



CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets

LONGWEN ZHANG*, ShanghaiTech University, China and Deemos Technology Co., Ltd., China

ZIYU WANG*, ShanghaiTech University, China and Deemos Technology Co., Ltd., China

QIXUAN ZHANG†, ShanghaiTech University and Deemos Technology Co., Ltd., China

QIWEI QIU, ShanghaiTech University, China and Deemos Technology Co., Ltd., China

ANQI PANG, ShanghaiTech University, China

HAORAN JIANG, ShanghaiTech University, China and Deemos Technology Co., Ltd., China

WEI YANG, Huazhong University of Science and Technology, China

LAN XU‡, ShanghaiTech University, China

JINGYI YU‡, ShanghaiTech University, China

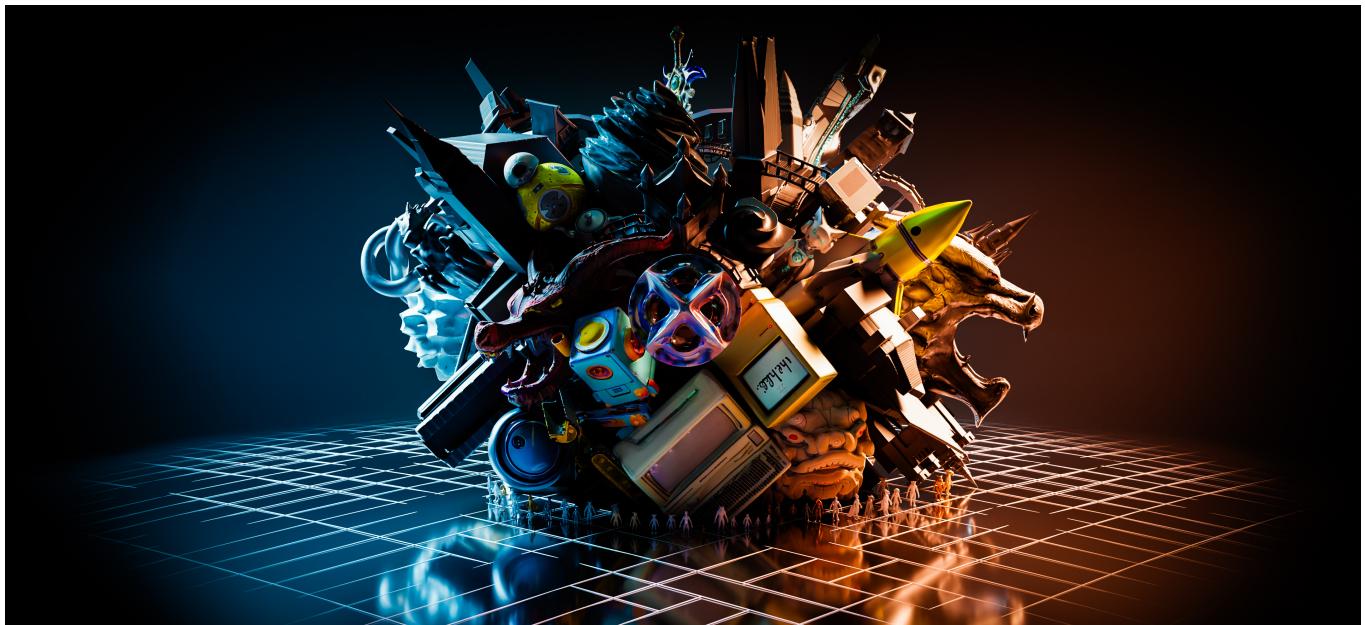


Fig. 1. A collection of 3D assets generated by CLAY, showcasing its capabilities in turning human imaginations into their digital forms.

*Equal contributions.

†Project leader.

‡Corresponding author.

Authors' Contact Information: Longwen Zhang, ShanghaiTech University, Shanghai, China and Deemos Technology Co., Ltd., Shanghai, China, zhanglw2@shanghaitech.edu.cn; Ziyu Wang, ShanghaiTech University, Shanghai, China and Deemos Technology Co., Ltd., Shanghai, China, wangzyb@shanghaitech.edu.cn; Qixuan Zhang, ShanghaiTech University and Deemos Technology Co., Ltd., Shanghai, China, zhangqx1@shanghaitech.edu.cn; Qiwei Qiu, ShanghaiTech University, Shanghai, China and Deemos Technology Co., Ltd., Shanghai, China, qiuqw@shanghaitech.edu.cn; Anqi Pang, ShanghaiTech University, Shanghai, China, pangaq@shanghaitech.edu.cn; Haoran Jiang, ShanghaiTech University, Shanghai, China and Deemos Technology Co., Ltd., Shanghai, China, jianghr1@shanghaitech.edu.cn; Wei Yang, Huazhong University of Science and Technology, Wuhan, China, weiyangs@hust.edu.cn; Lan Xu, ShanghaiTech University, Shanghai, China, xulan1@shanghaitech.edu.cn; Jingyi Yu, ShanghaiTech University, Shanghai, China, yujingyi@shanghaitech.edu.cn.

In the realm of digital creativity, our potential to craft intricate 3D worlds from imagination is often hampered by the limitations of existing digital tools, which demand extensive expertise and efforts. To narrow this disparity, we introduce CLAY, a 3D geometry and material generator designed to effortlessly transform human imagination into intricate 3D digital structures. CLAY supports classic text or image inputs as well as 3D-aware controls from diverse primitives (multi-view images, voxels, bounding boxes, point clouds, implicit representations, etc). At its core is a large-scale generative model composed of a multi-resolution Variational Autoencoder (VAE) and a minimalistic latent Diffusion Transformer (DiT), to extract rich 3D priors directly from a diverse range of 3D geometries. Specifically, it adopts neural fields to represent continuous and complete surfaces and uses a geometry generative module with pure transformer blocks in latent space. We present



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

© 2024 Copyright held by the owner/author(s).
ACM 1557-7368/2024/7-ART120
<https://doi.org/10.1145/3658146>

ACM Trans. Graph., Vol. 43, No. 4, Article 120. Publication date: July 2024.

a progressive training scheme to train CLAY on an ultra large 3D model dataset obtained through a carefully designed processing pipeline, resulting in a 3D native geometry generator with 1.5 billion parameters. For appearance generation, CLAY sets out to produce physically-based rendering (PBR) textures by employing a multi-view material diffusion model that can generate 2K resolution textures with diffuse, roughness, and metallic modalities. We demonstrate using CLAY for a range of controllable 3D asset creations, from sketchy conceptual designs to production ready assets with intricate details. Even first time users can easily use CLAY to bring their vivid 3D imaginations to life, unleashing unlimited creativity.

CCS Concepts: • Computing methodologies → Artificial intelligence.

Additional Key Words and Phrases: 3D Asset Generation, Multi-modal Control, Physically-based Rendering, Diffusion Transformer, Large-scale Model

ACM Reference Format:

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024. CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets. *ACM Trans. Graph.* 43, 4, Article 120 (July 2024), 20 pages. <https://doi.org/10.1145/3658146>

1 INTRODUCTION

Three-dimensional (3D) imagination allows us humans to visualize and design structures, spaces, and systems before they are physically constructed. When we were kids, we learned to build objects using this imagination, with as simple as clay, stones, or wood sticks, and for the lucky few, LEGO blocks. To us then, a building formed by a few simple blocks can imaginatively transform to a magnificent castle and a wood stick attached to a stone into a LightSaber, Jedi's or Sith's. In fact, with a diverse range of pieces in different shapes, sizes, and colors in hand, we once imagined having virtually unlimited capabilities for creating objects. This boundless imagination has fundamentally transformed the entertainment industry, from feature films to computer games, and has led to significant advances in the field of computer graphics, from modeling to rendering. In contrast, the capabilities of producing creative content by far fall far behind our imagination. For example, the current 3D creation workflow still requires immense artistic expertise and tedious manual labor. An ideal 3D creation tool should conveniently convert our kid-like vibrant imagination into digital reality - it should effortlessly craft geometry and textures and support diverse controllable strategies for creation, translating abstract concepts into tangible, digital forms.

Latest progresses on AI Generated Content (AIGC) [Po et al. 2023] reignite the hope and enthusiasm to bridge imagination and creation, epitomized by the text-based 2D image generation that benefits from the consolidation of large image datasets, effective neural network architectures (e.g., Transformer [Vaswani et al. 2017], Diffusion Model [Ho et al. 2020]), adaptation schemes (e.g., LoRA [Hu et al. 2022], ControlNet [Zhang et al. 2023b]), etc. It is not an exaggeration that the 2D creation workflow has largely been revolutionized, perhaps symbolized by the controversial triumph of Midjourney's AI-generated "Théâtre D'opéra Spatial" at a digital arts competition. In a similar vein, we have also witnessed rapid progress in 3D asset generation. Yet compared with 2D generation, 3D generation has not yet reached the same level of progress that can fundamentally reshape the 3D creation pipeline. Its model scalability and adaptation capabilities fall far behind mature 2D techniques. The challenges are

multi-fold, stemming from the limited scale of quality 3D datasets as well as the inherent entanglement of geometry and appearance of 3D assets.

State-of-the-art 3D asset generation techniques largely build on two distinct strategies: either lifting 2D generation into 3D or embracing 3D native strategies. In a nutshell, the former line of work leverages 2D generative models [Rombach et al. 2022; Saharia et al. 2022] via intricate optimization techniques such as score distillations [Poole et al. 2023; Wang et al. 2023], or further refines 2D models for multi-view generation [Liu et al. 2023c; Shi et al. 2024]. They address the diverse appearance generation problem by employing pretrained 2D generative models. As 2D priors do not easily translate to coherent 3D ones, methods based on 2D generation generally lack concise 3D controls (preserving lines, angles, planes, etc) that one would expect in a foundational model and they consequently fail to maintain high geometric fidelity. In comparison, 3D native approaches attempt to train generative models directly from 3D datasets [Chang et al. 2015; Deitke et al. 2023] where 3D shapes can be represented in explicit forms such as point clouds [Nichol et al. 2022], meshes [Nash et al. 2020; Siddiqui et al. 2024] or implicit forms such as neural fields [Chen and Zhang 2019; Zhang et al. 2023c]. They can better "understand" and hence preserve geometric features, but have limited generation ability unless they employ much larger models, as shown in concurrent works [Ren et al. 2024; Yariv et al. 2024]. Yet larger models subsequently require training on larger datasets, which are expensive to obtain, the problem that 3D generation aims to address in the first place.

In this paper, we aim to bring together the best of 2D-based and 3D-based generations by following the "pretrain-then-adaptation" paradigm adopted in text/image generation, effectively mitigating 3D data scarcity issue. We present CLAY, a novel Controllable and Large-scale generative scheme to create 3D Assets with high-quality geometry and appearance. CLAY manages to scale up the foundation model for 3D native geometry generation at an unprecedented quality and variety, and at the same time it can generate appearance with rich multi-view physically-based textures. The 3D assets generated by CLAY contain not only geometric meshes but also material properties (diffuse, roughness, metallic, etc.), directly deployable to existing 3D asset production pipelines. As a versatile foundation model, CLAY also supports a rich class of controllable adaptations and creations (from text prompts to 2D images, and to diverse 3D primitives), to help conveniently convert a user's imagination to creation.

The core of CLAY is a large-scale generative model that extracts rich 3D priors directly from a diverse range of 3D geometries. Specifically, we adopt the neural field design from 3DShape2VecSet [Zhang et al. 2023c] to depict continuous and complete surfaces along with a tailored multi-resolution geometry Variational Autoencoder (VAE). We customize the geometry generative module in latent space with an adaptive latent size. To conveniently scale up the model, we adopt a minimalist latent diffusion transformer (DiT) with pure transformer blocks to accommodate the adaptive latent size. We further propose a progressive training scheme to carefully increase both the latent size and model parameters, resulting in a 3D native geometry generator with 1.5 billion parameters. The quality of training samples is crucial for fine-grained geometry generation,

especially considering the limited size of available 3D datasets. We hence present a new data processing pipeline to standardize the diverse 3D data and enhance the data quality. Specifically, it includes a remeshing process that converts various 3D surfaces into occupancy fields, preserving essential geometric features such as sharp edges and flat surfaces. At the same time, we harness the capabilities of GPT-4V [OpenAI 2023] to produce robust annotations that accentuate these geometric characteristics.

The combination of new architecture, training scheme, and training data in CLAY leads to a novel 3D native generative model that can create high-quality geometry, serving as the foundation to downstream model adaptations. For appearance generation, the scarcity of abundant data poses a significant challenge for synthesizing material texture maps. To tackle this issue, CLAY sets out to generate multi-view physically-based rendering (PBR) textures, and subsequently project them onto geometry. We construct a multi-view material diffusion model analogous to 2D diffusion model [Rombach et al. 2022] but trained on high-quality PBR textures from Objaverse [Deitke et al. 2023], to efficiently generate diffuse, roughness, and metallic modalities while avoiding tedious distillation. We further extend the diffusion model to support super-resolution as well as to accurately map the multi-view textures onto the generated geometry. The modified model allows for much faster high-quality textures generation than traditional optimization methods, producing 2K resolution in the UV space for realistic rendering.

We further explore various adaptation schemes including LoRA-like fine-tuning and cross-attention-based conditioning, to support classic text or image-based creations as well as 3D-aware controls from diverse primitives (multi-view images, voxels, bounding boxes, point clouds, implicit representations, etc). These extensive adaptation capabilities of CLAY hence enable controllable 3D asset creation ranging from sketchy conceptual designs to more sophisticated ones with intricate details. Even first time users can use CLAY to bring their vivid 3D imaginations to life with our tailored interactive controls: a bustling village can be generated from scattered bounding boxes across a barren landscape, a spacecraft with futuristic wings and propulsion system from craft blocks with textual descriptions, and ultimately creations from imaginations.

2 RELATED WORK

3D generation is undoubtedly the fastest-growing research arena in AIGC. Efficient and high quality 3D asset creation via generation benefits entertainment and gaming industry as well as film and animation productions. Previous practices have explored different routes, ranging from directly training on 3D datasets, to imposing generated 2D images as priors, and to imposing 3D priors on top of 2D generation.

Imposing 2D Images as Prior. 3D generation methods in this category attempt to exploit significant strides made in 2D image generation, exemplified by latest advances such as DALL-E [Ramesh et al. 2021], Imagen [Saharia et al. 2022] and Stable Diffusion [Rombach et al. 2022]. Extending this prowess to 3D generation, many approaches have adopted image-based techniques, focusing on transforming 2D images into 3D structures or imposing 2D images as priors. DreamFusion [Poole et al. 2023] pioneered this practice by

introducing Score Distillation Sampling (SDS) and employed 2D image generation with viewpoint prompts to produce 3D shapes via NeRF [Mildenhall et al. 2021] optimization. Although the idea is intriguing, earlier attempts struggled to consistently produce high-quality and diverse results. Often, generating satisfactory results requires repeated adjustments to parameters and long waits of optimizations. Subsequent enhancements in SDS have explored the possibility of extending the idea to various neural fields [Chen et al. 2024; Huang et al. 2024; Lin et al. 2023; Wu et al. 2024; Yu et al. 2023b; Zhu et al. 2024], ranging from DMTet [Shen et al. 2021] to the most recent 3D Gaussian splatting [Kerbl et al. 2023]. Various modifications managed to elevate the performance [Chen et al. 2023a; Li et al. 2024; Metzer et al. 2023; Seo et al. 2024; Wang et al. 2023; Zhang et al. 2023a]. Yet a critical challenge remains: 2D image diffusion models utilized in SDS still lack an explicit understanding of neither geometry nor viewpoint. The lack of perspective information and explicit 3D supervision can lead to the multi-head Janus problem, where realistic 3D renderings do not translate to view consistency and every rendered view can be deemed as the front view.

To mitigate the problem, Zero-1-to-3 [Liu et al. 2023c] proposes to integrate view information into the image generation process. This can be achieved by training an additional mapping from the transformation matrix to the pretrained Stable Diffusion model, enabling the network to obtain some prior knowledge on view position and distribution. Alternative solutions attempt to employ SDS to optimize a coherent neural field [Qian et al. 2024; Sun et al. 2024; Tang et al. 2024; Zhang et al. 2023d], but they generally require long optimization time. Latest developments [Blattmann et al. 2023; Li et al. 2023; Liu et al. 2024a; Long et al. 2024; Qiu et al. 2024; Shi et al. 2023, 2024] have focused on directly generating multi-view images with view consistency, by employing enhanced attention mechanisms. These approaches have significantly improved multi-view image generation, achieving a higher level of consistency.

The downside there is the need to fine-tune Stable Diffusion using additional images either by conducting multi-view rendering [Deitke et al. 2023] or using auxiliary multi-view datasets [Reizenstein et al. 2021; Wu et al. 2023; Yu et al. 2023a]. Since the multi-view results can already be used to extract 3D shapes (e.g., via multi-view stereo or neural methods), techniques such as SyncDreamer [Liu et al. 2024a] and Wonder3D [Long et al. 2024] employed NeuS [Wang et al. 2021a] to accelerate generation. One-2-3-45 [Liu et al. 2023d] has gone one step further to train generalizable NeuS [Long et al. 2022] on 3D datasets, to tackle sparse view inputs. Since the starting point of all these approaches are 2D images, they unanimously focus on the quality of generated images without attempting to preserve geometric fidelity. As a result, the generated geometry often suffers from incompleteness and lacks details.

Imposing 3D Geometry as Priors. To address challenges in 2D-based techniques, an emerging class of solutions attempt to impose 3D shapes as priors. Even though One-2-3-45 [Liu et al. 2023d] is viewed as using 2D image priors, the clever use NeuS as geometry proxy reveals the possibility of imposing 3D shape priors. For example, Instant3D [Li et al. 2023], LRM [Hong et al. 2024; Wang et al. 2024], DMV3D [Xu et al. 2024] and TGS [Zou et al. 2024] further

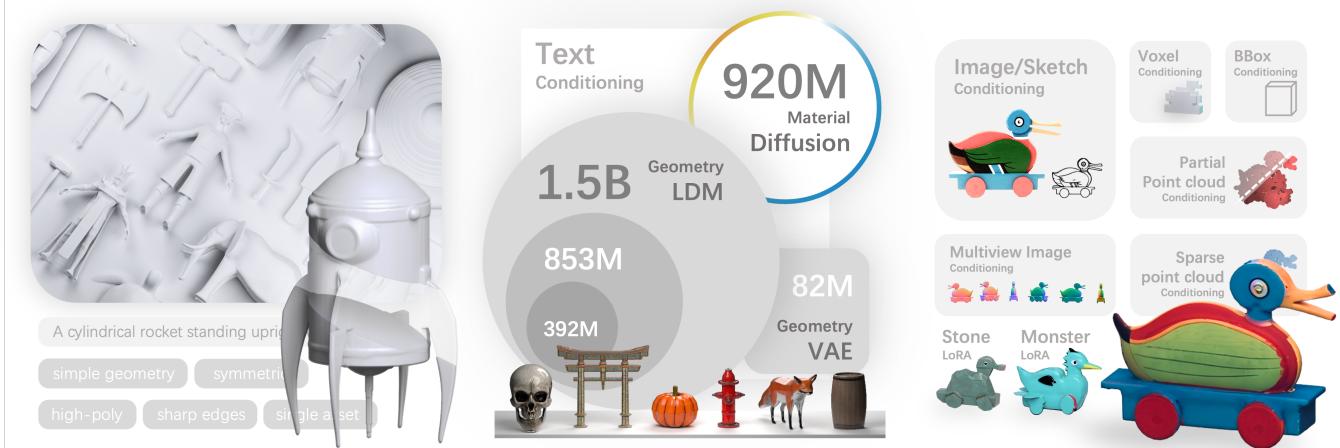


Fig. 2. An overview of our CLAY framework for 3D generation. Central to the framework is a large generative model trained on extensive 3D data, capable of transforming textual descriptions into detailed 3D geometries. The model is further enhanced by physically-based material generation and versatile modal adaptation, to enable the creation of 3D assets from diverse concepts and ensure their realistic rendering in digital environments.

utilized sparse-view or single-view reconstructors that leverage a Vision Transformer (ViT) as the vision backbone, coupled with a deep transformer architecture to directly reconstruct NeRF with both color and density attributes. They are hence commonly referred to Large Reconstruction Models (LRMs). Yet these techniques still focus on minimizing the volume rendering loss rather than explicitly generating surfaces, resulting in coarse or noisy geometry.

Apparently, the most straightforward practice to generate 3D would be to train on 3D datasets, rather than 2D images or image-induced 3D shapes. Early approaches [Choy et al. 2016; Fan et al. 2017; Groueix et al. 2018; Mescheder et al. 2019; Tang et al. 2019, 2021a] primarily utilized 3D convolutional networks to understand the 3D grid structure. Point-E [Nichol et al. 2022] took a pioneering step by leveraging a pure transformer-based diffusion model for denoising directly on the point clouds. This method is notable for its simplicity and efficiency, yet it faces great difficulties in transforming the generated point clouds into precise, common mesh surfaces. Polygon [Nash et al. 2020] and MeshGPT [Siddiqui et al. 2024] take a different approach by natively representing meshes through points and surface sequences. These models are capable of producing extremely high-quality meshes, but their dependence on small, high-quality datasets restricts their broader applicability. XCube [Ren et al. 2024] introduces a strategy that simplifies geometry into multi-resolution voxels before diffusion. It streamlines the process but faces challenges in managing complex prompts and supporting a broad range of downstream tasks, limiting its overall flexibility. It is worth mentioning that different 3D generation techniques have relied on different datasets. This is not surprising as they are based on different geometric representation but problematic as it is essential to have a unified dataset that includes all available shapes.

One such attempt is to represent geometry uniformly in terms of Signed Distance Field (SDF) [Park et al. 2019; Yariv et al. 2024], occupancy fields [Peng et al. 2020; Tang et al. 2021b], or both [Liu et al. 2024b; Zheng et al. 2023], and train directly on 3D datasets. Such approaches provide a more explicit mechanism than NeRF for learning and extracting surfaces but require the latent encoding of

watertight meshes for generation. Models such as DeepSDF [Park et al. 2019] and Mosaic-SDF [Yariv et al. 2024] utilize optimization techniques to create unique representations for each geometry in the training dataset, which is not efficient during training as they do not benefit from autoencoders. Other models such as SDFusion [Cheng et al. 2023] and ShapeGPT [Yin et al. 2023] adopt an intuitive 3D VAE (Variational Autoencoder) for encoding geometries and reconstructing SDF fields. These methods, primarily trained or tested on the ShapeNet [Chang et al. 2015] dataset, are limited in the diversity and variety of shapes they can generate. 3DGen [Gupta et al. 2023] employs a triplane VAE for both encoding and decoding SDF fields whereas Shap-E [Jun and Nichol 2023], 3DShape2VecSet [Zhang et al. 2023c], and Michelangelo [Zhao et al. 2023] adopt a different trajectory by utilizing transformers to encode the input point clouds into parameters for the decoding networks, signifying a shift towards more sophisticated neural network architectures in 3D generative models.

By far methods that aim to direct learning from 3D datasets, while capable of producing better geometries than 2D-based generation, still cannot match the hand-crafted ones by artists, in either detail or complexity. We observe, through the development of CLAY, this is mainly because they have not sufficiently explored rich geometric features embedded in the datasets. In addition, their small model size limits the capability of generalization and diversification. In CLAY, we resort to tailored geometry processing to mine a variety groups of 3D datasets as well as discuss effective techniques to scale up the generation model.

3 LARGE-SCALE 3D GENERATIVE MODEL

An effective 3D generative model should be able to generate 3D contents from different conditional inputs such as text, images, point clouds, and voxels. As aforementioned, the task is challenging in how to define a 3D model: should 3D asset be viewed in terms of geometry with per-vertex color or geometry with a texture map? should the 3D geometry be inferred from the generated appearance data or be directly generated? In CLAY, we adopt a minimalist

approach, i.e., we separate the geometry and texture generation processes. This indicates that we choose not to use 2D generation techniques which potentially help 3D geometry generation (e.g., through reconstruction). In our experiment, we find that once we manage to scale up the 3D generation model and train it with sufficiently large amount of high quality data, the directly generated 3D geometry by CLAY exceeds previous 2D generation based/assisted techniques by a large margin, in both diversity and quality (e.g., geometric details).

In a nutshell, CLAY is a large 3D generative model with 1.5 billion parameters, pretrained on high-quality 3D data. The significant upscaling from prior art is key to improving its capabilities in generation diversity and quality. Architecture-wise, CLAY extends the generative model in 3DShape2VecSet [Zhang et al. 2023c] with a new multi-resolution Variational Autoencoder (VAE). This extension enables more efficient geometric data encoding and decoding. In addition, we complement CLAY with an advanced latent Diffusion Transformer (DiT) for probabilistic geometry generation. Dataset-wise, we have developed a remeshing pipeline, along with annotation schemes powered by GPT-4V [OpenAI 2023], to standardize and unify existing 3D datasets. These datasets historically have not been used together for training a 3D generation model as they are in different formats and lack consistencies. Our combined dataset after processing maintains a consistent representation and coherent annotations. We show that putting the model architecture and training dataset together greatly improves 3D generation.

3.1 Representation and Model Architecture

Our approach for a 3D generative model emphasizes on learning to denoise 3D data in a compressed latent space, analogous to the foundation 2D generative models. This strategy significantly reduces the complexity and is computationally much more efficient than directly working in 3D space. We adopt the representation and architecture from 3DShape2VecSet but augment it with new scaling-up strategies. Specifically, we encode a 3D geometry into latent space by sampling a point cloud \mathbf{X} from a 3D mesh surface \mathbf{M} . This point cloud is encoded into a latent code with dynamic shape $Z = \mathbb{R}^{L \times 64}$ with a length L and channel size 64 using the encoder \mathcal{E} of a transformer-based VAE, expressed as $Z = \mathcal{E}(\mathbf{X})$. We then learn a DiT to denoise the latent code Z_t with noise at step t . Finally, the VAE decoder \mathcal{D} decodes the generated latent codes from DiT into a neural field, as $\mathcal{D}(Z_0, p) \rightarrow [0, 1]$, where p is a testing coordinate in space, and \mathcal{D} determines if p is inside or outside the 3D shape. Recall our objective is to achieve substantial scaling-up of this architectural model. To maintain robust scale-up while facilitating effective training, we develop a new scheme based on multi-resolution encoding. Such an extension not only enhances the model's capacity to manage large-scale data but also ensures refined training outcomes, underpinning the model's performance, scalability, and adaptability.

Multi-resolution VAE. In the design of our VAE module, we follow the structure outlined in 3DShape2VecSet. This involves embedding the input point cloud $\mathbf{X} \in \mathbb{R}^{N \times 3}$ sampled from a mesh \mathbf{M} into a latent code using a learnable embedding function and a cross-attention

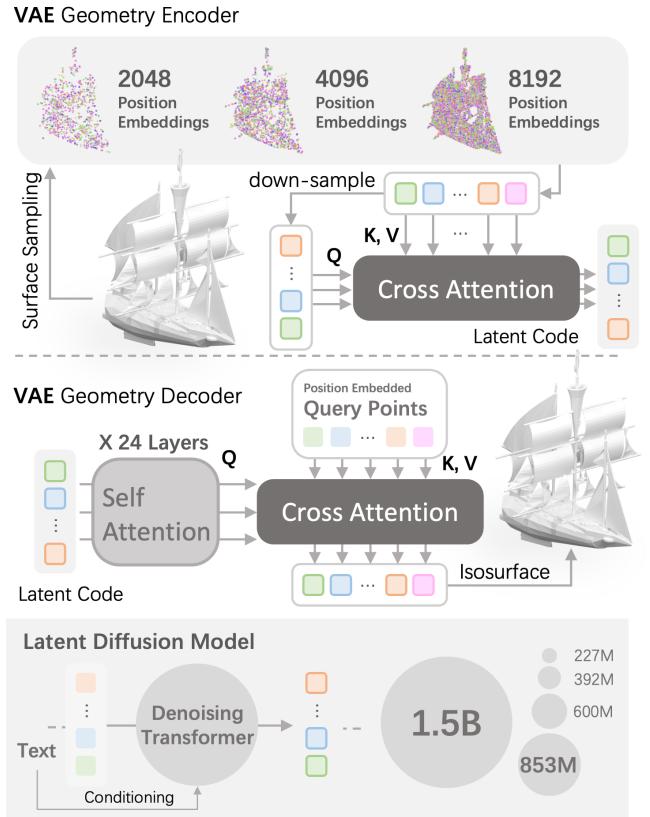


Fig. 3. Network design of our VAE and DiT. With a minimalist design, our DiT supports scalable training and VAE operates effectively across various geometric resolutions.

encoding module:

$$\mathbf{Z} = \mathcal{E}(\mathbf{X}) = \text{CrossAttn}(\text{PosEmb}(\tilde{\mathbf{X}}), \text{PosEmb}(\mathbf{X})), \quad (1)$$

where $\tilde{\mathbf{X}}$ denotes a down-sampled version of \mathbf{X} at 1/4 scale, effectively reducing the latent code's length L to a quarter of the input point cloud size N . The VAE's decoder, consisting of 24 self-attention layers and a cross-attention layer, processes these latent codes and a list of query points \mathbf{p} , outputting occupancy logits:

$$\mathcal{D}(\mathbf{Z}, \mathbf{p}) = \text{CrossAttn}(\text{PosEmb}(\mathbf{p}), \text{SelfAttn}^{24}(\mathbf{Z})). \quad (2)$$

Our VAE is dimensioned at 512 with 8 attention heads, culminating in a total of 82 million parameters. The latent code size is configured as $L \times 64$, with L varying based on the input point cloud size.

In 3DShape2VecSet, the point clouds are generally of small sizes and therefore are insufficient to capture fine geometric details. We adopt a multi-resolution approach. At each iteration, we first randomly choose a sampling size N from 2048, 4096, or 8192, to ensure variability. Next, we sample the corresponding number of surface points from the input mesh \mathbf{M} .

Coarse-to-fine DiT. Our DiT employs a minimalist yet effective structure, consisting of a 24-layer pure transformer, with added cross-attention mechanisms for accommodating text prompt conditions. The encoding process involves sampling $N = 4L$ surface

Table 1. DiT specifications and training hyper parameters.

Model size	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Latent length	Batch size	Learning rate
Tiny	227M	24	768	12	64	512	1024	1e-4
Small	392M	24	1024	16	64	512	16384	1e-5
						1024	8192	5e-6
Medium	600M	24	1280	16	80	512	16384	1e-4
						1024	8192	5e-5
Large	853M	24	1536	16	96	512	8192	1e-4
						1024	4096	1e-5
						2048	2048	5e-6
XL	1.5B	24	2048	16	128	512	4096	1e-4
						1024	2048	1e-5
						2048	1024	5e-6

points from a 3D mesh, which are subsequently encoded into a latent code $\mathbf{Z} \in \mathbb{R}^{L \times 64}$ using $\mathcal{E}(\cdot)$. In parallel, a pretrained language model, specifically CLIP-ViT-L/14 [Radford et al. 2021], processes the text prompt into textual features \mathbf{c} . The DiT’s role, defined as $\epsilon(\cdot)$, is to predict the noise in \mathbf{Z}_t at timestep t :

$$\epsilon(\mathbf{Z}_t, t, \mathbf{c}) = \{\text{CrossAttn}(\text{SelfAttn}(\mathbf{Z}_t \# \mathbf{t}), \mathbf{c})\}^{24}, \quad (3)$$

where the symbol $\#$ signifies concatenation, and for clarity, certain elements like projection and feed-forward layers are omitted from this description. To efficiently capture fine geometric details, we optimize the DiT on high-dimensional latent sets. Specifically, we employ a progressive training scheme, varying the latent code length for quicker convergence and time efficiency. Starting with a length of latent code $L = 512$ at a higher learning rate, we gradually increase to 1024, then to 2048, each time reducing the learning rate based on empirical observations. This progressive scaling method ensures robust and efficient training of our DiT.

Scaling-up Scheme. Scaling-up CLAY requires enhancing both the VAE and DiT architectures with pre-normalization and GeLU activation, to facilitate faster computation of attention mechanism. The feed-forward dimension is four times of the model dimension. For noise scheduling, a discrete scheduler with 1000 timesteps is employed, and a cosine beta schedule is utilized during training. Following the latest practice on diffusion training [Lin et al. 2024], we implement zero terminal SNR by rescaling betas and opt for “v-prediction” as our training objective, a strategy that promotes stable inference. To evaluate the impact of model size on performance, we train five DiTs with sizes varying from 227 million to 1.5 billion parameters, as outlined in Table 1. Our smallest model, designed for verification, can be trained on a single node with 8 NVidia A800 GPUs due to its smaller batch size, to support preliminary experiments. For larger models, we employed larger batch sizes, resulting in improved training stability and faster convergence rates. Our largest model, the XL, was trained on a cluster of 256 NVidia A800 GPUs, for approximately 15 days, with progressive training.

Following the insights in Gasmundo and Maile [2023] of *Head addition*, *Heads expansion* and *Hidden dimension expansion*, we progressively scale up the DiT during training. This approach offers benefits such as enhanced time efficiency, improved knowledge retention, and a reduced risk of the model trapped in the local optima. This scaling-up process in DiT training, leveraging the suggested

training techniques, is designed to optimize the model’s learning trajectory and overall performance.

Our model, once trained on our expanded dataset (Sec. 3.2), demonstrates strong capabilities to generate 3D objects from text prompts at a high quality and accuracy. During inference, we utilize a 100-timestep denoising process with linear-space timestep spacing for efficient 3D geometry generation. The model then engages in dense sampling at a 512^3 grid resolution with our VAE’s geometry decoder, precisely determining occupancy values for detailed geometry capture, which are then converted to mesh using Marching Cubes.

3.2 Data Standardization for Pretraining

The effectiveness and robustness of large-scale 3D generative models rely on the quality and the scale of 3D datasets. Unlike text and 2D images which are abundant and hence can support Stable Diffusion, 3D datasets such as ShapeNet [Chang et al. 2015] and Objaverse [Deitke et al. 2023] are limited in size or quality. To obtain large-scale high quality 3D data, it is essential to overcome challenges such as non-watertight meshes, inconsistent orientations and inaccurate annotation. Our solution is to apply a remeshing method for geometry unification and GPT-4V [OpenAI 2023] for precise automatic annotation. Our standardization starts with filtering out unsuitable data, such as complex scenes and fragmented scans, resulting in a refined collection of 527K objects from ShapeNet and Objaverse, laying a robust groundwork for enhanced model performance through tailored unification and annotation techniques.

Geometry Unification. To address the challenge of predicting a 3D shape’s occupancy field in the presence of non-watertight meshes after data filtration, we propose a standardized geometry remeshing protocol to ensure watertightness while avoiding discarding useful data in the training set. Popular remeshing tools such as Manifold [Huang et al. 2018a], while efficient, tend to smooth edges and corners, with its updated version, ManifoldPlus [Huang et al. 2020], showing improved but inconsistent results. Alternatives such as “mesh-to-sdf” [Marian 2021] and Dual Octree Graph Networks (DOGN) [Wang 2022; Wang et al. 2022] set out to compute Signed and Unsigned Distance Fields but they are computationally costly. As depicted in Fig. 4, the quality of training data for advanced 3D models is affected by these remeshing techniques, underscoring the need for a strategy that balances precision and efficiency. Specific criteria for effective remeshing include: (1) Geometric Preservation

ALGORITHM 1: Our remeshing method

Data: Input *Triangles* normalized in [-1,1]
Result: Output watertight *Mesh*
 Initialize *SDF* as a 3D array of dimensions $512 \times 512 \times 512$;
 Initialize *SDF* default value $+\infty$;
foreach *T* **in** *Triangles* **do**
 | **foreach** (i, j, k) **in** grid points within *T*'s bounding box **do**
 | | update the UDF value and store in *SDF*[*i, j, k*];
 | **end**
end
if common practice **then**
 | *Mesh* \leftarrow marchingcubes(*SDF*, 2/512);
 | *Mesh* \leftarrow the largest connected part in *Mesh*;
else
 | calculate grid-based visibility for each grid point using *SDF*;
 | **foreach** (i, j, k) **in** all grid points **do**
 | | **if** grid point (i, j, k) is not visible from any directions **then**
 | | | *SDF*[*i, j, k*] $\leftarrow 0$;
 | | **end**
 | **end**
 | *Mesh* \leftarrow marchingcubes(*SDF*, 2/512);
end

- maintaining essential geometric features with minimal alteration;
- (2) Volume Conservation - ensuring the integrity of all structural elements; and (3) Adaptability to Non-Watertight Meshes - proficiently managing non-watertight models to preserve volumetric accuracy essential for model training.

Inspired by DOGN [Wang 2022; Wang et al. 2022], we adopt the Unsigned Distance Field (UDF) representation because of its seamless conversion capabilities between mesh formats and correction of inconsistencies in vertex and face density. As outlined in Algorithm. 1, we start this process by normalizing the input mesh within a range of [-1,1]. UDF values are computed on a 512^3 resolution grid, with each triangle processed iteratively. We then extract an isosurface with a threshold of $\frac{2}{512}$, conferring an incremental thickness to each face while allowing for the depiction of single-face structures in the input mesh. Our experiments find that this is an important feature but was discarded by other remeshing techniques. In addition, the traditional Marching Cubes algorithm for isosurface extraction can produce a mere thin shell in scenarios involving mesh holes. To address this, we employ a grid-based visibility computation before isosurface extraction. Specifically, we label a grid point as “inside” when completely obscured from all angles, maximizing volume for stable VAE training. For efficiency, we only compute UDF value on grid points within triangles' bounding boxes, significantly reducing the computational load. The Marching Cubes algorithm is then applied to these points to extract the surface at a designated isosurface level. This modification greatly reduces the computation time, rendering the process more efficient and feasible.

Geometry Annotation. The impact of text prompts on 2D image generation by models such as Stable Diffusion [Rombach et al. 2022] and SDXL [Podell et al. 2023] reveals the importance of precise prompts in any successful 3D generative model. Previous studies have demonstrated how “magic prompts” guide specific content and

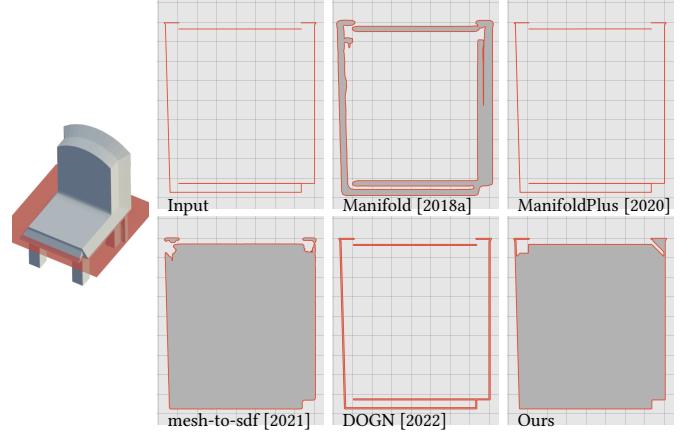


Fig. 4. Comparison against existing mesh preprocessing methods using cross-sectional analysis. The input is a non-watertight chair with its surface not closed. Red lines correspond to the faces of meshes, light gray indicates “outside” and dark gray indicates “inside”. Our method maximizes positive volume while faithfully preserving geometric features. This robustness extends to non-watertight input meshes, ensuring consistent and reliable results.

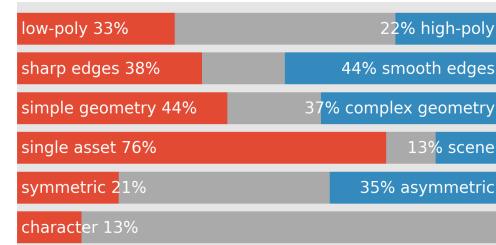


Fig. 5. Percentage of each annotated tag in our complete training data.

style. Recognizing this, we emphasize accurate textual prompts in our 3D model to capture geometric and stylistic details of objects. We have developed unique prompt tags and utilized GPT-4V [OpenAI 2023] for producing detailed annotation, enhancing the model’s capability to interpret and generate complex 3D geometries with nuanced details and diverse styles.

Our annotation for each 3D object is designed to be comprehensive, ensuring precise characterizations. In our annotation process, the Rotation and Symmetry aspects are marked using our own implemented schemes whereas other annotations are achieved by leveraging GPT-4V. Details of this specific algorithm will be discussed in followup sections. The overall annotation strategy aims to capture essential geometric properties of a 3D object in terms of attributes:

- Object Description: describes what it represents.
- Geometric Features: brief, descriptive phrases.
- Polygon Count: whether it is “low-poly” or “high-poly”.
- Edge Type: differentiates “sharp” edges and “smooth” edges.
- Geometry Complexity: “simple” or “complex”.
- Asset Type: distinguishes between “single asset” and “scene”.
- Symmetry: identifies as “symmetric” or “asymmetric”.
- Character Presence: determines if it represents a character.
- Rotation: aligns each object to a canonical orientation.



Fig. 6. Our Material Diffusion architecture and Asset Enhancement pipeline. Our Material Diffusion network, derived from existing diffusion models, facilitates efficient fine-tuning. Following mesh quadrification and atlasing, it generates textures through a multi-view approach and subsequently back-project them onto UV maps. The resultant materials, closely aligned with geometries and user inputs (text/image), faithfully respond to diverse lighting conditions, culminating in realistic renderings.

The annotation process begins with manually establishing the correct orientation for a subset of 10,000 3D objects, setting a standard for accuracy. A transformer network using CLIP-ViT-L/14’s vision encoder is trained on this dataset for orientation prediction in larger datasets. For symmetry annotation, we developed a scheme based on point cloud derived metrics, setting empirical thresholds for “symmetric” and “asymmetric” classifications. Further, objects are rendered in Blender from four 45-degree diagonal views for enhanced feature recognition. These images feed into GPT-4V, filling out the remaining annotation fields. The annotations are enriched with captions chosen randomly from GPT-4V-generated options, greatly improving data diversity and representativeness for training our 3D model.

4 ASSET ENHANCEMENT

To make the generated digital assets directly usable in existing CG pipelines, we further adopt a two-stage scheme: post-generation geometry optimization and material synthesis. Geometry optimization ensures structural integrity and compatibility as well as refines the model’s form aesthetically and functionally. Material synthesis is crucial for adding lifelike qualities through realistic textures and materials. Together, these steps transform coarse meshes into more engaging assets in digital environments.

Mesh Quadrification and Atlasing. In CLAY, the initial geometric meshes via the Marching Cubes algorithm typically consist of millions of uneven triangles. While suitable for early stages, such structure poses challenges in editing and application, notably when exported to mesh editing tools or game engines. In addition, it would require complicated automatic UV unwrapping – a crucial step in

texture mapping and material synthesis. To overcome these challenges, we transform these triangle-faced meshes into quad-faced ones using off-the-shelf tools [Blender Online Community 2024; Huang et al. 2018b], preserving key geometric features such as sharp edges and flat surfaces. This quadrification process is highly crucial for yielding high-quality final meshes, facilitating the effective conversion from coarse 3D models to the refined assets.

Material Synthesis. In addition to geometry generation, it is equally important to produce high quality textures in 3D generation. The physically-based rendering (PBR) materials, typically consisting of diffuse, metallic, and roughness textures, are essential for conveying convincing visual experiences in digital environments. Existing methods in PBR texture generation by far have focused on creating a very small subset of these materials. In addition, these approaches lack supervision on specific material attributes, limiting the rendering quality. For example, RichDreamer [Qiu et al. 2024] generates diffuse maps without roughness and metallic predictions. Fantasia3D [Chen et al. 2023a] and UniDream [Liu et al. 2023a] can produce roughness and metallic attributes but do not consider richer attributes. Therefore they cannot generate richer material types.

We aim to synthesize a wide range of PBR materials including diffuse, roughness, and metallic textures. From Objaverse [Deitke et al. 2023], we carefully choose over 40,000 objects, each characterized by high-quality PBR materials. Utilizing this dataset, we developed a multi-view Material Diffusion to synthesize textures with a significantly speed-up over existing methods, which are then accurately mapped onto the geometries’ UV space in a way similar to TEXTure [Richardson et al. 2023].

We modify MVDream [Shi et al. 2024], originally designed for image space generation, to suit the need for generation from texture attributes with additional channels and modalities. Inspired by HyperHuman [Liu et al. 2023b], we integrate three branches into its UNet’s outer most convolutional layers, each with skip connections, allowing concurrent denoising across various texture modalities and ensuring view consistency. Similar to MVDream, our training process includes selecting orthogonal-view rendered texture images for each 3D object in training data, and applying both full-parameter for add-on layers and LoRA-based fine-tuning for inside layers, focusing on generating high-quality, view-consistent PBR materials. Following the same training regimen, our model capably synthesizes texture images from four camera viewpoints, aligned precisely with the input geometry. This is achieved by applying the pretrained ControlNet [Zhang et al. 2023b], with each target view’s rendered normal map as inputs. Such an approach not only ensures geometric accuracy but also allows for image-based input customization via IPAdapter [Ye et al. 2023]. To further enhance texture detail, we employ a targeted inpainting approach as introduced in Text2Tex [Chen et al. 2023b], and integrate advanced super-resolution techniques Real-ESRGAN [Wang et al. 2021b] and MultiDiffusion [Bar-Tal et al. 2023], achieving 2K texture resolution sufficient for most realistic rendering tasks. Our Material Diffusion scheme enables the creation of high-quality textures, resulting in production quality rendering. Our generation results are of a much higher quality and visual pleasantness than previous 3D generation schemes enhancing engagement and realism of the generated 3D assets.

5 MODEL ADAPTATION

CLAY, when pretrained, also serves as a versatile foundation model. For example, CLAY directly supports Low-Rank Adaptation (LoRA) on the attention layers of our DiT. This allows for efficient fine-tuning, enabling the generation of 3D content targeted to specific styles, as illustrated in Fig. 7. Further, the minimalist architecture enables us to efficiently support various conditional modalities to support conditioned generation. We implement several exemplary conditions that can be easily provided by a user, including text, which is natively supported, as well as image/sketch, voxel, multi-view images, point cloud, bounding box, and partial point cloud with an extension box. These conditions, which can be applied individually or in combination, enable the model to either faithfully generate content based on a single condition or create 3D content with styles and user controls blended from multiple conditions, offering a wide range of creative possibilities.

5.1 Conditioning Scheme

Building upon our existing text prompt conditioning, we extend the model to incorporate additional conditions in parallel. Our use of pre-normalization [Xiong et al. 2020] converts the attention results into residuals, enabling the addition of extra conditions as parallel residuals alongside the text condition, which can be expressed as:

$$\mathbf{Z} \leftarrow \mathbf{Z} + \text{CrossAttn}(\mathbf{Z}, \mathbf{c}) + \sum_{i=1}^n \alpha_i \text{CrossAttn}_i(\mathbf{Z}, \mathbf{c}_i), \quad (4)$$



Fig. 7. Generation after LoRA fine-tuning on different specific datasets including the rock dataset and the pocket monster dataset. After generating a LEGO duck (center), which was one of the first toys designed by LEGO founder Ole Kirk Christiansen, CLAY can further generate variants in stone styles (left) and pocket styles (right).

where CrossAttn denotes the original text conditioning, CrossAttn_i denotes the i -th additional trainable module and \mathbf{c}_i is the i -th condition. The inclusion of scalar α_i in this residual framework allows for direct manipulation of the influence exerted by each additional condition.

While this conditioning scheme is general, obtaining the embedded condition \mathbf{c}_i requires careful calibration. For image/sketch conditions, we utilize the pretrained DINOv2 [Oquab et al. 2024] model to extract features as conditions and directly integrate using the cross-attention in the above equation. However, for spatially related modalities such as voxel, multi-view images, point cloud, bounding box, and partial point cloud with an extension box, directly applying cross-attention on features do not guarantee to preserve spatial information pertaining to those conditions. To maintain spatial integrity, we have devised a specific learning strategy.

Spatial Control. Our 3D geometry generative model incorporates conditions in 3D modalities, a unique feature absent in previous approaches. This allows for spatial controls similar to those in 2D diffusion models. However, different from 3D UNet structures with convolutional backbones that naturally maintain spatial resolution, our approach uses a VAE that dynamically generates latent codes interwoven with spatial coordinates, imposing a new set of challenges for achieving precise spatial controls.

To address the integration of 3D conditions, we set out to learn additional positional embeddings for spatial features. This allows our attention layer to differentiate point coordinates from their features effectively. We start by associating the feature embedding $\mathbf{f} \in \mathbb{R}^{M \times C}$, learned during fine-tuning or extracted from a backbone network, with sparse 3D points $\mathbf{p} \in \mathbb{R}^{M \times 3}$ sampled based on the type of condition being used, where M and C are the length and channels of specific conditioning embedding. The exact sampling strategy is tailored to each condition type and will be detailed subsequently. We then apply cross attention more specifically as:

$$\text{CrossAttn}_i(\mathbf{Z}, \mathbf{f} + \text{PosEmb}(\mathbf{p})), \quad (5)$$

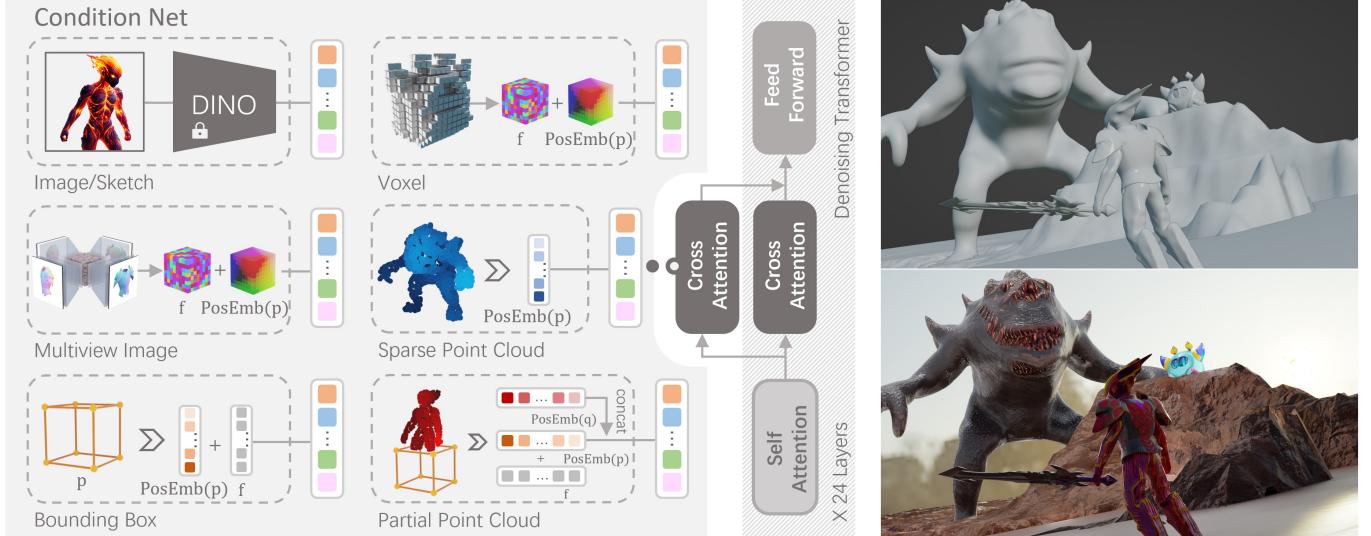


Fig. 8. Illustration of our network’s conditioning design across various modalities. When used together, they support the creation of cinematic scenes with lifelike renderings.

Table 2. Conditioning module specifications.

Conditioning	n_{params}	M	C	Backbone
Image/Sketch	352M	257	1536	DINOv2-Giant
Voxel	260M	8^3	512	/
Multi-view images	358M	8^3	768	DINOv2-Small
Point cloud	252M	512	512	/
Bounding box	252M	8	512	/
Partial point cloud	252M	$2048+8$	512	/

where $\text{PosEmb}(\cdot)$ is the learnable positional embedding. This method allows for the effective integration of various 3D modalities into our model.

5.2 Implementation

We discuss how to implement a variety of conditions for controlled 3D content generation. Each condition involves independently training an additional $\text{CrossAttn}_i(\cdot)$ while keeping other parameters fixed. Fig. 8 and Table. 2 showcase the specifications and hyperparameters of training for each condition. The base model and training data is described in Sec. 6.

Images and Sketches. For image and sketch conditions, we use the pretrained Vision Transformer (ViT) DINOv2 to extract both patch and global features. These features are integrated into CLAY via cross-attention, as indicated in Eqn. 4. This module is trained using rendered RGB images and corresponding sketches from our dataset, ensuring alignment between the generated 3D models and the visual characteristics of the conditioning images or sketches.

Voxel. Voxels represent spatial cubes and provide an intuitive medium for 3D construction. To integrate voxel-based guidance, we initially construct a 16^3 voxel grid for each 3D object in our dataset, marking each cell as occupied or vacant. These voxel grids are down-sampled to a 8^3 feature volume using 3D convolution. The volume features $f \in \mathbb{R}^{8^3 \times C}$, added with positional embeddings of

volume centers $\text{PosEmb}(p)$, are then flattened and integrated into the DiT through cross-attention. After training, CLAY can generate 3D geometries that correspond to user-defined voxel structures, effectively translating abstract voxel designs into intricate 3D forms.

Bounding Boxes. Bounding boxes provide a straightforward way for users to control the aspect ratio and position of 3D objects, essential in interactive generation applications. The bounding box features $f \in \mathbb{R}^{8 \times C}$, added with positional embeddings $\text{PosEmb}(p)$, are learned during condition fine-tuning, enabling precise spatial control.

Sparse Point Cloud. Point clouds offer an easily accessible abstraction for 3D shapes. CLAY can use sparse point clouds as conditions, to generate variants from input meshes or points. For this, we set feature embeddings $f = 0$, which indicates no feature embedding, and sample 512 points as p and learn the corresponding positional embedding $\text{PosEmb}(p)$. This allows CLAY to generate detailed 3D geometries based on sparse surface point clouds while maintaining the overall shape and appearance.

Multi-view Images. CLAY also supports multi-view images or multi-view normal maps as conditions, offering spatial control through projected views of 3D geometries. As a demonstration, we use DINOv2 to extract features from various views’ images generated by the Wonder3D. These features are back-projected into a 3D volume similar to previous method [Liu et al. 2024a], then down-sampled and flattened for integration into the DiT using cross-attention, a similar procedure to the voxel condition.

Partial Point Cloud with Extension Box. This condition specifically aims to address the point cloud completion task, where a certain bounding box indicates the generation region of missing parts. We merge the input point cloud with the corner points of an extension box, applying a similar approach for learning bounding box conditioning and sparse point cloud conditioning by concatenating

these two set of features. This integration is instrumental in the effective reconstruction of incomplete geometries, precisely within the specified extension areas.

6 RESULTS

We have trained five base models of different model sizes using our full training data with length of latent code $L = 1024$, ranging from Tiny-base to XL-base. Based on Large-base and XL-base, we have trained Large-P and XL-P on a high-quality subset of our training data including 300K objects, using length of latent code $L = 1024$. Based on Large-P and XL-P, we have further trained using the same subset data but with a longer length of latent code $L = 2048$. For adaptations including LoRA fine-tuning and conditioning, we have trained these modules based on XL-P using the same high-quality subset data, with each module independently trained for 8 hours.

Next, we demonstrate the generation results with various conditioning using the XL-P model of CLAY. Fig. 9 illustrates a sample collection of 3D models generated by CLAY, demonstrating its versatility in producing a wide range of objects with intricate details and textures. From ancient tools to futuristic spacecraft, the collection traces through a fascinating human history of imagination, celebrating the fusion of art, tech, and human ingenuity as well as embracing our rich cultural heritage. The array also includes technologically advanced vehicles, cultural artifacts, everyday items, and imaginative elements, all of which highlight the model’s capacity for high-fidelity and varied 3D creations suitable for applications in gaming, film, and virtual simulations.

Fig. 10 showcases CLAY’s conditioning capabilities across different modalities. With image conditioning, CLAY generates geometric entities that faithfully resemble the input images, be it real-world photos, AI-generated concepts, or hand-drawn sketches. CLAY also allows for the creation of entire towns or bedrooms from scattered bounding boxes. Using multi-view images, it reliably reconstructs 3D geometries from multiple perspectives and normal maps. CLAY further manages to generate from sparse point cloud, indicating it can also serve as an effective surface reconstruction tool, analogous but outperforming GCNO [Xu et al. 2023] from as few as 512 points in the “knot” case. Additionally, CLAY can be used to further improve 3D geometries generated by existing techniques while maintaining sharp edges and flat surfaces largely missing in prior art. Diversity wise, CLAY excels in generating rich varieties in shapes from the same voxel input, transforming the same coarse shape into anything from a futuristic monument to a Medieval Castle, from an SUV to a space shuttle, resembling our unlimited imagination. Finally, CLAY can be used to complete missing parts from partially available geometry and therefore serves as both a geometry completion tool and an editing tool. For example, it allows us to alter a monster’s body or turn a companion robot into a battle-ready counterpart, a Star Wars fantasy for many.

6.1 Evaluations

We have conducted comprehensive evaluations on CLAY, focusing on various aspects including model sizes, conditioning types, prompt engineering, multi-view conditioning, and geometry diversity.

Quantitative Evaluations. Here we evaluate nine versions of CLAY as illustrated in Table. 3. The text-to-shape evaluation employs metrics including render-FID, render-KID, P-FID, P-KID, CLIP, and ULIP-T, using a 16K text-shape pair validation set. We apply FID and KID to both 2D (image rendering) and 3D (point cloud) feature spaces. For render-FID and render-KID, images are rendered from eight views, and PointNet++ [Qi et al. 2017] is used to extract 3D features for P-FID and P-KID assessments. Additionally, we utilize CLIP-ViT-L/14 [Radford et al. 2021] for evaluating text-rendering similarity and ULIP-2 [Xue et al. 2023] for text-shape alignment. Specifically, ULIP-T is defined as $\text{ULIP-T}(T, S) = \langle E_T, E_S \rangle$, corresponding to the inner product of normalized ULIP features of caption T and generated geometry S . Table. 3 reveals the apparent trend that larger models excel over the smaller ones in text-to-shape generation tasks, demonstrated by higher scores and more accurate text-shape alignment.

We have also evaluated various conditioning modules, including image, multi-View normal, bounding box, and voxel, using XL-P as the base model. Additional metrics such as Chamfer Distance (CD), Earth Mover’s Distance (EMD), Voxel-IoU, and F-score are employed to assess conditioned shape generation accuracy. We further introduce ULIP-I to evaluate alignment between the condition image and the generated shapes. Both ULIP-T and ULIP-I are assessed across all conditions, except a few, such as voxel, that do not utilize text or image inputs. Table. 4 shows that with as few as a single condition, CLAY already manages to generate geometry of very high fidelity. Applying additional conditions further improves geometric details while maintaining high alignment with the ground truth text or image at the feature level. It is worth mentioning that among all settings, our multi-view normal (MVN) conditioning model exhibits one of the most outstanding performances. Therefore, CLAY can be also deemed as a reliable reconstruction back-end for other multi-view generation models [Long et al. 2024; Shi et al. 2024].

Prompt engineering. We further explore the effects of varied prompt tags on geometry generation, as illustrated in Fig. 11. For example, by incorporating “asymmetric geometry” into our prompts, CLAY successfully generates asymmetric table and church. Similarly, the transition from “sharp edges” to “smooth edges” prompts manages to modify Pikachu and a dog into more rounded shapes. Interestingly, typical 3D models composed of high-polygon meshes such as aircrafts and tanks can be transformed into low-polygon variants using CLAY. In contrast, the “complex geometry” tag prompts the generation of intricate details in a chandelier and a sofa. Adding “character” will transform inanimate objects such as a fireplug and a mailbox into anthropomorphic figures, reminiscent to magics taught at the Hogwarts. This experiment further indicates that specific annotated tags applied during training can effectively steer the model to produce geometries with desired complexities and styles, enhancing the quality and specificity of the generated shapes.

Geometry Diversity. CLAY also excels at generating high-quality geometries with rich diversity. In Fig. 12, we showcase the results generated by CLAY conditioned on either text or image inputs, alongside the most relevant samples retrieved from the dataset. To perform geometry retrieval, we utilize cosine similarity to compare the normalized ULIP feature of the generated geometry with that



Fig. 9. Evolution of human innovation, from primitive tools and cultural artifacts to modern electronics and futuristic imaginings, generated by CLAY.



Fig. 10. Sample creations using CLAY, with conditions marked in sky blue and input geometries for respective conditioning (if available) in sandy brown.

Table 3. Quantitative evaluation of Text-to-3D for models of different sizes.

Model name	Latent length	render-FID \downarrow	render-KID($\times 10^3$) \downarrow	P-FID \downarrow	P-KID($\times 10^3$) \downarrow	CLIP(I-T) \uparrow	ULIP-T \uparrow
Tiny-base	1024	12.2241	3.4861	2.3905	4.1187	0.2242	0.1321
Small-base	1024	11.2982	4.2074	1.9332	4.1386	0.2319	0.1509
Medium-base	1024	13.0596	5.4561	1.4714	2.7708	0.2311	0.1511
Large-base	1024	6.5732	2.3617	0.8650	1.6377	0.2358	0.1559
XL-base	1024	5.2961	1.8640	0.7825	1.3805	0.2366	0.1554
Large-P	1024	5.7080	1.9997	0.7148	1.2202	0.2360	0.1565
XL-P	1024	4.0196	1.2773	0.6360	1.0761	0.2371	0.1564
Large-P-HD	2048	5.5634	1.8234	0.6394	0.9170	0.2374	0.1578
XL-P-HD	2048	4.4779	1.4486	0.5072	0.5180	0.2372	0.1569

Table 4. Quantitative evaluation of Multi-modal-to-3D for different conditions and their combinations.

Condition	CD($\times 10^3$) \downarrow	EMD($\times 10^2$) \downarrow	Voxel-IoU \uparrow	F-Score \uparrow	P-FID \downarrow	P-KID($\times 10^3$) \downarrow	ULIP-T \uparrow	ULIP-I \uparrow
Image	12.4092	17.6155	0.4513	0.4070	0.9946	1.9889	0.1329	0.2066
MVN	0.9924	5.7283	0.7697	0.8218	0.3038	0.2420	0.1393	0.2220
Voxel	0.5676	8.4254	0.6273	0.6049	2.6963	5.0008	0.1186	0.1837
Image-Bbox	5.4733	14.0811	0.5122	0.4909	1.5884	3.2994	0.1275	0.2028
Image-Voxel	0.7491	8.1174	0.6514	0.6541	2.4866	6.8767	0.1262	0.2017
Text-Image	7.7198	14.5489	0.4980	0.4609	0.7996	1.4489	0.1407	0.2122
Text-MVN	0.7301	5.4034	0.7842	0.8358	0.2184	0.1233	0.1424	0.2240
Text-Bbox	5.6421	14.6170	0.4921	0.4659	2.0074	4.0355	0.1417	0.1838
Text-Voxel	0.6090	7.4981	0.6737	0.6689	1.0427	1.0903	0.1397	0.2036

of geometries in the dataset. With text inputs, CLAY manages to generate novel shapes that differ from any existing ones in the dataset. When presented with image inputs, CLAY faithfully reconstructs the content of the image while introducing novel structural combinations that are absent from the dataset. For instance, the airplane depicted at the bottom of Fig. 12 represents a novel concept art piece generated by AI. It features the fuselage of a passenger airplane, uniquely merged with square air intakes and the tail fins reminiscent of a fighter jet – a design composite that is never seen in the training data. Nevertheless, CLAY accurately generates its 3D geometry, capturing a high degree of resemblance to the provided image.

Effectiveness of MVN Conditioning. While single image conditioning tends to allow for more liberty in creation, multi-view conditioning harnesses multiple perspectives to deliver more detailed and precise control over the targeted generation, akin to a pixel-align sparse-view reconstruction approach. Fig. 13 shows an example where we use an initial image of a panther’s head (top left) as a starting point. This image, when processed through our single image conditioning, yields a solid 3D geometry (left column). In contrast, when the concept is further solidified using Wonder3D to generate multi-view images and corresponding normal maps, it results in a panther face mask with a notably thin surface (top right). Based on these multi-view images, our multi-view images conditioning using normal maps successfully harnesses these multiple views, leading to a faithful yet efficient synthesis of the thin surface (center column), distinct from the traditional NeuS method applied to Wonder3D’s outputs (right column). This comparison underscores the precision

Table 5. Quantitative comparison with state-of-the-art methods.

Method	CLIP	CLIP	ULIP-T	ULIP-I	Time
Text-to-3D	(N-T)	(I-T)			
Shap-E	0.1761	0.2081	0.1160	/	~10s
DreamFusion	0.1549	0.1781	0.0566	/	~1.5h
Magic3d	0.1553	0.2034	0.0661	/	~1.5h
MVDream	0.1786	0.2237	0.1351	/	~1.5h
RichDreamer	0.1891	0.2281	0.1503	/	~2h
CLAY	0.1948	0.2324	0.1705	/	~45s
Image-to-3D	(N-I)	(I-I)			
Shap-E	0.6315	0.6971	/	0.1307	~10s
Wonder3D	0.6489	0.7220	/	0.1520	~4min
DreamCraft3D	0.6641	0.7718	/	0.1706	~4h
One-2-3-45++	0.6271	0.7574	/	0.1743	~90s
Michelangelo	0.6726	/	/	0.1899	~10s
CLAY	0.6848	0.7769	/	0.2140	~45s

and efficiency of our multi-view image conditioning in guiding the generation of detailed 3D geometries.

Running Time. Regarding the inference timing breakdown, on a single Nvidia A100 GPU, it takes CLAY about 4 seconds for shape latent generation, 1 seconds to decode the latent due to the efficient adaptive sampling, 8 seconds for mesh processing, and 32 seconds for PBR generation, cumulatively resulting in a total generation time of 45 seconds.

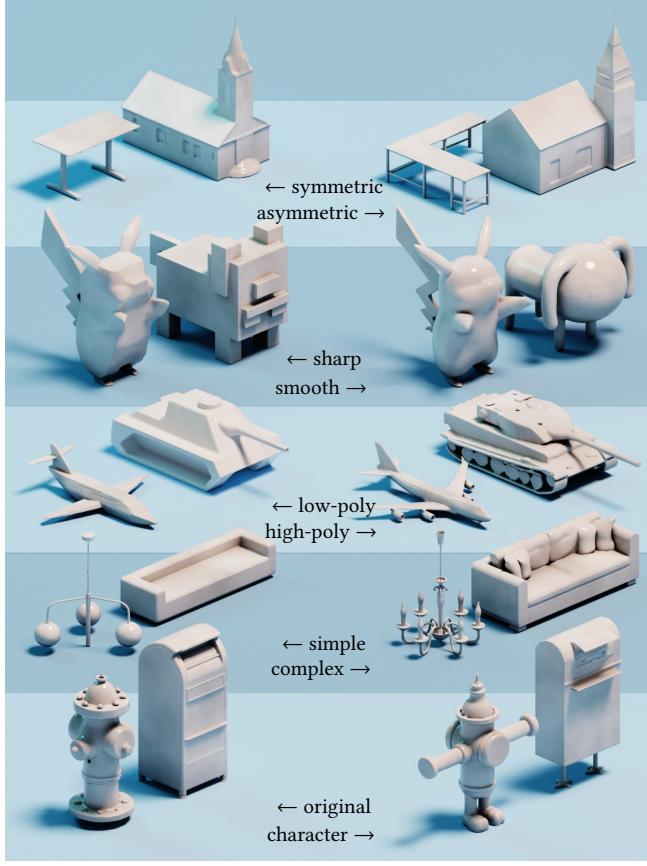


Fig. 11. Evaluation of the CLAY’s ability to alter generated content by incorporating different geometric feature tags in the prompt. We showcase precise controls over the geometry style, in the extreme case transforming a fireplug into a T-pose character.

6.2 Comparisons with SOTA

We compare our methods with leading text-to-3D approaches, namely Shap-E [Jun and Nichol 2023], DreamFusion [Poole et al. 2023], Magic3D [Lin et al. 2023], MVDream [Shi et al. 2024], and RichDreamer [Qiu et al. 2024]. We utilize the open-source code for Shap-E, MVDream, and RichDreamer, while for DreamFusion and Magic3D, we employ a third-party implementation [Guo et al. 2023].

Qualitative Comparison. On Text-to-3D tasks, Fig. 14 illustrates the comparison using normal maps, with text inputs such as “Mythical creature dragon”, “Stag deer”, “Interstellar warship”, “Space rocket”, and “Eagle, wooden statue”. Shap-E exhibits faster generation but lacks complete geometry structures. Pure SDS optimization methods like DreamFusion and Magic3D exhibit the multi-face Janus artifacts. MVDream and RichDreamer, which generate multi-view images for SDS, produce consistent geometries but exhibit a deficiency in surface smoothness and require long optimization times. In contrast, CLAY manages to produce high-quality 3D assets in roughly 45 seconds (5 seconds for geometry and 40 seconds for texture). The generated geometries exhibit smooth surfaces without compromising intricate details, better matching the text prompts.

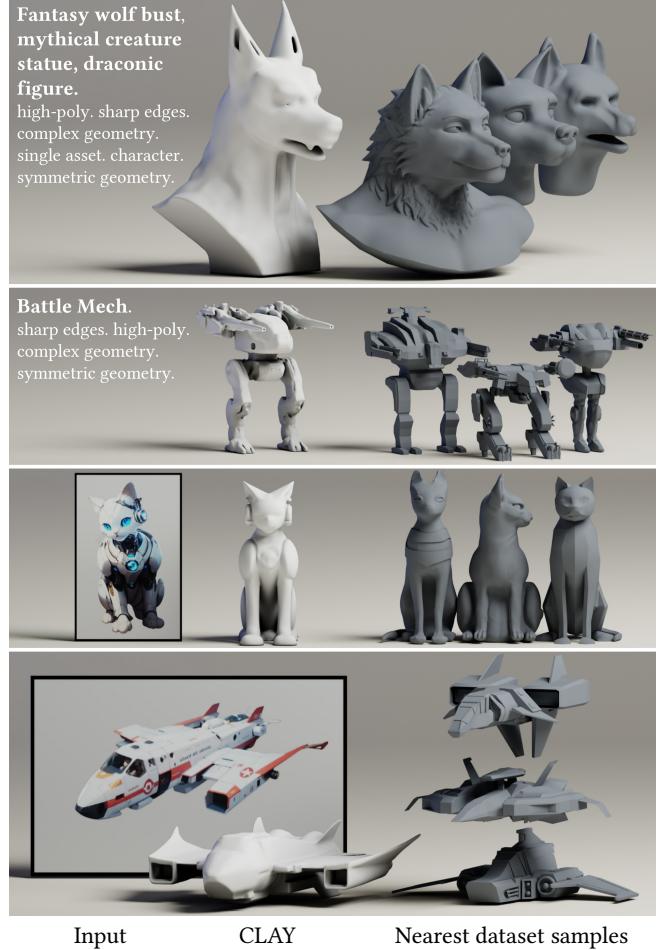


Fig. 12. Evaluation of the geometry diversity. We present top-3 nearest samples retrieved from the dataset. CLAY generates high-quality geometries that match the description but are distinct from the ones in the dataset.

We have further compared the image-to-3D generation quality of between CLAY and SOTA (Shap-E [Jun and Nichol 2023], Wonder3D [Long et al. 2024], One-2-3-45++ [Liu et al. 2024b], DreamCraft3D [Sun et al. 2024], and Michelangelo [Zhao et al. 2023]). We use the official code of respective techniques except One-2-3-45++ where only its online demo is available. Our evaluations include inputs like Chair, Car, Dragon Head, and Sword, detailed in Fig. 15. Note that Michelangelo produces only geometries and we manually assign a similar color for rendering. Shap-E, while fast, fails to accurately reconstruct the input images, resulting in incomplete geometries. Wonder3D, which relies on multi-view images and normal prediction followed by NeuS [Wang et al. 2021a] reconstruction, produces coarse and incomplete geometries due to inconsistencies among the multi-view output. One-2-3-45++ is efficient in creating smooth geometries but lacked details and does not fully maintain symmetry, especially on complex objects such as Chairs and Dragons. DreamCraft3D is an SDS optimization method that produces high-quality output, but is time-consuming and still results in uneven surfaces. CLAY in contrast manages to quickly generate

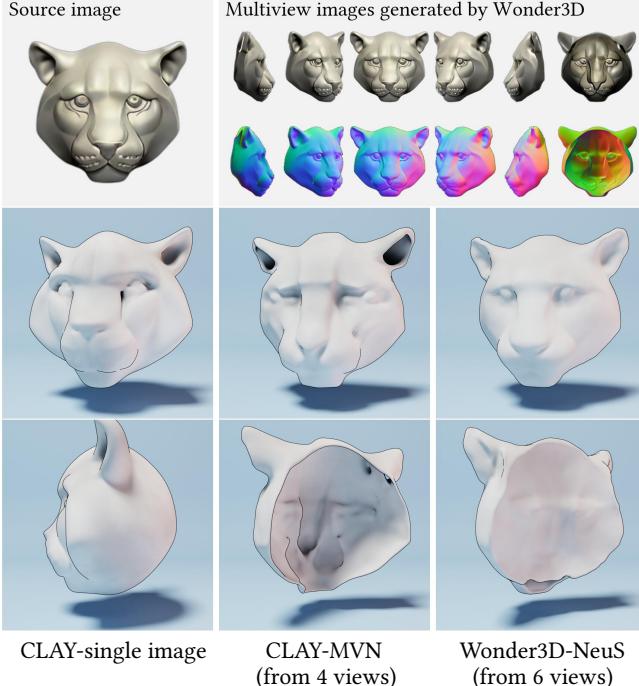


Fig. 13. Geometry generation via single image and multi-view image conditioning with multi-view RGB and normal images generated by Wonder3D.

detailed and high-quality geometries along with high quality PBR textures.

Quantitative Comparisons. We perform additional quantitative comparison using a GPT-4 generated test dataset that includes 50 images and 50 text prompts tailored for text-to-3D and image-to-3D evaluations, respectively. In addition to from ULIP-T and ULIP-I, we render 30 views of RGB images and normal maps for each generated 3D asset, respectively. We apply four CLIP-based metrics to these views, calculating the average to provide a comprehensive assessment. CLIP(N-I) and CLIP(N-T) gauge the geometric alignment of the normal map with the input image and text, respectively whereas CLIP(I-I) and CLIP(I-T) evaluate the appearance by measuring the similarity of rendered images with the input images and text. As shown in Table. 5, CLAY outperforms SOTA techniques in all metrics.

PBR Material Comparison. Another key component in CLAY is material generation. Here we show visual comparisons between CLAY and two leading methods, MVDream [Shi et al. 2024] and RichDreamer [Qiu et al. 2024], using the text prompt “Space rocket”. Fig. 16 illustrates that, under varying lighting conditions MVDream without PBR materials cannot fully reproduce specular highlights. RichDreamer, employing an albedo diffusion model, attempts to distinguish the albedo from complex lighting effects. In this case though, the highlights are modeled as fixed surface textures under changing environment lighting, e.g., on the rocket’s head. In contrast, CLAY faithfully models PBR materials where the rocket’s metallic surfaces exhibit realistic highlights that moves consistently

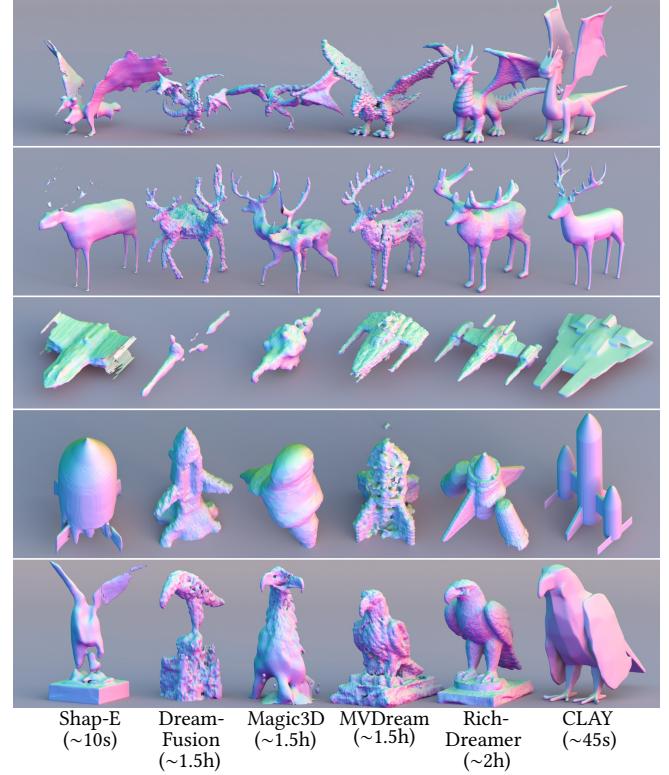


Fig. 14. Comparisons of CLAY vs. state-of-the-art methods on text-conditioned generation. From top to bottom: “Mythical creature dragon”, “Stag deer”, “Interstellar warship”, “Space rocket”, and “Eagle, wooden statue”.

with the moving environment lighting. This also showcases the potential advantages of separating generating geometry and texture.

User studies. We have conduct a comprehensive user study, structured around two primary evaluations: appearance quality for visualization and geometry quality for modeling. We have created a test set consisting of 5 text prompts generated by GPT-4 and 15 images generated by Stable Diffusion. A total of 150 volunteers participated in the study, each evaluating 15 randomly chosen questions to determine their preferred method. We compare CLAY with leading approaches on Text-to-3D and Image-to-3D tasks respectively. Fig 17 shows that CLAY outperforms others in both appearance and geometry in text-to-3D and image-to-3D tasks. Specifically, CLAY secured 67.4% of votes for appearance and 78.9% for geometry in text-to-3D, surpassing the second-ranked RichDreamer, which had a notably longer optimization time of ~2 hours compared to our ~45 seconds. In Image-to-3D, CLAY further garnered 85.4% and 91.2% votes in appearance and geometry, respectively.

7 DISCUSSIONS AND CONCLUSIONS

We have presented CLAY, a large-scale 3D generative model that supports multi-modal controls for high quality 3D asset generation, further bridging the gap between the vivid realms of human imagination and the tangible world of digital creation. By enabling

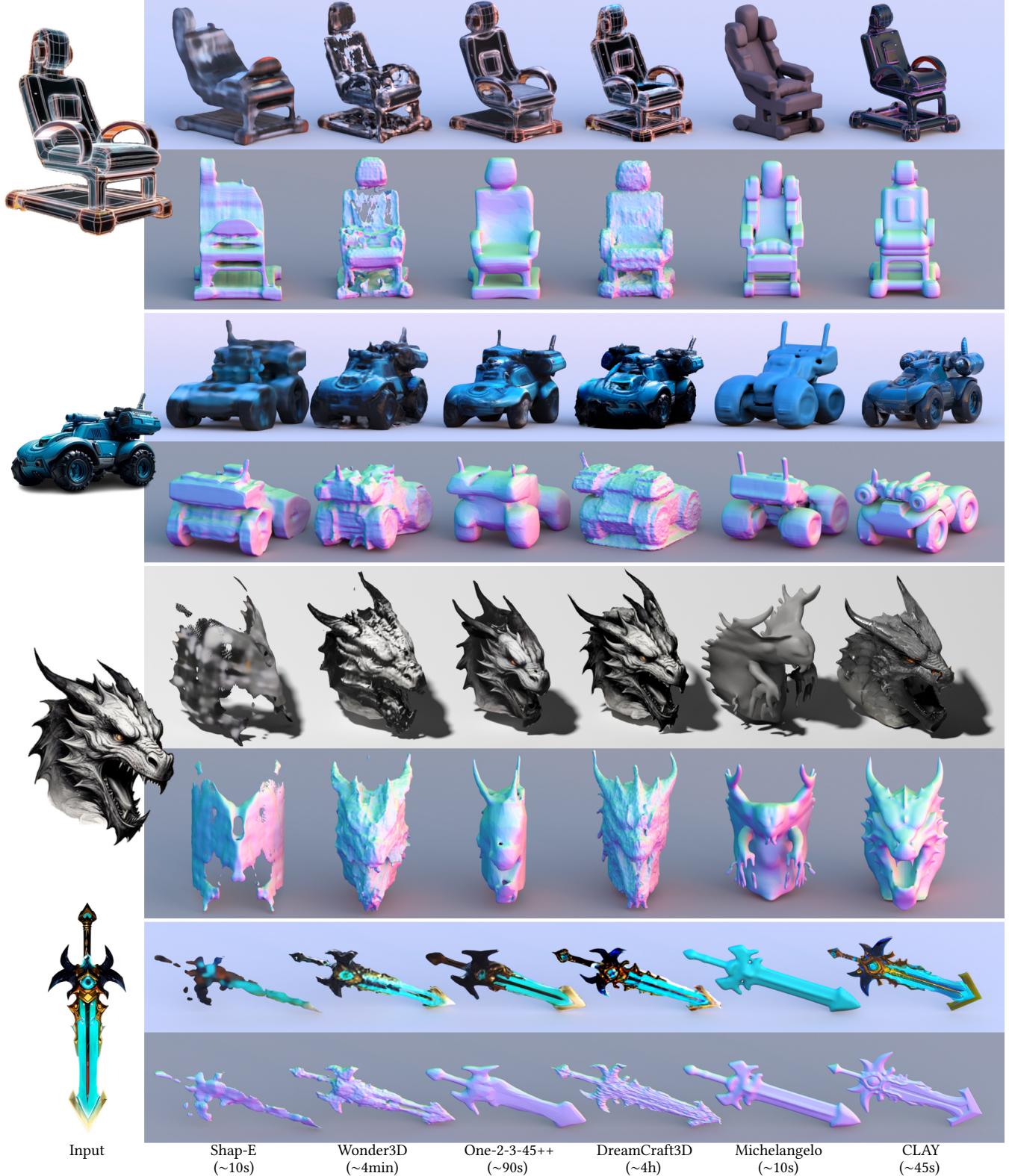


Fig. 15. Comparison with state-of-the-art methods on image-conditioned generation. Even without performing optimization using the target view, CLAY still generates high-quality and detailed geometries that faithfully resemble the input image, preserving essential geometric features, including straight lines and matching surface curvatures. Note that all input images are generated by Stable Diffusion. Colors of Michelangelo are manually set.



Fig. 16. Comparison of rendering results under two distinct lighting conditions. The light probes are displayed at the top-right corner. Our method showcases high-quality rendering with accurate specular highlights, whereas MVDream lacks matching highlights and RichDreamer misses view dependency by modeling highlights as fixed surface textures.

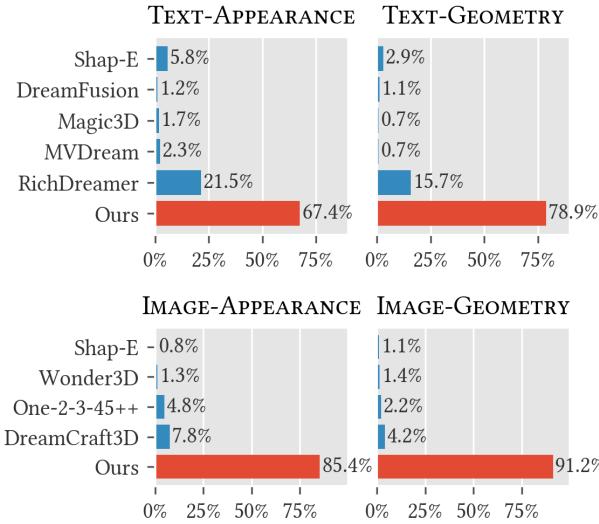


Fig. 17. User studies of CLAY vs. state-of-the-art methods indicates strong preferences of CLAY in generating both geometry and appearance.

users to effortlessly craft and manipulate digital geometry and textures, CLAY empowers both experts and novices alike to facilitate the seamless transformation of abstract concepts into detailed and realistic 3D models, expanding the horizons of digital artistry and design. At CLAY’s core is a large-scale generative framework enabled by a multi-resolution VAE and a DiT to accurately depict continuous surfaces and complex shapes. We have shown how to scale up CLAY efficiently through a progressive training scheme

to become a large 3D generative model. Its success is also largely attributed to our elaborately designed geometric data processing pipeline, including a standardized geometry remeshing protocol to ensure consistency in training, and the automatic annotation capabilities by GPT-4V. Comprehensive experimental evaluations and user studies have demonstrated CLAY’s efficacy and adaptability. Its high geometry quality, diversity in variety, and material richness position CLAY as one of the leading 3D generator in the field.

Ethics Statement. Same as 2D contents, 3D generation models have the potential to producing deceptive contents. Although we have implemented rigorous scrutiny processes for our training data, the utilization of pretrained feature encoders (CLIP [Radford et al. 2021] for text encoding and DINO [Oquab et al. 2024] for image encoding) in CLAY introduces a high-level of generalization capability that carries the risk of potential misuse. This means there is a possibility that our model could be used to generate virtual assets or scenes that violate regulations and propagate false information. We are committed to addressing these ethical issues, and along with the whole community, developing strategies to ensure the responsible use of CLAY.

Limitations and Future Work. It is important to note that CLAY is not yet complete end-to-end, as it entails distinct stages for generating geometry and materials, and requires additional steps such as remeshing and UV unwrapping. An immediate future step is to explore integrated model architectures to integrate geometry and PBR materials. This will require implementing automatic schemes to produce geometry with consistent topology. By far, CLAY has been trained on a substantially large dataset. However, there is still room for improvement in terms of both the quantity and quality of the training data, especially compared with 2D image datasets used to train Stable Diffusion. Further, we observe that CLAY shows robustness in generating assets composed of single objects but tends to be vulnerable when dealing with complex “composed objects”, such as “a tiger riding a motorcycle”, particularly with text-only inputs. The issue is largely attributed to insufficient training data of composed objects and the lack of detailed textual descriptions of these objects. The issue can potentially be mitigated through a text-to-image-to-3D workflow, akin to the approaches employed by Wonder3D [Long et al. 2024] and One-2-3-45++ [Liu et al. 2023d]. As the community augments the training dataset with a larger and more diverse collection of 3D shapes along with corresponding text descriptions, we expect CLAY as well as its concurrent works to reach a new level of geometry generation, in both quality and complexity. Finally, we intend to explore extends of CLAY to dynamic object generation. The generated results from CLAY indicate that it may be possible to semantically partition the geometry into meaningful parts, further facilitating motion and interaction, as in Singer et al. [2023] and Ling et al. [2024].

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China (2022YFF0902301), NSFC programs (61976138, 61977047), STCSM (2015F0203-000-06), and SHMEC (2019-01-07-00-01-E00003). We also acknowledge support from Shanghai Frontiers Science Center of

Human-centered Artificial Intelligence (ShangHAI) and MoE Key Lab of Intelligent Perception and Human-Machine Collaboration (ShanghaiTech University).

REFERENCES

- Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. 2023. MultiDiffusion: fusing diffusion paths for controlled image generation. , 16 pages.
- Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. 2023. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. arXiv:2311.15127 [cs.CV]
- Blender Online Community. 2024. Blender - a 3D modelling and rendering package. <http://www.blender.org>.
- Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. ShapeNet: An Information-Rich 3D Model Repository. arXiv:1512.03012 [cs.GR]
- Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023b. Text2Tex: Text-driven Texture Synthesis via Diffusion Models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 18512–18522. <https://doi.org/10.1109/ICCV51070.2023.01701>
- R. Chen, Y. Chen, N. Jiao, and K. Jia. 2023a. Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 22189–22199. <https://doi.org/10.1109/ICCV51070.2023.02033>
- Zilong Chen, Feng Wang, and Huaping Liu. 2024. Text-to-3D using Gaussian Splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5939–5948.
- Y. Cheng, H. Lee, S. Tulyakov, A. Schwing, and L. Gui. 2023. SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 4456–4465. <https://doi.org/10.1109/CVPR52729.2023.00433>
- Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII* 14. Springer, 628–644.
- M. Deitke, D. Schwenk, J. Salvador, L. Weihls, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsanit, A. Kembhavi, and A. Farhadi. 2023. Objaverse: A Universe of Annotated 3D Objects. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 13142–13153. <https://doi.org/10.1109/CVPR52729.2023.01263>
- Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 605–613.
- Andrea Gesmundo and Kaitlin Maile. 2023. Composable Function-preserving Expansions for Transformer Architectures. arXiv:2308.06103 [cs.LG]
- Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 2018. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 216–224.
- Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. 2023. threestudio: A unified framework for 3D content generation. <https://github.com/threestudio-project/threestudio>.
- Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 2023. 3DGen: Triplane Latent Diffusion for Textured Mesh Generation. arXiv:2303.05371 [cs.CV]
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6840–6851. https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d9671ab10179ca4b-Paper.pdf
- Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2024. LRM: Large Reconstruction Model for Single Image to 3D. In *The Twelfth International Conference on Learning Representations*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- Jingwei Huang, Hao Su, and Leonidas J. Guibas. 2018a. Robust Watertight Manifold Surface Generation Method for ShapeNet Models. arXiv:1802.01698 <http://arxiv.org/abs/1802.01698>
- Jingwei Huang, Yichao Zhou, and Leonidas Guibas. 2020. ManifoldPlus: A Robust and Scalable Watertight Manifold Surface Generation Method for Triangle Soups. arXiv:2005.11621 [cs.GR]
- Jingwei Huang, Yichao Zhou, Matthias Niessner, Jonathan Richard Shewchuk, and Leonidas J. Guibas. 2018b. QuadriFlow: A Scalable and Robust Method for Quadrangulation. *Computer Graphics Forum* 37 (2018). <https://doi.org/10.1111/cgf.13498>
- Yukun Huang, Jianan Wang, Yukai Shi, Boshi Tang, Xianbiao Qi, and Lei Zhang. 2024. DreamTime: An Improved Optimization Strategy for Diffusion-Guided 3D Generation. In *The Twelfth International Conference on Learning Representations*.
- Heewoo Jun and Alex Nichol. 2023. Shap-E: Generating Conditional 3D Implicit Functions. arXiv:2305.02463 [cs.CV]
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- Sixu Li, Chaojian Li, Wenbo Zhu, Boyang (Tony) Yu, Yang (Katie) Zhao, Cheng Wan, Haoran You, Huihong Shi, and Yingyan (Celine) Lin. 2023. Instant-3D: Instant Neural Radiance Field Training Towards On-Device AR/VR 3D Reconstruction. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (Orlando, FL, USA) (ISCA ’23)*. Association for Computing Machinery, New York, NY, USA, Article 6, 13 pages. <https://doi.org/10.1145/3579371.3589115>
- Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. 2024. SweetDreamer: Aligning Geometric Priors in 2D diffusion for Consistent Text-to-3D. In *The Twelfth International Conference on Learning Representations*.
- C. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M. Liu, and T. Lin. 2023. Magic3D: High-Resolution Text-to-3D Content Creation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 300–309. <https://doi.org/10.1109/CVPR52729.2023.00037>
- Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. 2024. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 5404–5411.
- Huan Ling, Seung Wool Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. 2024. Align Your Gaussians: Text-to-4D with Dynamic 3D Gaussians and Composed Diffusion Models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hanyang Chen, Chong Zeng, Jiayuan Gu, and Hao Su. 2024b. One-2-3-45++: Fast Single Image to 3D Objects with Consistent Multi-View Generation and 3D Diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. 2023d. One-2-3-45: Any Single Image to 3D Mesh in 45 Seconds without Per-Shape Optimization. In *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 22226–22246. https://proceedings.neurips.cc/paper_files/paper/2023/file/4683beb6bab325650db13af05d1a14a-Paper-Conference.pdf
- R. Liu, R. Wu, B. Van Hoornick, P. Tokmakov, S. Zakharov, and C. Vondrick. 2023c. Zero-1-to-3: Zero-shot One Image to 3D Object. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 9264–9275. <https://doi.org/10.1109/ICCV51070.2023.00853>
- Xian Liu, Jian Ren, Aliaksandr Siarohin, Ivan Skorokhodov, Yanyu Li, Dahu Lin, Xihui Liu, Ziwei Liu, and Sergey Tulyakov. 2023b. HyperHuman: Hyper-Realistic Human Generation with Latent Structural Diffusion. arXiv:2310.08579 [cs.CV]
- Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2024a. SyncDreamer: Generating Multiview-consistent Images from a Single-view Image. In *The Twelfth International Conference on Learning Representations*.
- Zexiang Liu, Yangguang Li, Youtian Lin, Xin Yu, Sida Peng, Yan-Pei Cao, Xiaojuan Qi, Xiaoshui Huang, Ding Liang, and Wanli Ouyang. 2023a. UniDream: Unifying Diffusion Priors for Relightable Text-to-3D Generation. arXiv:2312.08754 [cs.CV]
- Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuxin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. 2024. Wonder3D: Single Image to 3D using Cross-Domain Diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. 2022. SparseNeuS: Fast Generalizable Neural Surface Reconstruction from Sparse Views. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer Nature Switzerland, Cham, 210–227.
- Kleineberg Marian. 2021. mesh_to_sdf: Calculate signed distance fields for arbitrary meshes. https://github.com/marian42/mesh_to_sdf.
- Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4460–4470.
- G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or. 2023. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. In *2023 IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, Los Alamitos, CA, USA, 12663–12673. <https://doi.org/10.1109/CVPR52729.2023.01218>
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.
- Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. 2020. Polygon: An autoregressive generative model of 3d meshes. In *International conference on machine learning*. PMLR, 7220–7229.
- Alex Nichol, Heewoo Jun, Prafulla Dharwal, Pamela Mishkin, and Mark Chen. 2022. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. *arXiv:2212.08751 [cs.CV]*
- OpenAI. 2023. GPT-4V: Generative Pre-trained Transformer 4 for Vision. <https://www.openai.com/>.
- Maxime Oquab, Timothée Darzet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research* (2024).
- J. Park, F. Florence, J. Straub, R. Newcombe, and S. Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 165–174. <https://doi.org/10.1109/CVPR.2019.00025>
- Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. 2020. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III* 16. Springer, 523–540.
- Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T Barron, Amit H Bermano, Eric Ryan Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. 2023. State of the art on diffusion models for visual computing. *arXiv preprint arXiv:2310.07204* (2023).
- Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *arXiv:2307.01952 [cs.CV]*
- Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations*.
- Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: deep hierarchical feature learning on point sets in a metric space. 30 (2017), 5105–5114.
- Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. 2024. Magic123: One Image to High-Quality 3D Object Generation Using Both 2D and 3D Diffusion Priors. In *The Twelfth International Conference on Learning Representations*.
- Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. 2024. RichDreamer: A Generalizable Normal-Depth Diffusion Model for Detail Richness in Text-to-3D. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8821–8831. <http://proceedings.mlr.press/v139/ramesh21a.html>
- J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. 2021. Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 10881–10891. <https://doi.org/10.1109/ICCV48922.2021.01072>
- Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. 2024. XCube (X^3): Large-Scale 3D Generative Modeling using Sparse Voxel Hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. TEX-Ture: Text-Guided Texturing of 3D Shapes. In *ACM SIGGRAPH 2023 Conference Proceedings (Los Angeles, CA, USA) (SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 54, 11 pages. <https://doi.org/10.1145/3588432.3591503>
- R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 10674–10685. <https://doi.org/10.1109/CVPR52688.2022.01042>
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyr Ghaseimpour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 36479–36494. https://proceedings.neurips.cc/paper_files/paper/2022/file/ec795aeadae0b7d230fa35cbaf04c041-Paper-Conference.pdf
- Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Hyunsu Kim, Jaehoon Ko, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. 2024. Let 2D Diffusion Model Know 3D-Consistency for Robust Text-to-3D Generation. In *The Twelfth International Conference on Learning Representations*.
- Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. 2021. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.).
- Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. 2023. Zero123+: a Single Image to Consistent Multi-view Diffusion Base Model. *arXiv:2310.15110 [cs.CV]*
- Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. 2024. MV-Dream: Multi-view Diffusion for 3D Generation. In *The Twelfth International Conference on Learning Representations*.
- Yawar Siddiqui, Antonio Allegri, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2024. MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. 2023. Text-To-4D Dynamic Scene Generation. In *International Conference on Machine Learning, ICML 2023, 23–29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 31915–31929. <https://proceedings.mlr.press/v202/singer23a.html>
- Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. 2024. DreamCraft3D: Hierarchical 3D Generation with Bootstrapped Diffusion Prior. In *The Twelfth International Conference on Learning Representations*.
- Jiapeng Tang, Xiaoguang Han, Junyi Pan, Kui Jia, and Xin Tong. 2019. A skeletonon-bridged deep learning approach for generating meshes of complex topologies from single rgbs images. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 4541–4550.
- Jiapeng Tang, Xiaoguang Han, Mingkui Tan, Xin Tong, and Kui Jia. 2021a. Skeletonnet: A topology-preserving solution for learning mesh reconstruction of object surfaces from rgbs images. *IEEE transactions on pattern analysis and machine intelligence* 44, 10 (2021), 6454–6471.
- Jiapeng Tang, Jiaobiao Lei, Dan Xu, Feiyi Ma, Kui Jia, and Lei Zhang. 2021b. Sa-convolnet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6504–6513.
- Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. 2024. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. In *The Twelfth International Conference on Learning Representations*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dec91fb0d53c1c4a845aa-Paper.pdf
- Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021a. Neus: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *Advances in Neural Information Processing Systems* 34 (2021), 27171–27183.
- Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. 2024. PF-LRM: Pose-Free Large Reconstruction Model for Joint Pose and Shape Prediction. In *The Twelfth International Conference on Learning Representations*.
- Peng-Shuai Wang. 2022. mesh2sdf. <https://github.com/wang-ps/mesh2sdf>. Converts an input mesh to a signed distance field (SDF).
- Peng-Shuai Wang, Yang Liu, and Xin Tong. 2022. Dual octree graph networks for learning adaptive volumetric shape representations. *ACM Trans. Graph.* 41, 4, Article 103 (jul 2022), 15 pages. <https://doi.org/10.1145/3528223.3530087>
- X. Wang, L. Xie, C. Dong, and Y. Shan. 2021b. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. IEEE Computer Society, Los

- Alamitos, CA, USA, 1905–1914. <https://doi.org/10.1109/ICCVW54120.2021.00217>
- Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jinbo Wu, Xiaobo Gao, Xing Liu, Zhengyang Shen, Chen Zhao, Haocheng Feng, Jingtuo Liu, and Errui Ding. 2024. Hd-fusion: Detailed text-to-3d generation leveraging multiple noise estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3202–3211.
- T. Wu, J. Zhang, X. Fu, Y. Wang, J. Ren, L. Pan, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu. 2023. OmniObject3D: Large-Vocabulary 3D Object Dataset for Realistic Perception, Reconstruction and Generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 803–814. <https://doi.org/10.1109/CVPR52729.2023.00084>
- Rubin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*. PMLR, 10524–10533.
- Rui Xu, Zhiyang Dou, Ningna Wang, Shiqing Xin, Shuangmin Chen, Mingyan Jiang, Xiaohu Guo, Wenping Wang, and Changhe Tu. 2023. Globally Consistent Normal Orientation for Point Clouds by Regularizing the Winding-Number Field. *ACM Trans. Graph.* 42, 4, Article 111 (jul 2023), 15 pages. <https://doi.org/10.1145/3592129>
- Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. 2024. DMV3D: Denoising Multi-view Diffusion Using 3D Large Reconstruction Model. In *The Twelfth International Conference on Learning Representations*.
- Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. 2023. ULP-2: Towards Scalable Multimodal Pre-training for 3D Understanding. *arXiv:2305.08275 [cs.CV]*
- Lior Yariv, Omri Puny, Natalia Neverova, Oran Gafni, and Yaron Lipman. 2024. Mosaic-SDF for 3D Generative Models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. 2023. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. *arXiv:2308.06721 [cs.CV]*
- Fukun Yin, Xin Chen, Chi Zhang, Biao Jiang, Zibo Zhao, Jiayuan Fan, Gang Yu, Taihao Li, and Tao Chen. 2023. ShapeGPT: 3D Shape Generation with A Unified Multi-modal Language Model. *arXiv:2311.17618 [cs.CV]*
- Chaohui Yu, Qiang Zhou, Jingliang Li, Zhe Zhang, Zhibin Wang, and Fan Wang. 2023b. Points-to-3D: Bridging the Gap between Sparse Points and Shape-Controllable Text-to-3D Generation. In *Proceedings of the 31st ACM International Conference on Multimedia* (, Ottawa ON, Canada.) (MM '23). Association for Computing Machinery, New York, NY, USA, 6841–6850. <https://doi.org/10.1145/3581783.3612232>
- X. Yu, M. Xu, Y. Zhang, H. Liu, C. Ye, Y. Wu, Z. Yan, C. Zhu, Z. Xiong, T. Liang, G. Chen, S. Cui, and X. Han. 2023a. MVIImgNet: A Large-scale Dataset of Multi-view Images. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 9150–9161. <https://doi.org/10.1109/CVPR52729.2023.00883>
- Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 2023c. 3DShape2VecSet: A 3D Shape Representation for Neural Fields and Generative Diffusion Models. *ACM Trans. Graph.* 42, 4, Article 92 (jul 2023), 16 pages. <https://doi.org/10.1145/3592442>
- Longwen Zhang, Qiwie Qiu, Hongyang Lin, Qixuan Zhang, Cheng Shi, Wei Yang, Ye Shi, Sibei Yang, Lan Xu, and Jingyi Yu. 2023a. DreamFace: Progressive Generation of Animatable 3D Faces under Text Guidance. *ACM Trans. Graph.* 42, 4, Article 138 (jul 2023), 16 pages. <https://doi.org/10.1145/3592094>
- L. Zhang, A. Rao, and M. Agrawala. 2023b. Adding Conditional Control to Text-to-Image Diffusion Models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 3813–3824. <https://doi.org/10.1109/ICCV51070.2023.00355>
- Youjia Zhang, Junqing Yu, Zilkai Song, and Wei Yang. 2023d. Optimized View and Geometry Distillation from Multi-view Diffuser. *arXiv:2312.06198 [cs.CV]*
- Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. 2023. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems* (2023).
- Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. 2023. Locally Attentional SDF Diffusion for Controllable 3D Shape Generation. *ACM Trans. Graph.* 42, 4, Article 91 (jul 2023), 13 pages. <https://doi.org/10.1145/3592103>
- Junzhe Zhu, Peiyi Zhuang, and Sanmi Koyejo. 2024. HIFA: High-fidelity Text-to-3D Generation with Advanced Diffusion Guidance. In *The Twelfth International Conference on Learning Representations*.
- Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. 2024. Triplane Meets Gaussian Splatting: Fast and Generalizable Single-View 3D Reconstruction with Transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.