# Paint3D: Paint Anything 3D with Lighting-Less Texture Diffusion Models

## Appendix

This appendix provides more qualitative results (Sec. A), several additional experiments (Sec. B), discussion on the failure cases of our proposed texture generation approach (Sec. C), and implementation details(Sec. D).

## A. Qualitative Results

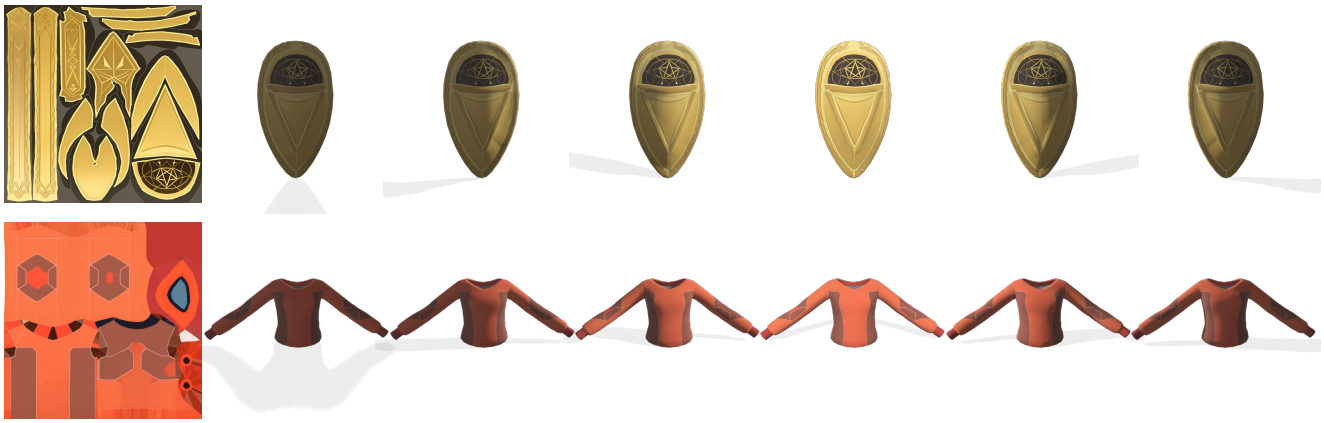



Figure 10. Lighting-less texture maps generated by Paint3D. These lighting-less textures produce appropriate shadows when the textured meshes are illuminated from different directions of light sources.

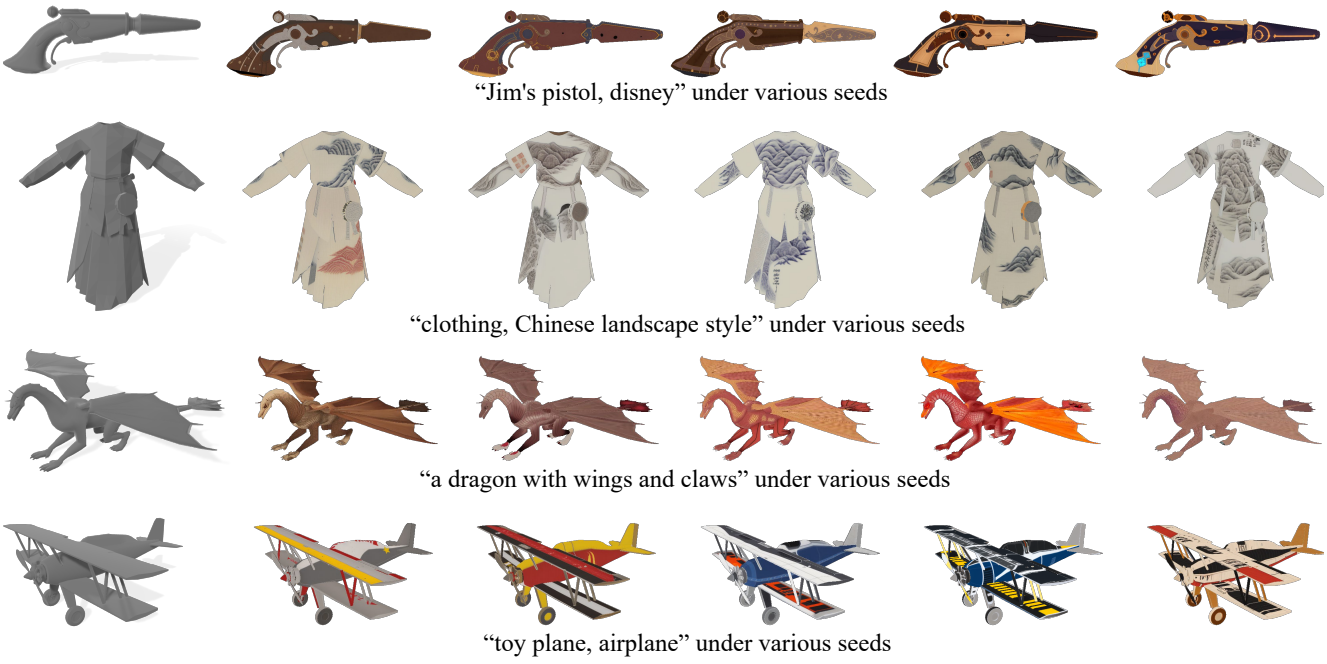



Figure 11. More samples from our best model for text-to-texture generation. Samples are generated with text prompts of the test set under various seeds.

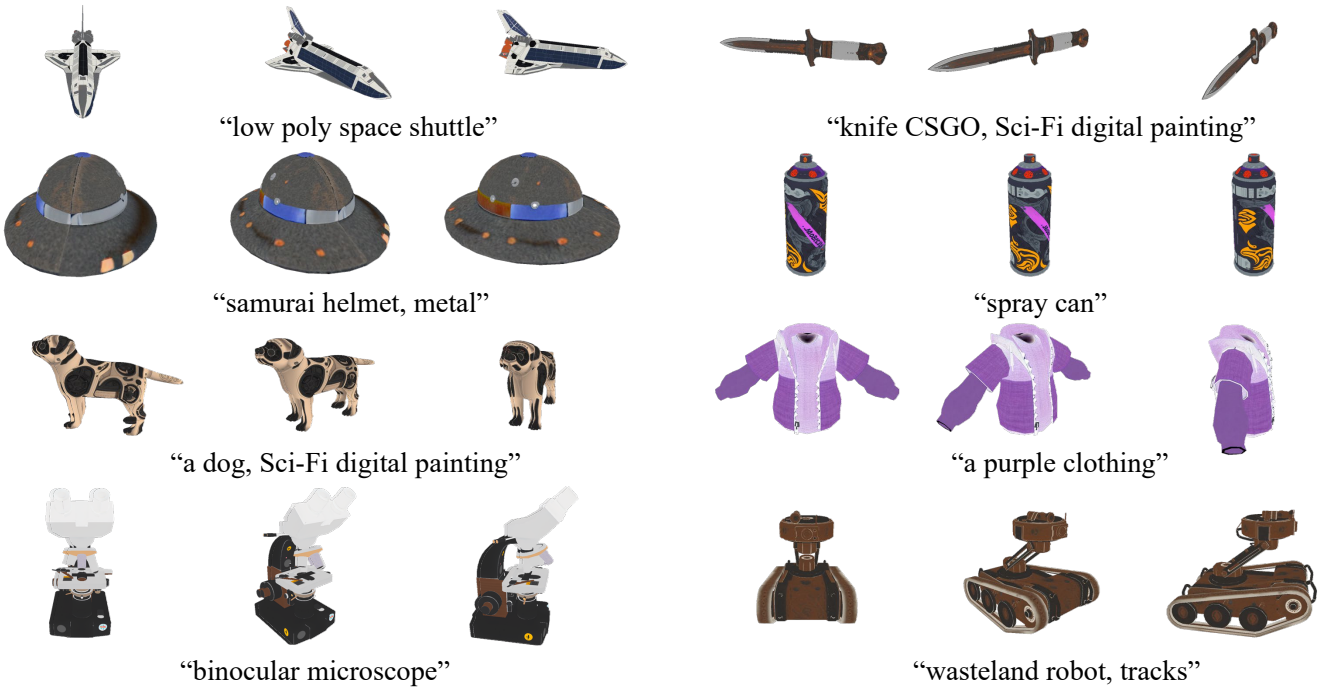



Figure 12. Additional texturing results generated by Paint3D on text-to-texture task. Each textured mesh is shown from three viewpoints.

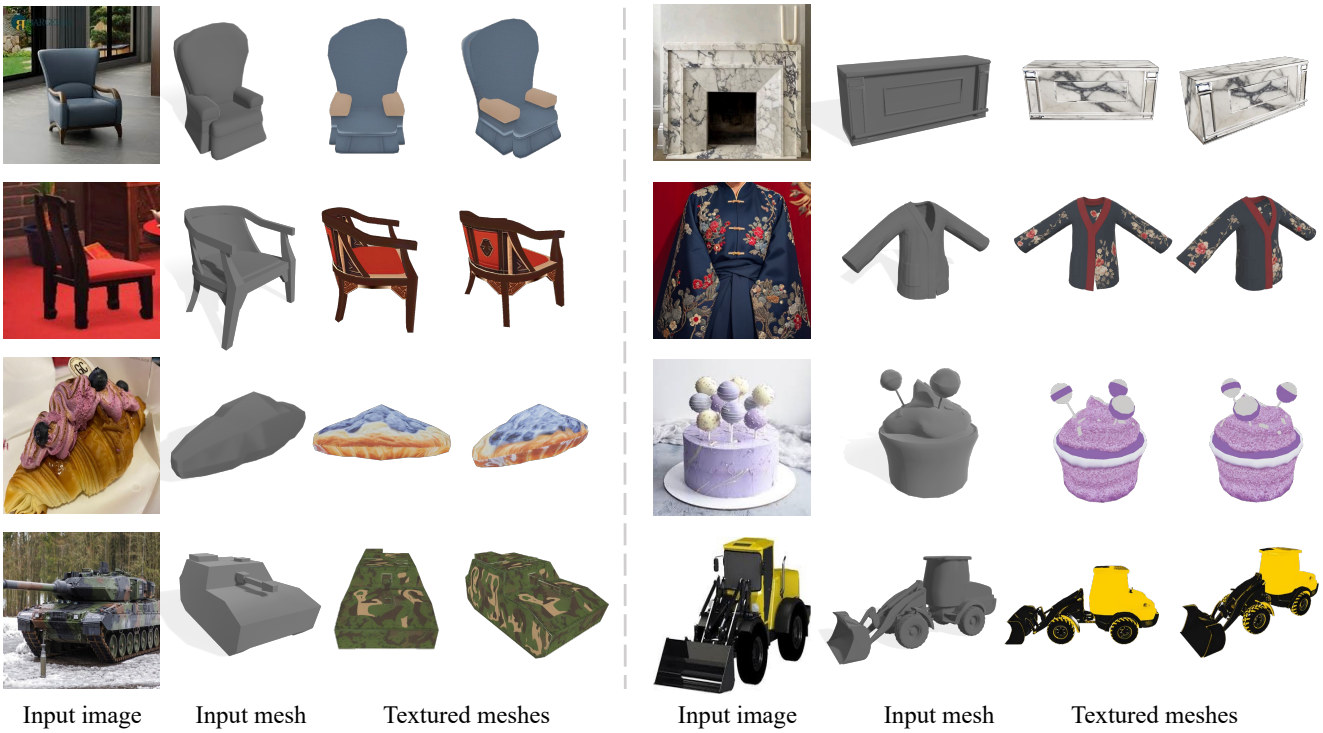



Figure 13. Additional samples from Paint3D for image-to-texture generation and each textured mesh is shown from two viewpoints. The input image conditions are collected in the wild.

## B. Additional Experiments

We first study the effectiveness of the position map in the UV inpainting and UVHD modules. Then, we provide more comparisons with category-specific texture generation approaches [69].

### B.1. Evaluation of Position Map

To demonstrate the effectiveness of position map in two texture refinement modules, UV inpainting and UVHD, we further conduct experiments on two baselines "UV inpainting w/o position map" and "UVHD w/o position map". The "UV inpainting w/o position map" configuration refers to inpainting the uncolored area without the guidance of the position map The "UVHD w/o position map" configuration represents the result of enhancing the texture map in UV space, without the position map. As indicated in Tab. 5, the performance shows a significant decrease when the position map is not utilized in UV inpainting or UVHD, indicating its irreplaceable function during texture refinement processing. We visualize the results of two baselines in Fig. 14 and Fig. 15. In both scenarios, the model produces inferior results compared to our full model.

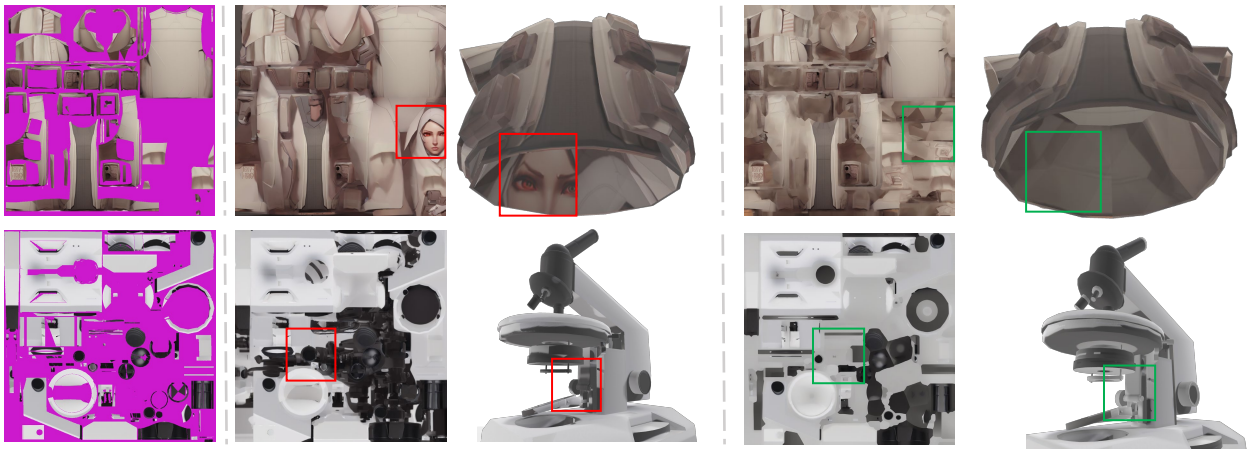



Figure 14. Visualization of the effect of the position map in the UV inpainting module. Without the position map, the inpainted texture is semantically confused. The purple area indicates the uncolored area.

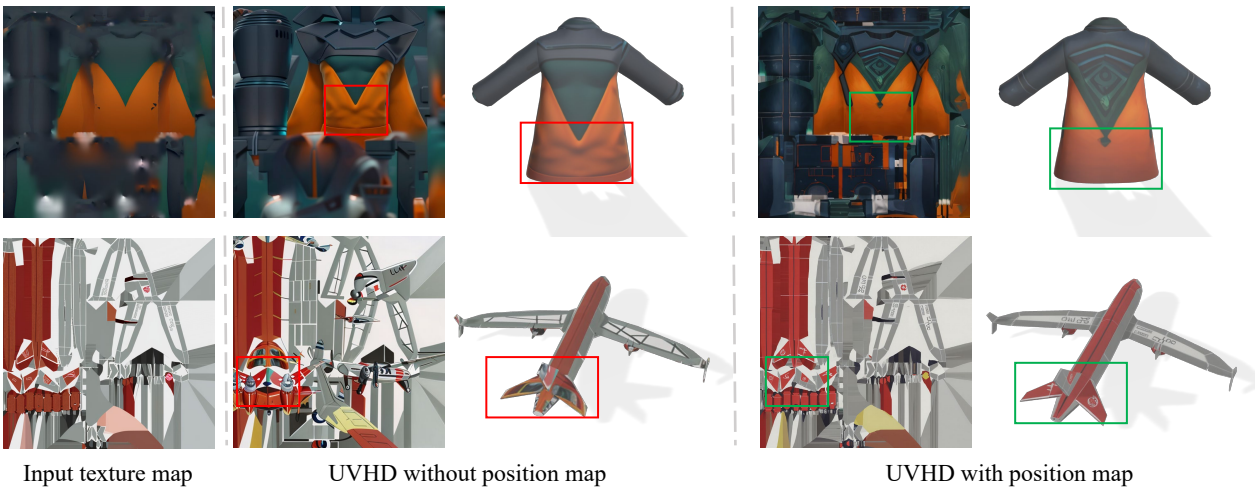



Figure 15. Visualization of the effect of the position map in the UVHD module. In the absence of the position map, the enhanced texture appears distorted (top) or lacks semantic coherence (bottom).

| Method | FID↓ | KID ↓ |
|---|---|---|
| UV inpainting w/o position map | 39.29 | 8.36 |
| UVHD w/o position map | 37.62 | 7.96 |
| Full model | **27.28** | **4.81** |

Table 5. Evaluation of the effectiveness of the position map in the UV inpainting and UVHD modules. This demonstrates the crucial role of the position map during the diffusion process in UV space.

### B.2. Comparisons with Category-Specific Model

In addition, we conduct comparison experiments with a category-specific approach on the chair and table categories of ShapeNet [4]. We choose Point-UV [69] as the baseline because 1) it represents the current state-of-the-art for category-specific texture generation, and 2) it has the conditional texture generation capability under both text and image conditions. For the input conditions, we utilize text and images as provided in [69]. As shown in Fig. 16, Paint3D achieves comparable results with Point-UV under both text and image conditions.

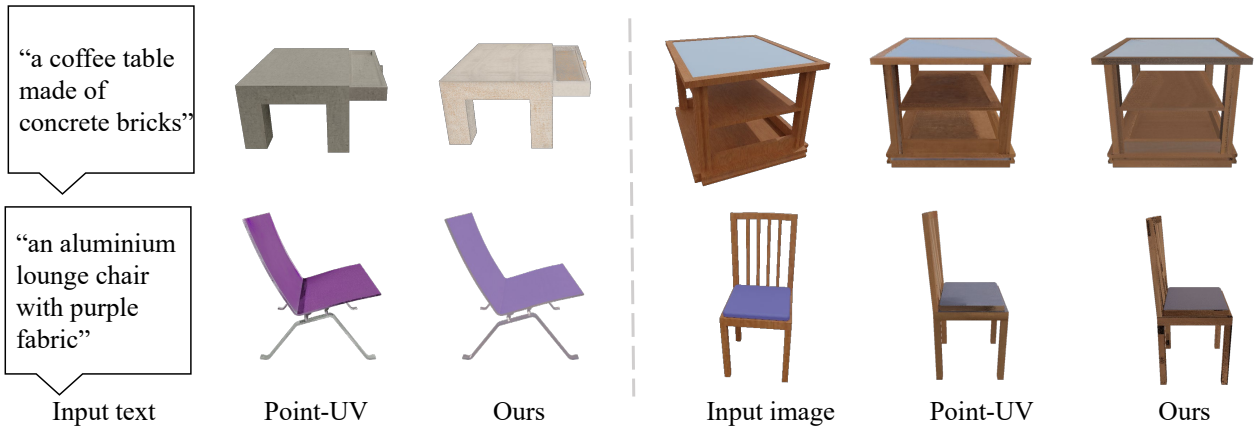



Figure 16. Qualitative comparisons on texture generation conditioned under text prompt (left) and image condition (right) on ShapeNet dataset [4]. We compare our textured mesh against those generated by the state-of-the-art category-specific approach, Point-UV [69]. In the categories of table and chair, Paint3D achieves comparable results with Point-UV under both text and image conditions.

## C. Discussion on failure case

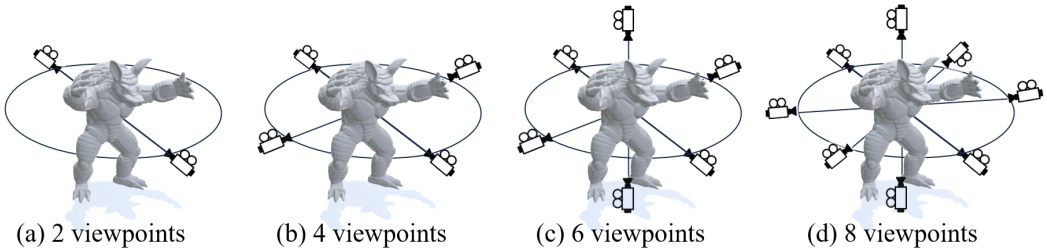
Our approach still suffers from the multi-faces problem in the coarse stage which will result in a failure case. This issue primarily arises from the inconsistency of multi-view texture images sampled by the pre-trained 2D diffusion model, as it is not explicitly trained on multi-view datasets. We believe that fine-tuning or retraining 2D diffusion models on large-scale multi-view datasets will improve the multi-view consistency of textures.


Figure 17. Visualization of our failure cases. Paint3D still suffers from the multi-faces problem in the coarse stage which will result in a failure case. Here, Paint3D generates duplicate mouse or lion faces in both the front and back views

# D. Implementation Details

## D.1. Multi-view Texture Sampling

We extend the texture sampling process (Eq. (1) and Eq. (2)) to the multi-view scene. Specifically, in the initial texture sampling, we utilize a pair of cameras to capture two depth maps $\{d_1, d_2\}$ from symmetric viewpoints. We then concatenate those two depth maps horizontally (in width) and compose a depth grid with a size of $1 \times 2$, denoted as $\mathbf{d_1}$. To perform multi-view depth-aware texture sampling, we replace the single depth image $d_1$ with the depth grid $\mathbf{d_1}$ in Eq. (1). Similarly, in the non-initial texturing, we horizontally concatenate renders, composing depth grid $\mathbf{d_k}$, RGB image grid $\hat{\mathbf{I}}_\mathbf{k}$, and mask grid $\mathbf{m_k}$. To perform multi-view depth-aware texture inpainting, we replace the inputs in Eq. (2) with those grids. As evaluated in Sec. 4.4, we also explore the effectiveness of the number of viewpoints. As shown in Fig. 18, we follow the viewpoint selection strategy of TEXTure. We select a sparse viewpoint set in our coarse stage since we design a followed texture refinement stage to inpaint texture holes.



(a) 2 viewpoints (b) 4 viewpoints (c) 6 viewpoints (d) 8 viewpoints

Figure 18. Visualization of the viewpoint sets in our coarse stage.

## D.2. Network Architecture

As shown in Fig. 19, we provide a detailed network architecture of UV inpainting and UVHD. In the inference, we simultaneously use the position encoder $\tau_p$ and inpainting encoder $\tau_i$ with a text2image model to perform texture inpainting in UV space, defined as UV inpainting module. Similarly, the combination of position encoder $\tau_p$ and image enhance encoder $\tau_t$ can achieve UVHD.

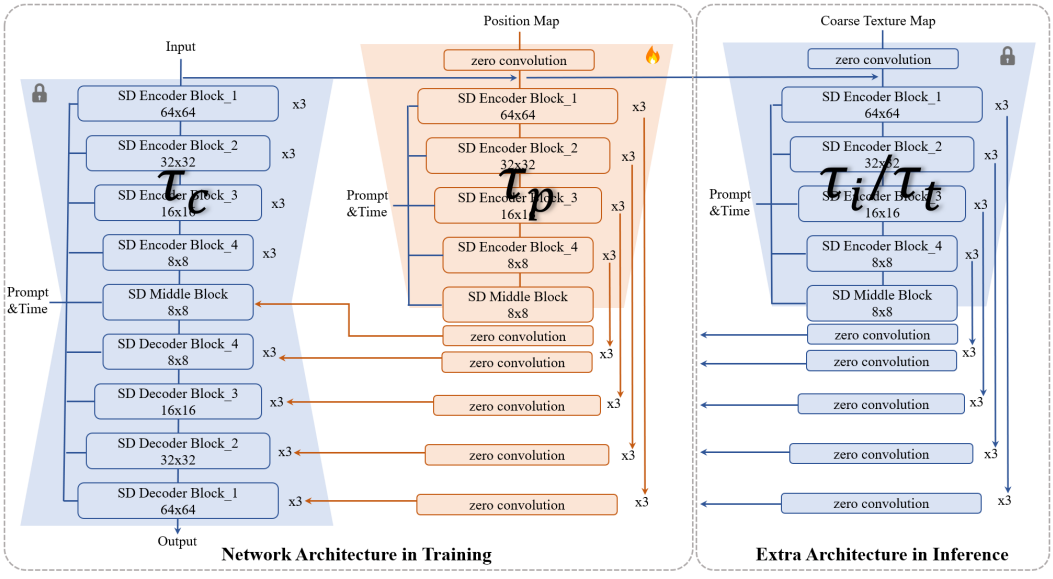



Figure 19. Network architecture of UV inpainting and UVHD. The module is UV inpainting when the extra encoder is set to $\tau_i$ and UVHD when the extra encoder is set to $\tau_t$.