DreamMat: High-quality PBR Material Generation with Geometry- and Light-aware Diffusion Models

YUQING ZHANG*, State Key Lab of CAD&CG, Zhejiang University, China
YUAN LIU*, Tencent Games, China
ZHIYU XIE, State Key Lab of CAD&CG, Zhejiang University, China
LEI YANG, Tencent Games, China
ZHONGYUAN LIU, Tencent Games, China
MENGZHOU YANG, Tencent Games, China
RUNZE ZHANG, Tencent Games, China
QILONG KOU, Tencent Games, China
CHENG LIN, Tencent Games, China
WENPING WANG, Texas A&M University, U.S.A
XIAOGANG JIN†, State Key Lab of CAD&CG, Zhejiang University, China

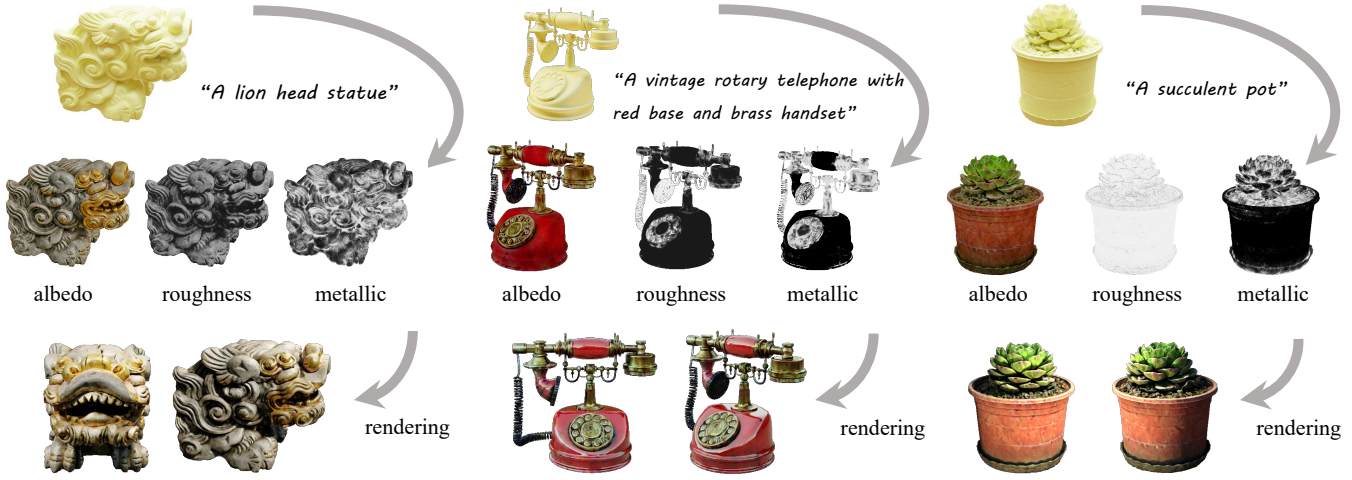

Fig. 1. Using untextured meshes and textual descriptions as input (top row), our method generates high-quality appearances consisting of albedo, roughness, and metallic (middle row) that can be applied in modern graphics engines for photo-realistic rendering under any new illumination environments (bottom row).

Recent advancements in 2D diffusion models allow appearance generation on untextured raw meshes. These methods create RGB textures by distilling a 2D diffusion model, which often contains unwanted baked-in shading effects and results in unrealistic rendering effects in the downstream applications. Generating Physically Based Rendering (PBR) materials instead of just RGB textures would be a promising solution. However, directly distilling the PBR material parameters from 2D diffusion models still suffers from incorrect material decomposition, such as baked-in shading effects in albedo. We introduce *DreamMat*, an innovative approach to resolve the aforementioned problem, to generate high-quality PBR materials from text descriptions. We find out that the main reason for the incorrect material distillation is that large-scale 2D diffusion models are only trained to generate final shading colors, resulting in insufficient constraints on material decomposition during distillation. To tackle this problem, we first finetune a new light-aware 2D diffusion model to condition on a given lighting environment and generate the shading results on this specific lighting condition. Then, by applying the same environment lights in the material distillation, DreamMat can generate

*Equal contribution
†Corresponding author.

Authors' addresses: Yuqing Zhang, State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang, China; Yuan Liu, Tencent Games, Shenzhen, China; Zhiyu Xie, State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang, China; Lei Yang, Tencent Games, Shenzhen, China; Zhongyuan Liu, Tencent Games, Shenzhen, China; Mengzhou Yang, Tencent Games, Shenzhen, China; Runze Zhang, Tencent Games, Shenzhen, China; Qilong Kou, Tencent Games, Shenzhen, China; Cheng Lin, Tencent Games, Shenzhen, China; Wenping Wang, Texas A&M University, Texas, U.S.A; Xiaogang Jin, State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang, China.

ACM 0730-0301/2024/7-ART39
https://doi.org/10.1145/3658170

high-quality PBR materials that are not only consistent with the given geometry but also free from any baked-in shading effects in albedo. Extensive experiments demonstrate that the materials produced through our methods exhibit greater visual appeal to users and achieve significantly superior rendering quality compared to baseline methods, which are preferable for downstream tasks such as game and film production.

Additional Key Words and Phrases: 3D generation, text-guided texturing, inverse rendering

ACM Reference Format:
Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, and Xiaogang Jin. 2024. DreamMat: High-quality PBR Material Generation with Geometry- and Light-aware Diffusion Models . *ACM Trans. Graph.* 43, 4, Article 39 (July 2024), 18 pages. https://doi.org/10.1145/3658170

1 INTRODUCTION

Creating high-quality appearances for objects is a critical task in computer graphics because they can significantly improve the realism of rendering in a variety of applications such as movies, games, and AR/VR. However, even experienced artists may find it time-consuming to create object appearances [Labschütz et al. 2011] due to the need for expertise with complex commercial 3D software such as Mari, ZBrush, and Substance Painter. This necessitates the development of new tools for efficiently creating object appearances, such that even novice users can do so with simple text prompts.

Several text-driven methods [Cao et al. 2023; Chen et al. 2023c; Knodt and Gao 2023; Le et al. 2023; Richardson et al. 2023; Yu et al. 2023a; Zeng et al. 2023] have been developed to generate RGB textures on untextured meshes. These approaches use powerful 2D text-to-image diffusion models [Ho et al. 2020; Rombach et al. 2022], such as the Stable Diffusion model [Rombach et al. 2022], to achieve impressive results. While these techniques make it easier to create object appearances, they frequently produce undesirable shading effects such as highlights and shadows, as shown in Fig. 2 (a). The baked-in shading effects in generated appearances cause unrealistic results in rendering, limiting their applicability in downstream tasks such as game or film production.

Using PBR materials instead of RGB textures can improve object appearance, but it may present some challenges. Directly training a material generation network is difficult and costly due to a scarcity of high-quality 3D assets with known PBR materials. Alternatively, recent research [Chen et al. 2023a; Xu et al. 2023] integrates material decomposition in distilling a powerful 2D text-to-image diffusion model, which does not require training on datasets with known PBR materials. Though these methods achieve impressive results in some examples, correct material generation remains a challenge, as illustrated in Fig. 2 (b). This is because 2D diffusion generative models can only generate final shading results, making it difficult to decompose correct material parameters due to the ill-posed nature of the material decomposition task.

In this paper, we begin with an in-depth analysis of the ill-posed problem of material decomposition in the context of diffusion distillation framework. Diffusion models are trained to generate natural RGB images which are the final shading results of some unknown environmental lights and materials. However, on different distillation steps, the generated images may correspond to different environmental lights, making it impossible to accurately estimate a fixing environmental light and thus leading to incorrect materials. Fantasia3D [Chen et al. 2023a] uses a single predefined environmental light to distill diffusion models to generate materials but the generated images from diffusion models may not be consistent with the given environment light, as shown in Fig. 3 (b), still resulting in incorrect materials.

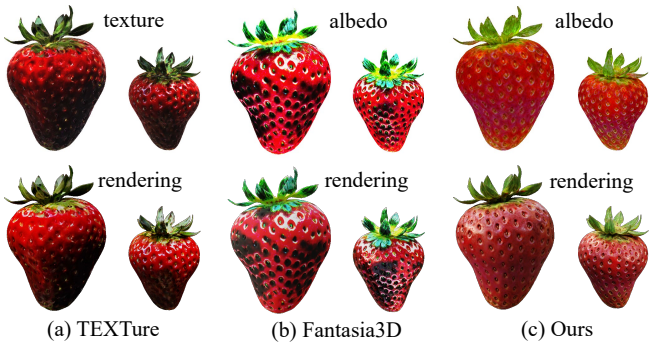

Fig. 2. **Generated albedo and rendering results in the same environment light.** (a) TEXTure [Yu et al. 2023a] generates an RGB texture map containing shading effects, leading to incorrect renderings in a new environment. (b) Fantasia3D [Chen et al. 2023a] directly distills a diffusion model to generate materials, which still contain unwanted shading effects in albedo. (c) Our method can generate correct materials, allowing for more photorealistic renderings in a new environment.

Fig. 3. **An untextured stool mesh and its generated images using different methods.** (a) An untextured mesh with a given light environment. (b) A generated image of a depth-to-image Stable Diffusion model, which is inconsistent with the given environment light and results in incorrect materials decomposition. (c) An image generated by our geometry- and light-aware diffusion model, which is consistent with the environment light.

Based on our analysis, we present *DreamMat*, a novel method to create high-quality appearances on an untextured mesh by generating PBR materials with a diffusion model. The key idea of DreamMat consists of two aspects. First, in the distillation process, we randomly select from a set of predefined HDR images as the environment light so that DreamMat can focus on generating the object materials. Second, we propose a novel geometry- and light-aware diffusion model, which is trained to generate images that are consistent with the given environment light, as shown in Fig. 3 (c). By distilling this new geometry- and light-aware diffusion, DreamMat is able to

accurately generate materials, enabling rendering photo-realistic images in various environments, outperforming baseline methods by a significant margin, and also being more compatible with modern graphics engines as shown in Fig. 2 (c).

Our contributions are summarized as follows:

- A geometry- and light-aware diffusion model to generate images consistent with geometric and light contexts.
- A novel framework for text-guided material generation on specified meshes with high-quality albedo, roughness, and metallic.

2 RELATED WORK

2.1 BRDF estimation

Surface BRDF estimation from images relies on inverse rendering techniques [Barron and Malik 2014; Nimier-David et al. 2019]. Many studies infer the geometry and material properties of real-world objects from image collections under controlled light conditions [Bi et al. 2020b; Nam et al. 2018; Xia et al. 2016] or domain-specific priors [Barron and Malik 2014; DUAN GAO et al. 2019; Guo et al. 2020; Li et al. 2020, 2018; Wimbauer et al. 2022; Ye et al. 2023] to reduce ambiguities in the inverse rendering. With the rise of neural rendering exemplified by NeRF [Mildenhall et al. 2020], inverse rendering has also made significant qualitative progress. To model spatially-varying bidirectional reflectance distribution function (SVBRDF) under more casual capture conditions, many methods [Boss et al. 2021a,b; Cheng et al. 2024; Dave et al. 2022; Munkberg et al. 2022; Sun et al. 2023a; Yariv et al. 2020; Ye et al. 2023; Zhang et al. 2022a, 2021a] have relied on implicit representation to provide geometry prior. Subsequent works improve the quality of the reconstruction results by introducing visibility prediction [Chen et al. 2022b; Srinivasan et al. 2021], modeling indirect illumination [Deng et al. 2022; Jin et al. 2023; Yang et al. 2023a; Yao et al. 2022; Zhang et al. 2023c, 2022b], applying Monte Carlo sampling [Hasselgren et al. 2022; Li et al. 2023b; Liu et al. 2023c; Luan et al. 2021; TG et al. 2023; Zhu et al. 2023], utilizing deep polarization information [Deschaintre et al. 2021; Zhao et al. 2022], setting multiple flashlights [Bi et al. 2020a; Cheng et al. 2021; Kuang et al. 2022; Li and Li 2022; Yang et al. 2022], and introducing material priors [Boss et al. 2021b; Zhang et al. 2021b].

Recent works [Gao et al. 2023; Jiang et al. 2023; Liang et al. 2023] apply 3D Gaussian Splatting [Kerbl et al. 2023] to inverse rendering, achieving speed improvements. Other works [Kocsis et al. 2023; Lyu et al. 2023] integrate a diffusion model into the traditional inverse rendering framework. Their improved decomposition results can be attributed to the strong learned prior of diffusion models trained on large-scale real-world images.

Contrary to the aforementioned works that perform inverse rendering based on ground truth images, our endeavor is not one of reconstruction but generation. In our framework, both geometry and lighting conditions are predefined, facilitating the generation of materials through a novel application of material decomposition integrated with a 2D diffusion model. Our method builds upon the second stage of NeRO [Liu et al. 2023c] and employs a simplified Disney BRDF [Burley and Studios 2012] to regulate material parameters across various established lighting scenarios.

2.2 Text-guided 3D appearance generation

Recent advances in large language models and image diffusion techniques demonstrate their remarkable capability for text-to-3D generation. The pioneering work Dreamfusion [Poole et al. 2022] first proposes the SDS loss to distill 3D assets from pre-trained text-to-image diffusion models. This idea inspires a series of following works to extend text-to-3D generation to material [Chen et al. 2023a; Liu et al. 2023a; Xu et al. 2023; Youwang et al. 2023], image-to-3D [Liu et al. 2023f,d; Metzer et al. 2022; Qian et al. 2023; Tewari et al. 2023; Zhou and Tulsiani 2023], and higher quality text-to-3D [Katzir et al. 2023; Li et al. 2023a; Lin et al. 2023; Sun et al. 2023b; Wang et al. 2023; Yu et al. 2023b; Zhu and Zhuang 2023] generation.

A notable trend in this field is the texture generation. TEXTure [Richardson et al. 2023] and Text2Tex [Chen et al. 2023c] use an iteratively inpainting method for texturing the given mesh with depth-conditioned Stable Diffusion. However, they exhibit noticeable artifacts at the viewpoint junctions. Subsequent works [Cao et al. 2023; Cheskidova et al. 2023; Knodt and Gao 2023; Liu et al. 2023e; Oh et al. 2023; Tang et al. 2023; Tang and He 2023; Yu et al. 2023a; Zhang et al. 2023b] improve the 3D consistency and quality of the generated textures. Many works on generating textures for human [AlBahar et al. 2023; Huang et al. 2023; Ma et al. 2023; Svitov et al. 2023] or rooms [Chen et al. 2023b; Höllein et al. 2023; Wen et al. 2023; Yang et al. 2023b] have also been proposed. However, the generated RGB texture maps often contain baked-in highlights or shadows, disabling realistic rendering in downstream tasks.

To address this limitation, recent works attempt to incorporate BRDF to generate more realistic appearances. Previous Methods [Sartor and Peers 2023; Vecchio et al. 2023a,b; Zhou et al. 2022] can generate PBR material maps given the input image. However, these methods primarily generate materials in the 2D image space and cannot generate appearances for 3D meshes. TANGO [Chen et al. 2022a] employs CLIP [Radford et al. 2021] loss to generate both lighting and material properties simultaneously for a given mesh. However, this method produces materials with less detail and the use of position encoding can lead to grid-like artifacts. Fantasia3D [Chen et al. 2023a] integrates material decomposition into a distillation-based text-to-3D framework but faces challenges with baked-in shading effects in albedo. There are some concurrent works Paint3D [Zeng et al. 2023], UniDream [Liu et al. 2023a] which train diffusion models to generate lightless albedo. In comparison, our method does not require ground-truth albedo data for training and is based on distillation loss. The concurrent work Matlaber [Xu et al. 2023] introduces a material prior distribution to address the ill-posed problem while our method relies on the fixed environmental light and light-aware diffusion model. Another concurrent work Paint-it [Youwang et al. 2023] is similar to Fantasia3D but with CNN-parameterized materials.

3 METHOD

Overview. Given an untextured mesh and a textual prompt, the target of our method is to generate high-quality Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF) materials on the mesh according to the text prompt. An overview of our method is illustrated in Fig. 4. First, to represent the SVBRDF

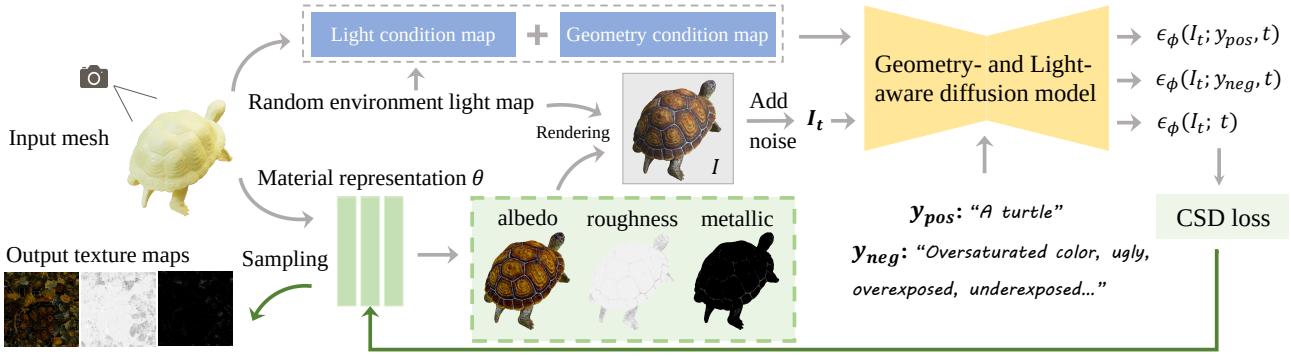

Fig. 4. **Overview of our pipeline.** DreamMat distills a diffusion model to generate PBR materials. We first use Monte Carlo sampling to render images of the object from its material representation and a randomly-selected predefined environment light. Then, we train the material representation by CSD loss on rendered images using a geometry- and light-aware diffusion model.

material, we adopt the hash-grid-based representation to store the SVBRDF parameters and apply the rendering equation to render images of the given mesh from this representation (see Sec. 3.1). Then, as introduced in Sec. 3.2, we add noises to the rendered images and apply a diffusion model to denoise the image, which results in a distillation loss to learn the parameters in the hash-grid representation. To avoid the baked-in lighting and shadows, we adopt a geometry- and light-aware diffusion model as the distillation diffusion model as stated in Sec. 3.3. Finally, the generated materials can be exported to contract material maps defined on the provided or extracted UV map of the mesh, which can be used in editing or arbitrary modern graphics engines.

3.1 Material Representation

In this section, we introduce our material representation and how to render images from these representations. We use the hash-grid-based representation from Instant-NGP [Müller et al. 2022] to represent the simplified Disney BRDF [Burley and Studios 2012]. The BRDF parameters on a point $\mathbf{p}$, including albedo $\mathbf{c}$, roughness α, and metalness m, are computed by:

$$(\mathbf{c}, \alpha, m) = \Gamma_\theta(\mathbf{p}), \tag{1}$$

where Γ_θ means the hash-grid-based representation with corresponding material parameters θ. Our goal is to learn the trainable parameters θ so that we can compute the BRDF parameters for any given point $\mathbf{p}$ on the object surface.

Following the rendering equation [Kajiya 1986], the rendering color $L(\mathbf{p}, \omega_\mathbf{o})$ for the point $\mathbf{p}$ on the direction ω_o is:

$$L(\mathbf{p}, \omega_\mathbf{o}) = \int_\Omega L_i(\omega_\mathbf{i}) f(\omega_\mathbf{i}, \omega_\mathbf{o})(\omega_\mathbf{i} \cdot \mathbf{n}) d\omega_i, \tag{2}$$

where $L(\omega_\mathbf{i})$ is the input environmental light, $\mathbf{n}$ is the normal direction, and the BRDF $f(\omega_\mathbf{i}, \omega_\mathbf{o})$ is a Cook-Torrance microfacet specular shading model [Cook and Torrance 1982] which is defined as:

$$f(\omega_\mathbf{i}, \omega_\mathbf{o}) = \frac{D\,F\,G}{4(\omega_\mathbf{o} \cdot \mathbf{n})(\omega_\mathbf{i} \cdot \mathbf{n})}, \tag{3}$$

where D, G and F are functions representing the GGX [Cook and Torrance 1982], the normal distribution function (NDF), geometric attenuation and Fresnel term, respectively.

Following the methodology proposed by [Karis and Games 2013], we adopt an importance-based Monte Carlo (MC) sampling strategy to separate the rendering equation into diffuse and specular components:

$$L(\mathbf{p}, \omega_\mathbf{o}) = L_\text{diffuse} + L_\text{specular}, \tag{4}$$

$$L_\text{diffuse} = \frac{\mathbf{c}}{N_d} \sum_{i=1}^{N_d} L(\omega_i), \tag{5}$$

$$L_\text{specular} = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{F(\mathbf{c}, m) G(\omega_o, \omega_i, \mathbf{n}, \alpha)(\omega_o \cdot \mathbf{h})}{(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \omega_o)} L(\omega_i), \tag{6}$$

where N_d and N_s denote the number of samples for diffuse and specular components, $\mathbf{h} = (\omega_i + \omega_o)/|\omega_i + \omega_o|$ is the half-way vector. The diffuse component is evaluated using a cosine-weighted hemisphere sampling, while the specular component employs sampling based on the GGX distribution.

Environmental lights. In an inverse rendering pipeline [Zhang et al. 2023c], both the environmental lights $L(\omega_i)$ and the material parameters θ should be estimated. Since we only want to generate material parameters for the given object, we fix the environment lights by randomly selecting a known HDR image as the environment light. This makes the inverse rendering problem less ill-posed for better material generation.

3.2 Distillation Loss for Material Generation

The material representation is randomly initialized at first and we follow the Score-Distillation Sampling (SDS) loss [Poole et al. 2022] to distill a text-to-image diffusion model to learn the parameters of the material representation. Given a rendered image I from the current material representation, we add a noise ϵ_t to it to get a noisy image $I_t = I + \epsilon_t$. Then the noisy image is denoised by the diffusion model to get a denoised image I'_t conditioned on the text prompt y_pos and the difference between the denoised image and the rendered image $\delta(I_t) = I'_t - I$ is computed as the distillation loss

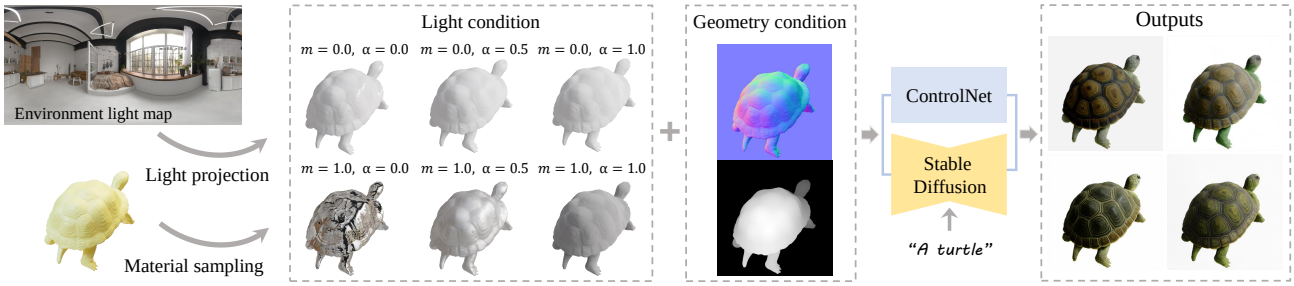

Fig. 5. **Our geometry- and light-aware diffusion model** uses an object's normal and depth maps as geometry conditions and six predefined materials with a given environment light as lighting conditions. Our model generates images that align with the given geometry and environment light.

$\mathcal{L}_{\text{Distill}}$ and we use the following gradient to optimize θ:

$$\nabla_\theta \mathcal{L}_{\text{Distill}} = \mathbb{E}_t \left[\delta(I_t) \frac{\partial I}{\partial \theta} \right]. \tag{7}$$

By denoising I_t to I'_t using the diffusion model, I'_t will be more consistent with the given text prompt y_{pos} than I. Thus, minimizing the difference $\delta(I_t) = I'_t - I$ makes the rendered image more aligned with the given text prompts. Thus, Eq. (7) gradually optimizes the material parameters θ to make the renderings consistent with the input texts.

Here, we adopt a variant of the original SDS loss [Poole et al. 2022] called Classifier Score Distillation (CSD) loss [Yu et al. 2023b], which shows better distillation performance than the SDS loss as demonstrated in Fig. 9. CSD loss is also defined by Eq. (7) but utilizes all the null, positive, and negative prompts to compute $\delta(I_t)$ as follows:

$$\delta(I_t) = \eta_1 \epsilon_\phi(I_t; y_{\text{pos}}, t) + (\eta_2 - \eta_1)\epsilon_\phi(I_t; t) - \eta_2 \epsilon_\phi(I_t; y_{\text{neg}}, t), \tag{8}$$

where t is a sampled time step, ϵ_ϕ means the noise predictor of the diffusion model, η_1 and η_2 are two predefined coefficients, and y_{pos} and y_{neg} are the positive and negative text prompts. In Fig. 4, y_{pos} represents the input text prompts, such as "A turtle", while y_{neg} contains predefined negative text prompts such as "oversaturated color", "ugly", "underexposed", and "overexposed".

Discussion. However, simply applying the Stable Diffusion model here for distillation would lead to erroneous material parameters in two aspects. First, the generated materials may not be consistent with the given geometry. An example is shown in Fig. 8 (a) because the diffusion model may generate images aligned with the prompt but not aligned with the geometry. Second, we render the image using a predefined environment light, but the generated images of the diffusion model may not be consistent with the given environment light as shown in Fig. 3. Inconsistencies can result in materials with incorrect albedo, including baked-in shadows or highlights as shown in Fig. 8 (b). To align images with text prompts, geometry, and environment light, we created a diffusion model that considers both geometry and light.

3.3 Geometry- and Light-aware Diffusion Model

We finetune the Stable Diffusion model by adding additional geometry and light conditions with ControlNet [Zhang et al. 2023a]. As shown in Fig. 5, the geometry condition is the rendered depth and normal maps. To represent the light condition, we assign a set of predefined materials to the object, which have the same white albedo color but different metallic and roughness. Then, we render images of this object using the given environment light and these predefined materials, which are used as the light condition to the ControlNet. Finally, we finetune the ControlNet with both light and geometry conditions on the Objaverse [Deitke et al. 2023] dataset. The output examples of the finetuned ControlNet are shown in Fig. 5. The resulting diffusion model generates images that are consistent with the geometry and the lights, which are used to calculate CSD loss to generate high-quality materials.

3.4 Material Generation

In this section, we summarize the material generation process of DreamMat, as shown in Fig. 4. Given the input mesh, we first precompute its light conditions on 128 random viewpoints using 5 predefined different environment lights because calculating the light conditions on the fly is time-consuming. During each distillation step, we randomly select a viewpoint and an environment light to render an image on the mesh using the material representation as stated in Sec. 3.1. Then, we add noise to the rendered image. The noisy rendered image is used in the computation of the CSD loss stated in Sec. 3.2 with the geometry- and light-aware diffusion model in Sec. 3.3, which uses the corresponding geometry and lighting condition of this viewpoint and this environment light. The CSD loss is backward to optimize the parameters in the material representation. Except for the CSD loss, following the previous works [Liu et al. 2023c; Yang et al. 2023a; Zhang et al. 2022b], we apply a material smoothness loss

$$\mathcal{L}_{\text{smooth}} = ||\Gamma_\theta(\mathbf{p}) - \Gamma_\theta(\mathbf{p} + \epsilon)||^2, \tag{9}$$

where ϵ is a small random perturbation vector sampled from a Gaussian noise with 0.05 as its standard deviation. This smoothness loss $\mathcal{L}_{\text{smooth}}$ makes the predicted materials (roughness, metallic, and albedo) more smooth on the mesh. Finally, we sample the trained material representation Γ_θ to construct material UV maps for compatibility with modern rendering engines.

4 EXPERIMENTS

4.1 Implementation Details

We train the geometry- and light-aware ControlNet from the images which are rendered on the objects in the LVIS subset of the Objaverse

Fig. 6. **Qualitative comparison.** We compared our method to TANGO [Chen et al. 2022a], TEXTure [Yu et al. 2023a], Text2Tex [Chen et al. 2023c], and Fantasia3D [Chen et al. 2023a]. We use NvDiffRec [Munkberg et al. 2022] to decompose the texture map produced by TEXTure and Text2Tex. Each object has three images: the albedo map on the left, the rendered image on the top right, and the roughness map on the bottom right.

Method	TANGO	TEXTure	Text2Tex	Fantasia3D	Ours
Overall Qual.	1.77	3.00	3.04	2.97	**4.39**
Text Fidelity	2.32	3.34	3.26	3.26	**4.41**
Albedo Qual.	1.67	2.92	2.73	2.95	**4.65**
Roughness Qual.	2.21	2.48	2.63	2.57	**4.41**
Metallic Qual.	1.75	2.71	2.75	3.01	**4.53**
Light/Mat. Disen.	1.95	2.52	2.49	2.92	**4.36**
Rendering Qual.	1.37	3.01	3.04	3.10	**4.75**

Table 1. **User study** conducted with 42 respondents. This table shows the average scores given by participants. We ask them to evaluate the albedo, roughness, metallic, and renderings of TANGO [Chen et al. 2022a], TEXTure [Yu et al. 2023a], Text2Tex [Chen et al. 2023c], and Fantasia3D [Chen et al. 2023a] to give scores in [1, 5], where a higher score means a better result. The evaluation criteria included overall quality, fidelity to the text prompt, effectiveness of albedo disentanglement from shading effects ("Light/Mat. Disen"), individual quality of the different material maps and quality of rendering under new environment lights rather than those used in generation. Scores were averaged across all responses and examples. The questionnaire used in this user study is in the supplementary material.

[Deitke et al. 2023]. Since the names and tags of objects in this dataset are rather noisy, we employ BLIP [Li et al. 2022] for captioning all rendered images. Following [Liu et al. 2023b], we render 16 random views for every object under randomly chosen environment light maps. The light condition maps are obtained by using ray tracing in Blender, which represents the radiance for different materials under the environment light. For normal maps, we transform the model's normal vectors into view space and flip the x-axis following ScanNet's [Dai et al. 2017] protocol. Depth maps are processed by inverting the real depth values and normalizing them. The filtered subset contains 1,242,880 entries, each consisting of conditional images and one rendered image. Our diffusion model is based on Stable Diffusion v2.1 and is trained with a batch size of 256 for 3 epochs, utilizing 8 V100 GPUs. More training details and results are presented in the appendix.

Our material generation pipeline is implemented in ThreeStudio [Guo et al. 2023]. As rendering light conditions is time-consuming, we randomly sample 128 viewpoints and render images with five different environment lights using Blender. In each iteration, we randomly select a viewpoint and environment light, optimizing the materials for 4,000 steps using an Adam optimizer with learning rate 0.01. We set the control scale at 1.0 in the ControlNet with a gradual decay to 0.8 after 700 steps. Adhering to the CSD loss with annealed negative prompts, we set the $\eta_1 = 1.05$ and η_2 was progressively reduced from 1.0 to 0.5.

4.2 Qualitative Results

Baselines. We compare our method with several state-of-the-art methods for 3D appearance generation, namely TANGO [Chen et al. 2022a], TEXTure [Richardson et al. 2023], Text2Tex [Chen et al. 2023c], and Fantasia3D [Chen et al. 2023a]. Since TEXTure and Text2Tex can only generate a single RGB texture map, we utilize NvDiffRec [Munkberg et al. 2022] to post-process the generated textures for material decomposition. As the mesh is given, we employ

Fig. 7. **More generated materials and editing results**.

only the second stage of Fantasia3D to optimize the materials with the fixed geometry, normal, and environment light map.

Comparison with baselines. We take diverse 3D meshes from real game assets and the Objaverse [Deitke et al. 2023] dataset that are not included in the training set. Fig. 6 shows the albedo and roughness generated from the same text prompts and untextured meshes. We also show the renderings of the generated materials under the same environment light. Tango successfully recognizes specific material descriptors like "golden" and "wooden" but falls short in generating detailed textures for objects. TEXTure and Text2Tex only generate pure RGB textures instead of PBR materials. Although the inverse rendering technique is employed for decomposing texture maps into materials, we are still unable to correctly disentangle albedo from environmental lights. Fantasia3D is capable of directly generating PBR materials but tends to retain many highlights and shadows in the albedo, due to the absence of lighting constraints in its generation process. Our method stands out by effectively achieving disentanglement of materials from environmental lighting with material diversity and texture detail preserved.

User study. To further validate the stability and quality of our model, we conduct a user study on the 20 generated materials. Each participant is provided with 5 examples, accompanied by the corresponding text prompts and meshes. They rated the materials generated by each method on various aspects. 42 feedbacks from 37 users were collected from the study with results detailed in Tab. 1.

More results and editing. The high-quality PBR material decoupled from lighting enables photo-realistic rendering in a modern graphics engine like Blender. Fig. 7 shows more results with the albedo, roughness, and metallic material generated for a given 3D model from text prompts, as well as their renderings under different environment lights. Moreover, by adjusting the overall roughness or metallic, or by altering the overall hue of the albedo, we can conveniently edit the material to achieve different visual appearances.

4.3 Quantitative Comparison

To evaluate the usability and robustness of DreamMat, we conduct a quantitative analysis in comparison with baseline methodologies using CLIP score [Hessel et al. 2021] and FID [Heusel et al. 2017] of rendered images as metrics. Our experiment involves generating materials for 10 different 3D meshes and selecting 5 distinct text prompts for each mesh for generation. To compute the metrics, we randomly chose 120 viewpoints for each object to render images. Then, the semantic alignment between the text prompts and the rendered images is quantitatively assessed by the CLIP Score [Hessel et al. 2021], where a higher score indicates a greater similarity between the generated appearance and the text prompts. Furthermore, the quality of the rendered images is evaluated by the Fréchet Inception Distance (FID) [Heusel et al. 2017], which compares the distribution's distance between the images generated by Stable Diffusion and the rendered images from the generated appearances. As

Method	TANGO	TEXTure	Text2Tex	Fantasia3D	Ours
CLIP Score ↑	76.15	78.55	78.52	77.30	**80.28**
FID ↓	165.40	135.72	144.86	131.86	**114.97**

Table 2. **Quantitative results.** We use 50 different text prompts on 10 meshes to generate appearances and calculate the CLIP score (similarity between rendered views and text prompts) and FID (distribution's distance between rendered images from the generated appearances and the generated images by Stable Diffusion) to assess the text fidelity and visual quality of the generated appearances.

illustrated in Tab. 2, DreamMat outperforms the baseline methods in achieving the best text fidelity and superior visual quality of the generated appearances.

4.4 Ablative Study

To validate the effectiveness of each component, we conduct an ablation study on the text prompt "a wooden treasure chest with metal accents and locks" to generate materials for a given untextured treasure chest mesh. The results of the ablation study are shown in Fig. 8 and more ablation study results on other meshes and prompts are included in the supplementary material.

(1) **Baseline distillation method**. In Fig. 8 (a), we directly combine our inverse rendering method with a text-to-image Stable Diffusion model to generate materials. Though some reasonable results are achieved in this baseline method, there is a noticeable inconsistency between the generated material and the given geometry as highlighted by the handle region of the chest.

(2) **Distillation with geometry-aware diffusion models**. An effective way to ensure the consistency between materials and geometry is to adopt the depth and normal-conditioned diffusion model. Fig. 8 (b) shows the distilled materials from a pre-trained normal and depth ControlNet [Zhang et al. 2023a]. This distillation method yields more consistency between the generated materials and the input mesh's geometry, as evidenced by the more pronounced texture details. However, because of the ill-posed nature of material decomposition, the generated albedo contains shading effects like highlights and the roughness and metallic are of low quality.

(3) **Distillation with light-aware diffusion models**. Fig. 8 (c) shows the distilled materials from a light-aware ControlNet without the use of geometry conditions, which results in geometry inconsistency. Although lighting conditions already carry geometric clues, primarily through shadows, when there are not adequate shadows, only using light conditions is not enough to capture accurate geometry.

(4) **Fixed environment light with geometry- and light-aware diffusion model**. In Fig. 8 (d), we apply the proposed geometry- and light-aware diffusion model to distill the material. However, instead of randomly selecting an environment light during the distillation, we always choose the same environment light. The results show that applying the geometry- and light-aware diffusion model improves the quality of the generated materials with better roughness and metallic while using the same environment light leads to overfitting in the inverse rendering. Thus, the resulting albedo still contains incorrect highlights.

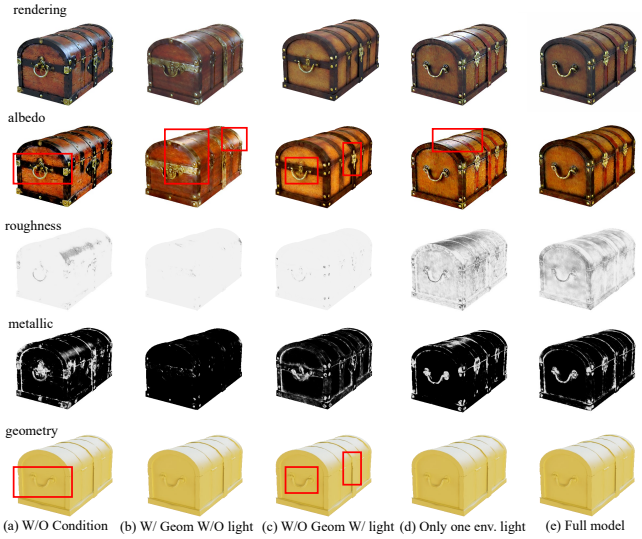

Fig. 8. **Ablation study** on the text prompt "a wooden treasure chest with metal accents and locks" applied to an untextured treasure chest mesh. (a) Baseline distillation method on a Stable Diffusion method with our inverse rendering scheme. (b) Adding "Geom" conditions, i.e. normal maps and depth maps, enables geometry-consistent material generation. (c) Employing solely lighting conditions in the absence of geometric constraints. (d) Distilling our geometry- and light-aware diffusion model but only using one environment light in the distillation. (e) Our full model with geometry- and light-aware diffusion model and randomly selected environment light.

(5) **Full model**. Fig. 8 (e) shows the results of our full method, which distills the geometry- and light-aware diffusion model and randomly selects an environment light in the distillation. Introducing the light condition enables the generation of light-consistent images. Meanwhile, by varying the light maps across iterations, we prevent the material's properties from overfitting to one environment light and remove the shading effects in the albedo. This produces the materials of the best quality and photorealistic shading results.

Discussion on distillation losses. In Fig. 9, we show the effects of using different distillation losses and different diffusion models. In Fig. 9(a), we adopt the SDS loss as DreamFusion [Poole et al. 2022], incorporating a ControlNet with Canny edges and depth for distillation. In Fig. 9(b), the CSD loss, following the texture generation approach described in [Yu et al. 2023c], also employs the ControlNet with depth and Canny edges as conditions. The material representation and rendering approach remains the same as DreamMat. CSD loss significantly reduces oversaturation and shows more details compared to SDS loss, yielding a more realistic appearance. However, CSD alone tends to incorporate lighting information into the albedo. Our solution, as shown in Fig. 9 (c), introduces a geometry- and light-aware diffusion model that better separates material properties from lighting, enhancing geometric fidelity. We also include results of using SDS loss with our geometry- and light-aware model in the Appendix A.6.

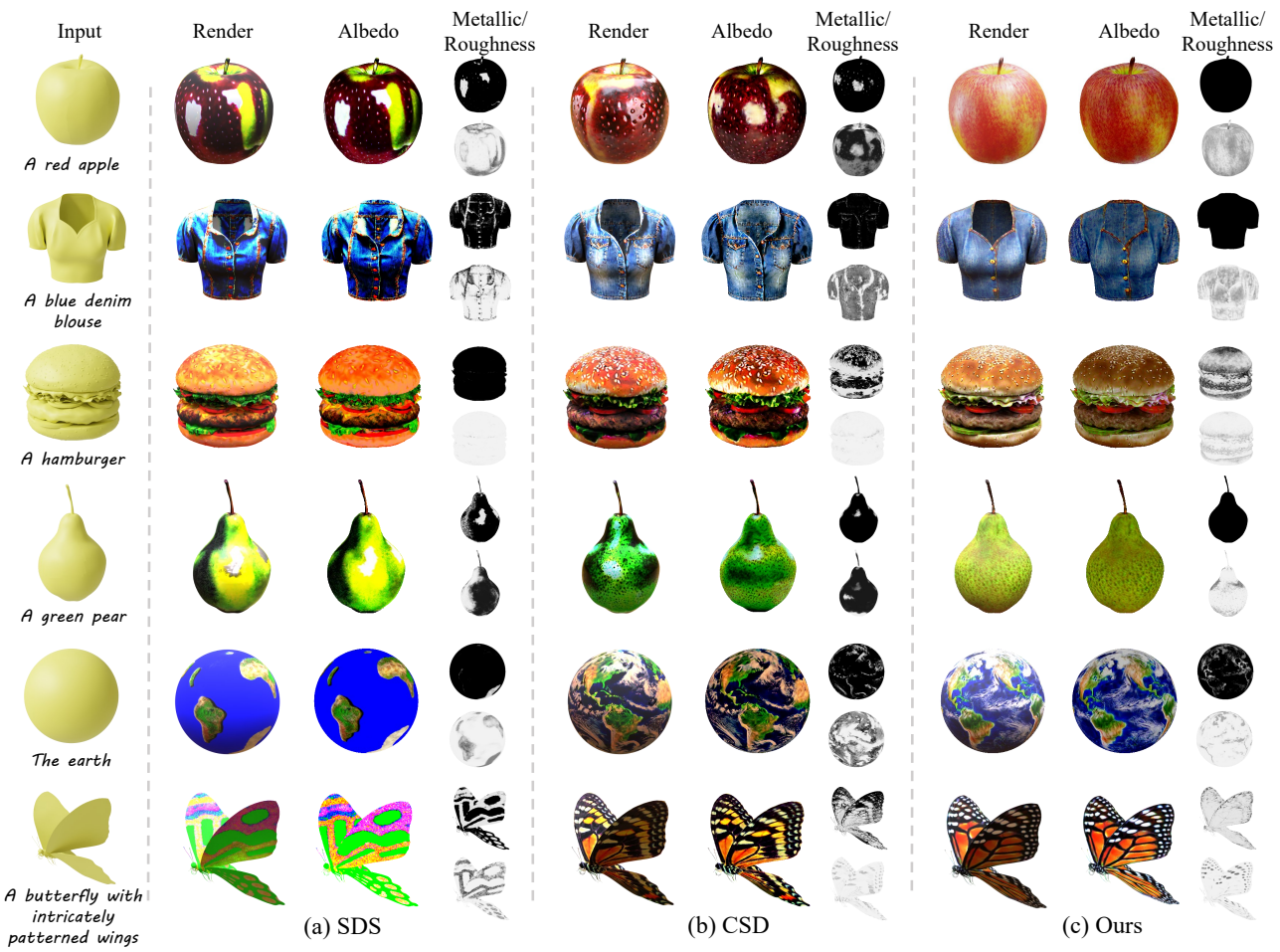

Fig. 9. **Comparison between the vanilla SDS loss, the vanilla CSD loss, and our method.** (a) Materials generated with vanilla SDS loss [Poole et al. 2022] and geometry condition. (b) Material generated using vanilla CSD loss [Yu et al. 2023b] with geometry condition. (c) Materials generated by our method which combines the CSD loss with our geometry- and light-aware diffusion model.

4.5 Generating Diverse Materials

Our method is able to generate different materials given different prompts on the same mesh. Fig. 10 shows the materials and the rendering results of the same mesh generated from different text prompts by the proposed methods. These results demonstrate that our method is able to generate diverse materials of different styles that align well with the text prompts with high fidelity.

4.6 Material Generation of Complex Objects

We demonstrate the generative capabilities of our method on a set of challenging examples, as shown in Fig. 11, which include complex self-occlusions, assemblies of multiple parts, and a variety of material compositions. Our method can directly take the whole mesh as input and generate textures on the mesh according to the given text prompt. Then, we export the generated materials into 2048x2048 resolution albedo, roughness, and metallic texture maps, which are utilized in the Blender to render the photo-realistic images in Fig. 11.

4.7 Material Generation for Object Sets

For a cluttered scene or a highly-detailed avatar, as shown in Fig. 12, it is difficult to directly generate materials for the whole scene or the whole body in one distillation process. Instead, we generate materials for each component separately and then combine these parts together to get the generated appearances for these two extremely complicated examples. For the "fruit" scene, we generate materials for each fruit separately and combine them. For the "avatar" mesh, the top-right figure shows the separated components with different colors. Utilizing ray tracing for rendering, our demonstration showcases the compatibility of our method with modern computer graphics rendering pipelines. The results achieved are of photo-realistic quality, illustrating the effectiveness of our approach in producing visuals that closely mimic real-world appearances.

4.8 Runtime Analysis

We conduct a runtime analysis of our method on one NVIDIA RTX 4090 graphics card. For the "Transformers" mesh displayed in Fig. 11,

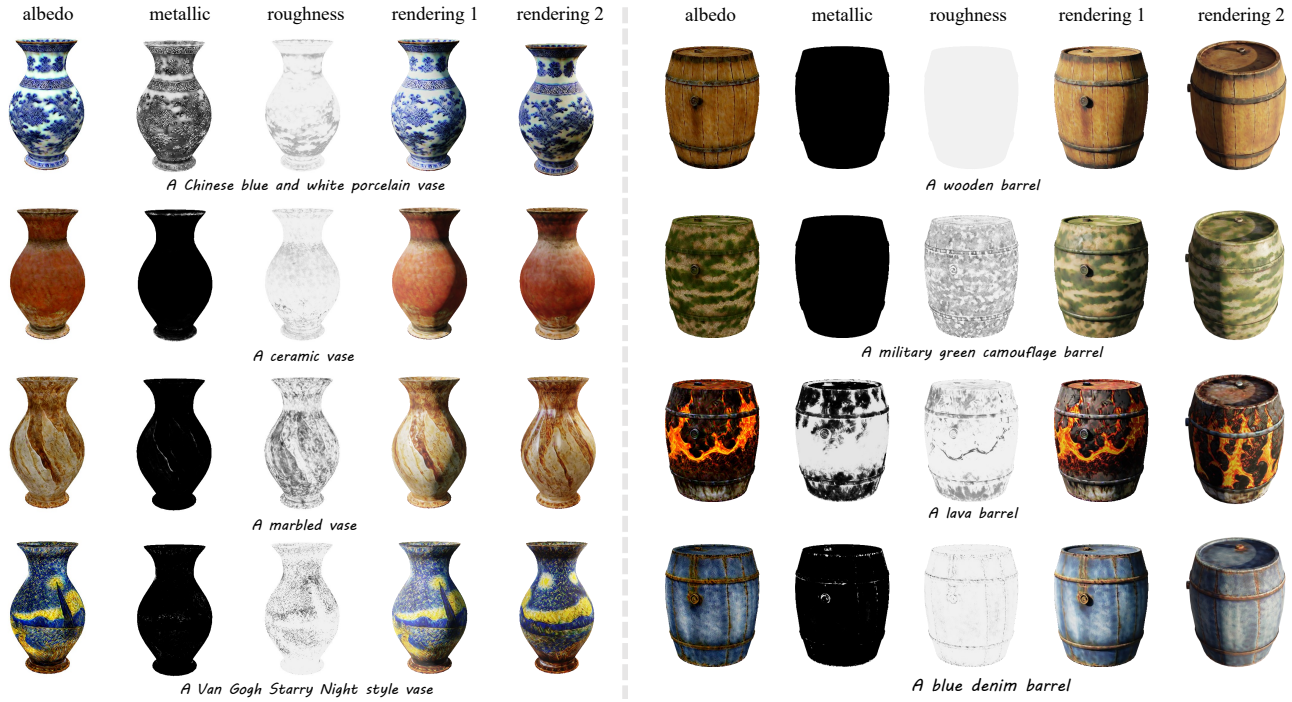

Fig. 10. **Diverse material generation**. Our method can generate different materials with different text prompts on the same mesh.

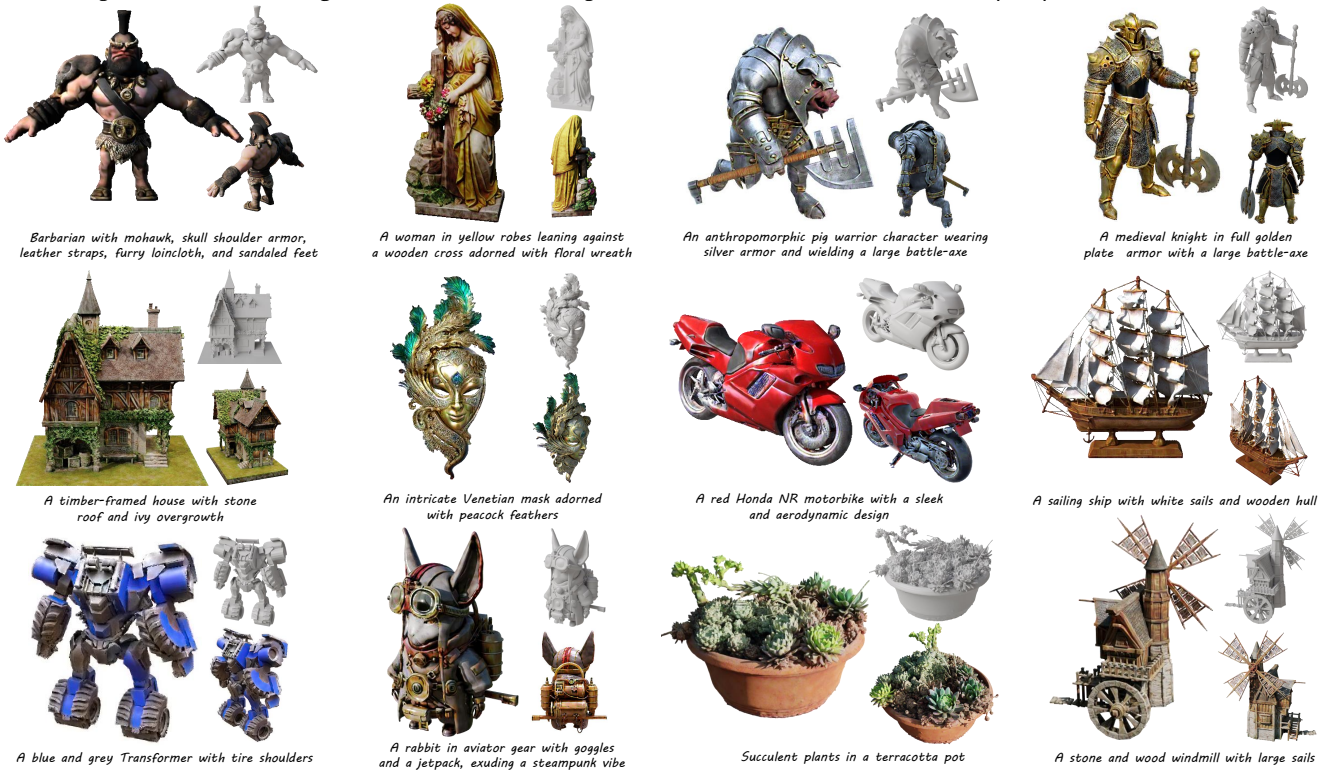

Fig. 11. **Material generation of complex objects**. On the given complex meshes shown in the right-top, our method can generate high-quality materials for the whole mesh in one distillation process, which enables photo-realistic renderings in Blender (left and right-bottom parts).

Fig. 12. **Combining generated materials of different components**. For extremely complex examples, our method can generate materials for each component of the meshes separately and then combine them to get the materials for the whole mesh.

our material distillation process costs about 18 minutes. The rendering and optimization of the material cost about 2/3 time (12 min) while querying the diffusion model costs about 1/3 time (6 min).

5 LIMITATIONS AND CONCLUSIONS

Limitations. Although our method successfully generates diverse and high-quality appearances according to the text prompts, it still has several limitations. Due to the ill-posed nature of material decomposition, DreamMat still exhibits inaccurate metallic and roughness in some cases. Also, our method has difficulty in dealing with materials that exhibit properties like transparency, high reflection, or subsurface scattering. This limitation is primarily caused by our choice of the Bidirectional Reflectance Distribution Function (BRDF) model, which cannot model more advanced and complex material. Meanwhile, the proposed method only accounts for the direct lights from the environment map but does not consider the indirect lights reflected from the object itself, which may lead to incorrect materials for highly reflective objects. Considering the indirect lights and more advanced BRDF may resolve this limitation but also bring more computation complexity in distillation, which we leave for future works. Another limitation is that the distillation of our method takes a relatively long time for a high-quality generation (about 20 minutes) while designers may want to use the material generation in an interactive environment. Our method may be further sped up with recent faster diffusion models and advanced representations in future works. Since DreamMat is based on Stable Diffusion [Rombach et al. 2022], it shares some limitations of Stable Diffusion, such as difficulty in precisely controlling the individual material components solely through text prompts.

Conclusions. In summary, we present a novel text-guided technique for generating detailed PBR materials specifically for given untextured 3D meshes. Our method includes a geometry- and light-aware diffusion model and an inverse rendering-based distillation method. The inverse rendering method renders images by Monte Carlo sampling and distills materials by a CSD loss. The key advantage of our method is the geometry- and light-aware diffusion model which can generate images consistent with the geometry and environment light. Distilling from this diffusion model avoids the common problem of baking shading effects into albedo. We demonstrate that the generated materials by our method are readily usable in modern graphics engines, offering enhanced realism for various applications in gaming and simulation.

6 ACKNOWLEDGMENTS

Xiaogang Jin was supported by the Key R&D Program of Zhejiang (No. 2023C01047) and the FDCT under Grant 0002/2023/AKP. This research work was supported by Information Technology Center and State Key Lab of CAD&CG, ZheJiang University.

REFERENCES

Badour AlBahar, Shunsuke Saito, Hung-Yu Tseng, Changil Kim, Johannes Kopf, and Jia-Bin Huang. 2023. Single-Image 3D Human Digitization with Shape-guided Diffusion. In *SIGGRAPH Asia*. 1–11.

Jonathan T Barron and Jitendra Malik. 2014. Shape, illumination, and reflectance from shading. *TPAMI* 37, 8 (2014), 1670–1687.

Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020a. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824* (2020).

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. 2020b. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *CVPR*.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. 2021a. Nerd: Neural reflectance decomposition from image collections. In

CVPR.

Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. 2021b. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *NeurIPS*.

Brent Burley and Walt Disney Animation Studios. 2012. Physically-based shading at disney. In *SIGGRAPH*.

Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and KangXue Yin. 2023. TexFusion: Synthesizing 3D Textures with Text-Guided Image Diffusion Models. In *ICCV*.

Dave Zhenyu Chen, Haoxuan Li, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023b. Scenetex: High-quality texture synthesis for indoor scenes via diffusion priors. *arXiv preprint arXiv:2311.17261* (2023).

Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023c. Text2Tex: Text-driven Texture Synthesis via Diffusion Models. In *ICCV*.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023a. Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *ICCV*.

Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. 2022a. TANGO: Text-driven Photorealistic and Robust 3D Stylization via Lighting Decomposition. In *NeurIPS*.

Ziyu Chen, Chenjing Ding, Jianfei Guo, Dongliang Wang, Yikang Li, Xuan Xiao, Wei Wu, and Li Song. 2022b. L-Tracing: Fast Light Visibility Estimation on Neural Surfaces by Sphere Tracing. In *ECCV*.

Tianhang Cheng, Wei-Chiu Ma, Kaiyu Guan, Antonio Torralba, and Shenlong Wang. 2024. Structure from Duplicates: Neural Inverse Graphics from a Pile of Objects. *arXiv preprint arXiv:2401.05236* (2024).

Ziang Cheng, Hongdong Li, Yuta Asano, Yinqiang Zheng, and Imari Sato. 2021. Multi-view 3d reconstruction of a texture-less smooth surface of unknown generic reflectance. In *CVPR*.

Evgeniia Cheskidova, Aleksandr Arganaidi, Daniel-Ionut Rancea, and Olaf Haag. 2023. Geometry Aware Texturing. In *SIGGRAPH Asia*. 1–2.

Robert L Cook and Kenneth E. Torrance. 1982. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)* 1, 1 (1982), 7–24.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*.

Akshat Dave, Yongyi Zhao, and Ashok Veeraraghavan. 2022. Pandora: Polarization-aided neural decomposition of radiance. In *ECCV*.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A universe of annotated 3d objects. In *CVPR*.

Youming Deng, Xueting Li, Sifei Liu, and Ming-Hsuan Yang. 2022. DIP: Differentiable Interreflection-aware Physics-based Inverse Rendering. *arXiv preprint arXiv:2212.04705* (2022).

Valentin Deschaintre, Yiming Lin, and Abhijeet Ghosh. 2021. Deep polarization imaging for 3D shape and SVBRDF acquisition. In *CVPR*.

Xiao Li DUAN GAO, Pieter Peers, Kun Xu, and Xin Tong. 2019. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Transactions on Graphics (ToG)* 38, 4 (2019), 1–15.

Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. 2023. Relightable 3D Gaussian: Real-time Point Cloud Relighting with BRDF Decomposition and Ray Tracing. *arXiv preprint arXiv:2311.16043* (2023).

Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli, and Shuang Zhao. 2020. MaterialGAN: reflectance capture using a generative SVBRDF model. *ACM Transactions on Graphics (ToG)* 39, 6 (2020), 1–13.

Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. 2023. threestudio: A unified framework for 3D content generation. https://github.com/threestudio-project/threestudio.

Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. 2022. Shape, light, and material decomposition from images using Monte Carlo rendering and denoising. *NeurIPS*.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In *EMNLP*.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. 2023. Text2room: Extracting textured 3d meshes from 2d text-to-image models. *arXiv preprint arXiv:2303.11989* (2023).

Xin Huang, Ruizhi Shao, Qi Zhang, Hongwen Zhang, Ying Feng, Yebin Liu, and Qing Wang. 2023. Humannorm: Learning normal diffusion model for high-quality and realistic 3d human generation. *arXiv preprint arXiv:2310.01406* (2023).

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. 2023. GaussianShader: 3D Gaussian Splatting with Shading Functions for Reflective Surfaces. *arXiv preprint arXiv:2311.17977* (2023).

Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. 2023. TensoIR: Tensorial Inverse Rendering. In *CVPR*.

James T. Kajiya. 1986. The rendering equation. In *SIGGRAPH*.

Brian Karis and Epic Games. 2013. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice* 4, 3 (2013), 1.

Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. 2023. Noise-free score distillation. *arXiv preprint arXiv:2310.17590* (2023).

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (ToG)* 42, 4 (July 2023).

Julian Knodt and Xifeng Gao. 2023. Consistent Mesh Diffusion. *arXiv preprint arXiv:2312.00971* (2023).

Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. 2023. Intrinsic Image Diffusion for Single-view Material Estimation. *arXiv preprint arXiv:2312.12274* (2023).

Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. 2022. NeROIC: Neural Rendering of Objects from Online Image Collections. In *SIGGRAPH*.

Matthias Labschütz, Katharina Krösl, Mariebeth Aquino, Florian Grashäftl, and Stephanie Kohl. 2011. Content creation for a 3D game with Maya and Unity 3D. *Institute of Computer Graphics and Algorithms, Vienna University of Technology* 6 (2011), 124.

Cindy Le, Congrui Hetang, Ang Cao, and Yihui He. 2023. EucliDreamer: Fast and High-Quality Texturing for 3D Models with Stable Diffusion Depth. *arXiv preprint arXiv:2311.15573* (2023).

Chenhao Li, Taishi Ono, Takeshi Uemori, Hajime Mihara, Alexander Gatto, Hajime Nagahara, and Yuseke Moriuchi. 2023b. NeISF: Neural Incident Stokes Field for Geometry and Material Estimation. *arXiv preprint arXiv:2311.13187* (2023).

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*.

Junxuan Li and Hongdong Li. 2022. Neural Reflectance for Shape Recovery with Shadow Handling. In *CVPR*.

Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. 2023a. SweetDreamer: Aligning Geometric Priors in 2D Diffusion for Consistent Text-to-3D. *arxiv:2310.02596* (2023).

Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2020. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *CVPR*.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2018. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia*.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. 2023. GS-IR: 3D Gaussian Splatting for Inverse Rendering. *arXiv preprint arXiv:2311.16473* (2023).

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3D: High-Resolution Text-to-3D Content Creation. In *CVPR*.

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. 2023f. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928* (2023).

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023d. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2023b. SyncDreamer: Learning to Generate Multiview-consistent Images from a Single-view Image. *arXiv preprint arXiv:2309.03453* (2023).

Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. 2023c. NeRO: Neural Geometry and BRDF Reconstruction of Reflective Objects from Multiview Images. In *SIGGRAPH*.

Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. 2023e. Text-Guided Texturing by Synchronized Multi-View Diffusion. *arXiv preprint arXiv:2311.12891* (2023).

Zexiang Liu, Yangguang Li, Youtian Lin, Xin Yu, Sida Peng, Yan-Pei Cao, Xiaojuan Qi, Xiaoshui Huang, Ding Liang, and Wanli Ouyang. 2023a. UniDream: Unifying Diffusion Priors for Relightable Text-to-3D Generation. *arXiv preprint arXiv:2312.08754* (2023).

Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. 2021. Unified shape and svbrdf recovery using differentiable monte carlo rendering. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 101–113.

Linjie Lyu, Ayush Tewari, Marc Habermann, Shunsuke Saito, Michael Zollhöfer, Thomas Leimkühler, and Christian Theobalt. 2023. Diffusion Posterior Illumination for Ambiguity-aware Inverse Rendering. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–14.

Yiwei Ma, Xiaoqing Zhang, Xiaoshuai Sun, Jiayi Ji, Haowei Wang, Guannan Jiang, Weilin Zhuang, and Rongrong Ji. 2023. X-Mesh: Towards Fast and Accurate Text-driven 3D Stylization via Dynamic Textual Guidance. In *ICCV*.

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2022. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. *arXiv preprint arXiv:2211.07600* (2022).

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–15.

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. 2022. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *CVPR*.

Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H. Kim. 2018. Practical SVBRDF Acquisition of 3D Objects with Unstructured Flash Photography. *ACM Transactions on Graphics (ToG)* 37, 6, Article 267 (2018), 12 pages.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Transactions on Graphics (ToG)* 38, 6, Article 203 (2019), 17 pages.

Yeongtak Oh, Jooyoung Choi, Yongsung Kim, Minjun Park, Chaehun Shin, and Sungroh Yoon. 2023. ControlDreamer: Stylized 3D Generation with Multi-View ControlNet. *arXiv preprint arXiv:2312.01129* (2023).

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. In *ICLR*.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. 2023. Magic123: One Image to High-Quality 3D Object Generation Using Both 2D and 3D Diffusion Priors. *arXiv preprint arXiv:2306.17843* (2023).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. Texture: Text-guided texturing of 3d shapes. In *SIGGRAPH*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*.

Sam Sartor and Pieter Peers. 2023. MatFusion: A Generative Diffusion Model for SVBRDF Capture. In *SIGGRAPH Asia*.

Sketchfab. [n. d.]. Sketchfab - The best 3D viewer on the web. https://www.sketchfab.com

Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. 2021. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*.

Cheng Sun, Guangyan Cai, Zhengqin Li, Kai Yan, Cheng Zhang, Carl Marshall, Jia-Bin Huang, Shuang Zhao, and Zhao Dong. 2023a. Neural-PBIR reconstruction of shape, material, and illumination. In *CVPR*.

Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. 2023b. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. *arXiv preprint arXiv:2310.16818* (2023).

David Svitov, Dmitrii Gudkov, Renat Bashirov, and Victor Lempitsky. 2023. DINAR: Diffusion Inpainting of Neural Textures for One-Shot Human Avatars. In *ICCV*.

Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. 2023. MVDiffusion: Enabling Holistic Multi-view Image Generation with Correspondence-Aware Diffusion. (2023).

Zhibin Tang and Tiantong He. 2023. Text-guided High-definition Consistency Texture Model. *arXiv preprint arXiv:2305.05901* (2023).

Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Joshua B. Tenenbaum, Frédo Durand, William T. Freeman, and Vincent Sitzmann. 2023. Diffusion with Forward Models: Solving Stochastic Inverse Problems Without Direct Supervision. In *NeurIPS*.

Thomson TG, Jeppe Revall Frisvad, Ravi Ramamoorthi, and Henrik Wann Jensen. 2023. Neural BSSRDF: Object Appearance Representation Including Heterogeneous Subsurface Scattering. *arXiv preprint arXiv:2312.15711* (2023).

Giuseppe Vecchio, Rosalie Martin, Arthur Roullier, Adrien Kaiser, Romain Rouffet, Valentin Deschaintre, and Tamy Boubekeur. 2023a. ControlMat: A Controlled Generative Approach to Material Capture. *arXiv preprint arXiv:2309.01700* (2023).

Giuseppe Vecchio, Renato Sortino, Simone Palazzo, and Concetto Spampinato. 2023b. MatFuse: Controllable Material Generation with Diffusion Models. *arXiv preprint arXiv:2308.11408* (2023).

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In *NeurIPS*.

Zehao Wen, Zichen Liu, Srinath Sridhar, and Rao Fu. 2023. AnyHome: Open-Vocabulary Generation of Structured and Textured 3D Homes. *arXiv preprint arXiv:2312.06644* (2023).

Felix Wimbauer, Shangzhe Wu, and Christian Rupprecht. 2022. De-rendering 3d objects in the wild. In *CVPR*.

Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. 2016. Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Transactions on Graphics (ToG)* 35, 6 (2016), 1–12.

Xudong Xu, Zhaoyang Lyu, Xingang Pan, and Bo Dai. 2023. MATLABER: Material-Aware Text-to-3D via LAtent BRDF auto-EncodeR. *arXiv preprint arXiv:2308.09278* (2023).

Bangbang Yang, Wenqi Dong, Lin Ma, Wenbo Hu, Xiao Liu, Zhaopeng Cui, and Yuewen Ma. 2023b. DreamSpace: Dreaming Your Room Space with Text-Driven Panoramic Texture Propagation. *arXiv preprint arXiv:2310.13119* (2023).

Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. 2022. PS-NeRF: Neural Inverse Rendering for Multi-view Photometric Stereo. In *ECCV*.

Ziyi Yang, Yanzhen Chen, Xinyu Gao, Yazhen Yuan, Yu Wu, Xiaowei Zhou, and Xiaogang Jin. 2023a. SIRe-IR: Inverse Rendering for BRDF Reconstruction with Shadow and Illumination Removal in High-Illuminance Scenes. *arXiv preprint arXiv:2310.13030* (2023).

Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. 2022. Neilf: Neural incident light field for physically-based material estimation. In *ECCV*.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. 2020. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *NeurIPS*.

Weicai Ye, Shuo Chen, Chong Bao, Hujun Bao, Marc Pollefeys, Zhaopeng Cui, and Guofeng Zhang. 2023. Intrinsicnerf: Learning intrinsic neural radiance fields for editable novel view synthesis. In *ICCV*.

Jounathan Young. 2021. xatlas. https://github.com/jpcy/xatlas.git

Kim Youwang, Tae-Hyun Oh, and Gerard Pons-Moll. 2023. Paint-it: Text-to-Texture Synthesis via Deep Convolutional Texture Map Optimization and Physically-Based Rendering. *arXiv preprint arXiv:2312.11360* (2023).

Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. 2023a. Texture Generation on 3D Meshes with Point-UV Diffusion. In *ICCV*.

Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. 2023b. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415* (2023).

Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. 2023c. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415* (2023).

Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. 2023. Paint3D: Paint Anything 3D with Lighting-Less Texture Diffusion Models. *arXiv preprint arXiv:2312.13913* (2023).

Junwu Zhang, Zhenyu Tang, Yatian Pang, Xinhua Cheng, Peng Jin, Yida Wei, Wangbo Yu, Munan Ning, and Li Yuan. 2023b. Repaint123: Fast and High-quality One Image to 3D Generation with Progressive Controllable 2D Repainting. *arXiv preprint arXiv:2312.13271* (2023).

Jingyang Zhang, Yao Yao, Shiwei Li, Jingbo Liu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. 2023c. NeILF++: Inter-Reflectable Light Fields for Geometry and Material Estimation. *arXiv preprint arXiv:2303.17147* (2023).

Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. 2022a. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *CVPR*.

Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021a. PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting. In *CVPR*.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023a. Adding Conditional Control to Text-to-Image Diffusion Models. In *ICCV*.

Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. 2021b. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)* 40, 6 (2021), 1–18.

Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. 2022b. Modeling Indirect Illumination for Inverse Rendering. In *CVPR*.

Jinyu Zhao, Yusuke Monno, and Masatoshi Okutomi. 2022. Polarimetric multi-view inverse rendering. *TPAMI* (2022).

Xilong Zhou, Milos Hasan, Valentin Deschaintre, Paul Guerrero, Kalyan Sunkavalli, and Nima Khademi Kalantari. 2022. TileGen: Tileable, Controllable Material Generation and Capture. In *SIGGRAPH Asia*.

Zhizhuo Zhou and Shubham Tulsiani. 2023. SparseFusion: Distilling View-conditioned Diffusion for 3D Reconstruction. In *CVPR*.

Jingsen Zhu, Yuchi Huo, Qi Ye, Fujun Luan, Jifan Li, Dianbing Xi, Lisha Wang, Rui Tang, Wei Hua, Hujun Bao, et al. 2023. I2-SDF: Intrinsic Indoor Scene Reconstruction and Editing via Raytracing in Neural SDFs. In *CVPR*.

Junzhe Zhu and Peiye Zhuang. 2023. HiFA: High-fidelity Text-to-3D Generation with Advanced Diffusion Guidance. *arXiv preprint arXiv:2305.18766* (2023).

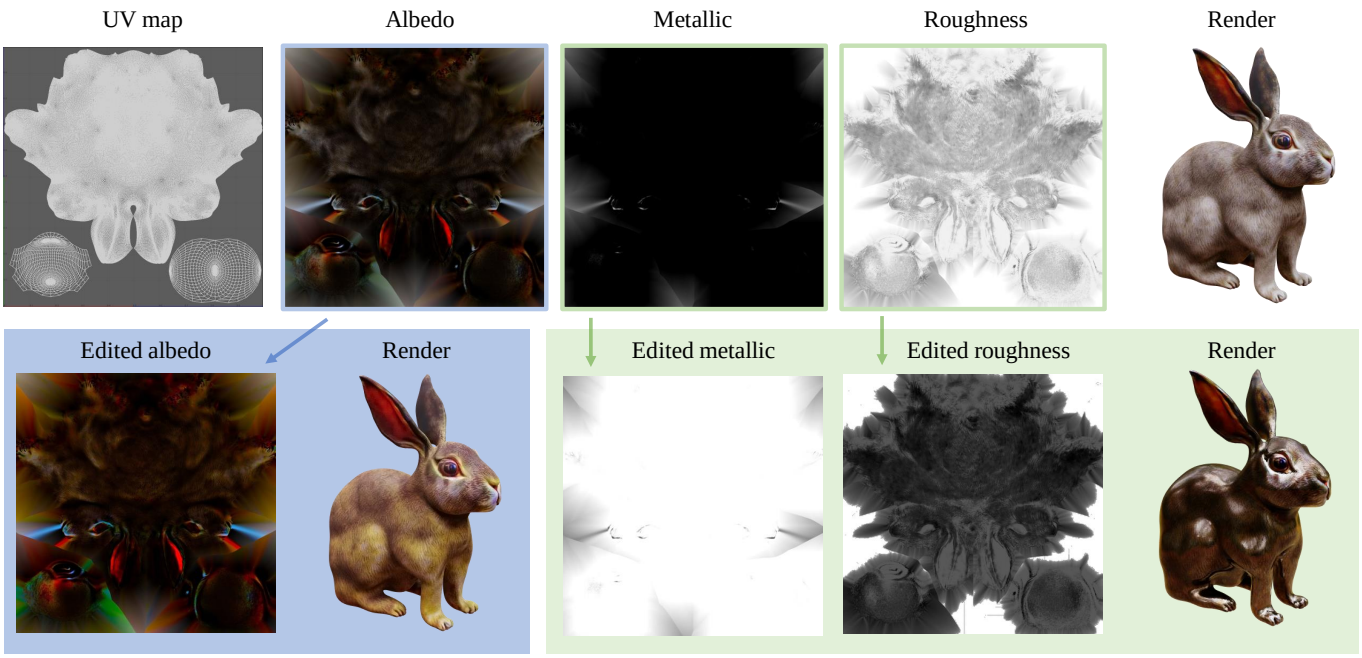

Fig. 13. **Texture maps and material editing.** Top: Generated material maps utilizing DreamMat and the rendering results. Bottom: Edited material maps using 2D image editing techniques and their rendering results.

A APPENDIX

A.1 More Implementation Details

Geometry- and Light-aware diffusion model Training. The geometry- and light-aware diffusion model is trained using six distinct environment light maps, as illustrated in Fig. 20. Within the Objaverse dataset [Deitke et al. 2023], we exclude objects that failed to meet specific criteria, such as those containing transparent parts, non-mesh structures, or too simple geometry (such as only a single plane). Then, we randomly choose one environment light map to render both the light condition maps and the final results. During the rendering of the light condition map, we set albedo to white color, metallic to 0.0 and 1.0, and roughness to 0.0, 0.5, and 1.0.

Our diffusion model uses a ControlNet [Zhang et al. 2023a] with a condition map of 22 channels. The learning rate is set as 1e-5 with a single-step gradient accumulation and a batch size of 256. The model is trained for a total of 3 epochs, which uses 8 V100 GPUs for 3 days. To demonstrate the versatility and quality of the ControlNet, Fig. 21 illustrates the generated results using different text prompts under different lighting conditions.

UV Mapping and Material Editing During the texture map output phase, we employ UV mapping to sample the generated appearance. The model's inherent UV map can be utilized, or alternatively, a UV map can be automatically generated using xatlas [Young 2021]. Following the previous methodologies [Chen et al. 2023a; Munkberg et al. 2022], we apply the UV edge padding technique to extend the boundaries of UV islands and fill in empty regions. The output texture maps are shown in Fig. 13, which can be seamlessly integrated into graphics engines.

Furthermore, these texture maps can be imported into various image editing software (e.g., Photoshop) for material editing. As demonstrated in the bottom row of Fig. 13, we adjust the albedo's saturation, invert the metallic properties, and modify the overall brightness of the roughness. These adjustments enable the achievement of diverse rendering outcomes under the same lighting conditions.

Baseline implementation details. For TANGO [Chen et al. 2022a], we follow the official implementation and set the learning rate to 5e-4, which is decayed by 0.7 in every 500 iterations. We iterate 1500-3000 times for each object until convergence. Since TANGO uses position encoding $\beta(l) = [\cos(2\pi Bl), \sin(2\pi Bl)]^T$ to provide high-frequency details of the generated materials, where B is a random Gaussian matrix whose entry is randomly drawn from $N(0, \sigma^2)$ (different examples in the open source code use different position encoding parameters), we find that the final result is greatly influenced by the PE parameters. Therefore, we carefully tune the σ and frequency number for each example to achieve the best performance.

For TEXTure [Richardson et al. 2023] and Text2Tex [Chen et al. 2023c], we use their original implementations to generate texture results, and then apply NVdiffrec [Munkberg et al. 2022] for subsequent material decomposition. In the material decomposition stage, we fix the geometry and lighting with a learning rate of 0.01 for 3000 iterations.

For Fantasia3D [Chen et al. 2023a], we only use its second stage to generate the objects' materials. In this stage, we fix the environmental lighting and geometry, optimizing the albedo, roughness, and metallic with prompts related to the viewpoint for 3000 iterations. The learning rate is set to 0.01.

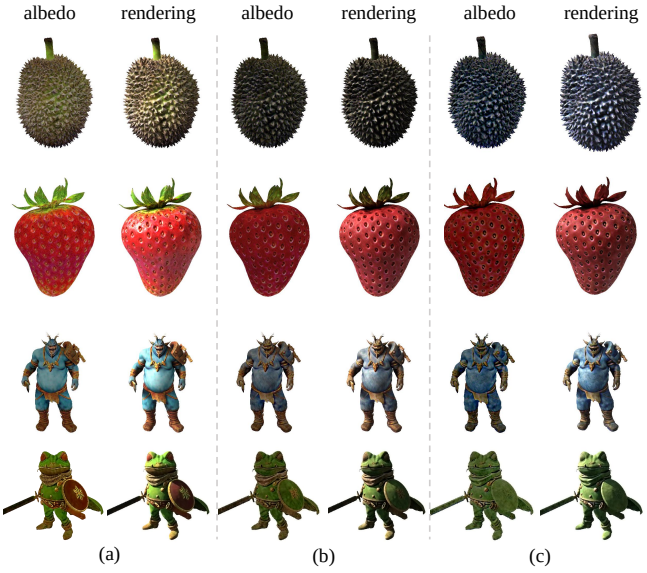

Fig. 14. (a) Our result, incorporating condition maps concurrently to $\epsilon_\phi(I_t; y_{\text{pos}}, t)$, $\epsilon_\phi(I_t; t)$, $\epsilon_\phi(I_t; y_{\text{neg}}, t)$; (b) The result when condition maps are only added to $\epsilon_\phi(I_t; y_{\text{pos}}, t)$, $\epsilon_\phi(I_t; y_{\text{neg}}, t)$; (c) The result with condition maps only added into $\epsilon_\phi(I_t; y_{\text{pos}}, t)$

A.2 Condition Maps to CSD Loss

A proper control strength would be important in the CSD distillation process. As written in Sec 3.2, $\delta(I_t)$ consists of three components: $\epsilon_\phi(I_t; y_{\text{pos}}, t)$, $\epsilon_\phi(I_t; t)$, and $\epsilon_\phi(I_t; y_{\text{neg}}, t)$, representing the scenarios with positive prompts, without prompts, and with negative prompts, respectively. Following the implementation in the CSD [Yu et al. 2023b] source code, we applied the condition map to all three components, resulting in the outcomes shown in Fig.14 (a). Additionally, we experimented with applying the condition map only to $\epsilon_\phi(I_t; y_{\text{pos}}, t)$ and $\epsilon_\phi(I_t; y_{\text{neg}}, t)$ and found that similar results could be achieved, but with some color distortion, as shown in Fig. 14(b). Moreover, applying the condition map solely to $\epsilon_\phi(I_t; y_{\text{pos}}, t)$ suffers from a similar problem of tonal distortion.

A.3 Entanglement of Materials and Lighting

In Fig. 15, we show the results of directly optimizing both lighting and materials using SDS losses, without imposing any constraints on the diffusion process or the materials themselves. For lighting representation, we adopt the model used in NeRO [Liu et al. 2023c], outputting the results as an environment lighting map after sampling. Notably, we observe that the learned environmental lighting contains substantial information about the material properties of the apple, while the apple's albedo retains numerous shading effects, such as highlights and shadows. This phenomenon stems from the ill-posed nature of inverse rendering, a challenge that becomes more pronounced when the lighting in stable diffusion-generated images is arbitrary. Consequently, the utilization of known environmental light maps is crucial to address these issues.

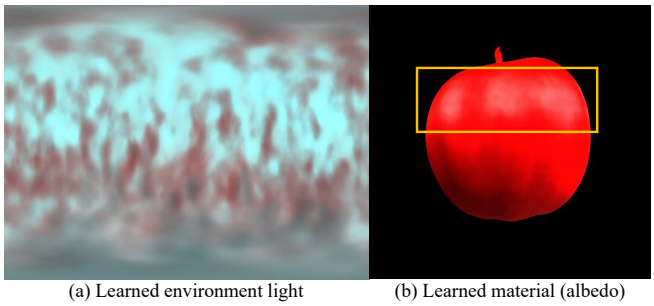

Fig. 15. **Results by directly distilling both materials nad environment light with an SDS [Poole et al. 2022] loss on a stable diffusion [Rombach et al. 2022] model.** (a) the environment lighting map derived from the SDS optimization process, where material attributes of the apple are inadvertently encoded. (b) the albedo of the apple, which exhibits shading effects including highlights and shadows, indicative of the entanglement with the environmental lighting. This visual evidence supports the need for employing light maps to mitigate the ill-posed challenges of inverse rendering.

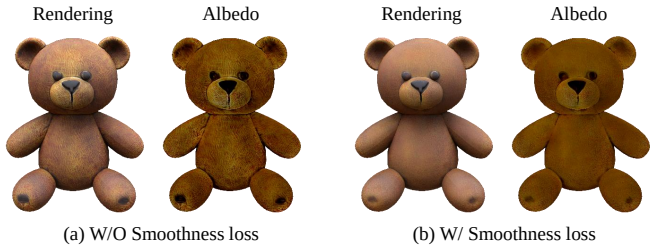

Fig. 16. **Ablation study** on the smoothness loss.

A.4 Different shading models in Objaverse

In our material generation process, we employ the Disney BRDF. However, the Objaverse [Deitke et al. 2023] dataset contains a variety of shaders within its .glb files, which results in a domain gap between renderings of some objects and their corresponding light condition maps during the training of our geometry- and light-aware diffusion model. However, we observe that this domain gap does not severely affect the quality of the generated materials of DreamMat. The reason is that we have excluded objects within Objaverse tagged with descriptors such as 'stylized,' 'style,' 'hand-painted,' 'pixelart,' 'npr,' and 'non-photorealistic', which shows a large deviation from our simplified Disney BRDF. Some remaining objects still do not have the same shading models as ours. However, the effect of lighting on rendering showed consistent patterns as our shader. For example, highlights in renderings are produced by intense lighting. Thus, this still enables us to train our geometry- and light-aware diffusion model.

A.5 Effects of smoothness loss

Following the previous methodologies, we incorporate a material smoothing loss, with the results depicted in Fig. 16. The inclusion of this smoothing term has resulted in a more refined albedo for the teddy bear. Users can adjust this term's coefficient to fine-tune the texture detail's granularity.

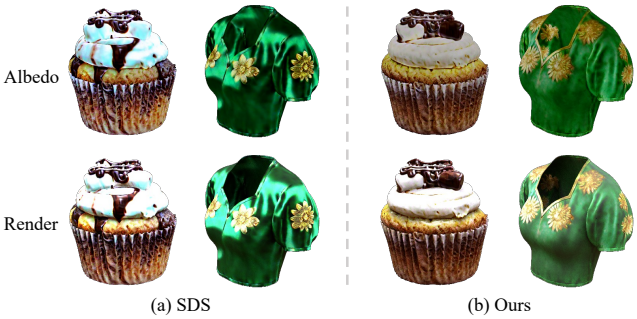

Fig. 17. Results using our geometry- and light-aware diffusion model with SDS loss (a) and CSD loss (b) for the prompts "A cupcake with marshmallow and chocolate drizzle toppin" and "a green silk blouse with golden flower embroidery".

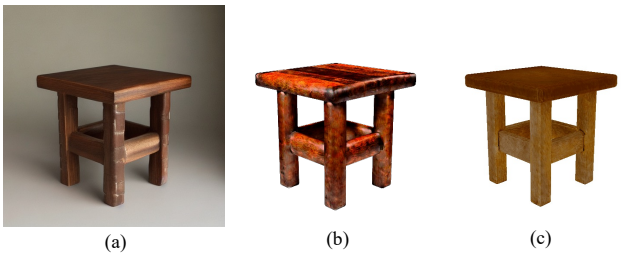

Fig. 18. (a) An albedo image generated directly by Stable Diffusion using text prompt "a wooden stool, diffuse albedo map". (b) An albedo map distilled from a 3D model using CSD loss with the text prompt "a wooden stool, diffuse albedo map". (c) An albedo map obtained by our method.

A.6 Applying SDS loss on our diffusion model

Figure 17 shows the results of using SDS loss with our geometry- and light-aware diffusion model. Notably, the use of SDS loss results in increased saturation and contrast in the generated appearances, which are less visually appealing. In comparison, our method with CSD loss produces better results.

A.7 Using text prompts to generate albedo

Directly using text prompts like "albedo maps" in Stable Diffusion cannot correctly generate albedo maps, as illustrated in Fig. 18. We incorporate the text prompts "diffuse albedo maps" to depth-conditioned Stable Diffusion to generate an image as shown in Fig. 18 (a) and also use this text prompt to distill an albedo map as shown in Fig. 18 (b). It is observable that despite the explicit inclusion of "diffuse albedo" in the text prompts, the albedo results still contain shading effects. This can be attributed to the inadequate text guidance and a lack of albedo training data within Stable Diffusion.

A.8 Different material representations

In Figure 19, we present the albedo maps achieved through different material representations including Multi-layer Perceptron (MLP) network with positional encoding (PE) [Mildenhall et al. 2020] and hash-grid representation [Müller et al. 2022]. During the distillation process, when employing PE-equipped MLPs, a higher frequency count tends to produce axis-aligned artifacts. Conversely, a lower frequency count fails to adequately capture the fine details of the texture. However, the adoption of a hash-grid-based representation effectively mitigates the aforementioned issues, additionally demonstrating a more rapid convergence rate.

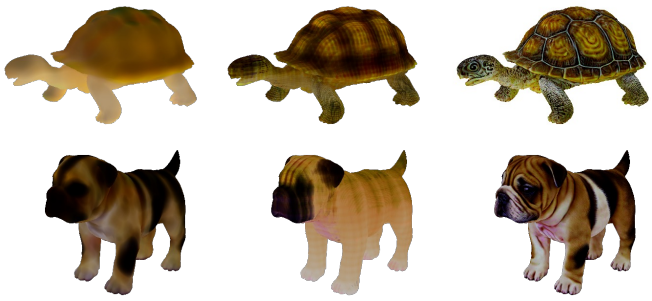

Fig. 19. The generated albedo maps utilizing positional encoding with different numbers of frequencies (a, b), and hash-grid encoding (c).

A.9 Addition Results

In Fig. 22, we further demonstrate several examples of materials generated based on corresponding text prompts, along with their rendered results under various lighting conditions. We can further perform material editing by adjusting the metallic and roughness values and re-rendering the edited objects.

A.10 3D Model Attribution

In this paper, we use 3D models sourced from the Objaverse dataset [Deitke et al. 2023] and Sketchfab [Sketchfab [n. d.]] under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. The models are utilized without their original textures to focus solely on the impact of our material generation method.

Each model used from Sketchfab is attributed as follows:

- "Bobcat machine" by mohamed ouartassi.
- "Molino De Viento _ Windmill" by BC-X.
- "MedivalHouse | house for living | MedivalVilage" by JFredchill.
- "Houseleek plant" by matousekfoto.
- Jagernaut (Beyond Human) by skartemka.
- "Grabfigur" by noe-3d.at.
- "Teenage Mutant Ninja Turtles - Raphael" by Hellbruch.
- "Cat with jet pack" by Muru.
- "Transformers Universe: Autobot Showdown" by Primus03.
- "PigMan" by Grigorii Ischenko.
- "Bulky Knight" by Arthur Krut.
- "Sir Frog" by Adrian Carter.
- "Infantry Helmet" by Masonsmith2020.
- "Sailing Ship Model" by Andrea Spognetta (Spogna).
- "Venice Mask" by DailyArt.
- "Bouddha Statue Photoscanned" by amcgi.
- "Bunny" by vivienne0716.

Fig. 20. Environment light maps used in geometry- and light-aware diffusion model training and material distillation.

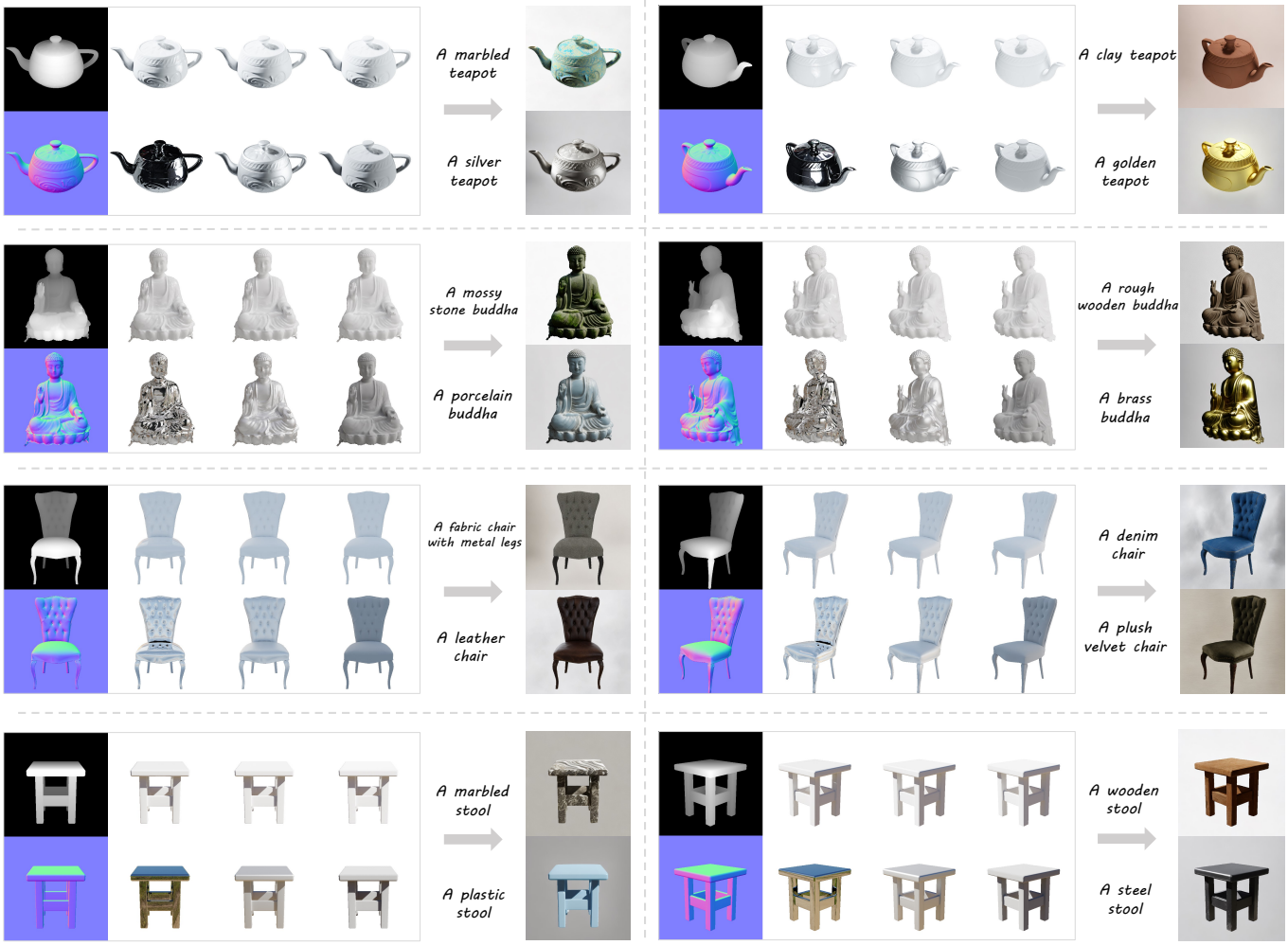

Fig. 21. Image generated by the geometry- and light-aware diffusion model with different text prompts.

Fig. 22. Generated materials and their renderings and editing results.