

# Consistency<sup>2</sup>: Consistent and Fast 3D Painting with Latent Consistency Models

Tianfu Wang Anton Obukhov Konrad Schindler

Photogrammetry and Remote Sensing, ETH Zurich

## Abstract

*Generative 3D Painting is among the top productivity boosters in high-resolution 3D asset management and recycling. Ever since text-to-image models became accessible for inference on consumer hardware, the performance of 3D Painting methods has consistently improved and is currently close to plateauing. At the core of most such models lies denoising diffusion in the latent space, an inherently time-consuming iterative process. Multiple techniques have been developed recently to accelerate generation and reduce sampling iterations by orders of magnitude. Designed for 2D generative imaging, these techniques do not come with recipes for lifting them into 3D. In this paper, we address this shortcoming by proposing a Latent Consistency Model (LCM) adaptation for the task at hand. We analyze the strengths and weaknesses of the proposed model and evaluate it quantitatively and qualitatively. Based on the Objaverse dataset samples study, our 3D painting method attains strong preference in all evaluations. Source code is available at <https://github.com/kongdai123/consistency2>.*

## 1. Introduction

Recently, we witnessed a breakthrough in generative 3D content creation and painting [6, 21, 24], empowering content creators and 3D artists to recycle old 3D assets or prototype texture design ideas using simple text prompts. These methods either involve end-to-end optimization with the diffusion model as guidance through score distillation sampling (SDS) [21], which takes thousands of iterations to converge, or they iteratively denoise multi-view images in latent space, which involves tens to hundreds of denoising iterations for standard Latent Diffusion Models (LDMs) [25]. On top of that, new techniques in diffusion sampling have been developed to speed up the generation process. In particular, Latent Consistency Models (LCMs) [16, 28] enable one-step generation by learning to directly map noise to data while retaining the option of multi-step sampling that trades runtime for better generation quality. Despite the significant speed-up that LCMs bring to image generation, exploration of their efficient operation in 3D has barely started.

We present *Consistency<sup>2</sup>*, a fast way to generate multi-view *consistent* surface textures of a mesh using latent *consistency* models. Our work takes accelerated text-to-image diffusion models, particularly LCMs, and lifts their generative power to 3D in the task of generative mesh painting. The key to successful lifting lies in achieving view-consistent painting while not compromising speed and generation quality. We tackle this problem from the perspective of *multi-view denoising*, where we tailor our painting procedure to the few-step sampling process of LCMs. Our method can generate high-quality mesh paintings in 4 timesteps per view, taking less than two minutes per mesh on a single consumer GPU. *Consistency<sup>2</sup>* provides a quick way for 3D content creators to prototype painting designs from text descriptions. Overall, it is much faster than most previous mesh painting methods.

Our contributions can be summarised in the following:

- We introduce the first recipe for mesh painting by multi-view denoising LCMs, benefiting from their few-step sampling process.
- We design a novel painting representation with separate noise and color textures. By applying appropriate interpolation techniques to each, we gain greater flexibility in texture resolution choice and camera view sampling.
- We apply our method to a diverse range of 3D meshes (Fig. 1), demonstrating competitive quality and runtime.

## 2. Related Work

**Diffusion and Consistency Models** Denoising diffusion probabilistic models (DDPMs) [11] have recently achieved state-of-the-art results in high-quality and photorealistic image generation. LDMs [20, 25] shift the denoising process to low dimensional latent space, reducing computational requirements while achieving competitive quality. Through frameworks such as ControlNet [34], these models have been further conditioned on various modalities such as depth maps, images, and segmentation masks.

Numerous improvements have been proposed to accelerate image generation [15, 26, 28]. Consistency models, proposed by Song *et al.* [28], emerge as a new category of diffusion models that enable one- or few-step generation from noise to data. Consistency models are formulated with



Figure 1. Selected painting results of Objaverse [8] meshes using Consistency<sup>2</sup>. Our method paints detailed high-resolution textures with very few denoising diffusion steps and allows for free camera pose selection for view painting.

the property of *self-consistency*, where points on the same Probability Flow ODE trajectory map to the same data sample [27]. LCMs [16] extend consistency distillation to the latent space, allowing distillation from LDMs.

### Generating 3D with Diffusion Guided Optimization

Alongside advancements in 2D image diffusion models, there has been an increasing interest in leveraging these models to generate 3D objects. One line of work is generation through end-to-end optimization of 3D scene parameters using a differentiable renderer, guided by the supervision of a 2D image diffusion model through Score Distillation Sampling (SDS) [6, 7, 17, 21, 22, 29, 30, 33]. This technique is first introduced in DreamFusion [21] with Neural Radiance Fields (NeRF) [18] as the 3D representation. Since then, this recipe has been applied to alternative 3D representations such as textured meshes [7, 17, 30], implicit surface representations [9, 22], BRDF materials [6], and 3D Gaussians [12, 29]. Despite their flexibility, SDS methods converge slowly and produce oversaturated results [21]. Moreover, they do not benefit from LCM speedups as LCM is tailored to generation through *denoising*.

### Multi-view and Multi-patch Diffusion Generation

Since diffusion models are often trained on images with a fixed resolution, there has been a lot of interest in using these models to generate surfaces of arbitrary size, such as panoramas [2, 14, 32, 35] and 3D in the context of mesh painting [4, 6, 24, 31]. For panorama generation, MultiDiffusion [2] proposes breaking down a large canvas into over-

lapping patches and fusing intermediate denoising results using a weighted average. Generative Powers-of-10 [32] generates view-consistent image zoom-ins by conducting multi-scale sampling across zoom levels in both color and noise spaces. On the 3D side, a line of work exists on generating textures for a given 3D geometry using *multi-view denoising* with the same image diffusion model [4, 5, 24, 31]. TEXTure [24] pioneered this approach by denoising each camera view one after another and back-projecting each iteration to a texture map using rasterizer-based differentiable rendering. One work that shares the idea of combining the intermediate multi-view denoising results with our approach is TexFusion [4]. However, that method uses a single latent texture map, which limits its flexibility in terms of texture resolution and camera pose selection.

## 3. Methods

**Prerequisites** Given a 3D mesh and a text description, we aim to achieve fast, view-consistent mesh painting. We consider only pre-trained depth- and text-conditioned LCMs to ensure fast operation. Inspired by works such as TexFusion [4] and TEXTure [24], we parameterize the surface with 2D texture maps. Compared to neural radiance or color fields exhibiting greater flexibility [31], mesh textures are a desirable representation for painting since they are inherently view-consistent (when mipmapped, see below) and come attached to the geometry. Unlike TexFusion [4] and TEXTure [24], we employ two different texture map layers to render the noise state and the predicted content separately. This design choice is inspired by Generative Powers-



Figure 2. We project a mesh textured with latent noise using various texture interpolation methods. The naive bilinear interpolation (**left**) disturbs the latent probability distribution and gives poor results. Our variance-preserving noise rendering (**middle**) results in a similar denoised quality of the mesh compared to just using a noise latent without any projection as a reference (**right**).  $\text{VarInit}$  denotes the variance of the initial projected noise.

of-10 [32], where it is demonstrated that such a separation helps fuse a stack of 2D images at different zoom levels. We extrapolate this idea to meshes in 3D and consider the process of rendering a mesh with multiple virtual cameras similar to sampling overlapping patches on the texture map with different degrees of magnification, which depend on the local mesh curvature and distance to the camera. However, Generative Powers-of-10 [32] uses a proprietary diffusion model [22] operating directly in the RGB pixel space. Contrary to that, modern high-quality open-source LDMs [25] operate in the latent space to save computational resources.

We empirically found manipulating multi-view latents challenging and resulting in poor novel view synthesis. On the other hand, the RGB pixel space has well-established and understood strategies for interpolation, such as mipmapping [1, 13]. Mipmapping involves an intermediate stack of downsampled textures. It automatically calculates the appropriate sampling level and the required texture pixels (texels) for interpolating each rendered pixel. With these observations in sight, the rationale for using an LCM as the generative model is twofold: (1) it enables fast and few-step generation (within ten timesteps), and (2) at each step, it can directly output a clean denoised latent sample, which can be immediately decoded into pixel space, and allows us to perform multi-view content fusion directly in the pixel space, taking advantage of mipmapping.

**Variance Preserving Noise Rendering** Generative Powers-of-10 [32] has shown that the image quality and consistency improve when noise is shared between patches. In the 3D painting case, we also want to render noise from multiple views to maximize sharing noise texels between views. Compared to nearest neighbor interpolation, the bilinear interpolation of the textured surface facilitates better texel sharing across views. However, the denoising diffusion

process requires the projected noise component of the data to follow the normal distribution strictly. As shown in Fig. 2, bilinear interpolation leads to subpar results due to violating distribution properties.

We will now consider sampling from a random normal noise texture using bilinear interpolation. Specifically, the standard convex combination  $Z' = rX + (1 - r)Y$ , of two i.i.d. random variables  $X, Y \sim \mathcal{N}(0, 1)$ , where  $r \in [0, 1]$ , is a random variable with non-unit variance:  $Z' \sim \mathcal{N}(0, r^2 + (1 - r)^2)$ . The 2D case of bilinear interpolation involves 4 variables and two coefficients  $r_u, r_v$ , but inherently suffers from the same issue. We designed a custom variance-preserving interpolation to alleviate this issue and ensure unit variance of the interpolated noise variable. In the 1D case, it takes the following form:

$$Z = \sqrt{r}X + \sqrt{1 - r}Y. \quad (1)$$

Here  $Z \sim \mathcal{N}(0, 1)$  for all interpolation coefficients  $r \in [0, 1]$ . For 2D variance-preserving interpolation, we repeatedly apply the 1D case in Eq. (1) in  $u$  and  $v$  dimensions.

**Multi-view Fusion Consistency Sampler** We adapt the multi-view sampling and painting procedure to the sampling mechanism of LCMs [16, 28]. First, the color texture is initialized to black, and the noise texture is initialized to random normal noise. The object is rendered independently textured with color and noise for each camera view in each iteration. Additionally, we store depth maps from the Z-buffer. Color textures undergo mipmapping for better quality and



Figure 3. We show a comparison between Texfusion’s Sequential Interlaced Multi-view Sampler (SIMS, **left**) [4] and our method (**right**) in accommodating high-resolution textures. Texfusion’s SIMS has rigid requirements on texture resolution and completely loses view consistency with a high-resolution texture. Our formulation of separating noise and color textures enables high texture resolutions without sacrificing view consistency.

Method	FID [10] ↓	KID ( $\times 10^{-3}$ ) [3] ↓
Text2Tex [5]	28.93	6.88 $\pm 0.05$
Consistency <sup>2</sup> (Ours)	<b>22.74</b>	<b>4.02</b> $\pm 0.03$

Table 1. **Quantitative comparison of mesh painting methods.** Generative metrics for Text2Tex [5] and our method demonstrate superior painting of Objaverse [8] meshes.

artifact-free view consistency, and the background is filled from an environment cubemap. Noise texture rendering is performed with our custom variance-preserving interpolation, and the background is filled with random normal noise. For each view  $i$ , we obtain the color latents  $\mathbf{x}_i$  by converting the color-textured projections into the latent space with the LCM encoder. We then compute a weighted average of the color ( $\mathbf{x}_i$ ) and noise ( $\mathbf{z}_i$ ) latents based on the LCM noise scheduling equation at timestep  $t$ :  $\hat{\mathbf{x}}_{i,t} = \alpha(t)\mathbf{x}_i + \sigma(t)\mathbf{z}_i$ . Refer to [16] for the notation on  $\alpha$  and  $\sigma$ . The combined latent is then fed into the LCM, with the text prompts and depth map for conditioning. The resulting clean latent is produced in a single step and decoded back into RGB pixel space for multi-view fusion. We perform the inverse rendering with mipmaps to update the color texture map. Specifically, we optimize it to fit the denoised RGB images, minimizing the L2 photometric loss. Due to the separation of noise and color textures and the usage of appropriate interpolation methods, we achieve flexibility in texture resolution, as shown in Fig. 3.

## 4. Experiments

**Implementation and Setup** We use the nvdiffrast [13] renderer to implement our variance-preserving interpolation. Additionally, PyTorch3D [23] framework is used for mesh and view processing. For the image generation model, we use SDXL [20] model distilled to an LCM, with additional depth conditioning from ControlNet [34]. A dome of 15 cameras covers the whole mesh located in the origin. For each iteration, we adjust the azimuth of each camera pose by  $10^\circ$  to promote coverage of the object from various view angles. Additionally, we add relative pose descriptions in the prompt for each view to facilitate view consistency. In our Multi-view Fusion Consistency Sampler, we use 4 denoising iterations, with text guidance scale 7.5 and depth guidance scale 0.4 on all iterations. The color texture resolution is 4K, and the noise texture resolution is  $768 \times 768$ .

**Comparison to State-of-the-Art** We compare our method with Text2Tex [5], one of the best open-source state-of-the-art mesh painting methods. Each method is evaluated using a subset of the Objaverse evaluation meshes [8] from Text2Tex, totaling 343 meshes in 203 categories. We include text descriptions of meshes provided by Objaverse in the prompt.

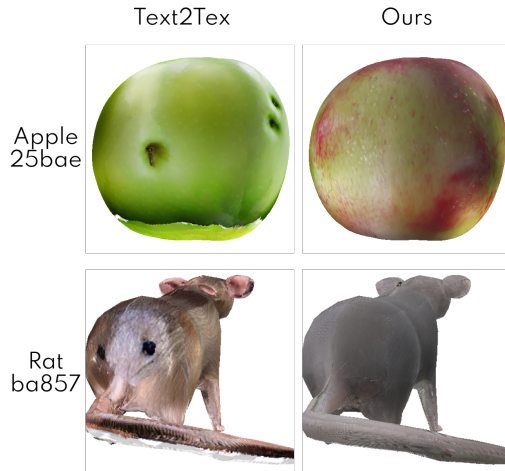


Figure 4. **Qualitative comparison of Objaverse [8] mesh painting.** Text2Tex’s [5] sequential nature makes it susceptible to irrecoverable artifacts such as seams and coherence issues. Our method is free of these limitations.

We sample views of the painted meshes from a set of object-centric views with various elevation angles. We also generate reference images using SDXL [20] pipeline with 50 denoising iterations guided by captions and depth maps obtained from the same views. This set of view-inconsistent images serves as a reference for the best generation quality attainable with a pure diffusion model.

We calculated [19] Frechet Inception Distance (FID) [10] and Kernel Inception Distance (KID) [3] between sets of painted and reference images. Our method outperforms Text2Tex in both metrics (Tab. 1). We also recorded  $\sim 7.5 \times$  runtime speed-up of our method (under 2 minutes per mesh) compared to Text2Tex (around 15 minutes per mesh) [5].

In a qualitative comparison, our approach of denoising views simultaneously with an LCM demonstrates better visual quality than sequential denoising methods such as Text2Tex. Specifically, our method is free of seams and the Janus (multi-head) problem (Fig. 4).

## 5. Conclusion

We presented a novel recipe for lifting LCM into 3D for fast generative painting. We introduced a creative mesh texture formulation, which separates noise and color and applies correct interpolation to both modalities, achieving unconstrained camera view selection and texture resolution. We conducted a study comparing our method with a state-of-the-art mesh painting method, Text2Tex, and showed advantages in generation quality and runtime. We believe our work is a step towards interactive and fast 3D painting that can benefit designers and content creators.

## References

- [1] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. Real-time rendering, 2019. [3](#)
- [2] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023. [2](#)
- [3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018. [4](#)
- [4] Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. Textfusion: Synthesizing 3d textures with text-guided image diffusion models. In *ICCV*, 2023. [2, 3](#)
- [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. [2, 4](#)
- [6] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. [1, 2](#)
- [7] Dale Decatur, Itai Lang, Kfir Aberman, and Rana Hanocka. 3d paintbrush: Local stylization of 3d shapes with cascaded score distillation. *arXiv preprint arXiv:2311.09571*, 2023. [2](#)
- [8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. [2, 4](#)
- [9] 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 *NeurIPS*, 2022. [2](#)
- [10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. [4](#)
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. [1](#)
- [12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. [2](#)
- [13] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. [3, 4](#)
- [14] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. In *NeurIPS*, 2023. [2](#)
- [15] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022. [1](#)
- [16] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. [1, 2, 3, 4](#)
- [17] 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. [2](#)
- [18] 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. *Communications of the ACM*, 65(1), 2021. [2](#)
- [19] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738. [4](#)
- [20] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. [1, 4](#)
- [21] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. [1, 2](#)
- [22] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. [2, 3](#)
- [23] 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. [4](#)
- [24] 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](#)
- [25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. [1, 3](#)
- [26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. [1](#)
- [27] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. [2](#)
- [28] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. [1, 3](#)
- [29] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. [2](#)
- [30] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. *arXiv:2304.12439*, 2023. [2](#)
- [31] Tianfu Wang, Menelaos Kanakis, Konrad Schindler, Luc Van Gool, and Anton Obukhov. Breathing new life into 3d assets with generative repainting. In *BMVC*, 2023. [2](#)
- [32] Xiaojuan Wang, Janne Kontkanen, Brian Curless, Steve Seitz, Ira Kemelmacher, Ben Mildenhall, Pratul Srinivasan, Dor Verbin, and Aleksander Holynski. Generative powers of ten. *arXiv preprint arXiv:2312.02149*, 2023. [2, 3](#)

- [33] 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. In *NeurIPS*, 2024. [2](#)
- [34] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. [1](#), [4](#)
- [35] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *CVPR*, 2023. [2](#)