

# Paint3D: Paint Anything 3D with Lighting-Less Texture Diffusion Models

Xianfang Zeng<sup>1\*</sup>

Xin Chen<sup>1\*</sup>

Zhongqi Qi<sup>1\*</sup>

Wen Liu<sup>1</sup>

Zibo Zhao<sup>1,3</sup>

Zhibin Wang<sup>1</sup>

BIN FU<sup>1</sup>

Yong Liu<sup>2</sup>

Gang Yu<sup>1†</sup>

<sup>1</sup>Tencent PCG

<sup>2</sup>Zhejiang University

<sup>3</sup> ShanghaiTech University

<https://github.com/OpenTexture/Paint3D>



Figure 1. A gallery of generated texture results by Paint3D. Our method is capable of generating lighting-less, high-quality, and high-fidelity textures across diverse objects from numerous categories.

## Abstract

This paper presents Paint3D, a novel coarse-to-fine generative framework that is capable of producing high-resolution, lighting-less, and diverse 2K UV texture maps for untextured 3D meshes conditioned on text or image inputs. The key challenge addressed is generating high-quality textures without embedded illumination information, which allows the textures to be re-lighted or re-edited within modern graphics pipelines. To achieve this, our method first leverages a pre-trained depth-aware 2D diffusion model to generate view-conditional images and perform multi-view texture fusion, producing an initial coarse texture map. However, as 2D models cannot fully represent 3D shapes and disable lighting effects, the coarse texture map exhibits incomplete areas and illumination artifacts.

facts. To resolve this, we train separate UV Inpainting and UVHD diffusion models specialized for the shape-aware refinement of incomplete areas and the removal of illumination artifacts. Through this coarse-to-fine process, Paint3D can produce high-quality 2K UV textures that maintain semantic consistency while being lighting-less, significantly advancing the state-of-the-art in texturing 3D objects.

## 1. Introduction

The rise of deep generative models has ushered the era of Artificial Intelligence Generated Content, catalyzing advancements in natural language generation [47, 59, 72], image synthesis [43, 49, 51, 52], and 3D generation [32, 44, 62]. These 3D generative technologies have significantly impacted various applications, revolutionizing the landscape of current 3D productions. However, the generated meshes, characterized by chaotic lighting textures and

\*These authors contributed equally to this work.

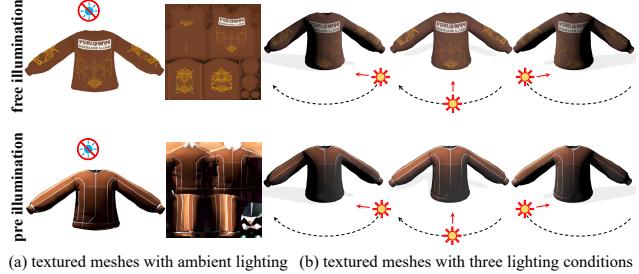
<sup>†</sup>Corresponding author (email: skicyyu@tencent.com).

complex wiring, are often incompatible with traditional rendering pipelines, such as physically based rendering (PBR). The lighting-less texture diffusion model, capable of generating diverse appearances of 3D assets, should augment these pre-existing 3D productions for the gaming industry, film industry, virtual reality, and so on.

Recent advancements in texture synthesis have shown significant progress, particularly in the utilization of 2D diffusion models such as TEXTure [50] and Text2tex [5]. These models effectively employ pre-trained depth-to-image diffusion models to generate high-quality textures through text conditions. However, these methods have issues with pre-illuminated textures. This can damage the quality of final renderings in 3D environments and cause lighting errors when changing lighting within common graphics workflows, as shown in the bottom of Fig. 2. Conversely, texture generation methods trained from 3D data offer an alternative approach such as PointUV [69] and Mesh2tex [2], which typically generate textures by comprehending the entire geometries for specific 3D objects. However, they are often hindered by a lack of generalization, struggling to apply their models to a broad range of 3D objects beyond their training datasets, as well as generate various textures through different textual or visual prompts.

Two challenges are crucial for texture generation. The first is achieving broad generalization across various objects using diverse prompts or image guidance, and the second is eliminating the coupled illumination on the generated results obtained from pre-training. Recent advancement of conditioned image synthesis works [51, 70] using billion-level images, capable of “rendering” diverse image results from 3D views, can help overcome the size limitation of 3D data in texture generation. However, the pre-illuminated textures can interfere with the final visual outcomes of these textured objects within rendering engines. Furthermore, since the pre-trained image diffusion models only provide 2D results in the view domain, they struggle to maintain view consistency for 3D objects due to the lack of comprehensive understanding of their shapes. Therefore, our focus is on developing a two-stage texture diffusion model for 3D objects. This model should be able to generalize to various pre-trained image generative models and learn lighting-less texture generation while preserving view consistency.

In this work, we propose a coarse-to-fine texture generation framework, namely Paint3D, that leverages the strong image generation and prompt guidance abilities of pre-trained image generative models for texturing 3D objects. To enable the generalization of rich and high-quality texture results from diverse prompts, we first progressively sample multi-view images from a pre-trained view depth-aware 2D image diffusion model and then back-project these images onto the surface of the 3D mesh to generate an initial texture map. In the second stage, Paint3D focuses on



(a) textured meshes with ambient lighting (b) textured meshes with three lighting conditions

Figure 2. Illustration of the pre-illumination problem. The texture map with free illumination is compatible with traditional rendering pipelines, while there are inappropriate shadows when relighting is applied on the pre-illumination texture.

generating lighting-less textures. To achieve this, we contribute separate UV Inpainting and UVHD diffusion models specialized in the shape-aware refinement of incomplete regions and removal of lighting influences. We train these diffusion models on UV texture space, using feasible 3D objects and their high-quality illumination-free textures as supervision. Through this coarse-to-fine process, Paint3D can generate semantically consistent high-quality 2K textures devoid of intrinsic illumination effects. Extensive experiments demonstrate that Paint3D achieves state-of-the-art performance in texturing 3D objects with different texts or images as conditional inputs and offers compelling advantages for graphics editing and synthesis tasks.

We summarize our contributions as follows: 1) We propose a novel coarse-to-fine generative framework that is capable of producing high-resolution, lighting-less, and diverse 2K UV texture maps for untextured 3D meshes; 2) We separately design a shape-aware UV Inpainting diffusion model and a shape-aware UVHD diffusion model as the refinement of incomplete regions and removal of lighting influences; 3) Our proposed Paint3D supports both textual and visual prompts as conditional inputs and achieves state-of-the-art performance on texturing 3D objects. The code will be released later.

## 2. Related Work

Traditional methods [19, 25, 26, 61, 63, 64, 73] of synthesizing texture to 3D assets concentrated on placing simple exemplar patterns on a surface or leveraging global optimization for painting the 3D shape. However, the recent learning-based approaches [8, 21, 29, 41, 45, 48, 57, 65, 74] have succeeded in generating plausible textures for more complex 3D shapes. The following discusses the related learning-based methods.

**Iteratively Texturing via 2D Diffusion Models.** The rapidly expanding large-scale 2D text-to-image (T2I) diffusion models [49, 51, 52] have yielded remarkable outcomes, and subsequently, [28, 32, 33, 53, 58] harness the capabilities of T2I models to facilitate texture synthesis on

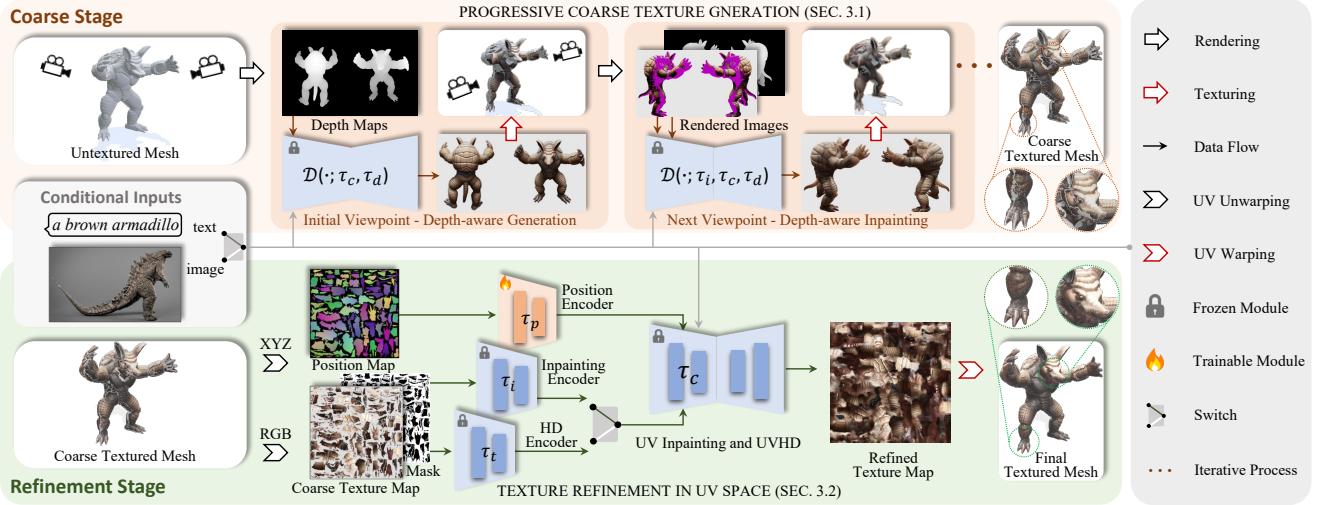


Figure 3. The overview of our coarse-to-fine framework. The coarse stage (Sec. 3.1) samples multi-view images from the pre-trained 2D image diffusion models, then back-project these images onto the mesh surface to create initial texture maps. The refinement stage (Sec. 3.2) generates high-quality textures with a diffusion model in UV space, conditioned on the position map and the coarse texture map.

3D assets. TEXTure [50] devises an iteratively texturing scheme and succeeds in synthesizing high-quality texture. It leverages a pretrained depth-to-image diffusion model and gradually paints the texture map of a 3D model from multiple viewpoints. Although TEXTure [50] samples a partial texture map under each viewpoint conditioned on previous results, the generative process still lacks global information modeling, leading to the view-inconsistency results. Later, TexFusion [3] proposes to aggregate texture information from different viewpoints during the denoising process and synthesize the entire texture map, which improves the view consistency. Besides, Text2tex [5] developed an automatic method to select viewpoints for saving human efforts. These methods improve the global texture modeling but still suffer from the inherited lighting bias from 2D Priors, leading to inconsistent results. In contrast, our framework involves a texture refinement model trained with illumination-free texture data, significantly alleviating the illumination artifacts.

**Optimization-based 3D Generation via 2D diffusion model.** Prior to the emergence of large-scale text-to-image models, early optimization-based texturing approaches [18, 27, 35, 37, 38] endeavored to utilize the large-scale vision-language model, CLIP [46], for optimizing texture map of 3D models. Subsequently, the introduction of Score Distillation Sampling (SDS) in DreamFusion [44] has paved the way for numerous text-to-3D approaches [7, 9, 31, 36, 55, 56, 60, 62]. Latent-nerf [36] and Fantasia3D [7] extend SDS for optimizing the texture map with texture-less 3D shapes as input. Those methods consider inputting an initial shape and simultaneously optimize the texture map and geometry. They could produce multi-view consistent texture but cannot guarantee geometry fidelity. Moreover, they

struggle with the Janus problem due to the semantically ambiguous. Different from these methods, our model learns on the whole texture map, preserving the 3D geometry.

**Generative Texturing from 3D Data.** Various learning-based approaches usually train generative texturing models based on the 3D data [12, 13, 20, 30, 34, 39] from scratch. Early methods [6, 15, 16, 40] learn implicit texture fields to assign a color to each pixel on the surface of the 3D shape. However, since the texture on the surface of 3D shapes is continuous, discrete supervision is unlikely to train a model for synthesizing high-quality textures. Texturify [54] defines texture maps on the surface of polygon meshes and devises a convolution operator for mesh structures by incorporating StyleGAN [22–24] architecture for predicting texture on each face. Such methods are limited by the mesh resolution and the lack of global information modeling, although the recent Mesh2tex [2] further integrates an implicit texture field branch for improvements. Moreover, some methods (AUV-net [10], LTG [68], TUVF [11], PointUV [69]) learn to synthesize UV-Maps for 3D shapes, avoiding the abovementioned limitations. Unfortunately, these methods usually struggle when handling more general objects due to the variations between 3D objects in different categories.

### 3. Method

To synthesize high-quality and diverse texture maps for 3D models based on desired conditional inputs like prompts or images, we propose a coarse-to-fine framework, Paint3D, which progressively generates and refines texture maps, as shown in Fig. 3. In the **coarse stage** (see Sec. 3.1), we sample multi-view images from the pre-trained 2D image diffusion models, then back-project these images onto the mesh

surface to create initial texture maps. In **the refinement stage** (see Sec. 3.2), we enhance coarse texture maps by performing a diffusion process in the UV space, achieving lighting-less, inpainting, High Definition (HD) functions to ensure the final texture’s completeness and visual appeal.

Given an uncolored 3D model  $M$  and an appearance condition  $c$ , such as text prompts [5, 50] or an appearance reference image [2], our Paint3D aims to generate the texture map  $T$  for the 3D model. Here, we represent the 3D model’s geometry using a surfaced mesh, denoted as  $M = (V, F)$ , with vertices  $V = \{v_i\}, v_i \in \mathbb{R}^3$  and triangular faces  $F = \{f_i\}$ , where each  $f_i$  is a triplet of vertices. The texture map is represented by a multi-channel image in UV space, denoted as  $T \in \mathbb{R}^{H \times W \times C}$ . The proposed Paint3D framework  $\mathcal{P}$  consists of two stages: the coarse texture generation stage  $\mathcal{C} : (M, c) \mapsto \hat{T}$  and the texture refinement stage  $\mathcal{F} : \hat{T} \mapsto T$ , that is  $T = \mathcal{P}(M, c) = \mathcal{F}(\mathcal{C}(M, c))$ . Furthermore, we define a conditional diffusion model as  $\mathcal{D}(\cdot; \tau_\theta)$ , where  $\tau_\theta$  is a domain-specific encoder and can be substituted for varying conditions.

### 3.1. Progressive Coarse Texture Generation

In this state, we generate a coarse UV texture map for untextured 3D meshes based on a pre-trained view depth-aware 2D diffusion model. Specifically, we first render the depth map from different camera views, then sample images from the image diffusion model with depth conditions, and finally back-project these images onto the mesh surface. To improve the consistency of textured meshes in each view, we alternately perform the three processes of rendering, sampling, and back-projection, progressively generating the entire texture map [5, 50].

**Initial Viewpoint.** With the set of camera views  $\{p_i\}_{i=1}^n$  focusing on the 3D mesh, we start to generate the texture of the visible region. We first render the 3D mesh to a depth map  $d_1$  from the first view  $p_1$ , where this rendering process is denoted as  $\mathcal{R} : (M, p_1) \mapsto d_1$ . We then sample a texture image  $I_1$  given an appearance condition  $c$  and a depth condition  $d_1$ , denoted as

$$I_1 = \mathcal{D}(z, c, d_1; \tau_c, \tau_d), \quad (1)$$

where  $z \in \mathbb{R}^{h \times w \times e}$  is a random initialized latent,  $\tau_c$  is appearance encoder, and  $\tau_d$  is depth encoder. Subsequently, we back-project this image onto the 3D mesh from the first view, generating the initial texture map  $\hat{T}_1$ , where this back-projecting process is denoted as  $\mathcal{R}^{-1} : (M, I_1, p_1) \mapsto \hat{T}_1$ .

**Next Non-initial Viewpoint.** For these viewpoints  $p_k$ , we execute a similar process as mentioned above but the texture sampling process is performed in an image inpainting manner. Specifically, taking into account the textured region from all previous viewpoints  $\hat{T}_{\{1, k-1\}}$ , the rendering process outputs not only a depth image  $d_k$  but

also a partially colored RGB image  $\hat{I}_k$  and an uncolored area mask  $m_k$  in the current view, denoted as  $\mathcal{R} : (M, p_k, \hat{T}_{\{1, k-1\}}) \mapsto (d_k, \hat{I}_k, m_k)$ . We use a depth-aware image inpainting model, with a new inpainting encoder  $\tau_i$ , to fill the uncolored area within the rendered RGB image, denoted as

$$I_k = \mathcal{D}(\hat{I}_k, m_k, c, d_k; \tau_i, \tau_c, \tau_d). \quad (2)$$

The inpainted image is back-projected onto the 3D mesh under the current view, generating the current texture map  $\hat{T}_k$  from the view  $p_k$ , denoted as  $\mathcal{R}^{-1} : (M, I_k, p_k) \mapsto \hat{T}_k$ . The textured region from previous viewpoints  $\hat{T}_{\{1, k-1\}}$  is kept and the uncolored area is updated by the current texture map  $\hat{T}_k$ , formatted as

$$\hat{T}_{\{1, k\}} = m_{k-1}^{UV} \odot \hat{T}_{\{1, k-1\}} + (1 - m_{k-1}^{UV}) \odot \hat{T}_k, \quad (3)$$

where  $m_{k-1}^{UV}$  is the colored area mask in the UV plane and can be calculated from the texture map  $\hat{T}_{\{1, k-1\}}$ . Therefore, the texture map is progressively generated view-by-view and arrives at the entire coarse texture map  $\hat{T} = \hat{T}_{\{1, n\}}$ .

**Multi-view Texture Sampling.** We extend the texture sampling process mentioned above (Eq. (1) and Eq. (2)) to the multi-view scene. Specifically, in the initial texture sampling, we utilize a pair of cameras to capture two depth maps  $\{d_1, d_2\}$  from symmetric viewpoints. We then concatenate those two depth maps horizontally (in width) and compose a depth grid with a size of  $1 \times 2$ , denoted as  $\mathbf{d}_1$ . To perform multi-view depth-aware texture sampling, we replace the single depth image  $d_1$  with the depth grid  $\mathbf{d}_1$  in Eq. (1). Similarly, in the non-initial texturing, we horizontally concatenate renders, composing depth grid  $\mathbf{d}_k$ , RGB image grid  $\hat{\mathbf{I}}_k$ , and mask grid  $\mathbf{m}_k$ . To perform multi-view depth-aware texture inpainting, we replace the inputs in Eq. (2) with those grids. As evaluated in Sec. 4.4, we also explore the effectiveness of the number of viewpoints.

### 3.2. Texture Refinement in UV Space

Although the appearance of the coarse texture map is coherent, it still has some issues like lighting shadows involved by the 2D image diffusion model, or the texture holes caused by self-occlusion during the rendering process. We propose to perform a diffusion process in the UV space based on the coarse texture map, aiming to mitigate these issues and further enhance the visual aesthetics of the texture map during texture refinement. However, refining texture maps in the UV space with mainstream image diffusion models [51, 71] presents the challenge of texture discontinuity [69]. The texture map is derived through UV mapping of the 3D surface texture, which cuts the continuous texture on the 3D mesh into a series of individual texture fragments in the UV plane. This fragmentation complicates the learning of the 3D adjacency relationships among the fragments in the UV plane, leading to texture discontinuity issues.

**Position Encoder.** To refine the texture map in UV space, we perform the diffusion process guided by adjacency information of texture fragments. Here, the 3D adjacency information of texture fragments is represented as the position map in UV space  $O \in \mathbb{R}^{H \times W \times 3}$ , where each non-background element is a 3D point coordinate. Similar to the texture map, the position map can be obtained through UV mapping of the 3D point coordinates. To fuse the 3D adjacency information during the diffusion process, we add an individual position map encoder  $\tau_p$  to the pretrained image diffusion model. Following the design principle of ControlNet [71], the new encoder has the same architecture as the encoder in the image diffusion model and is connected to it through zero-convolution layer.

Our texture diffusion model is trained using a dataset consisting of paired position maps and texture maps  $\{O_i, T_i\}_{i=1}^n$ . Given a set of conditions including time step  $t$ , appearance condition  $c$ , as well as a position map  $O$ , our texture diffusion model learns to predict the noise added to the noisy latent  $z_t$  with

$$\mathcal{L} = \mathbb{E}_{z_0, t, c, O, \epsilon \sim \mathcal{N}(0, 1)} \left[ \|\epsilon - \epsilon_\theta(z_t, t, c, \tau_p(O))\|_2^2 \right]. \quad (4)$$

For an image diffusion model with a trained denoiser  $\epsilon_\theta$ , we freeze  $\epsilon_\theta$  as suggested by [71] and only optimize the position encoder  $\tau_p$  with Eq. (4). Since texture maps in UV space are lighting-less, our model can learn this prior from data distribution, generating lighting-less texture.

**UV Inpainting.** We can simultaneously use the position encoder and other conditional encoders to perform various refinement tasks in UV space. Here we introduce two specific refinement capabilities, namely UV inpainting and UV High Definition (UVHD). The UV inpainting is used to fill texture holes within the UV plane, which can avoid self-occlusion problems during rendering. To achieve UV inpainting, we add the position map encoder  $\tau_p$  on an image inpainting diffusion model as

$$T_{inpainting} = \mathcal{D}(\hat{T}, m^{UV}, c, O; \tau_i, \tau_c, \tau_p), \quad (5)$$

which takes as input a coarse texture map  $\hat{T}$ , texture map mask  $m^{UV}$ , appearance condition  $c$ , and position map  $O$ , and produces as output an inpainted texture map  $T_{inpaint}$ .

**UV High Definition (UVHD)** is designed to enhance the visual aesthetics of the texture map. We use the position encoder  $\tau_p$  and an image enhance encoder  $\tau_t$  with a diffusion model  $\mathcal{D}(\cdot; \tau_c)$  to achieve UVHD, denoted as

$$T_{tiling} = \mathcal{D}(\hat{T}, c, O; \tau_t, \tau_c, \tau_p). \quad (6)$$

In our refinement stage, we perform UV inpainting followed by UVHD to get the final refined texture map  $T$ . By integrating the UV inpainting and UVHD, Paint3D is capable of producing lighting-less (Fig. 7), complete (Fig. 8), high-resolution, and diverse UV texture maps (Fig. 9).

## 4. Experiments

We provide extensive comparisons to evaluate our models on both quality and diversity in the following. Firstly, we introduce the datasets settings, evaluation metrics and implementation details Sec. 4.1. Importantly, we show the comparisons on two texture generation tasks, including text-to-texture (Sec. 4.2), image-to-texture (Sec. 4.3). Lastly, we conduct ablation studies to demonstrate the effectiveness of each module in our Paint3D (Sec. 4.4). More qualitative results, comparisons, and details are provided in supplements.

### 4.1. Implementation Details

We apply the text2image model from Stable Diffusion v1.5 [51] as our texture generation backbone. To handle the image condition, we employ the image encoder introduced in IP-Adapter [66]. For additional conditional controls such as depth, image inpainting, and image high definition, we utilize the domain encoders provided in ControlNet [71]. In the coarse texture generation, we define six axis-aligned principal viewpoints, and sample two texture images from a pair of symmetric viewpoints during a single diffusion progress. The denoising strengths are set as 1 and 0.75 for the coarse and refinement stages, respectively. Our implementation uses the PyTorch [42] framework, with Kaolin [14] used for rendering and texture projection. For the UV unwarping process, we utilize the original UV map if the mesh contains texture coordinates, or we use an open-source UV-Atlas tool [67] to perform UV unwarping.

**Datasets.** We conduct experiments on a subset of textured meshes from the Objaverse [13] dataset. We exclude meshes devoid of textures, those with monochromatic texture, and 3D scene objects composed of multiple meshes. The filtered subset contains 105,301 texture meshes, with 105,000 meshes utilized for training the position encoder and 301 meshes employed for evaluating our model. Additionally, we gather 30 meshes in the wild to assess our model. This brings the total to 331 high-quality textured meshes for evaluation.

**Evaluation metrics.** We access the generated textures with commonly used metrics for image quality and diversity. Specifically, we report the Frechet Inception Distance (FID) [17] and Kernel Inception Distance (KID  $\times 10^{-3}$ ) [1]. To calculate the generated image distribution, we render  $512 \times 512$  images of each mesh with the synthesized textures, captured from 20 fixed viewpoints. The real distribution is made up of renders of the meshes under identical settings, but using their original textures.

### 4.2. Comparisons on Text-to-Texture

We first evaluate the texture generation effect of Paint3D conditioned on the text prompt. We compare our method with state-of-the-art approaches, including Latent-Paint [36], TEXTure [50], and Text2Tex [5]. Latent-Paint

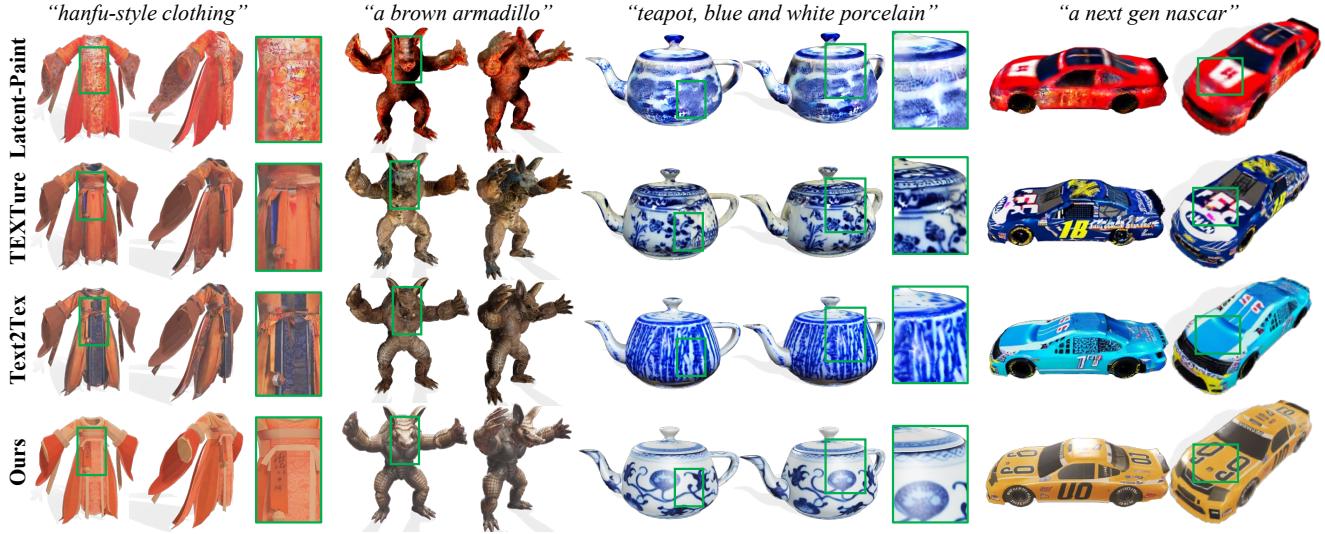


Figure 4. Qualitative comparisons on texture generation conditioned on text prompt. We compare our textured mesh against Latent-Paint [36], TEXTure [50], and Text2Tex [5]. Compared to the baselines, our method generates an illumination-free texture map, as well as more exquisite texture details (*c.f.* supplements for more our results).

Methods	FID↓	KID ↓	User Study	
			Overall Quality↑	Text Fidelity↑
Latent-Paint [36]	62.22	15.81	2.83	3.29
TEXTure [50]	43.13	11.13	3.36	4.12
Text2Tex [5]	38.93	7.94	3.57	4.27
Ours	<b>27.28</b>	<b>4.81</b>	<b>4.45</b>	<b>4.74</b>

Table 1. Quantitative comparisons on text-to-texture task. Ours outperforms other approaches on both FID and KID ( $\times 10^{-3}$ ).

is a texture generation variant of the NeRF-based 3D object generation framework, explicitly manipulating the texture map via the text2image model from Stable Diffusion. TEXTure devises an iterative texture generation scheme to manipulate the texture map, and successfully synthesizes high-quality textures. Following a similar principle, Text2Tex develops an automatic viewpoint selection strategy in the iterative process, representing the current state-of-the-art in the field of text-conditioned texture generation. For the category-specific texture generation approaches [2, 54, 69], we provide more comparisons in the supplements.

**Qualitative comparisons.** As shown in Fig. 4, our approach is able to generate an illumination-free texture map while excelling at synthesizing high-quality texture details. Firstly, Latent-Paint [36] tends to generate blurry textures, which can lead to suboptimal visual effects. Additionally, while TEXTure [50] is capable of generating clear textures, the generated textures may lack smoothness and exhibit noticeable seams or splicing(e.g., the teapot in Fig. 4). Lastly, even though Text2Tex [5] demonstrates the ability to generate smoother textures, it may compromise in generating fine textures with intricate details. Notably, all baselines generate pre-illumination texture maps that led to inappropriate shadows when relighting was applied.

**Quantitative comparisons.** In Tab. 1, we present the quantitative comparisons with the previous SOTA methods in text-driven texture synthesis. Following [5, 69], we report the FID [17] and KID [1] to access the quality and diversity of the generated texture maps. Our method outperforms all baselines by a significant margin (29.93% improvement in FID and 39.42% improvement in KID). These improvements demonstrate the superior capability of our method in generating high-quality textures across diverse objects from numerous categories.

**User study.** We further conduct a user study to analyze the overall quality of the generated textures and their fidelity to the input text prompts. We randomly select 60 meshes and corresponding text prompts to perform the user study. Those meshes are textured by both Paint3d and baseline models, and displayed to users in random sequence. Each object displays full-view texture details in the form of 360-degree rotation. Each respondent is asked to evaluate the results based on two aspects: (1) overall quality and (2) fidelity to the text prompt, using a scale of 1 to 5. We collected the evaluation results of 30 users, as presented in Tab. 1, where we show the average results across all prompts for each method. As can be seen, our approach outperforms all baselines in terms of both overall quality and text fidelity by a significant margin.

### 4.3. Comparisons on Image-to-Texture

We then evaluate the texture generation capability of Paint3D conditioned on the image prompt. Here, we provide TEXTure [50] as our comparison baseline. We use the texture transfer capability of TEXTure to generate its image-to-texture results. To handle the image condi-

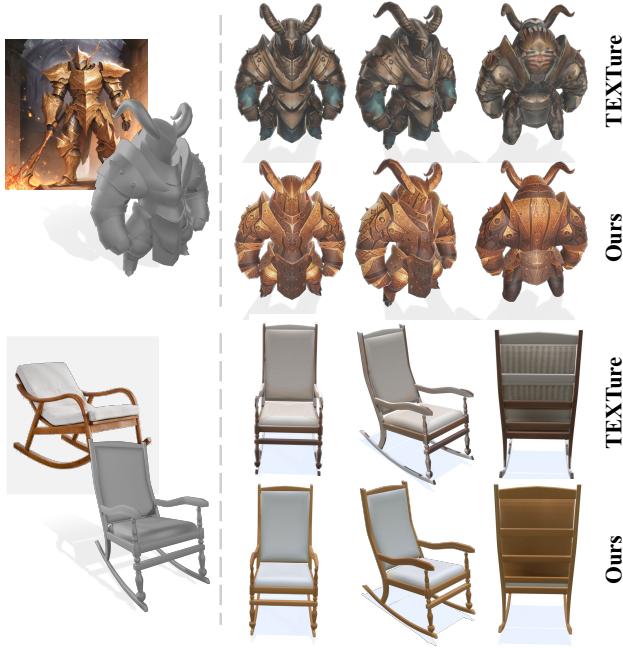


Figure 5. Qualitative comparisons on texture generation conditioned on image prompt. Compared to TEXTure, our method can better represent texture details contained in the image condition.

Methods	FID↓	KID↓	User Study	
			Overall Quality↑	Image Fidelity↑
TEXTure [50]	40.83	9.76	3.56	3.73
Ours	<b>26.86</b>	<b>4.94</b>	<b>4.71</b>	<b>4.89</b>

Table 2. Quantitative comparisons on image-to-texture task. Our method achieves a significant improvement over the baseline.

tion, our Paint3D employs the image encoder introduced in [66] based on the txt2image model from Stable Diffusion v1.5 [51]. As depicted in Fig. 5, our approach excels in synthesizing exquisite texture while maintaining high fidelity with respect to the image condition. TEXTure [50] is capable of generating a similar texture as the input image, but it struggles to accurately represent texture details in the image condition. For instance, in the samurai case, TEXTure generates a golden armor texture but fails to synthesize high-frequency line details present on the armor.

As shown in Tab. 2, we also report the FID [17] and KID [1] scores under the image condition. Our method demonstrates a significant improvement over the baseline, as evidenced by the FID score decreasing from 40.83 to 26.86 and the KID score decreasing from 9.76 to 4.94. For the user study, we follow a similar evaluation setting as described in Sec. 4.2, but replace the text prompt with the image prompt. Each participant needs to assess the generated texture based on its overall quality and fidelity to the image prompt, using a rating scale ranging from 1 to 5. The average scores of all users are reported in Tab. 2. Notably, Paint3D gets a 4.89 average score on image fidelity, indicat-

Coarse Stage	Refinement Stage		FID↓	KID↓
	UV inpainting	UVHD		
✓	✗	✗	41.84	10.91
✗	✓	✓	48.81	11.98
✓	✓	✗	37.84	7.13
✓	✗	✓	33.42	6.19
✓	✓	✓	<b>27.28</b>	<b>4.81</b>

Table 3. Evaluation of modules in the Paint3D framework. This demonstrates the effectiveness of each component, including the coarse stage, UV inpainting, and UVHD. By integrating the generation prior in the coarse stage and the illumination-free prior in the refinement stage, our full model achieves the optimal result.

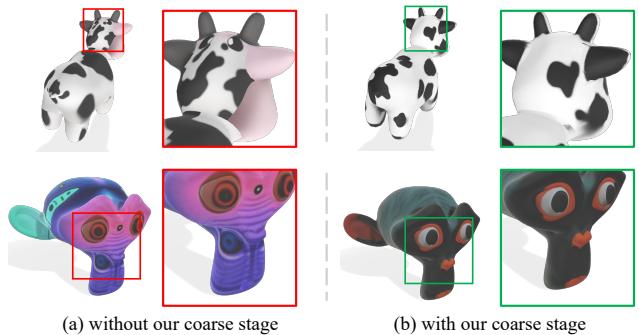


Figure 6. Illustration of the effect of the coarse stage. The absence of our coarse stage may result in semantic confusion in the texture.

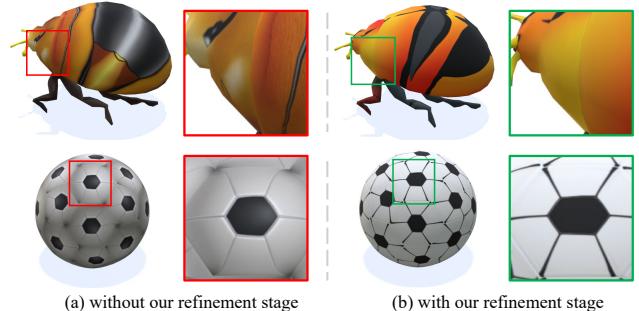


Figure 7. Visualization of the effect of the refinement stage. With our refinement stage, the generated textures are illumination-free.

ing our method is able to accurately represent texture details contained in the image condition.

#### 4.4. Ablation Studies

**Evaluation of Coarse-to-fine Framework.** To demonstrate the effectiveness of our coarse-to-fine texture generation framework, we conduct experiments on two baselines “w/o coarse stage” and “w/o refinement stage”. The “w/o coarse stage” configuration refers to directly generating the texture map using the texture refinement modules in UV space, performing UV inpainting followed by UVHD without initialization from the coarse stage. The “w/o refinement stage” configuration represents the outcome of the coarse stage, where the uncolored area is assigned a color

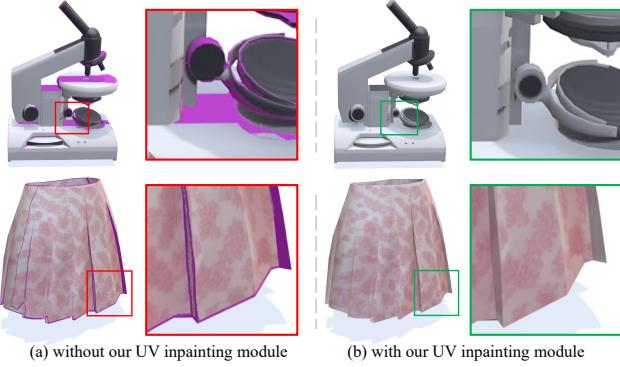


Figure 8. Illustration of the effect of UV inpainting. UV inpainting can effectively fill texture holes that are located in projecting blind spots (e.g. the inner side of a pleated skirt).

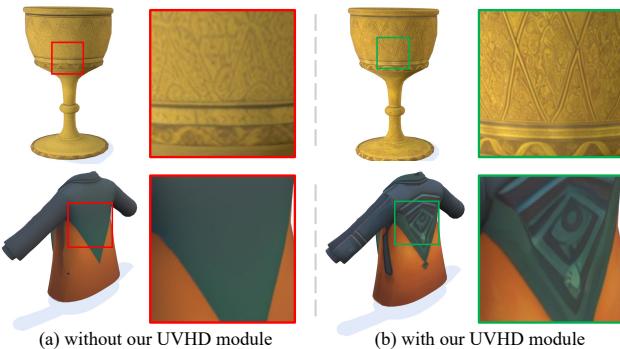


Figure 9. Illustration of the effect of UVHD module. This displays the capability of UVHD to enhance existing texture details and can even generate new textures in monochromatic areas.

using bilinear interpolation. In both scenarios, the model produces inferior results compared to our full model, as reported in Tab. 3. We visualize the results of “w/o coarse stage” in Fig. 6. Absent the coarse stage, the generated textures may display noticeable semantic problems, as the texture map in UV space consists of separate texture fragments. As shown in Fig. 7, without the refinement stage, the generated textures are pre-illuminated.

**Evaluation of UV inpainting and UVHD.** To demonstrate the effectiveness of two texture refinement modules, UV inpainting and UVHD, we further conduct experiments on two baselines “w/o UV inpainting” and “w/o UVHD”. The “w/o UV inpainting” configuration refers to filling the uncolored area with the bilinear interpolation instead of UV inpainting, followed by the UVHD module. The “w/o UVHD” configuration represents the inpainted result of the coarse stage with the UV inpainting module. As indicated in Tab. 3, the performance shows a significant decrease when UV inpainting or UVHD is not utilized, indicating their irreplaceable function during texture refinement processing. We visualize the results of “w/o UV inpainting” in Fig. 8. UV inpainting can effectively fill texture holes that are located in blind spots, as this inpainting process-

#Viewpoint Total	#Viewpoint One Iter	FID↓	KID ↓	#Viewpoint Total	#Viewpoint One Iter	FID↓	KID ↓
2	1	42.31	11.67	2	2	41.74	10.19
4	1	36.07	7.85	4	2	32.60	6.37
6	1	29.02	5.10	6	2	<b>27.28</b>	<b>4.81</b>
8	1	30.15	5.65	8	2	27.71	4.93

Table 4. Evaluation of the number of viewpoints in the coarse stage. The viewpoints are not the more the better, as the pretrained 2D image diffusion model may involve illumination artifacts.

ing is performed within the UV plane, without occlusion problems. As depicted in Figure 9, UVHD demonstrates its capability to enhance existing texture details and even generate new textures on monochromatic areas.

**Evaluation of the Number of Viewpoints.** The selection of viewpoints has shown a significant influence on the texture generation result in the coarse stage [5]. We conduct ablation studies to analyze the impact of the number of viewpoints on both overall coarse texture generation and single diffusion process. As shown in Tab. 4, we can see that increasing the number of viewpoints can improve the quality of generated textures, but it is not that the more the viewpoints the better the results. We achieve the best result when the viewpoint is set to 6. The result is further improved when we sample two texture images from a pair of symmetric viewpoints during a single diffusion process.

## 5. Discussion

This paper presents Paint3D, a novel coarse-to-fine generative framework that is capable of generating high-quality 2K UV textures that maintain semantic consistency while being lighting-less, conditioned on text or image inputs. To achieve this, our method first leverages a pre-trained depth-aware 2D diffusion model to generate view-conditional images and perform multi-view texture fusion, producing an initial coarse texture map. Subsequently, we train distinct UV Inpainting and UVHD diffusion models, specifically designed for shape-aware refinement of incomplete areas and the removal of illumination artifacts. Through this coarse-to-fine process, Paint3D can produce high-quality, lighting-less, and diverse texture maps, significantly advancing the state-of-the-art in texturing 3D objects.

Our method has inherent limitations as follows. Our approach still suffers from the multi-faces problem in the coarse stage which will result in a failure case. This issue primarily arises from the inconsistency of multi-view texture images sampled by the pre-trained 2D diffusion model, as it is not explicitly trained on multi-view datasets. It remains a challenge for Paint3D to generate material maps, which are commonly used in modern physically based rendering pipelines. Furthermore, unlike optimization-based 3D generation methods [7, 31, 36, 62], Paint3D is not capable to generate or edit the geometry of 3D assets.

## References

- [1] Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD gans. In *6th International Conference on Learning Representations, ICLR 2018*. 5, 6, 7
- [2] Alexey Bokhovkin, Shubham Tulsiani, and Angela Dai. Mesh2tex: Generating mesh textures from image queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8918–8928, October 2023. 2, 3, 4, 6
- [3] Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4169–4181, 2023. 3
- [4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 15
- [5] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18558–18568, October 2023. 2, 3, 4, 5, 6, 8
- [6] Qimin Chen, Zhiqin Chen, Hang Zhou, and Hao Zhang. Shaddr: Real-time example-based geometry and texture generation via 3d shape detailization and differentiable rendering. *arXiv preprint arXiv:2306.04889*, 2023. 3
- [7] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. 3, 8
- [8] Yiwen Chen, Chi Zhang, Xiaofeng Yang, Zhongang Cai, Gang Yu, Lei Yang, and Guosheng Lin. It3d: Improved text-to-3d generation with explicit view synthesis. *arXiv preprint arXiv:2308.11473*, 2023. 2
- [9] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585*, 2023. 3
- [10] Zhiqin Chen, Kangxue Yin, and Sanja Fidler. Auv-net: Learning aligned uv maps for texture transfer and synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1465–1474, 2022. 3
- [11] An-Chieh Cheng, Xuetong Li, Sifei Liu, and Xiaolong Wang. Tuvf: Learning generalizable texture uv radiance fields. *arXiv preprint arXiv:2305.03040*, 2023. 3
- [12] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21126–21136, 2022. 3
- [13] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 3, 5
- [14] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A pytorch library for accelerating 3d deep learning research. <https://github.com/NVIDIAGameWorks/kaolin>, 2022. 5
- [15] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022. 3
- [16] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023. 3
- [17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5, 6, 7
- [18] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535*, 2022. 3
- [19] Jingwei Huang, Justus Thies, Angela Dai, Abhijit Kundu, Chiyu Jiang, Leonidas J Guibas, Matthias Nießner, Thomas Funkhouser, et al. Adversarial texture optimization from rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1559–1568, 2020. 2
- [20] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 3
- [21] Animesh Karnewar, Niloy J Mitra, Andrea Vedaldi, and David Novotny. Holofusion: Towards photo-realistic 3d generative modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22976–22985, 2023. 2
- [22] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 3
- [23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
- [24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 3
- [25] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen, Dani Lischinski, and Tien-Tsin Wong. Solid texture synthesis from 2d exemplars. In *ACM SIGGRAPH 2007*

- papers*, pages 2–es. 2007. 2
- [26] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis. *ACM Transactions on Graphics (TOG)*, 25(3):541–548, 2006. 2
- [27] Jiabao Lei, Yabin Zhang, Kui Jia, et al. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. *Advances in Neural Information Processing Systems*, 35:30923–30936, 2022. 3
- [28] Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. Sweet-dreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. *arXiv preprint arXiv:2310.02596*, 2023. 2
- [29] Yuhang Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. *arXiv preprint arXiv:2308.10608*, 2023. 2
- [30] Yuchen Li, Ujjwal Upadhyay, Habib Slim, Ahmed Abdelreheem, Arpit Prajapati, Suhail Pothigara, Peter Wonka, and Mohamed Elhoseiny. 3d compat: Composition of materials on parts of 3d things. In *European Conference on Computer Vision*, pages 110–127. Springer, 2022. 3
- [31] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 3, 8
- [32] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 1, 2
- [33] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 2
- [34] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 3
- [35] Yiwei Ma, Xiaoqing Zhang, Xiaoshuai Sun, Jiayi Ji, Haowei Wang, Guannan Jiang, Weilin Zhuang, and Rongrong Ji. X-mesh: Towards fast and accurate text-driven 3d stylization via dynamic textual guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2749–2760, 2023. 3
- [36] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023. 3, 5, 6, 8
- [37] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 3
- [38] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 conference papers*, pages 1–8, 2022. 3
- [39] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 3
- [40] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4531–4540, 2019. 3
- [41] Zijie Pan, Jiachen Lu, Xiatian Zhu, and Li Zhang. Enhancing high-resolution 3d generation through pixel-wise gradient clipping. *arXiv preprint arXiv:2310.12474*, 2023. 2
- [42] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5
- [43] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1
- [44] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 3
- [45] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 2
- [46] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3
- [47] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 1
- [48] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508*, 2023. 2
- [49] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 1, 2
- [50] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In Erik Brunvand, Alla Sheffer, and Michael Wimmer, editors, *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6–10, 2023*, pages 54:1–54:11. ACM, 2023. 2, 3, 4, 5, 6, 7
- [51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of*

- the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 4, 5, 7
- [52] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 2
- [53] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2
- [54] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces. In *European Conference on Computer Vision*, pages 72–88. Springer, 2022. 3, 6
- [55] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. *arXiv preprint arXiv:2310.16818*, 2023. 3
- [56] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 3
- [57] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023. 2
- [58] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023. 2
- [59] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaee, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1
- [60] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. *arXiv preprint arXiv:2304.12439*, 2023. 3
- [61] Greg Turk. Texture synthesis on surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 347–354, 2001. 2
- [62] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 1, 3, 8
- [63] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the art in example-based texture synthesis. *Eurographics 2009, State of the Art Report, EG-STAR*, pages 93–117, 2009. 2
- [64] Li-Yi Wei and Marc Levoy. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 355–360, 2001. 2
- [65] Bangbang Yang, Wenqi Dong, Lin Ma, Wenbo Hu, Xiao Liu, Zhaopeng Cui, and Yuewen Ma. Dreamsphere: Dreaming your room space with text-driven panoramic texture propagation. *arXiv preprint arXiv:2310.13119*, 2023. 2
- [66] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ipadapter: Text compatible image prompt adapter for text-to-image diffusion models. 2023. 5, 7
- [67] Jonathan Young. xatlas, 2018. <https://github.com/jpcy/xatlas>. 5
- [68] Rui Yu, Yue Dong, Pieter Peers, and Xin Tong. Learning texture generators for 3d shape collections from internet photo sets. In *British Machine Vision Conference*, 2021. 3
- [69] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. Texture generation on 3d meshes with point-uv diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4206–4216, 2023. 2, 3, 4, 6, 14, 15
- [70] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2
- [71] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, October 2023. 4, 5
- [72] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 1
- [73] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics (ToG)*, 33(4):1–10, 2014. 2
- [74] Jingyu Zhuang, Chen Wang, Lingjie Liu, Liang Lin, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. *arXiv preprint arXiv:2306.13455*, 2023. 2

# Appendix

This appendix provides more qualitative results (Sec. A), several additional experiments (Sec. B), and discussion on the failure cases of our proposed texture generation approach (Sec. C).

## A. Qualitative Results

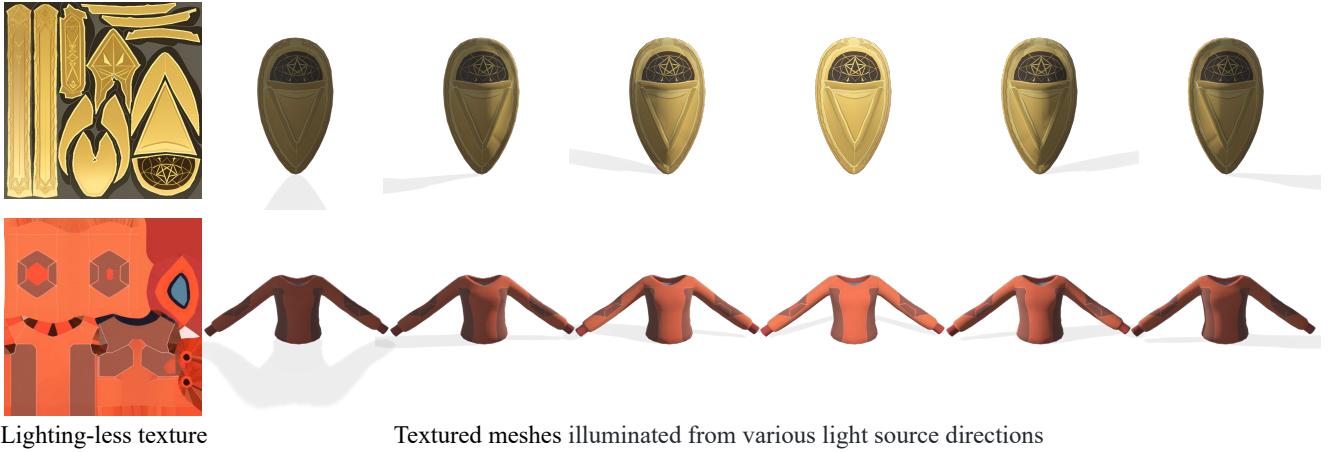


Figure 10. Lighting-less texture maps generated by Paint3D. These lighting-less textures produce appropriate shadows when the textured meshes are illuminated from different directions of light sources.



Figure 11. More samples from our best model for text-to-texture generation. Samples are generated with text prompts of the test set under various seeds. We recommend the supplemental video to see more results.



Figure 12. Additional texturing results generated by Paint3D on text-to-texture task. Each textured mesh is shown from three viewpoints.

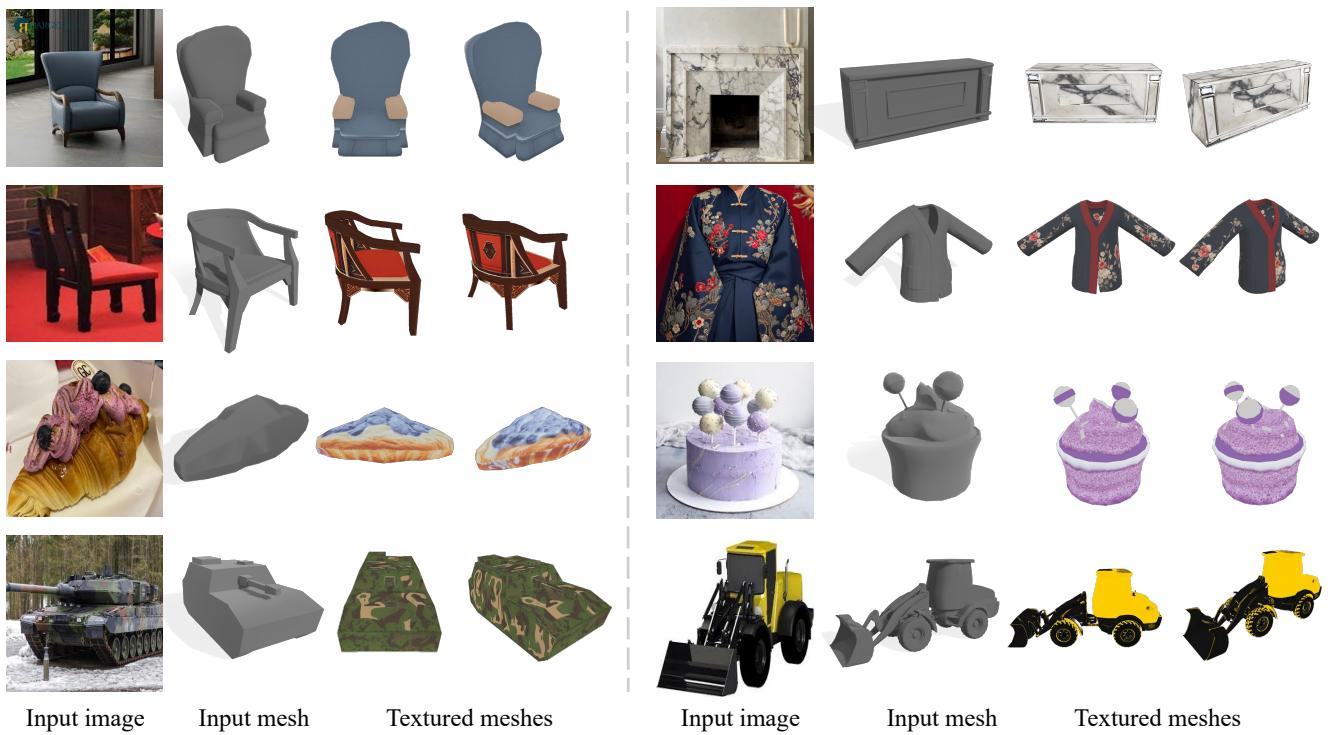


Figure 13. Additional samples from Paint3D for image-to-texture generation and each textured mesh is shown from two viewpoints. The input image conditions are collected in the wild. We recommend the supplemental video to see more results.

## B. Additional Experiments

We first study the effectiveness of the position map in the UV Inpaint and UVHD modules. Then, we provide more comparisons with category-specific texture generation approaches [69].

### B.1. Evaluation of Position Map

To demonstrate the effectiveness of position map in two texture refinement modules, UV inpainting and UVHD, we further conduct experiments on two baselines “UV inpainting w/o position map” and “UVHD w/o position map”. The “UV inpainting w/o position map” configuration refers to inpainting the uncolored area without the guidance of the position map. The “UVHD w/o position map” configuration represents the result of enhancing the texture map in UV space, without the position map. As indicated in Tab. 5, the performance shows a significant decrease when the position map is not utilized in UV inpainting or UVHD, indicating its irreplaceable function during texture refinement processing. We visualize the results of two baselines in Fig. 14 and Fig. 15. In both scenarios, the model produces inferior results compared to our full model.

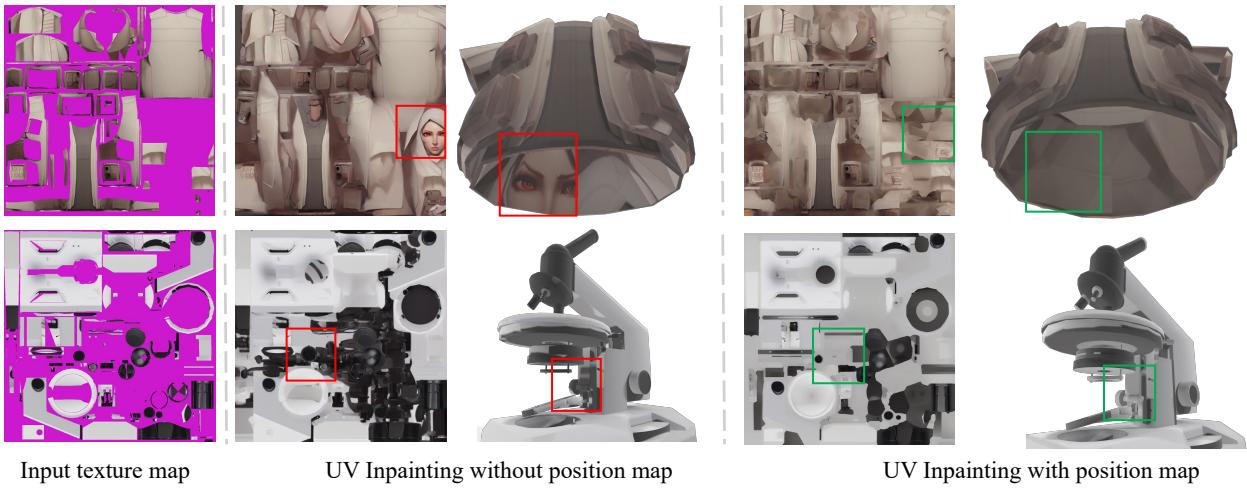


Figure 14. Visualization of the effect of the position map in the UV inpainting module. Without the position map, the inpainted texture is semantically confused. The purple area indicates the uncolored area.

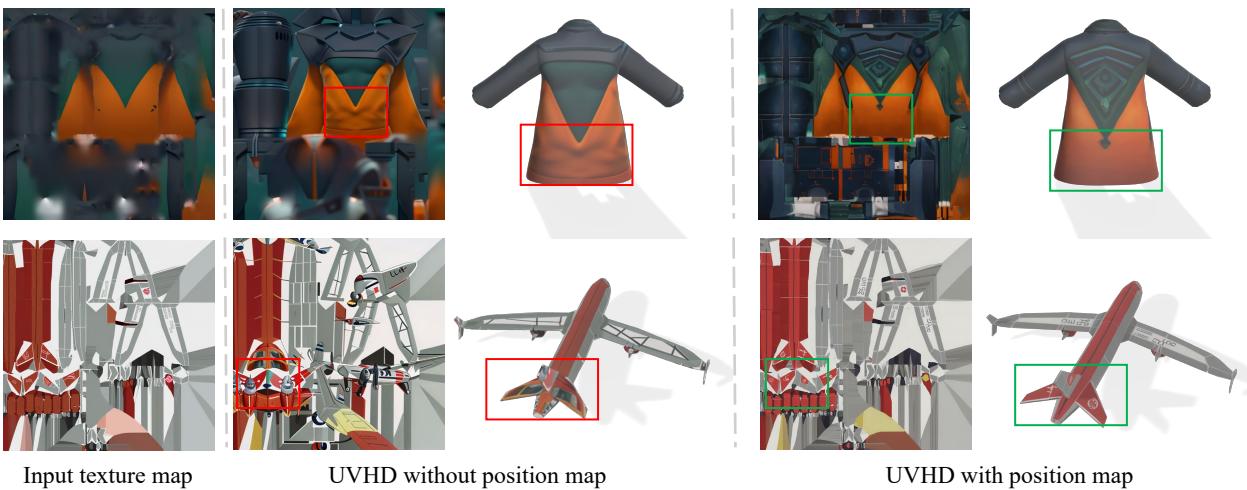


Figure 15. Visualization of the effect of the position map in the UVHD module. In the absence of the position map, the enhanced texture appears distorted (top) or lacks semantic coherence (bottom).

Method	FID $\downarrow$	KID $\downarrow$
UV inpainting w/o position map	39.29	8.36
UVHD w/o position map	37.62	7.96
Full model	<b>27.28</b>	<b>4.81</b>

Table 5. Evaluation of the effectiveness of the position map in the UV Inpaint and UVHD modules. This demonstrates the crucial role of the position map during the diffusion process in UV space.

## B.2. Comparisons with Category-Specific Model

In addition, we conduct comparison experiments with a category-specific approach on the chair and table categories of ShapeNet [4]. We choose Point-UV [69] as the baseline because 1) it represents the current state-of-the-art for category-specific texture generation, and 2) it has the conditional texture generation capability under both text and image conditions. For the input conditions, we utilize text and images as provided in [69]. As shown in Fig. 16, Paint3D achieves comparable results with Point-UV under both text and image conditions.

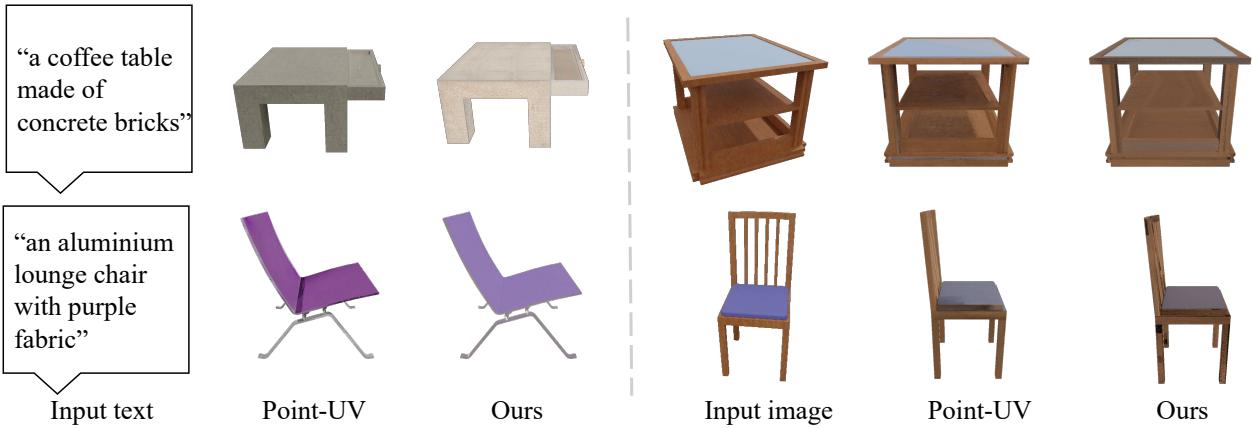


Figure 16. Qualitative comparisons on texture generation conditioned under text prompt (left) and image condition (right) on ShapeNet dataset [4]. We compare our textured mesh against those generated by the state-of-the-art category-specific approach, Point-UV [69]. In the categories of table and chair, Paint3D achieves comparable results with Point-UV under both text and image conditions.

## C. Discussion on failure case

Our approach still suffers from the multi-faces problem in the coarse stage which will result in a failure case. This issue primarily arises from the inconsistency of multi-view texture images sampled by the pre-trained 2D diffusion model, as it is not explicitly trained on multi-view datasets. We believe that fine-tuning or retraining 2D diffusion models on large-scale multi-view datasets will improve the multi-view consistency of textures.



Figure 17. Visualization of our failure cases. Paint3D still suffers from the multi-faces problem in the coarse stage which will result in a failure case. Here, Paint3D generates duplicate mouse or lion faces in both the front and back views