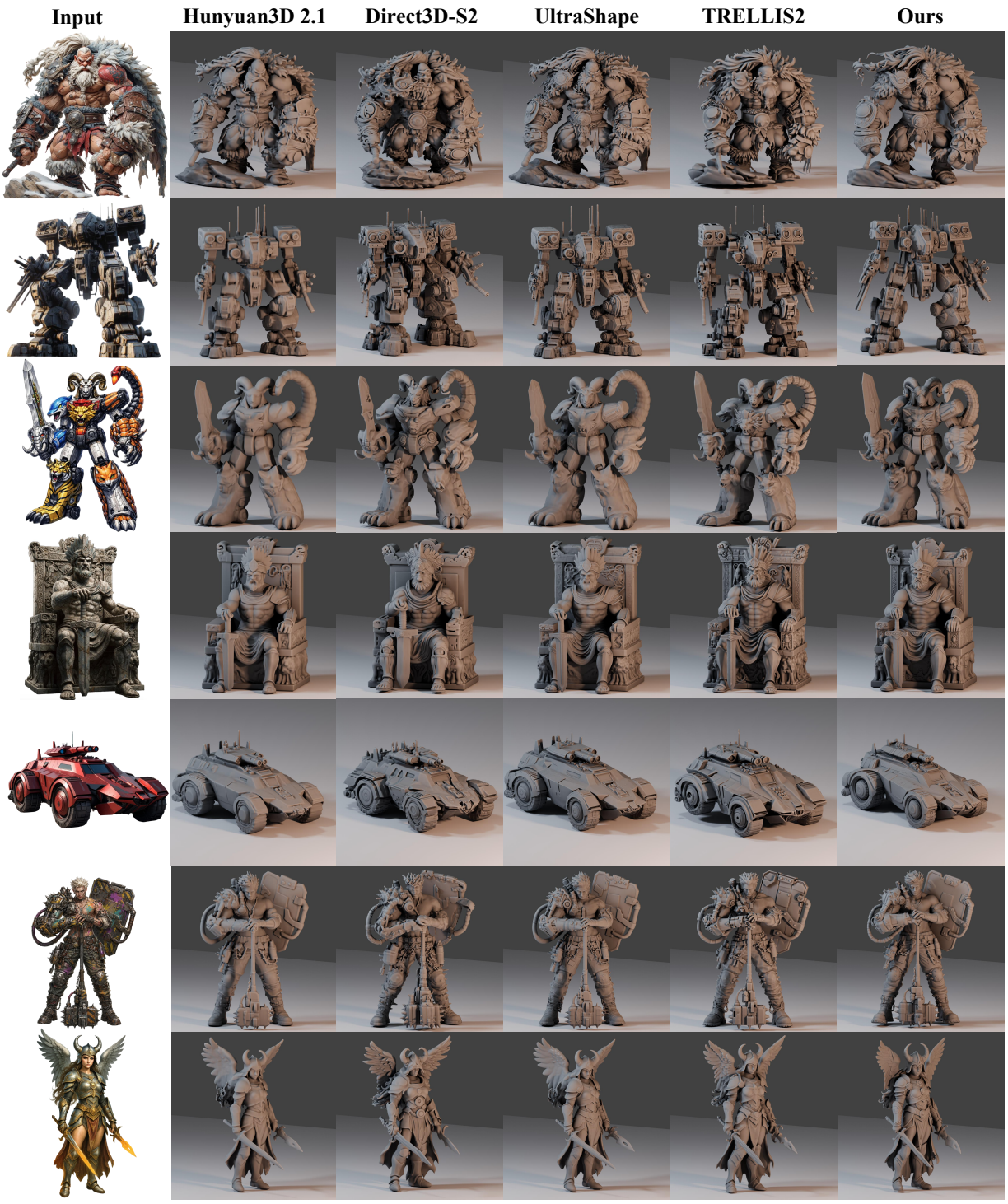


Fig. 15. Additional qualitative comparisons with recent image-to-3D generation methods.