Hunyuan3D 2.0: Scaling Diffusion Models for High Resolution Textured 3D Assets Generation

" *Living out everyone's imagination on creating and manipulating 3D assets.*"

— **Hunyuan3D Team** *

Abstract

We present Hunyuan3D 2.0, an advanced large-scale 3D synthesis system for generating high-resolution textured 3D assets. This system includes two foundation components: a large-scale shape generation model – Hunyuan3D-DiT, and a large-scale texture synthesis model – Hunyuan3D-Paint. The shape generative model, built on a scalable flow-based diffusion transformer, aims to create geometry that properly aligns with a given condition image, laying a solid foundation for downstream applications. The texture synthesis model, benefiting from strong geometric and diffusion priors, produces high-resolution and vibrant texture maps for either generated or hand-crafted meshes. Furthermore, we build Hunyuan3D-Studio – a versatile, user-friendly production platform that simplifies the re-creation process of 3D assets. It allows both professional and amateur users to manipulate or even animate their meshes efficiently. We systematically evaluate our models, showing that Hunyuan3D 2.0 outperforms previous state-of-the-art models, including the open-source models and closed-source models in geometry details, condition alignment, texture quality, and *etc*. Hunyuan3D 2.0 is publicly released in order to fill the gaps in the open-source 3D community for large-scale foundation generative models. The code and pre-trained weights of our models are available at: https://github.com/Tencent/Hunyuan3D-2.

---

* Hunyuan3D team contributors are listed in the end of report.
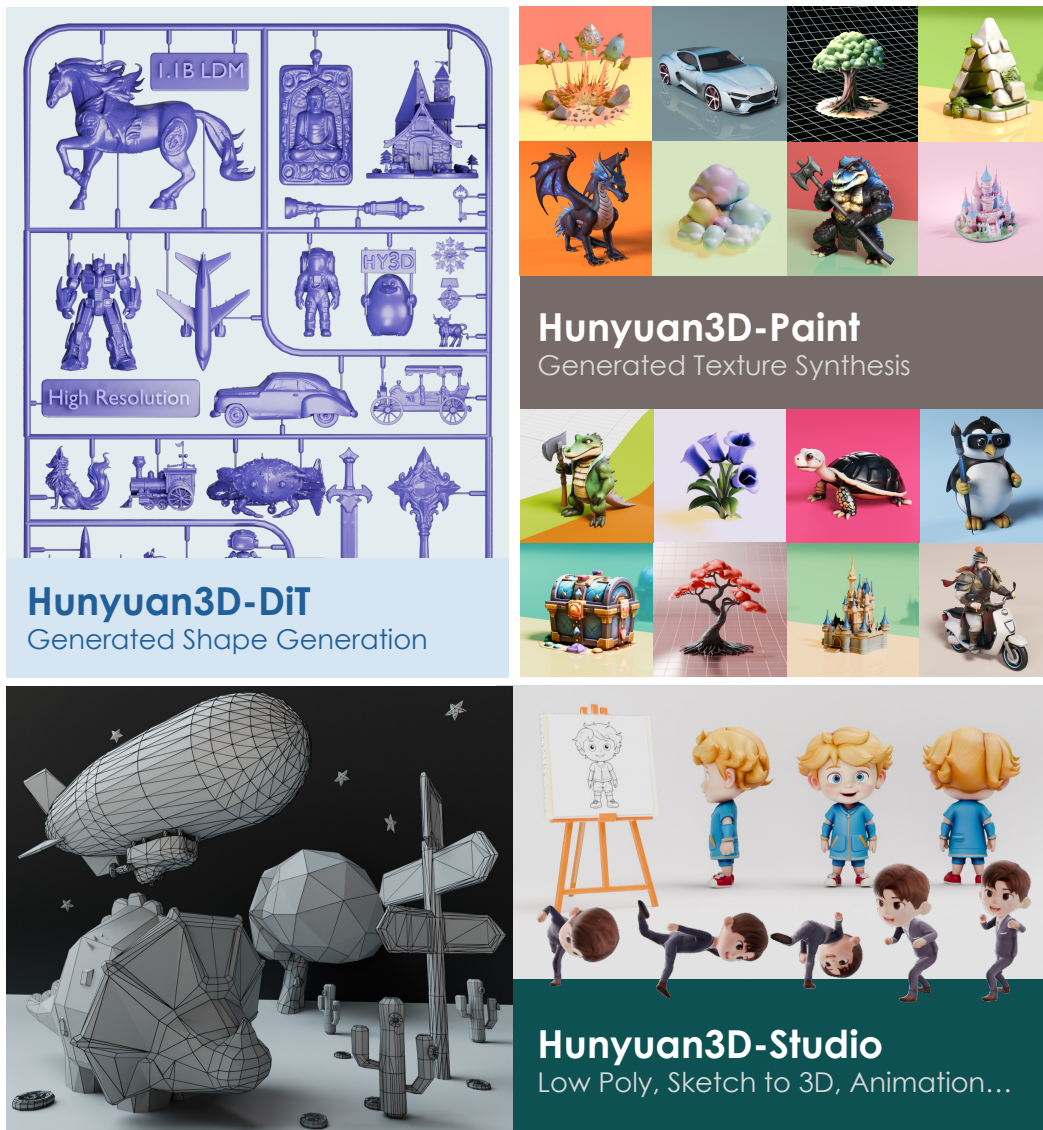
Figure 1: An overall of Hunyuan3D 2.0 system.

1 Introduction

Digital 3D assets have woven themselves into the very fabric of modern life and production. In the realms of gaming and film, these assets are vibrant expressions of creators' imaginations, spreading joy and crafting immersive experiences for players and audiences alike. In the fields of physical simulation and embodied AI, 3D assets serve as essential building blocks, enabling machines and robots to mimic and comprehend the real world. Yet, the journey of creating 3D assets is anything but straightforward; it is often a complex, time-consuming, and costly endeavor. A typical production pipeline may involve stages like sketch design, digital modeling, and 3D texture mapping, each demanding high expertise and proficiency in digital content creation software. As a result, the automated generation of high-resolution digital 3D assets has emerged as one of the most exciting and sought-after topics in recent years.

Despite the importance of automated 3D generation and rapid development in image and video generation fueled by the rise of diffusion models [33, 74, 24, 50, 43], the field of 3D generation appears to be relatively stagnant in the era of large models and big data, with only a handful of works making gradual progress [111, 118, 49]. Building on the 3DShape2Vectset [111], Michelangelo [118] and CLAY [113] gradually enhance shape generation performance, where CLAY is the first work to demonstrate the unprecedented potential of diffusion models in 3D asset generation. Nevertheless, progress in the 3D domain remains limited. As evidenced in other fields [114, 4, 3], the prosperity of a domain in the era of large models usually relies on a strong open-source foundational model, such as Stable Diffusion [74, 69, 24] for image generation, LLaMA [90, 91, 22] for language models, and HunyuanVideo [43] for video generation. To this end, we present *Hunyuan3D 2.0*, a 3D asset creation system with two strong open-sourced 3D foundation models: *Hunyuan3D-DiT* for generative shape creation and *Hunyuan3D-Paint* for generative texture synthesis.

Hunyuan3D 2.0 features a two-stage generation pipeline, starting with the creation of a bare mesh, followed by the synthesis of a texture map for that mesh. This strategy is effective for decoupling the difficulties of shape and texture generation [34, 106, 46, 47] and also provides flexibility for texturing either generated or handcrafted meshes. With this architecture, our shape creation model – Hunyuan3D-DiT, is designed as a large-scale flow-based diffusion model. As a prerequisite, we first train an autoencoder – Hunyuan3D-ShapeVAE using advanced techniques such as mesh surface importance sampling and variational token length to capture fine-grained details on the meshes. Then, we build up a dual-single stream transformer [45] on the latent space of our VAE with the flow-matching [53, 24] objective. Our texture generation model – Hunyuan3D-Paint is made of a novel mesh-conditioned multi-view generation pipeline and a number of sophisticated techniques for preprocessing and baking multi-view images into high-resolution texture maps.

We performed an in-depth comparison of Hunyuan3D 2.0 in relation to leading 3D generation models worldwide, including three commercial closed-source end-to-end products, an end-to-end open-sourced model Trellis [100], and several separate models [9, 37, 98, 110, 55, 59] for shape and texture generation. We report visual and quantitative evaluation results across three dimensions: generated textured mesh, bare mesh, and texture map. We also provided user study results on 300 test cases involving 50 participants. The comparison shows the superiority of Hunyuan3D 2.0 in alignment between conditional images and generated meshes, generation of fine-grained details, and human preference ratings.

2 Hunyuan3D 2.0 Architecture

In this section, we elaborate on the model architecture of Hunyuan3D 2.0, focusing on two main components: the shape generation model and the texture generation model. Fig. 2 illustrates the pipeline of Hunyuan3D 2.0 for creating a high-resolution textured 3D asset. Given an input image, Hunyuan3D-DiT initially generates a high-fidelity bare mesh via the shape generation model. This model comprises a Hunyuan3D-ShapeVAE and a Hunyuan3D-DiT, which will be discussed in Sec. 3. Subsequently, by leveraging strong geometric priors and the input image, we introduce Hunyuan3D-Paint as our texture generation model in Sec. 4. This model produces self-consistent multi-view outputs, which are used for baking high-definition texture maps.
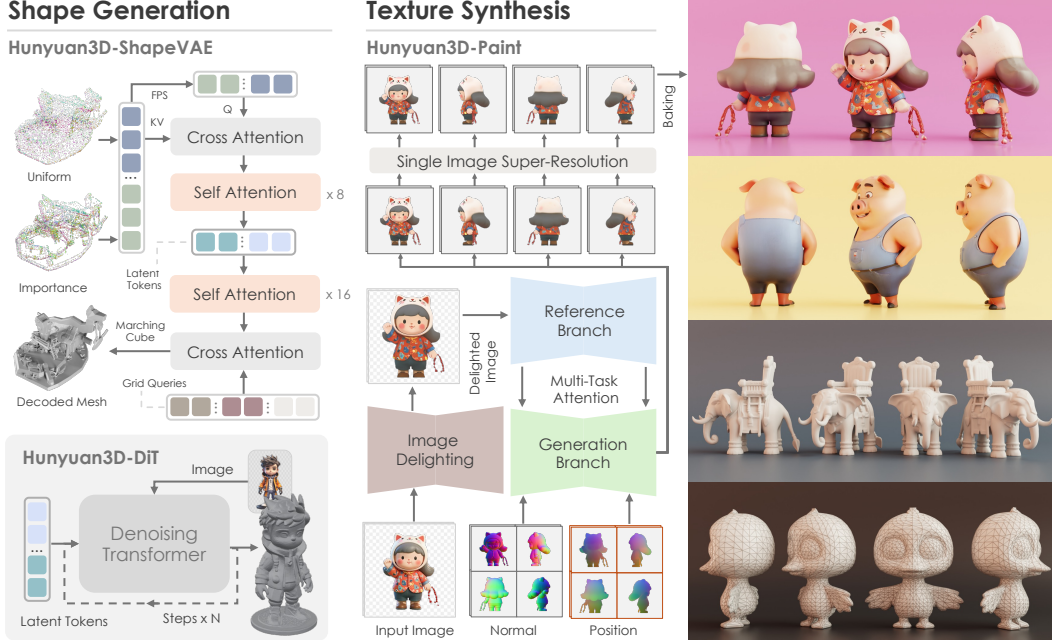
Figure 2: An overall of Hunyuan3D 2.0 architecture for 3D generation. It consists of two main components: Hunyuan3D-DiT for generating bare mesh from a given input image and Hunyuan3D-Paint for generating a textured map for the generated bare mesh. Hunyuan3D-Paint takes geometry conditions – normal maps and position maps of generated mesh as inputs and generates multi-view images for texture baking.

3 Generative 3D Shape Generation

Hunyuan3D 2.0 employs the architecture of the latent diffusion model [74, 111, 118, 113] for shape generation. This design is driven by the successful applications of latent diffusion models in image and video generation. Specifically, our shape generation model consists of (1) an autoencoder – Hunyuan3D-ShapeVAE (Sec. 3.1) that compresses the shape of a 3D asset represented by polygon mesh into a sequence of continuous tokens in the latent space; (2) a flow-based diffusion model – Hunyuan3D-DiT (Sec. 3.2), trained on the latent space of ShapeVAE for predicting object token sequences from a user-provided image. The predicted tokens are further decoded into a polygon mesh with VAE decoder. The details of these models are illustrated below.

3.1 Hunyuan3D-ShapeVAE

Hunyuan3D-ShapeVAE employs vector sets, a compact neural representation for 3D shapes proposed by 3DShape2VecSet [111], which has also been leveraged in the recent work Dora [11]. Followed by Michelangelo [118], we use a variational encoder-decoder transformer for shape compression and decoding. Besides, we choose 3D coordinates and the normal vector of point cloud sampled from the surface of 3D shapes as inputs for the encoder and instruct the decoder to predict the Signed Distance Function (SDF) of the 3D shape, which can be further decoded into triangle mesh via the marching cube algorithm. The overall network architecture is illustrated in Fig. 3.

Importance Sampled Point-Query Encoder. The encoder $\mathcal{E}_s$ aims to extract representative features to characterize 3D shapes. To achieve this, our first design utilizes an attention-based encoder to encode point clouds uniformly sampled from the surface of a 3D shape. However, this design usually fails to reconstruct the details of complex objects. We attribute this difficulty to the variations in the complexity of regions on the shape surface. Therefore, in addition to uniformly sampled point clouds, we designed an importance sampling method* that samples more points on the edges and corners of the mesh, which provides more complete information for describing complex regions.

---

*Concurrent work [11] also proposes similar importance sampling to improve the VAE reconstruction performance based on similar observations.

In detail, for an input mesh, we first collect uniformly sampled surface point clouds $P_u \in \mathbb{R}^{M \times 3}$, and importance sampled surface point clouds $P_i \in \mathbb{R}^{N \times 3}$. We use a layer of cross attention, to compress the input point clouds into a set of continuous tokens via a set of point queries [111]. To obtain point queries, we apply Farthest Point Sampling (FPS) separately to P_u and P_i to obtain the uniform point query $Q_u \in \mathbb{R}^{M' \times 3}$ and the importance point query $Q_i \in \mathbb{R}^{N' \times 3}$. The final point cloud $P \in \mathbb{R}^{(M+N) \times 3}$ and point query $Q \in \mathbb{R}^{(M'+N') \times 3}$ for the cross attention are constructed by concatenating both sources. Then, we encode the point clouds P and point queries Q with Fourier positional encoding followed by a linear projection, resulting $X_p \in \mathbb{R}^{(M+N) \times d}$ and $X_q \in \mathbb{R}^{(M'+N') \times d}$, where d is the width of the transformer. The encoded point cloud and point query are sent to the cross attention followed by a number of self-attention layers, which helps improve the feature representation, to obtain the hidden shape representation $H_s \in \mathbb{R}^{(M'+N') \times d}$. Since we adopt the design of variational autoencoder [41], an additional linear projection is applied on H_s to predict the mean $\mathrm{E}(Z_s) \in \mathbb{R}^{(M'+N') \times d_0}$ and variance $\mathrm{Var}(Z_s) \in \mathbb{R}^{(M'+N') \times d_0}$ of the final latent shape embedding in a token sequence, where d_0 is the dimension of latent shape embedding.

Decoder. The decoder $\mathcal{D}_s$ reconstructs the 3D neural field from the latent shape embedding Z_s from the encoder. $\mathcal{D}_s$ starts from a projection layer to transform the latent embedding from dimension d_0 back to the width of transformer d. Then, a number of self-attention layers further process the hidden embeddings, after which is another point perceiver that takes 3D grid $Q_g \in \mathbb{R}^{(H \times W \times D) \times 3}$ as queries to obtain 3D neural field $F_g \in \mathbb{R}^{(F_n \times W \times D) \times d}$ from the hidden embeddings. We use another linear projection on the neural field to obtain the Sign Distance Function (SDF) $F_{sdf} \in \mathbb{R}^{(F_o \times W \times D) \times 1}$, which can be decoded into triangle mesh with marching cube algorithms.

Training Strategy & Implementation. We employ multiple losses to supervise the model training, including (1) the reconstruction loss that computes MSE loss between predicted SDF $\mathcal{D}_s(x|Z_s)$ and ground truth $\mathrm{SDF}(x)$, and (2) the KL-divergence loss $\mathcal{L}_{KL}$ to make the latent space compact and continuous, which facilitates the training of diffusion models. Due to the dense computation required by complete SDF, the reconstruction loss is calculated as the expectation of losses on randomly sampled points in the space and shape surface. The overall training loss $\mathcal{L}_r$ can be written as,

$$\mathcal{L}_r = \mathbb{E}_{x \in \mathbb{R}^3}[\mathrm{MSE}(\mathcal{D}_s(x|Z_s), \mathrm{SDF}(x))] + \gamma \mathcal{L}_{KL} \tag{1}$$

where γ is the loss weight of KL loss. During training, we also utilize a multi-resolution strategy to
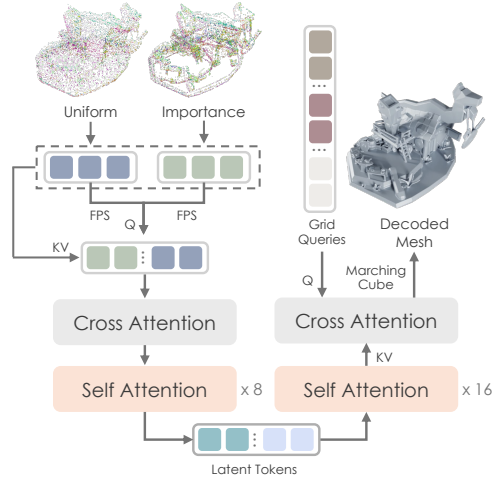

Figure 3: The overall architecture of Hunyuan3D-ShapeVAE. Instead of only using uniform sampling on mesh surface, We have developed an importance sampling strategy to extract high-frequency detail information from the input mesh surface, such as edges and corners. This allows the model to better capture and represent the intricate details of 3D shapes. Note that during the point query construction, the Farthest Point Sampling (FPS) operation is performed separately for the uniform point cloud and the importance sampling point cloud.

speed up model convergence, where the length of the latent token sequence is randomly sampled from a predefined set. A shorter sequence reduces the computation cost, and a longer sequence facilitates the reconstruction quality. The longest sequence length is 3072 in our released version, which could support high-resolution shape generation with fine-grained and sharp details.

3.2 Hunyuan3D-DiT

Hunyuan3D-DiT is a flow-based diffusion model aimed at producing high-fidelity and high-resolution 3D shapes according to given image prompts.

Network Structure. Inspired by FLUX [45], we adopt a dual- and single-stream network structure as in Fig. 4. In dual-stream blocks, latent tokens and condition tokens are processed with separate QKV projections, MLP, and *etc.*, but interact within an attention operation. In single-stream blocks,
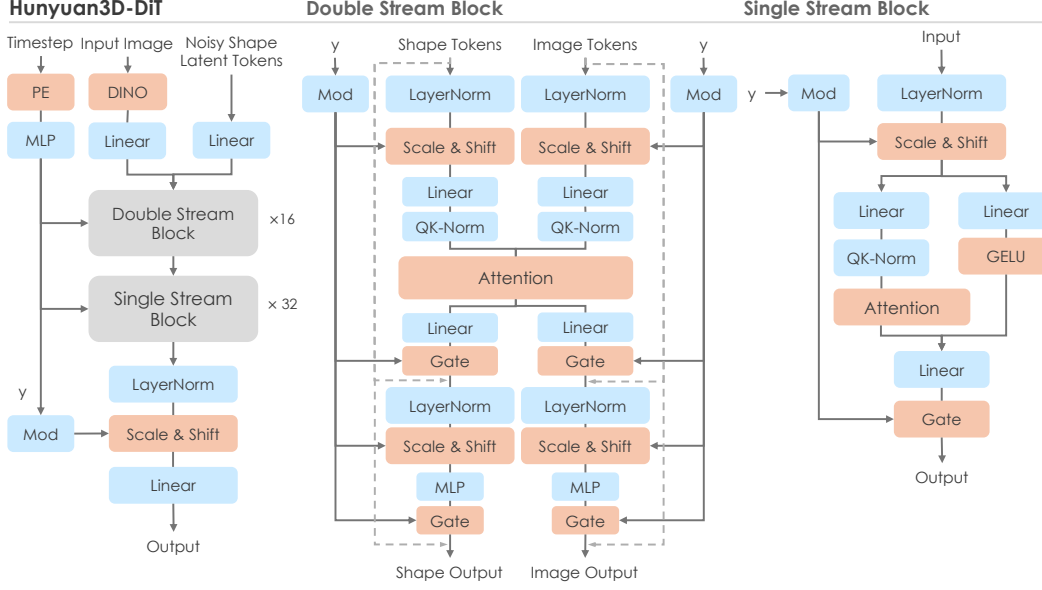
Figure 4: Overview of Hunyuan3D-DiT. It adopts a transformer architecture with both double- and single-stream blocks. This design benefits the interaction between modalities of shape and image, helping our model to generate bare meshes with exceptional quality. (Note that the orange blocks have no learnable parameters, the blue blocks contain trainable parameters, and the gray blocks indicate a module composed of more details.)

latent tokens and condition tokens are concatenated and processed by spatial attention and channel attention in parallel. We only use the embedding of the timestep for the modulation modules. Besides, we omit the positional embedding of the latent sequence as the specific latent token of our ShapeVAE in the sequence does not correspond to a fixed location in the 3D grid. Instead, the content of our 3D latent tokens themselves is responsible for figuring out the position/occupancy of the generated shape in the 3D grid, which is different from image/video generation where their tokens are responsible for predicting content in a specific location in 2D/spatial-temporal grid.

Condition Injection. We employ a pre-trained image encoder to extract conditional image tokens of the patch sequence including the head token at the last layer. To capture the fine-grained details in the image, we utilize a large image encoder – DINOv2 Gaint [64] and large input image size – 518×518. Besides, we also remove the background of the input image, resize the object to a unified size, reposition the object to the center, and fill the background with white, which helps to remove the negative impact of the background and increase the effective resolution of the input image.

Training & Inference. We utilize flow matching objective [53, 24] for training our model. Specifically, flow matching first defines a probability density path between Gaussian distribution and data distribution, then, the model is trained to predict the velocity field $u_t = \frac{x_t}{d_t}$ that drifting sample x_t towards data x_1. In our case, we adopt the affine path with the conditional optimal transport schedule specified in [54], where $x_t = (1 - t) \times x_0 + t \times x_1$, $u_t = x_1 - x_0$. Therefore, the training loss is formulated as,

$$\mathcal{L} = \mathbb{E}_{t,x_0,x_1}[\| u_\theta(x_t, c, t) - u_t \|_2^2], \tag{2}$$

where $t \sim \mathbb{U}(0, 1)$ and c denotes model condition. During the inference phase, we first randomly sample a start point $x_0 \sim \mathbb{N}(0, 1)$ and employ a first-order Euler ordinary differential equation (ODE) solver to solve x_1 with our diffusion model $u_\theta(x_t, c, t)$.

4 Generative Texture Map Synthesis

Given a 3D mesh without texture and an image prompt, we aim to generate a high-resolution and seamless texture map. The texture map should closely conform to the image prompt in the visible region, exhibit multi-view consistency, and maintain harmonious with the input mesh.
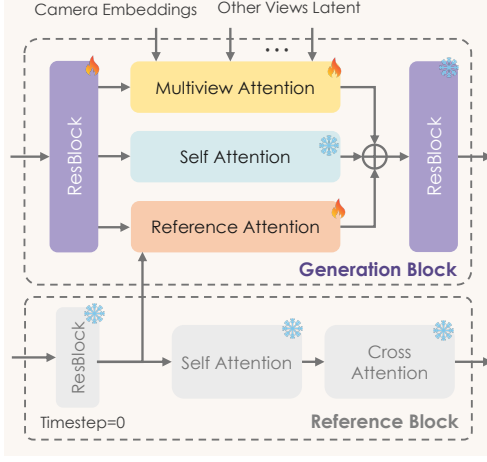
Hunyuan3D-Paint
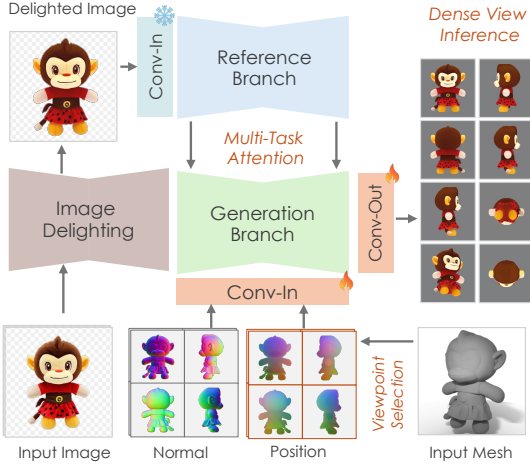
Figure 5: Overview of Hunyuan3D-Paint. We leverage an image delighting module to convert the input image to an unlit state to produce light-invariant texture maps. The system features a double-stream image conditioning reference-net, which provides faithfully conditional image features to the model. Furthermore, it facilitates the production of texture maps that conform closely to the input image. The multi-task attention module ensures that the model synthesizes multi-view consistent images. This module maintains the coherence of all generated images while adhering to the input.

To achieve these objectives, we employ a three-stage framework, including a pre-processing stage (Sec. 4.1), a multi-view image synthesis stage (Sec. 4.2, Hunyuan3D-Paint), and a texture baking stage based on dense multi-view inference (Sec. 4.3). Other details related to model training and text- and image-to-texture pipeline are provided in Sec. 4.4. Fig. 5 illustrates the complete pipeline of our texture map synthesis method.

4.1 Pre-processing

Image Delighting Module. The reference image typically exhibits pronounced and varied illumination and shadow, whether collected by the user or generated by T2I models. Directly inputting such images into the multi-view generation framework can cause illumination and shadows to be baked into the texture maps. To address this issue, we leverage a delighting procedure on the input image via an image-to-image approach [6] before multi-view generation. Specifically, to train such an image delighting model, we collect a large-scale 3D dataset and render it under the illumination of a random HDRI environmental map and an even white light to form the corresponding pair-wise image data. Benefiting from this image delighting model, our multi-view generation model can be fully trained on white-light illuminated images, enabling an illumination-invariant texture synthesis.

View Selection Strategy. In practical applications, to reduce the costs of texture generation (i.e., generate the largest area of texture with the minimum number of viewpoints), we employ a geometry-aware viewpoint selection strategy to support effective texture synthesis. By considering the coverage of the geometric surface, we heuristically select 8 to 12 viewpoints for inference. Initially, we fix 4 orthogonal viewpoints as basis since they cover most parts of the geometry. Subsequently, we iteratively add novel viewpoints using a greedy search approach. The specific process is illustrated in Algorithm 1, and the coverage function in the algorithm is defined as:

$$\mathcal{F}(v_i, \mathbb{V}_s, \mathbf{M}) = \mathcal{A}_{area} \left\{ \mathcal{UV}_{\text{cover}}(v_i, \mathbf{M}) \setminus \left[\mathcal{UV}_{\text{cover}}(v_i, \mathbf{M}) \cap \left(\bigcup_{s \in \mathbb{V}_s} \mathcal{UV}_{\text{cover}}(v_s, \mathbf{M}) \right) \right] \right\} \quad (3)$$

where $\mathcal{UV}_{\text{cover}}(v, \mathbf{M})$ is a function that returns the set of covering texels in UV space based on the input view v and mesh geometry $\mathbf{M}$, and $\mathcal{A}_{area}(\cdots)$ is a function that calculates the coverage area according to the given set of covering texels. This approach encourages the multi-view generation

model to focus on viewpoints with more unseen regions, together with the dense-view inference, alleviating the burden of post-processing (i.e., texture inpainting).

Algorithm 1 View Selection Algorithm

1: **Input:**
2:     Initialize a selected viewpoint set $\mathbb{V}_s$, a reference viewpoint set $\mathbb{V}_r$, a coverage function $\mathcal{F}$, and a mesh $\mathbf{M}$.
3:     Set $N_{max} = 12$ as the maximum number of iterative searches, and $N_{fixed} = 4$ as the initial fixed number of viewpoints.
4: **for** $i = N_{fixed}$ **to** $N_{max} - 1$ **do**
5:     Set $c_{max} = -1$
6:     Set $v_{max} = -1$
7:     **for** $v_i \in \mathbb{V}_r$ **do**
8:         Set $c_{cur} = \mathcal{F}(v_i, \mathbb{V}_s, \mathbf{M})$
9:         **if** $c_{cur} > c_{max}$ **then**
10:             $c_{max} = c_{cur}$
11:             $v_{max} = v_i$
12:         **end if**
13:     **end for**
14:     $\mathbb{V}_r \leftarrow \mathbb{V}_r \setminus \{v_{max}\}$
15:     $\mathbb{V}_s \leftarrow \mathbb{V}_s \cup \{v_{max}\}$
16: **end for**

4.2 Hunyuan3D-Paint

Geometry-conditioned multi-view image generation is a key component in the texture synthesis framework. In the context of image-guided texture synthesis, the design of the multi-view image generation must be meticulously crafted to achieve image alignment, geometric following, and multi-view consistency. These functionalities are realized in Hunyuan3D-Paint according to a bunch of techniques containing a double-stream image conditioning reference-net, a multi-task attention mechanism, and a strategy for geometry and view conditioning.

Double-stream Image Conditioning Reference-Net. In the aim of the image following, Hunyuan3D-Paint implements a reference-net conditioning approach [115], upon which we develop a series of alternative solutions. Specifically, rather than a noisy feature synchronized with the generation branch, we directly feed the original VAE feature of the reference image into the reference branch to maintain image details as much as possible. Since the feature is noiseless, we set the timestep of the reference branch to 0 to maintain the input image information faithfully. On the other hand, to regularize potential style bias introduced by the 3D rendering dataset, we abandon the shared-weight reference-net as in [88, 78] and freeze the weights of the original SD2.1 weights (which serves as the base model for our multi-view generation). We have found that the fixed-weights reference-net serves as a soft regularization that anchors the generated image distribution, preventing it from drifting away towards the rendered image distribution, significantly improving the performance on real-world image conditioning. Together, these two schemes form the double-stream image conditioning strategy. We leverage the zero-noised double-stream image conditioning reference-net by capturing a feature cache prior to each self-attention module. This cache is then fed into the multi-view diffusion model via a reference attention module.

Multi-task Attention Mechanism. We introduce two additional attention modules alongside the original self-attention to enable the image diffusion model to generate multi-view images guided by a reference image. The reference attention module integrates the reference image into the multi-view diffusion process. In contrast, the multi-view attention module ensures consistency across the generated views. To mitigate potential conflicts arising from these multi-functionalities, we design the additional two distinct attention modules in a parallel structure (illustrated in Fig. 5), which can be expressed as:

$$Z_{MVA} = Z_{SA} + \lambda_{ref} \cdot \text{Softmax}\left(\frac{Q_{ref}K_{ref}^T}{\sqrt{d}}\right) V_{ref} + \lambda_{mv} \cdot \text{Softmax}\left(\frac{Q_{mv}K_{mv}^T}{\sqrt{d}}\right) V_{mv} \quad (4)$$

where Z_{SA} represents the feature calculated by the original frozen-weight self-attention, and $Q_{ref}, K_{ref}, V_{ref}$ and Q_{mv}, K_{mv}, V_{mv} are the Query, Key, and Value projected features of reference attention and multi-view attention, respectively.

Geometry and View Conditioning. Following geometry is another unique feature in texture map synthesis. To enable effective training, we opt for an easy implementation of directly concatenating the geometry conditions with noise. Specifically, we first input the multi-view canonical normal maps and canonical coordinate maps (CCM)—two view-invariant geometry conditions we utilize—into a pre-trained Variational Autoencoder (VAE) to obtain geometric features. These features are then concatenated with latent noise and fed into the channel-extended input convolution layer of the diffusion model.

Other than geometry conditioning, we adopt a learnable camera embedding in our pipeline to boost the viewpoint clue for the multi-view diffusion model. Specifically, we assign a unique unsigned integer to each pre-defined viewpoint and set up a learnable view embedding layer to map the integer to a feature vector, which is then injected into the multi-view Diffusion model. We have found in our experiments that combining the geometry conditioning with a learnable camera embedding yields the best performance.

4.3 Texture Baking

Dense-view inference. Potential self-occlusion is a significant challenge in the context of texture synthesis within a multi-view image generation framework, particularly when handling irregular geometries produced by shape generative models. This issue necessitates modeling texture synthesis as a two-stage framework. The first stage focuses on multi-view image generation, while the second stage involves non-trivial inpainting to fill the holes caused by self-occlusion. In our texture-map synthesis framework, we alleviate the burden on the second stage of inpainting by facilitating dense-view inference during the multi-view generation stage. To enable effective dense-view inference, a view dropout strategy is introduced as a flexible training mechanism that allows the model to encounter all of the pre-set viewpoints, thereby enhancing its 3D perception capabilities and generalization. Specifically, we randomly select 6 viewpoints from a total of 44 pre-set viewpoints to serve as a batch input to the multi-view diffusion backbone network. During the inference phase, our framework is able to output the images of any specified viewpoints, supporting dense-view inference.

Single Image Super-resolution. To enhance texture quality, we apply a pre-trained single-image super-resolution model [95] to each generated image from different viewpoints. Experiments have demonstrated that this single-image super-resolution approach maintains consistency among multi-views, as it does not introduce significant variations to the images.

Texture Inpainting. After unwrapping the synthesized dense multi-view images into a texture map, a small set of patches in the UV texture remain that are not fully covered. To address this issue, we employ an intuitive inpainting approach. First, we project the existing UV texture into vertex texture. Then, we query each UV texel's texture by computing a weighted sum of the textures from the connected, textured vertices. The weights are set to be the reciprocal of the geometric distances between the texels and the vertices.

4.4 Implementation Details

Model Training. For training the multi-view image generation framework, we start by inheriting the ZSNR checkpoint of the Stable Diffusion 2 v-model [52]. We train our multi-view diffusion model using a self-collected large-scale 3D dataset. Multi-view images are rendered under the illumination of an even white light to accommodate our delighting model. Specifically, we render the reference image with a random azimuth and a fixed range of elevation from -20 to 20 degrees. This variation disrupts the consistency between the reference and generated images, thereby increasing the robustness of our texture generation framework. We directly train on 512×512 resolution with a total of 80,000 steps, a batch size of 48, and a learning rate of 5×10^{-5}. We use 1000 warm-up steps and the "trailing" scheduler proposed by ZSNR.

General Text- and Image-to-Texture. It is worth noting that Hunyuan3D-Paint not only generates high-quality texture maps for generated meshes but also supports arbitrary texture generation guided by any text or image input provided by the user for any geometric model. To achieve this, we leverage

	3DShape2VecSet [111]	Michelangelo [118]	Direct3D [98]	Hunyuan3D-ShapeVAE (Ours)
V-IoU(↑)	87.88%	84.93%	88.43%	**93.6%**
S-IoU(↑)	80.66%	76.27%	81.55%	**89.16%**

Table 1: Numerical comparisons. We evaluate the reconstruction performance of Hunyuan3D-ShapeVAE and baselines based on volume IoU (V-IoU) and Surface (S-IoU). The results indicate Hunyuan3D-ShapeVAE overwhelms all baselines in the reconstruction performance.

advanced T2I models and corresponding conditional generation modules, such as ControlNet [114] and IP-Adapter [107], to generate input images that align with geometric shapes based on user-provided text or image prompts. Benefitting from this paradigm, we are capable of texturing any specified geometry with arbitrary images, whether they are matched or mismatched. An application of using different images to texture the same geometry, dubbed as re-skinning, is illustrated in Fig. 9.

5 Evaluations

To thoroughly evaluate the performance of Hunyuan3D 2.0, we conducted experiments from three perspectives: (1) 3D Shape Generation (including Shape Reconstruction and Shape Generation), (2) Texture map synthesis, and (3) Textured 3D assets generation.

5.1 3D Shape Generation

Shape generation is crucial for 3D generation, as high-fidelity and high-resolution bare meshes form the foundation for downstream tasks. In this section, we compare and evaluate the capability of 3D shape generation in Hunyuan3D 2.0 from two perspectives: shape reconstruction and shape generation.

Baselines. We compare the reconstruction performance of Hunyuan3D-ShapeVAE with 3DShape2VecSet [111], Michelangelo [118], and Direct3D [98]. The mentioned methods represent the state-of-the-art ShapeVAE architecture, and the core differences are neural representations, where 3DShape2VecSet uses a downsampled vector set, point query; Michelangelo utilizes a learnable vector set, learnable query; Direct3D leverages learnable triplane; Hunyuan3D-ShapeVAE employees point query with importance sampling. Note that, except Direct3D, which requires a 3072 token length (suffering significant performance degeneration when reducing token length), all VAE models compare by 1024 token length. Hunyuan3D-DiT is compared with several state-of-the-art baselines. Open-source baselines are Michelangelo [118], Craftsman 1.5 [49], and Trellis [100]. Closed-source baselines are Shape Model 1, Shape Model 2, and Shape Model 3.

Metrics. We employ the Intersection of Union (IoU) to measure the reconstruction performance. Specifically, we compute randomly sampled volume points IoU (V-IoU) and near-surface region IoU (S-IoU) to reflect the reconstruction performance comprehensively. To evaluate shape generative performance, we employ ULIP [104] and Uni3D [121] to compute the similarity between the generated mesh and input images (ULIP-I and Uni3D-I) and the similarity between the generated mesh and images prompt synthesizing by the vision language model [13] (ULIP-T and Uni3D-T).

Shape Reconstruction Comparisons. The Numerical comparison of shape reconstruction is shown in Tab. 1. According to the table, Hunyuan3D-ShapeVAE overwhelms all baselines. Comparisons among Hunyuan3D-ShapeVAE, 3DShape2VecSet, and Michelangelo demonstrate the effectiveness of the importance sampling strategies. Fig. 6 illustrates the visual comparison of shape reconstruction, which shows that Hunyuan3D-ShapeVAE could faithfully recover the shape with fine-grained details and produce neat space without any floaters.

Shape Generation Comparisons. Tab. 2 shows the numerical comparison between Hunyuan3D-DiT and competing methods, which indicates that Hunyuan3D-DiT produces the most condition following results. Furthermore, according to the visual comparison in Fig. 7, results from Hunyuan3D-DiT follow the image prompt most, including clear human faces, surface bumps, logo texts, and layouts. Meanwhile, the generated bare mesh is holeless, which supports a solid basis for downstream tasks.
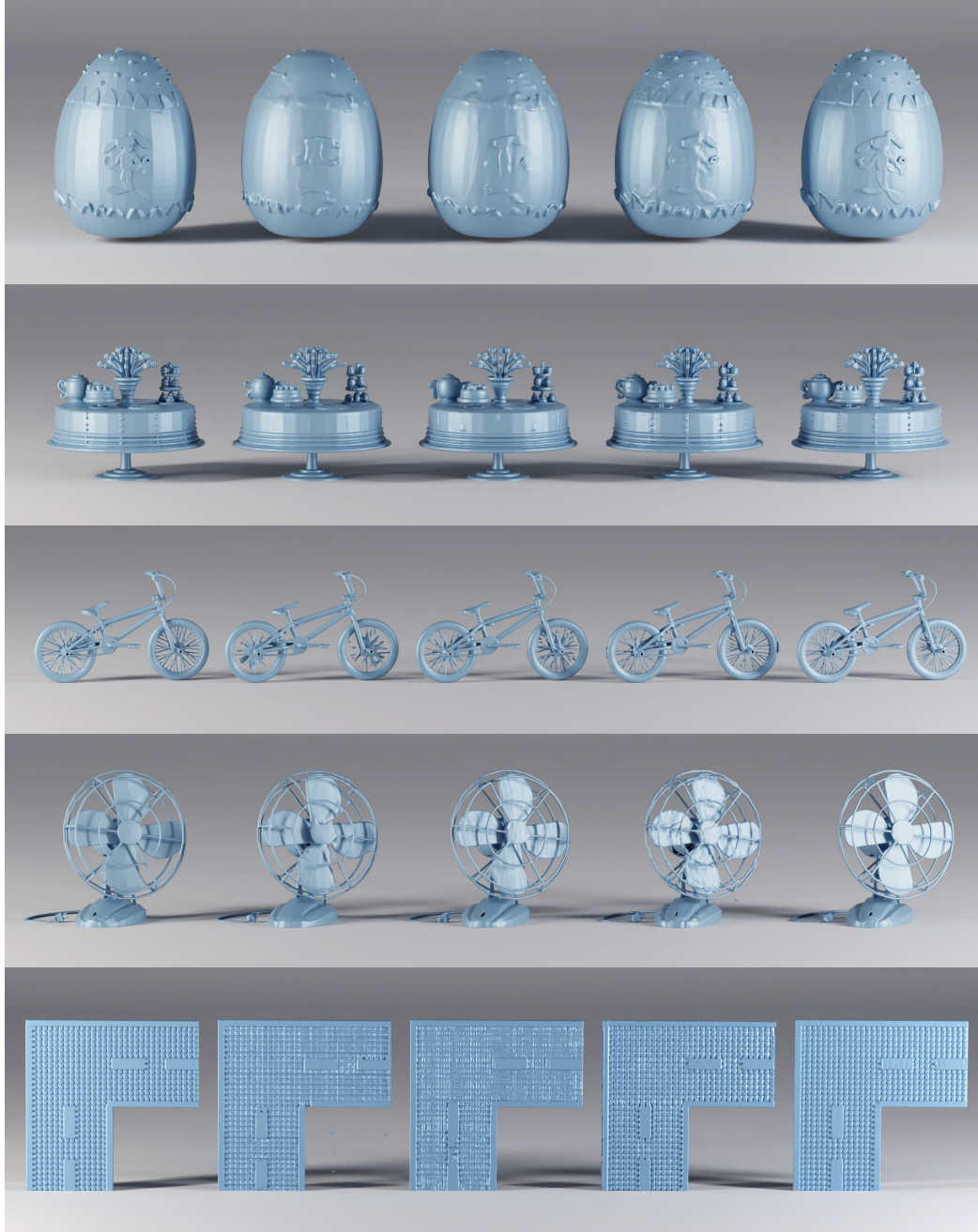
Ground Truth   Michelangelo   3DShape2Vectset   Direct3D   Hunyuan3D-ShapeVAE

Figure 6: Visual comparisons. We illustrate the reconstructed mesh (blue paint aims to show more details) in the figure, which showcases that only Hunyuan3D-ShapeVAE reconstructs mesh with fine-grained surface details and neat space. (Better viewed by zooming in.)

	ULIP-T($\uparrow$)	ULIP-I($\uparrow$)	Uni3D-T($\uparrow$)	Uni3D-I($\uparrow$)
Michelangelo [118]	0.0752	0.1152	0.2133	0.2611
Craftsman 1.5 [49]	0.0745	0.1296	0.2375	0.2987
Trellis [100]	0.0769	0.1267	0.2496	0.3116
Shape Model 1	<u>0.0799</u>	0.1181	0.2469	0.3064
Shape Model 2	0.0741	**0.1308**	0.2464	0.3106
Shape Model 3	0.0746	0.1284	<u>0.2516</u>	<u>0.3131</u>
Hunyuan3D-DiT (Ours)	**0.0771**	<u>0.1303</u>	**0.2519**	**0.3151**

Table 2: Numerical comparisons. By evaluating the shape generation performance on ULIP-T/I, Uni3D-T/I, demonstrating Hunyuan3D-DiT could produce the most condition followed results.

Input　Michelangelo　Craftsman　Trellis　Shape　Shape　Shape　Hunyuan3D-
　　　　　　　　　　1.5　　　　　　　Model 1　Model 2　Model 3　DiT

Figure 7: Visual comparisons. We display the input image and the generated bare mesh (blue paint aims to show more details) from all methods in the figure. The human faces and piano keys show that Hunyuan3D-DiT could synthesize detailed surface bumps, maintaining completeness. Several scenes or logos demonstrate that Hunyuan3D-DiT could generate intricate details. (Better viewed by zooming in.)

	CMMD($\downarrow$)	FID_{CLIP}($\downarrow$)	CLIP-score($\uparrow$)	LPIPS($\downarrow$)
TEXTure [73]	3.047	35.75	0.8499	0.0076
Text2Tex [9]	2.811	31.72	0.8680	0.0071
SyncMVD [59]	2.584	29.93	0.8751	0.0063
Paint3D [110]	2.810	30.29	0.8724	0.0063
TexPainter [112]	2.483	28.83	0.8789	0.0062
Hunyuan3D-Paint (Ours)	**2.318**	**26.44**	**0.8893**	**0.0059**

Table 3: Numerical comparisons. We compare Hunyuan3D-Paint with baselines on various metrics, and the results indicate that our model could produce the most condition-conforming texture maps.

5.2 Texture Map Synthesis

As texture maps directly influence the visual appeal of textured 3D assets, we conduct comprehensive text-conditioned texture map synthesis experiments to validate the performance of Hunyuan3D-Paint.

Baselines. We compare Hunyuan3D-Paint with the following texture generation methods, including TEXTure [73], Text2Tex [9], SyncMVD [59], Paint3D [110], and TexPainter [112]. All the baselines leverage geometric and diffusion priors to facilitate the overall generation quality of texture maps.

Metrics. We apply several frequently used image-level metrics to enable a fair comparison of texture map generation. Specifically, we leverage a CLIP-version of Fréchet Inception Distance FID_{CLIP} to compute the distance between the rendering of the generated textured map in semantic perspectives. We use the implementation of Clean-FID [66]. Besides, the recently introduced CLIP Maximum-Mean Discrepancy (CMMD) [38] is utilized to serve as another important criterion, which is a more accurate measurement of images with rich details. In addition to these two metrics, we also use CLIP-score [71] to validate semantic alignment between renderings of the generated texture map and given prompt and LPIPS [116] to estimate the consistency between renderings of the generated texture map and ground-truth images.

Comparisons. The numerical comparison of text-to-texturing is shown in Tab. 3, showcasing that Hunyuan3D-Paint achieves the best generative quality and semantic following. The visual comparison refers to Fig. 8. The fish and rabbit show that our model produces the most condition-following results. And the football demonstrates the ability of Hunyuan3D-Paint to produce clear texture maps. The texture map of the castle and bear contains rich texture patterns, showcasing that our model can produce intricate details.

Applications. All generated texture maps are seamless and lighting-invariant. Moreover, Hunyuan3D-Paint is flexible to produce various texture maps for bare mesh or hand-crafted mesh according to different prompts. As shown in the Fig. 9, our model produces different texture maps for a mesh with seamless and intricate details.

5.3 Textured 3D Assets Generation

In this section, we evaluate the generated textured 3D assets for reflecting the end-to-end generation capabilities of Hunyuan3D 2.0.

Baselines. We compare Hunyuan3D 2.0 against leading models in the field, including open-source model Trellis [100] and closed-source models Model 1, Model 2, and Model 3.

Metrics. We mainly measure the generative quality of textured 3D assets by their renderings. Similar to Sec. 5.2, we employ FID_{CLIP} to compute the image content distance, CLIP-score to reflect semantic alignment, CMMD to measure the similarity in the image details, and LPIPS to evaluate the consistency between rendering from generated textured 3D assets and given image prompts.

Comparisons. The numerical results reported in the Tab. 4 indicate that Hunyuan3D 2.0 surpasses all baselines in the quality of generated textured 3D assets and the condition following ability. The illustration in the Fig. 11 demonstrates that Hunyuan3D 2.0 produces the textured 3D assets with the highest quality. Even for the text in the image prompt, our model can produce the correct bumps on the shape surface and an accurate texture map according to the geometric conditions. The rest of the cases demonstrate the ability of our model to generate high-resolution and high-fidelity results with complex actions or scenes.

*A fish with **orange and pink scales***

*A bunny crafted from **colorful, hand-painted ceramic tiles***

*A futuristic dragon with sleek, **metallic blue and silver surfaces***

*A photo of a soccer ball showing a **pentagon-shaped black section** surrounded **by white panels***

*A castle with a central tower and four turrets, featuring a mix of **dark and light stone textures***

*A teddy bear wearing a **striped scarf**, standing on a **wooden base with grass trim***

| TEXTure | Text2Tex | SyncMVD | Paint3D | TexPainter | Ours |

Figure 8: Visual comparisons. We demonstrate several generated texture maps on different bare meshes. The fish and rabbit texture map showcases that Hunyuan3D-Paint produces the most text-conforming results. The football indicates that our model could synthesize seamless and clean texture maps. Moreover, Hunyuan3D-Paint could generate complex texture maps, like the castle and bear. (Better viewed by zooming in.)

14

Figure 9: Visual results. We generate different texture maps for two meshes, and the results validate the performance of Hunyuan3D-Paint on texture reskinning. (Better viewed by zooming in.)

	CMMD($\downarrow$)	FID$_{CLIP}$($\downarrow$)	FID$_{Incept}$($\downarrow$)	CLIP-score($\uparrow$)
Trellis [100]	3.591	54.639	289.287	0.787
Model 1	3.600	55.866	305.922	0.779
Model 2	3.368	49.744	294.628	0.806
Model 3	3.218	51.574	295.691	0.799
Hunyuan3D 2.0 (Ours)	**3.193**	**49.165**	**282.429**	**0.809**

Table 4: Numerical comparison. According to the results, Hunyuan3D-Paint produces the most condition-following texture maps.

User Study. In addition, we conducted a user study by randomly inviting 50 volunteers to evaluate 300 unselected results generated by Hunyuan3D 2.0 subjectively. The evaluation criteria included 1) overall visual quality, 2) adherence to image conditions, and 3) overall satisfaction (dissatisfaction in either 1 or 2 results in overall dissatisfaction). The user study results in Fig. 10 indicate that Hunyuan3D 2.0 outperforms comparative methods, particularly in its ability to adhere to image conditions.
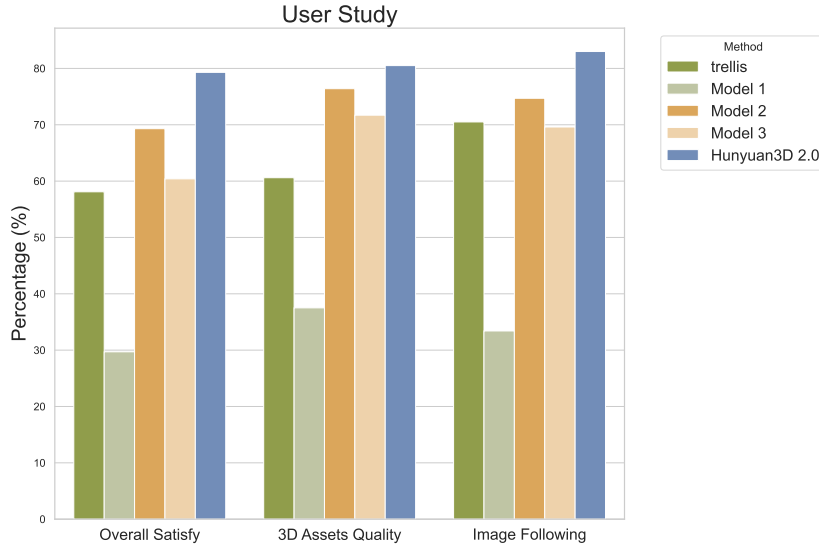

Figure 10: The results of user study.

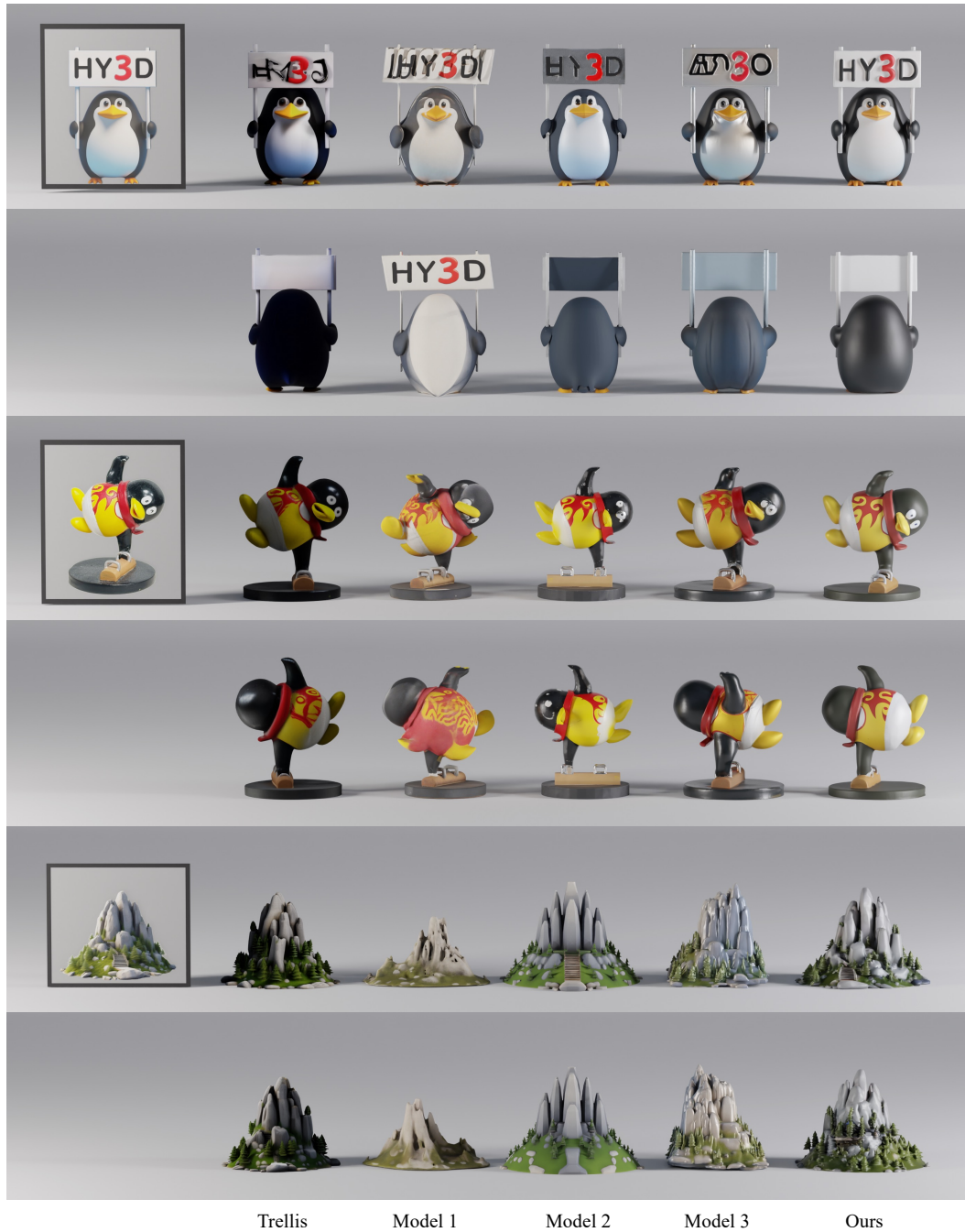|				
Trellis	Model 1	Model 2	Model 3	Ours

Figure 11: Visual comparisons. The first case reflects that Hunyuan3D 2.0 could synthesize detailed surface bumps and correct texture maps. The second penguin showcases our model's ability to handle complex actions. The last mountain demonstrates that Hunyuan3D-DiT could produce intricate structures, and Hunyuan3D-Paint can synthesize vivid texture maps. (Better viewed by zooming in.)

6 Hunyuan3D-Studio

We have developed Hunyuan3D-Studio. This platform includes a comprehensive set of tools for the 3D production pipeline, as illustrated in Fig. 1. Hunyuan3D-Studio aims to provide experts and novices with a no-frills way to engage in 3D generation production and research. In this section, we highlight several features of Hunyuan3D-Studio, including Sketch-to-3D, Low-polygon Stylization, and Autonomous Character Animator. These features aim to streamline the 3D creation process and make it accessible to a broader audience.

6.1 Sketch-to-3D

In game development and content creation, converting 2D sketches into 3D assets is a crucial technology that significantly enhances digital artistry design efficiency and flexibility. Previous methods [1, 120, 117, 32] suffer from the lack of the generative foundation model. They tackle this by training a small-scale generative or reconstruction model with sketch images as input directly on a limited dataset, significantly limiting the model's capabilities.

Benefitting from Hunyuan3D 2.0, we could convert sketches to images with rich details as input to the foundation 3D generative model. Specifically, the Hunyuan3D-Studio has developed the Sketch-to-3D module, which first converts sketches to images with rich details, maintaining original contours. Then, synthesize high-resolution and high-fidelity textured 3D assets, significantly reducing the barrier for users to engage in content creation.

As shown in Fig. 1, the Sketch-to-3D module can generate highly detailed and realistic 3D assets while maintaining close consistency with the original sketches. With this technology, users can synthesize 3D content with a simple sketch, providing a powerful tool for game developers and digital artists and a low-barrier creation platform for ordinary users.

6.2 Low-polygon Stylization

Low-polygon stylization is critical in many computer graphics (CG) pipelines, as the face count of a mesh significantly impacts the application of 3D assets. Low-polygon stylization can significantly reduce computational costs, making it an essential process in 3D asset management. To address this, we have established a low-polygon stylization module that efficiently converts the dense meshes generated by Hunyuan3D 2.0 into low-polygon meshes. This module operates in two steps: geometric editing and texture preserving.

For geometric editing, we employ a faster and more robust traditional method [28, 35], despite the recent auto-regressive transformer-based polygon-generation approaches [96, 103, 86, 12]. By setting an optimization criterion, we merge the vertices of the mesh to transform the dense mesh into a low-polygon mesh. As shown at the top-right of Fig. 1, each 3D model can be represented by only dozens of triangles after geometric editing. The change in the face count of the mesh causes significant deviations in the vertices and faces of the low-polygon mesh compared to the dense mesh. Therefore, to preserve the texture patterns of the textured 3D assets, we construct a KD-tree for the input dense mesh. We then use the nearest-neighbor search within the KD tree to query the texture colors for the vertices of the low-polygon mesh. Finally, we obtain the texture map for the low-polygon mesh by performing texture baking on the low-polygon mesh with vertex colors. This process ensures that the visual quality of the textures is maintained while optimizing the mesh structure for production-level textured 3D assets.

6.3 3D Character Animation

Hunyuan3D 2.0 generates static 3D assets with high-resolution shapes and texture maps. However, drivable 3D models yet have broad requirements [82, 81, 84, 83, 57], such as game development and animation production. To extend the range of applications of Hunyuan3D 2.0, we develop a 3D character animation function in Hunyuan3D-Studio. The animation algorithm inputs the generated character and extracts features from mesh vertices and edges. Then, we utilize the Graph Neural Network (GNN) to detect skeleton key points and assign skinning weights to the mesh surface. Finally, based on the predicted skeleton skinning and motion templates, the algorithm utilizes motion

retargeting to drive the character. Some frames are displayed in Fig. 1. With 3D character animation, the generated results from Hunyuan3D 2.0 can come to life.

7 Related Work

7.1 3D Shape Generation

Representations. The field of shape generation has undergone significant advancements, driven by the unique challenges associated with the 3D modality. Unlike other modalities, 3D data lacks a universal storage and representation format, leading to diverse approaches in shape generation research. The primary 3D representations include voxels, point clouds, polygon meshes, and implicit functions. With the advent of seminal works such as 3D ShapeNets [99], IF-Net [14], and 3D Gaussian Splatting [40], implicit functions, and 3D Gaussian Splitting have become prevalent in shape generation. However, even lightweight and flexible representations like implicit functions impose substantial modeling and computational burdens on deep neural networks. As a result, neural representations of 3D shapes have emerged as a new research focus, aiming to enhance efficiency. Pioneering methods such as 3DShape2VecSet [111], Michelangelo [118], CLAY [113], and Dora [11] represent 3D shapes using vector sets (one-dimensional latent token sequences proposed by 3DShape2VecSet [111]), significantly improving representation efficiency. Another approach involves structured representations (e.g., triplane [68, 7, 26] or sparse volume [63, 119, 72]) to encode 3D shapes, which better preserve spatial priors but are less efficient than vector sets. Inspired by recent advances in Latent Diffusion Models, Hunyuan3D 2.0 employs vector sets to represent 3D shapes' implicit functions, alleviating the compression and fitting demands on neural networks and achieving a breakthrough in shape generation performance.

Shape Generative models. The evolution of generative model paradigms has continually influenced shape generation research. Early works [97, 76, 105, 109] based on Variational Auto-encoder [41], generative adversarial networks (GANs) [29], normalizing flow [65], and auto-regressive modeling [31] demonstrated strong generative capabilities within several specific categories. The success of diffusion models [33] and their variants [58, 53] in text-conditioned image [74, 45] generation has spurred the popularity of diffusion-based shape generative models, with notable works [16, 118] achieving stable cross-category shape generation. Additionally, advancements in network architectures have propelled shape-generation research. The transition from early 3D Convolutional Neural Networks [30] to the now-common Transformer architectures [93, 67] has led to the development of classic shape generation networks, enhancing performance. Building on these advancements, Hunyuan3D 2.0 employs a flow-based scalable transformer, further improving the model's shape generation capabilities.

Large-scale Dataset. Large-scale datasets are the cornerstone of scaling laws. However, the scale of 3D data is much smaller than that in large language models and image generation fields. From 3Dscanrep [18, 44, 92] to ShapeNet [8], the growth of 3D datasets has been gradual [85, 122, 25, 77, 102, 21, 17]. The release of objaverse [20] and objaverse-xl [19] has been a significant driver in realizing the scaling law for shape generation. Leveraging these open-source 3D datasets, Hunyuan3D-2.0 can generate high-fidelity and high-resolution 3D assets.

Benefiting from these open-source algorithms and 3D datasets, Hunyuan3D 2.0 is capable of generating high-fidelity and high-resolution 3D assets. Therefore, we have released Hunyuan3D 2.0 to contribute to the open-source 3D generation community and further advanced 3D generation algorithms.

7.2 Texture Map Synthesis

High-quality texture-map synthesis has been a long-standing topic within the computer graphics community. Its significance has only increased with the growing demand for end-to-end 3D generation techniques, where it plays a crucial role for appearance modeling.

Text to Texture. Given a plain mesh, text/image to texture aims to generate a high-quality texture that aligns well with the given geometry according to a guided text and image. Early attempts tried to approach texture synthesis by harnessing the categorical information and train a generative model on a specified dataset [10, 5, 23, 80, 26, 27]. While achieving plausible texturing results, these

methods failed to generalize to objects of other categories, limiting their applicability in production environments.

More recently, Stable Diffusion [74], owing to its impressive text-guided image generation capability and flexible structure, has spawned a plethora of text-to-texture research. To take full advantage of pre-trained image diffusion models, most subsequent works have approached the texture synthesis problem as a geometry-conditioned multi-view images generation problem.

Initially, score distillation was adopted to harness the generation power of image diffusion models for 3D content (texture) synthesis [87, 51, 62, 70]. However, these methods are often limited by the over-saturated colors and misalignment with geometry.

Subsequently, optimization-free approaches pioneered by TEXTure [73] have been introduced [101, 15, 56, 112, 110, 59, 9]. To ensure consistency across multi-view images, these methods either adopt an inpainting framework by specifying viewpoint-related masks or employ a "synchronizing" operation during the denoising process. However, since Stable Diffusion is trained on a dataset with a noticeable forward-facing viewpoint bias [55], these training-free methods are limited and often suffer from severe performance issues, such as the Janus problem and multi-view inconsistency, which result in textures with significant artifacts.

With the development of extensive 3D datasets, training multi-view diffusion models has become a prevailing direction for texture generation [61, 2], exhibiting more powerful capabilities on texture consistency than the training-free approaches.

Image to Texture. In a related direction, image-guided texture generation has garnered attention in recent months, aligning closely with our research focus. This relatively unexplored area of image-guided texture synthesis demonstrates significant potential for further development since images provide more diverse information than text prompts, and text-to-texture generation can be fully replaced by a text-to-image and image-to-texture pipeline. Unfortunately, most of the existing works focus on semantic alignment with the reference image rather than precise alignment. FlexiTex [39] and EASI-Tex [42] both utilize an IP-Adapter [107] for image prompt injection. While TextureDreamer [108] employs a DreamBooth-like [75] approach to facilitate texture transfer across different objects.

However, we argue that there are two explicit advantages to exactly following every detail of the reference image. First, as part of an end-to-end image-guided 3D generation process, the geometry generated in the first stage strives to align with the reference image, while the appearance details are left for texture synthesis stage. Thus, one of the main objectives of our texture generation framework is to enhance the geometry with more detailed appearance features from the well-aligned reference image. Second, with the rapid development of image diffusion techniques, more exquisite reference images are now available. Carefully adhering to these details can significantly improve the quality of the generated textures. Based on these advantages, Hunyuan3D-Paint is designed with a detailed preserving image injection module according to the philosophy of aligning the reference image not only semantically but also following the details as closely as possible.

Multi-view Images Generation. Due to the viewpoint bias and multi-view inconsistency inherent in training-free image diffusion models, multi-view image diffusion was developed to alleviate these issues by utilizing large-scale 3D datasets, such as objaverse and objaverse-xl [20, 19].

Most works force the multi-view generated latents to communicate with each other by manipulating the self-attention layers with 3D-aware masks [36, 48, 88, 60, 94, 79, 78, 55]. For example, Zero123++ [78] first treats the multi-view attention as a self-attention on a large image, which is the spatial concatenation of six multi-view images. MVDiffusion [89] applies a correspondence-aware attention (CAA) to inform the model to focus only on the correlation among the spatially-close pixels. MVAdapter [36], following Era3D [48] implements a simpler but effective row-wise and column-wise attention to alleviate the computational burden of CAA and achieves comparable performance.

Inspired by these works, we propose a multi-view generation framework equipped with a multi-task attention mechanism to achieve both multi-view consistency and image alignment simultaneously. Benefiting from this careful design and being trained on a large 3D rendering dataset, Hunyuan3D-Paint is able to achieve high-quality, consistent textures with strong alignment to the reference image.

8 Conclusion

In this report, we introduce an open-source 3D creation system—Hunyuan3D 2.0 —for generating textured meshes from images. We present Hunyuan3D-ShapeVAE, which is trained using a novel importance sampling method. This approach compresses each 3D object into a few latent tokens while minimizing reconstruction losses. Building on our VAE, we developed Hunyuan3D-DiT, an advanced diffusion transformer capable of generating visually appealing shapes that align precisely with input images. Besides, we introduce Hunyuan3D-Paint, another diffusion model designed to create textures for both our generated meshes and user-crafted meshes. With several innovative designs, our texture generation model, in conjunction with our shape generation model, can produce high-resolution, high-fidelity textured 3D assets from a single image. As we continue to make progress, we hope that Hunyuan3D 2.0 will serve as a robust baseline for large-scale 3D foundation models within the open-source community and facilitate future research endeavors.

9 Contributors

- **Project Sponsors:** Jie Jiang, Yuhong Liu, Di Wang, Yong Yang, Tian Liu
- **Project Leaders:** Chunchao Guo, Jingwei Huang
- **Core Contributors:**
    - **Data:** Lifu Wang, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Chao Zhang, Yonghao Tan
    - **Shape Generation:** Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Zibo Zhao
    - **Texture Synthesis:** Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang
    - **Downstream Tasks:** Xianghui Yang, Huiwen Shi, Sicong Liu, Junta Wu, Yihang Lian, Fan Yang, Ruining Tang, Zebin He, Xinzhou Wang, Jian Liu, Xuhui Zuo
    - **Studio:** Zhuo Chen, Biwen Lei, Haohan Weng, Jing Xu, Yiling Zhu, Xinhai Liu, Lixin Xu, Changrong Hu, Tianyu Huang
- **Contributors:** Jie Xiao, Yangyu Tao, Jianchen Zhu, Jinbao Xue, Kai Liu, Chongqing Zhao, Xinming Wu, Zhichao Hu, Lei Qin, Jianbing Peng, Zhan Li, Minghui Chen, Xipeng Zhang, Lin Niu, Paige Wang, Yingkai Wang, Haozhao Kuang, Zhongyi Fan, Xu Zheng, Weihao Zhuang, YingPing He

References

[1] Hmrishav Bandyopadhyay, Subhadeep Koley, Ayan Das, Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Doodle your 3d: From abstract freehand sketches to precise 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9795–9805, 2024.

[2] Raphael Bensadoun, Yanir Kleiman, Idan Azuri, Omri Harosh, Andrea Vedaldi, Natalia Neverova, and Oran Gafni. Meta 3d texturegen: Fast and consistent texture generation for 3d objects. *arXiv preprint arXiv:2407.02430*, 2024.

[3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

[4] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[5] Alexey Bokhovkin, Shubham Tulsiani, and Angela Dai. Mesh2tex: Generating mesh textures from image queries. In *IEEE International Conference on Computer Vision (ICCV)*, October 2023.

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022.

[7] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022.

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

[9] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18558–18568, 2023.

[10] Qimin Chen, Zhiqin Chen, Hang Zhou, and Hao Zhang. Shaddr: Interactive example-based geometry and texture generation via 3d shape detailization and differentiable rendering. In *SIGGRAPH Asia 2023 Conference Papers*, 2023.

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

[12] Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555*, 2024.

[13] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[14] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5939–5948, 2019.

[15] Wei Cheng, Juncheng Mu, Xianfang Zeng, Xin Chen, Anqi Pang, Chi Zhang, Zhibin Wang, Bin Fu, Gang Yu, Ziwei Liu, et al. Mvpaint: Synchronized multi-view diffusion for painting anything 3d. *arXiv preprint arXiv:2411.02336*, 2024.

[16] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023.

[17] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. *CVPR*, 2022.

[18] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[19] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023.

[20] 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. *arXiv preprint arXiv:2212.08051*, 2022.

[21] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items, 2022.

[22] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[23] Aysegul Dundar, Jun Gao, Andrew Tao, and Bryan Catanzaro. Fine detailed texture learning for 3d meshes with generative models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

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

[25] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

[26] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022.

[27] Lin Gao, Tong Wu, Yu-Jie Yuan, Ming-Xian Lin, Yu-Kun Lai, and Hao Zhang. Tm-net: Deep generative networks for textured meshes. *ACM Trans. Graph.*, 40(6):1–15, 2021.

[28] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997.

[29] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[30] Ben Graham. Sparse 3d convolutional neural networks. *arXiv preprint arXiv:1505.02890*, 2015.

[31] Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. In *International Conference on Machine Learning*, pages 1242–1250. PMLR, 2014.

[32] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and Pascal Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13023–13032, 2021.

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

[34] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. In *The Twelfth International Conference on Learning Representations*, 2024.

[35] Hugues Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *Proceedings Visualization'99 (Cat. No. 99CB37067)*, pages 59–510. IEEE, 1999.

[36] Zehuan Huang, Yuanchen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. *arXiv preprint arXiv:2412.03632*, 2024.

[37] Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. Make-a-shape: a ten-million-scale 3d shape model. In *Forty-first International Conference on Machine Learning*, 2024.

[38] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 9307–9315, 2024.

[39] DaDong Jiang, Xianghui Yang, Zibo Zhao, Sheng Zhang, Jiaao Yu, Zeqiang Lai, Shaoxiong Yang, Chunchao Guo, Xiaobo Zhou, and Zhihui Ke. Flexitex: Enhancing texture generation with visual guidance. *arXiv preprint arXiv:2409.12431*, 2024.

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

[41] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[42] Perla Sai Raj Kishore, Yizhi Wang, Ali Mahdavi-Amiri, and Hao Zhang. EASI-Tex: Edge-aware mesh texturing from single-image. *ACM Transactions on Graphics (Special Issue of SIGGRAPH)*, 43(4), 2024.

[43] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[44] Venkat Krishnamurthy and Marc Levoy. Fitting smooth surfaces to dense polygon meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 313–324, 1996.

[45] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[46] Yushi Lan, Fangzhou Hong, Shuai Yang, Shangchen Zhou, Xuyi Meng, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff: Scalable latent neural fields diffusion for speedy 3D generation. In *European Conference on Computer Vision (ECCV)*, 2024.

[47] Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. Gaussiananything: Interactive point cloud latent diffusion for 3d generation. In *ICLR*, 2025.

[48] Peng Li, Yuan Liu, Xiaoxiao Long, Feihu Zhang, Cheng Lin, Mengfei Li, Xingqun Qi, Shanghang Zhang, Wenhan Luo, Ping Tan, et al. Era3d: High-resolution multiview diffusion using efficient row-wise attention. *arXiv preprint arXiv:2405.11616*, 2024.

[49] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner, 2024.

[50] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, Dayou Chen, Jiajun He, Jiahao Li, Wenyue Li, Chen Zhang, Rongwei Quan, Jianxiang Lu, Jiabin Huang, Xiaoyan Yuan, Xiaoxiao Zheng, Yixuan Li, Jihong Zhang, Chao Zhang, Meng Chen, Jie Liu, Zheng Fang, Weiyan Wang, Jinbao Xue, Yangyu Tao, Jianchen Zhu, Kai Liu, Sihuan Lin, Yifu Sun, Yun Li, Dongdong Wang, Mingtao Chen, Zhichao Hu, Xiao Xiao, Yan Chen, Yuhong Liu, Wei Liu, Di Wang, Yong Yang, Jie Jiang, and Qinglin Lu. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding, 2024.

[51] 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 *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2023.

[52] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5404–5411, 2024.

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

[54] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024.

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

[56] Shang Liu, Chaohui Yu, Chenjie Cao, Wen Qian, and Fan Wang. Vcd-texture: Variance alignment based 3d-2d co-denoising for text-guided texturing. In *European Conference on Computer Vision*, pages 373–389. Springer, 2025.

[57] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5904–5913, 2019.

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

[59] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

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

[61] Jiawei Lu, Yingpeng Zhang, Zengjun Zhao, He Wang, Kun Zhou, and Tianjia Shao. Genesistex2: Stable, consistent and high-quality text-to-texture generation. *arXiv preprint arXiv:2409.18401*, 2024.

[62] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 12663–12673, 2023.

[63] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022.

[64] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[65] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[66] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2022.

[67] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[68] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020.

[69] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

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

[71] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[72] 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*, pages 4209–4219, 2024.

[73] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023.

[74] 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*, pages 10684–10695, 2022.

[75] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[76] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18603–18613, 2022.

[77] Pratheba Selvaraju, Mohamed Nabail, Marios Loizou, Maria Maslioukova, Melinos Averkiou, Andreas Andreou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. Buildingnet: Learning to label 3d buildings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10397–10407, October 2021.

[78] 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.

[79] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *The Twelfth International Conference on Learning Representations*, 2023.

[80] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces. In *European Conference on Computer Vision (ECCV)*, pages 72–88. Springer, 2022.

[81] Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.

[82] Sebastian Starke, Paul Starke, Nicky He, Taku Komura, and Yuting Ye. Categorical codebook matching for embodied character controllers. *ACM Transactions on Graphics (TOG)*, 43(4):1–14, 2024.

[83] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)*, 39(4):54–1, 2020.

[84] Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics (TOG)*, 40(4):1–16, 2021.

[85] Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1798–1808, 2021.

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

[87] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

[88] Shitao Tang, Jiacheng Chen, Dilin Wang, Chengzhou Tang, Fuyang Zhang, Yuchen Fan, Vikas Chandra, Yasutaka Furukawa, and Rakesh Ranjan. Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 175–191. Springer, 2025.

[89] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv*, 2023.

[90] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[91] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[92] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, 1994.

[93] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

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

[95] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops (ECCVW)*, September 2018.

[96] Haohan Weng, Zibo Zhao, Biwen Lei, Xianghui Yang, Jian Liu, Zeqiang Lai, Zhuo Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, et al. Scaling mesh generation via compressive tokenization. *arXiv preprint arXiv:2411.07025*, 2024.

[97] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

[98] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *arXiv preprint arXiv:2405.14832*, 2024.

[99] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[100] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[101] Xiaoyu Xiang, Liat Sless Gorelik, Yuchen Fan, Omri Armstrong, Forrest Iandola, Yilei Li, Ita Lifshitz, and Rakesh Ranjan. Make-a-texture: Fast shape-aware texture generation in 3 seconds. *arXiv preprint arXiv:2412.07766*, 2024.

[102] Jiacong Xu, Yi Zhang, Jiawei Peng, Wufei Ma, Artur Jesslen, Pengliang Ji, Qixin Hu, Jiehua Zhang, Qihao Liu, Jiahao Wang, et al. Animal3d: A comprehensive dataset of 3d animal pose and shape. *arXiv preprint arXiv:2308.11737*, 2023.

[103] Jingwei Xu, Chenyu Wang, Zibo Zhao, Wen Liu, Yi Ma, and Shenghua Gao. Cad-mllm: Unifying multimodality-conditioned cad generation with mllm. *arXiv preprint arXiv:2411.04954*, 2024.

[104] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1179–1189, 2023.

[105] Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6239–6249, 2022.

[106] Xianghui Yang, Huiwen Shi, Bowen Zhang, Fan Yang, Jiacheng Wang, Hongxu Zhao, Xinhai Liu, Xinzhou Wang, Qingxiang Lin, Jiaao Yu, et al. Hunyuan3d-1.0: A unified framework for text-to-3d and image-to-3d generation. *arXiv preprint arXiv:2411.02293*, 2024.

[107] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023.

[108] Yu-Ying Yeh, Jia-Bin Huang, Changil Kim, Lei Xiao, Thu Nguyen-Phuoc, Numair Khan, Cheng Zhang, Manmohan Chandraker, Carl S Marshall, Zhao Dong, et al. Texturedreamer: Image-guided texture synthesis through geometry-aware diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4304–4314, 2024.

[109] Fukun Yin, Xin Chen, Chi Zhang, Biao Jiang, Zibo Zhao, Jiayuan Fan, Gang Yu, Taihao Li, and Tao Chen. Shapegpt: 3d shape generation with a unified multi-modal language model. *arXiv preprint arXiv:2311.17618*, 2023.

[110] Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4252–4262, 2024.

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

[112] Hongkun Zhang, Zherong Pan, Congyi Zhang, Lifeng Zhu, and Xifeng Gao. Texpainter: Generative mesh texturing with multi-view consistency. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

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

[114] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[115] Lyumin Zhang. https://github.com/mikubill/sd-webui-controlnet/discussions/1236. https://github.com/Mikubill/sd-webui-controlnet/discussions/1236, 2023.

[116] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[117] Song-Hai Zhang, Yuan-Chen Guo, and Qing-Wen Gu. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6012–6021, 2021.

[118] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems*, 36, 2024.

[119] Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. *ACM Transactions on Graphics (ToG)*, 42(4):1–13, 2023.

[120] Jie Zhou, Zhongjin Luo, Qian Yu, Xiaoguang Han, and Hongbo Fu. Ga-sketching: Shape modeling from multi-view sketching with geometry-aligned deep implicit functions. In *Computer Graphics Forum*. Wiley Online Library, 2023.

[121] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[122] Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.