Text-Guided Texture Generation for 3D Objects with Progressive Sampling

Anonymous CVPR submission

Paper ID 4556

(a) 3D Mesh (b) TEXTure (c) Text2Tex (d) Fantasia3D (e) ProlificDreamer (f) Ours

Figure 1. Given a 3D mesh, we present text-driven texture generation results from previous state-of-the-art approaches as well as our proposed method.

Abstract

Given a 3D mesh, we aim to synthesize 3D textures that correspond to arbitrary textual descriptions. In this paper, we present a novel texture sampling scheme leveraging a pre-trained text-to-image diffusion model. Current methods for generating and aggregating textures from sampled views often result in prominent seams or excessive smoothing. To tackle these issues, we propose to maintain a texture map that is parameterized by the denoising step and updated after each sampling step of the diffusion model to progressively reduce the view discrepancy and introduce an attention-guided aggregation strategy to broadcast the appearance information across views. Additionally, we develop a noise resampling technique that aids in the estimation of noise, generating inputs for subsequent denoising steps, as directed by the text prompt and current texture map. Through an extensive amount of qualitative and quantitative evaluations, we demonstrate that our proposed method produces significantly better texture quality for diverse 3D objects with a high degree of view consistency and rich appearance details, outperforming current state-of-the-art methods. Furthermore, our proposed texture generation technique can also be applied to texture editing while preserving the original identity.

1. Introduction

Generating high-quality 3D content is an essential component of visual applications in films, games, and upcoming AR/VR industries. While many prior works on 3D synthesis have focused on the geometric components of the assets, textures have garnered less attention which play a vital role in enhancing the realism of 3D assets. In this paper, our objective is to achieve automatic text-driven 3D texture synthesis for arbitrary meshes.

Recently, the research community has witnessed remarkable progress in text-to-image (T2I) generators by utilizing diffusion architectures trained on large-scale datasets [13, 14, 28, 30]. However, the generation of 3D assets still faces challenges due to the limited size of 3D datasets [5, 8, 35], characterized by overly simplified textures. To this end, the existing methods have been harnessing the visual information encoded in the image priors of pre-trained T2I diffusion models. A thread of studies, such as score distillation sampling (SDS) and variational score distillation (VSD) [7, 17, 25, 34], aim to distil the diffusion priors as score functions to optimize a 3D representation, ensuring that its rendered outputs align well with the high-likelihood image priors. However, despite remarkable successes in 2D-to-3D conversion, there are noticeable shortcomings. Specifically, textures generated using score distillation pipelines tend to exhibit over-saturation, as illustrated

in Fig. 1(d), or suffer from issues such as blurry edges and color artifacts, as seen in Fig. 1(e).

Beyond score distillation methods, another thread of studies for texture synthesis involves directly utilizing the image sampling process of diffusion models to create multi-view images, all the while ensuring consistency in the UV texture space. Several approaches have been developed to adapt an image inpainting pipeline to progressively aggregate the images generated from T2I models across different views onto the texture map [6, 27]. While these methods can produce high-fidelity textures for specific views, they often result in noticeable seams on the aggregated texture map. This issue mainly stems from the error accumulation during the autoregressive view inpainting process, as evident in Fig.1 (b) and (c).

Therefore, given the sequential characteristics of the denoising process in diffusion models, we are motivated to posit that tackling view inconsistencies at each sampling step could yield more effective results. A direct approach, as suggested by [4] is to maintain a consistent texture map at every sampling stage in the latent space and get the RGB texture by averaging the decoded latent features from different viewpoints. Nevertheless, the generated textures appear overly smooth, lacking fine details.

In response to the challenges of view inconsistency and over-smoothing in texture generation, we present a novel texture sampling scheme for synthesizing textures using pre-trained latent diffusion models. Specifically, we maintain a view-consistent texture map in the RGB space that is parameterized by the sampling step of diffusion model. It is updated at each denoising step to gradually reveal texture details as the denoising step proceeds. In detail, by leveraging Diffusion Denoising Implicit Models (DDIM), at each step of the diffusion model, we first predict the *denoised observations* of sequentially sampled views around the 3D objects, which are then decoded into RGB space and integrated into the texture map via a pipeline that employs attention-guided view aggregation. Second, the aggregated texture maps are utilized in the noise prediction phase of DDIM. More specifically, a novel text and texture-guided resampling mechanism is developed to predict the denoised input for the next sampling step, ensuring that these inputs of different views not only exhibit consistency but also retain the detail of the textures.

In summary, our key contributions can be outlined as follows: (1) we propose a novel texture sampling scheme for text-driven texture generation, leveraging a pre-trained 2D image generation model; (2) we demonstrate the effectiveness of our approach in texturing diverse 3D objects, showcasing superior performance compared to state-of-the-art methods. It is noteworthy that our proposed framework can naturally support text-driven texture editing as well.

2. Related Work

2.1. Diffusion Models in 3D Domain

Inspired by the success of 2D image generation with diffusion models, researchers have also attempted to utilize diffusion models to generate 3D objects in the form of various representations, such as point clouds [18, 24, 38, 40], and neural fields [23, 33]. For example, Point·E [24] trains a diffusion model using a large synthetic 3D dataset to produce a 3D RGB point cloud conditioned on a synthesized single view from a text prompt. However, these works mainly focus on geometry generation and do not specifically tackle 3D texture synthesis. Yu *et al*. [37] trains a diffusion model for mesh texture generation of specific object categories. Although Shap·E [15] is proposed to directly generate the parameters of implicit functions that can be rendered as both textured meshes and neural radiance fields, it cannot generalize to incorporate arbitrary text prompts. Moreover, the generated textures tend to be over smoothed with rather low quality as compared with the generated images from the T2I model.

2.2. Lifting pre-trained 2D generative models to 3D

Initially, the process of distilling 3D Objects from pre-trained 2D models has been enhanced by the development of text-image joint embedding, such as Contrastive Language-Image Pre-training (CLIP) [26]. For example, CLIP-Mesh [21] learns to generate a mesh with the guidance of the CLIP text embedding and the corresponding image embedding from the diffusion model. However, since the CLIP guidance is rather sparse, the generated 3D models for CLIP-based approaches [16, 20, 29] are rather coarse.

Recently, researchers have leveraged large-scale 2D T2I diffusion models to distil individual 3D objects in the form of neural radiance fields. Among various distilling approaches, a dominant one is Score Distillation Sampling (SDS). DreamFusion [25] pioneered the approach with many follow-up works works [7, 9, 17, 19, 31, 32, 34]. For example, Magic3D [17] proposed a coarse-to-fine strategy to improve generation quality. Latent-NeRF [19] performed distillation in the latent space of LDM. A crucial drawback of this line of work is that SDS typically requires strong guidance, resulting in low diversity and oversaturation of the generated textures. ProlificDreamer [34] proposed to address this issue with a Variational Score Distillation (VSD) algorithm that adopts a particle-based variational inference to estimate the distribution of the 3D scenes instead of a single point as in SDS. Yet, it still suffers from issues like blurry edges and color artefacts.

Texture Synthesis with Multiview Denoising. Instead of relying on the lengthy optimization of score distillation pipelines, an alternative research direction is directly leveraging the sampling process in diffusion models to synthe-

size UV textures. TEXTure [27] and Text2tex [6] adopt a depth-aware diffusion model [28] to progressively paint the mesh surface from different views and aggregate the images generated from the T2I model of sampled views into the texture map. While rich textures and details can be faithfully synthesized, there were obvious seams on the aggregated texture map due to error accumulation in the process of the autoregressive view update. To further reduce view inconsistencies, TexFusion [4] proposed to interleave texture aggregation with denoising steps in different camera views and maintained a latent texture map at each sampling step. To convert latent features to RGB textures, they optimized an intermediate neural color field on the decoding of 2D renders of the latent texture which would wash out the rich details [22]. Our proposed approach distinguishes itself from previous methods with its ability to generate 3D-consistent textures while preserving rich details in the meantime.

3. Proposed Method

3.1. Overview

In this section, we present an overview of our proposed Progressive Texture Sampling Scheme to synthesize view-consistent textures from a pre-trained T2I generation model. We first introduce the sampling process of the Denoising Diffusion Implicit Models (DDIM) [30], which forms the basis of our texture sampling approach.

DDIM Sampling. Assuming we sequentially sample N distinct views around a 3D mesh, the DDIM sampling process for each sampled viewpoint i at the denoising step t can be described as follows:

$$\hat{x}_0^i(x_t^i) = \frac{x_t^i - \sqrt{1-\alpha_t}\cdot\epsilon_\theta(x_t^i)}{\sqrt{\alpha_t}}, \tag{1}$$

$$x_{t-1}^i = \sqrt{\alpha_{t-1}}\cdot\hat{x}_0^i(x_t^i) + \sqrt{1-\alpha_{t-1}}\cdot\epsilon_\theta(x_t^i), \tag{2}$$

where x_t^i represents the noisy latent feature, and $\epsilon_\theta(x_t^i)$ represents the estimated noise from the pre-trained T2I diffusion model. At each denoising step t, we calculate $\hat{x}_0^i(x_t^i)$, representing the predicted x_0^i and dubbed as the *denoised observation* of x_t^i. α_t is the total noise variance parameterized via denoising step t.

Texture Sampling. To address the inconsistencies arising from independently generated different views, our texture sampling scheme is proposed by leveraging the sequential nature of the denoising process of the diffusion model and maintaining the 3D consistency of the generated texture at each denoising step.

In detail, at each denoising step t, we will conduct the following two steps. First, we compute a view-consistent denoised observation of a texture map $\hat{U}_0(x_t^{1\cdots N})$ by progressively aggregating the *denoised observations* $\hat{x}_0^i(x_t^i)$ where $i = 1,\ldots,N$ via an attention-guided view aggregation pipeline (Sec. 3.2). For brevity, we denote $\hat{U}_{0,t}^i$ as the partial texture map $\hat{U}_0(x_t^{1\cdots i})$ and $\hat{U}_{0,t}^N$ as the complete texture map $\hat{U}_0(x_t^{1\cdots N})$, both at denoising step t. Second, the calculation of the noisy latent feature for the upcoming denoising step $t-1$ is based on the current latent feature x_t^i, the *denoised observation* $\hat{x}_0^i(x_t^i)$, as well as the current texture map $\hat{U}_{0,t}^N$ as detailed in Sec. 3.3,

$$x_{t-1}^i \sim q(x_{t-1}^i | x_t^i, \hat{x}_0^i(x_t^i), \hat{U}_{0,t}^N). \tag{3}$$

Following the DDIM sampling, we go through the above process with T denoising steps to arrive at the final generated texture map $\hat{U}_{0,1}^N$. The texture sampling scheme is further illustrated in Fig. 2. We present the above-mentioned two major steps in the following sections.

3.2. View Sampling&Aggregation (VSA)

The objective of this stage is to determine the denoised observation of the texture map $\hat{U}_{0,t}^N$, based on the latent features of $x_t^{1\cdots N}$ at each viewpoint for the current step.

3.2.1 View Sampling.

Following DDIM sampling, for each sampled view i at time step t, the *denoised observation* $\hat{x}_0^i(x_t^i)$ can be computed as in Eq. 1. The latent features are then decoded into images I_t^i in the RGB space via the VAE decoder $\mathcal{D}$ of the pre-trained stable diffusion [28],

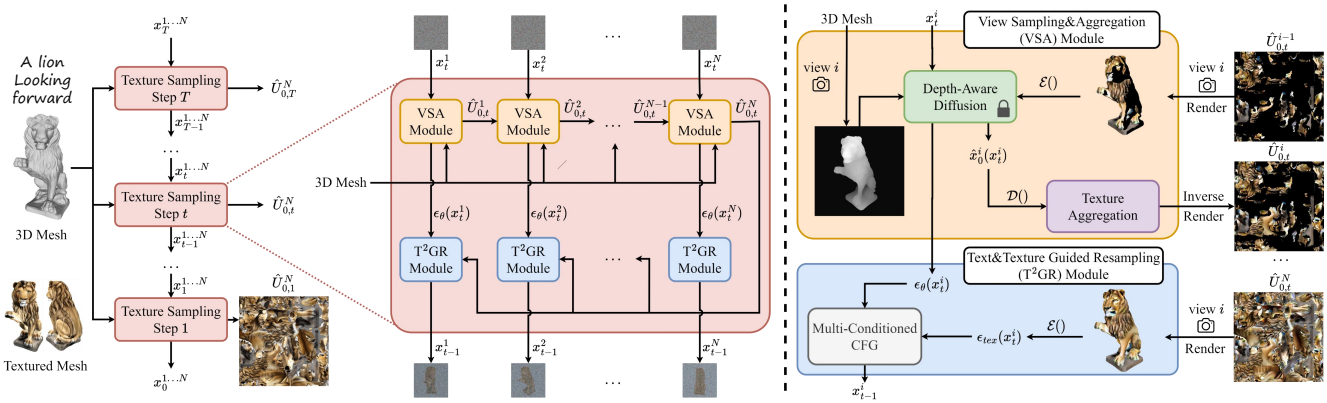
$$I_t^i = \mathcal{D}(\hat{x}_0^i(x_t^i)). \tag{4}$$

A naive solution of generating the denoised observation of texture map $\hat{U}_{0,t}^N$ is to directly fuse I_t^i (for $i = 1,\ldots,N$), after inverse rendering onto the UV space. However, it will lead to noticeable seams between adjacent views due to the separate and view-inconsistent generation of each I_t^i.

3.2.2 View Aggregation.

To address the issue of view inconsistency, we adopt an autoregressive generation strategy. This involves generating the *denoised observation* $\hat{x}_0^i(x_t^i)$ based on previous denoised views $\hat{x}_0^{1\cdots i-1}(x_t^{1\cdots i-1})$. Starting with the first viewpoint $i = 1$, we perform inverse rendering to map I_t^i onto the UV space, obtaining the partial texture map $\hat{U}_{0,t}^i$. Then for the subsequent viewpoint $i+1$, the prediction of $\hat{x}_0^{i+1}(x_t^{i+1})$ depends on the current partial texture map $\hat{U}_{0,t}^i$. More specifically, we render the partial texture map $\hat{U}_{0,t}^i$ onto viewpoint $i+1$, denoted as $Render^{i+1}(\hat{U}_{0,t}^i)$. The rendered output is then fed as input to the VAE encoder $\mathcal{E}$ for obtaining the latent features $G_{0,t}^{i+1}$,

$$G_{0,t}^{i+1} = \mathcal{E}(Render^{i+1}(\hat{U}_{0,t}^i)). \tag{5}$$



(a) Illustration of our texture sampling scheme (b) Details of a denoising step

Figure 2. Overview of our proposed method. First of all, we sample N viewpoints across the objects. Our texture sampling scheme is an interleaved process of multi-view texture aggregation and diffusion denoising. Specifically, our texture sampling process is structured into T steps of diffusion process. As shown in (a), at denoising step t, it takes the noisy latent features of sampled views ($x_t^{1\cdots N}$) as input to predict the noisy features for the next denoising step ($x_{t-1}^{1\cdots N}$) as well as a time-dependent texture map ($\hat{U}_{0,t}^N$). Upon completing T steps of sampling, the final texture map ($\hat{U}_{0,1}^N$) will be achieved. To elaborate more on each denoising step, we present two novel modules: View Sampling&Aggregation (VSA) module and Text&Texture Guided Resampling (T^{2}GR) module. As shown in (b), for view i, the VSA module is used to generate denoised observation $\hat{x}_0^i(x_t^i)$ which will be aggregated onto texture map to form $\hat{U}_{0,t}^{i+1}$. After iterating over all sampled views starting from $i=1$ to N, we obtain $\hat{U}_{0,t}^N$ for each denoising step. Conditioned on the current estimation of texture map $\hat{U}_{0,t}^N$, the T^{2}GR module will update the noise estimations $\epsilon_{tex}(x_t^i)$ to calculate the noisy latent feature x_{t-1}^i for the next denoising step.

The computation of the *denoised observation* for viewpoint $i+1$ at step t is performed as follows:

$$\hat{x}_0^{i+1}(x_t^{i+1}) = \frac{x_t^{i+1} - \sqrt{1-\alpha_t}\cdot\epsilon_\theta(x_t^{i+1})}{\sqrt{\alpha_t}}, \tag{6}$$

with

$$\begin{aligned} x_t^{i+1} =& x_t^{i+1}\odot\mathcal{M}_{\downarrow}^{i+1} + (\sqrt{\alpha_t}\cdot G_{0,t}^{i+1} \\ &+ \sqrt{1-\alpha_t}\cdot\epsilon)\odot(1-\mathcal{M}_{\downarrow}^{i+1}), \end{aligned} \tag{7}$$

where $\mathcal{M}^{i+1}$ denotes the mask for regions that are viewed for the first time at view $i+1$ in RGB space, and $\downarrow$ symbolizes downsampling to the resolution of latent features.

Attention-Guided View Synthesis. As indicated in Fig. 7, sequential generation across different viewpoints often fails to ensure appearance consistency. To address this, we introduce a novel attention-guided cross-view generation strategy. Drawing inspiration from the work of Cao *et al.* [3], we believe the Key and Value features in the self-attention module of the stable diffusion encapsulate the local contents and textures of generated images. In detail, we regard the front view as the reference view and propagate the Key and Value of the reference view to other views. The process can be outlined as follows:

$$\epsilon_\theta(x_t^{ref}),\ Q_t^{ref},\ K_t^{ref},\ V_t^{ref} \leftarrow Unet_\theta(x_t^{ref}), \tag{8}$$

$$\epsilon_\theta(x_t^i) \leftarrow Unet_\theta(x_t^i, K_t^{ref}, V_t^{ref}). \tag{9}$$

Herein, Q_t^{ref}, K_t^{ref}, and V_t^{ref} denote the Query, Key, and Value features from the self-attention module of the reference view, respectively. In Eq. 9, the Key and Value features for each viewpoint are substituted with those from the reference view to calculate its estimated noise. Following this substitution, for each viewpoint i, the *denoised observation* $\hat{x}_0^i(x_t^i)$ is updated in accordance with Eq. 1. As shown in Fig. 3, the texture details will gradually appear in the *denoised observation* as the diffusion process proceeds.

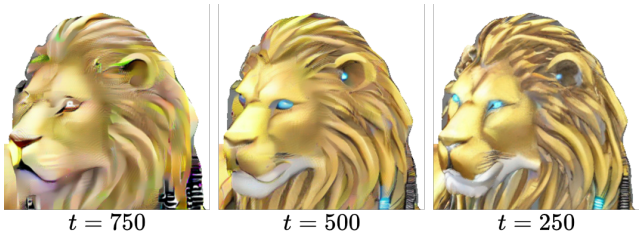


Figure 3. *Denoised observation* $\hat{x}_0(x_t^i)$ with text prompt "A Cyber Punk lion". The high-frequency information is gradually generated during sampling.

3.3. Text&Texture Guided Resampling (T^{2}GR)

Upon obtaining the current denoised observation of the texture map $\hat{U}_{0,t}^N$, we perform Text&Texture Guided Resampling (T^{2}GR) to update the noisy latent features $x_{t-1}^{1\cdots N}$ for the next denoising step.

As shown in Eq. 2, the derivation of x_{t-1}^i depends on the estimated noise $\epsilon_\theta(x_t^i)$ and the *denoised observation* $\hat{x}_0^i(x_t^i)$. Given that $\hat{x}_0^i(x_t^i)$ is expected to exhibit view consistency, as it is maintained by the texture map $\hat{U}_{0,t}^N$, recal-

culating the noise map $\epsilon_\theta(x_t^i)$ under the guidance of current denoised observation of texture map $\hat{U}_{0,t}^N$ ensures to preserve the view consistency. Specifically, in Eq. 1 we set $\hat{x}_0^i(x_t^i)$ equal to the current encoded render of the texture map $\hat{U}_{0,t}^N$ at view i. From this, we derive the recalculated noise map $\hat{\epsilon}_{tex}(x_t^i)$ as follows:

$$\hat{\epsilon}_{tex}(x_t^i) = \frac{x_t^i - \sqrt{\alpha_t} \cdot \mathcal{E}(Render^i(\hat{U}_{0,t}^N))}{\sqrt{1-\alpha_t}}. \quad (10)$$

This recalculated noise map is then utilized in place of $\epsilon_\theta(x_t^i)$ in Eq. 1 and Eq. 2 for the computation of x_{t-1}^i.

While our noise map update strategy ensures view consistency, it tends to result in over-smoothed images (as shown in Fig. 8). This is primarily because the VAE encoder $\mathcal{E}$ in the stable diffusion model compresses high-frequency details, referred to as *imperceptible details*, as noted by [28]. The repeated use of the encoder $\mathcal{E}$ leads to an accumulation of this detail compression, affecting the overall image quality.

To avoid over-smoothness, we take $\hat{U}_{0,t}^N$ as an additional condition to the diffusion model rather than directly replacing $\epsilon_\theta(x_t^i)$ with $\epsilon_{tex}(x_t^i)$. Basically, we want to compute a texture-conditioned noise estimation which we denote as $\epsilon_{tex}(x_t^i|\hat{U}_{0,t}^N)$. By analyzing the formulation of $\epsilon_\theta(x_t^i)$, we see that it is essentially a weighted combination of conditional noise prediction $\epsilon_\theta(x_t^i|c)$ and unconditional noise prediction $\epsilon_\theta(x_t^i|\varnothing)$, following the Classifier-Free Guidance (CFG) introduced in [13]:

$$\epsilon_\theta(x_t^i) = \epsilon_\theta(x_t^i|\varnothing) + \omega(\epsilon_\theta(x_t^i|c) - \epsilon_\theta(x_t^i|\varnothing)), \quad (11)$$

where c and $\varnothing$ represent the text prompt and null-text prompt, respectively, and ω is a user-specified weight.

Similarly, $\epsilon_{tex}(x_t^i)$ should follow the same formulation of CFG,

$$\epsilon_{tex}(x_t^i) = \epsilon_\theta(x_t^i|\varnothing) + \omega(\epsilon_{tex}(x_t^i|\hat{U}_{0,t}^N) - \epsilon_\theta(x_t^i|\varnothing)). \quad (12)$$

Thus, to disentangle the texture-conditioned noise estimation $\epsilon_{tex}(x_t^i|\hat{U}_{0,t}^N)$, we subtract the null-text conditioned noise estimation from $\epsilon_{tex}(x_t^i)$. Here we set $\epsilon_{tex}(x_t^i) = \hat{\epsilon}_{tex}(x_t^i)$. The computation for the texture-conditioned noise estimation $\epsilon_{tex}(x_t^i|\hat{U}_{0,t}^N)$ is as follows:

$$\epsilon_{tex}(x_t^i|\hat{U}_{0,t}^N) = \frac{1}{\omega}(\hat{\epsilon}_{tex}(x_t^i) - \epsilon_\theta(x_t^i|\varnothing)) + \epsilon_\theta(x_t^i|\varnothing). \quad (13)$$

In the end, we formulate our multi-conditioned CFG for final noise estimation, which is conditioned on both the textual prompt and texture map:

$$\begin{aligned}\epsilon_m(x_t^i) =& \epsilon_\theta(x_t^i|\varnothing) \\ &+ \omega_1(\epsilon_\theta(x_t^i|c) - \epsilon_\theta(x_t^i|\varnothing)) \\ &+ \omega_2(\epsilon_{tex}(x_t^i|\hat{U}_{0,t}^N) - \epsilon_\theta(x_t^i|\varnothing)),\end{aligned} \quad (14)$$

where $\omega_1 + \omega_2 = \omega$. We exploit a large ω_2 for early sampling steps, which will decrease linearly from ω to 0 in the process of denoising. The comprehensive derivation of Eq. 14 can be found in the supplementary materials. Finally, we compute x_{t-1}^i for the subsequent denoising step by letting $\epsilon_\theta(x_t^i) = \epsilon_m(x_t^i)$ in Eq. 1 and Eq. 2.

4. Experiments

4.1. Implementation Details

We employ the depth-aware diffusion model provided by ControlNet [39] as our T2I backbone with denoising steps $T = 40$. To render objects, we take eight different viewpoints around the object. The pose is sampled in spherical coordinates, with elevation angles being zero and azimuth angles uniformly sampled between [0°, 360°]. An additional top view is sampled. Additionally, we employ the Xatlas [36] tool to compute the UV atlas for a given mesh.

Dataset. Our experiment incorporates a diverse collection of 45 meshes, sourced from various datasets such as Objaverse [8] and ThreeDScans [1], with 2 to 3 distinct prompts for each mesh. Please refer to the supplementary for details.

4.2. Comparison Methods

We conduct experimental comparison over several state-of-the-art approaches, including TEXTure [27], Text2Tex [6], Fantasia3D [7], ProlificDreamer [34] and TexFusion [4]. For TEXTure, Text2Tex, and Fantasia3D, we use their respective publicly available codebase. As the official code for ProlificDreamer is not yet accessible, we adopt the implementation of ThreeStudio [10] and replaced its backbone with ControlNet [39] to recognize the depth. In the case of TexFusion, where the implementation is not available, our analysis is limited to a qualitative assessment using results extracted directly from the original paper. Notably, for all the compared approaches, the geometry remains fixed during texture generation.

4.3. Qualitative Comparison

We provide visual comparison in Fig. 4 and Fig. 5. Specifically, in Fig. 4, we showcase the robustness of our approach in addressing fragmented textures against progressively texture aggregation approaches, namely TEXTure [27] and Text2Tex [6]. This improvement is credited to our use of attention-guided view aggregation, combined with a distinct text&texture guided resampling approach that maintains view consistency at each denoising step to persistently enchance 3D consistency.

In Fig. 5, we compare with score distillation based approaches, namely Fantasia3D [7] and ProlificDreamer [34]. As demonstrated, Fantasia3D typically produces textures that are over-smoothed and over-saturated, while ProlificDreamer, though more detailed and contrasted, is marred

Figure 4. Visual comparison of our proposed method against TEXTure [27] and Text2Tex [6].

Methods	FID ↓	KID×10^{-3} ↓	CLIPScore ↑
TEXTure	99.06	7.23	19.73
Text2Tex	109.94	7.17	21.26
Fantasia3D	108.58	7.52	21.14
ProlificDreamer	94.51	7.00	21.25
Ours	**84.65**	**4.27**	**22.83**

Table 1. Quantitative comparison on generated textures.

by evident artifacts of blurry edges. In contrast, our method surpasses these distillation-based methods by generating more realistic high-quality results.

We also present a qualitative comparison of our method with TexFusion [4] in Fig. 6. TexFusion employed instantNGP [22] to mitigate inconsistencies post-decoding of latent features into RGB space, which often led to over-smoothed results. In contrast, our method effectively generates textures that are consistent across views and retain rich details. Please refer to supplementary materials for more visual results.

4.4. Quantitative Comparison

Evaluation Metrics. For quantitative evaluation of the generated texture, we employ two widely used image quality and diversity evaluation metrics, including Frechet Inception Distance (FID) [12] and Kernel Inception Distance (KID) [2]. These metrics are instrumental in measuring the distribution similarity between two sets of images. For each comparison method, we render a set of images by uniformly sampling 32 different views of the generated textured mesh. To establish a ground truth image set, we follow the approach outlined by Cao *et al*. [4] which used a depth-conditioned ControlNet to synthesize images conditioned on rendered depth maps and corresponding textual prompts. The background pixels have been removed from all images to mitigate the influence caused by unconstrained background. Additionally, we incorporate the CLIPScore metric [11] to assess the congruence and resemblance between the generated images and their associated text prompts. Specifically, for each method, we calculate the average CLIPScore across all rendered images relative to the given text prompts.

We present the quantitative evaluations of the above-mentioned methods on FID, KID and CLIPScore in Tab. 1. Notably, our approach demonstrates superior performance, outstripping the other methods by at least 10.4% in FID and 39.0% in KID. The figures showcase our method's capability to generate textures that not only are more realistic but also exhibit a wide variety of appearances across diverse

Figure 5. Visual comparison of our proposed method against Fantasia3D [7] and ProlificDreamer [34].

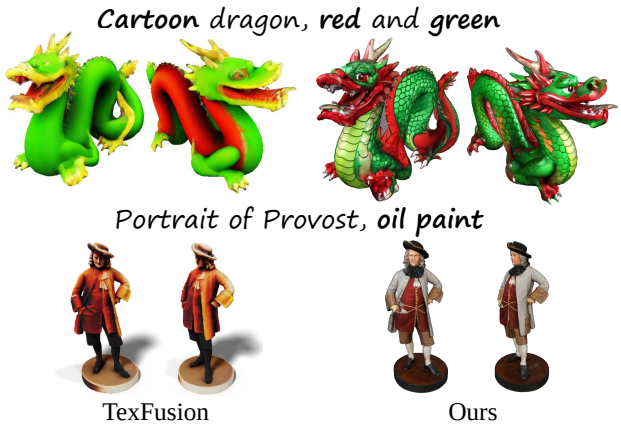


Figure 6. Visual comparison of our proposed method against TexFusion [4]. The results of TexFusion are directly copied from its original paper

objects.

User Study. To analyze the quality of the generated textures and their fidelity to the corresponding text prompts, we conducted a detailed user study of our method against four baseline methods. We randomly select 40 meshes from our collected data and feed them along with a text prompt as the input for each method. For each of these 40 selections, we generate 360° rotating view videos using both our method and one of the baseline methods and display them side-by-side. Participants in the study are then requested to select the video that not only better matched the given caption but also exhibited superior quality. The user study yielded a dataset of 2,480 responses from 62 participants. We report the user preferences in Tab. 2. The results indicate that our method is notably more effective in producing high-quality textures that are preferred by human evaluators.

	TEXTure ↑	Text2Tex ↑	Fantasia3D ↑	ProlificDreamer ↑
Ours	64.72%	71.46%	70.97%	69.18%

Table 2. User Study Preference: The entries in the table indicate our preference over other methods. A higher value represents a greater preference.

4.5. Ablation Study

We first visually evaluate the impact of the attention-guided view synthesis as shown in Fig. 7. The results demonstrate that our proposed method with attention-guided view synthesis is able to generate textures which have a consistent

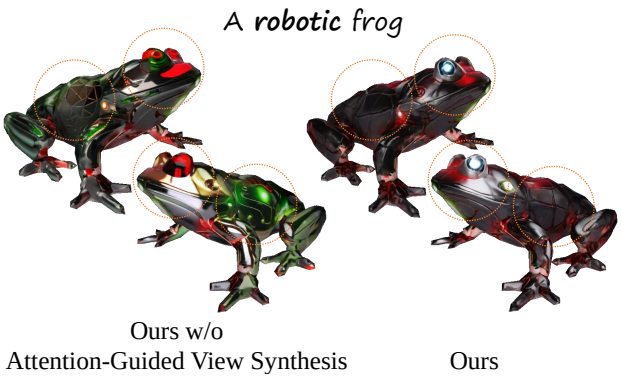


Figure 7. Visual comparison of ablation study over attention-guided view synthesis. Without attention-guided view synthesis, the frog has different appearance patterns and color tones over different sides such as eyes and back.

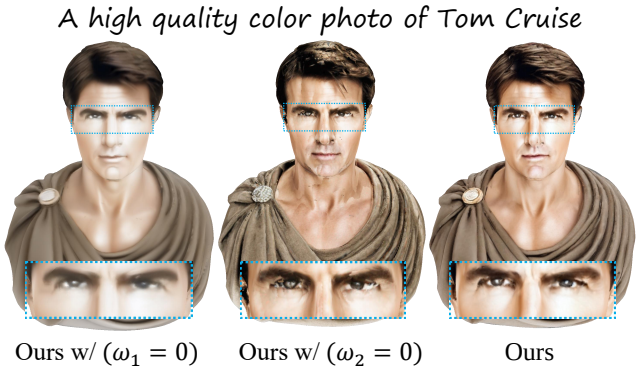


Figure 8. Visual comparison of ablation study over T^{2}GR module.

Methods	FID ↓	KID×10^{-3} ↓
Ours w/o AGVS	98.62	5.01
Ours w/ ($\omega_1 = 0$)	95.97	4.60
Ours w/ ($\omega_2 = 0$)	99.46	5.22
Ours	**84.65**	**4.27**

Table 3. Ablation study over Attention-Guided View Synthesis (AGVS) and T^{2}GR.

appearance in different viewpoints.

We also evaluate the impact of the T^{2}GR by keeping $\omega_1 = 0$ and $\omega_2 = 0$ in Eqn. 14, respectively. As shown in Fig. 8, the first figure with $\omega_1 = 0$ lacks high-frequency details and tends to be over-smoothed, while the middle figure with $\omega_2 = 0$ lacks texture guidance and the aggregated texture from all viewpoints is fragmented. We also evaluate the generation quality using FID and KID in Tab. 3, which shows that our method with attention-guided view synthesis and T^{2}GR outperforms other variants by a large margin.

Figure 9. Applications of our proposed texture sampling scheme for text-driven texture editing.

4.6. Application: Texture Editing

Our proposed texture sampling scheme can also be applied to texture editing, as shown in Fig. 9. It shares the same pipeline with texture generation, but here we replace the depth-aware ControlNet with the MultiControlNet [39] that combines both the depth-guided and edge-guided generation to preserve the original identity, where the canny edges are extracted from the generated views.

5. Conclusion

In this paper, we present a novel texture sampling scheme for text-driven texture generation on 3D meshes, leveraging depth-aware diffusion models. To address the significant challenges in 3D content generation, particularly in producing textures that are consistent across views and rich in detail, we first propose to maintain a time-dependent texture map that evolves with each denoising step to progressively reduce the view discrepancy. Specifically, at each denoising step, the texture is aggregated from the *denoised observations* of sampled views under our attention-guided inpainting process. It is then utilized in our Text&Texture Guided noise resampling procedure to further guide the estimated noise fed into the next denoising step. The effectiveness of our method is evident in its ability to generate superior-quality textures for diverse 3D objects as well as in its adaptability for texture editing purposes.

Discussion. While we have showcased superior performance in texture generation compared to existing approaches, the overall quality of the generated 3D textures still exhibits a gap when compared to 2D image generation. Striking a balance between 3D consistency and the generation quality of specific views remains a challenge. Additionally, a limitation of this work is the lack of disentanglement of material and lighting from the generated textures, which is crucial for tasks such as object relighting and material editing. We leave this as future work to explore.

References

[1] Three d scans. https://threedscans.com, 2012. 5

[2] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 6

[3] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv preprint arXiv:2304.08465*, 2023. 4

[4] 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. 2, 3, 5, 6, 7

[5] 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. 1

[6] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023. 2, 3, 5, 6, 4

[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. 1, 2, 5, 7

[8] 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. 1, 5

[9] Yanhui Guo, Xinxin Zuo, Peng Dai, Juwei Lu, Xiaolin Wu, Li Cheng, Youliang Yan, Songcen Xu, and Xiaofei Wu. Decorate3d: Text-driven high-quality texture generation for mesh decoration in the wild. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[10] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. https://github.com/threestudio-project/threestudio, 2023. 5

[11] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 6

[12] 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. 6

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1, 5

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[15] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2

[16] 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. 2

[17] 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. 1, 2

[18] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 2

[19] 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. 2

[20] 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. 2

[21] 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. 2

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 3, 6

[23] Gimin Nam, Mariem Khlifi, Andrew Rodriguez, Alberto Tono, Linqi Zhou, and Paul Guerrero. 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. *arXiv preprint arXiv:2212.00842*, 2022. 2

[24] 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. 2

[25] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[27] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 2, 3, 5, 6, 4

[28] 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, 3, 5

[29] Aditya Sanghi, Rao Fu, Vivian Liu, Karl DD Willis, Hooman Shayani, Amir H Khasahmadi, Srinath Sridhar, and Daniel Ritchie. Clip-sculptor: Zero-shot generation of high-fidelity and diverse shapes from natural language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18339–18348, 2023. 2

[30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 3

[31] 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. 2

[32] 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

[33] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4563–4573, 2023. 2

[34] 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, 2, 5, 7

[35] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 1

[36] Jonathan Young. xatlas. In *github.com/jpcy/xatlas*, 2016. 5

[37] 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

[38] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022. 2

[39] 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. 5, 8

[40] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 2

Text-Guided Texture Generation for 3D Objects with Progressive Sampling

Supplementary Material

6. Algorithm Details

To better illustrate the working flow of our proposed method, we present the detailed algorithm in Alg. 1

7. Derivation of Eq. 14

As discussed in Eq. 14 of Sec. 3.3, we apply the classifier-free guidance (CFG) on noise estimation with two conditions: the textual prompt c and the intermediate texture map $\hat{U}^i_{0,t}$. The original text guided diffusion model targets at learning $P(x_t|c)$ where x_t denotes the noisy latent feature at time step t. Now we extend the target of the original diffusion model to $P(x^i_t|c, \hat{U}^N_{0,t})$, which has an additional condition $\hat{U}^N_{0,t}$ to constrain the generated x^i_t to be view-consistent. We assume $P(c|x^i_t, \hat{U}^N_{0,t}) = P(c|x^i_t)$. Following Bayes' theorem, $P(x^i_t|c, \hat{U}^N_{0,t})$ can be reformulated as

$$P(x^i_t|c, \hat{U}^N_{0,t}) = \frac{P(x^i_t)P(c|x^i_t)P(\hat{U}^N_{0,t}|x^i_t)}{P(c, \hat{U}^N_{0,t})}. \tag{15}$$

By taking logarithm on both sides of the above equation, we get

$$\begin{aligned}\log(P(x^i_t|c, \hat{U}^N_{0,t})) =& \log(P(x^i_t)) \\ &+ \log(P(c|x^i_t)) + \log(P(\hat{U}^N_{0,t}|x^i_t)) \\ &- log(P(c, \hat{U}^N_{0,t})).\end{aligned} \tag{16}$$

As mentioned in [25], estimating $\epsilon_m(x^i_t)$ is related to predicting the score function $s_m(x^i_t)$ of the approximate marginal distribution $P(x^i_t|c, \hat{U}^N_{0,t})$, which can be formulated as:

$$s_m(x^i_t) = \nabla_{x^i_t} \log(P(x^i_t|c, \hat{U}^N_{0,t})), \tag{17}$$

$$\epsilon_m(x^i_t) = -\sigma_t s_m(x^i_t), \tag{18}$$

where σ_t is the standard deviation of the latent noise parameterized by denoising step t. The score function $\nabla_{x^i_t} \log(P(x^i_t|c, \hat{U}^N_{0,t}))$ can be further derived from Eq. 16 as:

$$\begin{aligned}\nabla_{x^i_t} \log(P(x^i_t|c, \hat{U}^N_{0,t})) =& \nabla_{x^i_t} \log(P(x^i_t)) \\ &+ \nabla_{x^i_t} \log(P(c|x^i_t)) \\ &+ \nabla_{x^i_t} \log(P(\hat{U}^N_{0,t}|x^i_t)),\end{aligned} \tag{19}$$

with

$$\begin{aligned}\nabla_{x^i_t} \log(P(c|x^i_t)) =& \nabla_{x^i_t} \log(P(x^i_t|c)) \\ &- \nabla_{x^i_t} \log(P(x^i_t)),\end{aligned} \tag{20}$$

$$\begin{aligned}\nabla_{x^i_t} \log(P(\hat{U}^N_{0,t}|x^i_t)) =& \nabla_{x^i_t} \log(P(x^i_t|\hat{U}^N_{0,t})) \\ &- \nabla_{x^i_t} \log(P(x^i_t)),\end{aligned} \tag{21}$$

which correspond to the terms in our multi-conditioned CFG in Eq. 14 as:

$$\epsilon_\theta(x^i_t|\varnothing) = -\sigma_t \nabla_{x^i_t} \log(P(x^i_t)), \tag{22}$$

$$\begin{aligned}\epsilon_\theta(x^i_t|c) - \epsilon_\theta(x^i_t|\varnothing) =& -\sigma_t(\nabla_{x^i_t} \log(P(x^i_t|c)) \\ &- \nabla_{x^i_t} \log(P(x^i_t))),\end{aligned} \tag{23}$$

$$\begin{aligned}\epsilon_{tex}(x^i_t|\hat{U}^N_{0,t}) - \epsilon_\theta(x^i_t|\varnothing) =& -\sigma_t(\nabla_{x^i_t} \log(P(x^i_t|\hat{U}^N_{0,t})) \\ &- \nabla_{x^i_t} \log(P(x^i_t))).\end{aligned} \tag{24}$$

Following CFG [13], we apply two guidance scales ω_1 and ω_2 on two guidance terms. Finally, we have the multi-conditioned CFG as:

$$\begin{aligned}\epsilon_m(x^i_t) =& \epsilon_\theta(x^i_t|\varnothing) \\ &+ \omega_1(\epsilon_\theta(x^i_t|c) - \epsilon_\theta(x^i_t|\varnothing)) \\ &+ \omega_2(\epsilon_{tex}(x^i_t|\hat{U}^N_{0,t}) - \epsilon_\theta(x^i_t|\varnothing)).\end{aligned} \tag{25}$$

8. Additional Experiments

8.1. Inference Time

In Tab. 4, we compare the inference time of our proposed method with that of baseline methods on a single NVIDIA Tesla V100 GPU.

8.2. More Qualitative Evaluations

More qualitative evaluations are shown in Fig. 10, Fig. 11, and Fig. 12.

9. User Study Details

We develop a WIX-based web application for the user study. As shown in Fig. 13, for each video pair, participants are required to choose the video that best illustrates the given textual prompt with the highest quality. They should then click the rounded check-box below the selected video and proceed to the next video pair. Finally, we determine the user preferences by counting all user selections.

10. Data Description

We present the details of our collected data in Tab. 5, Tab. 6, and Tab. 7 with corresponding textual prompts.

Algorithm 1: Progressive Texture Sampling

Input: A 3D Mesh
A textual prompt c
A set of viewpoints v_i, $i \in \{1, \dots, N\}$
Number of denoising step T
VAE encoder $\mathcal{E}$ and decoder $\mathcal{D}$
Depth conditioned ControlNet $Unet_\theta$
Output: Generated texture map $\hat{U}_{0,1}^N$

Randomly initialize $x_T^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $i \in \{1, \dots, N\}$
for $t = T, ..., 1$ **do**
  View Sampling&Aggregation (VSA):
  for $i = 1, \dots, N$ **do**
    Substitute the Key and Value features for viewpoint i with those from reference view to calculate $\epsilon_\theta(x_t^i)$ by Eq. 8 and Eq. 9
    Obtain the $\hat{x}_0^i(x_t^i)$ with x_t^i and $\epsilon_\theta(x_t^i)$ by Eq. 1
    Decode the $\hat{x}_0^i(x_t^i)$ to obtain I_t^i in RGB space by Eq. 4
    Inverse render the I_t^i to obtain the partial texture map $\hat{U}_{0,t}^i$
    if $i < N$ **then**
      Render and encode $\hat{U}_{0,t}^i$ to obtain $G_{0,t}^{i+1}$ by Eq. 5
      Update x_t^{i+1} with $G_{0,t}^{i+1}$ and view aggregation mask $\mathcal{M}^{i+1}$ by Eq. 7
    end
  end
  Text&Texture Guided Resampling (T^{2}GR):
  for $i = 1, \dots, N$ **do**
    Calculate the $\hat{\epsilon}_{tex}(x_t^i)$ by Eq. 10
    Obtain the texture-conditioned noise estimation $\epsilon_{tex}(x_t^i|\hat{U}_{0,t}^N)$ by Eq. 13
    Combine the texture-conditioned noise estimation $\epsilon_{tex}(x_t^i|\hat{U}_{0,t}^N)$, text-conditioned noise estimation $\epsilon_\theta(x_t^i|c)$, and unconditioned noise estimation $\epsilon_\theta(x_t^i|\varnothing)$ to calculate the final noise estimation $\epsilon_m(x_t^i)$ by Eq. 14
    if $t > 1$ **then**
      Substitute $\epsilon_\theta(x_t^i)$ with $\epsilon_m(x_t^i)$ in Eq. 1 and Eq. 2 to calculate the x_{t-1}^i for the next denoising step
    end
  end
end

Methods	Inference Time (minutes) ↓
TEXTure	3.94
Text2Tex	20.64
Fantasia3D	109.67
ProlificDreamer	483.92
Ours	46.83

Table 4. Inference time of compared methods using images with resolution of 512×512 on a single NVIDIA Tesla V100 GPU.

CVPR
#4556

Figure 10. More texture generation results of our proposed method.

Figure 11. Visual comparison of our proposed method against TEXTure [27] and Text2Tex [6].

Figure 12. Visual comparison of our proposed method against Fantasia3D [7] and ProlificDreamer [34].

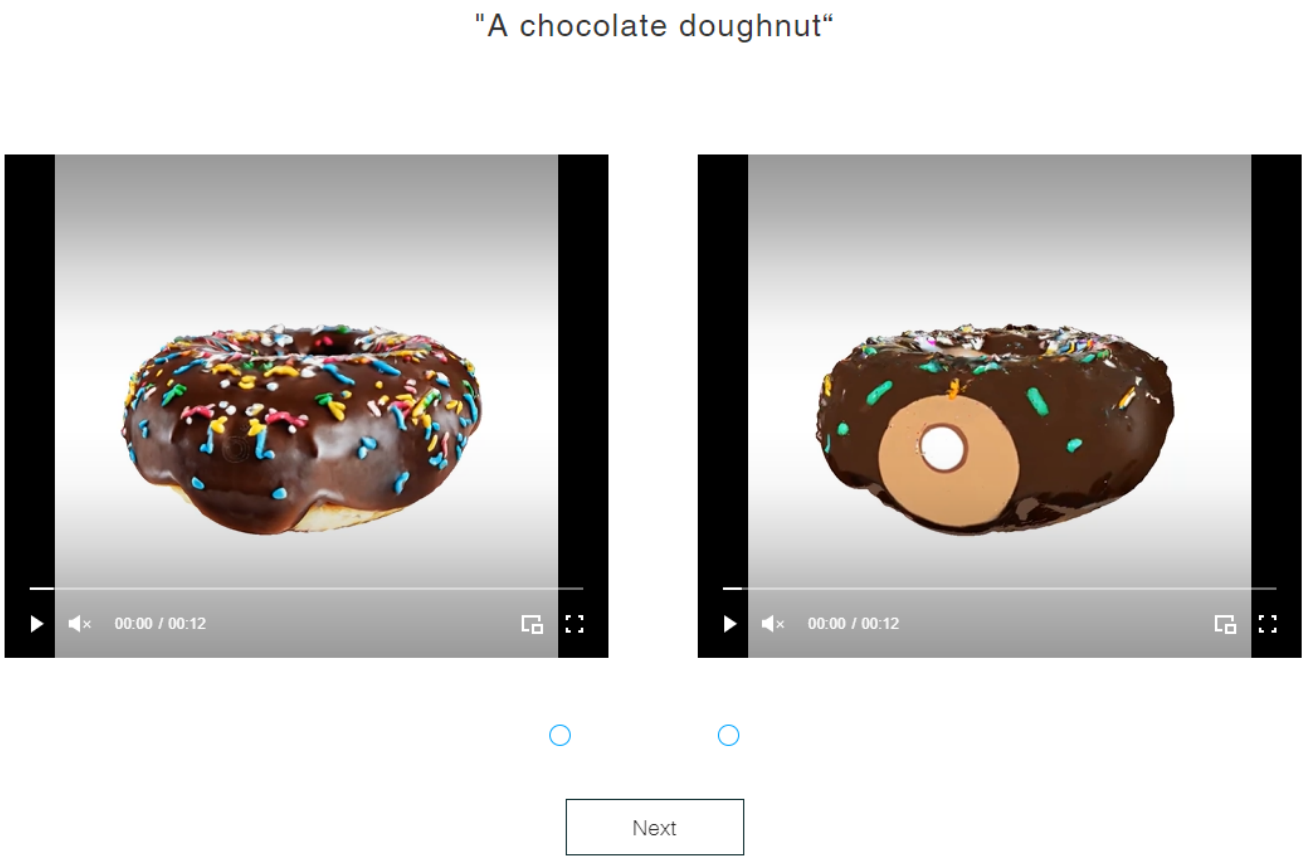


Figure 13. Screenshot of the user study web application

Object	Source	Description	Textual Prompts
eb219212147f4d84b88f8e103af8ea10	Objaverse	frog	"A robotic frog" "A green frog"
a8813ea1e0ce47ab97a416637a7520d7	Objaverse	helmet	"A Mandalorian helmet in silver" "A black helmet"
e0417d1e05984727a50f9ab1451d162d	Objaverse	lantern	"A stone lantern" "A medieval lantern"
9fa2da2c42234b58896e8d23393cac24	Objaverse	backpack	"A backpack in ironman style" "A backpack in spiderman style" "A 3D backpack"
a51751c9989940e592eb61be41ee35cc	Objaverse	baby owl	"A baby owl with fluffy wings" "A toy owl"
f73e2e1c8ad241ff859aca7e032ec262	Objaverse	lion	"A cute 3D cartoon lion with brown hair" "A marble lion"
91c5283b27c74583900d5e26e2fcd086	Objaverse	mug	"A wooden mug surrounded by silver rings" "A mug with cloud"
b6db59bd7f10424eae54c71d19663a65	Objaverse	car	"A next gen nascar in red" "A next gen nascar"
a2832b845e4e4edd9d439342cf4fd590	Objaverse	wolf	"Statue of a wolf" "A white wolf"
b19ef2650b4347348710eb6364ca90bd	Objaverse	penguin	"A black penguin" "A penguin covered by a blue sweater"
bd384d46514548cf8c4202f1ae6ea551	Objaverse	refrigerator	"A wooden refrigerator" "A high tech refrigerator"
f1aa479977a74a608d362679ed5ca721	Objaverse	piano	"A medieval piano" "A piano with flowers"
4c4690ba918f477b829990dd2e960c21	Objaverse	lion	"A golden lion" "A cyber punk lion"
f87caf6ac5a445ccad1a97653688e16e	Objaverse	dresser	"A wooden dresser" "A marble dresser"
f15298421b3d4e0fab4c43863a7e72fd	Objaverse	shark	"A deep ocean shark" "A dark blue shark"
d4c560493a0846c5943f3aeea58acb72	Objaverse	soccer ball	"A soccer ball in black and white" "A stone soccer ball"
c6509a8fe1f44a5eac8aebe12be2699e	Objaverse	tiger	"A tiger walking on the grass" "A plastic toy tiger"
bff537fb09b641c59b2ad123da0ca3dc	Objaverse	turtle	"A metal turtle with red eyes" "A sea turtle"
d726514a97f74f168b104fd6ba538331	Objaverse	vase	"An ancient vase" "A painted vase"
01ab0842feb1448bb18e8c7b85326d11	Objaverse	pottery	"An antique pottery" "A pottery with flowers"
f2d31eb0ddac4d21944df7dcc4af6d28	Objaverse	vending machine	"A coca cola vending machine" "A silver vending machine"

Table 5. Description of 3D Meshes in our collected data.

Object	Source	Description	Textual Prompts
e1f96691aaf648b885d927f5c3f5be61	Objaverse	apple	"A red apple" "An oil painted apple"
8a60954eccad433e987bbcafc7657140	Objaverse	armor	"A medieval armor" "A Japanese armor"
f98c5ee54c4a48f8b5eafd35a81dde4d	Objaverse	owl	"A metal owl with glowing eyes" "A wooden owl"
fadefc1eee3246a189f6b79c7c671343	Objaverse	lion	"A lion looking forward" "Statue of a lion"
9a0c52d350634e419aaf0eea1e67d9da	Objaverse	knight	"A golden knight" "A silver knight"
fa2c41a7a6c84fcb871a24016fa9a932	Objaverse	doughnut	"A chocolate doughnut" "An icecream doughnut"
f05b0c2f9bcf41cea188a4b4c848068a	Objaverse	fireplug	"A fireplug, red and yellow" "A fireplug with yellow top"
0db114d7753344d6825aa4f21ec56db9	Objaverse	crate	"A wooden crate" "A bronze crate"
72826cd5c17a42798a8e8e36c05c5035	Objaverse	clock	"A medieval clock" "A electric clock"
ac5df73de2c54239833643423a152592	Objaverse	dresser	"A wooden dresser" "A marble dresser"
90009fa6fa0b4d4bb1a1203431954097	Objaverse	keg	"A metal keg in silver" "A wooden keg"
b26a53419075442ca284cdf1d5541765	Objaverse	monitor	"A mac monitor" "An ironman monitor"
f75caead1dc1474195eb32a7f4c71117	Objaverse	control	"A game controller with black buttons on the top" "A PS5 controller"
edbeb81ef32645cea8bef89338f7e213	Objaverse	telephone	"A telephone with golden dials" "A classic telephone"
fc9cc06615084298b4c0c0a02244f356	Objaverse	piano	"A medieval piano" "A piano with flowers"
7adc9c74b75e4860b0a51c850bde9957	Objaverse	dress	"A princess dress" "A dress with spider patterns"
2fc0fc6ebe564a249c4617e6b3e6da93	Objaverse	fireplace	"A brick fireplace" "A stone fireplace"
14b8ae60eae240ff8bf1abdf9af5e49c	Objaverse	refrigerator	"A wooden refrigerator" "A high tech refrigerator"
62897c52e967469c85df9c6abdd09d16	Objaverse	doll	"A doll with yellow hairs" "A spiderman doll"
6f5480698a7a43c7a8c0a8b1e295e4a0	Objaverse	pumpkin	"A pumpkin with red eyes" "A Halloween pumpkin"

Table 6. Description of 3D Meshes in our collected data.

Object	Source	Description	Textual Prompts
Napoleon ler	ThreeDScans	statue	"A high quality color photo of Tom Cruise" "A high quality color photo of Benedict Cumberbatch" "A high quality color photo of Robert Downey Jr."
Plastic Dragon	ThreeDScans	statue	"Cartoon dragon, red and green" "A 3D dragon"
Francois	ThreeDScans	statue	"Spiderman with white hairs" "A boy in suits"
Provost	ThreeDScans	statue	"Portrait of Provost, oil paint" "A statue of Provost"

Table 7. Description of 3D Meshes in our collected data.