

# Meta 3D TextureGen: Fast and Consistent Texture Generation for 3D Objects

Raphael Bensadoun\*, Yanir Kleiman\*, Idan Azuri, Omri Harosh,  
Andrea Vedaldi, Natalia Neverova, Oran Gafni

GenAI, Meta

{raphaelbens, yanirk, idanazuri, omrih, vedaldi, nneverova, oran}@meta.com



Figure 1. Meta 3D TextureGen: examples of generated textures. Given a 3D shape and a textual prompt, our method generates globally consistent, high-quality textures in under 20 seconds, while maintaining text faithfulness for both realistic and stylized text prompts.

## Abstract

The recent availability and adaptability of text-to-image models has sparked a new era in many related domains that benefit from the learned text priors as well as high-quality and fast generation capabilities, one of which is texture generation for 3D objects. Although recent texture generation methods achieve impressive results by using text-to-image networks, the combination of global consistency, quality, and speed, which is crucial for advancing texture generation to real-world applications, remains elusive.

To that end, we introduce Meta 3D TextureGen: a new feedforward method comprised of two sequential networks aimed at generating high-quality and globally consistent textures for arbitrary geometries of any complexity degree in less than 20 seconds. Our method achieves state-of-the-art results in quality and speed by conditioning a text-to-image model on 3D semantics in 2D space and fusing them into a complete and high-resolution UV texture map, as demonstrated by extensive qualitative and quantitative evaluations. In addition, we introduce a texture enhancement network that is capable of up-scaling any texture by an arbitrary ratio, producing 4k pixel resolution textures.

\*Equal contribution

# 1. Introduction

3D generative models have advanced considerably, in part thanks to the impressive progress in text-to-image [13, 16, 43, 44, 46, 47] and text-to-video [17, 22, 52] generation. These advances concern three related fronts: (i) generation of 3D shapes, including the development of new and powerful shape representations [1, 10, 37, 51, 62], (ii) generation of textures [6, 8, 34, 45]; and (iii) combined generation of shape and texture, often called ‘text-to-3D’ [25, 26, 40, 49, 57]. As new shape representations usually include appearance information too, areas (i) and (iii) are converging. However, texture generation remains important, as it allows to control appearance independently of shape, and is applicable to any 3D asset, whether produced by an artist or generated automatically.

“Moonlight is sculpture; sunlight is painting”. After the subtleties of geometry, textures and colors add a remarkable layer of expressiveness, as implied in this famous quote by Nathaniel Hawthorne [19]. Creating textures is a key mode of expression for 3D artists and crucial to the impact of 3D content in applications such as gaming, animation, and virtual/mixed reality. However, creating high-quality and diverse textures, whether realistic or stylized, is difficult and time-consuming, particularly for complex 3D shapes, and requires specific professional skills.

Contrary to image and video generation, where billions of images and videos are available for training, 3D generation is hampered by the lack of large-scale 3D datasets. For this reason, 3D generation networks, including texture generation, are often *derived* from pre-trained image or video generation networks. This allows texture generators to inherit some of the qualities of their peers, including realism, faithfulness and open-ended nature, while only utilizing a comparatively small amount of 3D training data. However, there are still significant quality and speed gaps between texture and 2D image and video generation:

**(i) Global consistency and text faithfulness.** The gap between the image-text relationship when generating a single image compared to generating a sequence of images or views, translates to a lack of global consistency and text faithfulness in the generated texture. This is further intensified by the strong bias of text-to-image models towards frontal views, as well as their lack of 3D understanding. These inconsistencies range from small texture misalignments (often referred to as “seams”), to a lack of symmetry or an overall incoherent look, to catastrophic failures such as the “Janus effect” [59], where multiple instances of a given anatomical feature (e.g. a face or an eye) appear in multiple places across the object.

**(ii) Semantic alignment with the target 3D shape.** The text-to-image model is required to generate texture that fits the given 3D object, and must thus be *conditioned* on its shape. However, fusing fine 3D shape information into 2D

space in a coherent manner, such that fine 3D information is preserved yet translated efficiently to 2D space is difficult to achieve. Previous attempts generated texture by either conditioning in UV space on vertex or normal maps [66], or in image space, on depth maps [67]. However, they struggle with precise alignment and fine-detail preservation, resulting in lower texture quality for highly detailed 3D objects, which is a considerable limitation.

**(iii) Inference speed.** While previous methods rely on iterative generation for improving global consistency and gaining complete shape coverage, they require multiple generation steps, ranging from several to thousands of forward passes, such as via Score Distillation Sampling (SDS) [40]. This results in a long inference time of minutes, which is compute intensive and renders these methods unsuitable for many practical use cases, such as user-generated content applications, or allowing designers to perform quick iterations as part of their creative process.

We introduce Meta 3D TextureGen, a new texture generation method that successfully addresses these gaps, while attaining state-of-the-art results. Our method is fast, as it only requires a single forward pass over two diffusion processes. The method achieves excellent view and shape consistency, as well as text fidelity, by conditioning the first fine-tuned text-to-image model on 2D renders of 3D features, and generating all texture views jointly, accounting for their statistical dependencies and effectively eliminating global consistency issues such as the Janus problem.

The second image-to-image network operates in UV space, it creates a high-quality output by completing missing information, removing residual artifacts, and enhancing the effective resolution, bringing our generated textures to being close to application-ready. Moreover, we introduce an additional network that enhances the texture quality and increases resolution by an arbitrary ratio, effectively achieving a 4k pixel resolution for the generated textures.

To the best of our knowledge, this is the first approach to achieve high quality and diverse texturing of arbitrary meshes using merely two diffusion-based processes, without resorting to costly interleaved rendering or optimization-based stages. Moreover, this is the first work to explicit condition networks on geometry in 2D, such as position and normal renders in order to encourage local and global consistency, finally alleviating the Janus effect.

Samples of our generated textures are provided on a diverse set of shapes and prompts throughout the paper, as well as on static and animated shapes in the [video](#).

## 2. Related work

### 2.1. Image generation

A number of architectures have been proposed for text-to-image synthesis, including earlier efforts using Generative

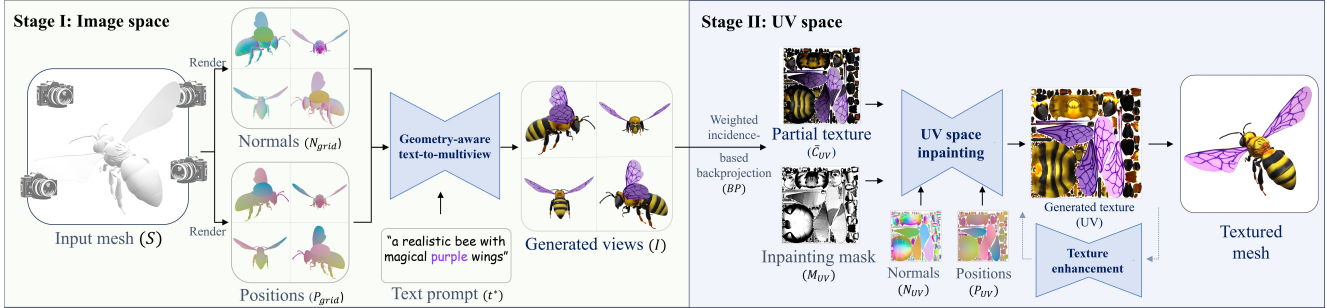


Figure 2. **Method overview.** Given an input shape and a text prompt, Meta 3D TextureGen generates a globally consistent high-quality texture in less than 20 seconds. The first stage (left) consists of a geometry-aware text-to-image model that generates a multi-view image of the generated texture, conditioned on renders of the normal and position maps over the input mesh. The second stage (right) consists of a projection of the generated texture renders back to UV space while taking into account the normals and camera angles (weighted incidence). The combined backprojections are then fed into the UV-space inpainting network along with a guiding inpainting mask, as well as the vertex and position UV maps, which generates a complete texture map in UV space. The generated texture map can optionally go through a MultiDiffusion texture enhancement network to increase the resolution by an arbitrary ratio.

Adversarial Networks [18]. Some more recent variants are based on transformers (e.g. DALL-E [42], CogView [15], Make-a-Scene [16], Parti [65]) and Muse [7]). Another popular class of text-to-image generators builds on pixel-space or latent diffusion models [21], including eDiff-I [2], Imagen [47], unCLIP [44], Stable Diffusion [46], SDXL [39], EMU [13] and others. In this work, we are starting with a pre-trained latent diffusion model with an architecture similar to EMU [13] and further extend it to our task.

## 2.2. Multi-view generation

The field of multi-view generation, which involves the generation of multiple perspectives of a single object or scene from noise or a few reference images, has demonstrated its utility in the generation of 3D shapes. Zero-1-to-3 [28] and Consistent-1-to-3 [63] generate novel views through viewpoint-conditioned diffusion model. Zero123++ [48], MVDream [49] and Instant 3D [25] opt for a grid-like generation of six and four views respectively. ConsistNet [61] use a different diffusion process for each view and introduce a 3D pooling mechanism to share information between views. Additional layers and architectures to enhance multi-view consistency are proposed by SyncDreamer [29], Consistent123 [58], DMV3D [60] and MVDiffusion++ [54] which denoise multiple views of the 3D object simultaneously. The obtained multi-view images in these works are then utilized as guidance to reconstruct the texture and geometry of a 3D object.

In contrast to our task of texture generation, these models are designed for the generation of 3D objects, where the geometry is not predetermined and is concurrently produced with the texture. This application inherently provides the flexibility to modify the geometry to achieve more consistent multi-view images, for both texture and geometry.

## 2.3. Texture generation

Texture generation aims to create high-quality and realistic or stylized textures for 3D objects based on textual descriptions. Early works, such as CLIP-Mesh [36] and Text2Mesh [35] proposed to optimize a texture via differentiable rendering, using CLIP [41] guidance to match the text prompt. Other optimization-based methods such as Fantasia3D [9], Latent-Paint [34] and Paint-It [64], combine differentiable rendering with SDS [40] to utilize gradients from diffusion models. Texturify [50] and Mesh2Tex [5] opt for a GAN-based approach incorporating a latent texture code and a mapping network similarly to StyleGAN [23]. The rapid emergence of large-scale text-to-image models, particularly diffusion models, has led to several advancements in texture generation. Several methods, such as TexDreamer [31] and Geometry Aware Texturing [11] aim to generate a UV map in a straight-forward manner, applying the diffusion process directly in UV space. While these methods tend to be fast, they are limited to human texture generation and clothing items respectively, and cannot generalize to arbitrary objects. Point-UV Diffusion [66] proposes a point-cloud diffusion approach to generate a colored point-cloud, which colors are subsequently projected onto the UV map for further refinement, yet requires to train a separate model for each object category, and does not generalize to arbitrary objects.

A significant area of work, which includes TEXTure [45], Text2Tex [8], Intex [53] and Paint3D [67], consists of iterative inpainting using pre-trained depth-to-image diffusion models in a zero-shot manner. This involves generating a single view at a time and iteratively rotating the mesh until a sufficient area is covered, using interleaved renderings as guidance for further inpainting steps. While these approaches are training-free, their inference

runtime is significant and can take a few minutes for a single generated texture. Moreover, they are not 3D-aware and are prone to producing artifacts such as the “Janus” effect. SyncMVD [30] adopted the same zero-shot approach while employing different diffusion processes for each view and synchronizing the output at each step, leading to better quality textures, yet suffering from the same global consistency issues. TexFusion [6] alleviates consistency issues by adding a module which performs denoising diffusion iterations in multiple camera views and aggregates them through a latent texture map after every denoising step. FlashTex [14], similarly to our approach, trains on a 3D dataset and generates a four-view grid. As conditioning, they use renderings of the shape with three different materials, which are then combined into a single three-channel image. Subsequently, they use an SDS optimization-based stage to distill information from their trained multi-view model, resulting in a significant runtime of 2 minutes. Meshy [33], a commercial product for which we do not have the complete technical details, tends to produce better results quality-wise than some methods mentioned above. Yet, their textures exhibit global inconsistencies, as well as over-saturated colors, text alignment issues and blurred inpainting of self occlusions.

### 3. Preliminaries and data processing

Our method takes a representation of the 3D shape features in the form of rendered images and baked texture maps in UV space, which are used in the first and second stage respectively. Here we detail the different channels that we render for each shape.

#### 3.1. Shape renders

We render the following channels for each shape. Each channel is rendered from four views which are stitched to a single image.

**Combined pass.** As ground truth data used for training, which is not extracted at inference time, we render the shape with all material properties. This render, often referred to as “beauty pass”, preserves lighting effects and material properties that are applied to the object. These are crucial to preserve to correctly represent different types of materials such as wood, plastic, metal, etc., which react differently to light and thus cannot be represented faithfully using only their diffuse color. We use Blender [12] to render the combined pass with even lighting from all directions.

**Position and normal passes.** These are used as conditioning for training and inference. Each pixel in the position pass represents the XYZ position of the corresponding point on the shape, and each pixel in the normal pass represents the normal direction of the shape at the corresponding point. Both are normalized to the range  $[0, 1]$  and rendered without lighting, hence written as-is to the output image.

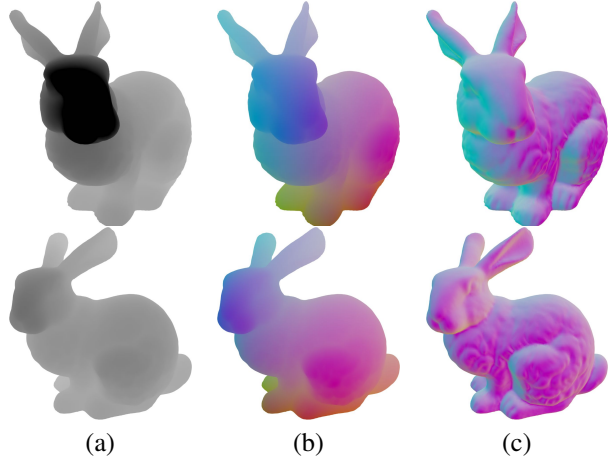


Figure 3. Contrary to (a) depth renders, (b) position renders are global rather than view-dependent, and (c) normal renders contain high-frequency details.

#### 3.2. UV maps

We bake each channel into a texture in UV space. This process involves producing a UV layout for each shape and baking the texture to an image.

**UV layout.** Our in-house dataset contains objects from various sources which may have various UV layouts, from layouts meticulously created by an artist, to scanned objects with a procedurally generated layout, and objects with partial or corrupt UV layout. A single object may contain many texture files, in which case the UV layout of each part may overlap the layout of parts that are mapped to a different texture. For our method, we require a UV layout that maps the shape onto a single square texture with no overlapping UV islands, so we automatically rearrange the UV islands of the shape such that there is no overlap between them. For objects that do not have a suitable UV map, we generate a new UV map using Blender’s *Smart Project* feature, and filter out objects for which this process fails to produce a desirable UV layout.

**Baked channels.** We use Blender to bake the *combined*, *position*, and *normal* passes mentioned above to the UV space. Baking a texture is a similar process to rendering an object, but the rendered pixel are written to the corresponding location on the UV map rather than being painted in the render view. The *combined* pass is used as the target image for training, while the *position* and *normal* passes are used as conditioning for the network.

**Backprojected textures.** To simulate the input textures that are produced by the first stage, we take the color renders of the shape and project them onto the texture in UV space, using the same process as described in Sec. 4.2.1. The network goal is to reconstruct the full texture map from these partial views.



Figure 4. **Qualitative comparison with previous work (local consistency, quality and text alignment).** Compared with previous work, our method results in higher-quality textures, while preserving local consistency and adhering to the text prompt.

## 4. Method

Given a 3D object and description of a desired texture, Meta 3D TextureGen produces as output a corresponding texture in UV space. As shown in Fig. 2, Meta 3D TextureGen employs a two-stage approach. The first stage operates in image space, conditioned on a text description and renders of the 3D shape features, and produces renders of the textured shape from multiple views. The second stage operates in UV space, taking a weighted incidence-based backprojection of the first stage output as condition, as well as the 3D shape features used for the first stage, but in UV space. The end result of the second stage is a complete UV texture map which is consistent between different views and matches the text prompt. An optional extension of the second stage is a texture enhancement network that extends the MultiDiffusion [3] approach from 1D to 2D image-patch overlaps, increasing the texture map resolution by  $\times 4$ .

As demonstrated in our experiments (Sec. 5), by conditioning the fine-tuned text-to-image model on renders of 3D shape features while generating all views in tandem, the first stage is able to generate diverse yet globally consistent renders of textured 3D shapes, while the second stage focuses on generating the missing areas that are occluded in image space and improving the overall quality of the generated texture map.

Next, we provide a detailed overview of each stage. We focus here on the novel or unusual aspects of our method

and refer the reader to the supplement for details.

### 4.1. Stage I: Generation in image space

The goal of the first stage is to generate globally consistent images of a given 3D object based on a textual description of the desired output. To this end, we use a diffusion-based neural network fine-tuned from a pre-trained image generator. In order to produce consistent views that match the given 3D object, the network takes as input a grid of position and normal renders from multiple angles, in addition to the text conditioning. Specifically, for each channel we produce a grid of 4 matching viewpoints and combine them to a single image. The four viewpoints are fixed at training and inference time, and provide a 360° view of the object at 90° intervals, with a fixed elevation angle of 20°.

#### 4.1.1 Geometry-aware 2D conditioning.

Multiple methods [8, 66, 67] use depth maps as a way to represent 3D assets in 2D images leveraging depth-conditioned pre-trained diffusion models in a zero-shot manner. In contrast, we advocate for the use of position and normal renders.

As seen in Fig. 3, the additional information in these representations provides the following benefits for using them as conditioning: (i) position values are global and not view-dependent, providing point correspondence between

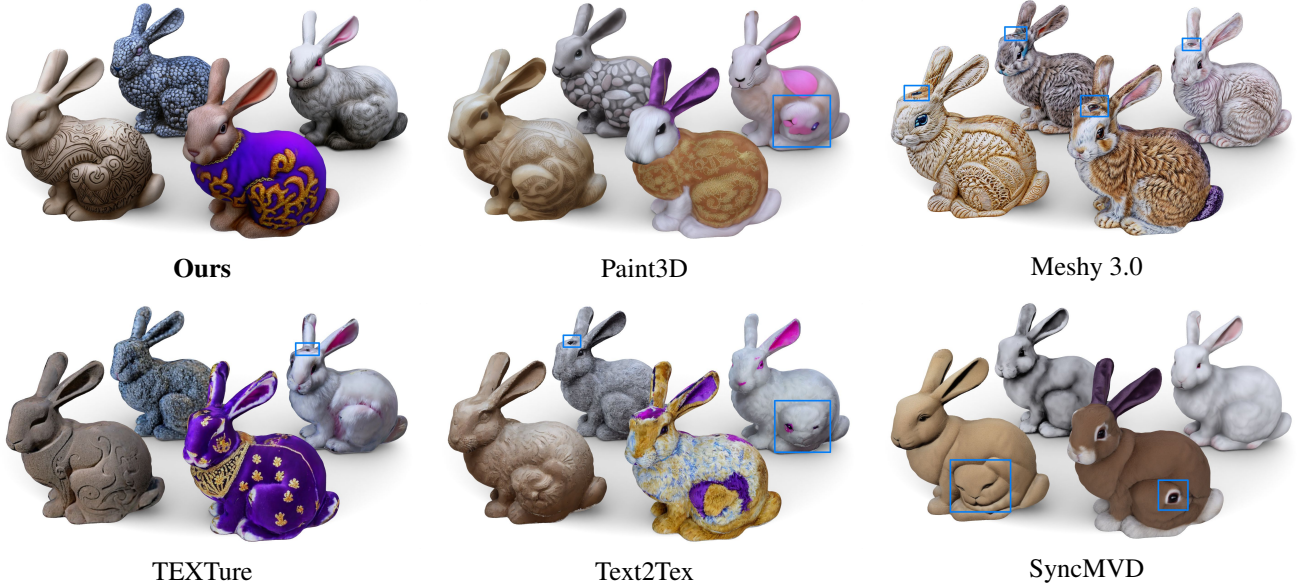


Figure 5. **Qualitative comparison with previous work (global consistency, quality and text alignment).** While previous methods result in global inconsistencies such as the Janus effect (blue rectangles), as well as text mis-alignments, our method returns a globally consistent and highly text-aligned textures. Text prompts: (i) **top-left**: “a bunny made out of *small pebbles* of many shades of gray”, (ii) **top-right**: “a realistic white rabbit with long fur, *pink eyes*, and *black paws*”, (iii) **bottom-left**: “a *sand sculpture* of a bunny with engraving of an intricate pattern”, (iv) **bottom-right**: “a bunny with a *velvet purple coat* with *intricate gold embroidery* along the edges”.

the same points on the object in different views, thus encouraging 3D consistency; (ii) normal renders provide orientation information and fine geometric details of the mesh to guide the generation model, which can be difficult to capture with depth.

#### 4.1.2 Multi-view image generation from text

The first stage consists of a U-Net based latent diffusion model, fine-tuned from a model with a similar architecture to Emu [13] denoted by  $f$ . Its goal is to generate a grid of four consistent views of an arbitrary mesh  $S$ , in image space, guided by a text prompt  $t^*$ , denoted by  $I$ . For this purpose, the diffusion model is conditioned on two grids of matching position and normal renders, denoted as  $P_{\text{grid}}(S)$  and  $N_{\text{grid}}(S)$  respectively.

The generated multi-view image grid  $I$  can then be formulated as follows:

$$I(S, t^*) = f(z, t^*, P_{\text{grid}}(S), N_{\text{grid}}(S)), \quad (1)$$

where  $z$  is 2D noise map where each pixel is sampled i.i.d. from a standard Gaussian distribution. Note that in this equation  $t^*$  denotes the textual prompt; in practice, this network is also conditioned on the diffusion step, sometime called ‘time’. We do not show it here explicitly for succinctness and clarity.

## 4.2. Stage II: Generation in UV space

The goal of the second stage is to generate the final texture in UV space. Given the viewpoints from the first stage output, the network aims at inpainting missing areas due to self occlusions and improving the overall quality of the generated texture, in UV space. The inputs for the second stage are the partial texture map, obtained by backprojecting and blending the views generated by the first stage, in addition to the position and normal UV maps.

### 4.2.1 Backprojection and incidence-based weighted blending

Backprojection is a technique where a 2D image or projection is mapped onto the UV texture map of a 3D model. This involves identifying the corresponding face on the 3D model for each non-background pixel in the image and assigning the color value at the corresponding coordinate in the texture map.

Although the first stage results in highly consistent views of the generated texture due to the conditioning on 3D semantics, we have observed, similarly to previous works [8, 30, 66], that textures generated over areas that are not facing the camera (low incidence angles) are less reliable. This can lead to artifacts when naïvely averaging different texture views together, particularly in areas with high frequency details such as fine patterns or writings. To overcome this

issue, similarly to SyncMVD [30], we blend the backprojections into a single UV map using a weighted average by the incidence angles. Specifically, we utilize the cosine similarity between the viewing direction and per-pixel normal vectors in image space to determine per-pixel weight contributions to the blended texture. Formally, the incidence of a pixel  $p$  in a rendering  $I_S^i$  of a 3D shape  $S$ , for each view  $i$  (which we denote by  $\phi(I_S^i, p)$ ) is defined as  $\phi(I_S^i, p) = \cos(\theta_{\vec{v}_i(p), \vec{n}(I_S^i, p)})$  where  $\theta_{\vec{x}, \vec{y}}$  is the angle between  $\vec{x}$  and  $\vec{y}$ ,  $\vec{v}_i(p)$  is the viewing direction from camera  $i$  to pixel  $p$ , and  $\vec{n}(I_S^i, p)$  is the normal vector at pixel  $p$  of the rendered shape  $S$  from camera  $i$ .

Finally, denoting the backprojection operation as BP, we define each pixel  $p$  of the blended partial texture  $\bar{C}_{UV}(S, t^*)$  as follows:

$$\bar{C}_{UV}^p(S, t^*) = \frac{\sum_{j=0}^n \text{BP}(I(S, t^*)_j^p) \odot \text{BP}(\phi(I(S, t^*)_j, p)^\alpha)}{\sum_{j=0}^n \text{BP}(\phi(I(S, t^*)_j, p)^\alpha) + \epsilon}, \quad (2)$$

where  $I(S, t^*)_j$  is the  $j$ 'th view of  $I(S, t^*)$  and  $I(S, t^*)_j^p$  is the pixel  $p$  of  $I(S, t^*)_j$ .  $\epsilon$  is a small constant to avoid zero division. We use  $n = 4$ , as the number of generated views and  $\alpha = 6$  for all of our experiments.

#### 4.2.2 UV-space inpainting network

The first stage followed by the weighted backprojection operator results in a texture map that is sparse in a varying degree, depending on the input shape. The degree of sparsity is determined by two factors: (i) occlusions caused by insufficient coverage of the selected views in respect to the shape structure, resulting in missing areas, and (ii) pixel-level ‘‘holes’’ resulting from the absence of one-to-one correspondence between each occupied pixel in the generated rendering and the UV map. To obtain the full texture, we opt for an inpainting approach.

Similarly to Stage I, the inpainting is modeled by a U-Net based latent diffusion model fine-tuned from the same pre-trained network which we denote by  $g$ .  $g$  is conditioned on the blended partial map  $\bar{C}_{UV}(S, t^*)$ , the inpainting mask denoting the missing areas and pixels to inpaint  $M_{UV}(S)$ , along with  $P_{UV}(S)$  and  $N_{UV}(S)$ , to obtain the final texture map  $\text{Texture}(S, t^*)$  as follows:

$$\text{Texture}(S, t^*) = g(z, \bar{C}_{UV}(S, t^*), M_{UV}(S), P_{UV}(S), N_{UV}(S)), \quad (3)$$

where  $z$  is a 2D noise map where each pixel is sampled i.i.d. from a standard Gaussian distribution.

#### 4.2.3 Texture enhancement network

Our two-stage texture generation approach yields a text-aligned, high-quality and consistent UV texture map at a resolution of  $1024 \times 1024$  pixels. While this resolution is

Table 1. Quantitative comparison with previous work. We evaluate the win-rate of our method in terms of better representation of the prompt and fewer artifacts compared with previous methods, as well as FID, KID ( $\times 10^{-3}$ ), and runtime. Overall, our textures were preferable over all baselines. The quantitative metrics show that we achieve better visual fidelity on this task of texturing artist-made assets.

	Preference	Artifacts	FID↓	KID↓	Runtime
TEXTure	78.5%	76.5%	91.4	8.4	90s
Text2Tex	81.9%	84.2%	92.1	6.9	287s
SyncMVD	67.4%	66.7%	77.7	3.8	81s
Paint3D	78.9%	79.5%	86.1	5.2	66s
Meshy 3.0	64.5%	68.4%	99.7	10.7	85s*
<b>Ours</b>	-	-	<b>73.0</b>	<b>3.6</b>	<b>19s</b>

satisfying for some applications, other applications may require a higher resolution of 4k ( $4096 \times 4096$ ) pixels. To that end, we introduce an additional, yet optional component to the second stage for up-scaling the generated texture map resolution and quality. This is the texture enhancement network, which is flexible in terms of the output resolution and ratio, as it operates in a patched-based fashion.

The reason for employing a patch-based approach [38, 55] is due to the memory limitations of current GPUs that do not support the generation of 4k resolution images. As patch-based prediction results in inconsistencies between different patches, manifesting both locally (seams) and globally as pattern/color mismatches, we extend the MultiDiffusion [3] approach from 1D image-patch overlaps to 2D (panoramas to square-shaped images) to mitigate these issues, aggregating the different latent patches and applying a weighted Gaussian average at each diffusion time step. In addition, we employ a tiled-VAE approach for the encoder-decoder to enable the encoding and decoding of high-resolution textures.

## 5. Experiments

We evaluate our method in comparison to state-of-the-art previous work, namely TEXTure [45], Text2tex [8], SyncMVD [30], Paint3D [67], and the commercial product Meshy 3.0 [33]. Our method achieves state-of-the-art results according to user studies and numerical metric comparisons. Samples supporting the qualitative advantage are provided in Figs. 4 and 5, while quantitative comparisons are provided in Tab. 1. Additionally, we provide a qualitative ablation study in Fig. 8 to better assess the effects of different contributions. Diverse sets of generated samples are provided in Fig. 1, Fig. 7, and Fig. 10, as well as in the appendix, including animated samples in the [video](#).

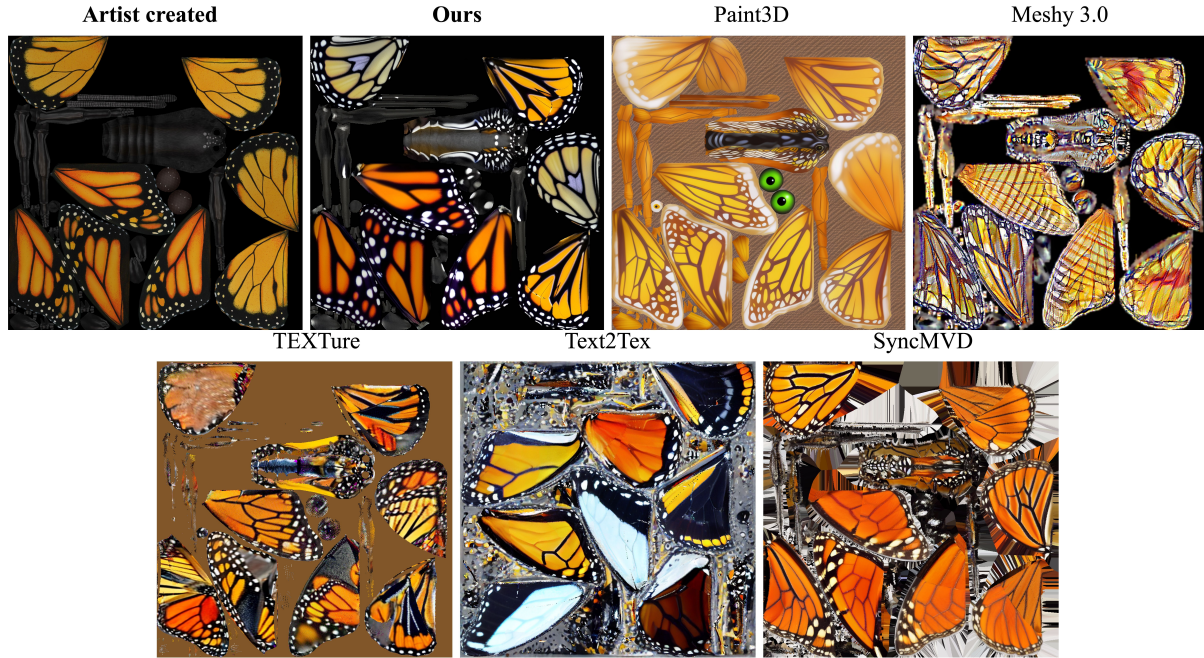


Figure 6. **Qualitative comparison with previous work (texture UV maps).** Our method produces a texture map which is cleaner and closer to the artist generated texture, making it more usable as part of the creative process.

## 5.1. Data

### 5.1.1 Training data

Our dataset consists of 260k textured 3D objects sourced from an in-house collection. Text captions are extracted for each object similarly to Cap3D [32].

### 5.1.2 Evaluation data

To evaluate our methods and the baselines quantitatively and qualitatively, we use a set of 54 objects with CC license that do not have a ‘No-AI’ tag from the Sketchfab website. In addition, we use 2 objects from the Stanford 3D Scanning Repository. For each 3D object, we provide 4 creative text prompts for generations, which we use for our user study. Additionally, we provide a single text prompt describing the original texture of each object, which is necessary for evaluation using metrics such as FID (Frechet Inception Distance) [20] and KID (Kernel Inception Distance) [4]. The complete list of objects and prompts is provided in the supplementary. Additionally, in all qualitative and quantitative comparisons, we do not employ the texture enhancement network in order to allow for fair comparison in terms of resolution, as both our method and the baselines generate texture maps at a resolution of  $1024 \times 1024$ .

## 5.2. Quantitative comparisons

In order to quantitatively evaluate our method, we employ the FID and KID metrics. These metrics aim at evaluat-

ing the quality of the generated textures. Furthermore, we conduct a user study to evaluate how well the generated texture represent the objects in terms of visual quality and text alignment, as well as the presence of artifacts.

### 5.2.1 User study

For the user study, we rendered  $360^\circ$  rotation videos of the generated textured meshes from our evaluation set. In each question we present two videos side-by-side, one generated by our model and another generated by one of the baselines, along with the text prompt used to generate the textures. The order of meshes, prompts and baselines are randomized, as well as the left-right ordering of the baseline and our method in order to eliminate bias. Similarly to [8], participants were asked to choose which object best represents the given prompt. This question captures both text alignment and overall visual quality, as textures of low quality do not represent the desired object well. In addition, we ask which object displays fewer visual artifacts to capture cases in which an object is generally of better quality, e.g., more detailed or realistic, but includes some errors or inconsistencies. The decision for each texture is determined by max-voting. An example question screenshot is provided in the supplementary. 33 users participated in the study, with 754 responses. A breakdown comparing our method with the baselines (see Sec. 2.3) can be seen in Tab. 1. Overall, our method was preferred over all baselines, both in terms of overall quality and when considering artifacts.



Figure 7. **Diversity of prompts.** Our method enables generating diverse prompts ranging from realistic to extremely fantastical creations. Here we show 33 different textures on the same llama model, and 11 textures for a voxel model for which the text prompts emphasize creation of low poly assets.

### 5.2.2 Metrics

For the FID and KID calculations, we render the ground-truth textured meshes and the generated textured meshes from 32 evenly spaced viewpoints under identical conditions, the standard image FID and KID scores are then calculated between these two sets of rendered images. For runtime, we compare inference time for each method, where we define inference time as the time it takes to generate a complete texture map for a given text prompt and pre-defined mesh. Even though we report faster runtime for the baselines compared with the numbers reported in the original papers, we emphasize that we could not run them in the same exact setup as ours (single H100 vs. A100 GPU), which should translate to some reduction in runtime. However, given our method’s advantage of not running multiple generation iterations, combined with the significant runtime difference, we expect that our method would be the fastest when running on the same GPU.

### 5.3. Qualitative comparisons

We provide several qualitative comparisons with previous work, focusing on different aspects of visual quality: text fidelity, global consistency, local consistency, and texture

map usability. Fig. 4 emphasizes the challenge in adhering to the text prompt while generating visually pleasing and geometrically coherent textures (e.g. style of Van Gogh, specifically texturing the shell as green, as well as fine-details of the armadillo’s face). Fig. 5 focuses on global consistency, where the Janus effect can be seen clearly in all baselines as additional sets of eyes or faces, as well as text fidelity, where previous methods struggle in maintaining alignment. Finally, UV texture maps generated by the different methods are illustrated to assess the potential usability of these maps, while comparing them with the original artist-created map in Fig. 6.

### 5.4. Ablation study

In order to assess the importance of different contributions to our method, we provide an ablation study in Fig. 8. We compare five cases: (a) excluding the first stage (no image space), (b) excluding the second stage (no UV space), (c) excluding the weighted-incidence blending (simple averaging), (d) our method without texture enhancement (SR), and (e) our method with texture enhancement.

**Omitting stage I** (no image space). In this scenario we fine-tuned a diffusion model that operates in UV space exclusively, similarly to the second stage. However, we omit the partial texture and inpainting mask conditioning and

\*Runtime for Meshy is estimated using the Meshy 3.0 API.

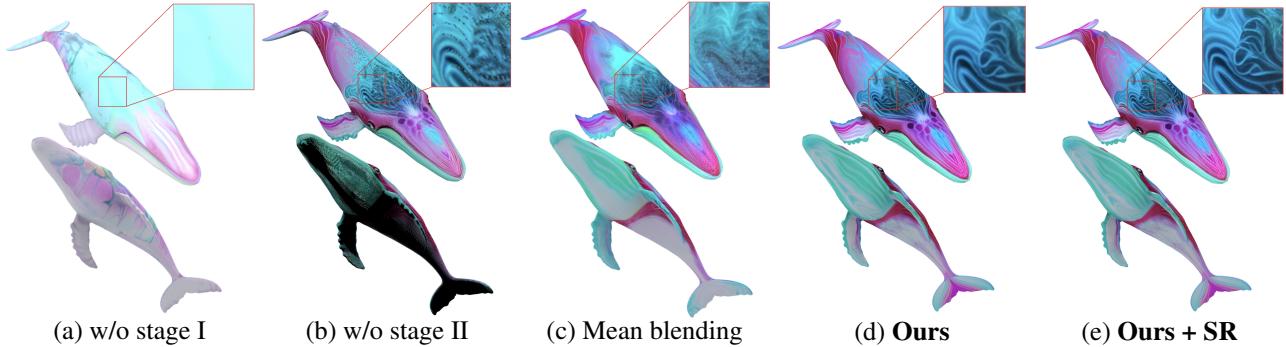


Figure 8. **Qualitative ablation results** for the text prompt “A whale with a pastel pink skin with swirls of mint green, lavender and blue creating a marbled effect”. Five scenarios are evaluated: **(a)** omitting stage I (no image space), **(b)** omitting stage II (no UV space), **(c)** backprojection average blending, **(d)** our result, and **(e)** our result with the texture enhancement network.

provide only position and normal UV maps ( $P_{UV}, N_{UV}$ ) as visual conditions. We additionally enable text conditioning for guidance. This setup proved to be challenging for a standard diffusion model, as it struggled to capture the 3D semantics presented in the exclusive form of UV maps. This resulted in generated textures that exhibit text-alignment issues, especially for non-global prompts, as well as significant local consistency issues (“seam”) appearing at the boundaries of UV fragments.

**Omitting stage II** (no UV space). Next, we directly evaluate the generated output of stage I after backprojection. In most cases four views are insufficient to cover an entire 3D object, resulting in several “unpainted” areas. Furthermore, we observed that the quality of the backprojected texture is inferior to that of the full method. This suggests that our UV-space stage not only inpaints the occluded areas, but also refines the existing areas of the partial texture and mitigates backprojection artifacts, thereby enhancing its quality and effective resolution.

**Average blending.** Lastly, we evaluate a straightforward averaging approach to merge the generated views, as opposed to using a weighted incidence-based blending technique. This results in a final output with blurry areas while lacking fine details.

## 6. Limitations

The generation of PBR material maps, such as tangent normals, metallic and roughness are not covered by this method, and are left as future work. Although Meta 3D TextureGen is currently the fastest method for texture generation, it is not real-time nor fast enough to cover all possible applications. However, the introduction of recent methods of speeding-up text-to-image models, such as Imagine-Flash [24], could directly translate into real-time texture generation, given that the bottlenecks are the text-to-image forward passes. While training on 3D datasets is crucial for achieving global consistency, the reliance on 3D datasets is

somewhat limiting for training large models compared with the size of image and video datasets.

## 7. Ethical considerations

The application of generative methods in general extends to a wide range of use cases, many of which are not covered in this work. Before implementing these methods in real-world scenarios, it is crucial to thoroughly examine the data, model, its potential uses, as well as considerations of safety, risk, bias, and societal impact. In the specific case of texture generation, the limitations of the existing shape provide some risk mitigation, as users would be bound to a pre-defined structure.

## 8. Conclusions

We introduce Meta 3D TextureGen, a new method for texturing 3D objects from text descriptions. While there has been impressive progress in this domain, our method brings texture generation to be significantly closer to an applicable tool for 3D artists and general users to create diverse textures for assets in gaming, animation and VR/MR. This is done by providing global consistency (e.g. eliminating the Janus problem), strong control (adherence to text prompts), speed, and high-resolution (4k) to the generation process.

## 9. Acknowledgements

We are grateful for the instrumental support of the multiple collaborators at Meta who helped us in this work. Emilien Garreau, Ali Thabet, Albert Pumarola, Markos Georgopoulos, Jonas Kohler, Filippos Kokkinos, Yawar Siddiqui, Uriel Singer, Lior Yariv, Amit Zohar, Yaron Lipman, Itai Gat, Ishan Misra, Mannat Singh, Zijian He, Jialiang Wang, Roshan Sumbaly. We thank Ahmad Al-Dahle and Manohar Paluri for their support.

## References

- [1] Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. Polydiff: Generating 3d polygonal meshes with diffusion models. *arXiv preprint arXiv:2312.11417*, 2023. [2](#)
- [2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhng, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. In *arXiv preprint arXiv:2211.01324*, 2022. [3](#)
- [3] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023. [5](#), [7](#)
- [4] Mikolaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. [8](#)
- [5] Alexey Bokhovkin, Shubham Tulsiani, and Angela Dai. Mesh2tex: Generating mesh textures from image queries. *arXiv preprint arXiv:2304.05868*, 2023. [3](#)
- [6] Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. Textfusion: 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](#), [4](#)
- [7] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. [3](#)
- [8] 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](#), [7](#), [8](#)
- [9] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22246–22256, 2023. [3](#)
- [10] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 45–54, 2020. [2](#)
- [11] Evgeniia Cheskidova, Aleksandr Arganai, Daniel-Ionut Rancea, and Olaf Haag. Geometry aware texturing. In *SIGGRAPH Asia 2023 Posters*, New York, NY, USA, 2023. Association for Computing Machinery. [3](#)
- [12] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2024. [4](#)
- [13] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. [2](#), [3](#), [6](#)
- [14] Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. Flashtex: Fast relightable mesh texturing with lightcontrolnet. *arXiv preprint arXiv:2402.13251*, 2024. [4](#)
- [15] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34, 2021. [3](#)
- [16] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. [2](#), [3](#)
- [17] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. [2](#)
- [18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. [3](#)
- [19] Nathaniel Hawthorne. *Passages from the American notebooks of Nathaniel Hawthorne*. Houghton, Mifflin, 1896. [2](#)
- [20] 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. [8](#)
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. [3](#)
- [22] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. [2](#)
- [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. [3](#)
- [24] Jonas Kohler, Albert Pumarola, Edgar Schönfeld, Artsiom Sanakoyeu, Roshan Sumbaly, Peter Vajda, and Ali Thabet. Imagine flash: Accelerating emu diffusion models with backward distillation. *arXiv preprint arXiv:2405.05224*, 2024. [10](#)
- [25] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023. [2](#), [3](#)
- [26] 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. 2
- [27] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5404–5411, 2024. 14
- [28] 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. 3
- [29] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 3
- [30] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. *arXiv preprint arXiv:2311.12891*, 2023. 4, 6, 7
- [31] Yufei Liu, Junwei Zhu, Junshu Tang, Shijie Zhang, Jiangning Zhang, Weijian Cao, Chengjie Wang, Yunsheng Wu, and Dongjin Huang. Texdreamer: Towards zero-shot high-fidelity 3d human texture generation. *arXiv preprint arXiv:2403.12906*, 2024. 3
- [32] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *Advances in Neural Information Processing Systems*, 36, 2024. 8
- [33] Meshy. Meshy 3.0. <https://docs.meshy.ai/>, 2024. Accessed: 2024-05-01. 4, 7
- [34] 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, 3
- [35] 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
- [36] 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
- [37] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020. 2
- [38] Ozan Özdenizci and Robert Legenstein. Restoring vision in adverse weather conditions with patch-based denoising diffusion models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 7
- [39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Romach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *arXiv preprint arXiv:2307.01952*, 2023. 3
- [40] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 3
- [41] 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. 3
- [42] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation (*ICML spotlight*), 2021. 3
- [43] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2
- [44] 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. 2, 3
- [45] 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, 7
- [46] 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. 2, 3
- [47] 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. 2, 3
- [48] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 3
- [49] 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, 3
- [50] 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
- [51] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. *arXiv preprint arXiv:2311.15475*, 2023. 2
- [52] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

- [53] Jiaxiang Tang, Ruijie Lu, Xiaokang Chen, Xiang Wen, Gang Zeng, and Ziwei Liu. Intex: Interactive text-to-texture synthesis via unified depth-aware inpainting. *arXiv preprint arXiv:2403.11878*, 2024. [3](#)
- [54] Shitao Tang, Jiacheng Chen, Dilin Wang, Chengzhou Tang, Fuyang Zhang, Yuchen Fan, Vikas Chandra, Yasutaka Furukawa, and Rakesh Ranjan. Mvdifffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. *arXiv preprint arXiv:2402.12712*, 2024. [3](#)
- [55] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023. [7](#)
- [56] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*. [14](#)
- [57] 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. *Advances in Neural Information Processing Systems*, 36, 2024. [2](#)
- [58] Haohan Weng, Tianyu Yang, Jianan Wang, Yu Li, Tong Zhang, CL Chen, and Lei Zhang. Consistent123: Improve consistency for one image to 3d object synthesis. *arXiv preprint arXiv:2310.08092*, 2023. [3](#)
- [59] Wikipedia. Janus — wikipedia, the free encyclopedia, 2024. [2024]. [2](#)
- [60] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023. [3](#)
- [61] Jiayu Yang, Ziang Cheng, Yunfei Duan, Pan Ji, and Hongdong Li. Consistnet: Enforcing 3d consistency for multi-view images diffusion. *arXiv preprint arXiv:2310.10343*, 2023. [3](#)
- [62] Lior Yariv, Omri Puny, Natalia Neverova, Oran Gafni, and Yaron Lipman. Mosaic-sdf for 3d generative models. *arXiv preprint arXiv:2312.09222*, 2023. [2](#)
- [63] Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. *arXiv preprint arXiv:2310.03020*, 2023. [3](#)
- [64] Kim Youwang, Tae-Hyun Oh, and Gerard Pons-Moll. Paint-it: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. *arXiv preprint arXiv:2312.11360*, 2023. [3](#)
- [65] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wangt, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Karagol Burcu Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. [3](#)
- [66] 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](#), [5](#), [6](#)
- [67] Xianfang Zeng. Paint3d: Paint anything 3d with lighting-less texture diffusion models. *arXiv preprint arXiv:2312.13913*, 2023. [2](#), [3](#), [5](#), [7](#)

## A. Additional implementation details

### A.1. Training details

All of the models presented in the manuscript have a similar architecture, and are fine-tuned from the same base text-to-image generation model that operates at a resolution of  $1024 \times 1024$ . Their multiple conditionings are encoded via the original image encoder matching to our base model and are concatenated altogether via channel-wise concatenation. To adapt the architecture to these new inputs, we simply add the relevant number of additional channels as zero-weighted input channels for the first convolution layer. The text-to-multiview network (Stage I) was fine-tuned to minimize the L2 loss, and both the UV space inpainting (Stage II) and texture enhancement networks to minimize the L1 loss. We use v-prediction formulation where the noise schedule was rescaled to enforce zero terminal SNR [27]. We empirically found that the latter is beneficial when training diffusion models on renderings and UV maps, which possess large background areas, such as rendering background and un-mapped pixels for UV maps.

We fine-tune all of our models with a learning rate of  $1e-5$  and a batch size of 256 on 32 H100 gpus. Stage I and Stage II models were fine-tuned for 15k steps each and the texture enhancement model was trained for 28k steps. For stage I and stage II we employ DDPM solver and use 60 diffusion steps for inference. For the multi-diffusion texture enhancement we employ DDIM solver with 50 diffusion steps.

### A.2. Texture enhancement model training pipeline

Our training pipeline for the diffusion model enhances image quality by addressing artifacts and upscaling the texture by an arbitrary ratio. The design of our upsampler draws inspiration from the widely utilized open-source Real-ESRGAN framework [56]. Despite its effectiveness, Real-ESRGAN’s degradations often produce artifacts such as over-smoothed textures, excessively sharpened edges, and patterns with high contrast, leading to noticeable ringing effects. We have noticed that our method does not exhibit these issues. Besides changing the architecture to a diffusion model and training on high quality texture maps, we modified the data degradation pipeline to empirically better match our needs, omitting the Unsharp Masking operation as well as the additive Gaussian noise. Our patch-based approach, followed by Multi-Diffusion blending, allows us to upsample an image by an arbitrary ratio, without introducing seams or noticeable artifacts between the patches. Moreover, we employ a tiled-VAE approach in order to overcome memory issues arising from encoding and decoding large images and latent maps. These choices resulted in a robust upsampler, tailor-made for upsampling texture maps to a very high resolution.

Table 2. Visualization dataset

Object	Ref.	Object	Ref.
Tree 1	<a href="#">Link</a>	Hot Air Balloon	<a href="#">Link</a>
Flower	<a href="#">Link</a>	Forest Skybox	<a href="#">Link</a>
Tree 2	<a href="#">Link</a>	Alien Skybox	<a href="#">Link</a>
Treasure Chest	<a href="#">Link</a>	Fairy Skybox	<a href="#">Link</a>
Treasure Chest 2	<a href="#">Link</a>	Swampland Skybox	<a href="#">Link</a>

## B. Experiments details

### B.1. Evaluation dataset

All meshes on our evaluation dataset were taken from Sketchfab, under [CC Attribution](#) license and respecting any NoAI requests by the artists. We present a list of all meshes, with credit to the artists, as well as the prompts we used, in Tabs. 4 to 9. Prompts not marked in bold were used during the user study, while those marked in bold were used for FID and KID calculation.

### B.2. Applications

The vast majority of texture generation evaluation is performed on a single asset detached from any environment (i.e. with a white background). While this is important for capturing fine details and artifacts, it lacks the broader context of a method’s ability to produce multiple assets that can blend in an environment, whether realistic or stylized, in a manner that is desirable for real-world applications. In addition to the single asset evaluations, we demonstrate the usability and applicability of our method in diverse real-world scenarios, utilizing it for building both realistic and stylized environments in virtual reality in Fig. 10 and the supplementary video.

### B.3. User study

We conducted a user study, presenting pair-wise comparisons between our method and five different baselines - TEXTure, Text2Tex, SyncMVD, Paint3D and Meshy on textured meshes. To eliminate biases, left-right ordering, as well as mesh, prompt and baseline orderings have all been randomized. A screenshot of the survey is shown in Fig. 14. Table 3 includes a breakdown of participants to different backgrounds, according to their familiarity and proficiency with 3D objects. Of the 33 participants we had in our study, 10 were 3D artists, 18 had some proficiency with 3D objects and 5 had no prior background.

## C. Visualization details

In addition to meshes used during evaluation, we made use of additional meshes and skyboxes for visualization purposes. These meshes are also under [CC Attribution](#) license, and can be found in Tab. 2.

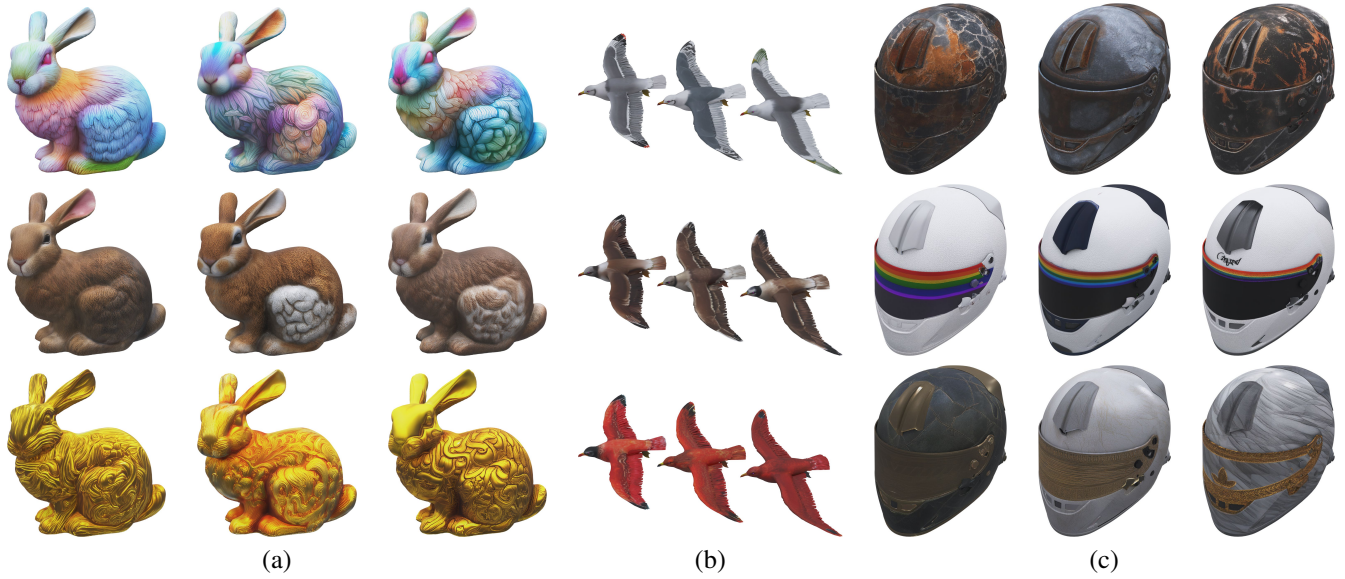


Figure 9. **Diverse samples.** For each column, each row was generated using the same prompt with a different seed.



Figure 10. Generated textures in realistic and stylized VR environments. Excluding the skybox (background), all textures are generated.

Table 3. Breakdown of user answers according to proficiency.

	3D Artists		Some Proficiency		No Background	
	Preference	Artifacts	Preference	Artifacts	Preference	Artifacts
TEXTure	78.3%	80.7%	77%	72.1%	62.5%	50%
Text2Tex	83.6%	89.6%	77.6%	75.5%	92.3%	92.3%
SyncMVD	64.3%	61.8%	67.2%	68.3%	84.6%	84.6%
Paint3D	87.3%	86.1%	68.3%	71.2%	76.9%	76.9%
Meshy3.0	69.1%	70.4%	56.3%	64%	80%	80%

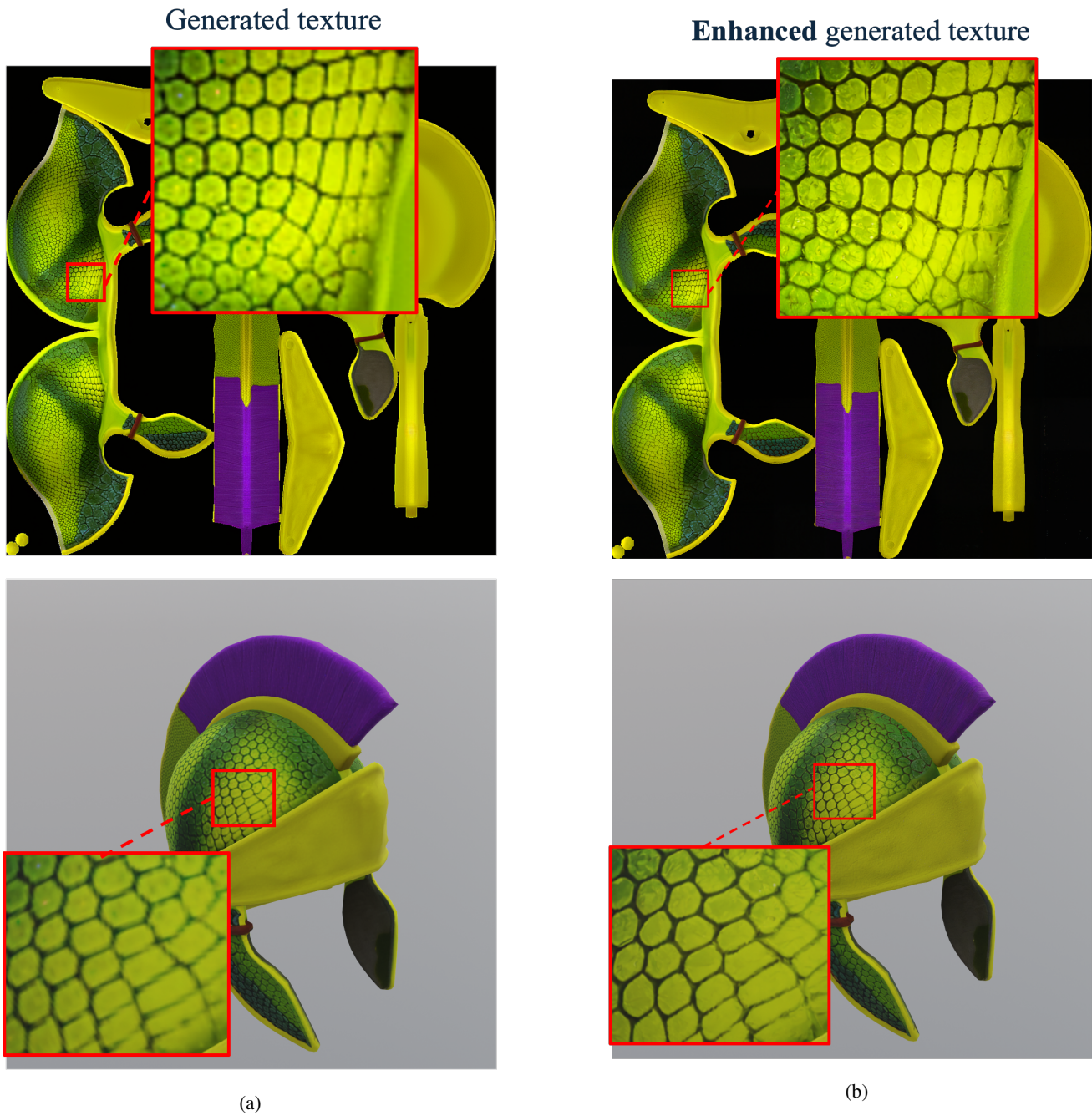
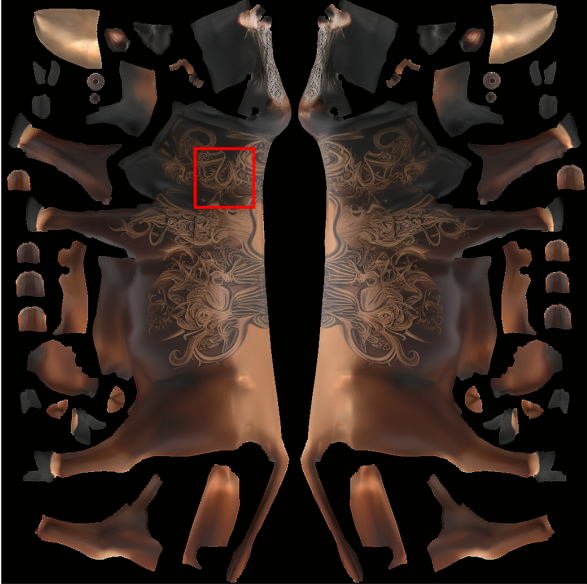
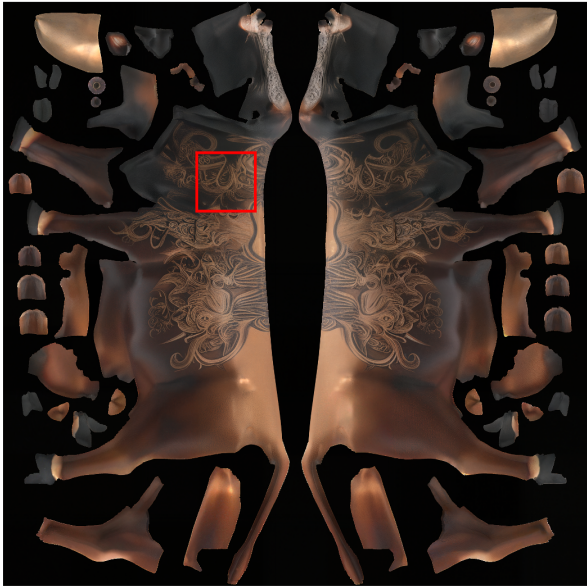


Figure 11. **Enhancing textures in UV space and 3D.** (a) Generated textures and (b) enhanced generated textures for the text prompt: “a yellow-green helmet made of snakeskin with a purple ruffle on top”. The top image represents a texture UV map, while the bottom image showcases a 3D render of the same patch from the UV space.

Generated texture



Enhanced generated texture



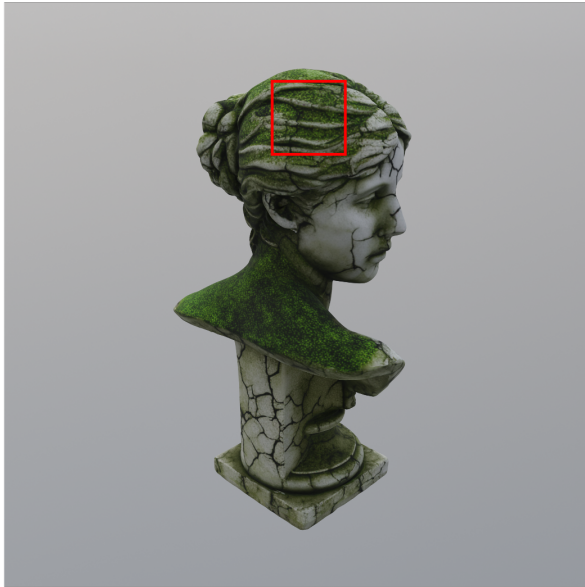
(a)



(b)

Figure 12. **Texture enhancement UV space.** (a) Generated textures and (b) enhanced generated textures for the text prompt: “a brown cow covered with an intricate tattoo”. The top row showcases these textures which, despite their initial high quality, have been further enhanced to reveal extremely fine details. The bottom row provides a closer look at these intricate details.

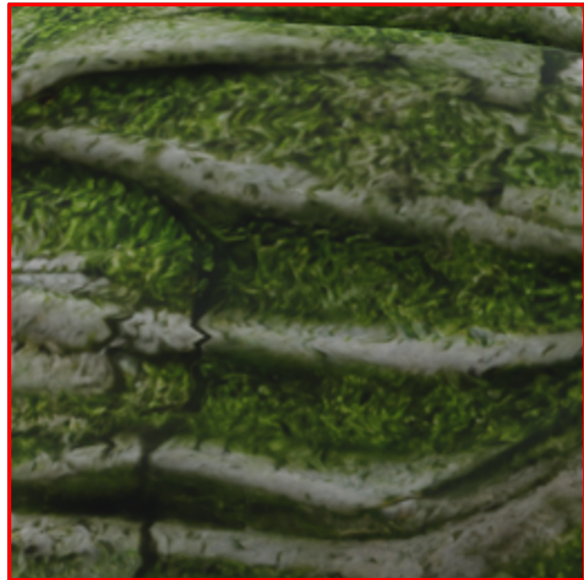
Generated texture



Enhanced generated texture



(a)



(b)

Figure 13. **Texture enhancement in 3D.** (a) Generated textures and (b) enhanced generated textures for the text prompt: “a moss-covered ancient statue made of cracked and semi-shattered stone”. The top row showcases these textures which, despite their initial high quality, have been further enhanced to reveal extremely fine details. The bottom row provides a closer look at these intricate details.

Add your name:

What is your background with 3D?

No prior experience

**Prompt: a fluffy fuzzy toy plushie bee**



FILL IN USERNAME

Which Object is a better representation of the prompt?

Left	Right
------	-------

Which Object displays fewer artifacts?

Left	Right
------	-------

Next
------

Figure 14. Screenshot from the user study screen.

Table 4. Evaluation dataset. **Bold** prompts were used for quantitative evaluation (continued in Tab. 5).

Object	Source	Prompts
Ant	<a href="#">Link</a>	a psychedelic colored ant a realistic fire ant an old ant robot made of rusted metal a radioactive green ant <b>a dark brown ant with light brown legs and black eyes</b>
Bee	<a href="#">Link</a>	rainbow striped bee a bee with a green snakeskin pattern on its body a fluffy fuzzy toy plushie bee a common honey bee <b>a black and yellow fuzzy bee with black eyes</b>
Bottle	<a href="#">Link</a>	a red wine bottle with a black label with a blue infinity logo a bottle filled with layers of colorful sand from the dead sea a bottle completely wrapped in old newspapers a beer bottle with GenAI written on the label
Boy Room	<a href="#">Link</a>	a diorama of a boy and a monster in a room made out of cardboard a dollhouse featuring a boy and a big monster papercraft diorama of a boy and a big monster, origami folding a boy wearing a red cape and a golden sword fighting a green swamp monster <b>an isometric green and gray room with a boy wearing a red cape fighting a purple-yellow monster.</b>
Bracelet	<a href="#">Link</a>	brown leather bracelet with steel studs, horses engraved on the leather a golden bracelet with precious stones inlaid around it a hand-worn communication device, with a led screen, buttons and lights bracelet made of rough weathered wood in a deep forest green color with visible wood grain <b>a dark brown leather bracelet with laces</b>
Burger	<a href="#">Link</a>	an ancient statue of a marble burger in greek or roman style, made of veined red, black, green marble a realistic burger with tomatoes, pickles and lettuce in a sesame bun a burger-shaped cake, made out of chocolate cake bun, candies and candy floss a simple wooden toy shaped like a burger, made out of natural oak wood <b>a succulent burger with poppy seeds on the bun, lettuce, cheese, tomatoes and pickles</b>
Bust	<a href="#">Link</a>	a painted greek sculpture, blonde hair, fair skin, red lipstick, bright blue eyes a moss covered ancient statue, made of cracked and semi-shattered stone a realistic bust of a heavily tattooed woman, with tribal tattoos covering her face and neck a sculpture of a woman painted in the style of Van Gogh <b>a marble sculpture bust of the woman Róza Loewenfeld</b>
Butterfly	<a href="#">Link</a>	a majestic monarch butterfly a magical butterfly, arcane sigils on its wings, sparkling glitter on its body a crochet butterfly toy from pastel pink and pastel green yarn venomous black butterfly with bright red and orange markings <b>a monarch butterfly with orange and yellow wings, white and yellow dots and a black body</b>
Cactus	<a href="#">Link</a>	alien blue cactus plant with red spikes and white flowers a realistic model of a cactus contemporary plastic statue of a green cactus with white geometric patterns on it cactus colored with vertical repeating stripes of purple, black and lavender <b>three green cartoonish cactus with red pink and yellow spikes</b>

Table 5. Evaluation dataset **Bold** prompts were used for quantitative evaluation (continued in Tab. 6).

Object	Reference	Prompts
Carriage	<a href="#">Link</a>	<p>a medieval wagon made out of light wood, with colorful striped orange and white awning</p> <p>a modern steel carriage with white fabric awning</p> <p>a mystical magician’s wagon, colored in purple with a yellow symbol of an eye painted on it</p> <p>a wagon made out of unprocessed wood, with awning made out of leaves, vines and branches</p> <p><b>a wooden carriage with white fabric and white cargo</b></p>
Cartoon Car	<a href="#">Link</a>	<p>a cartoon car in the style of 3D animation</p> <p>a futuristic space car</p> <p>an old rusted car for a post apocalyptic game</p> <p>a toy car made out of pastel colored plastic with red wheels</p> <p><b>a busted up cartoonish yellow car with some rust</b></p>
Cartoon Plane	<a href="#">Link</a>	<p>a realistic plane with blue wings and white body</p> <p>a futuristic robot car made of white and silver metal</p> <p>an old rusted car for a post apocalyptic game</p> <p>a toy car made out of blue and white plastic with red wheels</p> <p><b>a blue and white cartoonish light airplane</b></p>
Classic Car	<a href="#">Link</a>	<p>an old classic car made out of wood and brown leather</p> <p>a light blue old classic car</p> <p>a steampunk car lined with patches of leather</p> <p>a wooden carved car embossed with an intricate design</p> <p><b>a pale green vintage classic car</b></p>
Cow	<a href="#">Link</a>	<p>a cow made of rusted metal</p> <p>a realistic black and white holstein cow</p> <p>a brown cow covered with an intricate tattoo</p> <p>a white cow with golden horns and golden hooves</p> <p><b>a holstein cow with black and white spots and gray horns</b></p>
Cutlass	<a href="#">Link</a>	<p>toy sword made out of red and gold plastic</p> <p>a pirate sword with a bronze hilt and metallic blade, japanese kanji written across the blade</p> <p>a sword with a lattice of blue neon led-light across its blade</p> <p>a wooden sword with barnacles on the hilt</p> <p><b>a shiny metal sword with a golden blade wrapped with red tape</b></p>
Eclair	<a href="#">Link</a>	<p>a realistic raspberry eclair with cream and raspberries on top</p> <p>a realistic chocolate eclair with rainbow colored cream</p> <p>a yellow lemon and vanilla eclair</p> <p>a graphite drawing of an eclair with intricate shading patterns</p> <p><b>an eclair with pink whipped cream and raspberries</b></p>
Ender Dragon	<a href="#">Link</a>	<p>a realistic red dragon</p> <p>a steampunk dragon wearing a leather armor</p> <p>a cobalt dragon with lapis lazuli wings</p> <p>a stone statue of a majestic dragon decorated with jewelry</p> <p><b>a black shiny dragon with grayish wings</b></p>
Fish	<a href="#">Link</a>	<p>a white and orange koy fish</p> <p>a robotic koi fish made of orange metal</p> <p>an alabaster statue of a fish with delicate gold veins inlaid across its body</p> <p>a koi fish wearing bronze armor</p> <p><b>a white koi fish with red and orange spots</b></p>

Table 6. Evaluation dataset **Bold** prompts were used for quantitative evaluation (continued in Tab. 7).

Object	Reference	Prompts
Football Helmet	<a href="#">Link</a>	<p>a gamer vr football helmet, cyberpunk style with neon lights</p> <p>a football helmet with a mascot painting</p> <p>a rainbow football helmet</p> <p>a post apocalyptic rusty football helmet, with dirt, dust, and stains on it</p> <p><b>an old red football helmet with a white drawing of a creepy skull on the back</b></p>
Game Controller	<a href="#">Link</a>	<p>a game controller made out of wood with visible wood grain</p> <p>a game controller that looks like an exposed green circuit board</p> <p>an ancient stone statue of a game controller with colorful buttons</p> <p>a pink birthday cake in the shape of a game controller with chocolate buttons</p> <p><b>a white game controller</b></p>
Goggles	<a href="#">Link</a>	<p>goggles with butterflies on the strap</p> <p>steampunk goggles made out of brown leather and brass</p> <p>goggles with polka dots in the style of yayoi kusama</p> <p>goggles painted in the style of van gogh</p> <p><b>black goggles with silver rings</b></p>
Grapes	<a href="#">Link</a>	<p>a realistic cluster of red grapes</p> <p>magical glittering purple grapes, with gold dust on them</p> <p>crochet grapes, made of colorful thick wool yarn</p> <p>alien metallic grapes with strange engravings on them</p> <p><b>a bunch of black grapes</b></p>
Handbag	<a href="#">Link</a>	<p>a cute handbag with a whacky llama illustration on it</p> <p>an expensive pale leather handbag, high-end, high fashion</p> <p>a pink fluffy and fuzzy handbag with googly eyes</p> <p>a handbag made of bricks and metal</p> <p><b>a leather-like solid light beige handbag with gold-tone metal clasps</b></p>
Hovercraft	<a href="#">Link</a>	<p>an art deco hovercraft, with gold geometric patterns</p> <p>a neon cyberpunk hovercraft, in the style of japanese neon signs</p> <p>a desert hovercraft covered with camouflage nets</p> <p>a bright red metallic hovercraft</p> <p><b>a black futuristic hovercraft</b></p>
Ice Axe	<a href="#">Link</a>	<p>a magical axe with lightning powers set with blue gems</p> <p>an axe studded with diamonds and gems</p> <p>an old battle axe weathered with scratches and cracks and splattered with blood</p> <p>an axe carved out of a single piece of wood with intricate engraving. blade made of same wood as handle</p> <p><b>a cartoonish axe with a wooden handle and a gray blade with blue lightning</b></p>
Jellyfish	<a href="#">Link</a>	<p>neon colored purple and green jellyfish</p> <p>a robotic jellyfish made out of dark metal, with blue led lights across its tentacles</p> <p>a flying jellyfish with tentacles made out of delicate white and light blue feathers</p> <p>wooden carving of a jellyfish, unprocessed olive wood</p> <p><b>a pink and green jellyfish with intricate patterns</b></p>
Knight	<a href="#">Link</a>	<p>a knight wearing steel armor with a golden lion engraving on his chest</p> <p>a knight wearing armor with swirling rainbows on it</p> <p>a papercraft knight, origami with colorful papers covered by dots and geometric patterns</p> <p>a knight wearing samurai armor resembling a blooming sakura</p> <p><b>a knight dressed in full-body suit of armor</b></p>
Laptop	<a href="#">Link</a>	<p>a realistic laptop with a background picture of green fields and blue sky on its screen</p> <p>a laptop running a pixelated 80s style video game</p> <p>a laptop made entirely out of rock</p> <p>a gray laptop with a colorful keyboard and glowing keys</p> <p><b>a laptop showing an image of a city at night with a car, the laptop features mostly</b></p>

Table 7. Evaluation dataset **Bold** prompts were used for quantitative evaluation (continued in Tab. 8).

Object	Reference	Prompts
Mushrooms	<a href="#">Link</a>	<p>toadstool, red cap with white polka dots                      magical deep purple mushrooms, sparkling glitter on the cap                      cute cartoon mushrooms with a cute face, big eyes on the stalk                      realistic brown mushrooms  <b>dark red and orange mushrooms with white stems</b></p>
Ottomans	<a href="#">Link</a>	<p>3 wooden ottomans with portuguese azulejo patterned cushions                      colorful patchwork ottomans with elaborate wooden carvings on the bases                      ottomans made of dark wooden planks and deep scarlet velvet cushions                      Bamboo ottomans with simple bamboo cushion, ink paintings on the bases  <b>three wooden ottoman stools with floral patterns</b></p>
Potted Plant	<a href="#">Link</a>	<p>a realistic potted plant                      a realistic potted plant with colorful leaves                      a plant made out of colorful origami paper                      a potted plant made out of newspaper  <b>a simple potted plant with flat green leaves and a reddish pot</b></p>
Pterodactyl	<a href="#">Link</a>	<p>a white marble pterodactyl                      a steampunk pterodactyl wearing a leather armor                      a realistic pterodactyl with bat wings                      a pterodactyl made of rocks and lava  <b>a brown-yellow pterodactyl</b></p>
Roman Helmet	<a href="#">Link</a>	<p>a bronze metal helmet with vibrant red plume adorning it                      a crochet helmet in rainbow colors, a rainbow colored plume on top                      a yellow-green helmet made of snakeskin, with purple ruffle on top                      an old rusty cracked helmet  <b>a shiny roman corinthian helmet with a red plume adorned with intricate engravings and patterns</b></p>
Rose	<a href="#">Link</a>	<p>a red rose                      a rose with a pattern of pink-white stripes on its petals                      a rainbow colored rose with each petal leaf in a different color                      a magical blue rose with gold sparkling dusting  <b>a red rose with green leaves</b></p>
Row Boat	<a href="#">Link</a>	<p>a wooden red row boat, a black thick stripe at the bottom, a repeating pattern of shells across the body                      metallic row boat, made out of sheets of thin lightweight metal                      ancient egyptian boat with elaborate egyptian paintings and hieroglyphics painted on it                      a wooden boat made of untreated wood planks with visible wood grain  <b>an old wooden row boat</b></p>
Seahorse	<a href="#">Link</a>	<p>fantastic magical seahorse covered in sparkling gems                      a realistic yellow seahorse                      a seahorse wearing elaborate metal armor with engravings of sealife                      a futuristic robotic seahorse, with led lights and metal platings  <b>a yellow orange seahorse with orange-brown stripes and black eyes</b></p>

Table 8. Evaluation dataset **Bold** prompts were used for quantitative evaluation - continued in Tab. 9).

Object	Reference	Prompts
Shield	<a href="#">Link</a>	gold and blue high fantasy decorated shield black obsidian shield with a large ruby in its center shield made out of opaque yellow crystallized amber wooden shield with a painting of ouroboros on it <b>an elegant and elongated copper and blue shiny shield adorned with intricate designs and a blue gem stone in the middle</b>
Spider	<a href="#">Link</a>	a black widow spider with vibrant red markings on its back a monstrous flesh spider, sewn together from various parts a spider wearing an engraved bronze plated armor a pastel colored crochet spider <b>a scary black spider-like monster with a grayish-red bottom and yellow spikes coming out of the end of each leg</b>
Teacup	<a href="#">Link</a>	moroccan-style ceramic tea cup with arabesque patterns english ceramic tea cup with floral design, the painted flowers are pink brute concrete gray tea cup fine china tea cup, pink blue and green colors with design of a crane near a river <b>a white tea cup and coaster with blue intricate drawings</b>
Teapot	<a href="#">Link</a>	black ceramic teapot in the style of ancient greece, with red clay drawings of greek myths futuristic hi-tech smart teapot with led lights, sensors and a temperature measurement system a ceramic teapot with a intricate carvings on it a delicate floral teacup, vines and flowers spanning around the container <b>a grayish teapot with drawings of pink flower vines</b>
Telescope	<a href="#">Link</a>	a rusty brass telescope from the golden age of piracy a futuristic neon-lit spyglass, black chassis with neon pink and blue lights a steampunk spyglass made of leather an ancient moss covered rock telescope <b>an antique golden telescope with a wooden patch and black lenses</b>
Toad Hall	<a href="#">Link</a>	an alsatian timber house and a green frog wearing a black and white tux a frog wearing a white outfit in front of a santorini style white house with blue rooftop a brown toad wearing red near a scary halloween mansion a snow covered diorama of a house and a toad wearing a victorian outfit <b>a green toad with khaki clothes standing on a green grassy island with a wooden town and green trees</b>
Togo Cup	<a href="#">Link</a>	a togo coffee cup with a blue GenAI logo on it a old and dirty coffee cup with coffee stains a metal coffee cup that looks like a high-tech gadget a take-away coffee cup embossed with gold and precious gems <b>a brown take-away cup with white drawings of houses trees and windmills with a black top and a white bottom</b>
Toy Gun	<a href="#">Link</a>	a plastic toy gun made out of blue and orange plastic a high tech gun made out of metal and mechanical parts a toy gun with the word "GenAI" written on it a toy gun made out of wood with visible wood grain <b>a blue orange and black nerf gun with white markings</b>
Toy Plane	<a href="#">Link</a>	a toy plane made out of purple and yellow plastic with chrome propeller a toy plane made out of wood a realistic stunt biplane with red and blue stripes an old rusted war biplane <b>a purple cartoonish airplane with light yellow flame drawings</b>

Table 9. Evaluation dataset **Bold** prompts were used for quantitative evaluation - continued.

Object	Reference	Prompts
Tree Stump	<a href="#">Link</a>	<p>a light gray petrified wood on a dirt mound            snow covered tree stump during winter            a moss covered tree stump on fresh green grass            burnt wood on soot, scorch marks on the tree stump  <b>a wood stump covered with some snow laying on a snowy patch of ground</b></p>
Triceratops	<a href="#">Link</a>	<p>a hyperrealistic triceratops with pronounced grooves on its skin            a plush toy orange triceratops with white horns and white claws            a triceratops painted in the style of van gogh            a triceratops covered with tattoos of intricate designs  <b>a green triceratops with brown stripes on its back and yellow horns and nails</b></p>
Unicorn	<a href="#">Link</a>	<p>a pastel colored magical unicorn            a forest unicorn, covered with swirling blue shiny patterns amidst leaves covering its body            a unicorn with a snakeskin body in the colors of black and yellow            a nightmarish hell horse in the color of soot, fiery flaming mane and ruby eyes  <b>a white unicorn with black eyes and snout</b></p>
Vase	<a href="#">Link</a>	<p>an old vase made out of clay covered with ancient hieroglyphs            a priceless ming vase made out of white porcelain with a blue pattern            a black and red chinese vase with an intricate drawing            a cracked terracotta vase with various cracks  <b>a brown-orange base with some rust</b></p>
Watermelon	<a href="#">Link</a>	<p>a jelly cartoon watermelon            an autumn pumpkin pie            a slice of yellow lemon            a metallic round puzzle key with mysterious engravings on it  <b>a cartoonish black watermelon with green skin and black interior</b></p>
Wine Barrel	<a href="#">Link</a>	<p>a realistic wooden wine barrel with deep scarlet wine stains on them            a circus barrel, colored in bright simple colors with an image of a clown painted on it            a rusted metal barrel with graffiti on it            a dark red wooden barrel with steel hoops  <b>a wooden light colored wine barrel with metal stripes and brown base legs</b></p>
Wooden Crate	<a href="#">Link</a>	<p>an old wooden crate made out of rotten wood            a blue plastic crate covered with graffiti            an old rusted metal crate            a new wooden storage crate with chinese characters on it  <b>an old wooden crate with black markings</b></p>
Wooden Toy	<a href="#">Link</a>	<p><b>a wooden toy tower</b></p>
Stanford Bunny	<a href="#">Link</a>	<p>realistic white rabbit with long fur, pink eyes, and black paws            a bunny made out of small pebbles of many shades of gray            a futuristic bunny with dark fur and stripes of bright neon colors            a sand sculpture of a bunny with engraving of an intricate pattern</p>
Stanford Armadillo	<a href="#">Link</a>	<p>an armadillo man wearing a beautiful medieval armor crafted with gold and precious gems            an armadillo creature made out of a mosaic of small squares with red, blue, and white colors            a realistic armadillo creature with a shell like a green turtle on its back            an armadillo man action figure made out of pieces of brown and purple plastic with white claws and black eyes</p>