StyleMVD: Tuning-Free Image-Guided Mesh Texture Stylization by Synchronized Multi-View Diffusion

Kunho Kim[1], Sanghyeon An[1], and Minhyuk Sung[2]

[1] RebuilderAI
[2] KAIST



Fig. 1: Image-guided 3D texture stylization results with StyleMVD. Given a textured mesh and a style image, our mesh texture stylization method enables high quality texture style transfer while preserve the original content of mesh's texture using the pretrained diffusion model.

Abstract. In this paper, we propose StyleMVD, a high quality texture generation framework guided by style image reference for textured 3D meshes. Style transfer in images has been extensively researched, but there has been relatively little exploration of style transfer in 3D meshes. Unlike in images, the key challenge in 3D lies in generating a consistent style across views. While existing methods generate mesh textures using pretrained text-to-image (T2I) diffusion models, accurately expressing the style of an image in text remains challenging. To overcome this, we propose StyleMVD module that enables to transfer the consistency style among the different views without additional fine-tuning from the reference style image. Specifically, StyleMVD converts existing self-attention into

sample-wise self-attention to achieve global coherence, and decouples cross-attention in diffusion models to inject the style feature from the reference style image. Our experiments show that the proposed StyleMVD can achieve impressive results in both consistent image-to-texture style transfer and texture quality at high speed.

1 Introduction

Numerous compelling image stylization techniques, such as color transfer, texture transfer, and style transfer, have been extensively demonstrated over the years. This visual attribute transfer refers to copying visual attributes of one image such as color, texture, and style, to another image while preserving the original content. This methods enhance artist productivity by generating images tailored to the style of specific platforms, and hold particular appeal within the realm of social media, where photo sharing and entertainment are prominent.

In addition to advancements in image style transfer, a significant body of research focuses on 3D stylization based on neural radiance fields (NeRF) [25], a compact 3D scene representation that produces high-quality novel-view synthesis results. However, since the NeRF representation is typically constructed from a set of images, it faces challenges in accurately capturing the geometric properties of objects, making it difficult to extract precise meshes from the scene.

While numerous prior works on stylization concentrate on 2D images [11, 13, 14, 17, 19, 30, 40, 45] and NeRF scenes [1, 9, 16, 21, 27, 28, 47, 49, 50], the study of mesh texture stylization remains relatively underexplored, despite their critical role in realistic 3D assets, which are extensively used in various applications such as games, anmetaverse. Additionally, creating high-quality mesh textures is a time-consuming task, even for expert designers. Mesh texture stylization is thus a crucial technique for generating plausible variations of existing high-quality, artist-generated 3D assets or open dataset [6, 7].

Recent advances in text-to-image (T2I) diffusion models trained on large-scale data [29, 31, 35, 38] have enabled the generation of images with remarkable visual detail. To utilize this high performance of T2I diffusion models, existing research has mainly focused on applying them to texture synthesis through text-guided approaches [2, 3, 33, 44, 46]. Existing texture synthesis approaches render the untextured 3D mesh from various camera position to generate depth maps, using them as the conditioning input for T2I diffusion models. Based on these considerations, a straightforward approach to texture stylization involves extracting textual descriptions from an reference image and synthesizing texture accordingly. However, accurately expressing the style of an image through textual descriptions remains a significant challenge. On the one hand, directly applying image-based stylization methods for the pretrained diffusion model [13, 17, 30, 40] to mesh texture produces asynchronous result across different views result in unintented flickering artifacts, due to each view is processed independently, disregarding the detail and intrinsic 3D geometric property of mesh.

To address the above issues, we present StyleMVD, a high quality texture stylization framework guided by the style image reference for textured 3D meshes

that leverages pretrained diffusion model. Unlike existing approaches [2, 3, 33, 44, 46] that focus on text descriptions to utilize T2I diffusion models, our approach directly extracts features from the reference image and inject them into the synchronized diffusion process. Specifically, StyleMVD converts existing self-attention into sample-wise self-attention to generate consistency texture across the different views, and decouples cross-attention in the diffusion model to inject the image feature from the reference style image. By switching the existing attention mechanism in the diffusion model, our framework demonstrates the capability to generate the mesh texture that exhibit both the style of the reference image and the content of original mesh while preserving multi-view consistency, as shown in the Fig. 1. This is achieved in a zero-shot manner based on existing adapter-based method, without the need for fine-tuning of existing diffusion models.

In our experiments on image-guided texture generation using Stable Diffusion XL [29] and ControlNet [48] model, the results demonstrate that our method can more effectively reflect not only the style of reference image but also the original content of mesh. Quantitatively, the user study results show that a significant preference for our method 49.20% in terms of consistency across different views, 49.16% in terms of matching better style of a given style reference image among the three candidates. Moreover, we further demonstrate our method is capable of generating textures in only 68 seconds using a single RTX 4090 GPU.

2 Related Works

2.1 Image Style Transfer

In the 2D image domain, style transfer aims to generate a new stylized image that incorporates both the content of input image and style features of the reference image. One of the seminal works, Image Analogies [14], demonstrated that it is possible to learn textures, styles, and filters. However, it has the constraint that both content and style images must be semantically similar. This constraint was later addressed by Gatys et al. [11], who proposed an unrestrained mechanism that could perform stylization on arbitrary style images using a neural network architecture. To achieve generalization in style transfer, frameworks apply additional augmentation methods in the feature space, such as the whitening and coloring transform (WCT) [45].

Recent studies have integrated the adapter [41, 43] into diffusion models to transfer style using the pretrained diffusion models. The IP-Adapter architecture incorporates image priors into diffusion models. For instance, InstantStyle [40] introduces a tuning-free mechanism that efficiently decouples style and content based IP-Adapter [43]. We utilized the IP-Adapter for style guidance during the multi-view diffusion process by extracting style features.

2.2 3D Style Transfer

Recently, research on 3D style transfer has been actively pursued, with many works guided by text input. Text2Mesh [24] combine mesh surfaces and generalized

neural fields to stylize 3D meshes. TANGO [4] achieves photorealistic results by applying the SVBRDF rendering equation during the process. X-Mesh [23] introduces a stable and accurate stylization mechanism using a novel attention module. These studies commonly use an explicit mesh surface. On the other hand, 3DStyle-Diffusion [42] employs neural implicit representation with rendered images and depth maps, offering advantages like general semantic features due to the diffusion process.

For scene-level style transfer, recent approaches use implicit scene representation like NeRF [25] for 3D stylization. SNeRF [27] and ARF [47] achieves high-quality stylization but requires a long optimization time for each reference style. StylizedNeRF [16] and INS [9] apply latent vectors to set predefined styles, but they cannot achieve generalization. StyleRF [21] achieves high-quality stylization by using style transformation in the feature space of a radiance field. LipRF [50] aims for realistic style transfer and improves performance by employing a Lipschitz network. [49] applies a reference-based ray registration process and a template-based feature matching scheme to maintain geometrically and perceptually consistent stylization. LSNeRF [28] employs a dual-branch NeRF with hash encoding from InstantNGP [26] and a two-stage pipeline consisting of reconstruction and stylization stages. SINE [1] is a NeRF-based editing approach, but it requires a pre-trained radiance field. These methods generally require significant time and training for each sample to achieve style transfer due to the complexity of the radiance field.

Text-guided style transfers yield high-quality results, but image-guided research continues to progress. The Neural 3D Mesh Renderer [18] incorporates explicit 3D representation to enable learning of 3D awareness from images. Paparazzi [20] stylizes a 3D shape by adding geometric detail. These advancements underscore the ongoing innovation and potential in integrating style transfer techniques with 3D modeling.

2.3 Zero-shot Texture Generation Using Text-Image Prior

The development of texture generation research has been slow until recently due to the lack of large-scale 3D datasets. However, recent efforts aim to achieve zero-shot capabilities using publicly available 3D datasets and generative models such as diffusion. Following this trend, several works in texture generation, including TEXTure [33] and Text2Tex [3], have focused on generating seamless textures gradually by project-and-inpaint method using a trimap with depth-conditioned diffusion. However, TEXTure requires an additional tuning process that is referring the Textual Inversion [10] and DreamBooth [37] to texture transfer. To solve multi-view inconsistency problem of project-and-inpaint methods, TexFusion [2] utilizes latent diffusion models, applying the model's denoiser on 2D renders of the 3D object, and aggregates the denoising predictions on a shared latent texture map. The final RGB textures are generated by optimizing an intermediate neural color field. To achieve seamless texutre, SyncMVD [22] concatenates textures in UV space using a Multi-View Diffusion module. Zeng *et al*. propose Paint3D [46], coarse-to-fine generative framework that employs a depth-aware

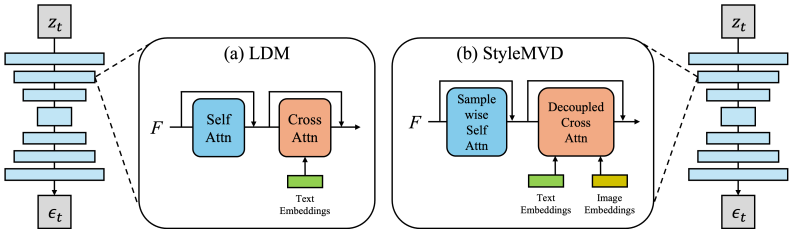

Fig. 2: Illustrations of the U-Net architectures in (a) Latent Diffusion Model (LDM) [29] and (b) StyleMVD. The basic blocks of U-Net in LDM comprise a residual block, self-attention, and cross-attention, initially designed to accept only text embeddings. In our StyleMVD, we modify the self-attention to sample-wise self-attention and the cross-attention to decoupled cross-attention to facilitate 3D mesh texture stylization from the reference image.

2D diffusion model for initial texture generation and separate UV Inpainting and UVHD models trained with the subset of Objaverse [7] for refining textures in UV space. TextureDreamer [44] employs a differentiable renderer to optimize BRDF parameters such as metallic and roughness. While it shows high-quality results, it requires DreamBooth [37] fine-tuning for each object. StyleMVD is a tuning-free architecture that can transfer arbitrary styles in a zero-shot manner. It captures style details effectively and maintains multi-view consistency.

3 Background - Attention Mechanism in Latent Diffusion Models

In this section, we breifly provide the description of latent diffusion models [29,35]. The Latent Diffusion Models (LDM) aim to generate a image from a textual prompt that utilizes a U-Net [36] as the noise prediction network. The noise predict network get a latent z_t, a timestep t and a text embedding $\mathbf{c}_{text}$ extracted from a text prompt using a pretrained text encoder, and it predicts the noise $\epsilon(z_t, t, \mathbf{c}_{text})$. The denoising U-Net ϵ_θ of LDM consists of a series of basic blocks. Each basic block contains a convolutional residual block [12], a self-attention module, and a cross-attention [39] module. At the denoising step t, the features from the previous (l-1)-th basic block are initially processed by the residual block to produce intermediate features f_t^l. These features are then restructured by a self-attention layer, and subsequently a cross-attention layer enables to control the synthesis process through text embedding c_{text} from the given text prompt P. Then, with the query feature $\mathbf{F}$, the attention mechanism can be defined as follows:

$$\mathbf{A}_{text} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}, \tag{1}$$

where $\mathbf{Q} = \mathbf{F}\mathbf{W}_q, \mathbf{K} = \mathbf{c}_{text}\mathbf{W}_k, \mathbf{V} = \mathbf{c}_{text}\mathbf{W}_v$ represent the query, key, and values matrices of the attention operation, and $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ represent the weight

of each matrix. During this procedure, the output feature from the preceding module is propagated via a residual connection as illustrated in Fig. 2-(a).

Additionally, ControlNet [48] employs zero convolution to transfer knowledge from the Latent Diffusion Model (LDM) utilizing input conditions such as depth, normal, or canny edge maps. The U-Net architecture in ControlNet predicts the noise $\epsilon(z_t, t, \mathbf{c}_{text}, C)$, conditioned on the timestep t, the text embedding $\mathbf{c}_{text}$, and an additional condition map C.

4 Method

4.1 Conditions for Mesh Texture Stylization

Stylization aims at transferring visual attributes, such as color, texture, and style, from one image to another, while preserving the original content. In 3D mesh textures, the key factors determining style are similar to those in 2D. Additionally, the elements that distinguish content in 3D meshes are the outlines of objects, such as the contours of eyes, nose, mouth, and ears on a face. With the given textured mesh and known camera poses, we can obtain the ground truth depth maps. Leveraging this aspect, previous texture synthesis approaches [3, 33, 44, 46] employ the latent diffusion model, Stable-Diffusion with depth conditioning (SD2-depth). They generate 2D views of the desired textured 3D object and project these views onto the mesh. The objective of these approaches is to generate textures for untextured meshes, thus the original texture of the input mesh is not considered. Here, to preserve the content of the original mesh, we use a canny edge map instead of a depth map. In the same setup (given the textured mesh and known camera poses), after obtaining rendered images from the textured mesh, canny edge maps can be easily extracted. This canny edge map can be used as conditioning input for pretrained diffusion models like a depth map as shown in Fig. 3.

With the conditioning canny edge map, we design an object stylization pipeline as described in Fig. 4, in order to perform tuning-free image-to-texture

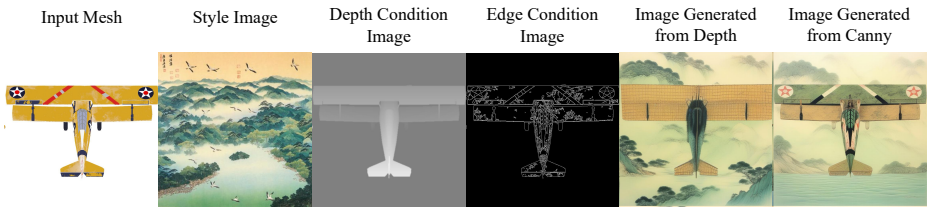

Fig. 3: Comparison of results based on depth and canny edge condition images. From rendered images of specific views of a biplane 3D mesh, depth and canny edge maps are obtained, and the resulting stylizations are presented. When using the depth map, the content of the original mesh is completely lost, whereas the stylized images produced using the canny edge map effectively reflect the detailed content of the original mesh.

stylization using a pretrained T2I diffusion model (SDXL [29]). With multiple viewpoints $\{v_1, v_2, \cdots, v_n\}$, the target 3D textured object $\mathbf{M}$ is rendered to generate a view-dependent canny edge map $\{C_{v_1}, C_{v_2}, \cdots, C_{v_n}\}$ of the object. Based on the viewpoints, we assign the directional prompt D_{v_i}, which indicates a text prompt of corresponding view direction of the rendered image (ex. front view, side view, back view). CLIP text encoder encodes each directional prompt D_{v_i} and generate text embedding $\mathbf{c}_{text}^{(v_i)}$. To extract a image feature $\mathbf{c}_{img}^{(v_i)}$ from a style reference image, we use pretrained CLIP image encoder. Extracted image feature is concatenated to each text embedding to create the embedding $\mathbf{c}^{(v_i)}$, which gives a style guidance during the denoising steps.

At the initial time step T, UV latent texture map W_T is initialized by a standard normal distribution. Subsequently, the texture of source mesh is assigned the noisy latent texture map and rendered from the set of views $\{v_1, v_2, \cdots, v_n\}$. This process yields 3D-consistent initial latents $\{z_T^{(v_1)}, z_T^{(v_2)}, \cdots, z_T^{(v_n)}\}$ of the object.

Then, to generate stylized images $\{I_{v_1}, I_{v_2}, \cdots, I_{v_n}\}$, a T2I diffusion process is assigned to each 2D views with initial noisy latent $\{z_t^{(v_1)}, z_t^{(v_2)}, \cdots, z_t^{(v_n)}\}$ as following:

$$\epsilon_\theta(z_t^{(v_i)}, \mathbf{c}^{(v_i)}, C_{v_i}), \tag{2}$$

where ϵ_θ is the diffusion model with the text-image embedding $\mathbf{c}^{v_i}$ as guidance and conditional image C_i as the condition. After obtaining the styled images I_{v_i} of each view, the 2D image pixels can be unprojected onto the 3D object surface using the rendering pipeline, resulting in the final UV texture map.

4.2 Replace Cross-attention Mechanism for Style Guidance

Ye *et al*. introduce IP-Adapter [43] that decoupled cross-attention mechanism for the existing T2I diffusion model. Inspired by this, we adopt decoupled cross-attention strategy to enhance the guidance of image feature extracted from the input style reference image as described in Fig. 2-(b).

Given the image features $\mathbf{c}_{img}$ and the query feature $\mathbf{F}$, the output of image cross-attention $\mathbf{CA}_{img}$ is described as follows:

$$\mathbf{CA}_{img} = \text{Softmax}(\frac{\mathbf{Q}(\mathbf{K}')^\top}{\sqrt{d}})\mathbf{V}', \tag{3}$$

where $\mathbf{Q} = \mathbf{F}\mathbf{W}_q, \mathbf{K}' = \mathbf{c}_{img}\mathbf{W}'_k, \mathbf{V}' = \mathbf{c}_{img}\mathbf{W}'_v$ represent the query, key, and values matrices from the image features, and $\mathbf{W}'_k, \mathbf{W}'_v$ represent the weight of each matrices initialized from $\mathbf{W}_k, \mathbf{W}_v$. Here, $\mathbf{Q}$ is the same query for image cross-attention as for text cross-attention in Eq. 1. The final output of decoupled cross-attention is defined as the simply add the output of text cross-attention

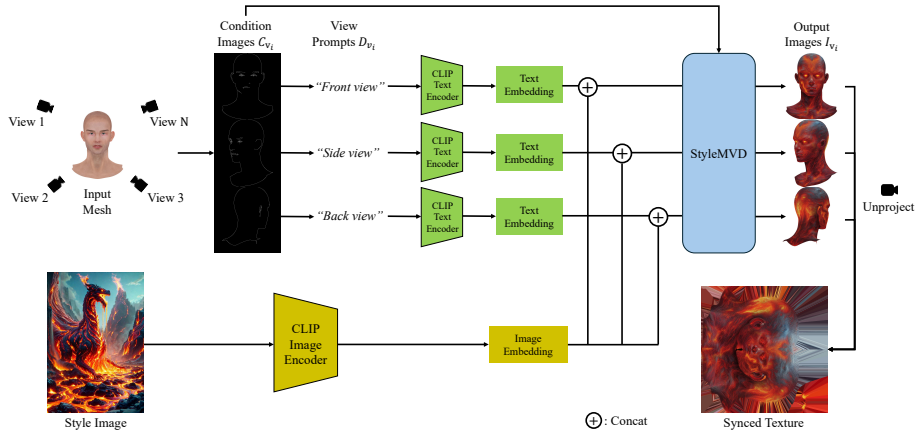

Fig. 4: The overview of StyleMVD pipeline. We first render the input textured mesh to generate condition images and view prompts based on the camera poses. Text embeddings are then extracted from the view prompts, while an image embedding is obtained from the input style image. Each view-dependent text embedding is concatenated with the image embedding and forwarded into the StyleMVD with condition images. After the view-dependent denoising process, the stylized images are unprojected onto the input mesh.

and the output of image cross-attention:

$$\mathbf{CA}_{new} = \mathbf{CA}_{text} + \lambda_{img} \cdot \mathbf{CA}_{img} \tag{4}$$

$$= \text{Softmax}(\frac{\mathbf{QK}^\top}{\sqrt{d}})\mathbf{V} + \lambda_{img} \cdot \text{Softmax}(\frac{\mathbf{Q}(\mathbf{K}')^\top}{\sqrt{d}})\mathbf{V}', \tag{5}$$

where λ_{img} is a weight factor for the image guidance.

While powerful guidance can be derived from a reference image, it influences both content and style. To address this, as proposed in [40], we inject image features exclusively into the 4th block (up blocks.0.attentions.1) of the attention layers in SDXL. This approach facilitates seamless style transfer without affecting the content of the style reference image.

4.3 Replace Self-attention Mechanism for View Consistency

Now we can give the style guidance from image, however, the diffusion process of each view does not guarantee to generate consistency result in terms of 3D mesh. Inspired by [22], we replace the self-attention mechanism with sample-wise self-attention in the existing T2I diffusion model as illustrated in Fig. 2-(b).

Based on the camera pose of each view $\{v_1, v_2, \cdots, v_n\}$, we split the batch group to implement a view-wise self-attention mechanism. From each batch, each main view is influenced by adjacent views. The key and values of the main view

and its adjacent views are concatenated, respectively. The self-attention operation for the main view with the adjacent views is then defined as follows:

$$\begin{aligned}\mathbf{SA}(v_i, \{v_{i-1}, v_i, v_{i+1}\}) = \text{Attention}(&\mathbf{Q}^{(v_i)}, \\ &\{\mathbf{K}^{(v_{i-1})}, \mathbf{K}^{(v_i)}, \mathbf{K}^{(v_{i+1})}\}, \\ &\{\mathbf{V}^{(v_{i-1})}, \mathbf{V}^{(v_i)}, \mathbf{V}^{(v_{i+1})}\})\end{aligned} \tag{6}$$

Furthermore, to enhance the guidance of a specific reference view v_{ref} (e.g., default front view) for the main view v_i, the self-attention operation with the reference view is defined as follows:

$$\mathbf{SA}(v_i, v_{ref}) = \text{Attention}(\mathbf{Q}^{(v_i)}, \mathbf{K}^{(v_{ref})}, \mathbf{V}^{(v_{ref})}) \tag{7}$$

During the denoising steps, to enforce a global consistency among different view, we define the sample-wise self-attention:

$$\textbf{Sample-wise SA} = \begin{cases} \lambda_{ref} \cdot \mathbf{SA}(v_i, v_{ref}) & \\ +(1-\lambda_{ref}) \cdot \mathbf{SA}(v_i, \{v_{i-1}, v_i, v_{i+1}\}) & \text{for } t > t_{ref} \\ \mathbf{SA}(v_i, \{v_{i-1}, v_i, v_{i+1}\}) & \text{otherwise} \end{cases} \tag{8}$$

where λ_{ref} denotes the weight factor for the reference view guidance. The early of denoising steps $t > t_{ref}$, we give a guidance of reference view to each main view to get the style consistency across the different views. In the later steps $t \leq t_{ref}$, we only use the sample-wise self-attention to eliminate the influence of the reference view, which can introduce unintended artifacts in final texture.

5 Experiments

Implementation Details Our diffusion pipeline was constructed using the Stable Diffusion XL and the ControlNet module from the Huggingface Diffusers library [8]. We adopt a conditioning scale annealing from 0.5 to 0.9 to achieve improved geometry alignment towards the final denoising steps when utilizing ControlNet. For the denoising pipeline scheduler, we utilize the DDPMScheduler. Additionally, we use PyTorch3D [32] as rendering pipeline. In all experiments with our pipeline, we utilize 10 viewpoints to encompass the object. Eight cameras are positioned along the equator at 45° intervals around the object, supplemented by two additional cameras at elevated positions directed towards the object's top. A latent resolution of each view is 96×96 and a resolution of UV latent texture map is 1,536×1,536. We configure the parameters as follows: classifier free guidance (CFG) [15] value is set to 5, λ_{img} and λ_{ref} are both set to 1, t_{ref} is set to 0.2 across a total of 20 inference steps, and v_{ref} is fixed as the front view. Our method requires approximately 22GB of VRAM. It the style reference image does not have 512x512 resolution, we resize the image such that the shorter side was adjusted to 512, followed by a center crop to produce a square image.

Datasets We collect 10 style images and 29 diverse textured objects from various sources. We conduct a comprehensive evaluation of each texture stylization technique across a dataset comprising 268 style-mesh pairs. Further details regarding the dataset are provided in the **supplementary materials**.

Baselines Recent advancements in texture generation methods have predominantly focused on text-driven approaches. Among them, TEXTure [33] and Paint3D [46] have image-guided texture transfer capability.

TEXTure uses project-and-inpaint scheme for texture generataion and employs the Textual Inversion technique [10, 37], which **fine-tune** a pretrained diffusion model to learn the concept of images and learning a new token $\langle S_{texture} \rangle$ that represents the generated texture.

Paint3D adopts a coarse-to-fine strategy utilizing a pretrained diffusion model and **their own texture diffusion model trained on 105,000 textured meshes** from the Objaverse [7] dataset. Additionally, they employ the image encoder introduced in IP-Adapter [43] to give an image condition, similar to our approach. Detailed descriptions of the experimental setup for each baseline are provided in the **supplementary materials**.

Method	Style (%)	Consistency (%)	Runtime (secs)
TEXTure [33]	38.01	37.74	710.14 ± 38.31
Paint3D [46]	12.79	13.11	70.38 ± 0.22
StyleMVD (Ours)	**49.20**	**49.15**	**67.96 ± 0.26**

Table 1: Quantitative comparison between StyleMVD and baseline methods. Style and view consistency are assessed through user studies.

5.1 Comparisons on Image-Guided Texture Stylization

Qualitative Comparisons We show qualitative comparison with our method and other baselines in Fig. 5. Each output textured mesh is rendered from the right side view, front view, left side view, and back view. Although TEXTure effectively captures the style of the reference image, it also incorporates content from the input image due to its focus on texture transfer. This results in a mismatched style transfer with the input mesh, as depicted in all cases of Fig. 5. The fourth and fifth rows of Fig. 5 demonstrate that Paint3D is unable to accurately reflect the style, as it relies on a texture diffusion model trained with standard synthetic data. Our StyleMVD, on the other hand, is capable of preserving the original properties of the input textured mesh while effectively reflecting the desired style. This is evidenced by the simple design of the luggage in the first row, the floral pattern on the robe in the fifth row, and the logo on the t-shirt in the fourth row of Fig. 5. Additional qualitative comparisons are provided in the **supplementary materials**.

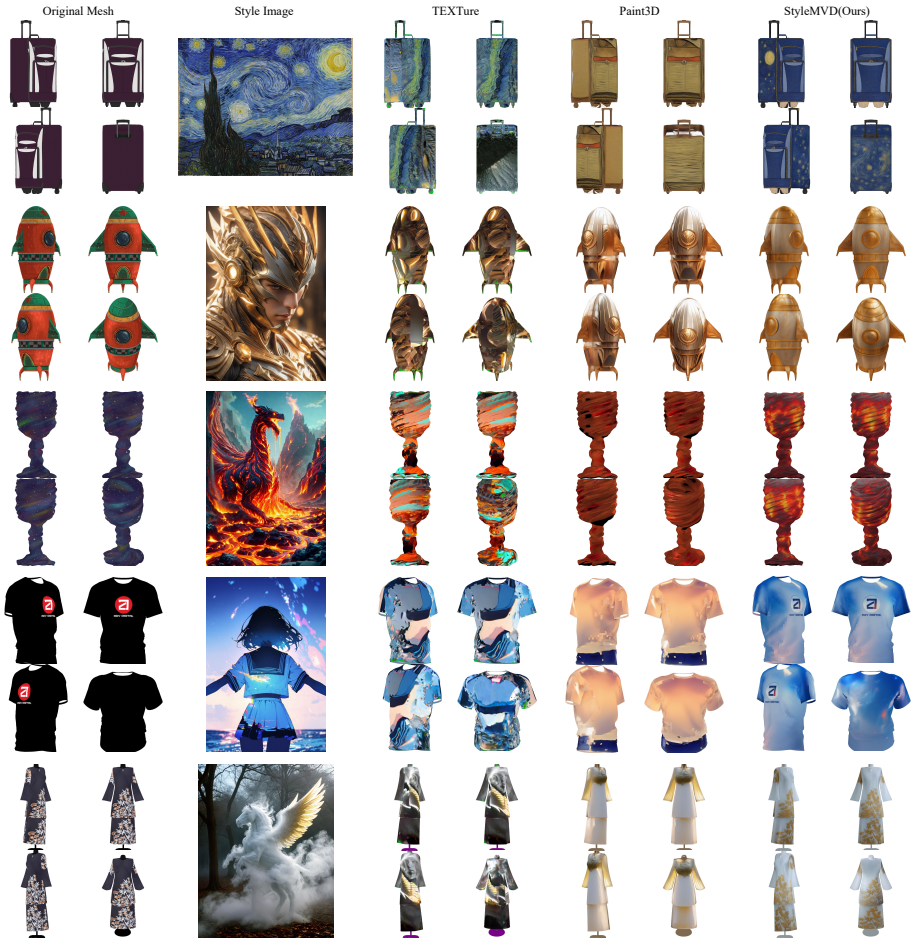

Fig. 5: Qualitative comparison of styled textures. We render the output textures of each method from four different views.

Quantitative Comparisons To evaluate the style matching between the input style image and stylized texture, we conducted two user studies following Ritchie [34]. In each user study, participants are asked the following questions to evaluate the methods: (1) "Which one matches the style of a given reference style image better while preserving the original content?" to assess style transfer (**Style**), and (2) "Which one produces more consistent results across different views while preserving the original content?" (**Consistency**) to evaluate view consistency. For each user study, we collected a total of 25 responses including 5 vigilance tasks from 138, 148 participants, respectively. The results presented in Tab. 1 indicate that human evaluators perceive our StyleMVD as generating more style-matched results compared to TEXTure and Paint3D, while also exhibiting

higher view consistency. Detailed setups for the user study can be found in the **supplementary materials**.

We further report the execution time of each method in the third column of Tab. 1. The single object texturing time for each method was measured five times using a single RTX 4090. TEXTure requires a significant amount of time (approximately 12 mins) due to the fine-tuning of diffusion models to convert image concepts into tokens. Paint3D, despite employing a two-stage process (Coarse and Refinement stages), achieves relatively fast execution times (70 secs). Our method, on the other hand, attains rapid execution times (68 secs) by utilizing only the view-dependent denoising process, without any fine-tuning or additional training.

5.2 Ablation Studies

Conditioning with a Canny Edge map vs. Depth map Fig. 6 presents the effects of using canny edge maps and depth maps as conditioning inputs for ControlNet. When using a depth map as a condition for ControlNet, the generated texture tends to overlook detailed elements such as the dog's collar and nose. In contrast, utilizing a canny edge map preserves these details effectively. These minor alterations significantly impact the stylization results, which need to preserve the original characteristics of input mesh.

Effect of Sample-Wise Self-Attention We also compare a comparison between the results obtained using pure self-attention and sample-wise self-attention in StyleMVD. In the absence of sample-wise self-attention, the application of pure self-attention can encounter difficulties in establishing a coherent representation of the object's appearance during the initial stages of the diffusion process. This may result in inconsistencies in the final rendered appearance. In severe scenarios, this can manifest as redundant features on the object's surface, such as eyes appearing on the back of a dog, as shown in the fourth column of Fig. 6.



Fig. 6: Ablation studies about input condition and sample-wise self-attention.

6 Conclusion

We presented StyleMVD, a tuning-free high quality texture stylization module designed to generate coherent texture across multiple views by modifying the self-attention and cross-attention mechanism in the pretrained text-to-image diffusion model. To maintain the integrity of the input mesh content, a canny edge map is utilized as a conditioning factor. We will focus on incorporating updates to the mesh's geometry based on the style input as the future work.

6.1 Limitations

While our StyleMVD module performs well in mesh texture stylization, the fixed condition to a canny edge map with fixed geometry limits its ability to effectively capture the geometrical style of the input image. Additionally, for elongated meshes, the object occupies a smaller proportion of the condition map, making it difficult to effectively reflect the style in the texture.

6.2 Potential Negative Societal Impacts

Texture generation models pose potential risks, including the creation of deepfakes, textures that resemble copyrighted material, discriminatory or prejudiced textured meshes, and others. Future research should aim to develop methods to prevent the generation of such harmful outputs and implement effective safeguards.

References

1. Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *CVPR*, 2023.
2. Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and KangXue Yin. TexFusion: Synthesizing 3D Textures with Text-Guided Image Diffusion Models. In *ICCV*, 2023.
3. Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2Tex: Text-driven Texture Synthesis via Diffusion Models. In *ICCV*, 2023.
4. Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. *NeurIPS*, 2022.
5. Civitai. Civitai.
6. Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A Universe of 10M+ 3D Objects. In *NeurIPS*, 2023.
7. 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.
8. Hugging Face. Diffusers: State-of-the-art diffusion models for image and audio generation in PyTorch.

9. Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. In *ECCV*, 2022.
10. Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. In *ICLR*, 2023.
11. Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. In *CVPR*, 2016.
12. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2015.
13. Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style Aligned Image Generation via Shared Attention. In *CVPR*, 2024.
14. Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. *ACM TOG*, 2001.
15. Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2022.
16. Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *CVPR*, 2022.
17. Jaeseok Jeong, Junho Kim, Yunjey Choi, Gayoung Lee, and Youngjung Uh. Visual Style Prompting with Swapping Self-Attention. *arXiv preprint arXiv:2402.12974*, 2024.
18. Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018.
19. Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual Attribute Transfer through Deep Image Analogy. *ACM TOG*, 2001.
20. Hsueh-Ti Derek Liu, Michael Tao, and Alec Jacobson. Paparazzi: surface editing by way of multi-view image processing. *ACM TOG*, 2018.
21. Kunhao Liu, Fangneng Zhan, Yiwen Chen, Jiahui Zhang, Yingchen Yu, Abdulmotaleb El Saddik, Shijian Lu, and Eric P Xing. Stylerf: Zero-shot 3d style transfer of neural radiance fields. In *CVPR*, 2023.
22. Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-Guided Texturing by Synchronized Multi-View Diffusion. *arXiv preprint arXiv:2311.12891*, 2023.
23. Yiwei Ma, Xiaoqing Zhang, Xiaoshuai Sun, Jiayi Ji, Haowei Wang, Guannan Jiang, Weilin Zhuang, and Rongrong Ji. X-mesh: Towards fast and accurate text-driven 3d stylization via dynamic textual guidance. In *ICCV*, 2023.
24. Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *CVPR*, 2022.
25. Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020.
26. Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022.
27. Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. SNeRF: stylized neural implicit representations for 3D scenes. *ACM TOG*, 2022.
28. Hong-Wing Pang, Binh-Son Hua, and Sai-Kit Yeung. Locally stylized neural radiance fields. In *ICCV*, 2023.
29. Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
30. Tianhao Qi, Shancheng Fang, Yanze Wu, Hongtao Xie, Jiawei Liu, Lang Chen, Qian He, and Yongdong Zhang. DEADiff: An Efficient Stylization Diffusion Model with Disentangled Representations. In *CVPR*, 2024.

31. Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125*, 2022.
32. Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with PyTorch3D. *arXiv preprint arXiv:2007.08501*, 2020.
33. Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. TEXTure: Text-Guided Texturing of 3D Shapes. *ACM TOG*, 2023.
34. Daniel Ritchie. Rudimentary framework for running two-alternative forced choice (2afc) perceptual studies on mechanical turk.
35. Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022.
36. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015.
37. Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-image Diffusion Models for Subject-Driven Generation. In *CVPR*, 2023.
38. Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *NeurIPS*, 2022.
39. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, 2017.
40. Haofan Wang, Matteo Spinelli, Qixun Wang, Xu Bai, Zekui Qin, and Anthony Chen. InstantStyle: Free Lunch towards Style-Preserving in Text-to-Image Generation. *arXiv preprint arXiv:2404.02733*, 2024.
41. Zhouxia Wang, Xintao Wang, Liangbin Xie, Zhongang Qi, Ying Shan, Wenping Wang, and Ping Luo. StyleAdapter: A Single-Pass LoRA-Free Model for Stylized Image Generation. In *AIGC*, 2023.
42. Haibo Yang, Yang Chen, Yingwei Pan, Ting Yao, Zhineng Chen, and Tao Mei. 3dstyle-diffusion: Pursuing fine-grained text-driven 3d stylization with 2d diffusion models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6860–6868, 2023.
43. Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.
44. 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*, 2024.
45. Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic Style Transfer via Wavelet Transforms. In *ICCV*, 2019.
46. 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.
47. Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *ECCV*, 2022.
48. Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *ICCV*, 2023.

49. Yuechen Zhang, Zexin He, Jinbo Xing, Xufeng Yao, and Jiaya Jia. Ref-npr: Reference-based non-photorealistic radiance fields for controllable scene stylization. In *CVPR*, 2023.
50. Zicheng Zhang, Yinglu Liu, Congying Han, Yingwei Pan, Tiande Guo, and Ting Yao. Transforming radiance field with lipschitz network for photorealistic 3d scene stylization. In *CVPR*, 2023.

A Appendix / supplemental material

In this supplementary material, we first describe the experimental setup of baseline methods (Sec. A.1) and dataset used in our experiments (Sec. A.2). We include the precise question presented to participants in the user study, along with a sample screen shot (Sec. A.3). Furthermore, we summarize additional qualitative comparison results with baselines (Sec. A.4).

A.1 Experimental Setups for Baseline Methods

In our experiments, we selected two baseline models: TEXTure [33] and Paint3D [46]. For TEXTure, we utilized the texture transfer functionality from the official implementation, influenced by Texture Inversion [10] and Dreambooth [37]. In the TEXTure pipeline, we fine-tune the model for 2000 steps using the input style image. Regarding Paint3D, although their paper references the use of the IP-Adapter [43], the official code of image-guided texture transfer has not been released. Consequently, we implemented it ourselves for the experiment. To ensure a fair comparison, we fix the inference step to 20 and used the default values for other parameters as specified in each implementation.

A.2 Dataset Description

We collect 10 style images and 27 meshes from various sources. A complete list of each images and meshes can be found in Tab. 2 and Tab. 3. Excluding the artwork "Starry Night" by Van Gogh, we obtained style images from Civitai [5]. 25 meshes are selected from Objaverse.v1 [7], obtained under the Objaverse license. Two meshes (human head and sneaker) is obtained from SyncMVD [22] code repository. We excluded cases that caused errors, resulting in a total of 268 style-mesh pairs used in the experiments.

Model ID	Source	Style description
Starry Night	Van Gogh	Starry Night
images/5902214	Civitai	volcaino
images/11979435	Civitai	black and white
images/12046120	Civitai	gold steal
images/12362231	Civitai	Japanese anime
images/12515146	Civitai	flame
models/56243	Civitai	Chinese ink painting
models/197434	Civitai	futuristic galaxy
models/241901	Civitai	storm cloud
models/443976	Civitai	foggy

Table 2: Description of all style references used in our evaluation.

Model ID	Source	Description
9871bf23b8a04d30a0a9b9731bae1d14	Objaverse.v1	Akatsuki coat
efa778ca442945b09300a70e34e68532	Objaverse.v1	Dior handbag
ed6a25f396a84d808977aa6d074d610e	Objaverse.v1	Torino1999 bus
83801b3d03004e4a8e5617534cd2f068	Objaverse.v1	airpot
be48f3f906ed431b98b1bf03ab7aadd6	Objaverse.v1	biplane
27120b66ac4e4385889f5542df0c2038	Objaverse.v1	black robe
6cdd3d87ff9e4b8c8dd80a7c66e6c303	Objaverse.v1	city hall
7c91bd719e594fcc930f3646db843265	Objaverse.v1	giraffe
e4f348e98ceb45e3abc77da5b738f1b2	Objaverse.v1	glasses
4ec412fc7e0648e0a7a4554143185c33	Objaverse.v1	goblet
ea66e95a48aa43aeb0cbd8b4a2664b29	Objaverse.v1	gundam
12a4585f80cc4f38b6a2b6bdd5cfd368	Objaverse.v1	headphone
8ec8f8a2022d491fa0d74279c4f07304	Objaverse.v1	human model 1
7830358a83ac427cbdd515ddb027ca52	Objaverse.v1	human model 2
d8c69ea106464c49aedb06e9164e017c	Objaverse.v1	luggage
0723b35415b0462eb5c01140b6b70340	Objaverse.v1	old chair
616761063ad845fe84e1d965bdce5595	Objaverse.v1	pickup truck
faef9fe5ace445e7b2989d1c1ece361c	Objaverse.v1	shiba dog
77b6a9cb27f6433bbf48c6ed8fc47827	Objaverse.v1	skull
ffedf6c2e1de40bcaef5c57301ca94fd	Objaverse.v1	space rocket
de9588f97f644bd689fc6502f262575d	Objaverse.v1	spike shoe
f2573d07dc5b437b9115ab4251e8f5f8	Objaverse.v1	t-shirts
0aecee43ac2749499a16ab4388a0baa2	Objaverse.v1	temple
7a0f2d413b5846f591b40580f783c53c	Objaverse.v1	tree stump
966522c1de144d43a5c5a9d68b7eff5f	Objaverse.v1	vans shoe
1d3ad2a23c444d5abfabf1a48eeb8c84	Objaverse.v1	wood mug
human head	SyncMVD	human head
sneaker	SyncMVD	sneaker

Table 3: Description of all meshes used in our evaluation.

A.3 Details of User Study

We created 10 distinct sets of user study samples, with the images in each set shuffled to ensure randomness. Each of the 268 questions featured pairs of stylized images, and for each question, we gathered 20 responses from participants on Amazon Mechanical Turk. Only participants who successfully completed our five vigilance tasks were included in the statistics. The vigilance tasks were designed to differentiate between outputs generated from the input style image and those generated from other style images. A total of 138 participants successfully completed all vigilance tasks for the style evaluation, and 148 participants did so for the consistency evaluation.



Fig. 7: Screen captures of user studies. Each row represents the example of **Style** user study and **Consistency** user study, and **vigilance task**, respectively.

We create 10 sets of user study sample and for each set, the order of the images was shuffled. Given a total of 268 questions with a random pair of images, we collected 20 responses each from the participants on Amazon Mechanical

Turk who passed our five vigilance tasks. The vigilance tasks were designed to distinguish our outputs generated by input style image from our outputs generated by other style image. 138 and 148 participants successfully completed all the vigilance tasks for the user study for style and consistency, respectively. The screen captures of each example are shown in the Fig. 7.

A.4 More Qualitative Comparisons




Input Mesh
Style Image
TEXTure [33]
Paint3D [46]
StyleMVD (Ours)


Input Mesh
Style Image
TEXTure [33]
Paint3D [46]
StyleMVD (Ours)