This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

MVPaint: Synchronized Multi-View Diffusion for Painting Anything 3D

Wei Cheng[1*†] Juncheng Mu[2,5*] Xianfang Zeng[1] Xin Chen[7] Anqi Pang[3]
Chi Zhang[6] Zhibin Wang[3] Bin Fu[3] Gang Yu[1] Ziwei Liu[4] Liang Pan[2‡]
[1] StepFun [2] Shanghai AI Laboratory [3] Tencent PCG [4] S-Lab, NTU
[5] Tsinghua University [6] AGI Lab, Westlake University [7] ByteDance
http://mvpaint.github.io



Figure 1. MVPaint generates **multi-view consistent** textures with **arbitrary UV unwrapping** and high **generation versatility**.

Abstract

*Texturing is a crucial step in the 3D asset production workflow, which enhances the visual appeal and diversity of 3D assets. Despite recent advancements in Text-to-Texture (T2T) generation, existing methods often yield subpar results, primarily due to local discontinuities, inconsistencies across multiple views, and heavy dependence on UV unwrapping outcomes. To tackle these challenges, we propose a novel generation-refinement 3D texturing framework called MVPaint, which can generate high-resolution, seamless textures while emphasizing multi-view consistency. MVPaint mainly consists of three key modules. **1) Synchronized Multi-view Generation (SMG).** Given a 3D mesh model, MVPaint first simultaneously generates multi-view images by employing a SMG model, which leads to coarse texturing results with unpainted parts due to missing observations. **2) Spatial-aware 3D Inpainting (S3I).** To ensure complete 3D texturing, we introduce the S3I method, specifically designed to texture previously unobserved areas effectively. **3) UV Refinement (UVR).** Furthermore, MVPaint employs a UVR module to improve the texture quality in the UV space, which first performs a UV-space Super-Resolution, followed by a Spatial-aware Seam-Smoothing algorithm for revising spatial texturing discontinuities caused by UV unwrapping. Moreover, we establish two T2T evaluation benchmarks: the Objaverse T2T benchmark and the GSO T2T benchmark, based on selected high-quality 3D meshes from the Objaverse dataset and the entire GSO dataset, respectively. Extensive experimental results demonstrate that MVPaint surpasses existing state-of-the-art methods. Notably, MVPaint could generate high-fidelity textures with minimal Janus issues and highly enhanced cross-view consistency.*

1. Introduction

3D texture generation remains a complex and critical aspect of asset creation, especially valuable in applications like gaming, animation, and virtual/augmented/mixed reality. Despite the scarcity of specialized 3D training data and the high computational demands of texture modeling, recent breakthroughs in text-to-image technologies [13, 15, 21, 41] have significantly advanced the field. These technologies facilitate Text-to-Texture (T2T) generation [5, 31, 40, 55] and their integration with 3D shapes [32, 37, 46], enhancing the visual diversity and realism of 3D models. However, achieving consistent and seamless textures across various viewing angles remains challenging, often hampered by local discontinuities and cross-view inconsistencies.

Recently, many texture generation methods have focused on leveraging 2D diffusion priors for guiding the generation process, which often utilizes conditional controls [57] (*e.g.*, depth) to produce more fitting textures. TEXTure [40] and Text2Tex [5] sample a series of camera viewpoints for iteratively rendering depth maps, which are then used to generate high-quality images through a pre-trained depth-to-image diffusion model. To avoid inconsistent textures due to using multiple independent generation processes, SyncMVD [31] introduced a method combining multi-view single-step denoising with UV space synchronization. However, its reused attention process is limited to nearby views, which frequently leads to Janus problems. Paint3D [55] developed a coarse-to-fine texture generation strategy, starting with a coarse texture obtained by iteratively painting from camera viewpoints, followed by inpainting and super-resolution in UV space. Similarly, Meta 3D TexGen [3] also employs UV position maps for UV inpainting and enhancement. Despite achieving remarkable 3D texturing results, both tools [3, 55] depend heavily on continuous mesh UV unwrapping. Discontinuities in texture often arise in scenarios where the UV atlases are randomly packed within the UV images. Consequently, many challenges remain in 3D texturing: **Multi-View Consistency**: Ensuring consistency across multiple viewpoints to prevent local style discontinuities and the presence of numerous seams. **Diverse Texture Details**: Avoiding overly smooth textures that lack detail, while aiming for high-resolution outputs. **UV Unwrapping Robustness**: Developing a method that does not rely heavily on UV unwrapping results to achieve robust automated generation.

To address these challenges, we propose MVPaint, a coarse-to-fine 3D texture generation framework capable of producing high-fidelity, seamless 3D textures while ensuring multi-view consistency and reducing dependence on UV unwrapping quality. MVPaint mainly consists of three stages for texture generation. (1) First, we employ the **Synchronized Multi-view Generation** (SMG) model that uses a multi-view diffusion model with cross-attention [48] and UV synchronization to initiate 3D texture generation conditioned on a given textural instruction, which effectively avoids the Janus problem and produces highly consistent multi-view images at low resolution. Following, we upsample and refine the coarse multi-view images by adding vivid texture details, subsequently projecting them into UV space (1K resolution) for further enhancement. (2) Second, we propose the **Spatial-aware 3D Inpainting** (S3I) method to ensure complete 3D texturing, particularly for areas that were not observed in the first stage. Specifically, S3I resolves the inpainting process in 3D space by considering the spatial relations among 3D points uniformly sampled from mesh surfaces. (3) Third, we introduce a **UV Refinement** (UVR) module, comprising a series of tailored texture enhancement operations in UV space. UVR first performs super-resolution to upscale the UV map. Afterward, we introduce a Spatial-aware Seam-smoothing Algorithm (SSA) to revise spatial discontinuous textures, especially for repairing the seams caused by UV unwrapping. Consequently, high-quality 3D UV textures could be obtained.

To facilitate the evaluation of T2T generation, we establish two benchmarks: the Objaverse [10] T2T benchmark and the GSO [14] T2T benchmark. The first benchmark comprises 1000 high-quality 3D meshes curated from the Objaverse dataset. Given that most T2T models are trained on a subset of the Objaverse dataset, we further establish the GSO T2T benchmark, which leverages all 1032 3D models from the GSO dataset to assess the generalizability of T2T models. For each 3D mesh, textual annotations are generated utilizing a large language model. Extensive experimental results on the Objaverse [10] and the GSO [14] T2T benchmarks demonstrate that MVPaint outperforms existing State-of-The-Art (SoTA) methods for 3D texture generation. We would like to emphasize that MVPaint is a robust 3D texturing method, significantly reducing occurrences of failed generations, such as missing areas, large inconsistencies, over-smoothness, and Janus issues. Qualitative texturing results of MVPaint could be visualized in Fig. 1.

In summary, our contributions are three-fold: **1)** We propose a robust 3D texturing framework, MVPaint, for generating diverse, high-quality and seamless 3D textures while ensuring multi-view consistency. **2)** We propose and study various 3D texturing models, operations, and strategies, including SMG, S3I, and UVR modules, which we believe will significantly advance future research in 3D texture generation. **3)** We conduct extensive experiments on the Objaverse and the GSO T2T benchmark, demonstrating that MVPaint achieves impressive 3D texture generation results, surpassing existing SoTA methods.

2. Related Work

Multi-View Generation. The generation of coherent multi-view images from diverse inputs, such as text, images, and

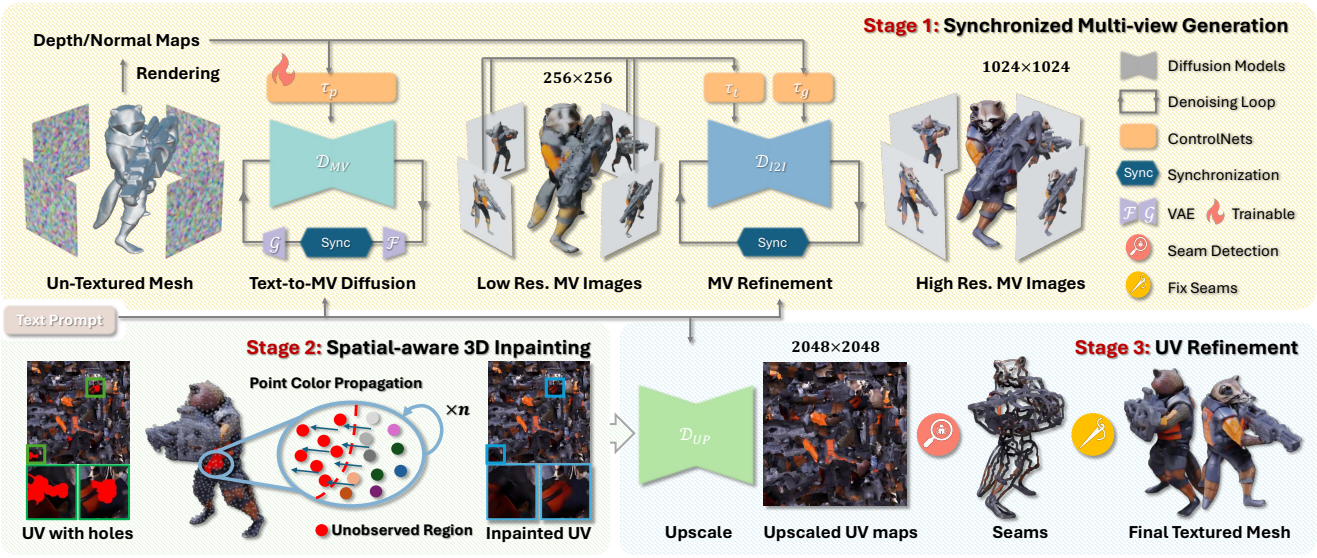


Figure 2. **The Framework Overview of MVPaint.** Given an input mesh, Stage 1 of MVPaint utilizes a synchronized multi-view generation (**SMG**) model, consisting of a control-based T2MV model and an I2I model, for 3D texture initialization. In Stage 2, the synchronized views are reprojected back to UV space, where inpainting is performed on the 3D point cloud to fill the holes (shown in red dots), hence completing the UV map. In Stage 3, the completed UV map undergoes super-resolution, adding finer details, followed by seam detection and 3D-aware smoothing to achieve a complete, seamless, and multi-view consistent 3D texture.

meshes, has become a crucial research area. This enables the creation of 3D assets with consistent appearances across various perspectives [16, 24, 53]. Groundbreaking studies like Zero-1-to-3 [30] and Consistent-1-to-3 [52] have employed viewpoint-conditioned diffusion models to synthesize novel views of objects, ensuring seamless transitions between the generated perspectives. Building upon these advancements, Zero123++ [43] and MVDream [44] have adopted an approach that tiles multiple view images into a single canvas for generation, respectively producing consistent hexa-view and tetra-view images.

3D Texture Generation. Traditionally, texture generation relies on manual or procedural techniques [27, 29, 47, 50], which were effective for basic applications but lacked complexity. The introduction of global optimization techniques [23, 49] allows for more detailed textures that better matched 3D model geometries. AI-based 3D texture generation is initially dominated by generative adversarial networks (GANs) [17, 33, 38, 58], and then the focus has shifted towards latent diffusion models (LDM) [20, 41], with models like Stable Diffusion [36, 41] showing promising results. With large-scale text-image prior models, texture generation methods [9, 31, 40] leverage text-image correlations to update rendered 3D views by extracting gradients from CLIP [34, 39, 42]. Iterative methods [11, 35, 40] enhance texture quality and consistency across 3D models by rendering depth or normal maps, with the help of ControlNet [57]. Other methods [3, 55] generate multi-view images with depth or normal conditioned sparse views and then apply inpainting and refinement directly on a UV map with a position-map-controlled diffusion network. However, it is challenging for UV diffusion models to directly generate correct and 3D-continuous texture patches, as they are frequently packed to separate UV regions. Recently, multi-path diffusion [2] synchronizes the wrapped latent [31] or images [26, 56] during multiple single-view DDIM [45] processes. Nonetheless, these methods [26, 31] tend to get trapped to the multi-face issues also known as the Janus problem which is a typical phenomenon in 3D generation using 2D priors. More discussions are provided in supplementary materials Sec.A.

3. Our Approach

Given an untextured mesh $\mathbf{M} = (\mathrm{V}, \mathrm{F})$ and a texture prompt c, where $\mathrm{V} = \{v_i\}$ representing the set of 3D vertex $v_i \in \mathbb{R}^3$ and $\mathrm{F} = \{f_i\}$ representing the set of triangular faces each defined by a triplet of vertices, MVPaint aims to generate a high-quality texture map (2K level) represented as a multi-channel UV image $\mathbf{T} \in \mathbb{R}^{H \times W \times C}$ conditioned on the texture prompt c. To achieve multi-view consistency, high-quality, and seamless 3D textures, MVPaint utilizes three major stages, including **1)** Synchronized Multi-view Generation (SMG) model - for simultaneously generating dense view images as the initial texturing (see Sec. 3.1); **2)** Spatial-aware 3D Inpainting (S3I) model - to inpaint and enhance the texturing based on spatial relations (see Sec. 3.2); and **3)** UV Refinement (UVR) module - to conduct upsampling and refinement for generating the final UV texture (see Sec. 3.3);. The overview of MVPaint is illustrated in Fig. 2.

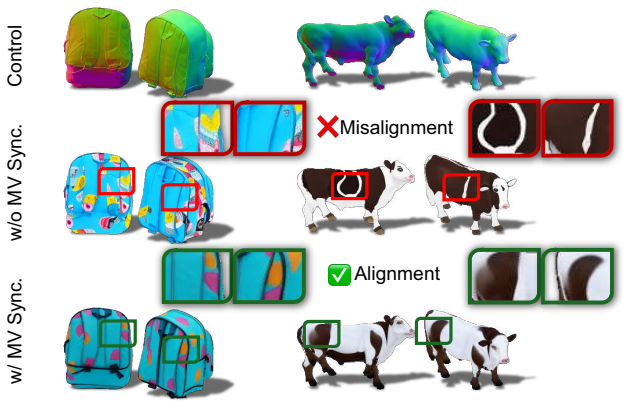


Figure 3. **The Effectiveness of Synchronization on Multi-view Image Generation.** Although T2MV models generate Janus-problem-free results, they still suffer from texture misalignment from different views. In contrast, the proposed SMG model can effectively enforce multi-view consistency for T2MV generation.

3.1. Synchronized Multi-View Generation

Building on the success of using 2D diffusion priors in recent Text-to-Image (T2I) models [13, 15, 21, 41], many 3D texturing methods [5, 31, 40, 55] initialize their textures using depth-conditioned 2D generation models. Specifically, they typically begin by rendering depth maps from various viewpoints sampled around the given 3D mesh model $\mathbf{M}$, then using a pre-trained depth-to-image diffusion model to generate multi-view images based on the text instructions. Despite leveraging 2D generation priors, existing methods often overlook 3D priors during generation, leading to low-quality multi-view results plagued by issues such as multi-view inconsistencies, the Janus problem, and over-smoothed textures lacking vivid details.

Text-to-Multi-View for 3D Texture Generation. In contrast, we utilize a Text-to-Multi-View (T2MV) diffusion model $\mathcal{D}_{\text{MV}}$, where the multi-view prior acts as a generalizable 3D prior [44], to generate consistent multi-view images in a single forward process conditioned on the text instruction c. In particular, we leverage the T2MV model $\mathcal{D}_{\text{MV}}$ by training a control model [57] τ_p to guide the generation process using the depth or normal map $\mathbf{P}^l_{\{1,2,\cdots,N\}}$ (abbr. as $\mathbf{P}^l_{\text{N}}$) from the corresponding view. Formally, the generation of initial low-resolution multi-view images $\mathbf{I}^l_{\{1,2,\cdots,N\}}$ (abbr. as $\mathbf{I}^l_{\text{N}}$) could be formulated as:

$$\mathbf{I}^l_{\text{N}} = \mathbf{I}^l_{\{1,2,\cdots,N\}} = \mathcal{D}_{\text{MV}}(z^{\text{MV}}, c, \mathbf{P}^l_{\text{N}}; \tau_p), \tag{1}$$

where z^{MV} is latent for MV generation.

Multi-View Synchronization. Existing T2I-model-based texturing methods [2, 28, 31] enhance the alignment of texture predictions from different views by integrating their denoising processes on the latent according to the shared UV space. Although significant differences across different views could lead to overly smoothed textures, the synchronization operation could eliminate minor discrepancies by adjusting the generation results of different views based on the same UV map. Unfortunately, the latent of a T2MV model $\mathcal{D}_{\text{MV}}$ usually has a low resolution (*e.g.*, 32×32), which complicates establishing a robust mapping to the UV space, especially when UV unwrapping is complex, hence hindering the effective capture of intricate visual relationships across different views.

Rather than synchronizing in latent space, we propose the Synchronized Multi-view Generation (**SMG**) model, which fuses the multi-view diffusion path by aligning the multi-view generation in the decoded image domain. Given the latent z^{MV}_t of $\mathcal{D}_{\text{MV}}$ at time step t, a clean intermediate state $z^{\text{MV}}_{0|t}$ (denoted as z'^{MV} in the following for simplicity) is obtained by removing the noise from z^{MV}_t. Afterward, z'^{MV} is converted into image space $\mathbf{I}'_{\text{N}}$ with a larger resolution (*e.g.*, 256×256) by using the pretrained decoder $\mathcal{G}_\psi(\cdot)$ of the VAE in $\mathcal{D}_{\text{MV}}$. Subsequently, we generate the synchronized UV map $\mathbf{T}'_{\text{sync}}$ by fusing the multi-view generation with inverse UV mapping, applying weights based on the cosine angle between the view direction $\mathbf{v}_i$ and the surface normal $\mathbf{n}_{uv}$ represented in the UV space. Afterward, the synchronized multi-view images $\mathbf{I}'_{\text{N}_{\text{sync}}}$ are generated by rasterizing the UV map, which is then encoded as synchronized latent $z'^{\text{MV}}_{\text{sync}}$ using the pretrained encoder $\mathcal{F}_\phi(\cdot)$ of the VAE in $\mathcal{D}_{\text{MV}}$. Consequently, the multi-view consistency could be enhanced, and the synchronized multi-view images $\mathbf{I}^l_{\text{N}_{\text{sync}}}$ could be obtained as follows:

$$\mathbf{I}^l_{\text{N}_{\text{sync}}} = \mathcal{D}_{\text{MV}}(\{z^{\text{MV}}; z'^{\text{MV}}_{\text{sync}}\}, c, \mathbf{P}^l_{\text{N}}; \tau_p), \tag{2}$$

$$\text{where} \begin{cases} z'^{\text{MV}}_{\text{sync}} = \mathcal{F}_\phi(\mathbf{I}'_{\text{N}_{\text{sync}}}), \quad \mathbf{I}'_{\text{N}_{\text{sync}}} = \mathcal{R}(\mathbf{T}'_{\text{sync}}), \\ \mathbf{T}'_{\text{sync}} = \sum_i^N \cos(\mathbf{v}_i, \mathbf{n}_{uv})\mathbf{T}'_i, \\ \mathbf{T}'_i = \mathcal{R}^{-1}(\mathbf{I}'_i), \quad \mathbf{I}'_{\text{N}} = \mathcal{G}_\psi(z'^{\text{MV}}), \end{cases} \tag{3}$$

$\mathcal{R}(\cdot)$ and $\mathcal{R}^{-1}(\cdot)$ denote the UV rasterization and UV mapping function, respectively. Notably, using a single synchronization step could be adequate, while many synchronized diffusion steps may result in unstable generations. The effectiveness of synchronization could be visualized in Fig. 3.

Synchronized Refinement. To enhance and upscale multi-view images $\mathbf{I}^l_{\text{N}_{\text{sync}}}$, we utilize an Image-to-Image (I2I) generation module $\mathcal{D}_{\text{I2I}}(\cdot)$ for synchronized texture refinement, which generates high-fidelity images $\mathbf{I}^h_{\text{N}_{\text{sync}}}$ at a higher resolution (1K level) while preserving multi-view consistency. Specifically, we leverage two pretrained control models [41, 57], including a texture refinement model τ_t and a geometry enforcement model τ_g, which adopts the synchronized denoising mechanism [31] to ensure consistency across different views. Formally, the high-quality multi-

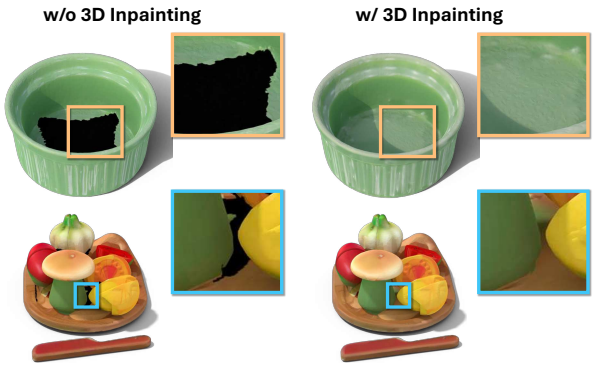


Figure 4. **Spatial-aware 3D Inpainting** could effectively accomplish texture completion for 3D structures with large unobserved areas or complex geometries.

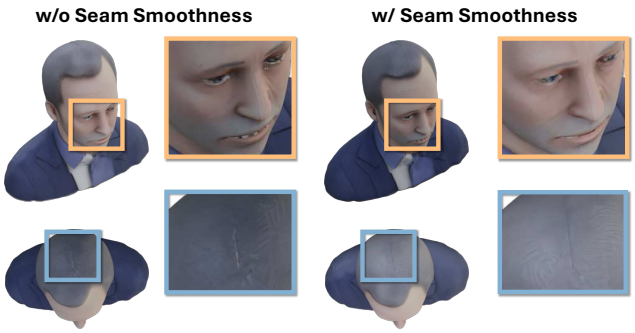


Figure 5. **Spatial-aware Seam-smoothing Algorithm** could revise texture seams from 2D UV unwrapping by smoothing color vectors using their 3D neighbors.

view image refinement could be formulated as:

$$\mathbf{I}^{h}_{\mathrm{N_{sync}}} = \mathcal{D}_{\mathrm{I2I}}(z^{\mathrm{I}}, \mathbf{I}^{l}_{\mathrm{N_{sync}}}, \mathbf{P}^{h}_{\mathrm{N}}, s_t, s_g; \tau_t, \tau_g), \tag{4}$$

where z^{I} is a random initialized latent for single image generation, $\mathbf{P}^{h}_{\mathrm{N}}$ is the set of N high-resolution geometric (*e.g.*, depth or normal) maps, and s_t, s_g is the user-defined strength of the two control models.

3.2. Texture Inpainting in 3D Space

Although high-quality multi-view images $\mathbf{I}^{h}_{\mathrm{N_{sync}}}$ cover most of the mesh surface, there are still unobserved regions that need to be inpainted. After performing UV mapping $\mathcal{R}^{-1}(\cdot)$, multi-view images $\mathbf{I}^{h}_{\mathrm{N_{sync}}}$ are projected into the UV space, achieving an incomplete UV map $\mathbf{T}_i$. To address the artifacts caused by complex self-occlusion issues during UV mapping $\mathcal{R}^{-1}(\cdot)$, we refine the projection area by identifying and limiting the regions affected by occlusion (check Sec.B.5 for details). For UV map completion, existing methods [31, 55] mostly perform inpainting directly in UV space. However, adjacent 3D areas are frequently mapped to non-adjacent 2D regions within $\mathbf{T}_i$, which becomes more pronounced when $\mathbf{T}_i$ is highly fragmented.

3D Point Cloud Inpainting. In light of this, we propose a Spatial-aware 3D Inpainting (**S3I**) module (shown in Stage 2 of Fig.2) to inpaint texture in the 3D space, which generates a full-coverage 3D texture $\mathbf{T}_c$ conditioned on the incomplete texture $\mathbf{T}_i$, while enforcing 3D geometry-aware spatial consistency. Specifically, we first generate a dense colored point cloud $\mathbf{P}_{uv} \in \mathbb{R}^{N_{uv}\times 6}$ by concatenating the 3D point coordinates generated based on each pixel from valid regions in $\mathbf{T}_i$ and their corresponding RGB values. Note that unpainted pixels from valid UV areas will also be used for generating 3D points $\mathbf{P}_u \subset \mathbf{P}_{uv}$ with zero-initialized color vector (*i.e.*, $(0, 0, 0)$). Accordingly, inpainting the UV texture could be reformulated as predicting a suitable color vector for each point $\mathbf{p} \in \mathbf{P}_u$ conditioned on the set of colored 3D points $\mathbf{P}_v \subset \mathbf{P}_{uv}$ generated from the visible area in $\mathbf{T}_c$. We highlight that the learning-free approach S3I is unaffected by the UV unwrapping results.

Spatial-aware Color Propagation. To address the 3D point inpainting problem, we propose a Spatial-aware Color Propagation (**SCP**) algorithm, which iteratively propagates the color value from $\mathbf{P}_v$ to $\mathbf{P}_u$. In each iteration, the k-nearest neighbors $\mathbf{N}^{\mathbf{p}_i}_{\mathrm{k}} = \{\mathbf{q}_j \in \mathbf{P}_v \,|\, 1 \le j \le k\}$ of each point $\mathbf{p}_i \in \mathbf{P}_u$ are selected, and the color vector of $\mathbf{p}_i$ is then estimated by applying a weighted sum of the color vectors from each neighbor within $\mathbf{N}^{\mathbf{p}_i}_{\mathrm{k}}$. For each neighboring point $\mathbf{q}_j \in \mathbf{N}^{\mathbf{p}_i}_{\mathrm{k}}$, the aggregation weight w_j is computed by considering both the Euclidean distance d_j and the surface normal similarity between $\mathbf{q}_j$ and $\mathbf{p}_i$:

$$w_j = \frac{1/d_j}{\sum_j 1/d_j} \times f(\mathbf{n}_j \cdot \mathbf{n}_i), \tag{5}$$

where $\mathbf{n}_j$ and $\mathbf{n}_i$ are surface normal of $\mathbf{q}_j$ and $\mathbf{p}_i$, respectively. $f(\cdot)$ is a robust mapping function, defined as:

$$f(x) = \begin{cases} 1 \times 10^{-8}, & \text{if } -1 \le x < 0.5, \\ x, & \text{if } 0.5 \le x < 0.9, \\ 10, & \text{if } 0.9 \le x \le 1. \end{cases} \tag{6}$$

After n iterations, all 3D points will be painted, resulting in the full-coverage UV texture $\mathbf{T}_c$ (1K level). Please refer to the supplementary (Sec.B.3) for algorithm details. The effectiveness of S3I could be visualized in Fig. 4.

3.3. UV Refinement

Even though a full-coverage UV texture $\mathbf{T}_c$ has been acquired, texture details may appear flawed caused by projection errors during $\mathcal{R}^{-1}(\cdot)$ and the interpolation filling process. To achieve a high-quality UV map $\mathbf{T}$, we propose a UV Refinement (**UVR**) module (shown in Stage 3 of Fig.2), mainly consisting of 1) a super-resolution module for UV texture upsampling and refinement; 2) and a spatial-aware seam-smoothing algorithm for repairing the texture seams caused by UV upsampling.



Figure 6. **Qualitative Results on Text-conditioned 3D Texture Generation.** MVPaint could constantly generate high-quality 3D textures, while existing methods frequently provide flawed results. Note that the input text prompts are simplified for better presentation.

UV Space Super-Resolution. In order to generate more aesthetically pleasing and higher-resolution texture mappings, we perform super-resolution on $\mathbf{T}_c$ in UV space. We use an Image-to-Image Upscale (UP) diffusion model $\mathcal{D}_{\text{UP}}$ for super-resolution in UV space with an upscale model τ_{up}:

$$\mathbf{T}_{\text{up}} = \mathcal{D}_{\text{UP}}(z^{\text{UV}}, \mathbf{T}_c; \tau_{\text{up}}), \tag{7}$$

where z^{UV} is a random initialized latent for high-resolution UV generation. The upscale model τ_{up} can be selected as either a tiling control network or an upscale network. Please refer to Sec.B.6 for detailed information.

Spatial-aware Seam-smoothing Algorithm. Although more delicate and intricate textures could be received in $\mathbf{T}_{\text{up}}$, the discontinuities caused by UV unwrapping frequently lead to abrupt changes. To resolve this problem, we introduce a Spatial-aware Seam-smoothing Algorithm (**SSA**) to revise the seams in 3D point cloud space. First, we extract the binary image $\mathbf{T}_{\text{valid}}$ to mark the valid pixels from $\mathbf{T}_{\text{up}}$, based on which we perform connectivity analysis and edge extraction for detecting the seam mask $\mathbf{m}_{\text{seam}}$. Similar to the projection operations in S3I, we resample $\mathbf{T}_{\text{up}}$ into 3D colored point cloud $\mathbf{P}_{\text{up}} \in \mathbb{R}^{N_2 \times 6}$, followed by applying the SSA algorithm for seam repairing. Specifically, we extract subsets $\mathbf{P}_{\text{seam}}$ and $\mathbf{P}_{\text{n-seam}}$ from $\mathbf{P}_{\text{up}}$ using the seam mask $\mathbf{m}_{\text{seam}}$. Afterward, we construct a kd-tree with $\mathbf{P}_{\text{n-seam}}$ and then refine $\mathbf{P}_{\text{seam}}$ with neighboring points from $\mathbf{P}_{\text{n-seam}}$. Finally, by calculating the normal vector cosine similarity and the distance for weighted coloring, we obtain the final high-quality seamless texture $\mathbf{T}$ (2K level). The effectiveness of SSA is illustrated in Fig. 5. More algorithm details are provided in Sec.B.4.

4. Experiments

In this section, extensive experiments have been conducted to evaluate the effectiveness of MVPaint in generating high-quality 3D textures from texture instructions.

Implementation Details. We use MVDream [44] as the base model for T2MV $\mathcal{D}_{\text{MV}}$ in SMG, adding a control module τ_p. For synchronized refinement, we use SDXL [36] deployed with two ControlNets [12, 51] for I2I refinement, refining 128×128 latents and synchronizing them at 512×512 $\mathbf{T}_{\text{sync}}$. All SMG steps use $N = 8$ views with evenly distributed azimuth angle and interleaved elevation of $\pm30°$. More details are in the supplementary materials.

4.1. Text-Instructed 3D Texture Generation

Dataset Details. We filter out 104k Objaverse [10] samples with valid single texture maps and bake the texture with Xatlas [54] wrapping, where 102k samples are selected for training and the rest for validation and evaluation. The training data consists of rendered per-view RGB images $\mathbf{I}_{\text{N}}$ and their corresponding control proxy images $\mathbf{P}^l_{\text{N}}$. The textual annotations are derived by instructing the CogVLM-2 model [22] to describe the category, texture, and appearance of the 3D objects, utilizing the multi-view images $\mathbf{I}_{\text{N}}$ as the model input. Afterward, keywords are summarized using another LLM [25].

Evaluation Benchmarks. To comprehensively evaluate arbitrary types of meshes for the text-to-texture (T2T) generation, we build two evaluation benchmarks: **(1)** the Objaverse T2T benchmark. To achieve a diverse Objaverse T2T benchmark, we integrate the Paint3D test set [55], containing 301 artist-crafted meshes from Objaverse, along with 3D scans and complex scenes, such as 3D character mod-

Table 1. **Quantitative Results** on the Objaverse T2T Benchmark.

Method	FID↓	KID↓	CLIP↑	User Study		
				Overall ↑	Seamless↑	Consistency↑
TEXTure [40]	28.03	7.60	**20.30**	3.81	3.66	3.31
Paint3D [55]	25.28	5.19	19.27	3.85	3.36	3.51
SyncMVD [31]	26.99	5.72	20.19	3.96	3.85	3.59
Ours	**20.89**	**3.45**	19.87	**4.19**	**4.25**	**3.98**

Table 2. **Quantitative Results** on the GSO T2T Benchmark.

Method	FID↓	KID↓	CLIP↑	User Study		
				Overall ↑	Seamless↑	Consistency↑
TEXTure [40]	24.76	5.50	**23.44**	3.78	3.97	3.65
Paint3D [55]	37.29	10.24	21.21	3.24	3.14	3.45
SyncMVD [31]	26.96	5.37	23.08	4.01	4.12	3.71
Ours	**20.02**	**3.12**	23.25	**4.13**	**4.51**	**4.21**

Table 3. **Ablation Study** on Objaverse [10] Benchmark.

Method	FID↓	KID↓	CLIP↑
w/o MV Sync.	21.42	3.72	19.90
w/o MV Diff	27.63	5.82	**20.44**
w/o Geo. Refinement	21.17	3.67	20.00
w/o 3D Inpainting	20.91	3.56	19.87
w/o Seam Smoothing	**20.82**	3.54	19.92
Full Design	20.89	**3.45**	19.87

els, which leads to an extensive test set with a total of 1000 models. **(2)** the GSO T2T benchmark. The Google Scanned Objects (GSO) dataset provides a curated collection of 1032 3D scanned common household items, each captured at high resolution to capture intricate details. As 3D texture generation methods are primarily trained on 3D objects from Objaverse, the GSO dataset could be used to assess their generalizability. Therefore, we establish the GSO T2T benchmark using the complete GSO dataset.

Evaluation metrics. After generating 3D textures, 512-resolution images of the mesh are rendered with the generated textures from 16 fixed viewpoints at the same elevation, *i.e.*, 15° for fairness. Then, we compare the rendered images with the true image distribution generated using the ground truth textures. For a thorough evaluation, we use common generative metrics: Fréchet Inception Distance (FID) [19], Kernel Inception Distance (KID) [4], and CLIP score [18] to assess image distribution, quality, and richness. The KID values are scaled by 10^3 in all tables.

User Study. To complement the quantitative results based on the generative metrics, we also conduct a user study to capture human preferences regarding the generated 3D textures. 10 participants are invited to evaluate the textured meshes in an interface that allows free navigation and observation of the 3D models. Each participant is asked to rate the following aspects on a scale of 1 to 5: overall quality, seam visibility, and overall consistency reflecting their preferences for each criterion.

Evaluation Results. We select all existing open-source SoTA methods on T2T for comparison, including TEXTure [40], Paint3D [55], and SyncMVD [31]. The quantitative results on the Objaverse T2T benchmark are reported in Tab. 1. Our method, MVPaint achieves the best scores in terms of FID and KID, outperforming previous SoTA methods by over 4.3 and 1.7, respectively, while TEXTure offers the best CLIP score. It is worth noting that the TEXTure often encounters the Janus problem, potentially resulting in a high CLIP score. According to the user study, MVPaint outperforms previous SoTA methods, achieving the highest ratings across all evaluated aspects, including overall quality, seam visibility, and consistency.

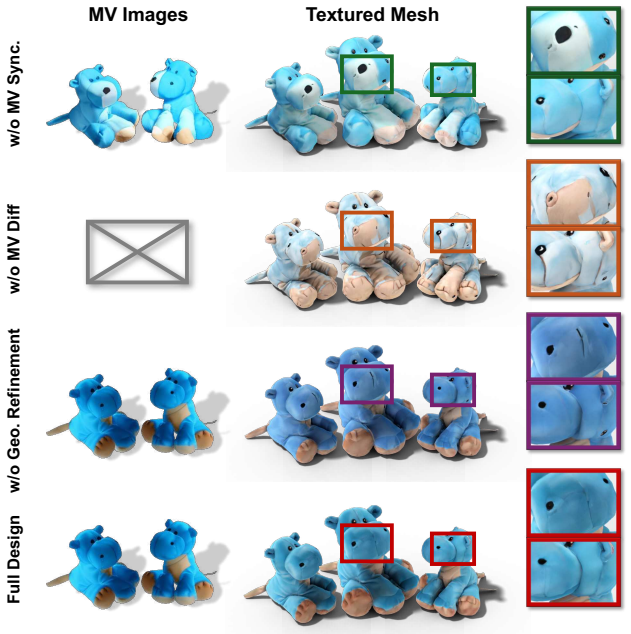


Figure 7. **Ablation Study on SMG Designs.**

Since none of the T2T methods incorporate the GSO dataset in their training, the GSO T2T benchmark could be used to evaluate their generalizability. The quantitative results on the GSO T2T benchmark are reported in Tab. 2. Similar to the results observed in the Objaverse T2T benchmark, MVPaint achieves the best objective FID, KID performance, subjective user study scores, and second-best CLIP scores on the GSO T2T benchmark. As Paint3D incorporates a crucial submodule that is trained on the Objaverse dataset, there is a noticeable decline in performance when evaluated on the GSO T2T benchmark, compared to its performance on the Objaverse T2T benchmark. In contrast, MVPaint maintains its ability to produce high-quality 3D textures on the GSO benchmark, despite being trained on a dataset curated from Objaverse. The qualitative comparisons are visualized in Fig. 6.

4.2. Ablation

SMG Designs. To validate the effectiveness of SMG module from Stage 1, we conduct a detailed ablation study over its three key designs, including the T2MV diffusion model, multi-view synchronization, and geometry-aware



Figure 8. **Diverse Texturing on the Same Model.** We use GPT4 [1] to generate 38 random texturing prompts without cherry-picking.

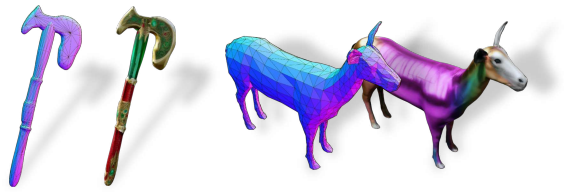


Figure 9. **Application of MVPaint.** Texturing for generated 3D assets from MeshXL [6] and MeshAnything [7, 8].

refinement. Specifically, in **w/o MV Sync.** we omit the synchronization module and use $\mathbf{I}^l_{\mathrm{N}}$ rather than $\mathbf{I}^l_{\mathrm{N_{sync}}}$ for T2MV model. In **w/o MV Diff** we omit the whole T2MV modol $\mathcal{D}_{\mathrm{MV}}$ and its output $\mathbf{I}^l_{\mathrm{N_{sync}}}$, and use the τ_g in the $\mathcal{D}_{\mathrm{I2I}}$ to generate from scratch. In **w/o Geo. Refinement**, we drop the geometry refinement control model τ_g and only use the tiling model τ_t. The quantitative results of the combined benchmarks are reported in Tab. 3 which testify to the efficacy of the proposed designs. The qualitative results are illustrated in Fig. 7. Without multi-view synchronization, inconsistent MV images result in poor initializations for subsequent refinement. In the absence of MV images, the refinement network struggles to identify the orientation of the untextured mesh across different views, leading to the Janus problem. Without geometry-guided refinement (τ_g), the refinement merely adds details to coarse MV images and exacerbates initialization artifacts. With the full design, MVPaint could produce multi-view consistent results, free of the Janus problem. Quantitative ablation results on the GSO benchmark are presented in Sec.C.2, as well as further analysis.

3D Inpainting and Seam Smoothing. We also validate the effectiveness of 3D inpainting (from Stage 2) and seam smoothing (from Stage 3), and the quantitative results are reported in Tab. 3. Their quantitative results are very close, with the FID showing slight variations within ± 0.1 and the KID consistently remaining below 0.2. Although these refinement operations have minimal impact on metrics, they could effectively eliminate texture artifacts.

4.3. Application

MVPaint create faithful 3D textures from text prompts, supporting applications such as: 1) Diverse 3D Texture Generation: For a given 3D model, MVPaint can produce diverse textures based on different text prompts, as illustrated in Fig. 8 with 38 unique textures for a unicorn model. 2) Texturing AI-Generated 3D Models: MVPaint generates high-quality textures independently of UV unwrapping quality, effectively texturing AI-generated meshes even if they contain minor artifacts, as shown in Fig. 9, where meshes are from MeshXL and MeshAnything [6–8].

5. Conclusion

We present MVPaint, a comprehensive framework for generating 3D textures from text in three stages: synchronized multi-view generation, 3D texture inpainting, and UV refinement. Utilizing synchronized multi-view diffusion, MVPaint initializes 3D textures based on generated multi-view images, ensuring high cross-view consistency. Then uncovered areas are inpainted in 3D space. Finally, a refinement module enhances and upscales the 3D mesh in the UV space, producing high-quality 2K textures. Extensive experiments demonstrate MVPaint's high-quality generation performance, surpassing existing SoTA methods.

Acknowledgments. This study is supported by the National Key R&D Program of China No.2022ZD0160102. This study is also supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOET2EP20221-0012, MOE-T2EP20223-0002), and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. In *arXiv preprint arXiv:2303.08774*, 2023. 8

[2] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: fusing diffusion paths for controlled image generation. In *ICML*, 2023. 3, 4

[3] Raphael Bensadoun, Yanir Kleiman, Idan Azuri, Omri Harosh, Andrea Vedaldi, Natalia Neverova, and Oran Gafni. Meta 3d texturegen: Fast and consistent texture generation for 3d objects. In *arXiv preprint arXiv:2407.02430*, 2024. 2, 3

[4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018. 7

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

[6] Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Yanru Wang, Zhibin Wang, Chi Zhang, et al. Meshxl: Neural coordinate field for generative 3d foundation models. In *arXiv preprint arXiv:2405.20853*, 2024. 8

[7] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. In *arXiv preprint arXiv:2406.10163*, 2024. 8

[8] Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. In *arXiv preprint arXiv:2408.02555*, 2024. 8

[9] Yiwen Chen, Chi Zhang, Xiaofeng Yang, Zhongang Cai, Gang Yu, Lei Yang, and Guosheng Lin. It3d: Improved text-to-3d generation with explicit view synthesis. In *AAAI*, 2024. 3

[10] 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 *CVPR*, 2023. 2, 6, 7

[11] Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. Flashtex: Fast relightable mesh texturing with lightcontrolnet. In *ECCV*, 2024. 3

[12] Diffusers. Controlnet depth sdxl 1.0. https://huggingface.co/diffusers/controlnet-depth-sdxl-1.0, 2023. 6

[13] 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. In *NeurIPS*, 2021. 2, 4

[14] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, 2022. 2

[15] 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 *ECCV*, 2022. 2, 4

[16] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. 3

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 3

[18] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 7

[19] 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. In *NIPS*, 2017. 7

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[21] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. In *JMLR*, 2022. 2, 4

[22] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. In *arXiv preprint arXiv:2408.16500*, 2024. 6

[23] Jingwei Huang, Justus Thies, Angela Dai, Abhijit Kundu, Chiyu Jiang, Leonidas J Guibas, Matthias Nießner, Thomas Funkhouser, et al. Adversarial texture optimization from rgb-d scans. In *CVPR*, 2020. 3

[24] Dong Huo, Zixin Guo, Xinxin Zuo, Zhihao Shi, Juwei Lu, Peng Dai, Songcen Xu, Li Cheng, and Yee-Hong Yang. Texgen: Text-guided 3d texture generation with multi-view sampling and resampling. In *ECCV*, pages 352–368. Springer, 2024. 3

[25] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. In *arXiv preprint arXiv:2310.06825*, 2023. 6

[26] DaDong Jiang, Xianghui Yang, Zibo Zhao, Sheng Zhang, Jiaao Yu, Zeqiang Lai, Shaoxiong Yang, Chunchao Guo, Xiaobo Zhou, and Zhihui Ke. Flexitex: Enhancing texture generation with visual guidance. In *arXiv preprint arXiv:2409.12431*, 2024. 3

[27] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen, Dani Lischinski, and Tien-Tsin Wong. Solid texture synthesis from 2d exemplars. In *SIGGRAPH*, 2007. 3

[28] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. In *NeurIPS*, 2023. 4

[29] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis. In *ACM TOG*, 2006. 3

[30] 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 *ICCV*, 2023. 3

[31] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. In *SIGGRAPH Asia*, 2024. 2, 3, 4, 5, 7

[32] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *CVPR*, 2024. 2

[33] Mehdi Mirza. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[34] 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. 3

[35] Sai Raj Kishore Perla, Yizhi Wang, Ali Mahdavi-Amiri, and Hao Zhang. Easi-tex: Edge-aware mesh texturing from single image. In *ACM TOG*, 2024. 3

[36] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *arXiv preprint arXiv:2307.01952*, 2023. 3, 6

[37] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 2

[38] Alec Radford. Unsupervised representation learning with deep convolutional generative adversarial networks. In *arXiv preprint arXiv:1511.06434*, 2015. 3

[39] 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 *ICML*, 2021. 3

[40] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *SIGGRAPH*, 2023. 2, 3, 4, 7

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 4

[42] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *CVPR*, 2022. 3

[43] 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. In *arXiv preprint arXiv:2310.15110*, 2023. 3

[44] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *ICLR*, 2024. 3, 4, 6

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3

[46] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024. 2

[47] Greg Turk. Texture synthesis on surfaces. In *SIGGRAPH*, pages 347–354, 2001. 3

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017. 2

[49] Li-Yi Wei and Marc Levoy. Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH*, 2001. 3

[50] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the art in example-based texture synthesis. In *Eurographics STAR*, 2009. 3

[51] Xinsir. Controlnet tile sdxl 1.0. https://huggingface.co/xinsir/controlnet-tile-sdxl-1.0, 2023. 6

[52] 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. In *3DV*, 2024. 3

[53] Yu-Ying Yeh, Jia-Bin Huang, Changil Kim, Lei Xiao, Thu Nguyen-Phuoc, Numair Khan, Cheng Zhang, Manmohan Chandraker, Carl S Marshall, Zhao Dong, et al. Texturedreamer: Image-guided texture synthesis through geometry-aware diffusion. In *CVPR*, pages 4304–4314, 2024. 3

[54] Jonathan Young. xatlas: A Library for Mesh Parameterization. GitHub repository, 2018. 6

[55] Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *CVPR*, 2024. 2, 3, 4, 5, 6, 7

[56] Hongkun Zhang, Zherong Pan, Congyi Zhang, Lifeng Zhu, and Xifeng Gao. Texpainter: Generative mesh texturing with multi-view consistency. In *SIGGRAPH*, 2024. 3

[57] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, 2023. 2, 3, 4

[58] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 3