MATLABER: Material-Aware Text-to-3D via LAtent BRDF auto-EncodeR

Xudong Xu[†]   **Zhaoyang Lyu**[†]   **Xingang Pan**[§]   **Bo Dai**[†]

[†]Shanghai AI Laboratory [§]S - Lab, Nanyang Technological University

[†]{xuxudong, lvzhaoyang, daibo}@pjlab.org.cn [§]xingang.pan@ntu.edu.sg

Abstract

Based on powerful text-to-image diffusion models, text-to-3D generation has made significant progress in generating compelling geometry and appearance. However, existing methods still struggle to recover high-fidelity object materials, either only considering Lambertian reflectance, or failing to disentangle BRDF materials from the environment lights. In this work, we propose Material-Aware Text-to-3D via LAtent BRDF auto-EncodeR (**MATLABER**) that leverages a novel latent BRDF auto-encoder for material generation. We train this auto-encoder with large-scale real-world BRDF collections and ensure the smoothness of its latent space, which implicitly acts as a natural distribution of materials. During appearance modeling in text-to-3D generation, the latent BRDF embeddings, rather than BRDF parameters, are predicted via a material network. Through exhaustive experiments, our approach demonstrates the superiority over existing ones in generating realistic and coherent object materials. Moreover, high-quality materials naturally enable multiple downstream tasks such as relighting and material editing. Code and model will be publicly available at `https://sheldontsui.github.io/projects/Matlaber`.

1   Introduction

3D asset creation is imperative for various industrial applications such as gaming, film, and AR/VR. Traditional 3D asset creation pipeline involves multiple labor-intensive and time-consuming stages [1], all of which rely on specialized knowledge and professional aesthetic training. Thanks to the recent development of generative models, recent text-to-3D pipelines that automatically generate 3D assets from purely textual descriptions have received growing attention, due to their rapid advances in generation quality and efficiency, as well as their potential of significantly reducing the time and skill requirement of traditional 3D asset creation.

Gradually optimizing the target 3D asset represented as NeRF [2] or DMTET [3] through the SDS loss [4], compelling geometry and appearance can be obtained by these text-to-3D pipelines [5, 6, 4, 7–9]. However, as shown in Figure1, they still struggle to recover high-fidelity object materials, which significantly limits their real-world applications such as relighting. Although attempts have been made to model Lambertian reflectance [4, 8] and bidirectional reflectance distribution function (BRDF) [7], in their designs, the neural network responsible for predicting materials has no sufficient motivation and clues to unveil an appropriate material that obeys the natural distribution, especially under fixed light conditions, where their predicted material is often entangled with environment lights.

In this work, we resort to existing rich material data to learn a novel text-to-3D pipeline that effectively disentangles material from environment lights. In fact, despite the inaccessibility of paired datasets of material and text descriptions, there exist large-scale BRDF material datasets such as MERL BRDF [10], Adobe Substance3D materials [11], and the real-world BRDF collections TwoShotBRDF [12]. Therefore, we propose Material-Aware Text-to-3D via LAtent BRDF auto-
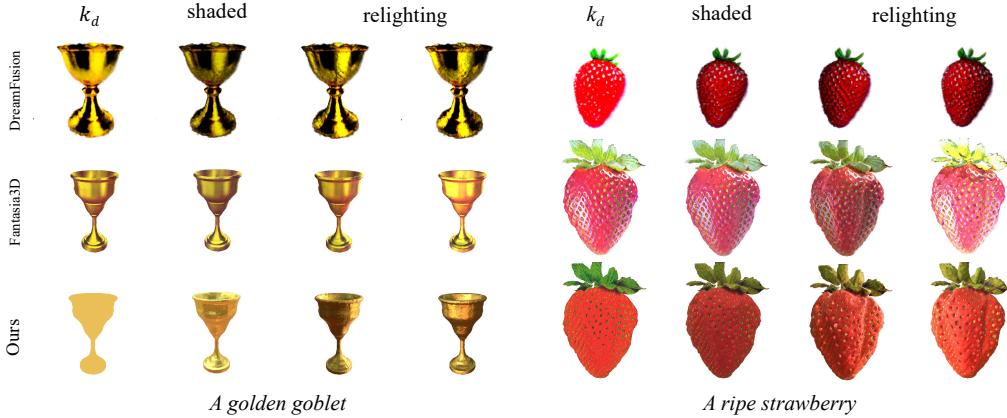
Figure 1: Text-to-3D generation aims to synthesize high-quality 3D assets aligning with given text descriptions. Despite the impressive appearance, representative methods like DreamFusion [4] and Fantasia3D [7] still fail to recover high-fidelity object materials. Specifically, DreamFusion only considers diffuse materials while Fantasia3D always predicts BRDF materials entangled with environment lights. Based on a latent BRDF auto-encoder, our approach is capable of generating natural materials for 3D assets, enabling realistic renderings under different illuminations.

EncodeR (**MATLABER**) that leverages a novel latent BRDF auto-encoder to synthesize natural and realistic materials that accurately align with given text prompts. The latent BRDF auto-encoder is trained to embed real-world BRDF priors of TwoShotBRDF in its smooth latent space, so that MATLABER can predict BRDF latent codes instead of BRDF values to focus more on choosing the most suitable material and worry less about the validity of predicted BRDF.

Thanks to the smooth latent space of the BRDF auto-encoder, our approach ensures the realism and coherence of object materials, achieving the ideal disentanglement of geometry and appearance. As shown in Figure 1, our approach can create 3D assets with high-fidelity material, outperforming previous state-of-the-art text-to-3D pipelines. More importantly, the effective estimation of object materials naturally allows various operations such as relighting, material editing, and scene manipulation, which can hardly be achieved before our work. It's noteworthy that these downstream tasks are extremely crucial for a series of real-world applications, paving the way for a more convenient paradigm of 3D content creation. Furthermore, by exploiting multi-modal datasets like ObjectFolder, our model has the potential to infer acoustic and tactile information from the obtained materials, which constitute the trinity of material for virtual objects.

2 Related Work

2.1 Text-to-Image Generation

In recent years, we have witnessed significant progress in text-to-image generation empowered by diffusion models. By training on large-scale text-image paired datasets, diffusion models can implicitly link semantic concepts and corresponding text prompts, and thus is capable of generating various and complex images of objects and scenes [13–16]. While GLIDE [13] obtains text embeddings via a pretrained CLIP [17] model, Imagen [14] and eDiff-I [18] adopt larger language models such as T5 [19] to achieve more diverse image synthesis and a deeper level of language understanding. To enable text-to-image training on limited computational resources, Stable Diffusion [16] leverages the latent diffusion model (LDM) and trains its diffusion model on the latent space instead of the pixel space, demonstrating highly competitive performance in terms of quality and flexibility. However, all of these works are constrained to the 2D domain and ignore the huge demand in the 3D field.

2.2 Text-to-3D Generation with 2D Supervision

With the great success of text-to-image synthesis, text-to-3D generation has gained significant attention in recent years as a promising way to generate high-quality 3D content from simple

textual input. Different from text-to-image generation, no massive text-3D paired data is available, making it infeasible to train a text-to-3D model as in training a text-to-image model. Alternatively, DreamField [5] and CLIP-mesh [6] explore the 2D supervision by using a pre-trained CLIP model to optimize the underlying 3D representations, such as NeRFs and meshes, to achieve high text-image alignment scores for multi-view renderings. Later, DreamFusion [4] utilizes, for the first time, a powerful 2D text-to-image diffusion model [14] as a prior and introduces an effective Score Distillation Sampling (SDS) loss to guide the optimization. Based on the promising SDS loss, Magic3D [8] optimizes 3D objects in two consecutive stages and further improves the rendered resolution of generated 3D objects from 64 to 512, showing impressive performance in text-to-3D generation. In Fantasia3D [7], Chen *et al* proposes to represent 3D objects with flexible DMTET representation [3] and model the appearance via the BRDF modeling, which can successfully generate compelling geometry and photorealistic object appearance. Recently, ProlificDreamer [9] proposes a more advanced guiding loss, Variational Score Distillation (VSD), for more diverse and high-fidelity object generation. Unlike these works, we aim to recover the inherent material information in text-to-3D generation, empowering more downstream applications such as relighting and material editing.

2.3 Material Estimation

In the community of computer vision and graphics, researchers aim to estimate surface materials for decades. The Bidirectional Reflection Distribution function (BRDF) is the most widely used material model, which characterizes how a surface reflects lighting from an incident direction toward an outgoing direction [20, 21]. One line of research works [22–25] targets recovering the reflectance from known 3D geometry and some methods [26–31] only focus on BRDF acquisition of 2D planar geometry. Moreover, the simultaneous acquisition of 3D geometry and materials has also gained a surge of interest and several solid papers [32–36] have already achieved compelling results on in-the-wild scenarios. Specifically, Neural-PIL [34] and NeRFactor [35] both leverage BRDF datasets for material priors. Unlike these methods, we aim to create high-quality surface materials aligning with the provided text prompts. Concurrent to this work, Fantasia3D [7] also explores the automatic creation of surface materials and object geometries from given language models. However, their obtained materials are unluckily entangled with lights and thus intractable to put into new environments.

3 Method

We present Material-Aware Text-to-3D via LAtent BRDF auto-EncodeR (**MATLABER**), aiming for photorealistic and relightable text-to-3D object generation. In the following, preliminaries on score distillation sampling are first presented in Section 3.1. Then in Section 3.2, we review appearance modeling in prior works and analyze their intrinsic deficiencies in the relighting scenarios. Finally, we introduce the latent BRDF auto-encoder in Section 3.3 and discuss how to incorporate it into material-aware text-to-3D generation in Section 3.4. An overview of our framework is illustrated in Figure 2.

3.1 Score Distillation Sampling (SDS)

DreamFusion [4] first achieves high-quality text-to-3D generation with a pretrained text-to-image diffusion model. It represents the scene with a modified Mip-NeRF [37], which can produce an image $x = g(\theta)$ at the desired camera pose. Here, g is a differentiable renderer, and θ is a coordinate-based MLP representing a 3D volume. The diffusion model ϕ comes with a learned denoising function $\epsilon_\phi(x_t; y, t)$ that predicts the sampled noise ϵ given the noisy image x_t, noise level t, and text embedding y. Specifically, to update the scene parameters θ, DreamFusion introduces Score Distillation Sampling (SDS), which computes the gradient as:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, x) = \mathbb{E}_{t,\epsilon}\left[w(t)(\epsilon_\phi(x_t; y, t) - \epsilon)\frac{\partial x}{\partial \theta}\right], \tag{1}$$

where $w(t)$ is a weighting function. Here, we follow Stable-dreamfusion [38] or Fantasia3D [7] and leverage the publicly available latent diffusion model (LDM) like Stable Diffusion [16] as our

guidance model. Therefore, the SDS loss now becomes:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, x) = \mathbb{E}_{t,\epsilon}\left[w(t)(\epsilon_\phi(z_t; y, t) - \epsilon)\frac{\partial z}{\partial x}\frac{\partial x}{\partial \theta}\right]. \qquad (2)$$

Intuitively, SDS loss will push all the rendered images to the high probability density regions given by the pretrained diffusion model [8]. Apart from the rendered RGB images, Fantasia3D [7] regards the surface normal maps n as special images and shows applying SDS loss on normal maps encourages to generate good object geometry with fine details.

3.2 Appearance Modeling in Text-to-3D Generation

To employ the SDS loss, prior works adopt different schemes to model the appearance of 3D objects. DreamFusion [4] leverages a reflectance model similar to [25, 39–41] and only considers diffuse reflectance [42, 43] while rendering multi-view images. For each point $\mathbf{x}$ on a specific ray, the RGB albedo $\boldsymbol{\rho}$ and volumetric density τ are predicted via a MLP, and the surface normal $\mathbf{n}$ is obtained with $\mathbf{n} = -\nabla_{\mathbf{x}}\tau/\|\nabla_{\mathbf{x}}\tau\|$. After accumulating each normal $\mathbf{n}$ and albedo $\boldsymbol{\rho}$ in a ray with volume rendering [44], assuming some point light source with 3D coordinate $\boldsymbol{\ell}$ and color $\boldsymbol{\ell}_\rho$, and an ambient light color $\boldsymbol{\ell}_a$, DreamFusion employs Lambertian shading to produce a color $\mathbf{c}$ for each pixel as:

$$\mathbf{c} = \boldsymbol{\rho} \circ (\boldsymbol{\ell}_\rho \circ \max(0, \boldsymbol{n} \cdot (\boldsymbol{\ell} - \mathbf{x})/\|\boldsymbol{\ell} - \mathbf{x}\|) + \boldsymbol{\ell}_a). \qquad (3)$$

Although generating various 3D objects with appealing appearances, DreamFusion fails to model specular reflectance, which is an indispensable term in the material that leads to photorealistic renderings under different illuminations. Magic3D [8] uses more advanced DMTET [3] as the scene representation and targets at higher-fidelity 3D models but still adopts Lambertian shading when texturing the object meshes extracted from DMTET. Hence, their synthesized 3D objects all lack complete material information, significantly limiting their applications in real-world scenarios.

To achieve more photorealistic and relightable rendering, Fantasia3D [7] introduces the spatially varying Bidirectional Reflectance Distribution Function (BRDF) into text-to-3D generation. They represent the material with three components [45], namely the diffuse term $\mathbf{k}_d \in \mathbb{R}^3$, the roughness and metallic term $\mathbf{k}_{rm} \in \mathbb{R}^2$ containing roughness k_r and metalness factor m, as well as the normal variation term $\mathbf{k}_n \in \mathbb{R}^3$. According to the convention in [46], the specular term $\mathbf{k}_s$ is computed with $\mathbf{k}_s = (1 - m) \cdot 0.04 + m \cdot \mathbf{k}_d$. For a specific surface point $\mathbf{x}$ with normal $\mathbf{n}$ and outgoing view direction $\boldsymbol{\omega}_o$, the final rendering $L(\mathbf{x}, \boldsymbol{\omega}_o)$ can be obtained following the rendering equation [47]:

$$L(\mathbf{x}, \boldsymbol{\omega}_o) = \int_\Omega L_i(\mathbf{x}, \boldsymbol{\omega}_i) f(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathbf{k}_d, \mathbf{k}_s)(\boldsymbol{\omega}_i \cdot \mathbf{n}) \mathrm{d}\boldsymbol{\omega}_i, \qquad (4)$$

where incident light L_i comes from the direction $\boldsymbol{\omega}_i$ and BRDF f is related to the diffuse term $\mathbf{k}_d$ and the specular term $\mathbf{k}_s$. However, in Fantasia3D [7], the metalness factor m is usually set to 0 and thus the specular term $\mathbf{k}_s$ is actually ignored during the appearance modeling. Moreover, they utilize a fixed HDR environment map with uniform brightness distribution as the environment lights all the time. Despite its appealing appearance under fixed illuminations, Fantasia3D always predicts materials entangled with environmental lights, which leads to unrealistic renderings under novel lighting conditions.

3.3 Latent BRDF Auto-Encoder

Confronted with the limitations of prior works, we take both diffuse and specular terms into consideration for appearance modeling. Specifically, for a surface point $\mathbf{x}$, we aim to estimate the Cook-Torrence [48] BRDF parameter $\mathbf{k} \in \mathbb{R}^7$ including diffuse $\mathbf{k}_d \in \mathbb{R}^3$, specular $\mathbf{k}_s \in \mathbb{R}^3$, and roughness $k_r \in \mathbb{R}$. However, directly optimizing on standard BRDF space lacks necessary constraints and may make the predicted materials fall into invalid BRDF regions. As mentioned in Section 1, we resort to data-driven BRDF priors that are learned from real-world BRDF data collections.

Inspired by previous literature on reflectance decomposition [35, 34], we leverage TwoShotBRDF [12] dataset to train a latent BRDF auto-encoder for such a material prior. TwoShotBRDF is a real-world BRDF material dataset containing $11,250$ high-quality SVBRDF maps of size 768×768 collected from various online sources, where each pixel represents an independent BRDF parameter. Following [49, 34], interpolating auto-encoders are adopted here for smoother interpolation in the
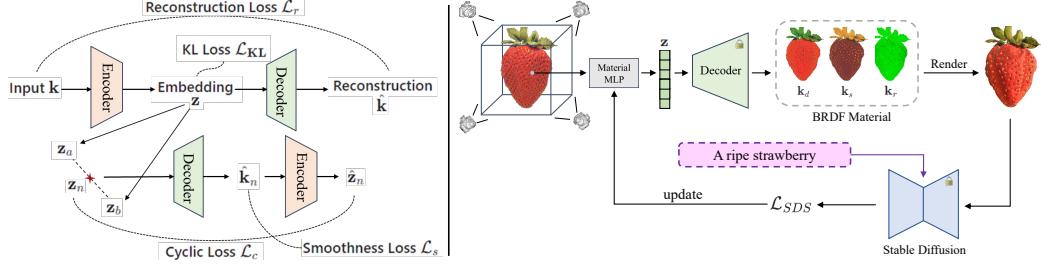
Figure 2: **Left**: Our latent BRDF auto-encoder is trained on the TwoShotBRDF dataset with four losses, *i.e.*, reconstruction loss, KL divergence loss, smoothness loss, and cyclic loss. Imposing KL divergence and smoothness loss on latent embeddings encourages a smooth latent space [34]. **Right**: Instead of predicting BRDF materials directly, we leverage a material MLP Γ to generate latent BRDF code $\mathbf{z}$, which is then decoded to 7-dim BRDF parameters via our pretrained decoder. Similar to prior works, the SDS loss can be applied to the rendered images, which empowers the training of our material MLP network. (Note that, roughness k_r is scalar and we visualize it with the green channel in this paper.)

latent space. As illustrated in Figure 2, our BRDF auto-encoder consists of an MLP encoder $\mathcal{E}$ and an MLP decoder $\mathcal{D}$. Given a BRDF parameter $\mathbf{k}$, the encoder will generate a 4-dim latent code $\mathbf{z} = \mathcal{E}(\mathbf{k})$, which is then reconstructed to a 7-dim BRDF $\hat{\mathbf{k}} = \mathcal{D}(\mathbf{z})$ via the decoder network. Here, apart from the standard L_2 reconstruction loss $\mathcal{L}_r$, we additionally employ a Kullback–Leibler (KL) divergence loss $\mathcal{L}_{\mathrm{KL}} = KL(p(\mathbf{z})||\mathcal{N}(\mathbf{0},\mathbf{I}))$ to encourage the smoothness of latent space. Moreover, for an ideal smooth latent space, the linearly interpolated embeddings $\mathbf{z}_n = \alpha \cdot \mathbf{z}_a + (1 - \alpha) \cdot \mathbf{z}_b$ between two random latent codes $\mathbf{z}_a$ and $\mathbf{z}_b$ could be also decoded to a reasonable BRDF parameter $\hat{\mathbf{k}}_n = \mathcal{D}(\mathbf{z}_n)$. We, therefore, impose a smoothness loss $\mathcal{L}_s = \sum_n (\nabla_\alpha \mathcal{D}(\mathbf{z}_n))^2$ for a batch of uniformly interpolated latent codes $\mathbf{z}_n$ following [34]. After adding the L_2 cyclic loss $\mathcal{L}_c$ between interpolated code $\mathbf{z}_n$ and the re-encoded counterpart $\hat{\mathbf{z}}_n = \mathcal{E}(\hat{\mathbf{k}}_n)$, the total loss for our latent BRDF auto-encoder training is a combination of four losses:

$$\mathcal{L} = \mathcal{L}_r + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}} + \lambda_s\mathcal{L}_s + \lambda_c\mathcal{L}_c, \tag{5}$$

where λ_{KL}, λ_s and λ_c are all balancing coefficients. With this total loss function, our encoder and decoder networks are trained jointly.

3.4 Material-Aware Text-to-3D

In this work, we divide the whole text-to-3D generation into two consecutive stages: geometry generation and appearance generation. Following Fantasia3D [7], we represent the 3D objects with the hybrid scene representation of DMTET owing to its superior performance in geometry modeling and photorealistic surface rendering. While generating the geometry, we employ the SDS loss on normal maps similar to strategies proposed in [7]. For the DMTET predicted by a geometry MLP ψ, a differentiable render g can render a normal map n from a randomly sampled camera pose c as follows: $n = g(\psi, c)$. Then, the SDS loss on normal maps n will help the geometry MLP ψ update until it converges to a satisfactory geometry aligning with the given text prompt.

Once obtaining the geometry of 3D objects, we can leverage our latent BRDF auto-encoder for appearance generation. As shown in Figure 2, for any point $\mathbf{x}$ on the surface, we first apply the hash-grid positional encoding $\beta(\cdot)$ [50] and then use a material MLP Γ parameterized as γ to predict its BRDF latent code $\mathbf{z_x}$, which is then transferred to 7-dim BRDF parameter $\mathbf{k_x}$ via the pre-trained BRDF decoder $\mathcal{D}$ following:

$$\mathbf{z_x} = \Gamma(\beta(\mathbf{x}); \gamma), \quad \mathbf{k_x} = \mathcal{D}(\mathbf{z_x}). \tag{6}$$

Equipped with this crucial material information $\mathbf{k_z} = [\mathbf{k}_d, \mathbf{k}_s, k_r]$, the point $\mathbf{x}$ can be rendered with the aforementioned rendering equation 4 under given incoming light $L_i(\boldsymbol{\omega}_i)$ from direction $\boldsymbol{\omega}_i$. Specifically, for the point $\mathbf{x}$ with normal $\mathbf{n}$ and outgoing view direction $\boldsymbol{\omega}_o$, the final rendering $L(\mathbf{x}, \boldsymbol{\omega}_o)$ is the summation of diffuse intensity L_d and specular intensity L_s as $L(\mathbf{x}, \boldsymbol{\omega}_o) = L_d(\mathbf{x}) +$

$L_s(\mathbf{x}, \boldsymbol{\omega}_o)$, and the two terms can be computed as follows:

$$L_d(\mathbf{x}) = \mathbf{k}_d(1-m) \int_{\Omega} L_i(\mathbf{x}, \boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \mathbf{n}) \mathrm{d}\boldsymbol{\omega}_i,$$

$$L_s(\mathbf{x}, \boldsymbol{\omega}_o) = \int_{\Omega} \frac{DFG}{4(\boldsymbol{\omega}_o \cdot \mathbf{n})(\boldsymbol{\omega}_i \cdot \mathbf{n})} L_i(\mathbf{x}, \boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \mathbf{n}) \mathrm{d}\boldsymbol{\omega}_i, \tag{7}$$

where D, G, and F are functions representing the GGX [51] normal distribution (NDF), geometric attenuation, and Fresnel term, respectively. It's noteworthy that the integration over the hemisphere Ω can be calculated using the split-sum method following [46, 36]. In particular, the specular intensity $L_s(\mathbf{x}, \boldsymbol{\omega}_o)$ is further approximated as:

$$F(\boldsymbol{\omega}_o, \mathbf{h}, k_r) = F_0 + \big(\max(1 - k_r, F_0) - F_0 \big) \big(1 - (\boldsymbol{\omega}_o \cdot \mathbf{h}) \big)^5, \tag{8}$$

$$L_s(\mathbf{x}, \boldsymbol{\omega}_o) = \big(F(\boldsymbol{\omega}_o, \mathbf{h}, k_r) B_0(\boldsymbol{\omega}_o \cdot \mathbf{n}, k_r) + B_1(\boldsymbol{\omega}_o \cdot \mathbf{n}, k_r) \big) \int_{\Omega} D(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \mathbf{n}, k_r) L_i(\mathbf{x}, \boldsymbol{\omega}_i) \mathrm{d}\boldsymbol{\omega}_i, \tag{9}$$

where $\mathbf{h} = (\boldsymbol{\omega}_i + \boldsymbol{\omega}_o)/|\boldsymbol{\omega}_i + \boldsymbol{\omega}_o|$ represents the half vector, F_0 is the specular albedo $\mathbf{k}_s$ [1], and B_0 and B_1 are two pre-computed 2D lookup textures (LUT) indexed by $(\boldsymbol{\omega}_o \cdot \mathbf{n})$ and the roughness k_r.

During the appearance modeling, we will leverage multiple environment maps and keep vertically rotating these maps to encourage the predicted BRDF materials to disentangle from environment lights. Given a sampled camera viewpoint, the rendered image x is actually the aggregation of rendered pixels along the outgoing view direction $\boldsymbol{\omega}_o$, which can be obtained with Equation 7. Accordingly, we can apply the aforementioned SDS loss in Equation 2 to the rendered image x and the material MLP Γ can be optimized with the gradient *w.r.t.* its parameter γ:

$$\nabla_{\gamma} \mathcal{L}_{\text{SDS}}(\phi, x) = \mathbb{E}_{t, \epsilon} \left[w(t)(\epsilon_{\phi}(z_t; y, t) - \epsilon) \frac{\partial z}{\partial x} \frac{\partial x}{\partial \mathbf{k}} \frac{\partial \mathbf{k}}{\partial \gamma} \right]. \tag{10}$$

Apart from the SDS loss, a material smoothness regularizer is extra utilized for enforcing smooth diffuse materials following [36, 35]. For a surface point $\mathbf{x}$ with diffuse $\mathbf{k}_d(\mathbf{x})$, the regularizer is defined as:

$$\mathcal{L}_{\text{mat}} = \sum_{\mathbf{x} \in \mathcal{S}} |\mathbf{k}_d(\mathbf{x}) - \mathbf{k}_d(\mathbf{x} + \boldsymbol{\epsilon})|, \tag{11}$$

where $\mathcal{S}$ denotes the object surface and $\boldsymbol{\epsilon}$ is a small random 3D perturbation.

4 Experiments

Implementation details. While training our latent BRDF auto-encoder, we will scatter all the SVBRDF maps to 7-dim BRDF parameters and randomly shuffle them for training. We set the batch size to 256, and loss balancing coefficients λ_{KL}, λ_s and λ_c to 2e-4, 0.05, and 1e-4, respectively. Our BRDF auto-encoder is trained for 30 epochs using the AdamW optimizer with a learning rate of 1e-4 on an NVIDIA A100 GPU. For the geometry generation, we initialize the DMTET with either a 3D ellipsoid, a 3D cylinder, or a customized 3D model provided by users following [7]. During the text-to-3D generation, the 3D object is optimized on 4 NVIDIA A100 GPUs with an AdamW optimizer, where each GPU loads 9 images rendered from randomly sampled camera poses. Specifically, our method spends $3,000$ iterations (learning rate 0.001) on geometry modeling and $2,000$ iterations (learning rate 0.01) on material generation. For the weight in score distillation sampling, we always adopt the *strategy I* proposed in Fantasia3D [7].

Qualitative results. Given text descriptions, our method MATLABER is capable of generating photorealistic 3D assets and simultaneously obtaining high-fidelity BRDF materials. In this section, we compare our approach with three representative methods, namely DreamFusion [4], Magic3D [8], and Fantasia3D [7]. As shown in Figure 4, we present the comparative results given the same text descriptions. For Fantasia3D, 3D assets are synthesized with their official code, while results of DreamFusion and Magic3D are borrowed from their project pages owing to the inaccessibility of

---

[1]The definition of specular albedo $\mathbf{k}_s$ is different from that mentioned in Section 3.2, which is actually equal to $F(\boldsymbol{\omega}_o, \mathbf{h}, k_r)$ computed here. For convenience, we still use the notation $\mathbf{k}_s$ for specular albedo.

A sliced loaf of fresh bread

A DSLR photo of a hamburger

A pineapple

A delicious croissant

A typewriter

A blue tulip

A plush dragon toy

A rabbit, animated movie character, high detail 3d model
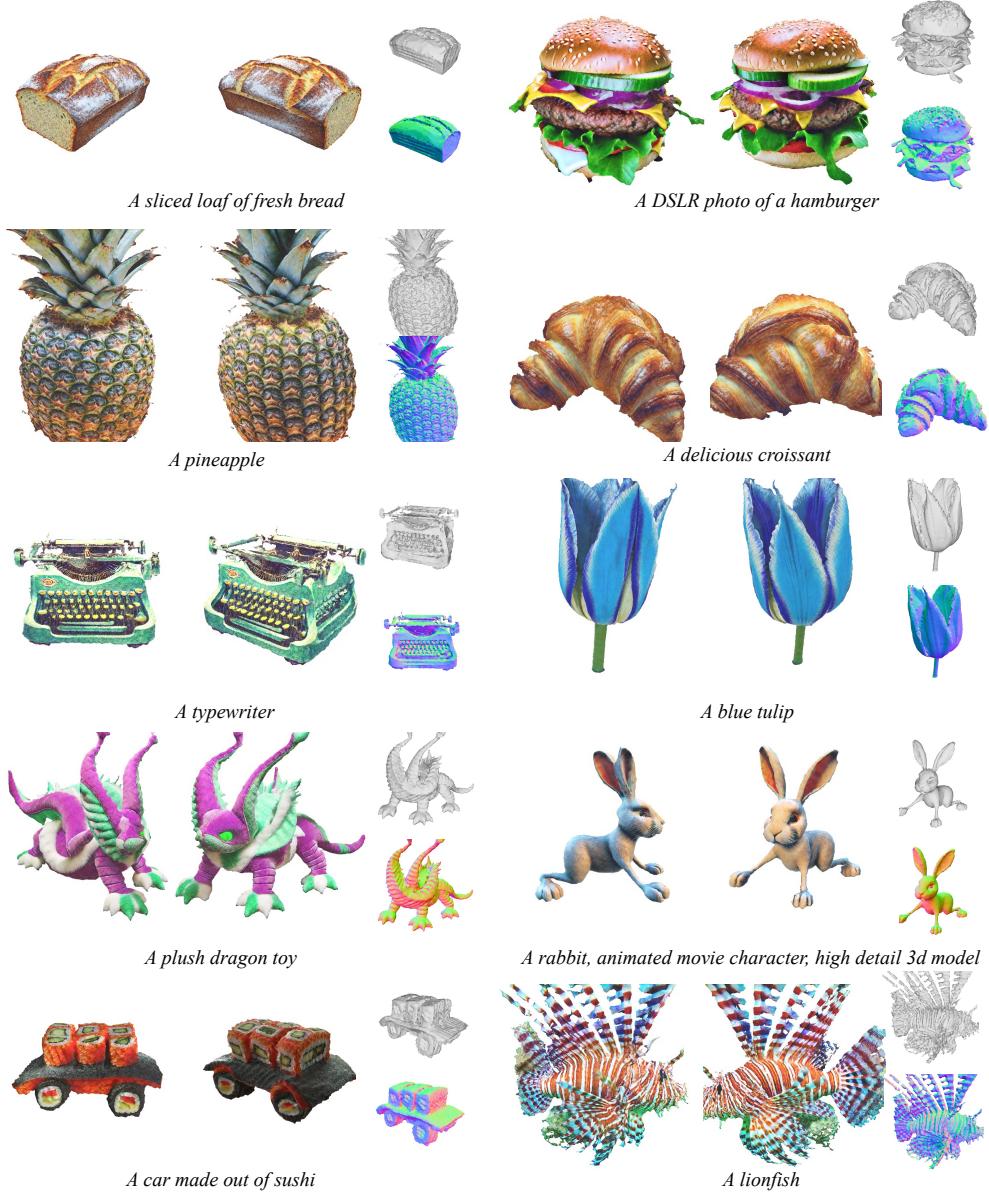
A car made out of sushi

A lionfish

Figure 3: The gallery of our text-to-3D generation results. Shapes, normal maps, and shaded images from two random viewpoints are presented here.

their model weights. Thanks to the latent BRDF auto-encoder, our method generates competitive geometry as Fantasia3D and demonstrates a better appearance with more natural textures and richer details. Moreover, the gallery of more 3D assets generated with MATLABER is provided in Figure 3.

User study. To evaluate the quality of generated 3D objects from human's perspective, we invited 80 volunteers to conduct a user study. For each participant, he/she will view 10 randomly selected 3D objects generated by our approach and three baseline methods. They are asked to evaluate these 3D assets in four different dimensions: 'alignment', 'realism', 'details', and 'disentanglement'. Alignment means how closely the generated 3D objects match the given text prompts, and realism reflects the fidelity of synthesized mesh textures. Besides, users also need to judge the richness of details on these 3D objects. Finally, we extra present the corresponding diffuse materials and ask them whether the diffuse is disentangled from the environment lights.

Totally, we collect 800 responses, and the comparative results are summarized in Table 1. Our method MATLABER achieves the best performance on three evaluation protocols, *i.e.*, realism, details, and

Table 1: Mean opinion scores in range $1 \sim 5$, where 1 means the lowest score and 5 is the highest score.

Method	Alignment	Realism	Details	Disentanglement
DreamFusion [4]	3.97 (± 0.66)	3.56 (± 0.43)	3.23 (± 0.61)	3.48 (± 0.59)
Magic3D [8]	**4.01** (± 0.59)	3.84 (± 0.72)	3.70 (± 0.66)	3.14 (± 0.89)
Fantasia3D [7]	3.76 (± 0.82)	4.17 (± 0.65)	4.27 (± 0.75)	2.93 (± 0.95)
Ours	3.81 (± 0.75)	**4.35** (± 0.60)	**4.31** (± 0.70)	**3.89** (± 0.65)

An ice cream sundae

A 3D model of an adorable cottage with a thatched roof

A plate piled high with chocolate chip cookies

Figure 4: Qualitative comparisons to baselines. Our results have more natural textures and richer details.

disentanglement. Especially for disentanglement, ours outperforms three baseline methods by a large margin, showing the effectiveness of our proposed latent BRDF auto-encoder model. In terms of alignment, DreamFusion [4] and Magic3D [8] both capitalize on large-scale language models such as T5 [19] and therefore demonstrate better alignment with the text descriptions. Unfortunately, these models are all not released for public usage.

Relighting and material editing. Since our model is capable of generating BRDF materials, we can implement relighting on these 3D assets. To validate the fidelity of our generated materials, we manually rotate an HDR environment map and thus obtain a series of HDR maps with different environment lights. As shown in Figure 5, the relit objects under different illuminations are all natural and photorealistic. Moreover, thanks to the smooth latent space of our BRDF auto-encoder, we can also conduct material interpolation if given two different material descriptions. Figure 6 demonstrates a smooth material morphing on the goblet and tulip. For example, given the text prompts of "A golden goblet" and "A silver goblet", we can obtain their corresponding material latent embeddings after the optimization, and the final results can be obtained via a naive interpolation on the latent embeddings.

5   Conclusion

In this work, we propose MATLABER, a novel latent BRDF auto-encoder for material-aware text-to-3D generation. This auto-encoder is trained with large-scale real-world BRDF collections, providing implicit material prior for the appearance modeling in text-to-3D generation. Thanks to such a material prior, our approach can generate high-quality and coherent object materials in text-to-3D synthesis, achieving the ideal disentanglement of geometry and appearance. Moreover, the generated BRDF materials also support various operations such as relighting, material editing, and scene
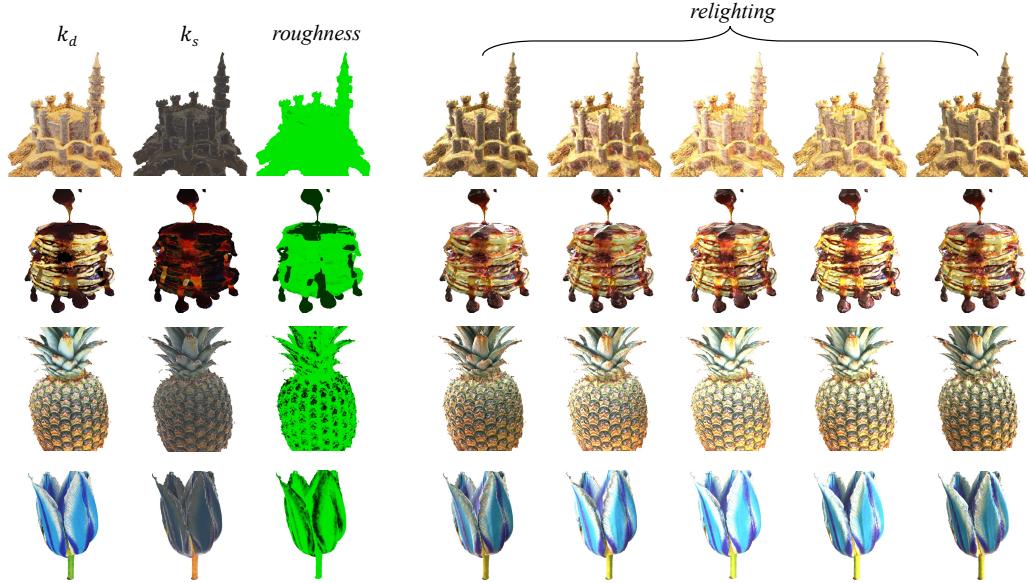
Figure 5: Relighting results. On the left side, we list the generated BRDF materials, including diffuse, specular, and roughness. The relit images under a rotating environment light are presented on the right side.
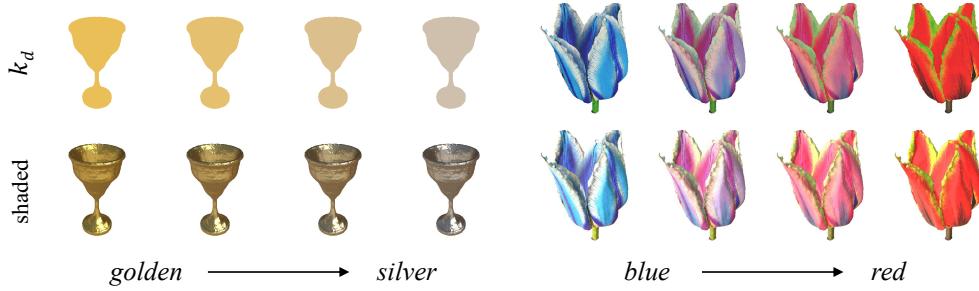

Figure 6: Material interpolation results. Thanks to the smooth latent space of our BRDF auto-encoder, we can conduct a linear interpolation on the BRDF embeddings. As can be observed here, a golden goblet will turn into a silver goblet while the color of the tulip changes from blue to red gradually.
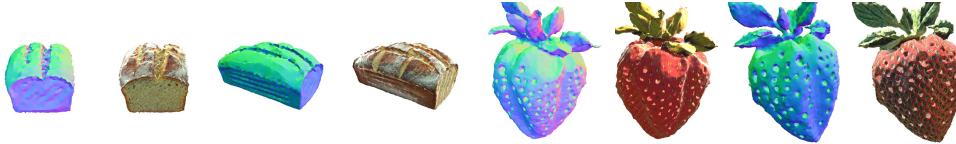

Figure 7: Failure cases. Owing to imperfect geometry, our generated 3D objects will present clear artifacts under some novel illuminations.

manipulations. Meanwhile, our latent BRDF auto-encoder potentially can be used in other tasks to predict materials instead of RGB values to enable the ability of relighting and material editing.

Limitations. Our generated 3D objects will present artifacts under some environment lights owing to imperfect geometry. As shown in Figure 7, there exist some weird textures on the top or the side of the bread, and the relit strawberry presents many undesirable hollows. The problem of imperfect geometry is not our focus here and is expected to be solved in future research. Besides, 3D assets generated by our current method still lack diversity and we are interested in circumventing it with promising Variational Score Distillation (VSD) loss proposed in [9].

9

References

[1] M. Labschütz, K. Krösl, M. Aquino, F. Grashäftl, and S. Kohl, "Content creation for a 3D game with maya and unity 3D," *Institute of Computer Graphics and Algorithms, Vienna University of Technology*, vol. 6, p. 124, 2011.

[2] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[3] T. Shen, J. Gao, K. Yin, M.-Y. Liu, and S. Fidler, "Deep marching tetrahedra: a hybrid representation for high-resolution 3D shape synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[4] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *International Conference on Learning Representations (ICLR)*, 2023.

[5] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 867–876, 2022.

[6] N. Mohammad Khalid, T. Xie, E. Belilovsky, and T. Popa, "Clip-mesh: Generating textured meshes from text using pretrained image-text models," in *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–8, 2022.

[7] R. Chen, Y. Chen, N. Jiao, and K. Jia, "Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation," *arXiv preprint arXiv:2303.13873*, 2023.

[8] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3D: High-resolution text-to-3D content creation," *arXiv preprint arXiv:2211.10440*, 2022.

[9] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation," *arXiv preprint arXiv:2305.16213*, 2023.

[10] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model," *ACM Transactions on Graphics*, vol. 22, pp. 759–769, July 2003.

[11] Adobe, "substance3d." https://substance3d.adobe.com/assets.

[12] M. Boss, V. Jampani, K. Kim, H. Lensch, and J. Kautz, "Two-shot spatially-varying brdf and shape estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3982–3991, 2020.

[13] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *Proceedings of Machine Learning Research (PMLR)*, 2021.

[14] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

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

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

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

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[20] F. E. Nicodemus, "Directional reflectance and emissivity of an opaque surface," *Applied optics*, vol. 4, no. 7, pp. 767–775, 1965.

[21] Z. Li, K. Sunkavalli, and M. Chandraker, "Materials for masses: Svbrdf acquisition with a single mobile phone image," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 72–87, 2018.

[22] H. P. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-based reconstruction of spatially varying materials," in *Rendering Techniques 2001: Proceedings of the Eurographics Workshop in London, United Kingdom, June 25–27, 2001 12*, pp. 103–114, Springer, 2001.

[23] H. P. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-based reconstruction of spatial appearance and geometric detail," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 2, pp. 234–257, 2003.

[24] D. Gao, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong, "Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images.," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 134–1, 2019.

[25] S. Bi, Z. Xu, P. Srinivasan, B. Mildenhall, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi, "Neural reflectance fields for appearance acquisition," *arXiv preprint arXiv:2008.03824*, 2020.

[26] M. Aittala, T. Weyrich, and J. Lehtinen, "Practical svbrdf capture in the frequency domain.," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 110–1, 2013.

[27] M. Aittala, T. Weyrich, J. Lehtinen, *et al.*, "Two-shot svbrdf capture for stationary materials.," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 110–1, 2015.

[28] N. Thanikachalam, L. Baboulaz, D. Firmenich, S. Süsstrunk, and M. Vetterli, "Handheld reflectance acquisition of paintings," *IEEE Transactions on Computational Imaging*, vol. 3, no. 4, pp. 580–591, 2017.

[29] Z. Hui, K. Sunkavalli, J.-Y. Lee, S. Hadap, J. Wang, and A. C. Sankaranarayanan, "Reflectance capture using univariate sampling of brdfs," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5362–5370, 2017.

[30] K. Kang, Z. Chen, J. Wang, K. Zhou, and H. Wu, "Efficient reflectance capture using an autoencoder," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–10, 2018.

[31] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau, "Single-image svbrdf capture with a rendering-aware deep network," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–15, 2018.

[32] G. Nam, J. H. Lee, D. Gutierrez, and M. H. Kim, "Practical svbrdf acquisition of 3d objects with unstructured flash photography," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–12, 2018.

[33] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 8, pp. 1670–1687, 2014.

[34] M. Boss, V. Jampani, R. Braun, C. Liu, J. Barron, and H. Lensch, "Neural-pil: Neural pre-integrated lighting for reflectance decomposition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10691–10704, 2021.

[35] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Transactions on Graphics (ToG)*, vol. 40, no. 6, pp. 1–18, 2021.

[36] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, and S. Fidler, "Extracting triangular 3D models, materials, and lighting from images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8280–8290, 2022.

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

[38] J. Tang, "Stable dreamfusion." https://github.com/ashawkey/stable-dreamfusion.

[39] X. Pan, X. Xu, C. C. Loy, C. Theobalt, and B. Dai, "A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20002–20013, 2021.

[40] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "Nerd: Neural reflectance decomposition from image collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12684–12694, 2021.

[41] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7495–7504, 2021.

[42] J. H. Lambert, "Photometria sive de mensura et gradibus luminis," *Colorum et Umbrae, Eberhard Klett, Augsberg, Germany*, 1760.

[43] R. Ramamoorthi and P. Hanrahan, "A signal-processing framework for inverse rendering," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 117–128, 2001.

[44] N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.

[45] S. McAuley, S. Hill, N. Hoffman, Y. Gotanda, B. Smits, B. Burley, and A. Martinez, "Practical physically-based shading in film and game production," in *ACM SIGGRAPH 2012 Courses*, pp. 1–7, 2012.

[46] B. Karis and E. Games, "Real shading in unreal engine 4," *Proc. Physically Based Shading Theory Practice*, vol. 4, no. 3, p. 1, 2013.

[47] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 143–150, 1986.

[48] R. L. Cook and K. E. Torrance, "A reflectance model for computer graphics," *ACM Transactions on Graphics (ToG)*, vol. 1, no. 1, pp. 7–24, 1982.

[49] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, "Understanding and improving interpolation in autoencoders via an adversarial regularizer," *arXiv preprint arXiv:1807.07543*, 2018.

[50] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[51] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet models for refraction through rough surfaces," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pp. 195–206, 2007.