Real3D: The Curious Case of Neural Scene Degeneration

Dengsheng Chen[1], **Jie Hu**[1], **Xiaoming Wei**[1], **Enhua Wu**[2,3,4]
[1]Meituan
[2]State Key Laboratory of Computer Science, ISCAS
[3]University of Chinese Academy of Sciences
[4]University of Macau
{chendengsheng, hujie39, weixiaoming}@meituan.com, ehwu@um.edu.mo

Abstract

Despite significant progress in utilizing pre-trained text-to-image diffusion models to guide the creation of 3D scenes, these methods often struggle to generate scenes that are sufficiently realistic, leading to "neural scene degeneration". In this work, we propose a new 3D scene generation model called Real3D. Specifically, Real3D designs a pipeline from a NeRF-like implicit renderer to a tetrahedrons-based explicit renderer, greatly improving the neural network's ability to generate various neural scenes. Moreover, Real3D introduces an additional discriminator to prevent neural scenes from falling into undesirable local optima, thus avoiding the degeneration phenomenon. Our experimental results demonstrate that Real3D outperforms all existing state-of-the-art text-to-3D generation methods, providing valuable insights to facilitate the development of learning-based 3D scene generation approaches.

Introduction

3D object generation has recently garnered significant attention in the fields of computer graphics and computer vision, owing to its numerous applications in areas such as gaming, virtual/augmented reality, and 3D printing.

Early approaches for 3D object generation primarily relied on categorical models, wherein the neural network could only synthesize a particular class of objects. However, in artistic contexts such as animation and game development, artists may require the ability to generate a diverse range of objects, posing a significant challenge for categorical models alone. One of the major obstacles to 3D object generation lies in the limited availability and diversity of 3D datasets. Compared to image and video content, 3D datasets are relatively scarce on the internet, primarily due to the exorbitant expenses involved in capturing and rendering 3D objects. Consequently, the existing 3D datasets are often confined to a handful of classes and cannot encompass a broad range of 3D objects, further impeding the 3D content creation process.

In the meantime, the recent advancements in diffusion models (Balaji et al. 2022; Nichol et al. 2021; Ramesh et al. 2022; Saharia et al. 2022a) have spurred significant progress in the generative modeling of images using text prompts, enhancing the creative process of generating image content (Ho, Jain, and Abbeel 2020; Sohl-Dickstein et al. 2015; Song and Ermon 2019). Therefore, recent studies (Lin et al. 2022; Wang et al. 2022; Poole et al. 2022a) try the feasibility of leveraging text-to-image diffusion models (Saharia et al. 2022b; Ho, Jain, and Abbeel 2020) for generating 3D scenes directly from text descriptions. These studies have demonstrated remarkable abilities in producing text-conditioned 3D content. The diffusion model functions as a critic to refine the 3D neural scenes, which can be represented by either implicit or explicit neural networks, such as a Neural Radiance Field, or NeRF (Gu et al. 2021). This refinement process involves using a score function to ensure that the rendered images align with the photo-realistic image distribution from different viewpoints, given an input text prompt. However, the neural scenes are prone to get stuck in certain undesirable local optima, which fully satisfy the description of the given prompt but lead to unnatural-looking scenes or "neural scene degeneration", as depicted in Fig. 2.

The challenge of scene degeneration poses a significant obstacle in effectively utilizing prior knowledge networks to enhance the quality and realism of 3D neural scenes. In this work, we use a discriminator to effectively address neural scene degeneration. This is a novel approach that has not been previously explored. Furthermore, we introduce Real3D, a dual-renderer text-to-3D generation model that combines the advantages of implicit and explicit rendering techniques. Our experimental results demonstrate that Real3D outperforms all existing state-of-the-art text-to-3D generation methods. Overall, our work offers valuable insights for the development of learning-based 3D scene generation approaches.

Related Work

Text-to-Image Generation. Text-to-image generation has witnessed significant advancements in recent years, owing to diffusion models (Balaji et al. 2022; Ramesh et al. 2022). These models integrate rich semantic concepts from the text, such as nouns, adjectives, and artistic styles, to generate highly realistic images of objects and scenes with exceptional detail and quality. Nevertheless, the image sampling process with diffusion models can be computationally expensive. To generate high-resolution images, state-of-the-art models use a cascade of super-resolution models (Saharia et al. 2022a) or employ a sampling strategy from a lower-resolution latent space, which is then decoded into high-resolution images (Rombach et al. 2022). Despite advancements in generat-

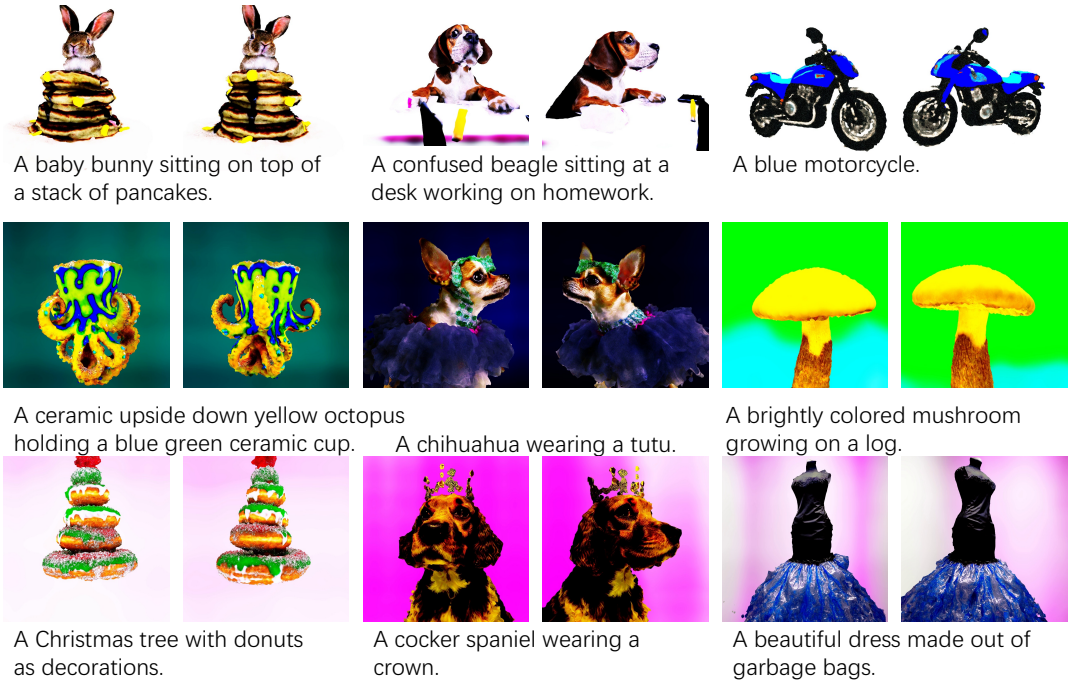


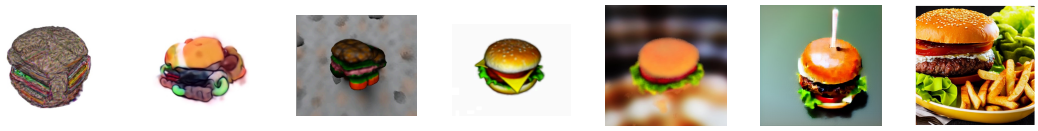
Figure 1: Real3D combines implicit and explicit rendering methods and utilizes a text-to-image diffusion model along with a discriminator to generate high-fidelity 3D models from text prompts.

ing high-resolution images, the problem of utilizing language to describe and control 3D properties, such as camera viewpoints, while preserving the 3D coherency of the generated images, remains a challenging and open research problem.

3D Neural Scene Representation. 3D generative modeling has been extensively researched, focusing on exploring different 3D neural scene representations, such as voxel grids (Lunz et al. 2020), point clouds (Zeng et al. 2022), meshes (Gao et al. 2022), implicit (Chen and Zhang 2019), and octree (Ibing, Kobsik, and Kobbelt 2021) representations. Recent studies have been influenced by the success of neural volume rendering (Mildenhall et al. 2021a), leading to a growing interest in 3D-aware image synthesis (Chan et al. 2022; Or-El et al. 2022; Schwarz et al. 2022). This technique offers several advantages, chief among them being the capacity to learn 3D generative models from images, which are more universally available. The neural radiance fields (Mildenhall et al. 2021b) (NeRF) technique for inverse rendering has achieved notable advancements in multi-view 3D reconstruction, particularly in surface geometry estimation and view synthesis (Martin-Brualla et al. 2021; Oechsle, Peng, and Geiger 2021; Yariv et al. 2021; Wang et al. 2021). In essence, NeRF represents a 3D asset through a dense grid of RGB colors and spatial density similar to that of alpha compositing (Max 1995). Although the volume is parameterized with a neural network, the dense 3D queries required for this system can be computationally expensive. An encouraging substitute for the computationally expensive dense 3D queries required by NeRF is the Voxel NeRF approach (Liu et al. 2020; Chen et al. 2022), which stores the volume as voxels and has demonstrated no deterioration in end task performance (Sun, Sun, and Chen 2022; Yu et al. 2021). Voxels are much less complex to query as they solely need a memory operation, unlike the feed-forward pass of a neural network.

Text-to-3D Scene Creation. In recent years, there has been noteworthy advancement in text-to-image generative modeling, which has instigated a growing interest in text-to-3D generation within the machine learning community. One earlier approach is CLIP-forge (Sanghi et al. 2022), which employs a normalizing flow model to sample shape embeddings from textual input. However, this approach necessitates 3D assets in voxel representations during training, which poses a challenge to scalability with data. Two recent works, namely DreamField (Jain et al. 2022b) and CLIP-mesh (Khalid et al. 2022b), address the issue of training data by utilizing a pre-trained image-text model (Radford et al. 2021a) to optimize the underlying 3D representations (NeRFs and meshes) such that all 2D renderings achieve high text-image alignment scores. In the realm of 3D synthesis, relying exclusively on pre-trained large-scale image-text models instead of costly 3D training data has become a popular methodology. One of the main limitations of text-to-3D synthesis models is the sub-optimal quality of the 2D renderings they often produce, which can lead to scene degeneration.

Recently, significant progress has been made to mitigate this issue through innovative approaches such as DreamFusion (Poole et al. 2022b), JSC (Wang et al. 2022), and Magic3D (Lin et al. 2022). These models leverage a powerful pre-trained text-to-image diffusion model (Saharia et al. 2022a) as a robust image prior to achieving remarkable text-



(a) (b) (c) (d) (e) (f) (g)

Figure 2: The cases of neural scene degeneration. Given a natural language description, such as "a realistic hamburger", a text-to-image diffusion model can effectively synthesize highly realistic images, as demonstrated in (g). These synthetic images are separately from (a) CLIPMesh (Khalid et al. 2022a), (b) DreamFields (Jain et al. 2022a), (c) DreamFields (reimpl. by (Poole et al. 2022a)), (d) DreamFusion (Poole et al. 2022a), (e) DreamFusion (our reimpl.), (f) SJC (Wang et al. 2022), (g) Stable diffusion model. However, when using pre-trained models, such as diffusion models and CLIP, as prior knowledge networks to guide the generation of 3D neural scenes, they fail to produce satisfactory results. Regardless of the loss function used (e.g. the logarithmic likelihood loss function in (a), score-based maximization function in (b) - (f)), all of the generated neural scenes are highly susceptible to degenerating in an unrealistic and toy-like manner. This phenomenon, commonly referred to as *neural scene degeneration*, results in images that are distant from their synthetic counterparts.

to-3D synthesis results. However, these approaches have not yet completely resolved the challenge of scene degeneration, and despite notable improvements in generation quality, this issue remains a persistent challenge.

Background

The diffusion model, one of the latent-variable generative models, gradually transforms a sample from a noise distribution to a data distribution by removing the structure from data through the forward process q and adding structure through the reverse process p (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020). In the forward process, noise is added to the previous latent at timestep t, and a Gaussian distribution is used to transition to a noisier latent at timestep $t+1$. By integrating intermediate timesteps, we can compute the marginal distribution of the latent variable at timestep t given an initial datapoint $\mathbf{x}$: $q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\alpha_t\mathbf{x}, \sigma_t^2\mathbf{I})$. The smoothed versions of the data distribution can be obtained by integrating out the data density $q(\mathbf{x})$ to compute the marginals $q(\mathbf{z}_t) = \int q(\mathbf{z}_t|\mathbf{x})q(\mathbf{x})d\mathbf{x}$. To ensure $q(\mathbf{z}_t)$ is close to the data density at the start of the process ($\sigma_0 \sim 0$) and close to Gaussian at the end of the forward process ($\sigma_T \sim 1$), the coefficients α_t and σ_t are chosen with $\alpha_t^2 = 1 - \sigma_t^2$ to preserve variance (Kingma et al. 2021; Song et al. 2020).

Training the generative model with a (weighted) evidence lower bound (ELBO) simplifies to a weighted denoising score matching objective for parameters ϕ (Ho, Jain, and Abbeel 2020; Kingma et al. 2021):

$$\mathcal{L}_{Diff}(\phi, \mathbf{x}) = \mathbb{E}_{t,\epsilon}\left[w(t)||\epsilon_\phi(\alpha_t\mathbf{x} + \sigma_t\epsilon; t) - \epsilon||_2^2\right], \quad (1)$$

where $t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $w(t)$ is a weighting function that depends on the timestep t. Diffusion model training can thereby be viewed as either learning a latent-variable model (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020), or learning a sequence of score functions corresponding to noisier versions of the data (Vincent 2011; Song and Ermon 2019; Song et al. 2020). We will use $p_\phi(\mathbf{z}_t; t)$ to denote the approximate marginal distribution whose score function is given by $\mathbf{s}_\phi(\mathbf{z}_t; t) = -\epsilon_\phi(\mathbf{z}_t; t)/\sigma_t$.

We have built upon text-to-image diffusion models that have the ability to learn $\epsilon_\phi(\mathbf{z}_t; t, y)$ by conditioning it on text embeddings y, as introduced in recent papers such as Saharia et al.; Ramesh et al.; Nichol et al..

The Curious Case of Neural Scene Degeneration

In this section, we provide a formalized description of text-to-3D generation using score distillation sampling (Poole et al. 2022a). Then, we conduct a deeper analysis of the issue of neural scene degeneration that inevitably arises within this paradigm and expound on the role that the discriminator plays in resolving this challenge. This leads us to a comprehensive understanding of the text-to-3D generation process.

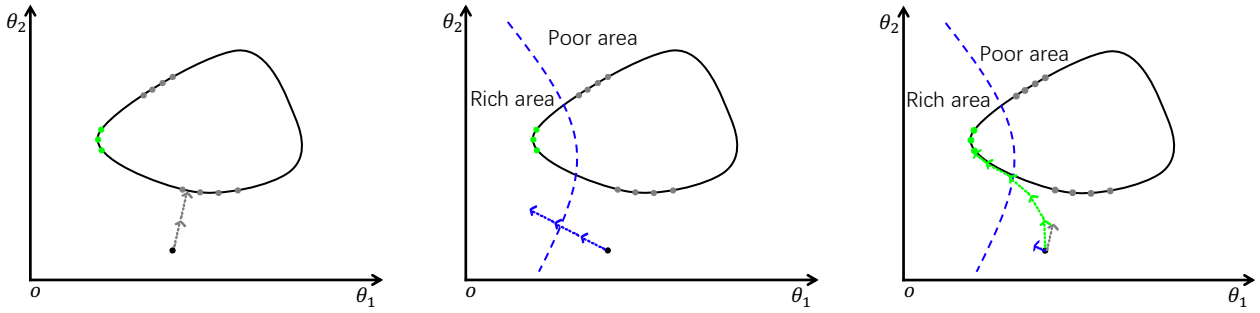
Prior-Knowledge-Based Text-to-3D Generative Models The **key idea** to sampling 3D scenes from diffusion models is to create neural scenes that look like good images when rendered from a random angle v. Such models can be specified as a differentiable image paramererization (DIP (Mordvintsev et al. 2018)), where a differential renderer g transforms neural scenes parameterized by θ to create an image $\mathbf{x}' = g_\theta(v)$. Here, we adapt the score distillation sampling as a loss function that optimizes over parameters θ such that $\mathbf{x}'$ looks like a sample from the frozen diffusion model p_ϕ. The diffusion model ϕ comes with a learned denoising function $\epsilon_\phi(\mathbf{x}'_t; y, t)$ that predicts the sampled noise ϵ given the noisy image $\mathbf{x}'_t$, noise level t, and text embedding y. It provides the gradient direction to update θ such that all rendered images are pushed to the high probability density regions conditioned on the text embedding under the diffusion prior, as shown in Fig. 3(a).

Specifically, score distillation sampling, which computes the gradient:

$$\nabla_\theta\mathcal{L}_{SDS}(\phi, \theta) = \mathbb{E}_{t,\epsilon}\left[w(t)(\epsilon_\phi(\mathbf{x}'_t; y, t) - \epsilon)\frac{\partial\mathbf{x}}{\partial\theta}\right]. \quad (2)$$

The differential renderer g can be viewed as a modular component of the framework, amenable to choice.

Neural Scene Degeneration In fact, Eq. 2 can be viewed as the gradient of a weighted probability density distillation loss (Oord et al. 2018), which utilizes the scored functions learned from the diffusion model:



(a) Score distillation sampling (b) Adversarial training (c) Real3D

Figure 3: Viewing the problem of neural scene degeneration from the perspective of parameter optimization. Here we illustrate two parameters for clarity. (a): Score distillation sampling (SDS) can drive θ to rapidly approach and eventually reach p_ϕ, but it could not further force θ to find a rich point along p_ϕ as the distribution of rich points on p_ϕ is very sparse. (b): While adversarial training can drive θ to a rich point, it fails to make θ belong to a specified content and usually causes *neural scene distortion*. (c): Real3D achieves a good balance by seamlessly continuing the search for a rich point along p_ϕ even after reaching one. (Dash arrow: optimization direction, Black dot: initial position, Gray dot: poor point, Green dot: rich point, Black solid line: p_ϕ, Blue dash line: the boundary between a rich area and poor area.)

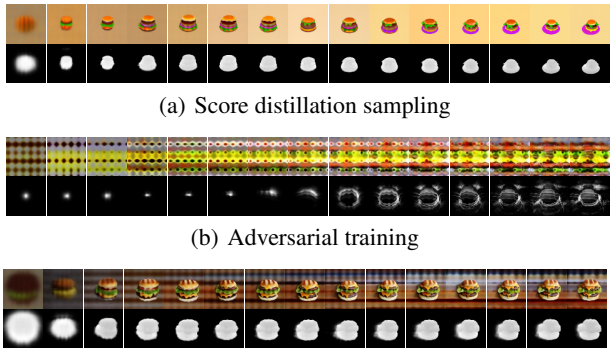
$$\nabla_\theta \mathcal{L}_{SDS}(\phi, \theta) = \nabla_\theta \mathbb{E}_t \left[\sigma_t/\alpha_t w(t) \mathrm{KL}(q(\mathbf{z}_t|g_\theta(v); y, t)||p_\phi(\mathbf{z}_t; y, t))\right]. \tag{3}$$

By updating θ using the gradient above, we can gradually make $\mathbf{x}' \sim p_\phi$. In addition, Eq. 1 helps p_ϕ to approach the distribution p_{data}. So, existing prior knowledge-based generative models assume that g_θ will eventually converge to p_{data}, resulting in the generated samples $\mathbf{x}'$ being similar to real data.

However, a crucial question remains unanswered: *is making $g_\theta \to p_\phi$ equivalent to making $g_\theta \to p_{data}$?*

The answer is an unequivocal no, and this underlying issue is a key factor in neural scene degeneration. Despite training the diffusion model with a large volume of data, it is unfortunate to acknowledge that p_ϕ cannot be deemed flawless. Hence, it cannot be assumed that all images $\tilde{\mathbf{x}}$ sampled from p_ϕ would be good samples. As demonstrated in Fig. 3(a), some of these samples might be outliers or not as high-quality as others. This could be attributed to various factors such as inappropriate random seeds or sample parameters, or the presence of erroneous inputs which might lead to the generation of suboptimal synthetic images. That means that *the score function fails to distinguish g_θ from rich and poor points* since they are all sampled from p_ϕ. This implies that when $\mathbf{x}'$ falls into the category of poor points, as tends to occur during the initial stages of optimization, the gradient obtained in Eq. 2 is no longer effective in updating θ and driving $\mathbf{x}' = g_\theta(v)$ towards rich points anymore. As a result, neural scene degeneration becomes an unsolvable problem under this regime.

As shown in Fig. 2, most existing prior-knowledge-based text-to-3D methods suffer severe scene degeneration problems, no matter how the loss function was designed or which prior knowledge network was used.



(a) Score distillation sampling (b) Adversarial training (c) Real3D

Figure 4: An illustration of evolutionary process among different methods. The images were rendered after every 1000 training iteration.

Adversarial Training Existing research has predominantly centered on enhancing loss functions or incorporating stronger prior knowledge networks to guide scene generation. The core objective of such approaches is to enhance the model's ability to transition toward rich points directly from initial points. However, these solutions prove ineffective when the model unintentionally encounters poor points, resulting from sub-optimal initialization parameters or unfavorable prompts. In such scenarios, these techniques fail to steer the model back towards the rich points, often leading to neural scene collapses, as depicted in Fig. 4(a). To overcome this challenge, we propose exploring alternative gradient calculation methods, distinct from the score function, to mitigate the problem of neural scene degeneration.

As mentioned previously, the score function fails to differentiate between rich and poor points in p_ϕ, resulting in degradation. To address this issue, a discriminator is used to

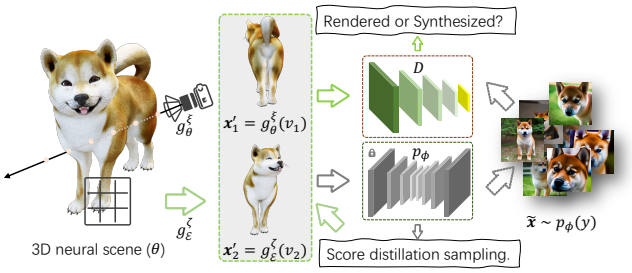


Figure 5: Real3D employs two renderers, one implicit and one explicit. The implicit renderer g^ξ_θ performs ray tracing sampling, which generates the image $\mathbf{x}'_1$. On the other hand, the explicit renderer $g^\zeta_\mathcal{E}$ samples around the mesh surface in parallel to produce the image $\mathbf{x}'_2$. Both $\mathbf{x}'_1$ and $\mathbf{x}'_2$ are then fed into a frozen stable diffusion model p_ϕ and a discriminator D. The gradient is then backpropagated along the green arrow to update θ.

guide the neural scene away from poor points through the following equation:

$$\min_{g_\theta} \max_{D} = \mathbb{E}_{\tilde{\mathbf{x}} \sim p_\phi(y)}[\log D(\tilde{\mathbf{x}})] + \mathbb{E}_{\mathbf{x}' \sim g_\theta(v)}[\log(1 - D(\mathbf{x}')] \tag{4}$$

where $\tilde{\mathbf{x}}$ is the high-quality synthesized image that comes from the rich points.

By optimizing Equation 4, the discriminator network can differentiate between high-quality synthetic images $\tilde{\mathbf{x}}$ and rendered images $\mathbf{x}'$. This process will continue to push g_θ toward rich areas as shown in Fig. 3(b).

While the discriminator can ensure that the generated neural scene moves towards the richer areas, i.e., $g_\theta \sim p_{\tilde{\mathbf{x}}}$, it cannot guarantee strict adherence to the prior knowledge of the network, i.e., $g_\theta \sim p_\phi$. This can result in serious distortion, as illustrated in Fig. 4(b). The ideal situation would be to update g_θ by minimizing Eq. 4 to avoid neural scene **degeneration** and update g_θ by Eq. 2 to avoid neural scene **distortion**. By doing so, g_θ can explore richer points after reaching a poorer point in the early stage, as shown in Fig. 3(c). Fig. 4(c) vividly demonstrates that a suitable balance between score distillation sampling and adversarial training can help g_θ reach richer areas and alleviate neural scene degeneration.

It is worth noting that the problem of scene degeneration is widely prevalent, and the discriminator strategy can be applied to other scenarios as well. In supplementary materials, we provide a detailed exposition of how to balance the impact of score distillation sampling and adversarial training on the neural scene during training.

Real3D

Relying solely on score distillation sampling and adversarial training is not enough to generate high-quality 3D scenes. While the discriminator plays an important role in ensuring that the rendered images $\mathbf{x}' = g_\theta(v)$ closely resemble rich points, it is not enough to guarantee that the underlying neural scenes are of sufficient detail and quality. To generate high-quality 3D models (meshes), it is therefore critical to design an effective framework.

To generate high-quality 3D models, we present Real3D, a text-to-3D generative model that incorporates a dual-renderer approach, as illustrated in Fig. 5. Unlike other existing methods, such as Magic3D (Lin et al. 2022), which adopt a two-stage coarse-to-fine framework, Real3D utilizes both an implicit renderer g^ξ_θ to generate a low-resolution image and an explicit renderer $g^\zeta_\mathcal{E}$ (where $\mathcal{E}$ is the explicit representation of θ) to generate a high-resolution image simultaneously from the same neural scene parameterized by θ.

Implicit Renderer g^ξ_θ We adapt Neural Radiance Filed, i.e., NeRF (Mildenhall et al. 2021b), as implicit renderer g^ξ_θ. NeRF is a technique for neural inverse rendering that consists of a volumetric raytracer and a multilayer perceptron (MLP). Rendering an image from a NeRF is done by casting a ray for each pixel from a camera's center of projection through the pixel's location in the image plane and out into the world. *Sampled 3D points μ_i along each ray* are then passed through an MLP, which produces 4 scalar values as output:

$$(\tau_i, c_i) = g^\xi_\theta(\mu_i) \tag{5}$$

where a volumetric density $\tau_i \in \mathbb{R}$ (how opaque the scene geometry at that 3D coordinate is) and an RGB color $c_i \in \mathbb{R}^3$. These densities and colors are then alpha-composited from the back of the ray towards the camera, producing the final rendered RGB value for the pixel:

$$\mathbf{C} = \sum_i w_i \mathbf{c}_i \tag{6}$$

$$w_i = \alpha_i \Pi_{j<i}(1 - \alpha_j) \tag{7}$$

$$\alpha_i = 1 - \exp(-\tau_i \parallel \mu_i - \mu_{i+1} \parallel) \tag{8}$$

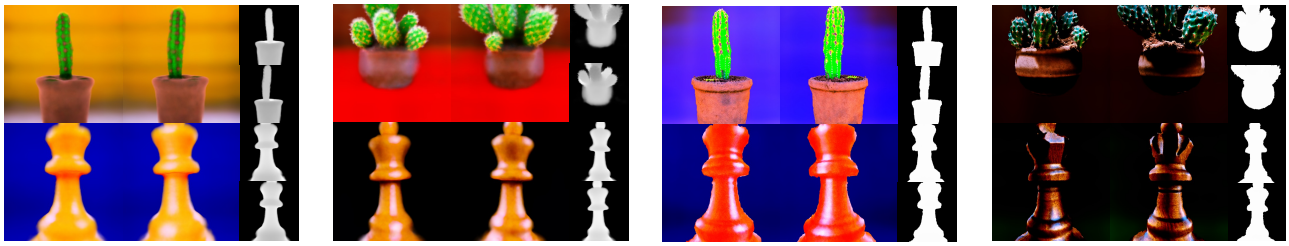
Explicit Renderer $g^\zeta_\mathcal{E}$ We utilize DMTET (Shen et al. 2021) as an explicit renderer denoted as g^ζ_θ. The core of DMTET contains a deformable tetrahedral grid Δ_{u_i} that discretizes a signed distance function and a differentiable marching tetrahedra layer ℓ that transforms the implicit signed distance representation to an explicit surface mesh representation. To render an image using DMTET, we rasterize the explicit surface mesh for each pixel from the camera's center of projection through the pixel's location in the image plane and out into the world. We can build the explicit surface mesh $\mathcal{E}$ by passing all valid u to the differentiable marching tetrahedra layer ℓ with the help of Eq. 5:

$$\mathcal{E} = \ell(\left\{\tau_i = g^\xi_\theta(u_i + \Delta_{u_i}); \tau_i > \varepsilon\right\}) \tag{9}$$

where we set ε as the threshold to filter out the empty voxels. We can obtain the final rendered RGB value for a specified pixel from view v using a differential rasterizing operation:

$$\mathbf{C} = g^\zeta_\mathcal{E}(v) \tag{10}$$

Advantages and Disadvantages of Implicit and Explicit Renderers On one hand, the approach of rendering images directly from implicit structures, such as NeRF, offers greater



(a) g_θ^ξ (b) g_θ^ξ, D (c) $g_\theta^\xi, g_\mathcal{E}^\zeta$ (d) $g_\theta^\xi, g_\mathcal{E}^\zeta, D$

Figure 6: Ablation study. (a) Baseline: DreamFusion which uses *Implicit Rendering (NeRF)* and score distillation sampling. (b) Baseline with *Discriminator*: It makes the generated 3D scene more realistic (although still lacking delicate details and contours, thus overall somewhat blurry). (c) Baseline with *Explicit Rendering*: It produces a very clear 3D scene but are less realistic. (d) Our Real3D (Baseline with *Discriminator* and *Explicit Rendering*): It can not only produce clear contours and details, but also provide highly realistic and faithful images.

flexibility in creating 3D scene content from scratch. This method can calculate the scene content at any location in space, allowing for the creation of new 3D scenes with ease. However, this approach suffers from difficulties in focusing on the surface of some objects, leading to often blurry 3D scene generation, as shown in Fig. 6(a). Though adversarial training can help to generate more realistic scenes, as shown in 6(b), it still failed to obtain a well enough mesh.

On the other hand, the method of generating images directly from the surface of the mesh through rasterizing can provide high-resolution images with clear visibility of contours and other small details, as shown in Fig. 5. It can optimize local details of existing structures with ease. However, it may not be suitable for creating 3D scenes from scratch.

To overcome the limitations of each approach and obtain more realistic and high-fidelity 3D scenes, Real3D combined g_θ^ξ and $g_\mathcal{E}^\zeta$ with score distillation sampling and adversarial training. The loss function can be described as:

$$\begin{aligned}\mathcal{L} =&\lambda_{SDS}(\mathcal{L}_{SDS}(g_\theta^\xi(v_1)) + \mathcal{L}_{SDS}(g_\mathcal{E}^\zeta(v_2)))+ \\ &\lambda_{adv}(\mathcal{L}_{adv}(g_\theta^\xi(v_1), \tilde{\mathbf{x}}_1) + \mathcal{L}_{adv}(g_\mathcal{E}^\zeta(v_2), \tilde{\mathbf{x}}_2))\end{aligned} \tag{11}$$

By adding explicit control over the geometry of the scene, generating higher-quality 3D geometry, and training a discriminator against the generated data, both the global and local consistency of generated scenes can be improved. As a result of this integration, we can obtain high-fidelity 3D scenes with detailed textures, realistic lighting, and authentic geometry, as shown in Fig. 6(c) and Fig. 6(d).

Experiments

We focus on comparing our method with existing text-to-3D methods, i.e., DreamFusion (Poole et al. 2022a), SJC (Wang et al. 2022) and Magic3D (Lin et al. 2022) on the text prompts taken from the online website[1]. Many implementation details play a crucial role in generating high-quality 3D models, which can be found in the supplementary materials.

Implementation Details. We use Adan (Xie et al. 2022) optimizer with an initial learning rate of 5×10^{-3}. A cosine annealing learning rate scheduler is applied with a minimum learning rate of 1×10^{-6}. In all experiments, we set $\lambda_{SDS} = 1$ and $\lambda_{adv} = 1 \times 10^{-2}$. Despite that, some geometry regularization described in Poole et al. also applied. To generate better 3D scenes, we perform experiments with multiple sets of random seed values. The SDF threshold ε is set as 0.1. We use p_ϕ to generate 1000 images as $\tilde{\mathbf{x}}$ for each prompt in advance. In Poole et al., it is claimed that converting part of the viewpoint information into text descriptions such as "front" and "back" and adding them to the prompt can improve the effectiveness. Here, we go further and use positional prompt embedding. Specifically, we interpolate different prompt embeddings for different front views based

[1]https://dreamfusion3d.github.io/gallery.html

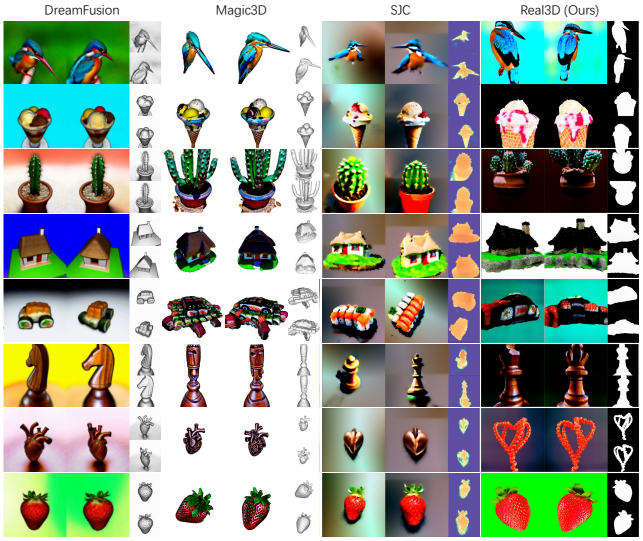


Figure 7: Qualitative example comparison among different methods. Prompts: a kingfisher bird, an ice cream sundae, a small cactus planted in a clay pot, a 3D model of an adorable cottage with a thatched roof, a car made of sushi, a beautifully carved wooden knight chess piece, a very beautiful tiny human heart organic sculpture made of copper wire and threaded pipes, a ripe strawberry.

	Rank	DF (%)	Magic3D (%)	SJC (%)	Real3D (%)
Official	@1	6.2	25.0	16.7	52.1
	@2	18.8	35.4	20.8	25.0
	@3	41.7	29.2	20.8	8.3
	@4	33.3	10.4	41.7	14.6
Unoff.	@1	4.5	21.6	17.0	56.8
	@2	18.2	38.6	19.3	23.9
	@3	42.0	29.5	21.6	6.8
	@4	35.2	10.2	42.0	12.5

Table 1: User preference study. In the upper four rows, we compared eight different 3D models generated using either official sources or official open-source code. As some algorithms do not have publicly-available 3D models or open-source code, we compared the performance of various methods int the rest four rows using replicated results from their original papers. We randomly selected 100 prompts from the 397 prompts provided by DreamFusion for generation and ensured that each prompt generated reasonable results for all algorithms.(DF indicates DreamFusion.)

	DF	Magic3D	SJC	Real3D
Score	25.27($\pm$5.82)	27.50($\pm$3.07)	27.21($\pm$4.00)	**27.97**($\pm$**2.91**)

Table 2: Image text similarity computed by CLIP-L (Radford et al. 2021b).(DF indicates DreamFusion.)

on the angle between the current view and the front view to obtain positional prompt embedding. Although we use both g_θ^ξ and $g_\mathcal{E}^\zeta$ at the same time, to ensure that the neural scene can converge more stably to a fixed rich point during the learning process, we use g_θ^ξ more frequently to render images in the early stage of training. This is to quickly generate meaningful geometric structures in the neural scene. In the later stage of training, we gradually stop using g_θ^ξ and use $g_\mathcal{E}^\zeta$ more to further optimize various aspects of the existing geometric structures.

Speed Evaluation. To evaluate the speed of our neural scene updates, we executed $10,000$ iterations with a batch size of 2, which included both an implicitly rendered image $\mathbf{x}'_1$ and an explicitly rendered image $\mathbf{x}'_2$. Experimentally, we found that the primary content of the neural scene is typically determined within the first few thousand iterations. The subsequent iterations mainly focus on optimizing some minor details. The entire process was carried out on a single A100 GPU, and it took approximately 25 minutes to complete.

Qualitative Comparisons. We present qualitative examples in Fig. 7. Compared to other methods, Real3D generates more realistic and intricate 3D models. More examples can be found in Fig. 1 and the supplementary materials.

Quantitative Comparisons. In order to provide quantitative evaluation metrics, we utilized the CLIP model to calculate the image-text similarity between the images rendered by the neural scene and the given prompts. As shown in Tab. 2, Real3D achieved the highest similarity score. It is worth noting that Real3D also achieved the smallest variance, indicating its ability to consistently generate better results, which is crucial in a controllable generation.

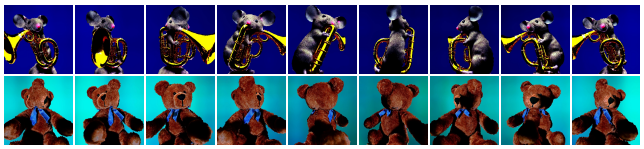


Figure 8: Failure cases and multi-face issues. Although Real3D can generate highly realistic and detailed 3D scenes, there are still some failure cases. We have carefully observed some scenes with poor generation results and found that the main problem is the presence of "multi-face" in the generated scenes. Specifically, "multi-face" refers to the phenomenon where some generated scenes appear very reasonable when viewed from a single angle, but when viewed as a whole, many repetitive structures appear, which makes the overall model illogical. As shown in the above figure, multiple faces appear when generating animals or plush toys. It is worth noting that "multi-face" is not limited to repeated facial structures and can also occur in other organs or tissues.

User Studies. Although image-text similarity can to some extent reflect the degree of matching between 3D scenes and text, it cannot effectively reflect more subjective visual effects such as blurriness, structural rationality, or aesthetics. To address this issue, we conducted user studies to evaluate the effectiveness of various methods based on user preferences. Specifically, we presented participants with four videos rendered from a canonical view using different algorithms, all generated from the same text prompt, and asked them to rank the videos based on realism and level of detail. Each prompt was evaluated by six users. The results, presented in Fig. 1, indicate that the majority of users preferred 3D models generated by Real3D, with over 50% of users rating our method as having the best quality.

Limitations

We adopt the latent stable diffusion (Rombach et al. 2021) as a prior knowledge network, which, as discussed in other works, also suffers from the issue of multi-face in many scenes generated during the training process, though the scenes themselves are highly realistic. This problem, as shown in Fig. 8, is related to some challenging prompts.

To address this issue, besides designing more reasonable prompts to avoid multi-face, we found that rendering two views with a certain angle offset using g_θ^ξ and $g_\mathcal{E}^\zeta$ respectively, can increase the field of view and alleviate the frequency of multi-face to some extent.

Conclusion

In this work, we propose a novel approach that combines adversarial training and distillation sampling to mitigate the issue of neural scene degeneration. By integrating the strengths of explicit and implicit rendering methods, the proposed method, *i.e.* Real3D, achieves promising results. However, we acknowledge that the performance of Real3D may be hindered by the quality of prior knowledge networks, which may impact its ability to generate high-quality scenes for certain prompts.

Ethics Statement

As the authors of this research paper on 'Real3D', we affirm that our work adheres to the highest ethical standards. All methods and procedures were conducted with integrity and transparency. The 'Real3D' methodology, while capable of generating 3D models from text prompts, is not designed or intended to be used for the production of harmful, illegal, or unethical content. We recognize the potential for misuse of this technology and strongly discourage such applications. Furthermore, we respect and uphold the principles of intellectual property rights. The text prompts used in our research were either created by us or sourced from the public domain, with due credits given where necessary. We also ensure that the generated 3D models do not infringe upon any existing copyrights or trademarks. We commit to using our research and its findings responsibly, intending to contribute positively to the field of 3D modeling and its potential applications. We believe that technology should serve society, and we strive to ensure that our work reflects this belief.

References

Balaji, Y.; Nah, S.; Huang, X.; Vahdat, A.; Song, J.; Kreis, K.; Aittala, M.; Aila, T.; Laine, S.; Catanzaro, B.; et al. 2022. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*.

Chan, E. R.; Lin, C. Z.; Chan, M. A.; Nagano, K.; Pan, B.; De Mello, S.; Gallo, O.; Guibas, L. J.; Tremblay, J.; Khamis, S.; et al. 2022. Efficient geometry-aware 3D generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16123–16133.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. TensoRF: Tensorial Radiance Fields. *arXiv preprint arXiv:2203.09517*.

Chen, Z.; and Zhang, H. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5939–5948.

Gao, J.; Shen, T.; Wang, Z.; Chen, W.; Yin, K.; Li, D.; Litany, O.; Gojcic, Z.; and Fidler, S. 2022. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*.

Gu, J.; Liu, L.; Wang, P.; and Theobalt, C. 2021. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*.

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

Ibing, M.; Kobsik, G.; and Kobbelt, L. 2021. Octree Transformer: Autoregressive 3D shape generation on hierarchically structured sequences. *arXiv preprint arXiv:2111.12480*.

Jain, A.; Mildenhall, B.; Barron, J. T.; Abbeel, P.; and Poole, B. 2022a. Zero-Shot Text-Guided Object Generation with Dream Fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Jain, A.; Mildenhall, B.; Barron, J. T.; Abbeel, P.; and Poole, B. 2022b. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 867–876.

Khalid, N.; Xie, T.; Belilovsky, E.; and Popa, T. 2022a. CLIP-mesh: Generating textured meshes from text using pretrained image-text models. *ACM Trans. Graph.*

Khalid, N. M.; Xie, T.; Belilovsky, E.; and Popa, T. 2022b. Clip-mesh: Generating textured meshes from text using pretrained image-text models. *arXiv preprint arXiv:2203.13333*.

Kingma, D.; Salimans, T.; Poole, B.; and Ho, J. 2021. Variational diffusion models. *Advances in neural information processing systems*, 34: 21696–21707.

Lin, C.-H.; Gao, J.; Tang, L.; Takikawa, T.; Zeng, X.; Huang, X.; Kreis, K.; Fidler, S.; Liu, M.-Y.; and Lin, T.-Y. 2022. Magic3D: High-Resolution Text-to-3D Content Creation. *arXiv preprint arXiv:2211.10440*.

Liu, L.; Gu, J.; Zaw Lin, K.; Chua, T.-S.; and Theobalt, C. 2020. Neural sparse voxel fields. In *Adv. Neural Inform. Process. Syst.*

Lunz, S.; Li, Y.; Fitzgibbon, A.; and Kushman, N. 2020. Inverse graphics gan: Learning to generate 3d shapes from unstructured 2d data. *arXiv preprint arXiv:2002.12674*.

Martin-Brualla, R.; Radwan, N.; Sajjadi, M. S.; Barron, J. T.; Dosovitskiy, A.; and Duckworth, D. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Max, N. 1995. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021a. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021b. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*.

Mordvintsev, A.; Pezzotti, N.; Schubert, L.; and Olah, C. 2018. Differentiable image parameterizations. *Distill*, 3(7): e12.

Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Oechsle, M.; Peng, S.; and Geiger, A. 2021. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Int. Conf. Comput. Vis.*

Oord, A.; Li, Y.; Babuschkin, I.; Simonyan, K.; Vinyals, O.; Kavukcuoglu, K.; Driessche, G.; Lockhart, E.; Cobo, L.; Stimberg, F.; et al. 2018. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, 3918–3926. PMLR.

Or-El, R.; Luo, X.; Shan, M.; Shechtman, E.; Park, J. J.; and Kemelmacher-Shlizerman, I. 2022. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13503–13513.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022a. DreamFusion: Text-to-3D Using 2D Diffusion. *arXiv preprint arXiv:2209.14988*.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022b. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021a. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021b. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2021. High-Resolution Image Synthesis with Latent Diffusion Models.

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

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S. K. S.; Ayan, B. K.; Mahdavi, S. S.; Lopes, R. G.; et al. 2022a. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S. K. S.; Ayan, B. K.; Mahdavi, S. S.; Lopes, R. G.; et al. 2022b. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487*.

Sanghi, A.; Chu, H.; Lambourne, J. G.; Wang, Y.; Cheng, C.-Y.; Fumero, M.; and Malekshan, K. R. 2022. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18603–18613.

Schwarz, K.; Sauer, A.; Niemeyer, M.; Liao, Y.; and Geiger, A. 2022. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. *arXiv preprint arXiv:2206.07695*.

Shen, T.; Gao, J.; Yin, K.; Liu, M. Y.; and Fidler, S. 2021. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Song, Y.; and Ermon, S. 2019. Generative Modeling by Estimating Gradients of the Data Distribution. In *Adv. Neural Inform. Process. Syst.*

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Sun, C.; Sun, M.; and Chen, H.-T. 2022. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Vincent, P. 2011. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674.

Wang, H.; Du, X.; Li, J.; Yeh, R. A.; and Shakhnarovich, G. 2022. Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation. *arXiv preprint arXiv:2212.00774*.

Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; and Wang, W. 2021. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*.

Xie, X.; Zhou, P.; Li, H.; Lin, Z.; and Yan, S. 2022. Adan: Adaptive Nesterov Momentum Algorithm for Faster Optimizing Deep Models. *arXiv preprint arXiv:2208.06677*.

Yariv, L.; Gu, J.; Kasten, Y.; and Lipman, Y. 2021. Volume rendering of neural implicit surfaces. In *Adv. Neural Inform. Process. Syst.*

Yu, A.; Fridovich-Keil, S.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2021. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*.

Zeng, X.; Vahdat, A.; Williams, F.; Gojcic, Z.; Litany, O.; Fidler, S.; and Kreis, K. 2022. LION: Latent Point Diffusion Models for 3D Shape Generation. *arXiv preprint arXiv:2210.06978*.