Computational Visual Media
https://doi.org/10.1007/s41095-0xx-xxxx-x

Review Article

Diffusion Models for 3D Generation: A Survey

Chen Wang[1], **Hao-Yang Peng**[2], **Ying-Tian Liu**[2], **Jiatao Gu**[3], **and Shi-Min Hu**[2](✉)

Abstract Denoising diffusion models have demonstrated tremendous success in modeling data distributions and synthesizing high-quality samples. In the 2D image domain, they have become the state-of-the-art and are capable of generating photo-realistic images with high controllability. More recently, researchers have begun to explore how to utilize diffusion models to generate 3D data, as doing so has more potential in real-world applications. This requires careful design choices in two key ways: identifying a suitable 3D representation and determining how to apply the diffusion process. In this survey, we provide the first comprehensive review of diffusion models for manipulating 3D content, including 3D generation, reconstruction, and 3D-aware image synthesis. We classify existing methods into three major categories: 2D space diffusion with pretrained models, 2D space diffusion without pretrained models, and 3D space diffusion. We also summarize popular datasets used for 3D generation with diffusion models. Along with this survey, we maintain a repository https://github.com/cwchenwang/awesome-3d-diffusion to track the latest relevant papers and codebases. Finally, we pose current challenges for diffusion models for 3D generation, and suggest future research directions.

Keywords Diffusion Models; 3D Generation; Generative Models; AIGC

1 Introduction

Human beings live in a 3D physical world. 3D data provide both the geometry and texture details of real-world objects and scenes, and contain much richer information than 2D images. The role of 3D digital assets is pivotal across a wide range of applications, from entertainment and gaming to the domains of virtual reality, robotics, architecture, and manufacturing. Although the development of 3D modeling technologies has made collecting and transmitting 3D assets much easier, their creation ab initio remains time-consuming and expensive. It also poses a great challenge for amateurs, since this process mandates extensive manual effort and prior experience. Consequently, techniques to generate 3D models with straightforward commands will undoubtedly benefit many people.

Generative models have greatly improved with deep learning and novel types of model, including variational autoencoders [1], generative adversarial networks [2], and normalizing flows [3]. Recently, denoising diffusion probabilistic models [4, 5] have recently become state-of-the-art generative models and have been widely applied to generate data of different forms, such as images, video, text, and voice. Notably, text-to-image diffusion models, including Stable Diffusion [5] and Imagen [6], can generate high-quality 2D images indistinguishable from real ones given prompts in natural language. However, 2D generation is still insufficient for real-world applications and researchers have made extensive efforts to develop 3D generative models with diffusion models.

Generating 3D data is inherently more challenging than generating 2D data. While 2D images are matrices of pixels that can be conveniently processed by modern neural networks, 3D representations have various forms, including explicit and implicit representations such as meshes, voxel grids, point clouds, and implicit functions [7]. Each representation has its own strengths and weaknesses, with no single representation being optimal. For example, implicit

1 Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pennsylvania, 19104, United States. E-mail: cw.chenwang@outlook.com.

2 Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: H.-Y. Peng, phy22@mails.tsinghua.edu.cn; Y.T. Liu, liuyingt23@mails.tsinghua.edu.cn; S.-M. Hu, shimin@tsinghua.edu.cn (✉).

3 Machine Learning Research, Apple AI/ML, New York, United States. Email: jiatao@apple.com.

Manuscript received: 2022-01-01; accepted: 2022-01-01

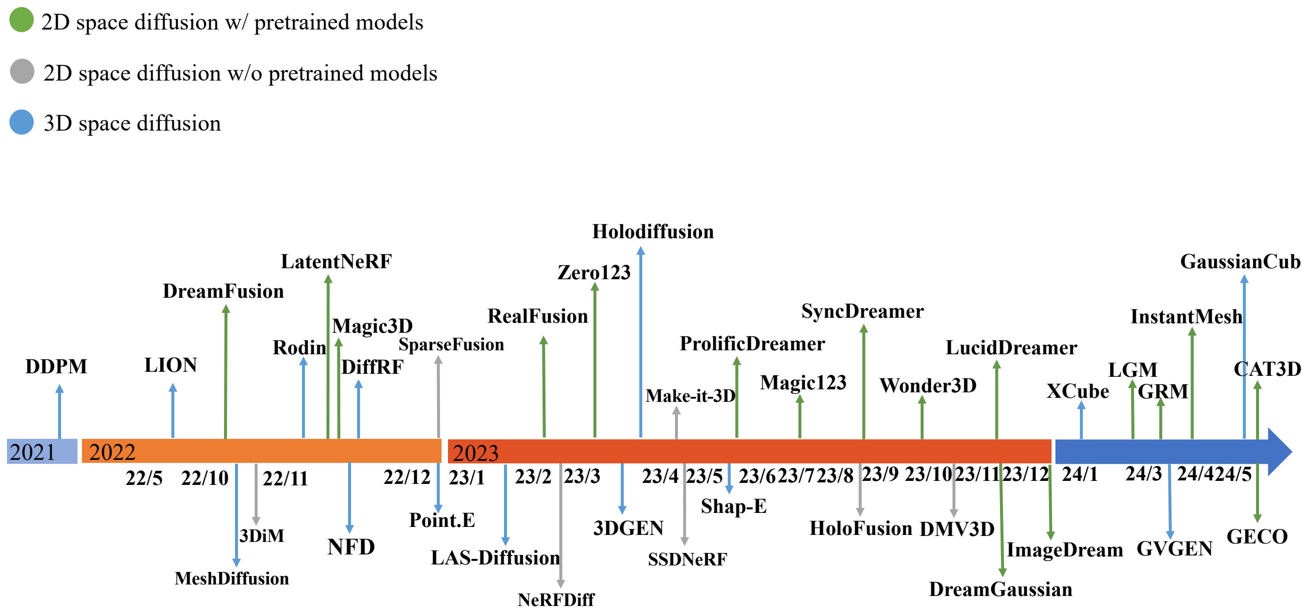


Fig. 1 A timeline of diffusion methods for 3D generation.

representations are easy to optimize but unsuitable for modern graphics pipelines. Critically, acquiring comprehensive real-world 3D datasets, essential for 3D prior learning, is far more difficult than capturing realistic images, posing a major challenge.

Given the popularity of 3D generation via diffusion models, this survey aims to provide a systematic review of recent progress in this field. The use of diffusion models by existing works can be classified from two aspects: what data the diffusion process (2D or 3D) operates on, and whether a pretrained diffusion model is used. This results in three categories of recent works: 2D space diffusion with pretrained models, 2D space diffusion without pretrained models and 3D space diffusion. It is important to note that they also have different requirements for input data. In methods performing 2D diffusion with pretrained diffusion models, a 3D model is learned by ensuring that its renderings lie in the distribution modeled by the pretrained model. They treat the diffusion model as a plug-in and do not perform any training of it, so no additional data is needed. Other approaches learn diffusion processes on 2D posed images and make them 3D-aware in different ways, so the synthesized novel views are still view-consistent. For diffusion using 3D representations, 3D raw data is needed and the most common approach is to directly convert the data into an intermediate representation, e.g., triplanes, on which the diffusion process is performed. As the choice of 3D representation is essential to diffusion learning of 3D spaces, we review this line of work according to the representation. A timeline of representative works for each category can be found in Fig. 1.

The contributions of this survey are:

- the first comprehensive review of diffusion models for 3D generation, covering up-to-date research,
- a classification of related methods according to the data the diffusion process operates on and whether a pretrained diffusion model is used, and
- suggestions for future research directions for 3D content generation by diffusion models.

Section 2 considers related surveys and clarifies the scope of our survey. Section 3 introduces the basic concepts of diffusion models and 3D representations. We summarize existing methods for 3D generation with 2D diffusion or 3D diffusion in Sections 4–6. Popular datasets used for 3D generation are summarized in Section 7. Section 8 presents existing challenges and provides suggestions for future research. Finally, Section 9 contains the conclusions drawn from our study.

2 Related Surveys

Recent surveys have provided comprehensive overviews of general diffusion models as well as 3D generation and reconstruction. In the former category, Yang et al. [8] summarize the theory of diffusion models and briefly introduce their applications to different fields. Zhang et al. [9] review methods that use text guided diffusion models for image generation and editing. Although Li et al. [10] provides a survey of 3D generation with diffusion, it only includes methods that optimize a scene-specific 3D representation. Turning to surveys related to 3D generation, Shi et al. [11] focuses on approaches that use generative models to directly model unconditional distributions and conditional distributions (conditioned on e.g., image, 3D or text inputs) of 3D data. Our survey includes methods that use diffusion models to manipulate 3D content that comes in both implicit and explicit representations. These mainly include methods that use diffusion models to assist the generation or editing of 3D data in a per-scene optimization manner, methods that infer 3D novel views with diffusion guidance, and methods that use diffusion models to learn the 3D data distribution from existing datasets.

3 Preliminaries

3.1 Diffusion Models

Probabilistic diffusion models are a class of generative models that convert simple known distributions (e.g., a Gaussian) into complex data distributions. They gradually perturb the input in the forward diffusion process with Gaussian noise and learn to estimate the perturbations through variational inference during the reverse process [4, 12]. Both the forward process and reverse process are parametrized using Markov Chains. In notation, given $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, the forward process q is a fixed Markov chain that adds Gaussian noise to $\mathbf{x}_0$ and generates latent variables $\mathbf{x}_1, \ldots, \mathbf{x}_T$ with the same dimension with a predetermined variance schedule $\beta_1, \ldots, \beta_T$:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

Ideally, the final latent variable $\mathbf{x}_T$ should be from a standard Gaussian distribution. Therefore, the reverse process starts denoising from $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ by learning the Gaussian transitions from $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \tag{2}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{3}$$

The training procedure aims to maximize the negative data log-likelihood. In DDPM [4], $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ is set to time-dependent constants and only the mean $\boldsymbol{\mu}$ in the reverse process is trainable. In practice, we use a trainable network (U-Net) to approximate the noise $\boldsymbol{\epsilon}$ added in the forward process through parametrization:

$$\mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\phi(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{4}$$

Please refer Ho et al. [4] for a complete explanation.

Diffusion probabilistic models are also called score-based generative models, so can be viewed from a stochastic differential equation (SDE) perspective [13]. The forward process is expressed as:

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\boldsymbol{\omega}_t \tag{5}$$

where $\boldsymbol{\omega}_t$ is a standard Wiener process, and $\mathrm{d}t$ is an infinitesimal negative timestep. The reverse process is also an SDE:

$$\mathrm{d}\mathbf{x}_t = [-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)]\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\bar{\boldsymbol{\omega}}_t \tag{6}$$

where $\bar{\boldsymbol{\omega}}_t$ is a standard Wiener process in which time flows backward from T to 0. $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ is known as the score function. We can estimate the score for all t using a neural network, allowing the reverse process to be determined. See [13] for more details.

Unless otherwise noted, in this paper, we use $\mathbf{x}_t$ to denote the sample with noise for diffusion models at timestep t, ϵ for the added noise and ϵ_ϕ for the noise predictor. Thus, $\mathbf{x}_t$ can be a rendered image, latent variable, or 3D shape, depending on the actual method.

3.2 3D Data Representations

The main traditional 3D data representations include point clouds, meshes and voxel grids. Point clouds are a collection of 3D point coordinates and their attributes (colors). Meshes represent 3D shapes by storing vertex positions and edge connections. Voxel grids can be seen as an extension of image pixels, with each point regularly distributed in 3D space. To save memory, sparsification techniques such as voxel hashing are used to prune empty voxels.

With the advance of deep learning, neural fields [14] have gained in popularity as a way of representing scene geometry and appearance. Fields refer to spatial-varying quantities and a neural field parameterizes a field in part or fully with a neural network. Chen et al. [15] proposed a unified framework to represent existing neural fields:

$$\mathbf{s}(\mathbf{x}) = \mathcal{P}\left(\prod_{i=1}^{N}\mathbf{f}_i(\gamma_i(\mathbf{x}))\right), \tag{7}$$

where $\gamma_i : \mathbb{R}^D \rightarrow \mathbb{R}^{F_i}$ is a coordinate transformation, $\mathbf{f}_i : \mathbb{R}^{F_i} \rightarrow \mathbb{R}^K$ are the factor fields (features for a coordinate), and $\mathcal{P} : \mathbb{R}^K \rightarrow \mathbb{R}^Q$ is a projection function. $\prod$ denotes the element-wise product of a sequence of factors. To derive the final observations, the projection function $\mathcal{P}$ may also perform post-processing steps. For example, in NeRF [7], the output signal provides per-point color and density ($D = 5$, $Q = 4$), which requires the volumetric rendering step to produce an image at given viewpoints. Commonly used neural field components in Eq. (7) are shown in Fig. 2. For 3D tasks, widely utilized neural fields and their formulations are listed in Table 1.

3D Gaussians [19] have appeared as a popular type of 3D representation since they can provide high-quality, high-speed rendering. 3D Gaussians represent 3D scenes as a set of Gaussians that contain attributes of position, color, scale and

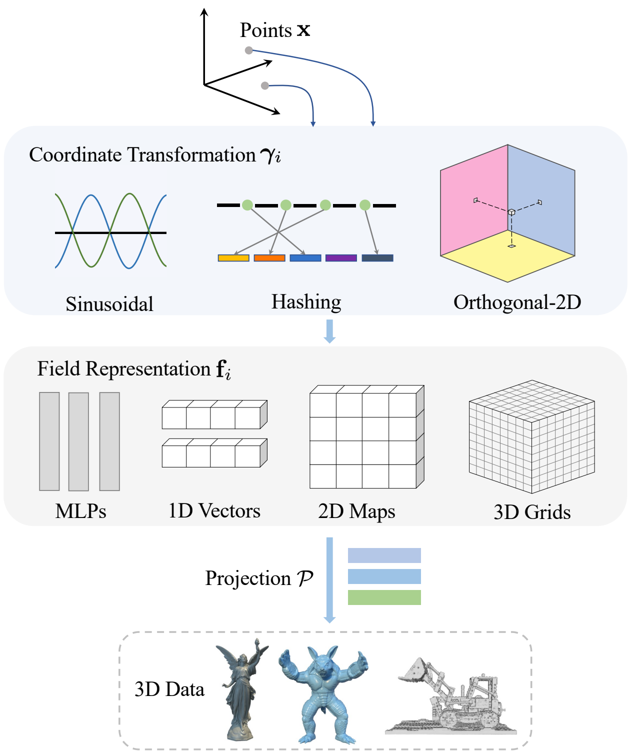


Fig. 2 Common neural field components, following [15].

Table 1 Common Neural Field 3D Representations

Name	N	$\gamma_i(\mathbf{x})$	$\mathbf{f}_i(\mathbf{x})$	$\mathcal{P}(\mathbf{x})$
NeRF [7]	1	Sinusoidal($\mathbf{x}$)	$\mathbf{x}$	MLP($\mathbf{x}$)
iNGP [16]	1	Hashing($\mathbf{x}$)	Vectors($\mathbf{x}$)	MLP($\mathbf{x}$)
Triplane [17]	1	Orthogonal-2D($\mathbf{x}$)	2D-Maps($\mathbf{x}$)	MLP($\mathbf{x}$)
Plenoxels [18]	1	$\mathbf{x}$	3D-Grids($\mathbf{x}$)	$\mathbf{x}$ or SH($\mathbf{x}$)

opacity, which can be rasterized into images and optimized with rendering loss.

4 Diffusion in 2D Space with Pre-trained Models

Pretrained text-to-image diffusion models are powerful enough to generate photorealistic 2D images from text input. Researchers have leveraged this capability for various 3D generation tasks with score distillation techniques. We show results from representative works in Fig. 4 later.

4.1 Preliminary: Score Distillation Sampling

Dreamfusion [20] learns a 3D scene from 2D pre-trained text-to-image diffusion models. Given a datapoint $\mathbf{x} = g(\theta)$ generated by a differentiable generator g with parameters θ, Dreamfusion [20] adds Gaussian noise of level t and turns it into $\mathbf{x}_t$. It then uses a pre-trained diffusion model with denoising function $\epsilon_\phi(\mathbf{x}_t; y, t)$ to predict the noise with text embedding y to update θ. The proposed *score distillation sampling* (SDS) is written as:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, g(\theta)) = \mathbb{E}_{t,\epsilon}\left[w(t)(\hat{\epsilon}_\phi(\mathbf{x}_t;y,t)-\epsilon)\frac{\partial \mathbf{x}}{\partial \theta}\right] \tag{8}$$

$$\hat{\epsilon}_\phi(\mathbf{x}_t;y,t) = (1+w_{\text{g}})\epsilon_\phi(\mathbf{x}_t,y,t) - w_{\text{g}}\epsilon_\phi(\mathbf{x}_t,t) \tag{9}$$

where $w(t)$ is a weighting function, w_{g} is the guidance scale between unconditional and conditional generation. For 3D tasks, θ is the neural field, mostly represented by a multi-layer perceptron, $\mathbf{x}$ is a rendered image (or latent image) given a random camera viewpoint, and g represents the volume rendering process. SDS loss can also be written as:

$$\mathcal{L}_{\text{SDS}} = \mathbb{E}_{t,\epsilon}\left[||\mathbf{x}-\hat{\mathbf{x}}_0||^2\right] \tag{10}$$

where $\mathbf{x}$ and $\mathbf{x}_0$ are the rendered image and denoised image respectively. Therefore, optimizing SDS encourages the renderings of the neural field to be similar to the generated 2D images of diffusion models given a text condition t. SJC [21] arrives at the same training objective from the perspective of estimating the scores of 3D data with 2D scores. As the pseudo-ground-truth of SDS is stochastic, LucidDreamer [22] uses interval score matching that applies DDIM inversion and DDIM sampling to the 3D renderings, for more accurate supervision.

VSD [23] further models the 3D representations to be learned as a distribution and aligns its samples with the pretrained diffusion model by solving a variational inference problem. The final loss function used by VSD is:

$$\begin{aligned}&\nabla_\theta \mathcal{L}_{\text{VSD}}(\phi, g(\theta)) = \\ &\mathbb{E}_{t,\epsilon,c}\left[w(t)(\hat{\epsilon}_\phi(\mathbf{x}_t;y,t)-\epsilon_\tau(\mathbf{x}_t;y,t,c))\frac{\partial \mathbf{x}}{\partial \theta}\right]\end{aligned} \tag{11}$$

where ϵ_τ estimates the score of noisy rendered images, trained with a standard diffusion objective, and c denotes the camera parameters used for rendering. Note that both SDS and VSD are typically applied to 2D images since 2D pretrained diffusion models are well-established and accurate in estimating 2D scores. However, they are general distillation methods and can be directly applied to 3D representations given 3D pretrained models, in which case $\mathbf{x}$ would be some 3D data.

4.2 Text-to-3D Generation

4.2.1 Object Level

Dreamfusion [20] and SJC [21] were first to achieve text-guided 3D generation by using SDS to optimize MipNeRF [24] and a voxel grid respectively. Most following works employ a similar framework to that shown in Figure 3, which updates a 3D representation using pretrained diffusion models and improves the generation quality in various ways, including the choice of 3D representation, the sampling of

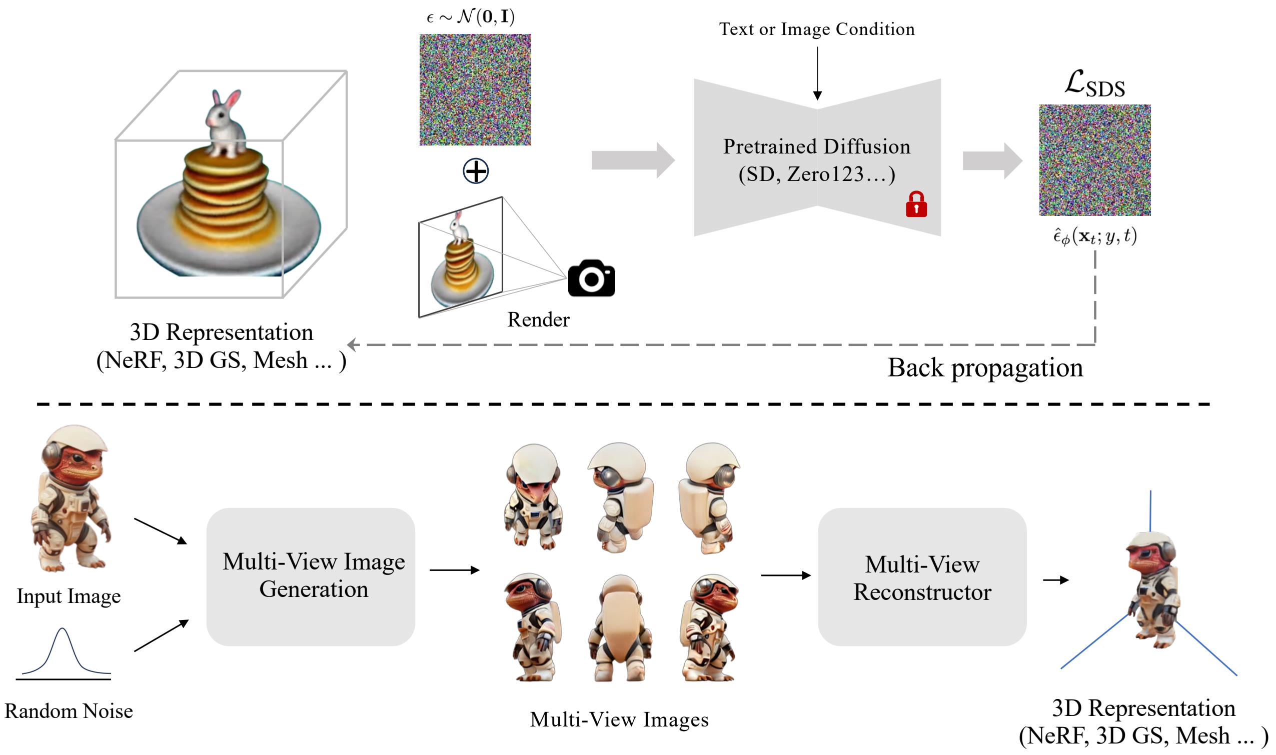


Fig. 3 (Above) Distilling 3D models from 2D pretrained diffusion models. In each iteration, random Gaussian noise is added to the image rendering of the 3D scene and a conditional pretrained diffusion model denoises the noisy image. The difference between the added noise and estimated noise is used to calculate the SDS loss for gradient backpropagation. (Below) Recent works have fine-tuned pretrained diffusion models to generate multi-view images which can then be directly reconstructed into 3D representations.

diffusion models and the choice of diffusion model. DreamGaussian [25] speeds up SDS optimization into minutes by using 3D Gaussians.

Magic3D [26] generates high resolution 3D content in a coarse-to-fine manner with a latent diffusion model. It first optimizes an Instant-NGP [16] model with SDS in low-resolution image space. Then it extracts a textured mesh from it and further fine-tunes the mesh with a high-resolution latent diffusion model using SDS again.

TextMesh [27] aims to generate high-quality 3D content in mesh representation. It directly optimizes an SDF neural field with SDS to allow easy extraction of meshes. Further, it uses another diffusion model conditioned on the renderings from the mesh to re-texture the mesh with another SDS optimization stage.

Latent-NeRF [28] also performs SDS in the latent space. Then it fine-tunes a shallow MLP encoder with SDS to turn latent information into an image. Latent-NeRF also allows guiding the generation process with coarse shape initialization or colorizing a given mesh by optimizing its u-v map with SDS.

Fantasia-3D [29] generates disentangled geometry and appearance of 3D objects. It first represents the geometry with DMTet and optimizes its parameters with SDS by using the rendered normal map as the input to the pre-trained diffusion model. Then, it optimizes the appearance of the object by predicting the material parameters of the bidirectional reflectance distribution function (BRDF) using another round of image-based SDS.

DITTO-NeRF [30] firstly generates a point cloud for the input image by depth estimation methods. It divides viewing angles into inside-boundary (IB) and outside-boundary (OB) according to whether the point cloud can be projected to the view. It enforces image rendering loss in the IB regions and uses SDS based on a pre-trained inpainting diffusion model to generate the OB regions. Finally, a refinement step is performed to ensure consistency of IB and OB regions.

3DFuse [31] makes the SDS process 3D-aware by injecting depth information into the pre-trained diffusion model. Given a text prompt, it first generates a 2D image and optimizes the text embedding e. Then the depth map of the image, predicted by an off-the-shelf estimator, is injected into the diffusion U-Net through feature addition. It also fine-tunes the diffusion model with additional LoRA layers to adapt it to the embedding e and maintain semantic consistency.

As SDS approaches based on latent diffusion models operate at a limited 64×64 resolution, Liao et al. [32] present a generic approach to achieve more detailed guidance: besides the SDS loss, it aligns the features of the input latent information and the predicted latent information by inputting them into the UNet decoder of Stable Diffusion [5] and computing the difference between the multi-level features. It also uses KL loss to keep the optimized latent information close to the prior distribution during training.

Perp-Neg [33] aims to overcome the *multi-face Janus problem* of text-to-3D generation: the Janus problem refers to the phenomenon that the generated 3D content may show the canonical view from multiple viewpoints, *i.e.,* resulting in more than one face. Perp-Neg resolves this problem by making the 2D diffusion model generate images more conforming to the view angles. It generates each view with different positive and negative prompts by making Equation (9) view-dependent:

$$\hat{\epsilon}_\phi(\mathbf{x}_t; y, v, t) = \epsilon_\phi(\mathbf{x}_t, t) + w_\mathrm{g}[\epsilon_\phi^{\mathrm{pos}_v} - \sum_i w_v^i \epsilon_\phi^{\mathrm{neg}_v^{(i)\perp}}] \tag{12}$$

$$\epsilon_\phi^{\mathrm{pos}_v} = \epsilon_\phi(\mathbf{x}_t, t, y_{\mathrm{pos},v}) - \epsilon_\phi(\mathbf{x}_t, t) \tag{13}$$

$$\epsilon_\phi^{\mathrm{neg}_v^{(i)}} = \epsilon_\phi(\mathbf{x}_t, t, y_{\mathrm{neg}_v^{(i)},v}) - \epsilon_\phi(\mathbf{x}_t, t) \tag{14}$$

where y_v refers to the positive/negative text embedding for view direction v, $\epsilon_\phi^{\mathrm{neg}_v^{(i)\perp}}$ is the perpendicular component of $\epsilon_\phi^{\mathrm{neg}_v^{(i)}}$ on $\epsilon_\phi^{\mathrm{pos}_v}$. The perpendicular gradient prevents the negative prompt from influencing the semantics of the positive prompt and makes the generation better conditioned on the prompts.

HiFA [34] rewrites Equation (8) to:

$$\nabla_\theta \mathcal{L}_\mathrm{SDS}(\phi, g(\theta)) = w(t) \frac{\sqrt{\bar{\alpha}_t}}{2\sqrt{1-\bar{\alpha}_t}} (\mathbf{x} - \hat{\mathbf{x}}_\mathrm{1step}) \frac{\partial \mathbf{z}}{\partial \theta} \tag{15}$$

$$\hat{\mathbf{x}}_\mathrm{1step} = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \epsilon_\phi(\mathbf{x}_t; y, t)) \tag{16}$$

where $\mathbf{x}$ is the latent image. Standard SDS directly compares the rendered latent image to the latent image predicted by the diffusion model with one-step inference. HiFA replaces $\hat{\mathbf{x}}_\mathrm{1step}$ by more accurate step-by-step estimation. Furthermore, it supervises the rendered depth using a pre-trained depth estimation model and regularizes the distribution of NeRF weights for each ray to generate a crisp surface. HifA and DreamTime [35] also study the choice of t during optimization, opting for a large t during the initial training iterations and gradually reducing it to capture fine details.

Building upon previous works, ATT3D [36] was the first to achieve 3D generation with pre-trained diffusion without per-scene optimization. It trains a mapping network to map prompts to NeRFs, allowing the training of a set of prompts collectively. Latte3D [37] further scales up ATT3D and achieves much better quality.

4.2.2 *Scene Level*

To generate 3D scenes with pre-trained diffusion models, one line of work generates a proportion of the scene, then iteratively extends it by inpainting using novel viewpoints with the help of depth information. The popular underlying 3D representations include meshes and neural fields.

SceneScape [38] generates zoom-out trajectories for a 3D scene from text prompts. It represents the scene using a unified mesh and outpaints the mesh iteratively. At each step, a new frame with depth is projected from the mesh and completed using a pre-trained text-to-image diffusion model. It leverages a pre-trained model to predict the depth of the generated image and further fine-tunes the depth model for consistent geometry by encouraging the predictions to be consistent in the projected regions. Finally, the mesh is updated using the predicted depth map.

Text2Room [39] generates a mesh-based 360° 3D scene based on pre-trained text-to-image diffusion models. It has a similar spirit to SceneScape [38], because it also maintains a global mesh and samples at predefined poses to generate the whole scene by completing RGBD renderings step-by-step. The difference is that Text2Room performs depth alignment and mesh filtering to obtain an optimal next mesh patch for each pose. It also samples additional poses to fill in the remaining unobserved regions.

Given a text prompt, Text2NeRF [40] synthesizes an initial view and estimates its depth with a pre-trained diffusion model and depth estimation model. Then it warps the initial view to other viewpoints to initialize a NeRF scene. Then it renders from novel viewpoints and adopts diffusion models to complete the missing regions. Using a depth alignment step, the newly completed image is added to NeRF training.

For panoramas, PanoGen [41] generates 360° indoor scenes with recursive outpainting over a single image generated from the text caption. MVDiffusion [42] fine-tunes Stable Diffusion by adding attention layers to generate consistent images across views and stitch them to panoramas.

Another line of work creates 3D scenes in a compositional manner, *i.e.,* generating each object separately and then blending them into a scene. Given user-specified 3D bounding boxes each denoting the location and size of an object, Po and Wetzstein [43] render both images and segmentation maps. The optimization is still using SDS loss, but the denoising steps of each semantic region in an image are based on the text prompt of the corresponding object. Set-the-Scene [44] and

CompoNeRF [45] also generates 3D scenes from 3D bounding boxes, but they have both object-level and scene-level neural fields. During the optimization, they either optimize each object individually or optimize the whole scene with SDS.

4.3 Image-to-3D Generation

Pre-trained diffusion models contain 3D knowledge inherently since they can generate images from various viewpoints. By exploiting the 3D priors in them, existing works can reconstruct an object from only one view or a few views.

Neural-Lift360 [46] and NeRDi [47] lift a single in-the-wild image to 360° views with diffusion guidance. They use a pre-trained Stable Diffusion [5] to denoise the NeRF renderings to ensure the generation is aligned with the input image. They also incorporate relative depth ranking information from pre-trained monocular depth estimation to regularize the geometry of radiance fields. Similarly, RealFusion [48] achieves 360° mesh reconstruction from a single image using a pre-trained diffusion model. It adapts the diffusion prior to the input image by using textual inversion [49] on the augmented images of the input. Given the customized diffusion model, a coarse-to-fine NeRF is optimized with SDS and smooth normal objectives.

To make the synthesized novel views more faithful to the given view, Make-it-3D [50] proposes a two-stage optimization pipeline. The first stage extends NeRDi [47] with an additional CLIP loss to force the rendered image to look more like the input. The second stage builds a point cloud and textures visible points using reference images, and invisible points from the first stage NeRF, with a learnable deferred renderer.

DreamSparse [51] achieves view synthesis from sparse views with a pre-trained diffusion model. It extracts geometry features from input views with a 3D geometry module and learns a spatial guidance model to condition the pre-trained diffusion model with the extracted features. In this way, the synthesized images from the pre-trained models are view-consistent with the input object. Dreambooth3D [52] lifts a set of casually captured images of an object to 3D without camera poses. Since DreamBooth tends to overfit input views, naively combining it with SDS leads to inconsistent 3D models. Therefore, DreamBooth3D first *partially* fine-tunes a DreamBooth and uses SDS to optimize a 3D-consistent but not subject-specific NeRF. Then the renderings of the NeRF are translated to detailed multi-view subject images using a fully-trained DreamBooth model. Those images are used to further fine-tune the partial DreamBooth into a multi-view DreamBooth for final SDS optimization.

Instead of using frozen models, Zero123 [53] constructs a synthetic dataset containing paired images and their relative camera parameters to fine-tune a pre-trained Stable Diffusion. The training objective is to synthesize one image using the other image and relative poses as the denoising condition. Once trained, the model can generate new images of the same object under a given camera transformation. Once trained, Zero123 can be used as a pre-trained model for image-to-3d generation with SDS. Magic123 [54] utilizes Zero123 (precise geometry but oversimplified texture) as the 3D prior and Stable Diffusion (detailed texture but imprecise geometry) as the 2D prior for 3D reconstruction from a single image. Zero123 can also generate multi-view images of an object to assist the training of a generalized single image reconstruction model [55].

Since Zero123 [53] generates images in different poses separately, results for the same object are inconsistent in 3D. Following work [56–61] fine tunes Stable Diffusion to synthesize multi-view images at the same time and model their connections with attention mechanisms. The synthesized multi-view images can be fused into 3D representations such as 3D Gaussians and meshes using reconstruction models [62–64]. In a recent development, GECO [65] distills the multi-view diffusion model into one-step using VSD [23] and trains a feedforward 3D generative model that can naturally handles the back-view of the input image. CAT3D [66] generates a large set of synthetic views from a multi-view latent diffusion model conditioned on the input views, and directly trains a NeRF on those views. Benefiting from large-scale training, the model also works well on images with backgrounds.

4.4 3D Human and Animal Generation

For human and animal generation, shape priors such as parametric human models can be incorporated into the optimization process to ensure reasonable geometry.

DreamFace [67] generates personalized 3D faces using text guidance. It first selects a coarse geometry from the shape space of a parametric model called ICT-FaceKit [68]. To achieve detailed geometry, it renders the coarse mesh with vertex displacements and normal maps which are learned with SDS. In the texture generation step, it again uses SDS on both the latent space and image space similar to Latent-NeRF [28].

To generate human avatars from text prompts, parametric models such as SMPL provide an ideal geometry initialization for the 3D representation. AvatarCraft [69] optimizes a template avatar initialized from the SMPL model using SDS, but introduces a pixel-level silhouette loss to avoid SDS changing the geometry greatly. DreamAvatar [70] utilizes two SDSs to optimize a canonical template avatar and an observation avatar jointly; the former is obtained by deforming the latter. The canonical pose minimizes self-occlusion and is easy to generate. For 3D consistent SDS, DreamWaltz [71] extracts a body skeleton from SMPL to replace the pretrained diffusion model with a skeleton-conditioned ControlNet. TADA [72] further improves the text-to-human optimization pipeline by optimizing geometry and texture simultaneously and introduces animations throughout the optimization process to make the generated avatar semantically consistent with SMPL-X so that it can be easily animated. As an alternative to SMPL, implicit statistic models like imGRUM [73] are more compatible with NeRF and are also utilized with SDS for 3D human body generation and animation.

In addition to generating avatars from text prompts, other works create them from input images. Given a single human image, ZeroAvatar [74] first estimates a SMPL mesh and u-v map. The recovered mesh is used in two ways: to initialize the density field of NeRF and to render the depth at novel views. The final loss terms include depth-guided SDS, RGB loss from the u-v map, and depth correlation loss. AvatarBooth [75] creates personalized avatars from casually captured face or body images. It optimizes NeuS with SDS with fine-tuned Stable Diffusion models on the input images.

Pretrained diffusion models can also assist in the single-view reconstruction of articulated objects. Farm3D [76] learns an articulated category-level model using only virtual data generated by Stable Diffusion. It encodes an input image into an articulated shape, appearance, viewpoint, and light direction with a single-forward pass. The encoder is learned using both SDS loss on sampled virtual views and reconstruction loss on the input view. ARTIC3D [77] aims to achieve the same goal with sparse web images of an animal species instead. However, it calculates pixel-level gradients with Stable Diffusion. Specifically, the latent image of a rendered image I is updated using multiple steps with score distillation and then decoded to I'. Pixel-level L2 loss between I and I' is utilized to update the reconstruction module.

4.5 3D Editing

Instruct-NeRF2NeRF [78] and Instruct-3D-to-3D [79] edit a trained NeRF scene with an image-conditioned Instruct-Pix2Pix [80] diffusion model. The former edits the NeRF renderings with InstructPix2Pix given a text instruction and then uses them as supervision signals to optimize the NeRF (known as *dataset update*). By repeating the procedure, the original NeRF scene is gradually aligned with the prompt. The latter contains a frozen NeRF model and a target model. Both models render the same viewpoint and the rendering from the frozen model is edited with InstructPix2Pix. Finally, the edited image and the rendering from the target model are used to update the target NeRF with SDS. Edit-DiffNeRF [81] also edits NeRFs with Instruct-Pix2Pix, but fine-tunes the diffusion model in the target scene for more accurate changes and better semantic consistency. Control4D [82] applies dataset update for dynamic scene editing and trains a discriminator to mitigate the issue of inconsistent supervision arising from the edited dataset.

Other works use the Stable Diffusion model for editing tasks. RePaint-NeRF [83] first trains a CLIP feature field in NeRF to select the target object and then uses a text prompt to edit the selected region with SDS. DreamEditor [84] automatically locates the regions to be edited by using the fact that the attention maps in the pre-trained diffusion model reflect the relationship between each keyword and a pixel in the generated image. SDS is only performed in the editing region for precise editing. FocalDreamer [85] allows adding independent and reusable 3D parts to existing 3D models. It optimizes the added parts in selected regions with SDS by feeding the renderings of the whole object to pre-trained diffusion models. Style and geometric consistency losses are applied to ensure localized change and congruent overall appearance.

5 Diffusion in 2D Space for View Synthesis

5.1 View Synthesis of 3D Objects

To synthesize novel views of 3D objects, current works learn to make the diffusion process 3D-aware by exploiting the cross-view relationships in multi-view data, or using the inductive bias of 3D representations, such as NeRF.

3DiM [87] learns to denoise a Gaussian-noised target view by conditioning the 2D diffusion model with an input view and relative camera pose. It also uses stochastic conditioning at inference time for better 3D consistency. Similarly, Chan et al. [88] also conditions a 2D diffusion model on the input image and the relative camera pose. However, it incorporates geometry priors by concatenating the input with a pixel-aligned feature image that is created by warping input image features to the target view for 3D consistency.

NeRFDiff [89] jointly trains a triplane-based Pixel-NeRF [90] with 3D-aware conditional diffusion to model the uncertainly of single-image view synthesis. The diffusion process learns to denoise at the target viewpoint given Pixel-NeRF rendering as the condition. It also uses NeRF-guided distillation to alternately update the NeRF representation and guide the multi-view diffusion process.

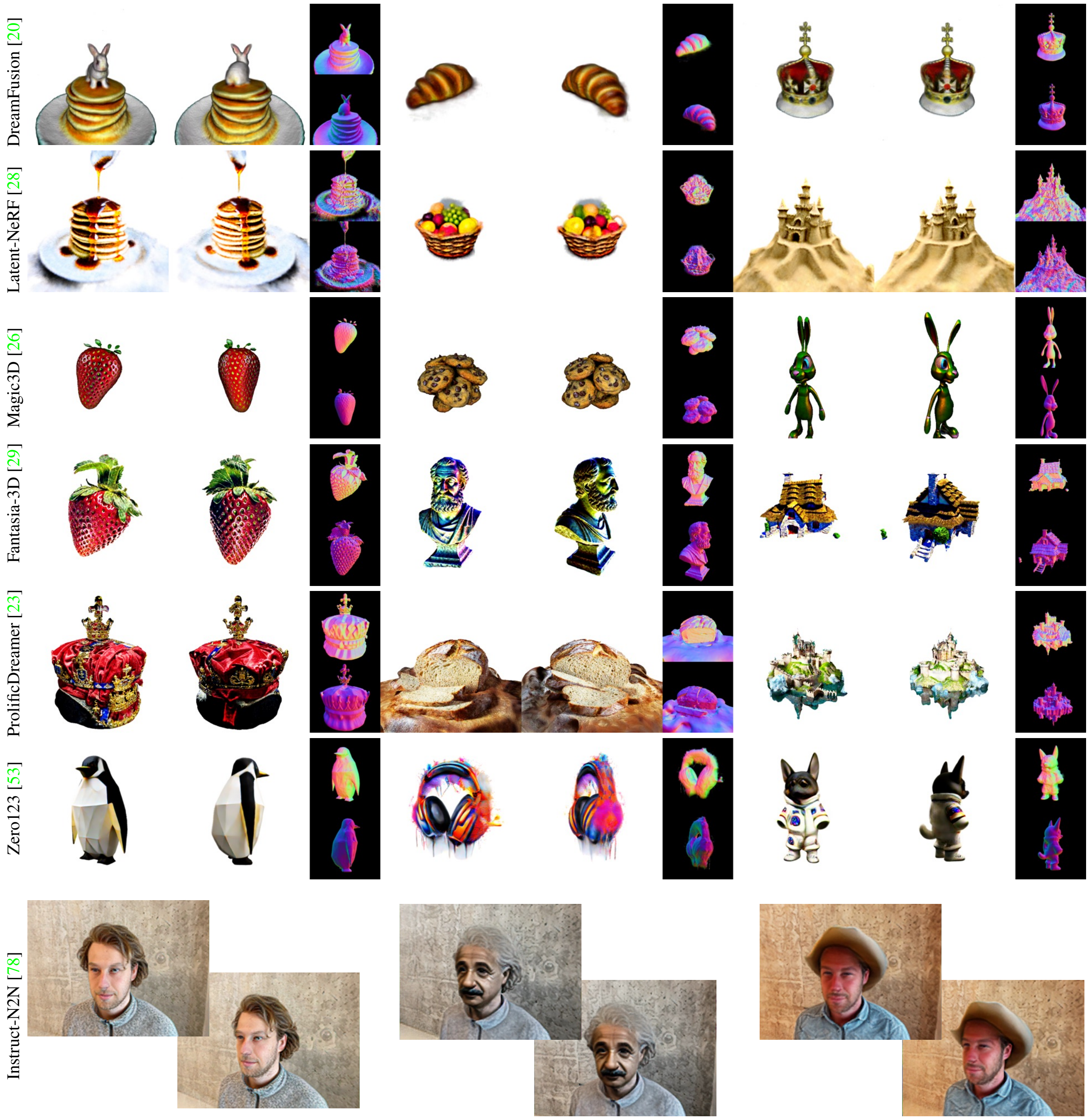


Fig. 4 A gallery of 3D generation results from different categories, obtained with threestudio [86]. Please refer to the Github Repo for up-to-date results.

3DDesigner [91] also consists of a NeRF module and diffusion module. It concatenates the noised image with a coarse rendering from NeRF as the conditional information. It jointly denoises two images from different viewpoints to enhance multi-view consistency and computes cross-view feature interactions in attention blocks.

SparseFusion [92] first learns a diffusion model on the features extracted from an epipolar transformer to model the distribution of $p(\mathbf{x}|\boldsymbol{\pi}, C)$, where $\mathbf{x}$ is the 2D image, $\boldsymbol{\pi}$ is the target pose, and C denotes the input views and poses. To sample from this distribution, it distills a NeRF by encouraging the NeRF rendering $g_\theta(\boldsymbol{\pi})$ to be close to denoised images $\hat{\mathbf{x}}_T$:

$$\mathcal{L}_{\text{distillation}} = \mathbb{E}_{\boldsymbol{\pi},\epsilon,t}[w_t \| g_\theta(\boldsymbol{\pi}) - \hat{\mathbf{x}}_T \|]$$

RenderDiffusion [93] makes the denoiser 3D-aware to introduce inductive bias for 3D generation with only single-view 2D data. It replaces the popular UNet [94] denoiser by a latent 3D structure, consisting of a triplane encoder that transforms a single noisy image into a triplane, and a triplane volume renderer that renders it back to a denoised 2D image for supervision. Similarly, Tewari et al. [95] also learn to generate novel views by denoising one image, but train on multi-view datasets and condition the denoising process with renderings using PixelNeRF. ViewsetDiffusion [96] further jointly denoises multiple noisy images with multi-view aggregation given any number of clean images for conditioning, allowing for sampling of 3D reconstructions. The denoising function targets reconstruction and rendering of a 3D volume. DMV3D [97] further scales up RenderDiffusion [93] to highly diverse datasets with the LRM [98] 3D denoiser architecture.

Xiang et al. [99] directly train a 2D diffusion model on ImageNet [100]. They model the distribution of 3D scenes $p(\mathbf{x}_{\text{3d}})$ as the joint distribution of their multiview renderings:

$$\begin{aligned} p(\mathbf{x}_{\text{3d}}) &= p(\mathbf{x}_{\boldsymbol{\pi}_0}, \mathbf{x}_{\boldsymbol{\pi}_1}, \ldots, \mathbf{x}_{\boldsymbol{\pi}_N}) \\ &= p(\mathbf{x}_{\boldsymbol{\pi}_0}) \cdot p(\mathbf{x}_{\boldsymbol{\pi}_1} | \mathbf{x}_{\boldsymbol{\pi}_0}) \cdot \cdots \cdot p(\mathbf{x}_{\boldsymbol{\pi}_N} | \mathbf{x}_{\boldsymbol{\pi}_0}, \ldots, \mathbf{x}_{\boldsymbol{\pi}_{N-1}}) \end{aligned}$$

Then, they learn an unconditional diffusion model to generate the first view and a conditional diffusion model with previous views as the condition to synthesize novel views. To train without multi-view data, they replace the condition image by the forward-backward depth-warped target view.

5.2 View Synthesis of 3D Scenes

Tseng et al. [101] train an image diffusion model to synthesize a long-term video of novel views from a single image. The model takes the source view image and camera poses as the condition and denoises the image from the target viewpoint. It adds an epipolar attention layer after each self-attention layer in the UNet denoiser. Therefore, the denoising process is augmented by the epipolar features linking source and target views. They also use stochastic conditioning and fixed noise in the backward process to reduce flicker.

Also dealing with view synthesis from a single image, Yu et al. [102] propose a two-stream architecture using two U-Nets with shared weights to process the novel view and the conditioning view. The two networks interact with each other through cross-attention layers, which are inserted after every spatial attention layer. They also incorporate camera pose information into the queries and keys of the attention layers.

DiffDreamer [103] uses diffusion models for view synthesis of a long camera trajectory with only internet-collected images of nature scenes. It creates training pairs by projecting the ground truth RGBD image $(I_{\text{gt}}, D_{\text{gt}})$ to a previous camera pose and then projecting back to get $(I_{\text{corrupt}}, D_{\text{corrupt}}$. The diffusion model learns to inpaint and refine the corrupted image $(I_{\text{corrupt}}, D_{\text{corrupt}})$ with ground truth $(I_{\text{gt}}, D_{\text{gt}})$. During inferencing, the sampling is conditioned on the anchored frame and future frame to preserve temporal consistency.

6 Diffusion in 3D Space

Diffusion in 2D images requires no data or only images. With available 3D datasets, another popular line of research directly performs 3D generation with diffusion models using 3D data, which has several different representations, e.g., point clouds, meshes, and neural fields. For these methods, the forward and reverse diffusion processes are applied to certain 3D representations $\mathbf{z}$. The whole network intends to directly learn the prior distribution of 3D space and aims to generate 3D shapes without further training during inferencing. The main training objective for the denoising process is similar to that in Section 3.1:

$$\mathbb{E}_{t\sim(0,T),\mathbf{x}_0\sim q(\mathbf{x}_0),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)\right\|^2\right] \tag{17}$$

where c is the condition. Prompt-guided generation, including text-guided and image-guided, can be achieved by adding conditions c to the diffusion training, *i.e.,* the denoiser is conditioned on c. One main approach utilizes the cross-attention mechanism to add connections between conditions and denoised 3D representations. Another applies adaptive group normalization (AdaGN) to combine the embedded condition with the denoising layers.

It is important to design a proper representation for diffusion models to learn the prior distribution. Thus most 3D generation methods using 3D diffusion contain two stages: they first train a network to convert the input explicit 3D data (e.g., mesh, point cloud) to a more usable form such as

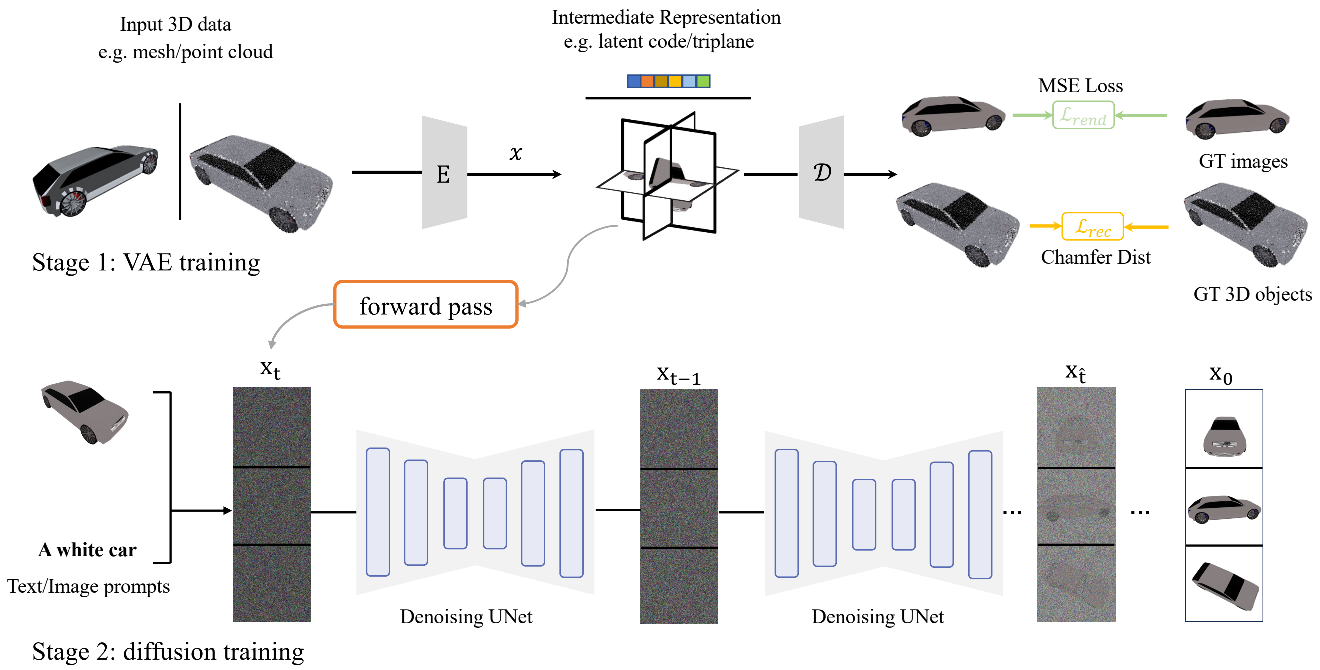


Fig. 5 A common two-stage pipeline for 3D generation with diffusion models. First, an encoder-decoder network (e.g., VAE) is applied to learn an intermediate representation (e.g., triplane) of 3D data. Then diffusion models are utilized to learn the prior distributions of this representation; $\mathbf{x}_{\hat{t}}$ denotes an intermediate timestep. One can thus generate 3D data by sampling from this learned distribution.

tri-planes or latent shape. This step usually includes training an autoencoder or a VAE architecture supervised by 3D reconstruction loss or neural rendering loss. Then diffusion models are utilized to learn the distribution of the intermediate representation. This procedure is visualized in Fig. 5. We categorize 3D generation methods based on which representation the diffusion process adopts. Representative results are presented later in Fig. 6.

6.1 3D Diffusion using Tri-planes

The tri-plane representation is a hybrid explicit–implicit representation, which is widely used for 3D tasks. Thus, we first review diffusion models on triplanes. A triplane consists of three axis-aligned feature maps with the same resolution $N \times N \times C$, where N is the spatial resolution and C is the number of channels. EG3D [17] introduce the tri-plane representation to generate 3D human faces in an efficient and expressive manner. To obtain the features of any 3D point, we can project it onto the three axis-aligned planes to get corresponding 2D features (F_{xy}, F_{yz}, F_{xz}) and then aggregate them via summation or multiplication. Since the features contain rich information, they can then be processed and used in volume rendering or shape reconstruction using shallow networks, with high efficiency.

Since the $N \times N \times C$ tri-planes can be viewed as C channel 2D images, NFD [104] suggests that we can directly utilize existing 2D diffusion backbones to generate normalized tri-planes. NFD first learns a dataset of tri-planes and a shared decoder that decodes tri-plane images into occupancy representation on a class of objects. It then trains the reverse process of the diffusion model on the generated tri-plane dataset with DDPM. During inferencing, the tri-plane distribution is sampled and decoded to a 3D shape by the shared decoder.

Rodin [105] also uses diffusion on tri-plane features for volume rendering of human faces. It uses a latent representation extracted from image, text, or random noise to condition the base diffusion model at a resolution of 64 × 64. Then it further trains a diffusion upsampler to lift the low-resolution tri-planes to 256 × 256, which helps to generate 3D structures with high fidelity. The low-resolution tri-planes serve as the condition for the diffusion process using 256 × 256 tri-planes. Instead of directly using a 2D convolution as in NFD [104], a 3D-aware convolution of the three planes is used to reinforce the cross-plane connections. Eq. 18 shows how this convolution works in practice, where $(.)$ indicates mean pooling on an axis.

$$F^{'}_{xz} = \mathrm{Conv2d}(\mathrm{concat}(F_{xz}, F_{(.)z}, F_{x(.)})) \qquad (18)$$

3DGen [106] adopts a VAE structure with tri-plane features as the intermediate representation. However, 3DGen decodes the tri-plane into an SDF, and then applies DmTet [107] and

NvDiffRast [108] to render the RGB and depth map from the reconstructed mesh for rendering-based supervision. The whole pipeline of 3DGen is pretrained on the Objaverse dataset [109], which substantially improves the quality of the generation results.

SSDNeRF [110] unifies the auto-encoding and diffusion stages of previous methods [106] into a single-stage diffusion model, thereby reducing the noise and artifacts introduced by intermediate latent codes in two-stage training. The training target of SSDNeRF is to minimize the variational upper bound on the negative data log-likelihood. The loss function consists of two items: a rendering loss to learn the tri-plane decoder and a diffusion denoising loss to learn the diffusion priors for tri-planes, as shown in Eq. 19. $\mathbf{x}$ denotes the tri-planes, π is the rendering camera pose, and y as the ground truth image. SSDNeRF supports both unconditional generation and image-based reconstruction with learned diffusion priors. For reconstruction during inferencing, SSDNeRF fine tunes the tri-plane with the training loss terms, but with a lower weight for the diffusion loss.

$$\begin{aligned}\mathcal{L} = \underbrace{\mathbb{E}[\sum_j \frac{1}{2}||y(\pi) - \text{render}(\mathbf{x},\pi)||^2]}_{\text{rendering loss}} + \\ \underbrace{\mathbb{E}_{t,\boldsymbol{\epsilon}}[\frac{1}{2}w^{(t)}||\text{denoiser}(\mathbf{x}_t;t) - \mathbf{x}||^2]}_{\text{diffusion loss}}\end{aligned} \tag{19}$$

Control3Diff [111] combines the strengths of diffusion models and GANs for versatile controllable 3D-aware image synthesis using single-view datasets for a class (e.g., FFHQ, AFHQ). It trains EG3D [17] to synthesize an infinite number of pairs of the control signal and tri-planes. Then it adopts a diffusion model with optional image guidance to jointly learn the prior distributions of tri-planes and camera poses of input images. During the inference stage, it allows additional rendering guidance for camera calibration and prediction.

6.2 3D Diffusion on Latent Space

Instead of tri-planes, some work encodes 3D objects or scenes into latent spaces that represent geometry or texture. The latent representations take various forms, e.g., 1D vectors or 3D grids, which are compressed and more suitable for transformer-based backbones. The diffusion models are thus trained in the latent space.

DiffusionSDF [112] applies a VAE-based architecture as the backbone for generation. It first trains the VAE to encode the input point clouds into the latent SDF space. It then trains a DDPM model on the latent representation. To allow conditional generation from partial clouds or images, it adds an additional cross-attention layer in each block of the diffusion model. During inferencing, it samples latent representations from a Gaussian distribution and decodes them with the SDF network. Diffusion-SDF [113] also adopts a VAE autoencoder to learn latent SDF space, but it instead encodes patch-level truncated signed distance functions (TSDF) into voxelized latent codes and introduces a voxelized diffusion model. It uses a UniU-Net architecture to replace the U-Net in DDPM; the former contains $1 \times 1 \times 1$ convolution layers to learn independent patch-focused information. Spatial transformer networks capture inter-patch relationships.

3D-LDM [114] adopts a VAE architecture to encode input SDF objects into compact latent codes for the diffusion process. It achieves multi-modal conditions through a cross-attention mechanism and classifer-free guidance (CFG). SD-Fusion [115] also employs conditioned diffusion networks on the latent codes from SDF inputs, and further improves the quality of the textures through SDS optimization [20] of generated geometry.

LION [116] uses a hierarchical VAE with PVCNN [117] backbones to encode both latent shape and latent points. It trains diffusion models on both latent spaces. The regularized latent points are more effective and expressive compared to raw point clouds, while the global latent shape is used to augment the model.

3DShape2VecSet [118] directly optimizes a set of latent codes to represent 3D objects. It first maps point clouds to positional embeddings and encodes them into a set of latent codes through a cross-attention module. Then the latent space is regularized with KL-divergence loss. 3Dshape2VecSet generates final objects by querying the decoded latent features f_i with an attention mechanism, as given below:

$$\hat{O}(x) = \text{FC}\left(\frac{\sum_i^m v(f_i)\exp\left(q(x)^T k(f_i)/\sqrt{d}\right)}{\sum_i^m \exp\left(q(x)^T k(f_i)/\sqrt{d}\right)}\right) \tag{20}$$

EDM [119] is used as the denoising network on the shape latent space.

Shap-E [120] adopts NeRF, and a signed distance and texture field (STF) to represent 3D objects. It learns an encoder to produce the parameters of NeRF and STF. The encoder first produces a latent representation of input 3D assets and then decodes it to MLP parameters. A diffusion prior is learned on the latent space.

3D VADER learns the denoising process on normalized 3D latent voxel grids and adopts EDM [119] in the inferencing step. It first trains the auto-decoder to decode a robustly-normalized latent voxel grid from the input 1D object embeddings. The final radiance volume representations are extracted from the latent grid through rendering supervision.

For scene generation, GAUDI [121] utilizes disentangled latent codes to represent scenes and camera poses. It first jointly optimizes latent codes and reconstruction networks with neural volume rendering losses. In the second step, GAUDI employs a DDPM model to learn the distribution of the latent codes. NeuralField-LDM [122] generates real-world 3D scenes with a three-stage pipeline. It first learns to encode scenes into a neural field with density and feature voxel grids. Then, the voxel grids are further compressed to a set of 3D coarse, 2D fine and 1D global latent representations. The diffusion model is trained on the tri-latent representation for 3D scene generation.

The diffusion process can also be performed in the latent style space of StyleGAN [123] as in StyleAvatar3D [124], which applies ControlNet [125] to introduce view, attribute, and style conditions for generating stylized images of humans. EG3D [17] is trained on the data from which image and style vector pairs can then be sampled. Finally, it applies the denoising process in the latent space of StyleGAN to allow 3D avatar generation from single-view image conditions.

6.3 3D Diffusion using Implicit Representation

Since the emergence of neural fields, implicit representations have become a popular form of encoding 3D assets. Existing works have also explored 3D generation using diffusion models on implicit representations, such as SDF, NeRF, or even MLP weights.

Yang et al. [130] follows a common two-stage pipeline and suggests that ReLU-fields are suitable for NeRF-based 3D generation. They first train the voxelized ReLU-fields with rendering and density losses. For the diffusion process, they utilize 3D convolution to update the U-Net structure used in DDPM [4], which suits the volume representation better.

Nikolai et al. [131] utilize a diffusion process to generate tetrahedral meshes. They adopt the VAE architecture and use a subdivision-based convolution and pooling operation for upsampling and subsampling tetrahedral grids. The diffusion process is carried out on the signed distance and displacement stored at the tetrahedra vertices. Similarly, MeshDiffusion [127] also represents meshes using tetrahedral grids fitted from random RGBD views. It applies DMTet [107] to extract meshes from normalized tetrahedral grids for differentiable rendering supervision. The diffusion model treats normalized signed distance values as floats and adds a refinement step for the deformation vectors to improve quality.

Hui et al. [132] use neural wavelets to represent 3D objects. They sample grid TSDF and apply multi-scale wavelet decomposition to generate both coarse and detailed wavelet coefficients. The diffusion model is applied to the coarse coefficient grids, which are refined by a detail predictor module. An explicit 3D representation can be obtained through an inverse wavelet transform on the detailed coefficients. Hu and Hui et al. [133] extend this framework for shape inversion and shape manipulation processes by adding a latent shape code as the condition in the denoising stage.

Since a high-resolution SDF grid is both memory and computationally expensive, LAS-Diffusion [134] uses a two-stage diffusion network: the first stage generates a low-resolution occupancy field to approximate the rough shape and the second stage generates detailed SDF values inside the occupied region. To incorporate 2D sketches for conditional generation, LAS-Diffusion introduces a view-aware local attention mechanism that uses local patch features of the input sketch to interact with the voxel feature via cross-attention.

HyperDiffusion [135] first applies diffusion models in MLP weight space and generates neural fields by predicting their weights. It first overfits a set of MLPs to faithfully represent individual dataset instances. The parameters are later sent to train the denoising network. The HyperDiffusion architecture supports both 3D shape generation and 4D mesh animation thanks to the flexibility of the weight space design.

DiffComplete [136] leverages diffusion models with voxelized TSDF and TUDF for 3D shape completion tasks. It formulates the completion task as TSDF shape generation conditioned on incomplete shapes. Instead of using the time-consuming cross-attention mechanism, DiffComplete uses an independent conditional branch to encode the incomplete corrupted shape conditions into the voxelized TSDF. As in ControlNet [125], the condition branch is merged with the TUDF voxels in the main branch by simple voxel addition.

For scene-level generation, DiffRoom [137] learns the denoising process on the cropped sparse room space with TSDF representation. It adopts a two-stage curriculum learning strategy, which first uses TSDF extracted by NeuralRecon [138] and then Gaussian noise as condition signals to train the 3D sparse denoising networks. For scene generation, DiffRoom splits the whole scene into overlapping crops and utilizes stochastic fusion on the crops to generate the final room geometries.

6.4 3D Diffusion using Explicit Representation

Diffusion models applied to explicit representations largely target point clouds. DPM [139] introduces a diffusion model to directly generate point clouds with a target shape code as the condition. It parameterizes the prior distribution of shape

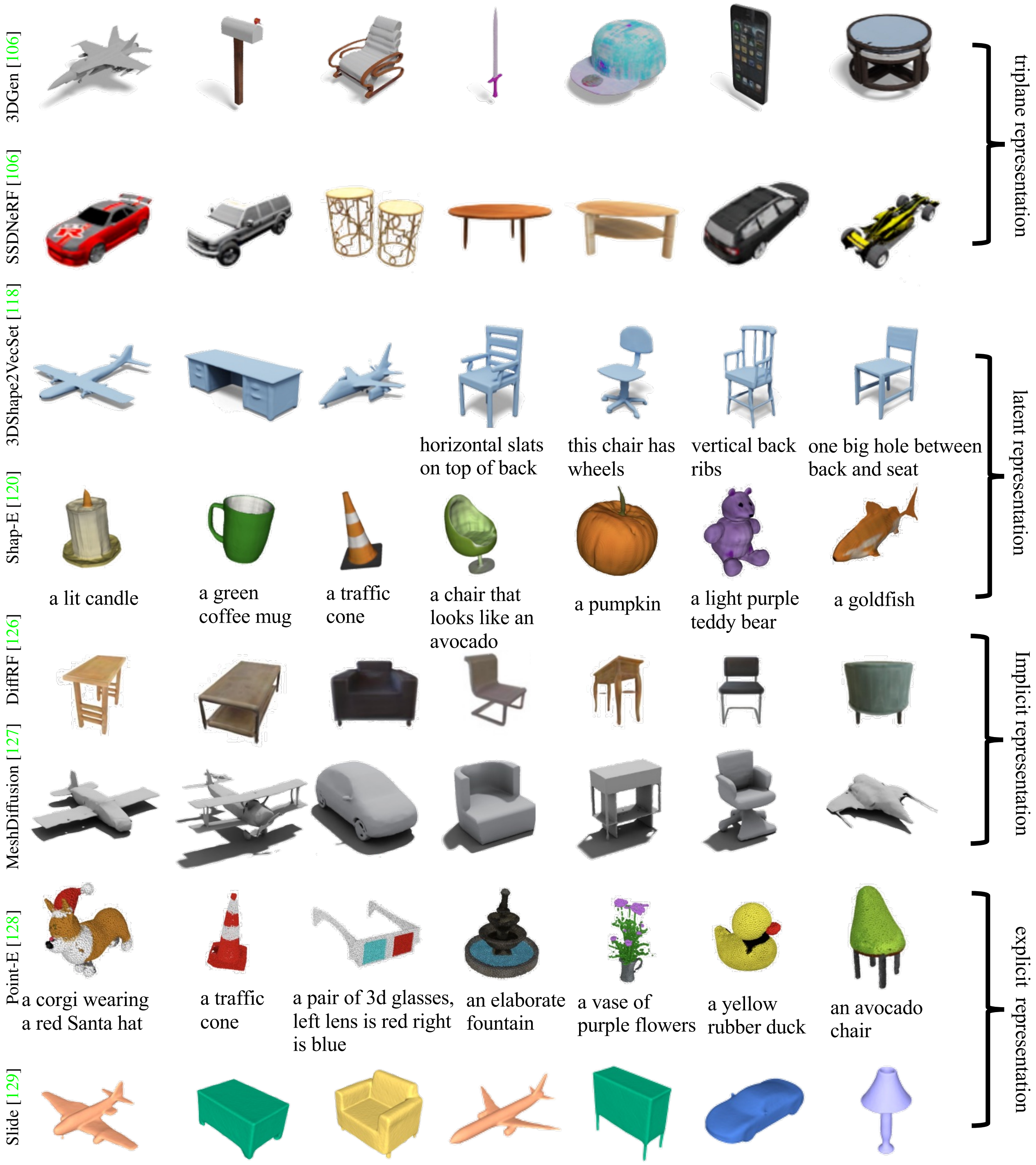


Fig. 6 A gallery of 3D generation results from 4 categories of methods performing 3D diffusion. Images with captions indicate text-to-3D results with specified prompts.

codes $p(\mathbf{z})$ using a normalizing flow and learns it end-to-end. An additional loss $\mathcal{L}_{\mathbf{z}}$ is added to control the latent distribution of generated point cloud $\mathbf{x}_0$ in the VAE close to $p(\mathbf{z})$:

$$\mathcal{L}_{\mathbf{z}} = D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}_0)||p(\mathbf{z})) \tag{21}$$

Kong et al. [140] extend DPM's network architecture for sketch-to-point cloud generation. They replace the conditional shape codes with sketch embedding and apply an adversarial loss to further refine the diffusion process. Later, PVD [141] directly trained a DDPM on point clouds by using the PVCNN [117] architecture as the denoising backbone, which voxelizes the point clouds for 3D convolution. With the unified 3D structure, PVD can handle both 3D point cloud generation and completion by varying training objectives.

Beyond unconditional generation, Point-E [128] provides the first framework for text-to-point cloud generation. A 2D image given a text prompt is first generated as the condition for the denoising process. It employs a transformer network as the denoising backbone with both noise inputs and image fed in as tokens. The generation process is cascaded: a 1K point cloud with the LR diffusion model is first generated and then a hierarchical upsampling network conditioned on the base points produces the final 4k point cloud.

STPD [142] combines sketch and text conditions to gain better control over point cloud generation. The sketch and text embeddings are fused through cascaded attention networks to create geometry and appearance conditions C_g, C_a. It generates point clouds by disentangling geometry component $\boldsymbol{g}_0$ and texture component $\boldsymbol{c}_0$. The generation process contains two stages, each of which is a conditioned diffusion process: the geometry stage learns the distribution of $p_{\theta_1}(\boldsymbol{g}_0|C_g)$ and the texture stage learns $p_{\theta_2}(\boldsymbol{a}_0|\boldsymbol{g}_0, C_a)$.

Point clouds can also be used as an intermediate representation for 3D generation. SLIDE [129] generates diverse meshes by generating point clouds first and then reconstructing surfaces from them. It uses a point cloud autoencoder consisting of an improved PointNet++ [143] to encode input point clouds as sparse points and hierarchical point up-sampling modules to recover point clouds of the original size. Then it learns the distributions of point positions and point features with two separate DDPMs for controllable generation.

HOLODiffusion [144] learns a diffusion model over the distribution of 3D voxel grids using 2D images as supervision. Specifically, it generates intermediate 3D-aware features conditioned only on the posed input images and applies 3D UNet to remove the noise added to this intermediate representation. The denoising loss is defined as the photometric error between rendered and input images. Based on HOLODiffusion, HOLOFusion [145] additionally trains an upsampling diffusion model to increase the quality of generated shapes.

DiffFacto [146] further studies part-based point cloud generation and editing with diffusion models. The whole pipeline consists of three parts: it first learns the latent codes $\mathbf{z}$ from each part of shape S. Then it fits the distribution of part transformation $P(\mathbf{T}|\mathbf{z})$ conditioned on part latent information. Finally, DiffFacto models the conditional distribution $P(S|\mathbf{z}, \mathbf{T})$ to sample part-level point clouds with a *cross diffusion network*, in which the cross attention layer pays attention to to m (number of parts) tokens each being the concatenation of $(\mathbf{x}_t, \mathbf{z}, T_s, j, t)$. The training objective of the diffusion model is:

$$\mathcal{L}_{\text{recon}} = \sum_{j=1}^{m} \sum_{x \in S_j} E_{\boldsymbol{\epsilon},\mathbf{z},t}[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\phi(\mathbf{x}_t, \mathbf{z}, T_s, j, t)||_2^2] \tag{22}$$

Other than using point clouds, DiffRF [126] was the first approach to apply diffusion models to directly generate neural fields. It utilizes an explicit voxel grid as a NeRF to represent 3D objects. It first fits the explicit grid to input multi-view images using neural volume rendering. When training diffusion models, it combines the denoising loss in diffusion models and weighted RGB rendering loss in each time step using a time-conditioned 3D UNet.

GVGEN [147] and GaussianCube [148] train diffusion models to generate 3D Gaussians. Both found that directly generating all attributes of 3D Gaussians is challenging and proposed to anchor the Gaussians' positions first.

7 Open-Source 3D Datasets

3D datasets are essential for diffusion-based 3D-3D generation: diffusion models need sufficient training data to learn the prior distributions of 3D representations. Here we list and categorize open-source datasets widely used in 3D generation. Some [105, 120, 128] rely on their own sizable collections of 3D object data.

7.1 Datasets of Objects

ShapeNet [149] (including ShapeNetSem and ShapeNetCore) is the most widely used dataset in diffusion-based 3D-3D generation applications. It contains nearly 3M textured objects in 3,135 categories. Its subsets, such as cars, chairs, and planes, are frequently used in per-class generation tasks. Acronym [151] provides watertight meshes across 262 categories, produced from the original ShapeNet dataset, making it easy to convert them to other representations, e.g., SDF.

ModelNet [150] also provides a large number of models for 3D deep learning. It contains 151,128 objects belonging

Table 2 Datasets for Diffusion-based 3D Object Generation

Dataset	Year	Data Type	Categories	Size	Source of 3D Data
ShapeNet [149]	2015	mesh with texture	3135	3,000,000	synthetic
ModelNet [150]	2015	mesh	660	151,728	synthetic
Acronym [151]	2021	Watertight mesh	262	8872	synthetic
Objaverse [109]	2023	mesh with texture	-	800,000	synthetic
BuildingNet [152]	2021	mesh	5	2,000	synthetic
RedWood [153]	2016	mesh	-	-	scanned
YCB [154]	2015	mesh	5	75	scanned
PhotoShape Chairs [155]	2018	mesh	1	29,133	synthetic
ABO [156]	2022	mesh	63	147,702	reconstructed
Text2Shape [157]	2019	voxel-text pair	2	15,038	synthetic
ShapeGlot [158]	2019	mesh-text pair	1	7,000	synthetic
Pix3D [159]	2018	image-mesh	9	395	synthetic
Co3D [160]	2021	videos	50	18,619	scanned
MVImgNet [161]	2023	image-point cloud	238	219,188	scanned

Table 3 Datasets for Diffusion-based 3D Scene Generation

Dataset	Year	Data Type	Size	Source of 3D Data
Matterport3D [162]	2017	images with annotations	10,800 views with 194,400 images	scanned
Realestate10K [163]	2018	images	10M images	website
CLEVR [164]	2017	images with questions	100K images, 853K questions	synthetic
LHQ [165]	2021	images	91,963 images	nature
ARKitScenes [166]	2021	images	5047 captures	scanned
VizDoom [167]	2016	synthesized scenes	1 configurable scene	synthetic
Replica [168]	2019	mesh with texture	18 scenes	scanned
Carla [169]	2017	synthesized scenes	2 scenes	synthetic

to 660 unique categories and is thus widely used in various 3D tasks, e.g., 3D object classification.

Objaverse [109] is a recent large 3D dataset with nearly 800K textured objects and corresponding captions. It has been widely utilized in pretraining neural networks for various 3D tasks, such as 3D generation [106] and 3D segmentation [170].

BuildingNet [152] provides nearly 2K building objects with more than 513K annotated mesh primitives.

RedWood [153] contains more than 10K objects along with 23M images scanned from the real world.

YCB [154] reconstructs meshes from 600 scanned RGBD images and contains 75 high-quality objects in total.

PhotoShape Chairs [155] provides more than 29,000 synthetic relightable chair objects with photorealistic materials.

Amazon Berkeley Objects (ABO) [156] contains more than 140K models of 63 kinds of products on Amazon shopping websites. Like ShapeNet, diffusion models are usually trained with a single category such as tables from the ABO dataset for 3D-3D generation tasks.

Datasets with shape-text pairs can be used in training diffusion models conditioned on text prompts.

Text2Shape [157] provides captions for chair and table subsets in ShapeNet and contains nearly 75K shape-text pairs. A hybrid sampling strategy is applied to voxelized 3D mesh objects.

ShapeGlot [158] also provides more than 70K utterances for the ShapeNet chair subset. Each prompt is accompanied by additional distractors.

Some methods utilize 2D-3D reconstruction datasets for image-conditioned 3D-3D generation.

Pix3D [159] provides well-aligned real-world image-shape pairs containing 10697 images and 395 shapes. Each 3D shape is associated with a diverse set of images and has a precise 3D pose annotation.

FFHQ [123] and *AFHQ* [171] provide a large number of various high-resolution human and animal faces respectively. They are widely used in GAN-based generation tasks and can be used as sources of image-conditioning for diffusion models. FFHQ contains 70,000 images varying of age, ethnicity, etc. AFHQ comprises AFHQ-Cat, AFHQ-Dog, and AFHQ-Wild, each containing 15,000 images.

Co3D [160] focuses on providing a tremendous range of real-world objects. It contains almost 19,000 videos in 50 categories, with accurate camera parameters checked manually.

MVImgNet [161] provides multi-view images of 219,188 real-world objects in 238 classes. Each object also contains annotations, including masks and scanned point clouds.

Among the above, Objaverse [109] is currently the most used dataset for object generation. Co3D [160] and MVImgNet [161] are also becoming popular because they contain backgrounds.

7.2 Datasets of Scenes

For scene generation, popular 3D scene datasets include the following.

Matterport3D [162] provides 10,800 panoramic views of 90 large real world indoor scenes. It contains comprehensive annotations of view camera poses, surface reconstructions, and 2D & 3D semantic segmentation results.

Realestate10K [163] provides timestamps and camera trajectories for more than 10 million images from videos on YouTube based on SLAM algorithms. It collects images in the real estate category, and features both indoor and outdoor scenes.

CLEVR [164] is a diagnostic dataset aiming at testing visual-question answering (VQA) systems in visual reasoning tasks. It contains 100K rendered images of annotated 3D objects and nearly 853K unique questions.

LHQ (Landscapes High-Quality) [165] contains 91,693 high-resolution natural landscapes, each with resolution higher than 1024^2. The raw images were collected from Unsplash and Flickr websites and then filtered by a blacklist of keywords and a Mask R-CNN network in turn.

Vizdoom [167] is a Doom-based platform for reinforcement learning and provides a simple scene setting.

Replica [168] consists of 18 high-quality real-world indoor scene reconstructions, each containing high-resolution meshes, HDR texture and per-primitive semantic class and instance information.

ARKitScenes [166] provides more than 5K indoor scans with rendered images and depth maps along given trajectories, captured by Apple's LiDAR scanner. Reconstructed surfaces and corresponding object bounding boxes are also accessible in ARKitScenes.

CARLA [169] is an open-world simulator specifically designed for studying autonomous driving. CARLA is implemented based on the UE4 engine and contains two scenes containing 40 buildings, 16 vehicles, and 50 pedestrian models.

8 Future Directions

In this section, we highlight current challenges and potential research directions for 3d generation by diffusion models.

8.1 Generation Quality

Unlike 2D generative models that can synthesize realistic images almost indistinguishable from real ones, the quality of generated 3D output still remains unsatisfactory.

Methods that extract 3D representations from pretrained 2D diffusion models have results fully determined by these 2D models. By harnessing the capabilities of these 2D models, current methods can synthesize detailed and fairly realistic 3D models of diverse kinds of objects, pose variations and artistic styles. They can even deal with intricate and abstract textual prompts, such as 'a robot and dinosaur playing chess'. Nonetheless, these approaches are susceptible to issues like the multi-face Janus problem, which is known to be associated with the training data distribution of the 2D diffusion models.

Methods trained on 3D data are often constrained to generating relatively simple objects and exhibit over-smooth textures due to their heavy reliance on existing 3D datasets, which, unfortunately, comprise mostly basic objects. Also, these methods are mostly trained on a single class of objects and haven't demonstrated the ability to generate in-the-wild objects. Recently, Objaverse-XL [172] was introduced to include over 10 million diverse 3D objects. Future works might explore how to utilize this large dataset to push the limits of 3D generation. This requires work on several issues, including architectural design, data representation and training strategy.

Only a few methods deal with scene-level generation. Existing works have explored compositional generation, iterative generation and synthesizing a long trajectory from a 2D image. However, they often require dedicated design (depth warping, inpainting etc) to create a valid scene. This process is not always successful: the resulting geometry may contain artifacts like holes or distortion, and the texture may be oversmooth or view-inconsistent. Moreover, it is more challenging to generate reasonable geometry for outdoor scenes, as they are more complex and have drastic depth continuities. Further, there is no work to train on 3D scenes, to generate scene assets directly, possibly because of insufficient scene-level data and high computational load. Future work can explore scene generation with flexible user control, material decomposition from lighting and better generation quality.

8.2 Efficiency

Utilizing pretrained diffusion models for 3D scene generation introduces certain challenges in terms of both efficiency and resource requirements. These methods necessitate per-scene optimization for many iterations for every provided prompt, a procedure that can potentially take hours to achieve satisfactory geometry. Moreover, they often require more than 20 GB of GPU memory. Although ATT3D [36] turns optimization into direct inference, generation is limited to the training set and it cannot handle arbitrary prompts. Multi-view generation followed by a reconstruction pipeline has greatly speeded up the generation process, with GECO [65] further achieving feedforward generation by distilling multi-view diffusion models into one-step. However, the results are still limited by multi-view generation. As for methods training on 3D representations, the training process often takes several days to converge since the diffusion process uses high-dimension data. Also, many of them include various data processing steps, e.g., training an auto-encoder for 3D shapes. After training, 3D shapes can be obtained via direct inference.

8.3 Evaluation Protocol

Evaluation of 3D generation has always been challenging since there is no ground truth data to compare with. No direct metric can measure how ‘good’ a 3D model is. Methods trained on 3D datasets often use Fréchet Inception Distance (FID) and Inception Score (IS) to evaluate image quality, and use Coverage Score (COV) and Minimum Matching Distance (MMD) with Chamfer Distance (CD) to evaluate geometric quality. However, these metrics are limited to simple, single-class objects.

For zero-shot text-to-3D generation, existing methods use CLIP R-Precision to measure the consistency of rendered images and text prompts. There are no suitable metrics to quantify view consistency and geometry quality of 3D assets. It is also essential to have a diverse and representative test set covering different objects and scenes to fairly evaluate the capabilities of the generative models. Recently, T3-Bench [173] partly addresses this problem by providing a benchmark of 100 prompts and employing text-image scoring models (e.g., CLIP) to detect the consistency of rendered 2D views.

8.4 Towards Real-world Applications

Generated 3D assets are not yet suitable for practical real-world applications. Unlike 2D generation, where output images can be easily edited using well-established photo-editing tools, 3D generation involves the optimization of geometry and structure through neural networks, making the editing of generated assets more challenging. Moreover, these assets may not fully encompass the logical and operational nuances of 3D modeling as understood by human experts. Additionally, current methods for 3D generation lack the necessary flexibility to allow precise control and editing of the finer details of the output. Consequently, modifying AI-generated 3D models is still challenging, especially for amateurs.

9 Conclusions

3D generation has captured significant attention in recent years and has made notable advances through the evolution of diffusion models. In this survey, we have systematically reviewed and summarized recent works on 3D generation utilizing diffusion models. We first outlined the foundational concepts of diffusion models and 3D data representations. Subsequently, we reviewed existing works according to how they use diffusion and whether they exploit pretrained diffusion models. We discussed the advantages and disadvantages of different works, indicating the architecture, and shown results for representative methods. Additionally, we summarized widely employed datasets for training 3D generative models. We finished by outlining potential directions for future research. As the first survey on 3D generation with diffusion models, we hope this paper offers researchers a concise overview of relevant works and the path of development. We also hope our survey will inspire more researchers to delve into this domain and contribute more advanced techniques.

Acknowledgements

The authors would like to thank Lingjie Liu and Meng-Hao Guo for helpful suggestions.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Kingma DP, Welling M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[2] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. *Communications of the ACM*, 2020, 63(11): 139–144.

[3] Rezende D, Mohamed S. Variational inference with normalizing flows. In *International conference on machine learning*, 2015, 1530–1538.

[4] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020, 33: 6840–6851.

[5] Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 10684–10695.

[6] Saharia C, Chan W, Saxena S, Li L, Whang J, Denton EL, Ghasemipour K, Gontijo Lopes R, Karagol Ayan B, Salimans T, et al.. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 2022, 35: 36479–36494.

[7] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021, 65(1): 99–106.

[8] Yang L, Zhang Z, Song Y, Hong S, Xu R, Zhao Y, Shao Y, Zhang W, Cui B, Yang MH. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

[9] Zhang C, Zhang C, Zhang M, Kweon IS. Text-to-image diffusion model in generative ai: A survey. *arXiv preprint arXiv:2303.07909*, 2023.

[10] Li C, Zhang C, Waghwase A, Lee LH, Rameau F, Yang Y, Bae SH, Hong CS. Generative AI meets 3D: A Survey on Text-to-3D in AIGC Era. *arXiv preprint arXiv:2305.06131*, 2023.

[11] Shi Z, Peng S, Xu Y, Liao Y, Shen Y. Deep generative models on 3d representations: A survey. *arXiv preprint arXiv:2210.15663*, 2022.

[12] Sohl-Dickstein J, Weiss EA, Maheswaranathan N, Ganguli S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, 2256–2265.

[13] Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[14] Xie Y, Takikawa T, Saito S, Litany O, Yan S, Khan N, Tombari F, Tompkin J, Sitzmann V, Sridhar S. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, 2022, 641–676.

[15] Chen A, Xu Z, Wei X, Tang S, Su H, Geiger A. Factor fields: A unified framework for neural fields and beyond. *arXiv preprint arXiv:2302.01226*, 2023.

[16] Müller T, Evans A, Schied C, Keller A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 2022, 41(4): 1–15.

[17] Chan ER, Lin CZ, Chan MA, Nagano K, Pan B, De Mello S, Gallo O, Guibas LJ, Tremblay J, Khamis S, et al.. Efficient geometry-aware 3D generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 16123–16133.

[18] Fridovich-Keil S, Yu A, Tancik M, Chen Q, Recht B, Kanazawa A. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, 5501–5510.

[19] Kerbl B, Kopanas G, Leimkühler T, Drettakis G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 2023, 42(4): 139–1.

[20] Poole B, Jain A, Barron JT, Mildenhall B. Dreamfusion: Text-to-3d using 2d diffusion. *International Conference on Learning Representations*, 2022.

[21] Wang H, Du X, Li J, Yeh RA, Shakhnarovich G. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 12619–12629.

[22] Liang Y, Yang X, Lin J, Li H, Xu X, Chen Y. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, 6517–6526.

[23] Wang Z, Lu C, Wang Y, Bao F, Li C, Su H, Zhu J. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 2024, 36.

[24] Barron JT, Mildenhall B, Tancik M, Hedman P, Martin-Brualla R, Srinivasan PP. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 5855–5864.

[25] Tang J, Ren J, Zhou H, Liu Z, Zeng G. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *The Twelfth International Conference on Learning Representations*, 2023.

[26] Lin CH, Gao J, Tang L, Takikawa T, Zeng X, Huang X, Kreis K, Fidler S, Liu MY, Lin TY. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 300–309.

[27] Tsalicoglou C, Manhardt F, Tonioni A, Niemeyer M, Tombari F. Textmesh: Generation of realistic 3d meshes from text prompts. In *2024 International Conference on 3D Vision (3DV)*, 2024, 1554–1563.

[28] Metzer G, Richardson E, Patashnik O, Giryes R, Cohen-Or D. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 12663–12673.

[29] Chen R, Chen Y, Jiao N, Jia K. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, 22246–22256.

[30] Seo H, Kim H, Kim G, Chun SY. DITTO-NeRF: Diffusion-based Iterative Text To Omni-directional 3D Model. *arXiv preprint arXiv:2304.02827*, 2023.

[31] Seo J, Jang W, Kwak MS, Ko J, Kim H, Kim J, Kim JH, Lee J, Kim S. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023.

[32] Liao TH, Ge S, Xu Y, Lee YC, AlBahar B, Huang JB. Text-

driven Visual Synthesis with Latent Diffusion Prior. *arXiv preprint arXiv:2302.08510*, 2023.

[33] Armandpour M, Zheng H, Sadeghian A, Sadeghian A, Zhou M. Re-imagine the Negative Prompt Algorithm: Transform 2D Diffusion into 3D, alleviate Janus problem and Beyond. *arXiv preprint arXiv:2304.04968*, 2023.

[34] Zhu J, Zhuang P. HiFA: High-fidelity Text-to-3D with Advanced Diffusion Guidance. *arXiv preprint arXiv:2305.18766*, 2023.

[35] Huang Y, Wang J, Shi Y, Qi X, Zha ZJ, Zhang L. DreamTime: An Improved Optimization Strategy for Text-to-3D Content Creation. *arXiv preprint arXiv:2306.12422*, 2023.

[36] Lorraine J, Xie K, Zeng X, Lin CH, Takikawa T, Sharp N, Lin TY, Liu MY, Fidler S, Lucas J. Att3d: Amortized text-to-3d object synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 17946–17956.

[37] Xie K, Lorraine J, Cao T, Gao J, Lucas J, Torralba A, Fidler S, Zeng X. Latte3d: Large-scale amortized text-to-enhanced3d synthesis. *arXiv preprint arXiv:2403.15385*, 2024.

[38] Fridman R, Abecasis A, Kasten Y, Dekel T. Scenescape: Text-driven consistent scene generation. *Advances in Neural Information Processing Systems*, 2023, 36.

[39] Höllein L, Cao A, Owens A, Johnson J, Nießner M. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 7909–7920.

[40] Zhang J, Li X, Wan Z, Wang C, Liao J. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[41] Li J, Bansal M. Panogen: Text-conditioned panoramic environment generation for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 2024, 36.

[42] Tang S, Zhang F, Chen J, Wang P, Furukawa Y. MVDiffusion: Enabling Holistic Multi-view Image Generation with Correspondence-Aware Diffusion. *arXiv preprint arXiv:2307.01097*, 2023.

[43] Po R, Wetzstein G. Compositional 3d scene generation using locally conditioned diffusion. In *2024 International Conference on 3D Vision (3DV)*, 2024, 651–663.

[44] Cohen-Bar D, Richardson E, Metzer G, Giryes R, Cohen-Or D. Set-the-scene: Global-local training for generating controllable nerf scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 2920–2929.

[45] Lin Y, Bai H, Li S, Lu H, Lin X, Xiong H, Wang L. Componerf: Text-guided multi-object compositional nerf with editable 3d scene layout. *arXiv preprint arXiv:2303.13843*, 2023.

[46] Xu D, Jiang Y, Wang P, Fan Z, Wang Y, Wang Z. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 4479–4489.

[47] Deng C, Jiang C, Qi CR, Yan X, Zhou Y, Guibas L, Anguelov D, et al.. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 20637–20647.

[48] Melas-Kyriazi L, Laina I, Rupprecht C, Vedaldi A. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 8446–8455.

[49] Gal R, Alaluf Y, Atzmon Y, Patashnik O, Bermano AH, Chechik G, Cohen-Or D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[50] Tang J, Wang T, Zhang B, Zhang T, Yi R, Ma L, Chen D. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, 22819–22829.

[51] Yoo P, Guo J, Matsuo Y, Gu SS. Dreamsparse: Escaping from plato's cave with 2d diffusion model given sparse views. *Advances in Neural Information Processing Systems*, 2024, 36.

[52] Raj A, Kaza S, Poole B, Niemeyer M, Ruiz N, Mildenhall B, Zada S, Aberman K, Rubinstein M, Barron J, et al.. Dreambooth3d: Subject-driven text-to-3d generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, 2349–2359.

[53] Liu R, Wu R, Van Hoorick B, Tokmakov P, Zakharov S, Vondrick C. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, 9298–9309.

[54] Qian G, Mai J, Hamdi A, Ren J, Siarohin A, Li B, Lee HY, Skorokhodov I, Wonka P, Tulyakov S, et al.. Magic123: One Image to High-Quality 3D Object Generation Using Both 2D and 3D Diffusion Priors. *The Twelfth International Conference on Learning Representations*, 2023.

[55] Liu M, Xu C, Jin H, Chen L, Xu Z, Su H, et al.. One-2-3-45: Any Single Image to 3D Mesh in 45 Seconds without Per-Shape Optimization. *Advances in Neural Information Processing Systems*, 2024, 36.

[56] Shi Y, Wang P, Ye J, Long M, Li K, Yang X. Mvdream: Multi-view diffusion for 3d generation. *The Twelfth International Conference on Learning Representations*, 2023.

[57] Liu Y, Lin C, Zeng Z, Long X, Liu L, Komura T, Wang W. SyncDreamer: Generating Multiview-consistent Images from a Single-view Image. *arXiv preprint arXiv:2309.03453*, 2023.

[58] Shi R, Chen H, Zhang Z, Liu M, Xu C, Wei X, Chen L, Zeng C, Su H. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[59] Long X, Guo YC, Lin C, Liu Y, Dou Z, Liu L, Ma Y, Zhang SH, Habermann M, Theobalt C, et al.. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, 9970–9980.

[60] Li J, Tan H, Zhang K, Xu Z, Luan F, Xu Y, Hong Y, Sunkavalli K, Shakhnarovich G, Bi S. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *The Twelfth International Conference on Learning Representations*, 2023.

[61] Wang P, Shi Y. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

[62] Tang J, Chen Z, Chen X, Wang T, Zeng G, Liu Z. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

[63] Xu Y, Shi Z, Yifan W, Chen H, Yang C, Peng S, Shen Y, Wetzstein G. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024.

[64] Xu J, Cheng W, Gao Y, Wang X, Gao S, Shan Y. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[65] Wang C, Gu J, Long X, Liu Y, Liu L. GECO: Generative Image-to-3D within a SECOnd. *arXiv preprint arXiv:2405.20327*, 2024.

[66] Gao R, Holynski A, Henzler P, Brussee A, Martin-Brualla R, Srinivasan P, Barron JT, Poole B. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

[67] Zhang L, Qiu Q, Lin H, Zhang Q, Shi C, Yang W, Shi Y, Yang S, Xu L, Yu J. DreamFace: Progressive Generation of Animatable 3D Faces under Text Guidance. *arXiv preprint arXiv:2304.03117*, 2023.

[68] Li R, Bladin K, Zhao Y, Chinara C, Ingraham O, Xiang P, Ren X, Prasad P, Kishore B, Xing J, et al.. Learning formation of physically-based face attributes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, 3410–3419.

[69] Jiang R, Wang C, Zhang J, Chai M, He M, Chen D, Liao J. Avatarcraft: Transforming text into neural human avatars with parameterized shape and pose control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 14371–14382.

[70] Cao Y, Cao YP, Han K, Shan Y, Wong KYK. Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, 958–968.

[71] Huang Y, Wang J, Zeng A, Cao H, Qi X, Shi Y, Zha ZJ, Zhang L. DreamWaltz: Make a Scene with Complex 3D Animatable Avatars. *Advances in Neural Information Processing Systems*, 2024, 36.

[72] Liao T, Yi H, Xiu Y, Tang J, Huang Y, Thies J, Black MJ. Tada! text to animatable digital avatars. In *2024 International Conference on 3D Vision (3DV)*, 2024, 1508–1519.

[73] Kolotouros N, Alldieck T, Zanfir A, Bazavan E, Fieraru M, Sminchisescu C. Dreamhuman: Animatable 3d avatars from text. *Advances in Neural Information Processing Systems*, 2024, 36.

[74] Weng Z, Wang Z, Yeung S. ZeroAvatar: Zero-shot 3D Avatar Generation from a Single Image. *arXiv preprint arXiv:2305.16411*, 2023.

[75] Zeng Y, Lu Y, Ji X, Yao Y, Zhu H, Cao X. AvatarBooth: High-Quality and Customizable 3D Human Avatar Generation. *arXiv preprint arXiv:2306.09864*, 2023.

[76] Jakab T, Li R, Wu S, Rupprecht C, Vedaldi A. Farm3d: Learning articulated 3d animals by distilling 2d diffusion. In *2024 International Conference on 3D Vision (3DV)*, 2024, 852–861.

[77] Yao CH, Raj A, Hung WC, Rubinstein M, Li Y, Yang MH, Jampani V. Artic3d: Learning robust articulated 3d shapes from noisy web image collections. *Advances in Neural Information Processing Systems*, 2024, 36.

[78] Haque A, Tancik M, Efros AA, Holynski A, Kanazawa A. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 19740–19750.

[79] Kamata H, Sakuma Y, Hayakawa A, Ishii M, Narihira T. Instruct 3D-to-3D: Text Instruction Guided 3D-to-3D conversion. *arXiv preprint arXiv:2303.15780*, 2023.

[80] Brooks T, Holynski A, Efros AA. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 18392–18402.

[81] Yu L, Xiang W, Han K. Edit-DiffNeRF: Editing 3D Neural Radiance Fields using 2D Diffusion Model. *arXiv preprint arXiv:2306.09551*, 2023.

[82] Shao R, Sun J, Peng C, Zheng Z, Zhou B, Zhang H, Liu Y. Control4D: Dynamic Portrait Editing by Learning 4D GAN from 2D Diffusion-based Editor. *arXiv preprint arXiv:2305.20082*, 2023.

[83] Zhou X, He Y, Yu FR, Li J, Li Y. RePaint-NeRF: NeRF Editting via Semantic Masks and Diffusion Models. *arXiv preprint arXiv:2306.05668*, 2023.

[84] Zhuang J, Wang C, Lin L, Liu L, Li G. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*, 2023, 1–10.

[85] Li Y, Dou Y, Shi Y, Lei Y, Chen X, Zhang Y, Zhou P, Ni B. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024, 3279–3287.

[86] Guo YC, Liu YT, Wang C, Zou ZX, Luo G, Chen CH, Cao YP, Zhang SH. threestudio: A unified framework for 3D content generation. https://github.com/threestudio-project/threestudio, 2023.

[87] Watson D, Chan W, Martin-Brualla R, Ho J, Tagliasacchi A, Norouzi M. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022.

[88] Chan ER, Nagano K, Chan MA, Bergman AW, Park JJ, Levy A, Aittala M, De Mello S, Karras T, Wetzstein G. Generative novel view synthesis with 3d-aware diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 4217–4229.

[89] Gu J, Trevithick A, Lin KE, Susskind J, Theobalt C, Liu L, Ramamoorthi R. NerfDiff: Single-image View Synthesis with NeRF-guided Distillation from 3D-aware Diffusion. *arXiv preprint arXiv:2302.10109*, 2023.

[90] Yu A, Ye V, Tancik M, Kanazawa A. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 4578–4587.

[91] Li G, Zheng H, Wang C, Li C, Zheng C, Tao D. 3DDesigner: Towards Photorealistic 3D Object Generation and Editing with Text-guided Diffusion Models. *arXiv preprint arXiv:2211.14108*, 2022.

[92] Zhou Z, Tulsiani S. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 12588–12597.

[93] Anciukevičius T, Xu Z, Fisher M, Henderson P, Bilen H, Mitra NJ, Guerrero P. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 12608–12618.

[94] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 2015, 234–241.

[95] Tewari A, Yin T, Cazenavette G, Rezchikov S, Tenenbaum J, Durand F, Freeman B, Sitzmann V. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *Advances in Neural Information Processing Systems*, 2024, 36.

[96] Szymanowicz S, Rupprecht C, Vedaldi A. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 8863–8873.

[97] Xu Y, Tan H, Luan F, Bi S, Wang P, Li J, Shi Z, Sunkavalli K, Wetzstein G, Xu Z, et al.. DMV3D: Denoising Multi-view Diffusion Using 3D Large Reconstruction Model. *The Twelfth International Conference on Learning Representations*, 2023.

[98] Hong Y, Zhang K, Gu J, Bi S, Zhou Y, Liu D, Liu F, Sunkavalli K, Bui T, Tan H. Lrm: Large reconstruction model for single image to 3d. *The Twelfth International Conference on Learning Representations*, 2023.

[99] Xiang J, Yang J, Huang B, Tong X. 3D-aware Image Generation using 2D Diffusion Models. *The Eleventh International Conference on Learning Representations*, 2023.

[100] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009, 248–255.

[101] Tseng HY, Li Q, Kim C, Alsisan S, Huang JB, Kopf J. Consistent View Synthesis with Pose-Guided Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 16773–16783.

[102] Yu JJ, Forghani F, Derpanis KG, Brubaker MA. Long-term photometric consistent novel view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 7094–7104.

[103] Cai S, Chan ER, Peng S, Shahbazi M, Obukhov A, Van Gool L, Wetzstein G. Diffdreamer: Towards consistent unsupervised single-view scene extrapolation with conditional diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 2139–2150.

[104] Shue JR, Chan ER, Po R, Ankner Z, Wu J, Wetzstein G. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 20875–20886.

[105] Wang T, Zhang B, Zhang T, Gu S, Bao J, Baltrusaitis T, Shen J, Chen D, Wen F, Chen Q, et al.. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, 4563–4573.

[106] Gupta A, Xiong W, Nie Y, Jones I, Oğuz B. 3DGen: Triplane Latent Diffusion for Textured Mesh Generation. *arXiv preprint arXiv:2303.05371*, 2023.

[107] Shen T, Gao J, Yin K, Liu MY, Fidler S. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 2021, 34: 6087–6101.

[108] Laine S, Hellsten J, Karras T, Seol Y, Lehtinen J, Aila T. Modular Primitives for High-Performance Differentiable Rendering. *ACM Transactions on Graphics*, 2020, 39(6).

[109] Deitke M, Schwenk D, Salvador J, Weihs L, Michel O, VanderBilt E, Schmidt L, Ehsani K, Kembhavi A, Farhadi A. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 13142–13153.

[110] Chen H, Gu J, Chen A, Tian W, Tu Z, Liu L, Su H. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, 2416–2425.

[111] Gu J, Gao Q, Zhai S, Chen B, Liu L, Susskind J. Control3diff: Learning controllable 3d diffusion models from single-view images. In *2024 International Conference on 3D Vision (3DV)*, 2024, 685–696.

[112] Chou G, Bahat Y, Heide F. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, 2262–2272.

[113] Li M, Duan Y, Zhou J, Lu J. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2023, 12642–12651.

[114] Nam G, Khlifi M, Rodriguez A, Tono A, Zhou L, Guerrero P. 3D-LDM: Neural Implicit 3D Shape Generation with Latent Diffusion Models. *arXiv preprint arXiv:2212.00842*, 2022.

[115] Cheng YC, Lee HY, Tulyakov S, Schwing AG, Gui LY. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 4456–4465.

[116] Vahdat A, Williams F, Gojcic Z, Litany O, Fidler S, Kreis K, et al.. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 2022, 35: 10021–10039.

[117] Liu Z, Tang H, Lin Y, Han S. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 2019, 32.

[118] Zhang B, Tang J, Niessner M, Wonka P. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 2023, 42(4): 1–16.

[119] Karras T, Aittala M, Aila T, Laine S. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 2022, 35: 26565–26577.

[120] Jun H, Nichol A. Shap-E: Generating Conditional 3D Implicit Functions. *arXiv preprint arXiv:2305.02463*, 2023.

[121] Bautista MA, Guo P, Abnar S, Talbott W, Toshev A, Chen Z, Dinh L, Zhai S, Goh H, Ulbricht D, et al.. Gaudi: A neural architect for immersive 3d scene generation. *Advances in Neural Information Processing Systems*, 2022, 35: 25102–25116.

[122] Kim SW, Brown B, Yin K, Kreis K, Schwarz K, Li D, Rombach R, Torralba A, Fidler S. NeuralField-LDM: Scene Generation with Hierarchical Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 8496–8506.

[123] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, 4401–4410.

[124] Zhang C, Chen Y, Fu Y, Zhou Z, Yu G, Wang B, Fu B, Chen T, Lin G, Shen C. StyleAvatar3D: Leveraging Image-Text Diffusion Models for High-Fidelity 3D Avatar Generation. *arXiv preprint arXiv:2305.19012*, 2023.

[125] Zhang L, Rao A, Agrawala M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 3836–3847.

[126] Müller N, Siddiqui Y, Porzi L, Bulo SR, Kontschieder P, Nießner M. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 4328–4338.

[127] Liu Z, Feng Y, Black MJ, Nowrouzezahrai D, Paull L, Liu W. MeshDiffusion: Score-based Generative 3D Mesh Modeling. *arXiv preprint arXiv:2303.08133*, 2023.

[128] Nichol A, Jun H, Dhariwal P, Mishkin P, Chen M. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[129] Lyu Z, Wang J, An Y, Zhang Y, Lin D, Dai B. Controllable mesh generation through sparse latent point diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, 271–280.

[130] Yang G, Kundu A, Guibas LJ, Barron JT, Poole B. Learning a Diffusion Prior for NeRFs. *arXiv preprint arXiv:2304.14473*, 2023.

[131] Kalischek N, Peters T, Wegner JD, Schindler K. Tetrahedral Diffusion Models for 3D Shape Generation. *arXiv preprint arXiv:2211.13220*, 2022.

[132] Hui KH, Li R, Hu J, Fu CW. Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers*, 2022, 1–9.

[133] Hu J, Hui KH, Liu Z, Li R, Fu CW. Neural wavelet-domain diffusion for 3d shape generation, inversion, and manipulation. *ACM Transactions on Graphics*, 2024, 43(2): 1–18.

[134] Zheng XY, Pan H, Wang PS, Tong X, Liu Y, Shum HY. Locally attentional sdf diffusion for controllable 3d shape generation. *ACM Transactions on Graphics (ToG)*, 2023, 42(4): 1–13.

[135] Erkoç Z, Ma F, Shan Q, Nießner M, Dai A. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, 14300–14310.

[136] Chu R, Xie E, Mo S, Li Z, Nießner M, Fu CW, Jia J. Diffcomplete: Diffusion-based generative 3d shape completion. *Advances in Neural Information Processing Systems*, 2024, 36.

[137] Ju X, Huang Z, Li Y, Zhang G, Qiao Y, Li H. DiffRoom: Diffusion-based High-Quality 3D Room Reconstruction and Generation. *arXiv preprint arXiv:2306.00519*, 2023.

[138] Sun J, Xie Y, Chen L, Zhou X, Bao H. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, 15598–15607.

[139] Luo S, Hu W. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 2837–2845.

[140] Kong D, Wang Q, Qi Y. A Diffusion-ReFinement Model for Sketch-to-Point Modeling. In *Proceedings of the Asian Conference on Computer Vision*, 2022, 1522–1538.

[141] Zhou L, Du Y, Wu J. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 5826–5835.

[142] Wu Z, Wang Y, Feng M, Xie H, Mian A. Sketch and text guided diffusion model for colored point cloud generation. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, 8929–8939.

[143] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 2017, 30.

[144] Karnewar A, Vedaldi A, Novotny D, Mitra NJ. HOLODIFFUSION: Training a 3D Diffusion Model Using 2D Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 18423–18433.

[145] Karnewar A, Mitra NJ, Vedaldi A, Novotny D. Holofusion: Towards photo-realistic 3d generative modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 22976–22985.

[146] Nakayama GK, Uy MA, Huang J, Hu SM, Li K, Guibas L. Difffacto: Controllable part-based 3d point cloud generation with cross diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 14257–14267.

[147] He X, Chen J, Peng S, Huang D, Li Y, Huang X, Yuan C, Ouyang W, He T. Gvgen: Text-to-3d generation with volumetric representation. *arXiv preprint arXiv:2403.12957*, 2024.

[148] Zhang B, Cheng Y, Yang J, Wang C, Zhao F, Tang Y, Chen D, Guo B. GaussianCube: Structuring Gaussian Splatting using Optimal Transport for 3D Generative Modeling. *arXiv preprint arXiv:2403.19655*, 2024.

[149] Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H, et al.. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[150] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, 1912–1920.

[151] Eppner C, Mousavian A, Fox D. Acronym: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, 6222–6227.

[152] Selvaraju P, Nabail M, Loizou M, Maslioukova M, Averkiou M, Andreou A, Chaudhuri S, Kalogerakis E. BuildingNet: Learning to label 3D buildings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 10397–10407.

[153] Choi S, Zhou QY, Miller S, Koltun V. A Large Dataset of Object Scans. *arXiv:1602.02481*, 2016.

[154] Calli B, Singh A, Walsman A, Srinivasa S, Abbeel P, Dollar AM. The YCB object and Model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, 2015, 510–517, doi:10.1109/ICAR.2015.7251504.

[155] Park K, Rematas K, Farhadi A, Seitz SM. Photoshape: Photorealistic materials for large-scale shape collections. *arXiv preprint arXiv:1809.09761*, 2018.

[156] Collins J, Goel S, Deng K, Luthra A, Xu L, Gundogdu E, Zhang X, Vicente TFY, Dideriksen T, Arora H, et al.. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 21126–21136.

[157] Chen K, Choy CB, Savva M, Chang AX, Funkhouser T, Savarese S. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, 2019, 100–116.

[158] Achlioptas P, Fan J, Hawkins R, Goodman N, Guibas LJ. ShapeGlot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 8938–8947.

[159] Sun X, Wu J, Zhang X, Zhang Z, Zhang C, Xue T, Tenenbaum JB, Freeman WT. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, 2974–2983.

[160] Reizenstein J, Shapovalov R, Henzler P, Sbordone L, Labatut P, Novotny D. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, 10901–10911.

[161] Yu X, Xu M, Zhang Y, Liu H, Ye C, Wu Y, Yan Z, Zhu C, Xiong Z, Liang T, et al.. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, 9150–9161.

[162] Chang A, Dai A, Funkhouser T, Halber M, Niessner M, Savva M, Song S, Zeng A, Zhang Y. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[163] Zhou T, Tucker R, Flynn J, Fyffe G, Snavely N. Stereo Magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2018, 37.

[164] Johnson J, Hariharan B, Van Der Maaten L, Fei-Fei L, Lawrence Zitnick C, Girshick R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, 2901–2910.

[165] Skorokhodov I, Sotnikov G, Elhoseiny M. Aligning latent and image spaces to connect the unconnectable. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 14144–14153.

[166] Baruch G, Chen Z, Dehghan A, Dimry T, Feigin Y, Fu P, Gebauer T, Joffe B, Kurz D, Schwartz A, et al.. ARKitScenes–A Diverse Real-World Dataset For 3D Indoor Scene Understanding Using Mobile RGB-D Data. *arXiv preprint arXiv:2111.08897*, 2021.

[167] Kempka M, Wydmuch M, Runc G, Toczek J, Jaśkowski

W. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE conference on computational intelligence and games (CIG)*, 2016, 1–8.

[168] Straub J, Whelan T, Ma L, Chen Y, Wijmans E, Green S, Engel JJ, Mur-Artal R, Ren C, Verma S, et al.. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[169] Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V. CARLA: An open urban driving simulator. In *Conference on robot learning*, 2017, 1–16.

[170] Xue L, Yu N, Zhang S, Panagopoulou A, Li J, Martín-Martín R, Wu J, Xiong C, Xu R, Niebles JC, et al.. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, 27091–27101.

[171] Choi Y, Uh Y, Yoo J, Ha JW. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, 8188–8197.

[172] Deitke M, Liu R, Wallingford M, Ngo H, Michel O, Kusupati A, Fan A, Laforte C, Voleti V, Gadre SY, et al.. Objaverse-XL: A Universe of 10M+ 3D Objects. *arXiv preprint arXiv:2307.05663*, 2023.

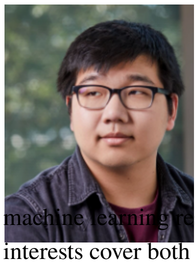
[173] He Y, Bai Y, Lin M, Zhao W, Hu Y, Sheng J, Yi R, Li J, Liu YJ. T^3 Bench: Benchmarking Current Progress in Text-to-3D Generation. *arXiv preprint arXiv:2310.02977*, 2023.

Author biographies

Chen Wang is a Ph.D. student in the University of Pennsylvania. He received his bachelor's and master's degrees in computer science from Tsinghua University, Beijing. His research interests include 3D/4D generation and reconstruction.

Hao-Yang Peng is currently a master's student in the Department of Computer Science and Technology, Tsinghua University. His research interests include computer graphics, 3D understanding, and 3D generation.

Ying-Tian Liu received a B.S. degree in Computer Science and Technology from Tsinghua University in 2020, where he is currently pursuing a Ph.D. degree in the Department of Computer Science and Technology. His research interests include font representation, 3D generation, and diffusion modeling.



Jiatao Gu obtained his Ph.D. degree from the Department of Electrical and Electronic Engineering, University of Hong Kong in 2018. He obtained his bachelor's degree from the Electronic Engineering Department, Tsinghua University in 2014. He is currently a machine learning researcher at Apple AI/ML (MLR). His research interests cover both representation learning and generative models for multiple modalities, including natural languages, images, 3D and speech.

Shi-Min Hu received his Ph.D. degree from Zhejiang University in 1996. He is currently a professor in the Department of Computer Science and Technology, Tsinghua University. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He is the editor-in-chief of Computational Visual Media, and is on the editorial boards of several other journals, including Computer Aided Design and Computer & Graphics. He is a senior member of ACM, and a fellow of IEEE, CCF and SMA.