SceneTex: High-Quality Texture Synthesis for Indoor Scenes via Diffusion Priors

Dave Zhenyu Chen[1] Haoxuan Li[1] Hsin-Ying Lee[2] Sergey Tulyakov[2] Matthias Nießner[1]
[1]Technical University of Munich [2]Snap Research
https://daveredrum.github.io/SceneTex/

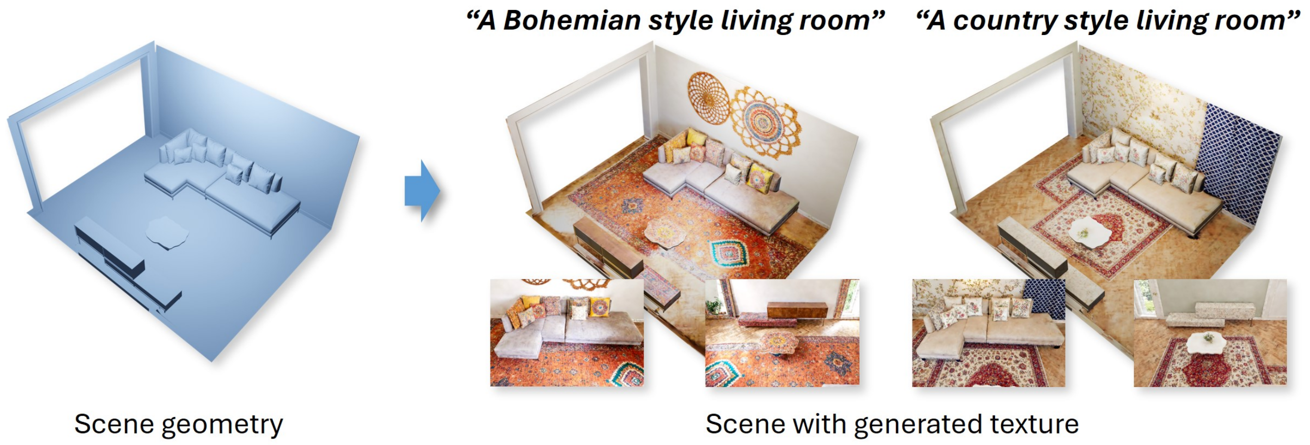

"A Bohemian style living room"
"A country style living room"
Scene geometry
Scene with generated texture

Figure 1. We introduce SceneTex, a text-driven texture synthesis architecture for 3D indoor scenes. Given scene geometries and text prompts as input, SceneTex generates high-quality and style-consistent textures via depth-to-image diffusion priors.

Abstract

We propose SceneTex, a novel method for effectively generating high-quality and style-consistent textures for indoor scenes using depth-to-image diffusion priors. Unlike previous methods that either iteratively warp 2D views onto a mesh surface or distillate diffusion latent features without accurate geometric and style cues, SceneTex formulates the texture synthesis task as an optimization problem in the RGB space where style and geometry consistency are properly reflected. At its core, SceneTex proposes a multiresolution texture field to implicitly encode the mesh appearance. We optimize the target texture via a score-distillation-based objective function in respective RGB renderings. To further secure the style consistency across views, we introduce a cross-attention decoder to predict the RGB values by cross-attending to the pre-sampled reference locations in each instance. SceneTex enables various and accurate texture synthesis for 3D-FRONT scenes, demonstrating significant improvements in visual quality and prompt fidelity over the prior texture generation methods.

1. Introduction

Synthesizing high-quality 3D contents is an essential yet highly demanding task for numerous applications, including gaming, film making, robotic simulation, autonomous driving, and upcoming VR/AR scenarios. With an increasing number of 3D content datasets, the computer vision and graphics community has witnessed a soaring research interest in the field of 3D geometry generation [2, 9, 31, 33, 35, 54, 61, 66]. Despite achieving a remarkable success in 3D geometry modeling, generating the object appearance, i.e. textures, is still bottlenecked by laborious human efforts. It typically requires a substantially long time for designing and adjustment, and immense 3D modelling expertise with tools such as Blender. As such, automatic designing and augmenting the textures has not yet been fully industrialized due to a huge demand for human expertise and financial expenses.

Leveraging the recent advances of 2D diffusion models, tremendous progress has been made for text-to-3D generation, especially for synthesizing textures of given shapes [5, 34, 45]. Seminal work such as Text2Tex [5] and Latent-Paint [34] have achieved great success in generating high-quality appearances for objects, facilitating high-fidelity texture synthesis from input prompts. Despite the fascinating results on objects, upscaling these methods to generating textures for an entire scene still confronts several challenges. On one hand, methods that autoregressively project 2D views to 3D object surface [5, 45] usually suffer from texture seams, accumulated artifacts, and loop closure issues. It is also quite difficult to maintain style consistency in the scene if every object is textured individually. On the other hand, score-distillation-based approaches [34] perform texture optimization in the low-resolution latent space, often resulting in blurry RGB textures and incorrect geometry details. As such, previous text-driven attempts fail to deliver high-quality textures for 3D scenes.

To address the aforementioned challenges, we propose SceneTex, a novel architecture to generate high-quality and style-consistent texture for indoor scene meshes by leveraging depth-to-image diffusion priors. Unlike previous methodologies that iteratively warp 2D views onto mesh surfaces, we take a different approach by framing the texture synthesis as a texture optimization task in RGB space via diffusion priors. At its core, we introduce a multiresolution texture field to implicitly represent the appearance of the mesh. To faithfully represent the texture details, we adopt a multiresolution texture to store texture features at multiple scales. This enables our architecture to flexibly learn both low and high frequency appearance information. To secure the style consistency of the generated texture, we incorporate a cross-attention decoder to reduce style incoherence introduced by self-occlusion. Concretely, each decoded RGB values are produced by cross-attending to the pre-sampled reference surface locations scattered across each object. This way, we further secure the global style consistency within each instance, as every visible location receives a global reference to the whole instance appearance.

We show that SceneTex has the capacity to facilitate versatile and accurate texture synthesis for indoor scenes with given language cues. We demonstrate in extensive experiments that SceneTex places a strong emphasis on both style and geometry consistency. The proposed method performs favorably against other text-driven texture synthesis methods in terms of 2D metrics such as CLIP score [43] and Inception Score [54], and user study on a subset of the 3D-FRONT dataset [16].

We summarize our technical contributions as follows:

- We design a novel framework for generating high-quality scene textures in high resolution using depth-to-image diffusion priors.
- We propose an implicit texture field to encode the object appearance at multiple scales, leveraging a multiresolution texture to faithfully represent rich texture details.
- We incorporate a cross-attention texture decoder to secure the global style-consistency for each instance, producing more visually appealing and style-consistent textures for 3D-FRONT scenes compared against previous synthesis methods.

2. Related work

Feed-forward 3D Generation. The advancement of 3D generation has adhered to the progress of 2D generative models. Adopting different backbone techniques, from variational autoencoders [29], generative adversarial networks [20], autoregressive transformers [58], to the recent diffusion models [15, 23], 3D models have been trained on 3D data of various representations, including voxels [7, 31, 50, 54, 61], point clouds [2, 33, 67], meshes [41, 66], signed distance functions [9–11, 13, 35], and more. However, unlike the ubiquity of 2D images and videos, 3D data is inherently scarce and poses significant challenges in terms of acquisition and annotation. Recent efforts have sought to address this issue by utilizing differentiable rendering techniques to learn from 2D images [1, 3, 4, 17, 21, 36, 39, 48, 49, 51, 53, 62, 63]. Although these models typically demonstrate proficiency in specific shape categories, they are incapable of handling 3D generation from free-form texts.

3D Generation with 2D diffusion models. Recent advancements in text-to-image generation, particularly diffusion models trained on large-scale image collections [15, 23, 24, 38, 47], have prompted the integration of pretrained 2D diffusion models as priors to facilitate 3D generation. Two main streams of work have emerged. The first branch directly incorporates the output of the 2D diffusion models along with the depth information. TEXTure [45] and Text2Tex [5] perform texturing on given meshes with a depth-aware variation of diffusion models. Other methods generate 3D scenes, where the geometry information is either jointly predicted [55] or obtained from off-the-shelf depth estimator [26]. The second branch of methods [6, 30, 34, 42, 59, 60] attempt to distill knowledge from pretrained 2D diffusion models with the Score Distillation Sampling (SDS) [42] technique and its subsequent improved variations in a per-prompt optimization manner. In contrast, we take advantages of the distilled knowledge from the depth-conditioned 2D diffusion priors to enable high-quality 3D texture synthesis.

3D Scene Texturing. In this work, we focus on generating high-quality textures for 3D scenes. 3D scene texturing has been studied by applying the 2D style transfer techniques [18, 19, 28] to 3D domain [8, 12, 22, 25, 27, 64]. However, these methods often emphasize low-level styles without semantic understanding. While existing 3D generation methods leveraging 2D diffusion models can theoretically be applied to 3D scene texturing, those based on inpainting [5, 45] suffer from visible seams and accumulated artifacts, while distillation-based methods [34] often

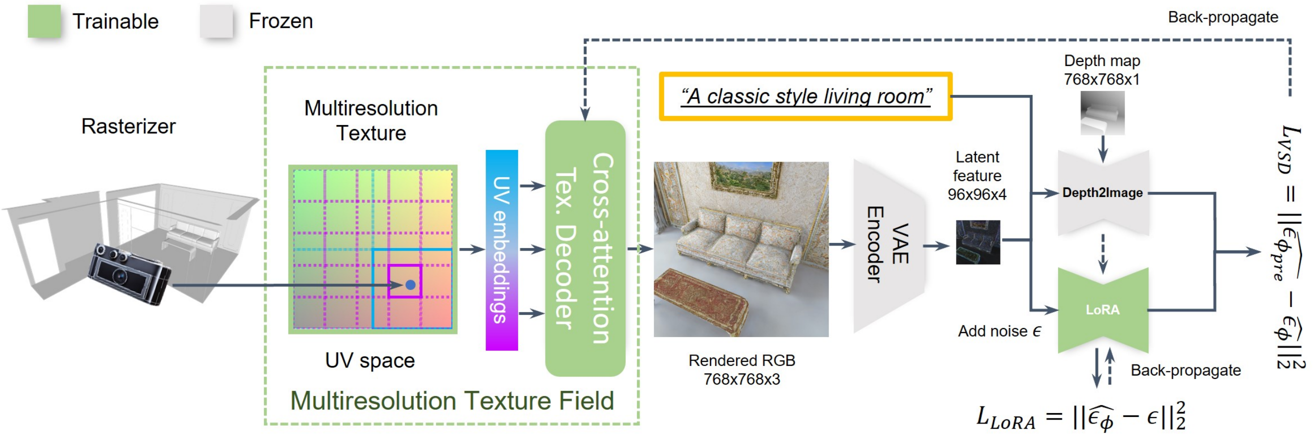

Trainable
Frozen
Back-propagate
Rasterizer
Multiresolution Texture
UV space
UV embeddings
Cross-attention Tex. Decoder
Multiresolution Texture Field
"A classic style living room"
Rendered RGB 768x768x3
VAE Encoder
Latent feature 96x96x4
Depth map 768x768x1
Depth2Image
LoRA
Add noise ϵ
Back-propagate
$L_{VSD} = ||\widehat{\epsilon_{\phi_{pre}}} - \widehat{\epsilon_\phi}||_2^2$
$L_{LoRA} = ||\widehat{\epsilon_\phi} - \epsilon||_2^2$

Figure 2. **Texture synthesis pipeline.** The target mesh is first projected to a given viewpoint via a rasterizer [32]. Then, we render an RGB image with the proposed multiresolution texture field module. Specifically, each rasterized UV coordinate is taken as input to sample the UV embeddings from a multiresoultion texture. Afterward, the UV embeddings are mapped to an RGB image of shape 768 × 768 × 3 via a cross-attention texture decoder. We use a pre-trained VAE encoder to compress the input RGB image to a 96 × 96 × 4 latent feature. Finally, the Variational Score Distillation loss [60] is computed from the latent feature to update the texture field.

produce blurry textures with incorrect geometry details. In contrast, we optimize the target scene texture with accurate geometric cues and decode the high-resolution scene appearance via the proposed multiresolution texture field module, facilitating 3D scene texture synthesis with much better visual quality.

3. Method

The objective of our work is to texture an entire 3D scene with diffusion priors as the critic. In this section, we begin by introducing a Multiresolution Texture Field module to produce high-quality RGB textures, which consists of two key components: Multiresolution Texture and Cross-attention Texture Decoder. The Multiresolution Texture is integrated to faithfully represent both the low- and high-frequency texture details at various scales (Sec. 3.1). Subsequently, to tackle the style-inconsistency issue brought by limited field of view and self-occlusion, the Cross-attention Texture Decoder module is incorporated to enforce a global style-awareness for each object in the scene (Sec. 3.2). Finally, we adopt a pretrained diffusion model as training critic to dynamically distillate realistic scene appearance from the 2D depth-conditioned diffusion priors. (Sec. 3.3). The entire synthesis architecture is presented in Fig. 2.

3.1. Multiresolution Texture Field

The core of texture synthesis with 2D priors lies in generating RGB values visible to a series of pre-defined viewpoints. Previous methods maintain a 64 × 64 × 4 latent map and operate directly on it with the SDS loss [34, 42]. This latent map is decoded via the variational autoencoder of the pre-trained diffusion model after convergence. The optimization process is technically view-consistent, as the diffusion priors are leveraged from numerous perspectives. Notwithstanding, we observe that the decoded RGB textures often carry patch-like artifacts and are subsequently inconsistent with the given geometry. This is caused by the mismatches between the low-resolution latent map and high-resolution RGB images, and the lack of perspective transformation of the same latent code in different views.

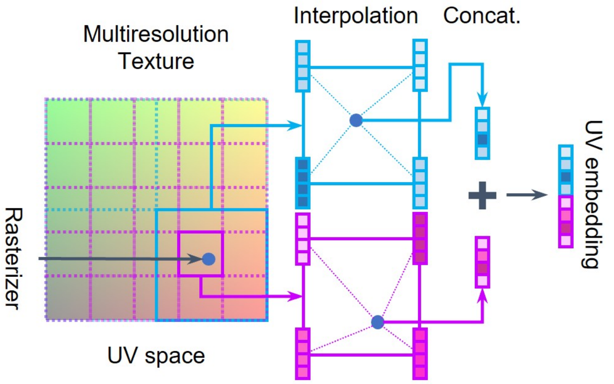


Figure 3. **Multiresolution Texture.** We use a multiresolution feature grid to encode positional features at different scale in the UV space. For a query UV coordinate, we interpolate the grid features at respective resolutions. The interpolated grid features are concatenated as the final UV embedding for the query UV coordinate.

To tackle those inherent disadvantages of representing the target texture via a low-resolution latent map, we adopt an implicit texture field that queries the texture features with given UV coordinates. At its core, we integrate a multiresolution texture to prevent oversimplified appearance with-

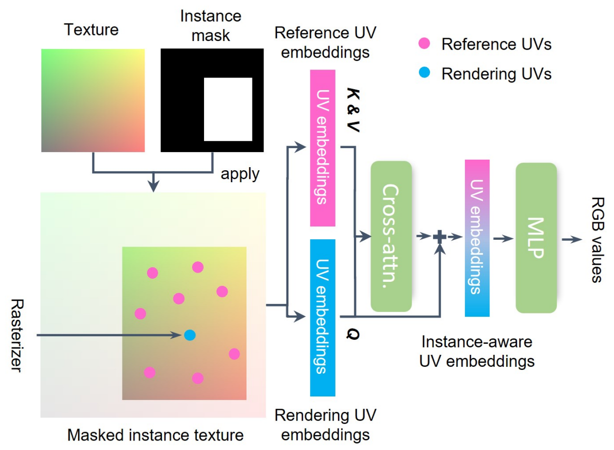


Figure 4. **Cross-attention Texture Decoder.** For each rasterized UV coordinate, we apply a UV instance mask to mask out the corresponding instance texture features. Then, we obtain the rendering UV embeddings for the rasterized locations in the view. At the same time, we extract the texture features for the pre-sampled UVs scattered across this instance as the reference UV embeddings. We deploy a multi-head cross-attention module to produce the instance-aware UV embeddings. Here, we treat the rendering UV embeddings as the Query, and the reference UV embeddings as the Key and Value. Finally, a shared MLP maps the instance-aware UV embeddings to RGB values in the rendered view.

out any texture details. In particular, as shown in Fig. 3, we encode texture features for all query locations q at each scale, and concatenate those features as the output UV embeddings $\mathcal{E}(q)$ to faithfully represent all texture details. The UV embeddings are then decoded to the final RGB texture by the cross-attention texture decoder introduced in the next section.

3.2. Cross-attention Texture Decoder

Since the texture is optimized in image space, instance textures are often constrained by limited field of view and self-occlusion. As a result, the optimized texture often suffers from style-inconsistency. Therefore, we propose a simple yet effective rendering module with global instance awareness to predict RGB values from UV embeddings. This is done by incorporating a multi-head cross-attention module to the texture features. As Fig 4 illustrates, for each rasterized UV coordinate, we apply a UV instance mask to mask out the corresponding instance texture features. Then, we obtain the rendering UV embeddings for the rasterized locations in the view. At the same time, we extract the texture features for the pre-sampled UVs scattered across this instance as the reference UV embeddings. We deploy a multi-head cross-attention module to produce the instance-aware UV embeddings. Here, we treat the rendering UV embeddings as the Query, and the reference UV embeddings as the Key and Value. Finally, a shared MLP maps the global-aware UV embeddings to RGB values in the rendered view. We denote the whole rendering process as $\mathcal{C} = f(\mathcal{E}(q);\theta)$, where $\mathcal{C}$ represents an RGB image at arbitrary resolution, $f(\theta)$ is a differentiable function resembles the entire texture field with trainable parameters θ.

3.3. Texture Field Optimization via VSD

We adopt a pre-trained ControlNet model as a critic to optimize the texturing module $f(\theta)$ following the strategy of Latent-Paint [34], as shown in Fig. 2. Here, the UNet of a pre-trained latent diffusion model (LDM) [46] applied to calculate the gradients based on a low-resolution 96 × 96 latent map. We observe that such low-resolution rendering often lead to broken visual quality and unsatisfactory view consistency. This is primarily due to the size mismatches between the 96 × 96 optimization target and the final 768 × 768 RGB output. Additionally, prior work exclude geometric cues from the diffusion priors, resulting in poor consistency between the generated textures and target geometry. To address those issues, we directly render an 768 × 768 RGB image via querying the texture field $\mathcal{C} = f(\mathcal{E}(q);\theta)$. In each iteration, we first optimize $\mathcal{C}$ via the VSD objective [60] with a pre-trained frozen depth-conditioned diffusion prior ϕ_{pre} and a trainable LoRA module ϕ:

$$\mathcal{L}_{\text{VDS}}(\theta) := \mathbb{E}_{t,\epsilon}[w(t)(\epsilon_{\phi_{\text{pre}}} - \epsilon_{\phi})\frac{\partial f(\theta)}{\partial \theta}] \tag{1}$$

where $\epsilon_{\phi_{\text{pre}}} = \phi_{\text{pre}}(f(\theta);y,d,t)$ and $\epsilon_{\phi} = \phi(f(\theta);y,d,t))$. We draw time step randomly by $t \sim \mathcal{U}(0.02, 0.98)$. The injected noise is $\epsilon \sim \mathcal{N}(0,1)$. d is the depth map in the current viewpoint produced by the rasterizer. y is the noised input to the UNet. The weighting function $w(t)$ is empirically set as $w(t) = \sqrt{1-\prod_{s=1}^{t}\alpha_s}$. Note that ϕ_{pre} and ϕ are kept frozen when updating the parameters of the texture field θ. After $\mathcal{C} = f(\mathcal{E}(q);\theta)$ is updated via the gradients of VSD, we unfreeze and update the LoRA module ϕ:

$$\mathcal{L}_{\text{LoRA}}(\phi) = \min_{\phi} \sum_{i=1}^{n} \mathbb{E}_{t,\epsilon}[||\epsilon_{\phi}(f(\theta);y,d,t) - \epsilon||_2^2] \tag{2}$$

3.4. Inference

Since the texture field $f(\theta)$ only receives the UV coordinates as input, producing the final RGB texture is straightforward. For each position q_i in UV space, the corresponding pixel value c_i of the output RGB texture $\mathcal{C}$ can be simply queried by $c_i = f(\mathcal{E}(q_i);\theta)$. Thanks to the multiresolution grid encoding, it is worth mentioning that there is no specification for the size of the final RGB texture, i.e. the resolution of the texture can be adjusted according to the computational resources.

4. Results

4.1. Implementation Details

We apply the ControlNet Depth model [65] for VSD optimization. The multiresolution texture is implemented by multiresolution hash encoding [37]. During each optimization iteration, we randomly pick 1 viewpoint from all perspective points scattered across the scene. We set the learning as 0.001 for optimizing the texture field and 0.0001 for fine-tuning the LoRA module. The entire optimization uses $5,000$ viewpoints and takes $30,000$ iterations to converge. To generate a more visually appealing appearance via VSD, We adopt the time annealing scheme following ProlificDreamer [60], where we sample time steps $t \sim \mathcal{U}(0.02, 0.98)$ for the first $5,000$ steps and then anneal into $t \sim \mathcal{U}(0.02, 0.50)$ for the rest of the optimization. For the proposed cross-attention decoder, we pre-sampled $4,096$ UV coordinates scattered across each instance. To enable cross-attention in such a long context, we implement the cross-attention module with Flash Attention v2 [14]. Each synthesis process takes around 20 hours to converge on an NVIDIA RTX A6000. After convergence, we generate a high-resolution $4,096 \times 4,096$ RGB image as the final scene texture. Our implementation uses the PyTorch [40] framework, with PyTorch3D [44] for rendering and texture projection.

4.2. Quantitative Analysis

We compare our method against texture synthesis methods appeared recently in Tab. 1, including Latent-Paint [34], MVDiffusion [57], and Text2Tex [5]. We experiment all methods on 10 3D-FRONT [16] scenes with 2 different text prompts for each scene. Here, we calculate CLIP score (CLIP) [43] and Inception Score (IS) [54] to measure the fidelity with input prompts and texture quality, respectively. Our method outperforms all baselines on the 2D automated metrics by a significant margin. We additionally report the User Study results from 75 participants about the Visual Quality (VQ) and Prompt Fidelity (PF) on a scale of 1-5. Our method is shown to be more favored by human users.

4.3. Qualitative Results

We show the qualitative comparisons in Fig. 5. Latent-Paint suffers from the over-saturation issue and hallucinates non-existing objects, such as the huge frame on the wall (see the first example in the first row in Fig. 5). Those unrealistic texture components are produced by the inaccurate geometric cues and the mismatch between the optimized latent representation and final texture. MVDiffusion [57] produces overall smooth but blurry and dimmed texture. It also fails to reflect the iconic properties in the prompts, such as "baroque" and "luxury". Text2Tex [5] generates plausible textures for individual objects, but fails to achieve global style consistency across objects. In contrast, our method synthesizes high-quality with overall coherent styles within and across objects, reflecting the representative traits in the prompts with high-fidelity (see the baroque paintings above and the golden pillows below). We additionally visualize our texture synthesis results for different 3D-FRONT [16] scenes and input prompts in top-down views and close-up renderings in Fig. 6, further demonstrating the supreme texture quality and fidelity produced by our method.

Method	2D Metrics		User Study	
	CLIP ↑	IS ↑	VQ ↑	PF ↑
Latent-Paint [34]	18.37	1.96	1.57	2.11
MVDiffusion [57]	18.47	2.83	3.09	3.12
Text2Tex [5]	20.83	2.87	2.62	3.04
SceneTex (Ours) w/o texture field	15.77	1.56	1.23	1.11
SceneTex (Ours) w/o multires. tex.	19.87	2.79	2.11	2.39
SceneTex (Ours) w/ cross-attn.	20.94	3.29	3.94	4.05
SceneTex (Ours)	**22.18**	**3.33**	**4.40**	**4.29**

Table 1. **Quantitative comparisons.** We report the 2D metrics and User Study results for quantitative comparisons, including: CLIP score (CLIP) [43], Inception Score (IS) [54], Visual Quality (Visual Quality), and Prompt Fidelity (PF). We show that our method produces textures with the highest quality.

4.4. Ablation Studies

We conduct ablation experiments on the key components of our method, including multiresolution texture (Sec. 3.1), and cross-attention decoder (Sec. 3.2). All comparisons are shown in Tab. 1 and Fig. 7.

Does texture field produce better textures than RGB tensors? Since only a few UVs are sampled by the rasterizer during each iteration, directly optimizing an RGB tensor as the output texture leads to noisy artifacts, as shown in the first column in Fig. 7. Additionally, the optimization is often difficult to converge with different gradient scales across the whole RGB tensor. As a result, the optimized texture appears to be broken and unrealistic. In contrast, the MLP in the proposed texture field module effectively smoothens the back-propagated gradients, producing much smoother and detailed textures.

Does multiresolution texture improve the visual quality? As previous studies indicate, implicit representations via MLPs tend to learn low-frequency information [52, 56]. As such, the texture field with single resolution produces an over-simplified appearance without texture details. Such texture lacks the characteristic properties of the input prompt, and carries noisy bubble artifacts, as shown in the second column in Fig. 7. We show that the multiresolution texture is capable of producing a visually appealing and

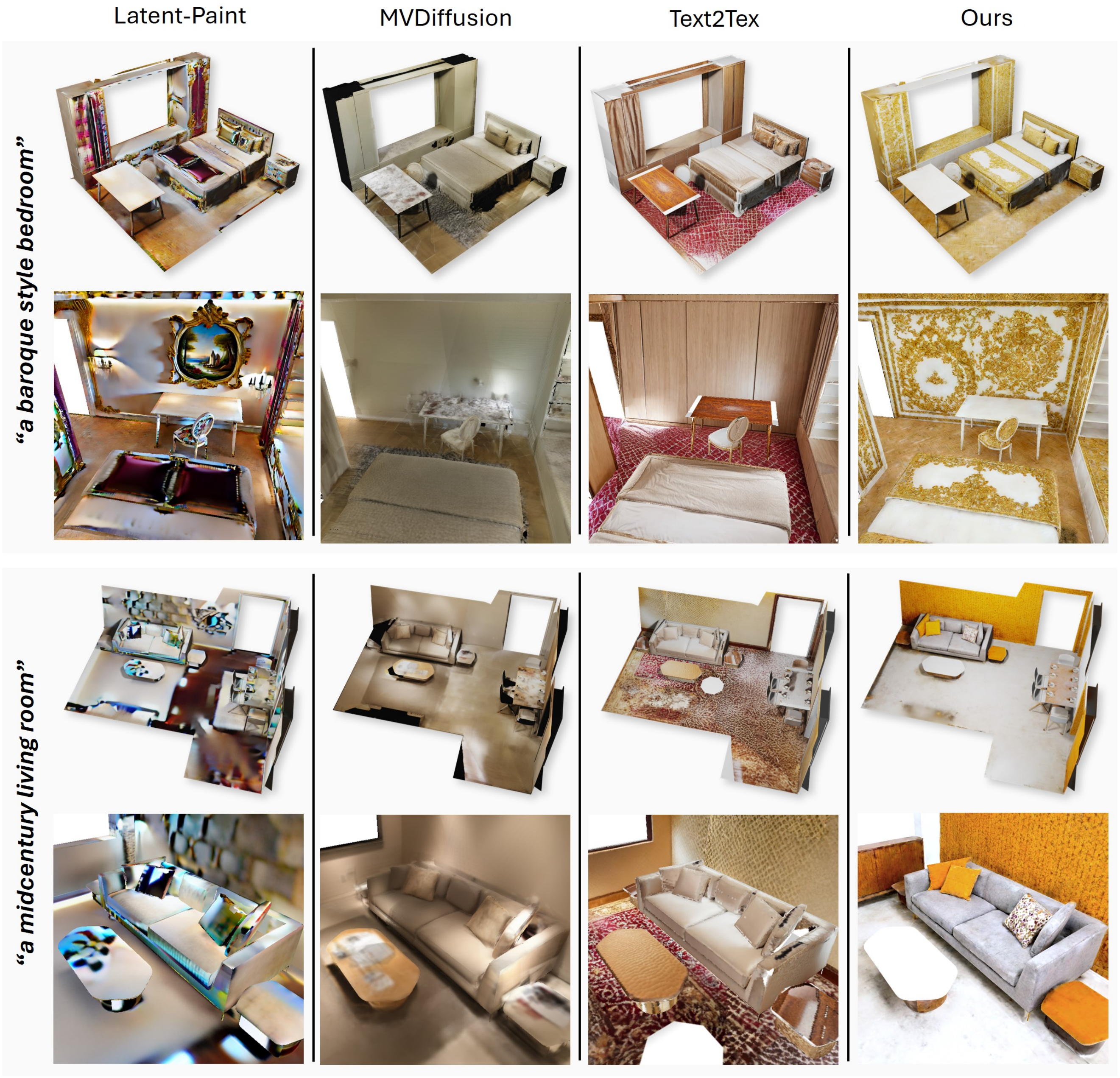
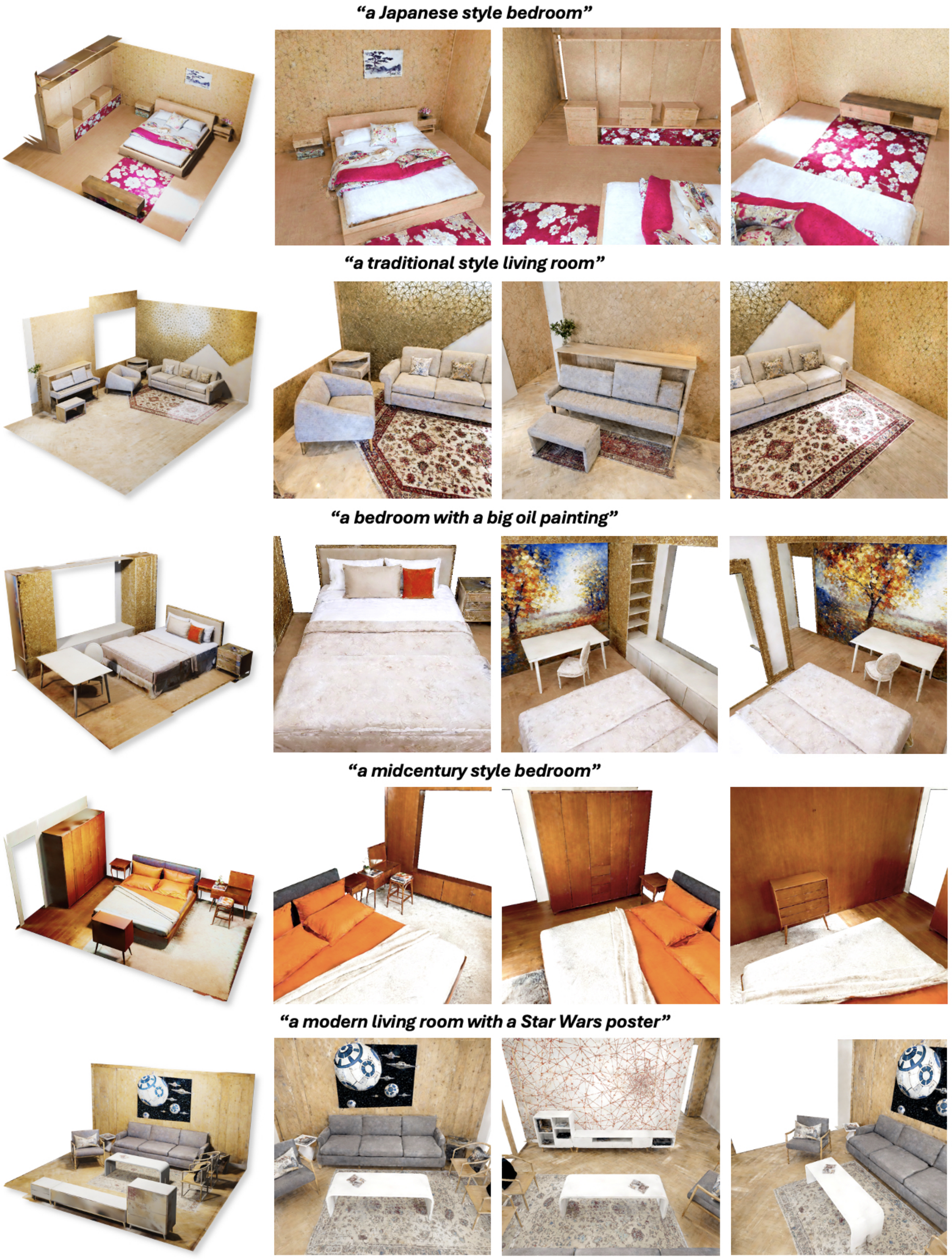


Figure 5. **Qualitative comparisons.** Latent-Paint [34] suffers from over-saturation and hallucinates scene components. MVDiffusion [57] delivers blurry textures and fails to reflect the input prompts. Text2Tex [5] struggles to keep all instances style-consistent. In contrast, our method produces high-quality textures and maintains overall style-consistency across instances in the scenes. Ceilings and back-facing walls are excluded for better visualizations. Images best viewed in color.

highly detailed mesh appearance.

Does cross-attention strengthen the style consistency? Replacing the cross-attention decoder module with a simple MLP also produces plausible textures. However, such replacement exposes global style inconsistency issue. Due to limited field of view and self-occlusion, the appearance of the same object can be synthesized differently. As shown in Fig. 7, big objects such as the carpet do not share a coherent pattern. It is difficult for the big objects to maintain style consistency during optimization, if there is no global information shared across views. The proposed cross-attention decoder effectively tackles this issue by globally sharing the style features within each object. This enforces the in-



"a Japanese style bedroom"

"a traditional style living room"

"a bedroom with a big oil painting"

"a midcentury style bedroom"

"a modern living room with a Star Wars poster"

Figure 6. **Synthesized textures for 3D-FRONT scenes.** Our method generates high-quality style-coherent textures, and reflects the iconic traits in the prompts. Ceilings and back-facing walls are excluded for better visualizations. Images best viewed in color.

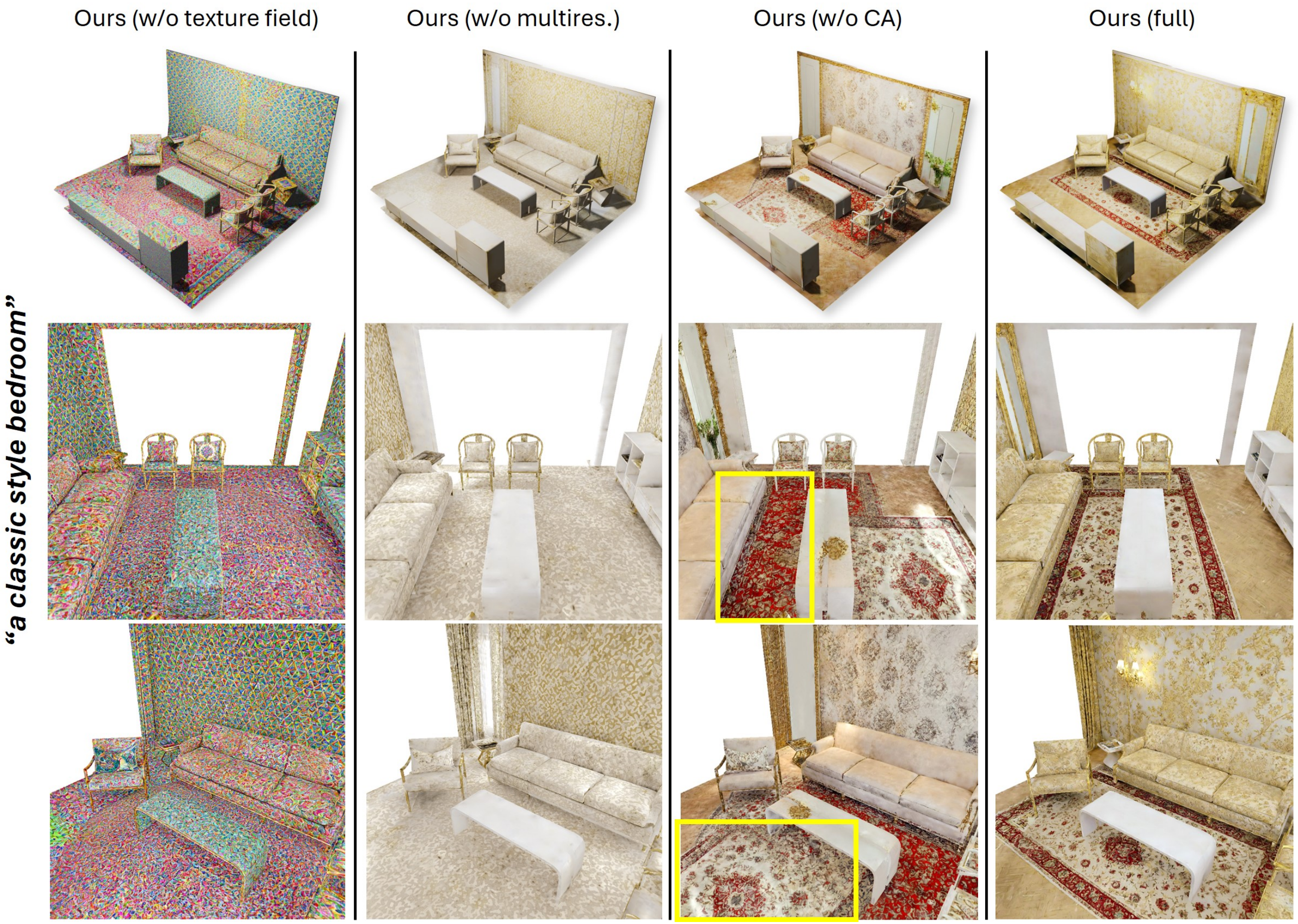


Figure 7. **Ablation studies on the key components.** Optimizing an RGB texture directly without the proposed texture field results in extreme noisy and unrealistic textures. A single-resolution texture fails to capture texture details and produces bubble artifacts. Removing the cross-attention decoder leads to style inconsistency, especially for big instances such carpet, as shown in the yellow boxes. In contrast, our full method produces high-quality and style-consistent textures without aforementioned artifacts.

stance style awareness, and therefore produces more style-consistent textures for all instances across the scene.

4.5. Limitations

Although our method enables high-quality texture synthesis for indoor scenes, we still notice that our method tends to generate textures with shading effects. This phenomenon becomes more obvious when the scene structure indicates the existence of lighting such as lamp, window, or even mirror. We believe this issue can be properly addressed by carefully fine-tuning the diffusion priors on the indoor scene images without shading effects. We acknowledge this challenge and leave it to future research.

5. Conclusion

We introduce SceneTex, a novel method for effectively generating high-quality and style-consistent textures for indoor scenes using depth-to-image diffusion priors. At its core, SceneTex proposes a multiresolution texture field to implicitly encode the mesh appearance. We optimize the target texture via a score-distillation-based objective function in respective RGB renderings. To further secure the style consistency across views, we introduce a cross-attention decoder to predict the RGB values by cross-attending to the pre-sampled UV coordinates within each instance. We show that the proposed texture field with multiresolution texture is capable of generating visually appealing high-quality texture. Moreover, the proposed cross-attention decoder further strengthens the global style awareness for each instance, resulting in style-coherent appearance in the target scene. Extensive analysis show that SceneTex enables various and accurate texture synthesis for 3D-FRONT scenes, demonstrating significant improvements in visual quality and prompt fidelity over the prior texture generation methods. Overall, we hope our work can inspire more future research in the area of text-to-3D generation.

Acknowledgements

This work was supported by a gift by Snap Inc., the ERC Starting Grant Scan2CAD (804724), and the German Research Foundation (DFG) Research Unit "Learning and Simulation in Visual Computing". We thank Angela Dai for the video voiceover.

References

[1] Rameen Abdal, Hsin-Ying Lee, Peihao Zhu, Menglei Chai, Aliaksandr Siarohin, Peter Wonka, and Sergey Tulyakov. 3davatargan: Bridging domains for personalized editable avatars. In *CVPR*, 2023. 2

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, 2018. 1, 2

[3] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. 2

[4] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2

[5] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023. 1, 2, 5, 6

[6] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *ICCV*, pages 22246–22256, 2023. 2

[7] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. 2019. 2

[8] Yaosen Chen, Qi Yuan, Zhiqiang Li, Yuegen Liu, Wei Wang, Chaoping Xie, Xuming Wen, and Qien Yu. Upst-nerf: Universal photorealistic style transfer of neural radiance fields for 3d scene. *arXiv preprint arXiv:2208.07059*, 2022. 2

[9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 1, 2

[10] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander Schwing, and Liangyan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *CVPR*, 2023.

[11] Zezhou Cheng, Menglei Chai, Jian Ren, Hsin-Ying Lee, Kyle Olszewski, Zeng Huang, Subhransu Maji, and Sergey Tulyakov. Cross-modal 3d shape generation and manipulation. In *ECCV*, 2022. 2

[12] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. 2022. 2

[13] Angela Dai, Yawar Siddiqui, Justus Thies, Julien Valentin, and Matthias Nießner. Spsg: Self-supervised photometric scene generation from rgb-d scans. In *CVPR*, 2021. 2

[14] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 5, 12, 13

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. 2021. 2

[16] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 2, 5, 12

[17] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022. 2

[18] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 2

[19] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *CVPR*, 2017. 2

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 2

[21] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *ICLR*, 2022. 2

[22] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 2

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 2020. 2

[24] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021. 2

[25] Lukas Höllein, Justin Johnson, and Matthias Nießner. Stylemesh: Style transfer for indoor 3d scene reconstructions. In *CVPR*, 2022. 2

[26] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *ICCV*, 2023. 2

[27] 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. 2

[28] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[30] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 2

[31] Chieh Hubert Lin, Hsin-Ying Lee, Willi Menapace, Menglei Chai, Aliaksandr Siarohin, Ming-Hsuan Yang, and Sergey Tulyakov. Infinicity: Infinite-scale city synthesis. In *ICCV*, 2023. 1, 2

[32] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 3

[33] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021. 1, 2

[34] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, 2023. 1, 2, 3, 4, 5, 6

[35] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3d completion, reconstruction and generation. In *CVPR*, 2022. 1, 2

[36] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4328–4338, 2023. 2

[37] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 5, 12

[38] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. 2021. 2

[39] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[41] Dario Pavllo, Graham Spinks, Thomas Hofmann, Marie-Francine Moens, and Aurelien Lucchi. Convolutional generation of textured 3d meshes. 2020. 2

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

[43] 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 *ICLR*, 2021. 2, 5

[44] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

[45] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 1, 2

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

[47] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE TPAMI*, 2022. 2

[48] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. 2020. 2

[49] Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Kyle Olszewski, Hsin-Ying Lee, Jian Ren, Menglei Chai, and Sergey Tulyakov. Unsupervised volumetric animation. In *CVPR*, 2023. 2

[50] Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Kyle Olszewski, Jian Ren, Hsin-Ying Lee, Menglei Chai, and Sergey Tulyakov. Unsupervised volumetric animation. In *CVPR*, 2023. 2

[51] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces. In *ECCV*, 2022. 2

[52] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 5

[53] Ivan Skorokhodov, Aliaksandr Siarohin, Yinghao Xu, Jian Ren, Hsin-Ying Lee, Peter Wonka, and Sergey Tulyakov. 3d generation on imagenet. In *ICLR*, 2023. 2

[54] Edward J Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. In *Conference on Robot Learning*, pages 87–96. PMLR, 2017. 1, 2, 5

[55] Gabriela Ben Melech Stan, Diana Wofk, Scottie Fox, Alex Redden, Will Saxton, Jean Yu, Estelle Aflalo, Shao-Yen Tseng, Fabio Nonato, Matthias Muller, et al. Ldm3d: Latent diffusion model for 3d. *arXiv preprint arXiv:2305.10853*, 2023. 2

[56] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 5

[57] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023. 5, 6

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

[59] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, 2023. 2

[60] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 2, 3, 4, 5

[61] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. Learning descriptor

networks for 3d shape synthesis and analysis. In *CVPR*, 2018. 1, 2

[62] Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Ivan Skorokhodov, Aliaksandr Siarohin, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, et al. Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. In *CVPR*, 2023. 2

[63] Rui Yu, Yue Dong, Pieter Peers, and Xin Tong. Learning texture generators for 3d shape collections from internet photo sets. In *BMVC*, 2021. 2

[64] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *ECCV*, 2022. 2

[65] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 5

[66] Song-Hai Zhang, Yuan-Chen Guo, and Qing-Wen Gu. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *CVPR*, 2021. 1, 2

[67] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021. 2

Supplementary Material

In this supplementary material, we present additional results and analysis in Sec. A to show the effectiveness of the proposed texture synthesis method. We also provide details of the user study in Sec. B. For reproducibility, we produce the architecture details of the proposed Cross-attention Texture Decoder in Sec. C, and the hash encoding [37] configurations of the multiresolution texture in Sec. D.

A. Additional Qualitative Results

To further showcase the effectiveness of the proposed method, we present more texture synthesis results on 3D-FRONT dataset [16].

Realistic indoor scene decorations. Our method is capable of generating high-quality realistic scene appearance for most common indoor decoration styles, such as "Scandinavian" and "luxury", as shown in Fig. 9. Additionally, our method can move beyond common indoor styles to more challenging ones, such as "Lego" and "Game of Throne", as shown in Fig. 10. It demonstrates the flexibility of our method for synthesizing high-quality texture while loyally reflecting iconic properties in the input prompts. We believe that those synthesis results indicate a great potential to stylize more high-quality 3D textures for indoor scene.

Stylizing the same scene. In Fig. 11, we show that our method can generate different texturing results on the same objects. Here, we use the same prompt template "a ⟨STYLE⟩ living room" with 4 different styles ("Bohemian", "Baroque", "French country", and "Japanese"). All our textures are highly detailed and loyal to the style, demonstrating a great potential to generate diverse and various 3D scenes appearances.

B. User Study Details

We develop a Django-based web application for the user study. In Fig. 12, we show the interface for the questionnaire. We randomly select textured scenes from each baseline and our method to form a batch of 8 samples. To better visualize the textured scene, we render multi-view images for those scenes from 10 different viewpoints. After presenting the 10 rendered images, we ask the users to rate the appearance of the scenes from 1 to 5 in terms of aesthetics, realism, smoothness, etc. We also ask the users to measure how well the scene appearances match the given descriptions. To avoid biases and cheating in this user study, we shuffle scenes so that there is no positional hint of our method. In the end, we gather 100 responses from 75 participants to calculate the user study results.

Parameter	Value
Number of levels	20
Hash table size	2^{24}
Number of feature dimensions	4
Min. resolution	16^2
Max. resolution	4096^2

Table 2. **Multiresolution texture configuration.** We configure the hash encoding parameters following iNGP [37].

C. Cross-attention Texture Decoder

We implemented a neural texture decoder with cross-attention mechanism to strengthen the style-consistency within each object in the scene. As Fig. 8 shows, we map the UV embeddings with 2-layer MLPs in a 4-head cross-attention module implemented with Flash Attention v2 [14]. A skip connection is applied between the original rendering UV embeddings and the output features as the final instance-aware UV embeddings. The final RGB renderings are produced by a 4-layer MLP with hidden size 256.

D. Multiresolution Texture Configuration

In Tab. 2, we illustrate the implementation details of our multiresolution texture for reproducibility, following the configuration in iNGP [37]. Here, we use a deep hierarchy with 20 layers of texture features to learn scene appearance at various scales. This ensures our synthesis architecture to faithfully encode detailed texture information irrespective of the distance between the viewpoint and mesh surface. To reduce possible hash collision in UV space, we choose a comparatively big hash table size of 2^{24}. Our multiresolution texture encodes texture features from 16×16 to $4,096 \times 4,096$.

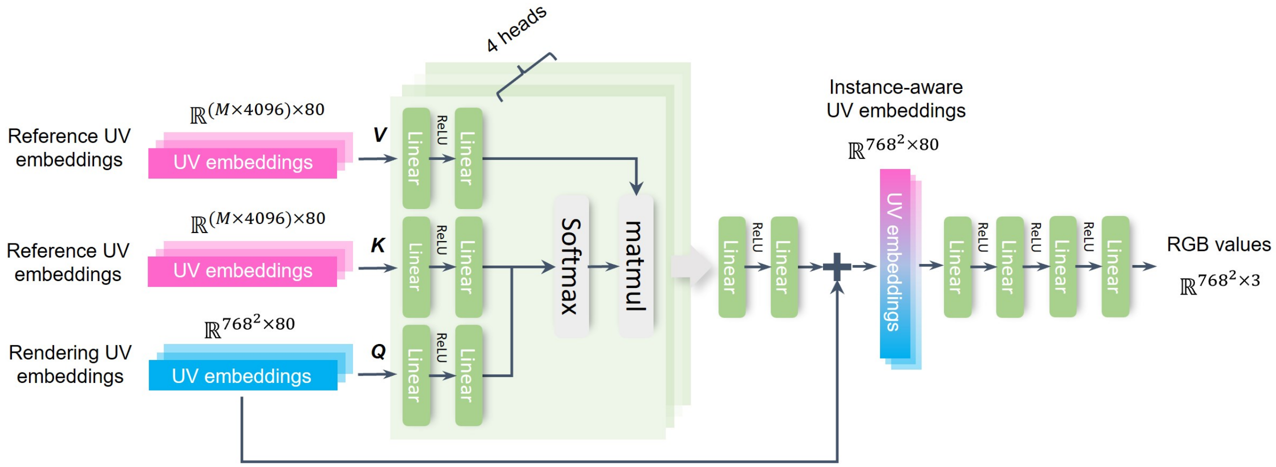
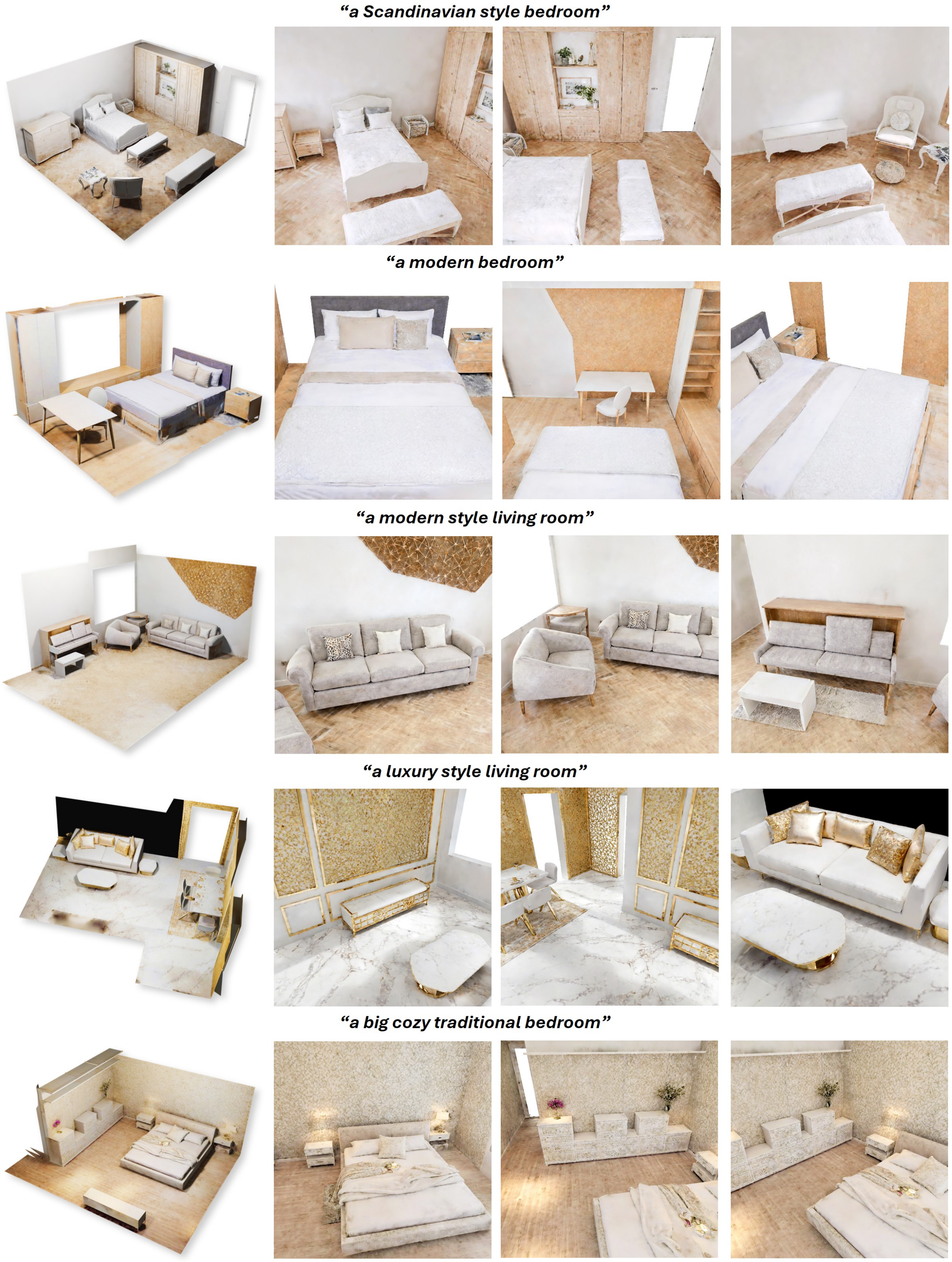

4 heads
Reference UV embeddings
$\mathbb{R}^{(M\times4096)\times80}$
UV embeddings
V
Reference UV embeddings
$\mathbb{R}^{(M\times4096)\times80}$
UV embeddings
K
Rendering UV embeddings
$\mathbb{R}^{768^2\times80}$
UV embeddings
Q
Linear
ReLU
Linear
Softmax
matmul
Instance-aware UV embeddings
$\mathbb{R}^{768^2\times80}$
UV embeddings
RGB values
$\mathbb{R}^{768^2\times3}$

Figure 8. **Architecture details for Cross-attention Texture Decoder.** For the target 768×768 image, we directly obtain the rendering UV embeddings in shape $\mathbb{R}^{768^2\times80}$. Then, for the current scene with M objects, we query the reference UV embeddings in shape $\mathbb{R}^{(M\times4096)\times80}$ of the pre-sampled $4,096$ positions for each object. We apply 2-layer MLPs to map the UV embeddings to the Key, Value, and Query for the cross-attention module. We use a 4-head cross-attention module implemented with Flash Attention v2 [14]. The output features are mapped with a 2-layer MLP, and added to the original rendering UV embeddings as the final instance-aware UV embeddings. Finally, a 4-layer MLP predicts the RGB renderings of the current viewpoint.



“a Scandinavian style bedroom”

“a modern bedroom”

“a modern style living room”

“a luxury style living room”

“a big cozy traditional bedroom”

Figure 9. **Synthesized scene textures with modern styles.** Our method is capable of generating high-quality scene appearance for common indoor decoration styles. Ceilings and back-facing walls are excluded for better visualizations. Images best viewed in color.

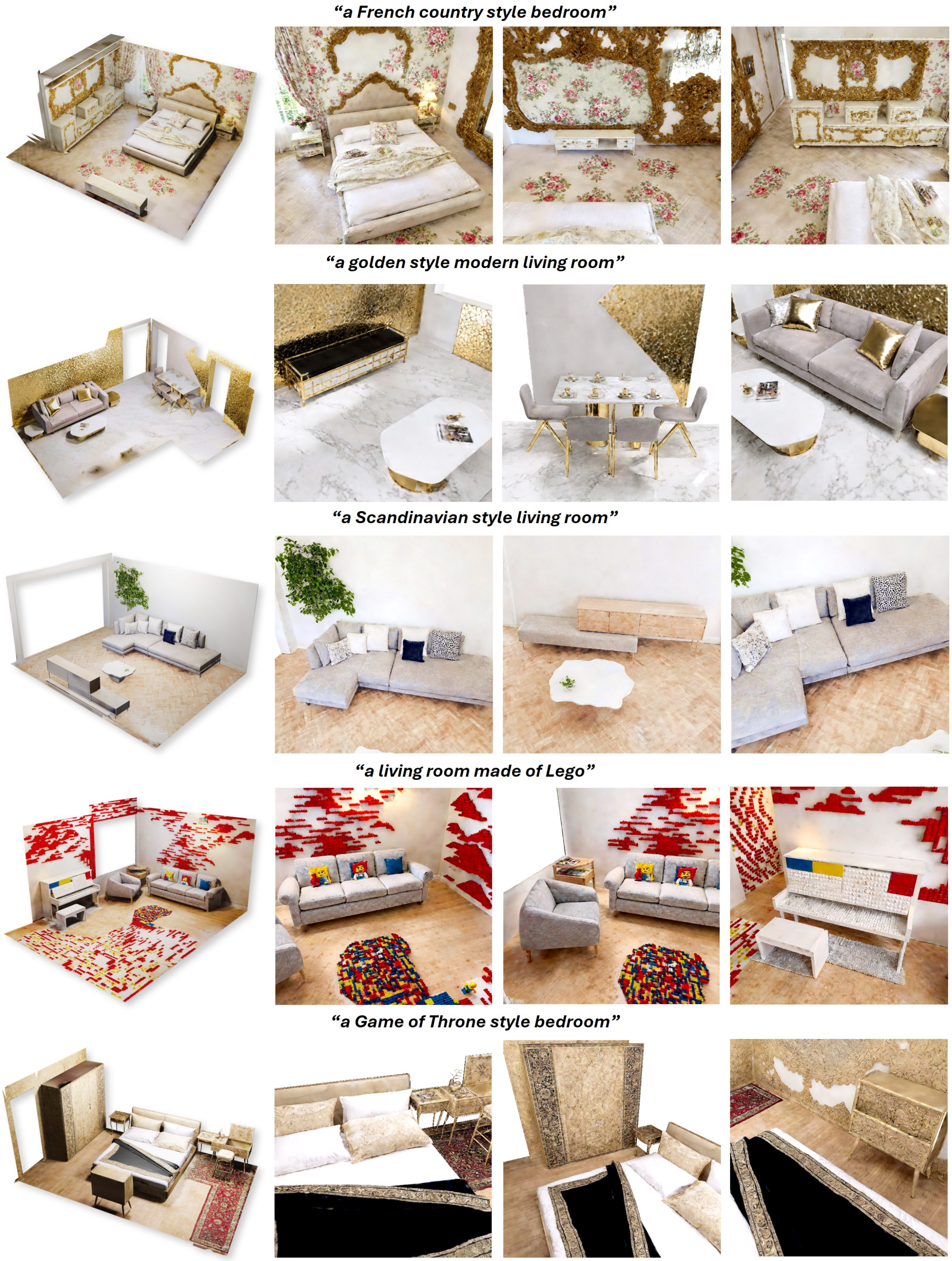

“a French country style bedroom”
“a golden style modern living room”
“a Scandinavian style living room”
“a living room made of Lego”
“a Game of Throne style bedroom”

Figure 10. **Synthesized scene textures with creative styles.** Our method can also generate textures for challenging and creative scene styles. Ceilings and back-facing walls are excluded for better visualizations. Images best viewed in color.

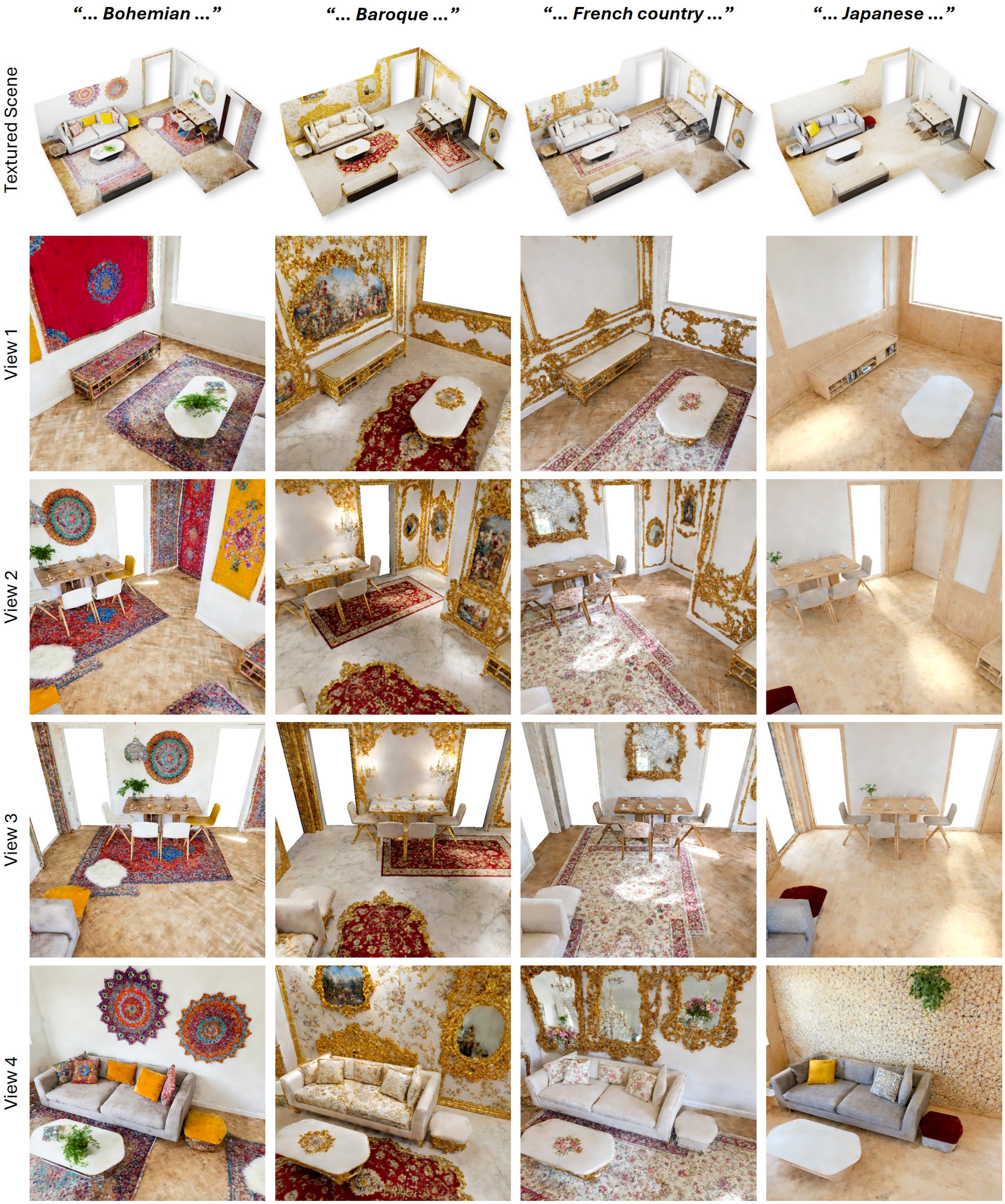
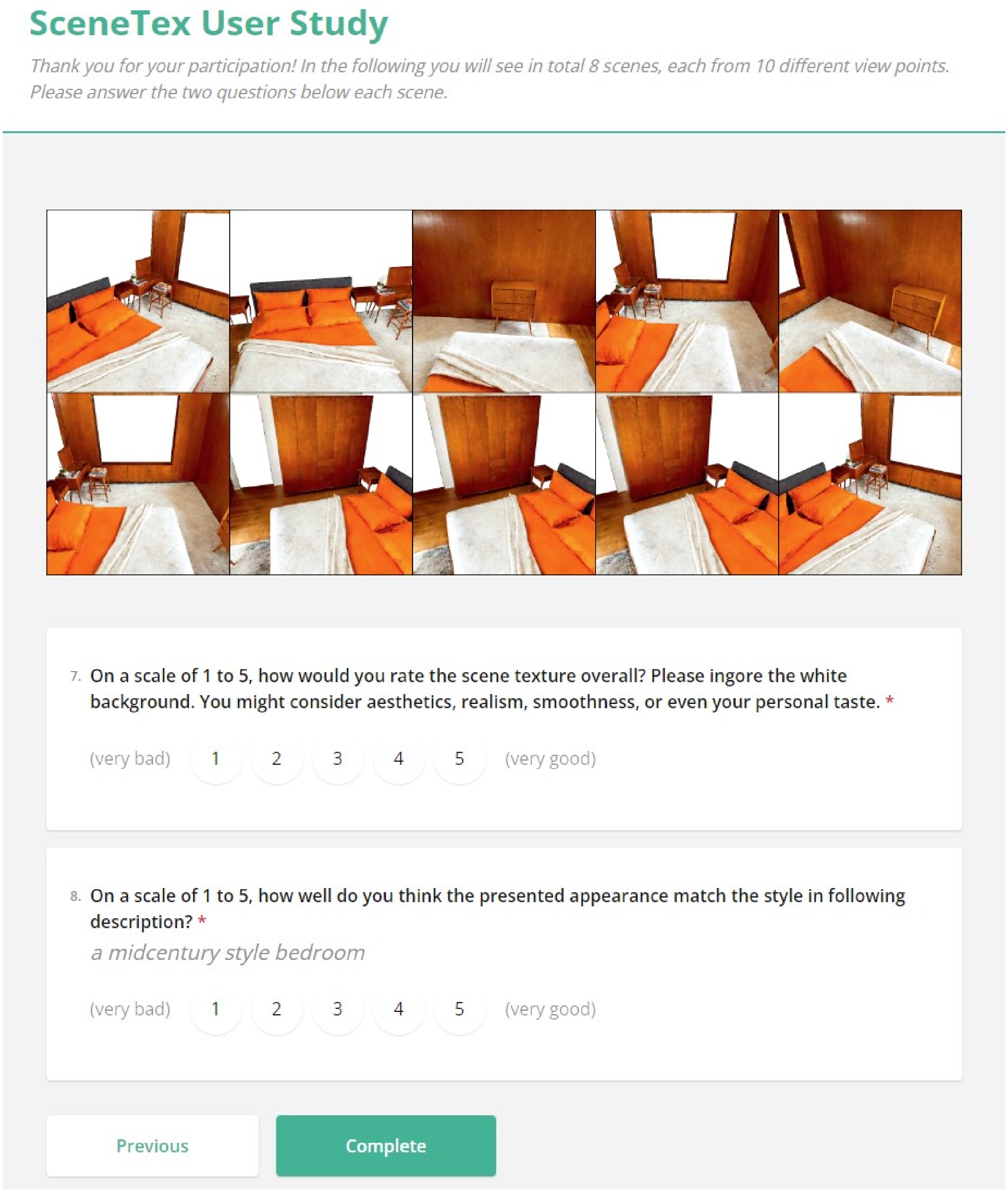


Figure 11. **Synthesized textures for 3D-FRONT scenes.** Our method generates different textures for the same input scene. We use the prompt template "a ⟨STYLE⟩ living room" with 4 different styles for texture generation.



SceneTex User Study

Thank you for your participation! In the following you will see in total 8 scenes, each from 10 different view points. Please answer the two questions below each scene.

7. On a scale of 1 to 5, how would you rate the scene texture overall? Please ingore the white background. You might consider aesthetics, realism, smoothness, or even your personal taste. *

(very bad) 1 2 3 4 5 (very good)

8. On a scale of 1 to 5, how well do you think the presented appearance match the style in following description? *

a midcentury style bedroom

(very bad) 1 2 3 4 5 (very good)

Figure 12. **Screenshot of the user study interface.** We present 10 rendered views from 8 different texturing results to each human user, and ask them to rate the appearance and the similarity with input prompts on a scale of 1-5.