FlashTex: Fast Relightable Mesh Texturing with LightControlNet

Kangle Deng[2⋆], Timothy Omernick[1], Alexander Weiss[1], Deva Ramanan[2], Jun-Yan Zhu[2], Tinghui Zhou[1], and Maneesh Agrawala[1,3]

[1] Roblox
[2] Carnegie Mellon University
[3] Stanford University

Abstract. Manually creating textures for 3D meshes is time-consuming, even for expert visual content creators. We propose a fast approach for automatically texturing an input 3D mesh based on a user-provided text prompt. Importantly, our approach disentangles lighting from surface material/reflectance in the resulting texture so that the mesh can be properly relit and rendered in any lighting environment. We introduce LightControlNet, a new text-to-image model based on the ControlNet architecture, which allows the specification of the desired lighting as a conditioning image to the model. Our text-to-texture pipeline then constructs the texture in two stages. The first stage produces a sparse set of visually consistent reference views of the mesh using LightControlNet. The second stage applies a texture optimization based on Score Distillation Sampling (SDS) that works with LightControlNet to increase the texture quality while disentangling surface material from lighting. Our algorithm is significantly faster than previous text-to-texture methods, while producing high-quality and relightable textures.

1 Introduction

Creating high-quality textures for 3D meshes is crucial across industries such as gaming, film, animation, AR/VR, and industrial design. Traditional mesh texturing tools are labor-intensive, and require extensive training in visual design. As the demand for immersive 3D content continues to surge, there is a pressing need to streamline and automate the mesh texturing process (Figure 1).

In the past year, significant progress in text-to-image diffusion models [41, 43, 44] has created a paradigm shift in how artists create images. These models allow anyone who can describe an image in a text prompt to generate a corresponding picture. More recently, researchers have proposed techniques for leveraging such 2D diffusion models for automatically generating textures for an input 3D mesh based on a user-specified text prompt [7, 8, 28, 42]. But these methods suffer from three significant limitations that restrict their wide-spread adoption in commercial applications: (1) slow generation speed (taking tens of

⋆ Work done when interning at Roblox.

minutes per texture), (2) potential visual artifacts (e.g., seams, blurriness, lack of details), and (3) baked-in lighting causing visual inconsistency in new lighting environments (Figure 2). While some recent methods address one or two of these issues, none adequately address all three.

Fig. 1: We propose an efficient approach for texturing an input 3D mesh given a user-provided text prompt. Our generated texture can be relit properly in different lighting environments. The light probe shows the varied lighting environment. We suggest the readers check our video results of rotating lighting in our supplementary material.

In this work, we propose an efficient approach for texturing an input 3D mesh based on a user-provided text prompt that disentangles the lighting from surface material/reflectance to enable relighting (Figure 1). Our method introduces **LightControlNet**, an illumination-aware text-to-image diffusion model based on the ControlNet [61] architecture, which allows specification of the desired lighting as a conditioning image for the diffusion model. Our text-to-texture pipeline uses LightControlNet to generate relightable textures in two stages. In stage 1, we use **multi-view visual prompting** in combination with the LightControlNet to produce visually consistent reference views of the 3D mesh for a small set of viewpoints. In stage 2, we perform a new **texture optimization** procedure that uses the reference views from stage 1 as guidance, and extends Score Distillation Sampling (SDS) [38] to work with LightControlNet. This allows us to increase the texture quality while disentangling the lighting from surface material/reflectance. We show that the guidance from the reference views allows our optimization to generate textures with over 10x speed-up than previous SDS-based relightable texture generation methods such as Fantasia3D [8].

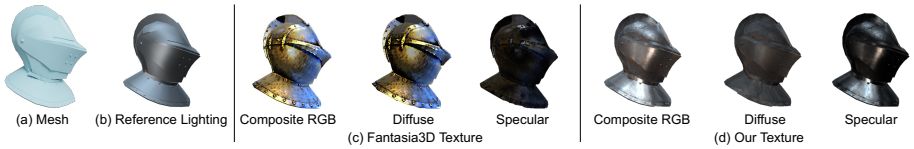

Fig. 2: Given a 3D mesh of a helmet (a) and a lighting environment L, the reference rendering (b) depicts the "correct" highlights on the mesh due to L, by treating its surface reflectance as half-metal and half-smooth with a gray diffuse color. (c) The texture generated by the leading method Fantasia3D [8] is not properly relit as Fantasia3D bakes most of the lighting into the diffuse texture for the mesh and does not capture the bright highlights in the specular texture. (d) In contrast, our pipeline disentangles lighting from material, better capturing the diffuse and specular components of the metal helmet in this environment. Text prompt: "A medieval steel helmet."

Furthermore, our experiments show that the quality of our textures is generally better than those of existing baselines in terms of FID, KID, and user study.

2 Related Work

Text-to-Image generation. Recent years have seen significant advancements in text-to-image generation empowered by diffusion models [41, 43, 44]. Stable Diffusion [43], for example, trains a latent diffusion model (LDM) on the latent space rather than pixel space, delivering highly impressive results with affordable computational costs. Further extending the scope of text-based diffusion models, works such as GLIGEN [22], PITI [54], T2IAdapter [30], and ControlNet [61] incorporate spatial conditioning inputs (e.g., depth maps, normal maps, edge maps, etc.) to enable localized control over the composition of the result. Beyond their power in image generation, these 2D diffusion models, trained on large-scale text-image paired datasets, also contribute valuable priors to various other tasks such as image editing [14,27], 3D generation [38,40], and 3D editing [12,18,52,64].

Text-to-3D synthesis. The success of text-to-image synthesis has sparked considerable interest in its 3D counterpart. Some approaches [20, 33, 47, 63] train a text-conditioned 3D generative model akin to 2D models, while others employ 2D priors from pre-trained diffusion models for optimization [8,21,24,28,38,49,53,55] and multi-view synthesis [26, 46]. For instance, DreamFusion [38] and Score Jacobian Chaining [53] were the first to propose Score Distillation Sampling to optimize a 3D representation using 2D diffusion model gradients. Zero-1-to-3 [26] synthesizes novel views using a pose-conditioned 2D diffusion model. Yet, these methods often produce blurry, low-frequency textures that bake lighting into surface reflectance. Fantasia3D [8] can generate more realistic textures by incorporating physics-based materials. However, the resulting materials remain entangled with lighting, making it difficult to relight the textured object in a new lighting environment. In contrast, our method effectively disentangles the lighting and surface reflectance texture. Concurrent to our work, MATLABER [58]

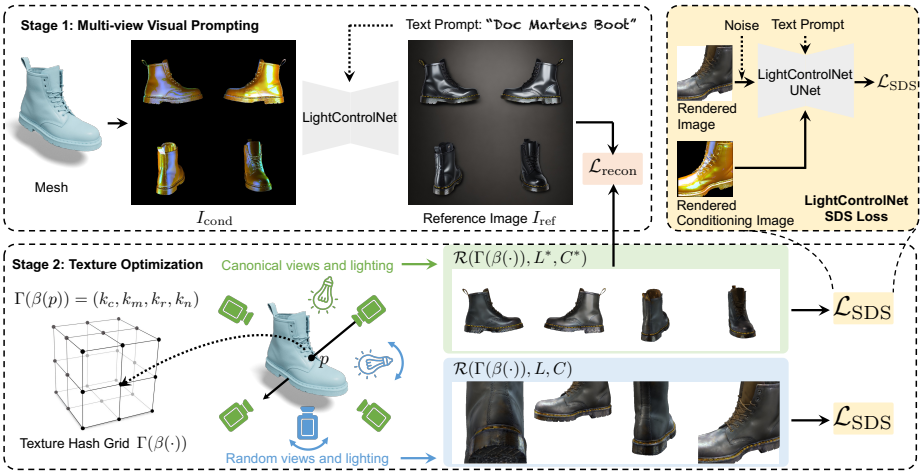

Fig. 3: Text-to-Texture pipeline. Our method efficiently synthesizes relightable textures given a 3D mesh and text prompt. In stage 1 (top left), we use ***multi-view visual prompting*** with our LightControlNet to generate four visually consistent canonical views of the mesh under fixed lighting, concatenated into a reference image I_{ref}. In stage 2, we apply a new ***texture optimization*** procedure using I_{ref} as guidance along with a multi-resolution hash-grid representation of the texture $\Gamma(\beta(\cdot))$. For each iteration, we render two batches of images using $\Gamma(\beta(\cdot))$ – one using the canonical views and lighting of I_{ref} to compute a reconstruction loss $\mathcal{L}_{\text{recon}}$ and the other using randomly sampled views and lighting to compute an SDS loss $\mathcal{L}_{\text{SDS}}$ based on LightControlNet.

aims to recover material information in text-to-3D generation using a material autoencoder. Our method, however, differs in approach and improves efficiency.

3D texture generation. The area of 3D texture generation has evolved over time. Earlier models either directly took 3D representations as input to neural networks [4, 48, 59] or used them as templates [35, 37]. While some methods also use differentiable rendering to learn from 2D images [4, 13, 37, 59], the models often fail to generalize beyond the limited training categories. Closest to our work are the recent works that use pre-trained 2D diffusion models and treat texture generation as a byproduct of text-to-3D generation. Examples include Latent-Paint [28], which uses Score Distillation Sampling in latent space, Text2tex [7], which leverages depth-based 2D ControlNet, and TEXTure [42], which exploits both previous methods. Nonetheless, similar to recent text-to-3D models, such methods produce textures with entangled lighting effects and suffer from slow generation. On the other hand, TANGO [9], generates material textures using a Spherical-Gaussian-based differentiable renderer, but struggles with complex texture generation. A concurrent work, Paint3D [60], aims to generate lighting-less textures, yet it cannot produce material-based textures like ours.

Material generation. Bidirectional Reflection Distribution Function (BRDF) [34] is widely used for modeling surface materials in computer vision and graphics. Techniques for recovering material information from images often leverage

neural networks to resolve the inherent ambiguities when applied to a limited range of view angles or unknown illuminations. However, these methods often require controlled setups [23] or curated datasets [2, 11, 56], and struggle with in-the-wild images. Meanwhile, material generation models like ControlMat [50], Matfuse [51], and Matfusion [45] use diffusion models for generating Spatially-Varying BRDF (SVBRDF) maps but limit themselves to 2D generation. In contrast, our method creates relightable materials for 3D meshes.

3 Preliminaries

Our text-to-texture pipeline builds on several techniques that have been recently introduced for text-to-image diffusion models. Here, we briefly describe these prior methods and then present our pipeline in Section 4.

ControlNet. ControlNet [61] is an architecture designed to add spatially localized compositional controls to a text-to-image diffusion model, such as Stable Diffusion [43], in the form of conditioning imagery (e.g., Canny edges [5], OpenPose keypoints [6], depth images, etc.). In our work, where we take a 3D mesh as input, the conditioning image $I_{\text{cond}}(C)$ is a rendering of the mesh from a given camera viewpoint C. Then, given text prompt y,

$$I_{\text{out}} = \text{ControlNet}(I_{\text{cond}}(C), y),$$

where the output image I_{out} is conditioned on y and I_{cond}. ControlNet introduces a parameter s that sets the strength of the conditioning image. When $s = 0$, the ControlNet simply produces an image using the underlying Stable Diffusion model, and when $s = 1$, the conditioning is strongly applied.

Score Distillation Sampling (SDS). DreamFusion [38] optimizes a 3D NeRF representation θ [1,29] conditioned on text prompts using a pre-trained 2D text-to-image diffusion model. A differentiable renderer $\mathcal{R}$ applied to θ with a randomly sampled camera view C then generates a 2D image $x = \mathcal{R}(\theta, C)$. A small amount of noise $\epsilon \sim \mathbb{N}(0, 1)$ is then added to x to obtain a noisy image x_t. DreamFusion leverages a diffusion model ϕ (Imagen [44]) to provide a score function $\hat{\epsilon}_\phi(x_t; y, t)$, which predicts the sampled noise ϵ given the noisy image x_t, text prompt y, and noise level t. This score function can update the scene parameters θ, using the gradient calculated by SDS:

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

where $w(t)$ is a weighting function. During each iteration, to calculate the SDS loss, we randomly choose a camera view C, render the NeRF θ to form an image x, add noise ϵ to it, and predict the noise using the diffusion model ϕ. DreamFusion optimzes for 5,000 to 10,000 iterations. In our work, we introduce an illumination-aware SDS loss to optimize surface texture on a mesh to suppress inconsistency artifacts and simultaneously separate lighting from the material.

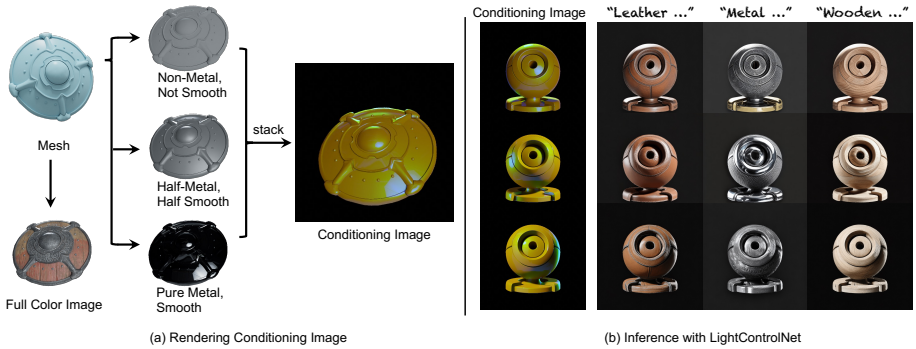

Fig. 4: (a) LightControlNet requires a conditioning image that specifies desired lighting L for a view C of a 3D mesh. To form the conditioning image, we render the mesh with the desired L and C using three different materials: (1) non-metal, not smooth, (2) half-metal, half-smooth, and (3) pure metal, smooth, and then combine the renderings into a single three-channel image. (b) LightControlNet is a diffusion model conditioned on such light-conditioning images and text prompts.

4 Method

Our text-to-texture pipeline operates in two main stages to generate a relightable texture for an input 3D mesh with a text prompt (Figure 3). In Stage 1, we use **multi-view visual prompting** to obtain visually consistent views of the object from a small set of viewpoints, using a 2D ControlNet. Simply backprojecting these sparse views onto the 3D mesh could produce patches of high-quality texture but would also generate visible seams and other visual artifacts where the views do not fully match. The resulting texture would also have lighting baked-in, making it difficult to relight the textured mesh in a new lighting environment. Therefore, in Stage 2, we apply a **texture optimization** that uses a ControlNet in combination with Score Distillation Sampling (SDS) [38] to mitigate such artifacts and separate lighting from the surface material properties. In both stages, we introduce a new illumination-aware ControlNet that allows us to specify the desired lighting as a conditioning image for an underlying text-to-image diffusion model. We call this model **LightControlNet** and describe how it works in Section 4.1. We then detail each stage in Section 4.2 and Section 4.3, respectively.

4.1 LightControlNet

LightControlNet adapts the ControlNet architecture to enable control over the lighting in the generated image. Specifically, we create a conditioning image for a 3D mesh by rendering it using three pre-defined materials and under known lighting conditions (Figure 4). These renderings encapsulate information about the desired shape and lighting for the object, and we stack them into a three-channel conditioning image. We have found that setting the pre-defined materials to (1) non-metal, not smooth; (2) half-metal, half-smooth; and (3) pure metal, extremely smooth, respectively, works well in practice.

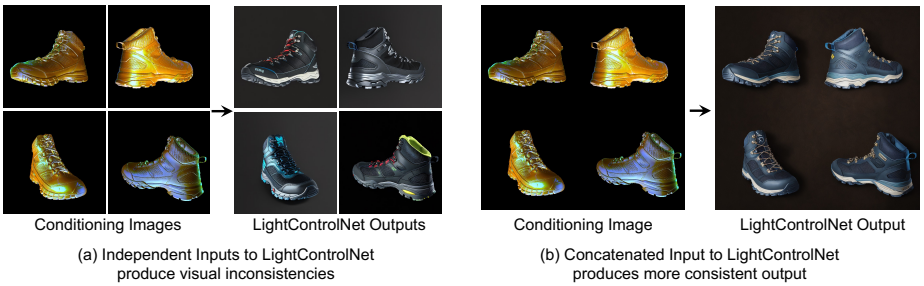

Fig. 5: Multi-view visual prompting. (a) When we independently input four canonical conditioning images to LightControlNet, it generates four very different appearances and styles even with a fixed random seed. (b) When we concatenate the four images into a 2×2 grid and pass them as a single image into LightControlNet, it produces a far more consistent appearance and style. Text prompt: "A hiking boot".

To train our LightControlNet, we use 40K objects from the Objaverse dataset [10]. Each object is rendered from 12 views using a randomly sampled camera C and lighting L sampled from 6 environment maps sourced from the Internet. L is also subject to random rotation and intensity scaling. For each resulting (L, C) pair, we render the conditioning image using the pre-defined materials, as well as the full-color rendering of the object using its original materials and textures. We use the resulting 480K pairs of (conditioning images, full-color rendering) to train LightControlNet using the approach of Zhang et al. [61].

Once LightControlNet is trained, we can specify the desired view and lighting for any 3D mesh. We first render the conditioning image with the desired view and lighting and then pass it along with a text prompt into LightControlNet, to obtain high-quality images. These images are spatially aligned to the desired view, lit with the desired lighting, and contain detailed textures (Figure 4).

Distilling the encoder. We improve the efficiency of SDS by distilling the image encoder in Stable Diffusion (SD) [43], the base diffusion model in the ControlNet architecture. The original SD encoder consumes almost 50% of the forward and backward time of SDS calculation, primarily in downsampling the input image. Metzer et al. [28] have found the image decoder can be closely approximated by per-pixel matrix multiplication. Inspired by this, we distill the encoder by removing its attention modules and training it on the COCO dataset [25] to match the original output. This distilled encoder runs 5x faster than the original one, resulting in an approximately 2x acceleration of our text-to-texture pipeline without compromising output quality (Table 3).

4.2 Stage 1: Multi-view Visual Prompting

In Stage 1, we leverage LightControlNet to synthesize high-quality 2D images for a sparse set of views of the 3D mesh. Specifically, we create conditioning images for four canonical views C^* around the equator of the 3D mesh using a

fixed lighting environment map L^* sampled from a set of environment maps. One approach to generating the complete texture for the mesh would be to apply the LightControlNet independently with each such conditioning image, but using the same text prompt, and then backprojecting the four output images to the surface of the 3D mesh. In practice, however, applying the LightControlNet to each view independently produces inconsistent images of varying appearance and style, even when the text prompt and random seed remain fixed (Figure 5).

We use a multi-view visual prompting approach to mitigate this multi-view inconsistency issue. We concatenate the conditioning images for the four canonical views into a single 2×2 grid and treat it as a single conditioning image. We observe that applying LightControlNet to all four views simultaneously, using this combined multi-view conditioning image, results in a far more consistent appearance and style across the views, compared to independent prompting (Figure 5). We suspect this property arises from the presence of similar training data samples – grid-organized sets depicting the same object – in Stable Diffusion's training set, which is also observed in concurrent works [57, 62]. Formally, we generate the conditioning image $I_{\text{cond}}(L^*, C^*)$ under a fixed canonical lighting condition L^* using four canonical viewpoints C^*. We then apply our LightControlNet with text prompt y to generate the corresponding reference image I_{ref}:

$$I_{\text{ref}} = \text{ControlNet}(I_{\text{cond}}(L^*, C^*), y).$$

4.3 Stage 2: Texture Optimization

In Stage 2, we could directly backproject the four reference views output in Stage 1 onto the 3D mesh using the camera matrix C associated with each view. While the resulting texture would contain some high-quality regions, it would also suffer from two problems: (1) It would contain seams and visual artifacts due to remaining inconsistencies between overlapping views, occlusions in the views that leave parts of the mesh untextured, and loss of detail when applying the backprojection transformation and resampling the views. (2) as lighting is baked into the LightControlNet's RGB images, it would also be baked into the backprojected texture, making it difficult to relight the mesh.

To address both issues, we employ texture optimization using SDS loss. Specifically, we use a multi-resolution hash-grid [31] as our 3D texture representation. Given a 3D point $p \in \mathbb{R}^3$ on the mesh, our hash-grid produces a 32-dim multi-resolution feature, which is then fed to a 2-layer MLP Γ to obtain the texture material parameters for this point. Similar to Fantasia3D [8], these material parameters consist of metallicness $k_m \in \mathbb{R}$, roughness $k_r \in \mathbb{R}$, a bump vector $k_n \in \mathbb{R}^3$ and the base color $k_c \in \mathbb{R}^3$. Formally,

$$(k_c, k_m, k_r, k_n) = \Gamma(\beta(p)),$$

where β is the multi-resolution hash encoding function. Notably, this 3D hash-grid representation can be easily converted to 2D UV texture maps, which

are more friendly to downstream applications. Given the mesh M, the texture $\Gamma(\beta(\cdot))$, a camera view C and lighting L we can use nvdiffrast [19], a differentiable renderer $\mathcal{R}$ to produce a 2D rendering of it, x, as

$$x = \mathcal{R}(M, \Gamma(\beta(\cdot)), L, C).$$

More details about the rendering equation are in the arXiv version. Since the mesh geometry is fixed, we omit M in the remainder of the paper.

Recall that the optimization approach of DreamFusion [38] randomly samples camera views C, generates an image for C using diffusion model ϕ, and supervises the optimization using the SDS loss. We extend this optimization in two ways. First, we use four fixed reference images I_{ref} with their canonical views C^* and lighting L^* to guide the texture optimization through a reconstruction loss:

$$\mathcal{L}_{\text{recon}} = ||I_{\text{ref}} - \mathcal{R}(\Gamma(\beta(\cdot)), L^*, C^*)||_2 + \mathcal{L}_{\text{perceptual}}(I_{\text{ref}}, \mathcal{R}(\Gamma(\beta(\cdot)), L^*, C^*)),$$

where both L2 loss and perceptual loss [17] are used. For a non-canonical view C, we sample a random lighting L and use the SDS loss to supervise the optimization, but with our LightControlNet as the diffusion model ϕ_{LCN}, so

$$\nabla_{\Gamma,\beta}\mathcal{L}_{\text{SDS}}(\phi_{\text{LCN}}, x) = \mathbb{E}_{t,\epsilon}\left[w(t)(\hat{\epsilon}_{\phi_{\text{LCN}}}(x_t; y, t, I_{\text{cond}}(L, C)) - \epsilon)\frac{\partial x}{\partial \Gamma(\beta(\cdot))}\right],$$

where $x = \mathcal{R}(\Gamma(\beta(\cdot)), L, C)$ and $w(t)$ is the weight.

Finally, we employ a material smoothness regularizer on every iteration to enforce smooth base colors, using the approach of nvdiffrec [32]. For a surface point p with base color $k_c(p)$, the smoothness regularizer is defined as

$$\mathcal{L}_{\text{reg}} = \sum_{p \in S} |k_c(p) - k_c(p + \epsilon)|,$$

where S denotes the object surface and ϵ is a small random 3D perturbation. We use $\lambda_{\text{recon}} = 1000$ and $\lambda_{\text{reg}} = 10$ to reweight the loss $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{reg}}$.

Scheduling the optimization. We warm up the optimization by rendering the four canonical views and applying $\mathcal{L}_{\text{recon}}$ for 50 iterations. We then add in iterations using $\mathcal{L}_{\text{SDS}}$ and optimize over randomly chosen camera views and randomly selected lighting from a pre-defined set of environmental lighting maps. Specifically we alternate iterations between using $\mathcal{L}_{\text{SDS}}$ and $\mathcal{L}_{\text{recon}}$. In addition, for a quarter of the SDS iterations, we use the canonical views rather than randomly selecting the views. This ensures that the resulting texture does not overfit to the reference images corresponding to the canonical views. The warm-up iterations capture the large-scale structure of our texture and allow us to use relatively small noise levels ($t \leq 0.1$) in the SDS optimization. We sample the noise following a linearly decreasing schedule [16] with $t_{\text{max}} = 0.1$ and $t_{\text{min}} = 0.02$. We also adjust the conditioning strength s of our LightControlNet in $\mathcal{L}_{\text{SDS}}$ linearly from 1 to 0 over these iterations so that LightControlNet is only lightly applied by the

end of the optimization. We also experimented with a recent variant Variational Score Distillation [55], but did not observe notable improvement. We have experimentally found that we obtain high-quality textures after 400 total iterations of this optimization and this is far fewer iterations than other SDS-based texture generation techniques such as Fantasia3D [8] which requires 5000 iterations.

Faster pipeline without relightability. Our two-stage pipeline is also compatible with off-the-shelf depth ControlNet and Stable Diffusion [43] as the backbone replacement of LightControlNet. Specifically, we can replace the LightControlNet in Stage 1 with a depth ControlNet that uses a depth rendering of the mesh as the conditioning image, and uses Stable Diffusion based SDS in Stage 2. In scenarios where texture relightability is not required, this variant offers an additional 2× speed-up (as shown in Table 1), since it eliminates the additional computation required by LightControlNet forward pass in the SDS optimization.

5 Experiments

In this section, we present comprehensive experiments to evaluate the efficacy of our proposed method for relightable, text-based mesh texturing. We perform both qualitative and quantitative comparisons with existing baselines, along with an ablation study on the significance of each of our major components.

Dataset. As illustrated in Figure 3, we employ Objaverse [10] to render paired data to train our LightControlNet. Objaverse consists of approximately 800k objects, of which we use the names and tags as their text descriptions. We filter out objects with low CLIP similarity [39] to their text descriptions and select around 40k as our training set. To evaluate baselines and our method, we hold out 70 random meshes from Objaverse [10] as the test set. We additionally gather 22 mesh assets from 3D online games with 5 prompts each to assess our method.

Baselines. We compare our approach with existing mesh texturing methods. Specifically, Latent-Paint [28] employs SDS loss in latent space for texture generation. Text2tex [7] progressively produces 2D views from chosen viewpoints, followed by an inverse projection to lift them to 3D. TEXTure [42] utilizes a similar lifting approach but supplements it with a swift SDS optimization post-lifting. Beyond these texture generation methods, text-to-3D approaches serve as additional baselines, given that texture is a component of 3D generation. Notably, we choose Fantasia3D [8] as a baseline, the first to use a material-based representation for textures in text-to-3D processing.

Quantative evaluation. In Table 1, we compare our method with the baselines on the Objaverse [10] test set. For each method, we generate 16 views and evaluate Frechet Inception Distance (FID) [15,36] and Kernel Inception Distance (KID) [3] compared with ground-truth renderings. Two variations of our method are assessed. Both variants use ourr two-stage pipeline, and the first employs a standard depth ControlNet, while the second uses our proposed LightControlNet. Our method outperforms the baselines in both quality and runtime.

Qualitative analysis. In Figure 6, our method can generate highly detailed textures that can be rendered properly with the environment lighting across various

Fig. 6: Sample results from our method applied to Objaverse test meshes (top half) and 3D game assets (bottom half). To illustrate the efficacy of our relightable textures, for each textured mesh, we fix the environment lighting and render the mesh under different rotations. As shown above, our method is able to generate textures that are not only highly detailed, but also relightable with realistic lighting effects.

meshes. We also visually compare our method and the baselines in Figure 7. Our method produces textures with higher visual fidelity for both the relightable and non-relightable variants. In particular, when compared with Fantasia3D [8], a recent work that also aims to generate material-based texture, our results not only have superior visual quality, but also disentangle the lighting more successfully.

User study. To further evaluate the texture quality quantitatively, we conduct a user study comparing our results with each of the baselines on the Objaverse test set in Table 2. We asked 30 participants to evaluate (1) the realism of the results, (2) the consistency of the generated texture with the input text, and (3) the plausibility of the results when placed under varying lighting conditions. Each result is presented in the form of 360-degree rotation to display full texture details. The reference lighting is provided alongside when participants evaluate (3). Across all three aspects, participants consistently prefer our method.

(a) Close-up Comparison with Fantasia3D.
Left Prompt: "A medieval steel helmet" ; Right Prompt: "A leather horse saddle".

(b) Comparison with relightable and non-relightable baselines.
Top Prompt: "A hiking boot"; Bottom Prompt: "A leather horse saddle".

Fig. 7: Qualitative analysis. (a) We compare with Fantasia3D [8] that also attempts to generate Physically Based Rendering (PBR) texture. However, their results often exhibit baked-in lighting, leading to artifacts when put under varied lighting. (b) We also compare with other baselines that only generate non-relightable (RGB) texture. For non-relightable texture generation, we can replace our LightControlNet with depth ControlNet and generate textures with a shorter runtime. More details are in Table 1.

Ablation study. We perform an ablation analysis on different aspects of our method in Table 3. When replacing our distilled encoder with the original SD encoder, performance is twice as slow without noticeably superior quality. On the other hand, without the multi-view visual prompting for the initial generation, the system requires 2000 iterations (a 5x slowdown compared to our 400 itera-

Table 1: Quantitative Evaluation. We test our methods and baselines on 70 test objects from Objaverse [10] and 22 objects curated from 3D game assets. With depth ControlNet, our method yields superior results to all baselines while being three times as fast as the fastest baseline. Using LightControlNet (Ours) within our model improves the lighting disentanglement while maintaining comparable image quality.

	Objaverse test set		Game Asset		Runtime ↓
	FID ↓	KID ↓ ($\times 10^{-3}$)	FID ↓	KID ↓ ($\times 10^{-3}$)	(mins)
Latent-Paint [28]	73.65	7.26	204.43	9.25	10
Fantasia3D [8]	120.32	8.34	164.32	9.34	30
TEXTure [42]	71.64	5.43	103.49	5.64	6
Text2tex [7]	95.59	4.71	119.98	5.21	15
Ours (w/ depth)	**60.49**	3.96	85.92	3.87	**2**
Ours	62.67	**2.69**	**83.32**	**3.34**	4

Table 2: User study. We conduct a user preference study to evaluate (1) result realism, (2) texture consistency with input text, and (3) plausibility under varied lighting. Participants consistently prefer our results over all baselines in these respects.

Preferred Percentage	Objaverse test set		
	Realistic	Consistent with text	Relightable
Ours v.s. Latent-Paint [28]	92.6%	74.5%	84.3%
Ours v.s. Fantasia3D [8]	81.9%	67.6%	74.3%
Ours v.s. TEXTure [42]	70.8%	57.3%	87.1%
Ours v.s. Text2tex [7]	75.4%	61.6%	88.6%

Table 3: Ablation study on algorithmic components. We analyze the role of our distilled encoder (1st row) and multi-view visual prompting (2nd row). Replacing the distilled encoder with the original encoder doubles the running time without a noticeable improvement. When removing the multi-view visual prompting for initial generation, the system requires 2,000 iterations (5x compared to our 400 iterations) to produce reasonable results, which produces slightly worse texture quality.

Objaverse test set	FID ↓	KID ($\times 10^{-3}$) ↓	Runtime ↓ (mins)
Ours (w/o dist. enc.)	**60.34**	2.84	8
Ours (w/o m.v.v.p)	74.23	3.54	19
Ours	62.67	**2.69**	**4**

tions) to produce reasonable results while still leading to slightly worse texture quality. In Section 4.1, we render a conditioning image using three pre-defined materials to encompass a broad range of feasible effects. Table 4 shows omitting any one of these bases degrades quality. Table 5 evaluates our selection of four

Table 4: Ablation study on material bases. We verify the impact of the material bases in rendering conditioning images. Omitting any one of these degrades quality.

Material Basis			FID ↓	KID ($\times 10^{-3}$) ↓
non-metal, not smooth	half-metal, half-smooth	pure metal, smooth		
✓	✓	✓	**62.67**	**2.69**
	✓	✓	66.34	3.11
✓		✓	64.32	3.42
✓	✓		67.43	4.12
	✓		72.13	4.53

Table 5: Ablation study on canonical view selection in Section 4.2. Using only front and back views provides insufficient supervision while adding top and bottom views worsens quality. This likely stems from pre-trained 2D diffusion models struggling with top and bottom views. Additionally, stacking more views reduces each view's resolution, leading to poorer initialization for Stage 2.

Num. of canonical views	FID ↓	KID ($\times 10^{-3}$) ↓
2 views (front, back)	67.43	3.47
4 views (**Ours:** front, back, left, right)	**62.67**	**2.69**
6 views (front, back, left, right, top, bottom)	70.14	3.72

canonical views in Section 4.2. Relying on only the front and back views provides insufficient supervision. Interestingly, incorporating top and bottom views degrades the performance. We hypothesize that this is likely due to the limitation of 2D diffusion model backbones in reliably generating top and bottom views. Furthermore, stacking more views within a single image results in a decreased resolution for each view, given the fixed resolution of the multi-view image.

6 Discussion

We proposed an automated texturing technique based on user-provided prompts. Our method employs an illumination-aware 2D diffusion model (LightControlNet) and an improved optimization process based on the SDS loss. Our approach is substantially faster than previous methods while yielding high-fidelity textures with illumination disentangled from surface reflectance/albedo. We demonstrated the efficacy of our method through quantitative and qualitative evaluation on the Objaverse dataset and meshes curated from game assets.

Limitations. Our approach still poses a few limitations: (1) Baked-in lighting can still be found in certain cases, especially for meshes that are outside of the training data distribution; (2) The generated material maps are sometimes not fully disentangled and interpretable as metallicness, roughness, etc.

Acknowledgements

We thank Benjamin Akrish, Victor Zordan, Dmitry Trifonov, Derek Liu, Sheng-Yu Wang, Gaurav Parmer, Ruihan Gao, Nupur Kumari, and Sean Liu for their discussion and help. This work was done when KD was an intern at Roblox. The project is partly supported by Roblox. JYZ is partly supported by the Packard Fellowship. The Microsoft Research PhD Fellowship supports KD.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: IEEE International Conference on Computer Vision (ICCV) (2021)
2. Bi, S., Xu, Z., Srinivasan, P., Mildenhall, B., Sunkavalli, K., Hasan, M., Hold-Geoffroy, Y., Kriegman, D., Ramamoorthi, R.: Neural reflectance fields for appearance acquisition. arXiv preprint arXiv:2008.03824 (2020)
3. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. In: International Conference on Learning Representations (ICLR) (2018)
4. Bokhovkin, A., Tulsiani, S., Dai, A.: Mesh2tex: Generating mesh textures from image queries. In: IEEE International Conference on Computer Vision (ICCV) (2023)
5. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence pp. 679–698 (1986)
6. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7291–7299 (2017)
7. Chen, D.Z., Siddiqui, Y., Lee, H.Y., Tulyakov, S., Nießner, M.: Text2tex: Text-driven texture synthesis via diffusion models. In: IEEE International Conference on Computer Vision (ICCV) (2023)
8. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In: IEEE International Conference on Computer Vision (ICCV) (2023)
9. Chen, Y., Chen, R., Lei, J., Zhang, Y., Jia, K.: Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
10. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
11. Gao, D., Li, X., Dong, Y., Peers, P., Xu, K., Tong, X.: Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. In: ACM SIGGRAPH (2019)
12. Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: IEEE International Conference on Computer Vision (ICCV) (2023)
13. Henderson, P., Tsiminaki, V., Lampert, C.: Leveraging 2D data to learn textured 3D mesh generation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

14. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
16. Huang, Y., Wang, J., Shi, Y., Qi, X., Zha, Z.J., Zhang, L.: Dreamtime: An improved optimization strategy for text-to-3d content creation. arXiv preprint arXiv:2306.12422 (2023)
17. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 694–711. Springer (2016)
18. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. Advances in Neural Information Processing Systems **35**, 23311–23330 (2022)
19. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. In: ACM SIGGRAPH (2020)
20. Li, M., Duan, Y., Zhou, J., Lu, J.: Diffusion-sdf: Text-to-shape via voxelized diffusion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
21. Li, W., Chen, R., Chen, X., Tan, P.: Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. arxiv:2310.02596 (2023)
22. Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., Lee, Y.J.: Gligen: Open-set grounded text-to-image generation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
23. Li, Z., Sunkavalli, K., Chandraker, M.: Materials for masses: Svbrdf acquisition with a single mobile phone image. In: European Conference on Computer Vision (ECCV) (2018)
24. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
25. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context (2015)
26. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: IEEE International Conference on Computer Vision (ICCV) (2023)
27. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (ICLR) (2022)
28. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
29. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision (ECCV) (2020)
30. Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)

31. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. In: ACM SIGGRAPH (2022)
32. Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., Müller, T., Fidler, S.: Extracting Triangular 3D Models, Materials, and Lighting From Images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
33. Nam, G., Khlifi, M., Rodriguez, A., Tono, A., Zhou, L., Guerrero, P.: 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. arXiv preprint arXiv:2212.00842 (2022)
34. Nicodemus, F.E.: Directional reflectance and emissivity of an opaque surface. Applied optics **4**(7), 767–775 (1965)
35. Park, K., Rematas, K., Farhadi, A., Seitz, S.M.: Photoshape: Photorealistic materials for large-scale shape collections. In: ACM SIGGRAPH Asia (2018)
36. Parmar, G., Zhang, R., Zhu, J.Y.: On aliased resizing and surprising subtleties in gan evaluation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
37. Pavllo, D., Kohler, J., Hofmann, T., Lucchi, A.: Learning generative models of textured 3d meshes from real-world images. In: IEEE International Conference on Computer Vision (ICCV) (2021)
38. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: International Conference on Learning Representations (ICLR) (2023)
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML) (2021)
40. Raj, A., Kaza, S., Poole, B., Niemeyer, M., Ruiz, N., Mildenhall, B., Zada, S., Aberman, K., Rubinstein, M., Barron, J., et al.: Dreambooth3d: Subject-driven text-to-3d generation. arXiv preprint arXiv:2303.13508 (2023)
41. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
42. Richardson, E., Metzer, G., Alaluf, Y., Giryes, R., Cohen-Or, D.: Texture: Text-guided texturing of 3d shapes. In: ACM SIGGRAPH (2023)
43. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
44. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
45. Sartor, S., Peers, P.: Matfusion: a generative diffusion model for svbrdf capture. In: ACM SIGGRAPH Asia (2023)
46. Shi, Y., Wang, P., Ye, J., Mai, L., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv:2308.16512 (2023)
47. Shue, J.R., Chan, E.R., Po, R., Ankner, Z., Wu, J., Wetzstein, G.: 3d neural field generation using triplane diffusion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
48. Siddiqui, Y., Thies, J., Ma, F., Shan, Q., Nießner, M., Dai, A.: Texturify: Generating textures on 3d shape surfaces. In: European Conference on Computer Vision (ECCV) (2022)
49. Sun, J., Zhang, B., Shao, R., Wang, L., Liu, W., Xie, Z., Liu, Y.: Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior (2023)

50. Vecchio, G., Martin, R., Roullier, A., Kaiser, A., Rouffet, R., Deschaintre, V., Boubekeur, T.: Controlmat: Controlled generative approach to material capture. arXiv preprint arXiv:2309.01700 (2023)
51. Vecchio, G., Sortino, R., Palazzo, S., Spampinato, C.: Matfuse: Controllable material generation with diffusion models. arXiv preprint arXiv:2308.11408 (2023)
52. Wang, C., Chai, M., He, M., Chen, D., Liao, J.: Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3835–3844 (2022)
53. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
54. Wang, T., Zhang, T., Zhang, B., Ouyang, H., Chen, D., Chen, Q., Wen, F.: Pretraining is all you need for image-to-image translation. arXiv preprint arXiv:2205.12952 (2022)
55. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In: Advances in Neural Information Processing Systems (NeurIPS) (2023)
56. Wang, Z., Philion, J., Fidler, S., Kautz, J.: Learning indoor inverse rendering with 3d spatially-varying lighting. In: IEEE International Conference on Computer Vision (ICCV) (2021)
57. Weber, E., Holynski, A., Jampani, V., Saxena, S., Snavely, N., Kar, A., Kanazawa, A.: Nerfiller: Completing scenes via generative 3d inpainting. In: arXiv (2023)
58. Xu, X., Lyu, Z., Pan, X., Dai, B.: Matlaber: Material-aware text-to-3d via latent brdf auto-encoder. arXiv preprint arXiv:2308.09278 (2023)
59. Yu, R., Dong, Y., Peers, P., Tong, X.: Learning texture generators for 3d shape collections from internet photo sets. In: British Machine Vision Conference (BMVC) (2021)
60. Zeng, X., Chen, X., Qi, Z., Liu, W., Zhao, Z., Wang, Z., FU, B., Liu, Y., Yu, G.: Paint3d: Paint anything 3d with lighting-less texture diffusion models (2023)
61. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: IEEE International Conference on Computer Vision (ICCV) (2023)
62. Zhao, M., Zhao, C., Liang, X., Li, L., Zhao, Z., Hu, Z., Fan, C., Yu, X.: Efficientdreamer: High-fidelity and robust 3d creation via orthogonal-view diffusion prior. In: arXiv (2023)
63. Zhou, L., Du, Y., Wu, J.: 3d shape generation and completion through point-voxel diffusion. In: IEEE International Conference on Computer Vision (ICCV) (2021)
64. Zhuang, J., Wang, C., Lin, L., Liu, L., Li, G.: Dreameditor: Text-driven 3d scene editing with neural fields. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–10 (2023)