Hunyuan3D 2.1: From Images to High-Fidelity 3D Assets with Production-Ready PBR Material

Tencent Hunyuan

https://github.com/Tencent-Hunyuan/Hunyuan3D-2.1

Figure 1: Gallery of 3D assets generated by **Hunyuan3D 2.1.**

Abstract

3D AI-generated content (AIGC) is a passionate field that has significantly accelerated the creation of 3D models in gaming, film, and design. Despite the development of several groundbreaking models that have revolutionized 3D generation, the field remains largely accessible only to researchers, developers, and designers due to the complexities involved in collecting, processing, and training 3D models. To address these challenges, we introduce Hunyuan3D 2.1 as a case study in this tutorial. This tutorial offers a comprehensive, step-by-step guide on processing 3D data, training a 3D generative model, and evaluating its performance using Hunyuan3D 2.1, an advanced system for producing high-resolution, textured 3D assets. The system comprises two core components: the Hunyuan3D-DiT for shape generation and the Hunyuan3D-Paint for texture synthesis. We will explore the entire workflow, including data preparation, model architecture, training strategies, evaluation metrics, and deployment. By the conclusion of this tutorial, you will have the knowledge to finetune or develop a robust 3D generative model suitable for applications in gaming, virtual reality, and industrial design.

1 Introduction

While recent breakthroughs in 2D image and video generation—powered by diffusion models [1, 2, 3, 4, 5, 6]—have revolutionized content creation, the field of 3D generative modeling lags behind. Current methods for 3D asset synthesis remain fragmented, with incremental progress in foundational techniques such as latent representation learning [7], geometric refinement [8, 9, 10], and texture synthesis [11, 12, 13]. Among these, CLAY [11] marks a milestone as the first framework to demonstrate the viability of diffusion models for high-quality 3D generation. Yet, unlike the thriving open-source ecosystems in image (*e.g.*, Stable Diffusion [2]), language (*e.g.*, LLaMA [14]), and video (*e.g.*, HunyuanVideo [5], and Wan 2.1 [6]), the 3D domain lacks a robust, scalable foundation model to drive widespread innovation.

To bridge this gap, we introduce *Hunyuan3D 2.1*, a comprehensive 3D asset creation system to generate a textured mesh from a single image input. It is mainly built on two fully open-source foundation models: 1) Hunyuan3D-DiT: A shape-generation model combining a flow-based diffusion architecture with a high-fidelity mesh autoencoder (Hunyuan3D-ShapeVAE); 2) Hunyuan3D-Paint: a mesh-conditioned multi-view diffusion model for PBR material generation, producing high-quality, multi-channel-aligned, and view-consistent textures.

For shape generation, we leverage Hunyuan3D-ShapeVAE and Hunyuan3D-DiT to achieve high-quality and high-fidelity shape generation. Specifically, Hunyuan3D-ShapeVAE employs mesh surface importance sampling to enhance sharp edges and variational token length to improve intricate geometric details. Hunyuan3D-DiT inherits the recent advanced flow matching models [15, 3] to construct a scalable and flexible diffusion model.

For texture synthesis, Hunyuan3D-Paint introduces a multi-view PBR diffusion that generates albedo, metallic, and roughness maps for meshes. Notably, Hunyuan3D-Paint incorporates a spatial-aligned multi-attention module to align albedo and MR maps, 3D-aware RoPE to enhance cross-view consistency, and an illumination-invariant training strategy to produce light-free albedo maps robust to varying lighting conditions.

Hunyuan3D 2.1 separates shape and texture generation into distinct stages, an more advanced strategy proven effective upon previous large reconstruction models [16, 17, 18, 19, 20, 21, 22, 23]. This modularity allows users to generate untextured meshes only or apply textures to custom assets, enhancing flexibility for industrial applications.

We rigorously evaluate *Hunyuan3D 2.1* against leading commercial and recent open-source models, *e.g.*, Michelangelo [8], Craftsman 1.5 [24], Trellis [25], TripoSG [9], Step1X-3D [26] and Direct3D-S2 [27]. Quantitative metrics and visual comparisons confirm its superiority in geometric detail preservation, texture-photo consistency, and human preference.

This tutorial unpacks the architecture, data processing, training, and evaluation of *Hunyuan3D 2.1*, providing practitioners with the tools to harness its capabilities for diverse 3D generation tasks.

2 Data Processing

In this section, we aim to describe the data processing for training the shape generation model and texture model. We start to introduce the dataset preparation, and then present how to obtain the relevant training and testing data for the shape generation model and texture model.

2.1 Dataset collection

For shape generation, we collect 100K+ textured and untextured 3D data from public datasets and custom datasets. The public dataset comes mainly from ShapeNet [28], ModelNet40 [29], Thingi10K [30], and Objaverse [31, 32]. For texture synthesis, we filter 70K+ human-annotated high-quality data following strict curation protocols from Objaverse-XL [32].

2.2 Data preprocessing for shape generation

2.2.1 Normalization

The normalization process begins by calculating the axis-aligned bounding box for each 3D object, ensuring all subsequent operations work in a standardized coordinate space. We apply uniform scaling to fit the object within a unit cube centered at the origin, preserving aspect ratios while maintaining consistent scale across the entire dataset. This spatial normalization is particularly crucial for neural networks to learn consistent geometric patterns, as it eliminates size variations that could otherwise dominate the learned features. For point cloud data, the implementation involves centering the cloud by subtracting its centroid, then scaling all points by the maximum Euclidean distance from the center, as shown in the provided Python snippet. This approach guarantees that all objects occupy approximately the same volume in the normalized space while preserving their original geometric relationships.

2.2.2 Watertight

The *IGL* library generates watertight surfaces by constructing a signed distance field (SDF) from defective geometry. We initialize a uniform 3D query grid encompassing the input mesh. For each query point $\mathbf{q} \in Q_g$, IGL computes:

$$\text{SDF}(\mathbf{q}) = \underbrace{\text{distance\_to\_mesh}(\mathbf{q}, V, F)}_{\text{nearest surface distance}} \cdot \underbrace{\text{sign}(\omega(\mathbf{q}))}_{\text{inside/outside sign}}$$

where V and F represent input vertices and faces. The sign is determined by the generalized winding number $\omega(\mathbf{q})$ where $\omega \approx 1$ indicates interior points and $\omega \approx 0$ exterior points.

Sign consistency is enforced using IGL's winding number calculation. This resolves ambiguous signs near self-intersections by thresholding $\omega > 0.5$ for interior classification. The watertight mesh is extracted at the zero-level isosurface via marching cubes. The output $(V_{\text{iso}}, F_{\text{iso}})$ forms a topologically closed surface without boundary discontinuities.

2.2.3 SDF Sampling

In our approach, the creation of signed distance fields (SDF) serves as the core mathematical framework for representing 3D shapes. To achieve this, we employ a strategy of randomly selecting query points in two distinct ways: either close to the surface of the shape or evenly distributed throughout the entire $[-1, 1]^3$ space. We then compute the SDF values for these points using the IGL computing library. The SDF values obtained from points near the surface are crucial for capturing the intricate details of the shape's surface. This allows the model to accurately represent fine features and subtle variations in the geometry. The SDF values from uniformly sampled points provide the model with a broader understanding of the overall structure and form of the 3D shapes. This dual sampling approach ensures that the model gains a comprehensive understanding of both detailed and general aspects of the shapes.

2.2.4 Surface Sampling

Our hybrid sampling strategy combines the strengths of both uniform and feature-aware approaches to capture complete geometric information. Uniform sampling guarantees even coverage across the surface, forming approximately 50% of the final point set. The remaining 50% of points are strategically placed near high-curvature features through importance sampling based on local surface derivatives. The sampling density automatically adapts to geometric complexity, increasing point concentration in regions with intricate details while maintaining sparser sampling in simpler areas. This balanced approach ensures that sharp edges, corners, and other defining features receive adequate representation without unnecessarily dense sampling of planar regions, optimizing both the quality and efficiency of the resulting point set.

2.2.5 Condition Render

To render condition images for shape generation training, we sample 150 cameras uniformly distributed on a sphere centered at the origin using the Hammersley sequence algorithm with a randomized offset $\delta \in [0, 1)^2$. An augmented dataset is generated with randomized FoVs $\theta_{\text{aug}} \sim \mathcal{U}(10^\circ, 70^\circ)$.

in the meanwhile camera's radius is adjusted between $r_{\text{aug}} \in [1.51, 9.94]$ to ensure consistent object framing.

Algorithm 1 3D Data Preprocessing Pipeline

```
Require: Raw 3D mesh $X = (V, F)$ (vertices and faces)
1. Normalization:
$V_{norm} \leftarrow Normalize(V)$
2. Watertight Processing:
Initialize empty SDF grid $\mathcal{G}$
$SDF \leftarrow \text{IGL}(\mathcal{G}, V_{norm}, F)$
$(V_{iso}, F_{iso}) \leftarrow \text{MarchingCube}(SDF, \text{level} = 0)$
3. SDF Sampling:
$P_{surface} \leftarrow \text{sample\_surface}(V_{iso}, F_{iso}, N_{near})$    ▷ $N_{near} = 249,856$ total points
$P_{near} \leftarrow \text{sample\_near\_surface}(V_{iso}, F_{iso}, N_{uniform})$    ▷ $N_{uniform} = 249,856$ total points
Query points $P_{query} \leftarrow P_{near} \cup P_{uniform}$
$SDF_{query} \leftarrow igl.signed\_distance(P_{query}, V_{iso}, F_{iso})$
4. Surface Sampling:
$P_{random} \leftarrow \text{RandomSample}(V_{iso}, F_{iso}, N)$
$P_{sharp} \leftarrow \text{SharpSample}(V_{iso}, F_{iso}, N)$    ▷ $N = 124928$ total points
5. Hammersley Condition Rendering:
Generate Hammersley sequence $H_{150}$ on unit sphere
Apply random offset $\delta \sim \mathcal{U}([0, 1)^2)$ to $H_{150}$
for each camera position $\mathbf{c}_i \in H_{150}$ do
    Sample FoV $\theta_i \sim \mathcal{U}(10°, 70°)$
    Compute radius $r_i \sim \mathcal{U}(\theta_{min}, \theta_{max})$
    $Img_i \leftarrow \text{render\_image}(X, \mathbf{c}_i, r_i)$
end for
return $P_{query}, SDF_{query}, P_{random}, P_{sharp}, \{Img_i\}_{i=1}^{150}$
```

2.3 Data preprocessing for texture synthesis

The texture synthesis heavily relies on 3D assets with rich texture details. Our training dataset consists of 70k+ human-annotated high quality data following strict curation protocols, which is filtered from Objaverse [31] and Objaverse-XL [32]. For each 3D object, we rendered data from four elevation angles: $-20°$, $0°$, $20°$, and a random angle. At each elevation angle, we select 24 views that are uniformly distributed across azimuth dimension, generating corresponding albedo, metallic, roughness maps, and HDR/Point-light images of 512×512 resolution. We probabilistically render reference images using: (1) Randomly sampled viewpoints (elevation: [-30°, 70°]) (2) Stochastic illumination: point lights (p=0.3) or HDR maps (p=0.7).

3 Training

3.1 Hunyuan3D-Shape

Shape generation serves as the cornerstone of 3D generation, playing a crucial role in determining the usability of a 3D asset. Drawing inspiration from the success of the latent diffusion model [2, 7, 8, 11] in shape generation, we have adopted the generative diffusion model as the architecture for our shape model. Our shape generation model is composed of two main components: (1) an autoencoder, Hunyuan3D-ShapeVAE (Sec. 3.1.1), which compresses the shape of a 3D asset, represented by a polygon mesh, into a sequence of continuous tokens within the latent space; and (2) a flow-based diffusion model, Hunyuan3D-DiT (Sec. 3.1.2), which is trained on the latent space of ShapeVAE to predict object token sequences from a user-provided image. These predicted tokens are then decoded back into a polygon mesh using the VAE decoder. The specifics of these models are detailed below.

3.1.1 Hunyuan3D-ShapeVAE

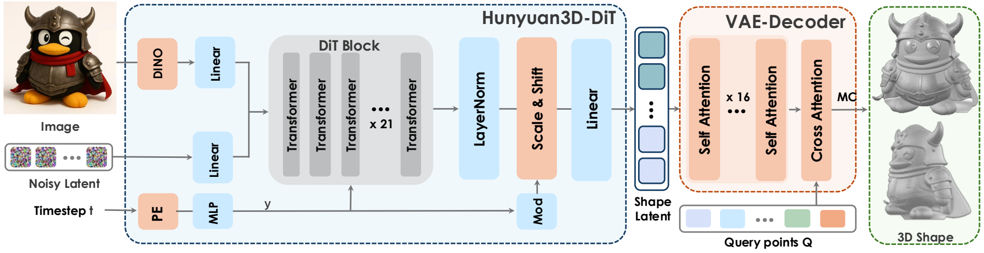

Figure 2: Overall pipeline for shape generation. Given a single image input, combining Hunyuan3D-DiT and Hunyuan3D-VAE can generate a high-quality and high-fidelity 3D shape.

Hunyuan3D-ShapeVAE utilizes vector sets introduced by 3DShape2VecSet [7], and also used in the recent project Dora [11, 33]. Following these works, we employ a variational encoder-decoder transformer for compact shape representations. We use 3D coordinates and the normal vectors from point clouds sampled from the surfaces of 3D shapes as inputs for the encoder. The decoder is designed to predict the Signed Distance Function (SDF) of the 3D shape, which can be further transformed into a triangle mesh using the marching cube algorithm.

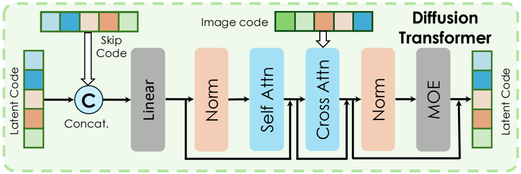

Figure 3: Overview of DiT block. We adopt the DiT implemented by Hunyuan-DiT [4] in our pipeline.

Encoder. For an input mesh, we first gather uniformly sampled surface point clouds $P_u \in \mathbb{R}^{M\times 3}$ and importance sampled point clouds $P_i \in \mathbb{R}^{N\times 3}$. The encoding process begins by applying Farthest Point Sampling (FPS) separately to P_u and P_i to generate query points $Q_u \in \mathbb{R}^{M'\times 3}$ and $Q_i \in \mathbb{R}^{N'\times 3}$ respectively. We then concatenate these points to form the final point cloud $P \in \mathbb{R}^{(M+N)\times 3}$ and query set $Q \in \mathbb{R}^{(M'+N')\times 3}$. Both P and Q are encoded by Fourier positional followed by linear projection, yielding encoded features $X_p \in \mathbb{R}^{(M+N)\times d}$ and $X_q \in \mathbb{R}^{(M'+N')\times d}$, where d is the dimension. These features are processed through cross-attention and self-attention layers to obtain the hidden shape representation $H_s \in \mathbb{R}^{(M'+N')\times d}$. Following the variational autoencoder framework, we apply final linear projections to 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 latent shape embedding, with d_0 being the latent dimension.

Decoder. The decoder $\mathcal{D}_s$ reconstructs a 3D neural field from the latent shape embedding Z_s. Initially, a projection layer maps the d_0-dimensional latent embedding to the transformer's hidden dimension d. Subsequent self-attention layers refine these embeddings, followed by a point perceiver module that queries a 3D grid $Q_g \in \mathbb{R}^{(H\times W\times D)\times 3}$ to generate a neural field $F_g \in \mathbb{R}^{(F_n\times W\times D)\times d}$. A final linear projection converts F_g into a Sign Distance Function (SDF) $F_{sdf} \in \mathbb{R}^{(F_o\times W\times D)\times 1}$, which is subsequently converted to a triangle mesh via marching cubes during inference.

Training Strategy & Implementation. We employ two 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. 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. To optimize computational efficiency, we implement a multi-resolution training strategy where latent token sequence lengths vary dynamically, with a maximum sequence length of 3072.

3.1.2 Hunyuan3D-DiT

Hunyuan3D-DiT is a flow-based diffusion model designed to generate detailed and high-resolution 3D shapes based on image conditions.

Condition encoder. To capture detailed image features, we employ a large image encoder, DINOv2 Giant [34] with an image size of 518×518. Additionally, we remove the background from the input image, resize the object to a standard size, center it, and fill the background with white.

DiT block. Inspired by Hunyuan-DiT [4] and TripoSG [9], we adopt transformers structure as shown in Fig. 2. We stack the 21 Transformer layers to learn the latent codes. As shown in Fig. 3, in each Transformer layer, we leverage the dimension concatenation to introduce the skip connection of the latent code. Similar to previous methods [11, 24], we employ the cross-attention layer to project the image condition into the latent code. In addition, an MOE layer is used to enhance the representation learning of the latent code.

Training & Inference. We train our model using the flow matching objective [15, 3]. Flow matching defines a probability path between Gaussian and data distributions, training the model to predict the velocity field $u_t = \frac{x_t}{d_t}$ that moves sample x_t towards data x_1. We use the affine path with a conditional optimal transport schedule as specified in [35], where $x_t = (1-t) \times x_0 + t \times x_1$, $u_t = x_1 - x_0$. 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 represents model condition. During inference, we randomly sample a starting point and use a first-order Euler ordinary differential equation (ODE) solver to compute x_1 with our diffusion model $u_\theta(x_t, c, t)$.

3.2 Hunyuan3D-Paint

Traditional color textures are no longer sufficient to meet the demands for photorealistic 3D asset generation. Therefore, we introduce a PBR material texture synthesis framework advancing beyond conventional RGB texture maps. We adhere to the BRDF model and simultaneously output albedo, roughness, and metallic maps from multiple viewpoints, aiming to accurately describe the surface reflectance properties of generated 3D assets and precisely simulate the distribution of geometric micro-surfaces, resulting in more realistic and detailed rendering effects. Further, we introduce 3D-Aware RoPE to inject spatial information, significantly improving cross-view consistency and enabling seamless texturing.

Basic Architecture. Building on the multiview texture generation architecture of Hunyuan3D-2 [36], we introduce a novel material generation framework, as is shown in the left side of Fig.4. The framework implements the Disney Principled BRDF model [37] to generate high-quality PBR material maps. We retain the reference image feature injection mechanism of ReferenceNet, while concatenating both geometry-rendered normal maps and CCM (canonical coordinate map) with latent noise.

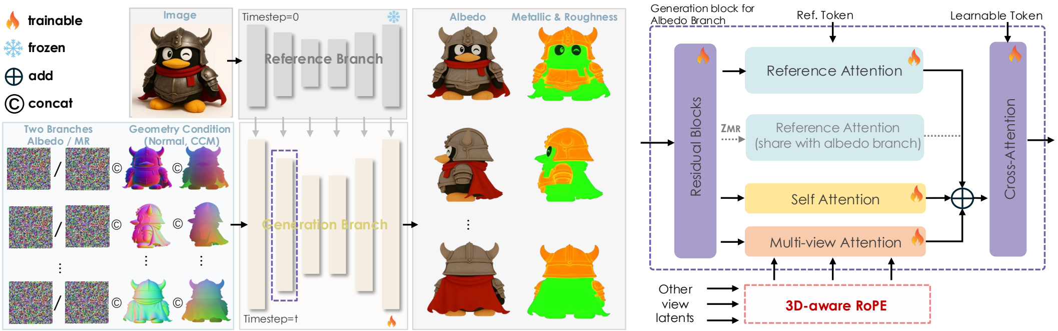

Figure 4: Overview of material generation framework.

Spatial-Aligned Multi-Attention Module. We employ a pre-trained VAE for multi-channel material image compression while implementing a parallel dual-branch UNet architecture [38] for material generation. As shown in the right side of Fig.4, we implement parallel multi-attention modules [39] comprising self-attention, multi-view attention, and reference attention for both albedo and metallic-roughness (MR) maps. To model the physical relationships between albedo/MR maps and reference images, and to achieve spatial alignment between MR and albedo maps, we directly propagate the computed outputs from the albedo reference attention module to the MR branch.

Models	ULIP-T (↑)	ULIP-I (↑)	Uni3D-T (↑)	Uni3D-I (↑)
Michelangelo [8]	0.0752	0.1152	0.2133	0.2611
Craftsman 1.5 [24]	0.0745	0.1296	0.2375	0.2987
TripoSG [9]	0.0767	0.1225	0.2506	0.3129
Step1X-3D [26]	0.0735	0.1183	0.2554	0.3195
Trellis [25]	0.0769	0.1267	0.2496	0.3116
Direct3D-S2 [27]	0.0706	0.1134	0.2346	0.2930
Hunyuan3D-DiT	**0.0774**	**0.1395**	**0.2556**	**0.3213**

Table 1: The quantitative comparison for shape generation. The Hunyuan3D-DiT presents the best performance.

3D-Aware RoPE. To address texture seams and ghosting artifacts caused by local inconsistencies across neibor views, 3D-Aware RoPE [39] is introduced into the multiview attention block for enhanced cross-view coherence. Specifically, by downsampling the 3D coordinate volume, we construct multi-resolution 3D coordinate encodings aligned with the UNet hierarchy levels. These encodings are additively fused with corresponding hidden states, thereby integrating cross-view interactions into 3D space to enforce multi-view consistency.

Illumination-Invariant Training Strategy. To generate light- and shadow-free albedo map and accurate MR map, we posit an intuitive insight: while rendering results of the same object differ under diverse lighting, its intrinsic material properties should remain consistent. Consequently, we design an illumination-invariant training strategy [38] to enforce this property. Specifically, consistency loss is computed by adopting two sets of training samples containing reference images of the same object rendered by different lighting conditions.

Experimental Setup. Our model is initialized from the Zero-SNR checkpoint [40] of Stable Diffusion 2.1 and optimized using the AdamW at a learning rate of 5×10^{-5}. The training protocol incorporates 2000 warm-up steps, requiring approximately 180 GPU-days.

4 Evaluation

To assess the effectiveness of a 3D generative model, we conduct experiments focusing on three key areas: (1) 3D Shape Generation (untextured shape evaluation), (2) Texture Synthesis, and (3) Complete 3D asset creation (textured 3D shapes).

4.1 3D Shape Generation

Shape generation is crucial for 3D generation, as detailed and high-resolution meshes provide the groundwork for subsequent tasks. In this section, we evaluate the 3D shape generation capabilities of Hunyuan3D-DiT, focusing on shape creation.

Metrics. To evaluate shape generation performance, we used ULIP [41] and Uni3D [42] to measure the similarity between the generated mesh and input images. Specifically, we sampled 8,192 surface points from the generated mesh as the point cloud modality. We then utilized the caption of the input image obtained from an existing VLM model as the text modality. Finally, we applied the ULIP models to obtain the ULIP-I and ULIP-T scores, which measure the similarity between the point cloud and text, as well as the similarity between the point cloud and image, respectively. In this context, the text caption comes from a VLM model. We also employed the same process to obtain the Uni3D-I and Uni3D-T scores based on the Uni3D model.

Comparison with Shape Generation Models. We compared Shape Quality with several leading models, including open-source options like Direct3D-S2 [27], Step1X-3D [26], and TripoSG [9]. Table 1 presents a numerical comparison between Hunyuan3D-DiT and other methods, showing that Hunyuan3D-DiT delivers the most accurate results. Additionally, the visual comparison in Fig. 5 confirms the adherence of Hunyuan3D-DiT to image prompts, including the faithful capture of intricate details (details of roly-poly toys, the number of calculator buttons, the number of teeth on a rake, and the structure of a fighter jet), and its ability to produce watertight meshes ready for downstream applications.

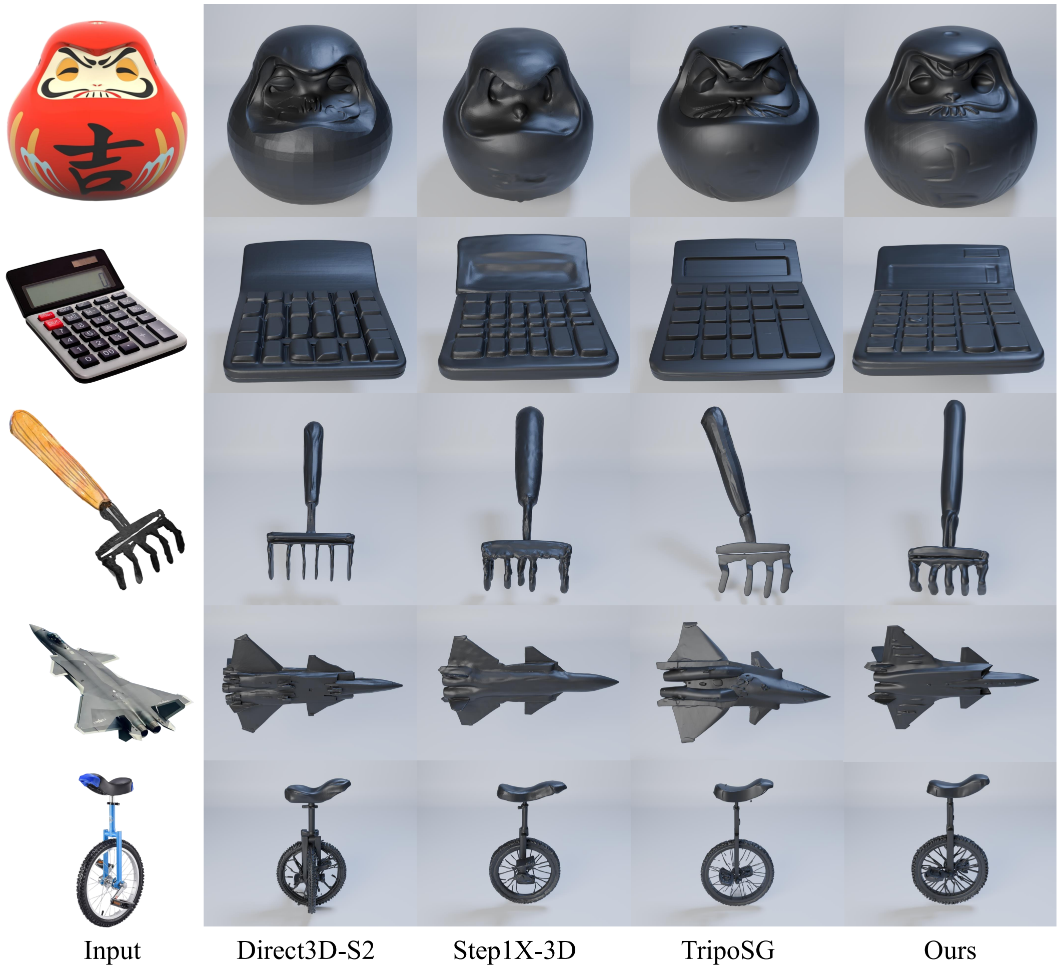

Figure 5: The qualitative comparisons for image-to-shape generation.

4.2 Texture Map Synthesis

As texture maps directly influence the visual appeal of textured 3D assets, we conduct comprehensive quantitative and qualitative comparisons of texture generation methodologies across both academic and industrial domains.

Comparison with Texture Synthesis Models. To quantify the similarity between generated textures and ground truth, we employ Fréchet Inception Distance (FID) [43], CLIP-based FID (CLIP-FID) [44], and Learned Perceptual Image Patch Similarity (LPIPS) [45] metrics on Hunyuan3D-Paint and baseline image-to-texture models, including SyncMVD-IPA [13], TexGen [46] and Hunyuan3D-2.0 [36]. Given an untextured shape and a single image, we compare these models with our results through both quantitative and qualitative evaluations. The quantitative results are shown in Tab. 2, while the qualitative results are illustrated in Figure 6. These evaluations clearly demonstrate the superiority of our method over all comparative approaches.

Comparison with Image-to-3D Models. We also conduct visualized comparison with publicly accessible 3D generation algorithms, including the open-source Step1X-3D [26] and 3DTopia-XL [47], alongside the commercial Model 1 and Model 2. Given a single image, all compared methods can form geometry and corresponding PBR material maps. Specifically, we assess the end-to-end quality across these methods, as shown in Fig. 7. These results demonstrate that our model not only generates PBR material maps with the highest fidelity but also effectively mitigates shortcomings associated with lower-quality geometries. This leads to superior end-to-end performance compared to existing methods.

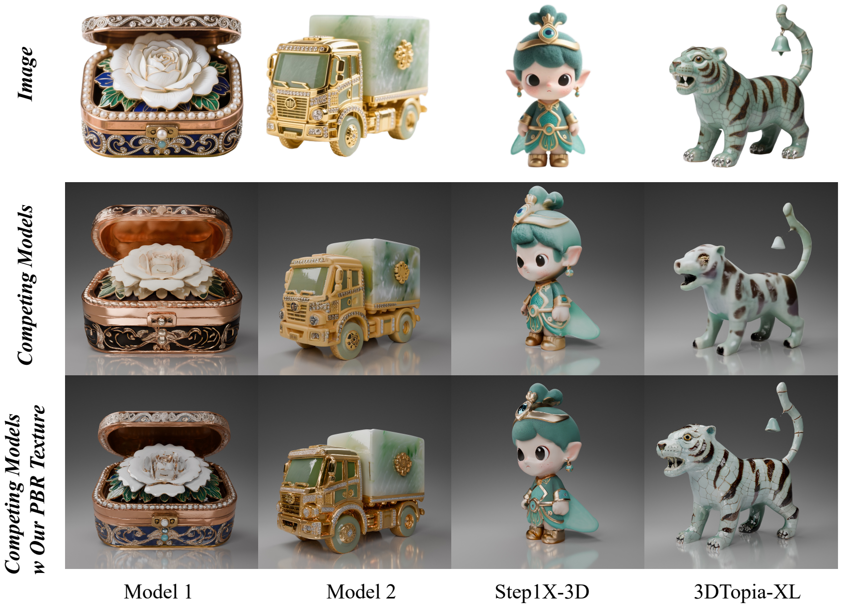

Figure 6: The qualitative comparisons for texture synthesis.

Method	CLIP-FID (↓)	CMMD (↓)	CLIP-I (↑)	LPIPS (↓)
SyncMVD-IPA [13]	28.39	2.397	0.8823	0.1423
TexGen [46]	28.24	2.448	0.8818	0.1331
Hunyuan3D-2.0 [36]	26.44	2.318	0.8893	0.1261
Hunyuan3D-Paint	**24.78**	**2.191**	**0.9207**	**0.1211**

Table 2: The quantitative comparison for texture generation. Hunyuan3D-Paint achieves the best performance.

5 Conclusion

Hunyuan3D 2.1 introduces a groundbreaking approach for production-ready 3D content creation by unifying high-fidelity geometry generation and PBR material synthesis within an open-source framework. Its architecture, which combines a DiT for shape generation and a multi-view conditioned painter for PBR material synthesis, allows for the rapid creation of studio-quality assets with exceptional visual fidelity. By open-sourcing the entire data processing, training pipelines, and model weights, this system makes advanced 3D AIGC accessible to a wider audience, revolutionizing workflows in gaming, virtual reality, and industrial design. Quantitative metrics demonstrate its superiority in both geometric accuracy and material quality. As the first fully open-sourced solution for PBR-textured 3D asset generation, *Hunyuan3D 2.1* bridges the gap between academic research and scalable content creation, encouraging global collaboration to shape the future of 3D generative AI.

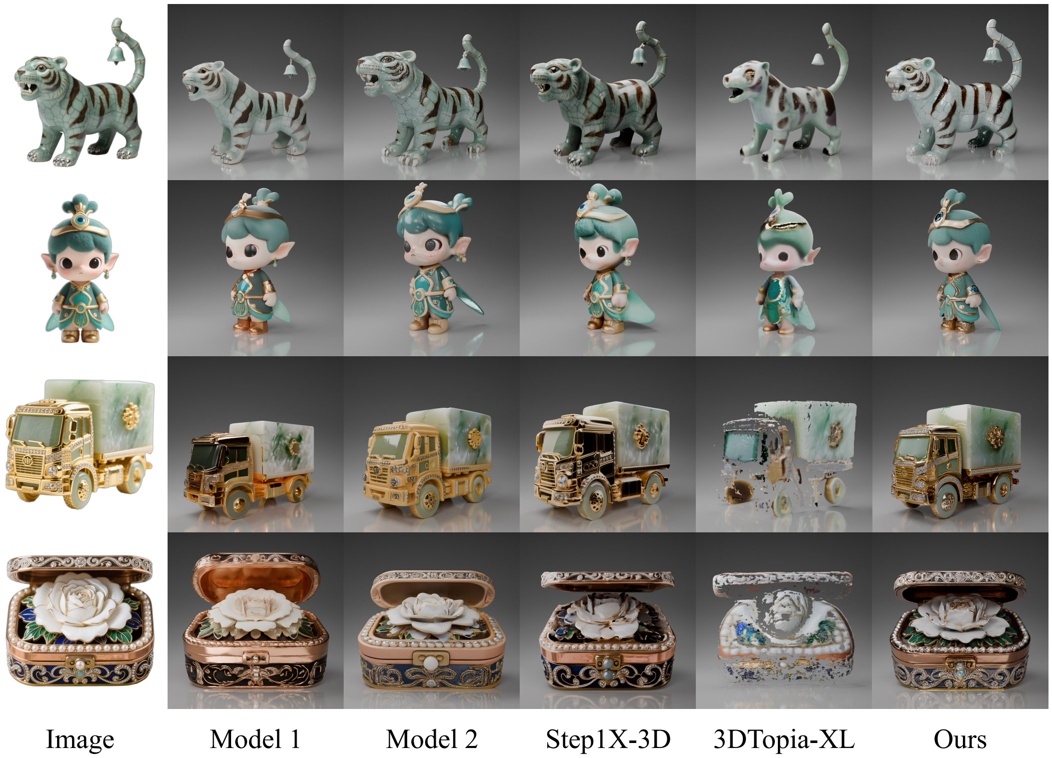

Figure 7: The qualitative comparisons for image-to-3D generation.

6 Contributors

- **Project Sponsors:** Jie Jiang, Linus, Yuhong Liu, Di Wang, Tian Liu, Peng Chen
- **Project Leaders:** Chunchao Guo, Jingwei Huang
- **Core Contributors:**
  - **Data:** Lifu Wang, Sicong Liu, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Dongyuan Guo, Junlin Yu, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Shida Wei, Chao Zhang, Yonghao Tan
  - **Shape Generation:** Haolin Liu, Yunfei Zhao, Qingxiang Lin, Zeqiang Lai, Xianghui Yang, Huiwen Shi, Zibo Zhao, Bowen Zhang, Hongyu Yan
  - **Texture Synthesis:** Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo
  - **Infra:** Yifu Sun, Lin Niu, Shirui Huang, Bojian Zheng, Shu Liu, Shilin Chen, Xiang Yuan, Xiaofeng Yang, Kai Liu, Jianchen Zhu

References

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

[2] 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.

[3] 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.

[4] 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.

[5] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2024.

[6] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[7] 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.

[8] 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*, 36:73969–73982, 2023.

[9] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

[16] 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. *arXiv preprint arXiv:2311.04400*, 2023.

[17] 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.

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

[19] Xianghui Yang, Yan Zuo, Sameera Ramasinghe, Loris Bazzani, Gil Avraham, and Anton van den Hengel. Viewfusion: Towards multi-view consistency via interpolated denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9870–9880, 2024.

[20] Haohan Weng, Tianyu Yang, Jianan Wang, Yu Li, Tong Zhang, CL Chen, and Lei Zhang. Consistent123: Improve consistency for one image to 3d object synthesis. *arXiv preprint arXiv:2310.08092*, 2023.

[21] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023.

[22] 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.

[23] 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.

[24] Weiyu Li, Jiarui Liu, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024.

[25] 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.

[26] Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025.

[27] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Philip Torr, Xun Cao, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412*, 2025.

[28] 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.

[29] 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.

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

[31] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023.

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

[33] 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.

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

[35] 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.

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

[37] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM Siggraph*, volume 2012, pages 1–7. vol. 2012, 2012.

[38] Zebin He, Mingxin Yang, Shuhui Yang, Yixuan Tang, Tao Wang, Kaihao Zhang, Guanying Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, and Wenhan Luo. Materialmvp: Illumination-invariant material generation via multi-view pbr diffusion, 2025.

[39] Yifei Feng, Mingxin Yang, Shuhui Yang, Sheng Zhang, Jiaao Yu, Zibo Zhao, Yuhong Liu, Jie Jiang, and Chunchao Guo. Romantex: Decoupling 3d-aware rotary positional embedded multi-attention network for texture synthesis, 2025.

[40] 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.

[41] 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.

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

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

[44] 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.

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[46] Xin Yu, Ze Yuan, Yuan-Chen Guo, Ying-Tian Liu, Jianhui Liu, Yangguang Li, Yan-Pei Cao, Ding Liang, and Xiaojuan Qi. Texgen: a generative diffusion model for mesh textures. *ACM Transactions on Graphics*, 43(6):1–14, 2024.

[47] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. *arXiv preprint arXiv:2409.12957*, 2024.