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

Lafite: A Generative Latent Field for 3D Native Texturing

Chia-Hao Chen[1] Yuan-Chen Guo[2] Zi-Xin Zou[2] Ze Yuan[3]
Guan Luo[1] Xiaojuan Qi[3] Ding Liang[2] Yan-Pei Cao [2†] Song-Hai Zhang [1†]

[1]Tsinghua University [2]VAST [3]The University of Hong Kong

Figure 1. A gallery of diverse, high-fidelity, and seamless textures generated by our 3D-native framework, Lafite.

Abstract

Generating high-fidelity, seamless textures directly on 3D surfaces, a process we term 3D-native texturing, is a fundamental open challenge, promising to overcome the limitations of traditional UV-based and multi-view projection methods. While promising, existing native approaches are bottlenecked by the lack of a powerful and versatile latent representation, severely limiting the fidelity of generated results. In this work, we identify this representation gap as the central roadblock to progress. We introduce ***Lafite****, a framework that resolves this challenge by learning to generate textures as a 3D generative sparse latent color field. At its core, Lafite leverages a variational autoencoder (VAE) to encode complex surface appearance into a sparse, structured latent space and then decoded into a continuous color field. This novel representation achieves unprecedented fidelity, i.e., yielding a* ***>10 dB PSNR*** *improvement in reconstruction over state-of-the-art methods, by effectively disentangling texture from mesh topology and UV parameterizations. Building on this superior representation, a conditional rectified flow model synthesizes textures with state-of-the-art quality and coherence. Extensive experiments show that Lafite not only sets a new standard for native texturing but also enables flexible downstream applications like editing and material synthesis, paving the way for the next generation of 3D content creation workflows.*

1. Introduction

The automated generation of high-quality 3D content has become a cornerstone of modern computer graphics and vision. While significant strides have been made in generating complex 3D geometry [73, 86], the creation of detailed, seamless, and coherent textures remains a major bottleneck. Textures often contribute more to the visual richness and realism of an asset than the geometry itself. As such, developing powerful generative models for texturing is a critical and defining challenge for the next generation of AI-driven content creation.

Current approaches for texture generation predominantly fall into two categories, both of which are inherited from 2D-centric paradigms and suffer from fundamental limitations. The most common approach involves generating multiple 2D images of an object from different viewpoints and then projecting them back onto the mesh surface [10, 26, 41, 44, 53]. This "multi-view projection" strategy is inherently flawed; it struggles to reconcile inconsistent predictions caused by occlusions and view-dependent lighting, inevitably leading to conspicuous seams and artifacts that require complex post-processing to fix. An alternative approach is to generate textures directly in a 2D UV-unwrapped space [82]. However, this method trades one set of problems for another. It becomes critically dependent on a mesh's UV parameterization, which is non-unique and often introduces severe distortions, leading to textures that are warped or contain seams along UV island boundaries. These approaches are symptomatic treatments, not a cure for the underlying challenge of creating truly 3D-native appearance.

We argue that the only way to guarantee spatial coherence and seamlessness is to generate textures *directly in 3D space*. This native approach sidesteps the inherent contradictions of 2D-based methods. If this path is so clearly superior, why has it not yet dominated? The central roadblock, and the primary focus of this work, has been the lack of a suitable **3D texture representation**. An ideal representation for generative modeling must satisfy three criteria: it must be **expressive** enough to capture high-frequency details; it must be **decoupled** from the mesh's specific topology and UV coordinates; and it must be **compact and structured** enough to be learned effectively by a generative model.

To address this critical representation gap, we introduce **Lafite**, a framework built on the central insight that a powerful generative model of texture must first learn a powerful representation of it. We propose to model texture as a 3D generative **sparse latent color field**, and design a novel variational autoencoder (VAE) to learn its structure from data. Our VAE encodes the appearance of a 3D asset, represented as a dense colored point cloud, into a sparse, structured latent feature. This representation is powerful because it is **local and continuous**; it concentrates modeling capacity near the object's surface and can be queried at any point in space to reconstruct fine-grained details. This fundamentally disentangles texture from the discrete, and often problematic, mesh connectivity and UV coordinates.

Furthermore, Lafite introduces a second key insight in how it conditions the generative process. We found that the geometry of the mesh is implicitly but powerfully captured by our VAE's encoder. By encoding an uncolored ("white") version of the mesh, we can extract a pure geometry latent that provides a complete, occlusion-free 3D shape prior. This elegant maneuver allows us to condition our generative model on a perfect 3D geometric signal that is inherently aligned with the texture representation, all without the need for a separate, complex geometry encoder. This synergistic design of representation and conditioning is the key to Lafite's ability to synthesize textures with state-of-the-art quality and coherence.

Our main contributions are summarized as follows:

- **A High-Fidelity Latent Field Representation:** We are the first to propose and learn a generative latent field for native texturing via a VAE, achieving state-of-the-art reconstruction fidelity (**>10 dB PSNR** improvement over prior work) that is fully decoupled from mesh topology.
- **A Unified Texture Generation Framework:** We introduce a highly effective method for geometry conditioning by reusing our texture encoder to create a pure geometry latent, ensuring perfect alignment and eliminating the need for auxiliary networks.
- **State-of-the-Art Native Texturing:** Lafite significantly outperforms all existing native and projection-based methods, delivering seamless, detailed, and coherent textures across a diverse range of complex 3D assets.

2. Related Work

3D Representations and Generative Models. The field of 3D content generation has rapidly evolved, driven by advances in both underlying 3D representations and the neural models that create them. Research has explored a wide range of representations, from implicit fields like SDFs [13, 55], UDFs [22, 55], or hybrid ones [69, 76] to explicit structures like meshes [12, 51, 64], point clouds, and 3D Gaussian Splatting [33, 72]. The generative models and methods themselves have progressed from score-based distillation [24, 40, 57, 68, 70, 81] on differentiable representations to powerful 3D-native models. A key component in many modern systems is the intermediate neural representation, such as triplanes [9, 29, 37, 87, 92], sparse latents [25, 71, 73], or VecSets [38, 67, 86, 88], which encode 3D information for the generative model. While this body of work provides the foundation, it has largely focused on geometry; our work tackles the distinct challenge of a generative representation for texture.

Multi-view Texture Synthesis. A recent common paradigm for texturing is to synthesize multiple 2D views and project them onto a mesh. Initial works, enabled by powerful 2D diffusion models, faced a trade-off between slow, per-object optimization [23] and progressive generation [8, 10, 61] that risked view misalignment. Subsequent research has therefore focused on improving multi-view consistency [14, 20, 44, 46], either by synchronizing the denoising process across views or by jointly generating all views [6, 26, 41, 78, 83, 88]. Orthogonal to consistency, a major thrust has been generating physically-based rendering (PBR) materials [7, 11, 18, 26, 31, 50, 65, 79, 89, 91] to support realistic relighting [28, 35, 85]. The flexible view selection scheme [34] helps reduce occlusion, while making it even harder to maintain consistency due to the diversified geometry. While increasingly sophisticated, these methods all grapple with the fundamental ill-posedness of reconciling multiple 2D images into a single, coherent 3D surface, a problem our native approach sidesteps entirely.

Native Texture Generation. To circumvent the inherent issues of 2D projection, native approaches generate texture directly in the object's own space. Some methods operate in a 2D UV-unwrapped space [30, 82, 83], but this makes them dependent on a non-unique and often distorted parameterization. Others try to adopt a colored point cloud attached to the surface as texture [19], suffering from an excessive number of tokens for intricate details during the diffusion process and rendering issues. A more robust direction, and the one we pursue, is to represent textures natively in 3D. Prior works have explored this by learning on mesh surfaces [63], 3D texture fields [32, 41, 53, 73, 74], and generating colored point clouds or 3DGS [42, 75, 80] to define spatial texture. However, these pioneering native works have been limited by their chosen representations, which can struggle to capture high-frequency detail within a compact latent space suitable for high-fidelity generative modeling.

3. Method

We introduce Lafite, a generative framework for seamless, high-fidelity native texturing on arbitrary mesh topologies. At its core is a novel latent representation of 3D texture, which we detail along with its VAE architecture (Sec. 3.1). We then describe our conditional generation model (Sec. 3.2), various downstream applications (Sec. 3.3), and our data curation pipeline (Sec. 4).

3.1. The Lafite Representation: A VAE for Sparse Latent Color Field

Unlike dense representations such as triplanes [9, 62] or dense grids [49], we adopt a more efficient sparse structure that has been widely used in recent advanced models [25, 39, 73]. Specifically, we discretize 3D space into a voxel grid, storing information only in active voxels near the object's surface. Within each active voxel, a continuous implicit function defines the local color field. This hybrid approach efficiently concentrates modeling capacity and memory, handles arbitrary topology, and enables the querying of continuous, high-frequency details. To learn this representation, we design a variational autoencoder (VAE) that maps a densely sampled colored point cloud into a compact, sparse latent space, from which the continuous color field is subsequently decoded.

Texture Encoding. Recent 3D generative models [73] typically achieve this by projecting pre-trained 2D features (e.g., DINO) [54, 66] from multiple camera views onto the on-surface sparse voxels. While effective, this strategy exhibits three fundamental limitations: 1) The projection process is view-dependent, causing inconsistencies or ambiguity in occluded regions; 2) Pretrained 2D features have limited resolution, restricting high-frequency surface detail; 3) Features are tied to the 2D foundation knowledge, which may not optimally represent fine-grained 3D surface attributes such as PBR materials.

In contrast, we propose constructing a representation that directly and unbiasedly captures the high-fidelity texture of a 3D asset by using a colored point cloud $\mathcal{P} = \{x_i = (p_i, n_i, c_i)\}$ densely sampled from the textured mesh $\mathcal{M}$. Specifically, we uniformly sample points p_i together with their corresponding normals n_i and raw colors c_i, which serve as the input to the texture encoder. This provides the encoder with native, unoccluded, and view-independent surface information, minimizing information loss and forming a more accurate ground truth for representation learning.

Given this point cloud, we adopt an encoder architecture similar to SparseFlex [25], but replacing the PointNet [58, 59, 77] with a specialized point-voxel attention mechanism for robustly fusing point features within each voxel. This encoder compresses the detailed, high-frequency texture information into sparse, structured latent features $\mathcal{Z} = \{z_k\}$, where k indexes the set of L on-surface voxels. The point-voxel attention mechanism comprises an intra-voxel self-attention module and a point-voxel cross-attention module. The intra-voxel self-attention module processes all points within each voxel, allowing them to interact and aggregate local geometric and appearance features into a set of point-wise features $\tilde{x}$:

$$\tilde{x}_i = \sum_{j=1}^{n} \text{softmax}\left(\frac{Q_{x_i} K_{x_j}^T}{\sqrt{d}}\right) \cdot V_{x_j}, \tag{1}$$

where x_j denotes the initial point features within the same voxel as x_i.

The point-voxel cross-attention module then consolidates these detailed point-wise features into voxel-wise features $\tilde{v}_k$, representing the encoded texture information for each voxel. This is achieved by attending from a shared learnable voxel feature v_k to all point-wise features within

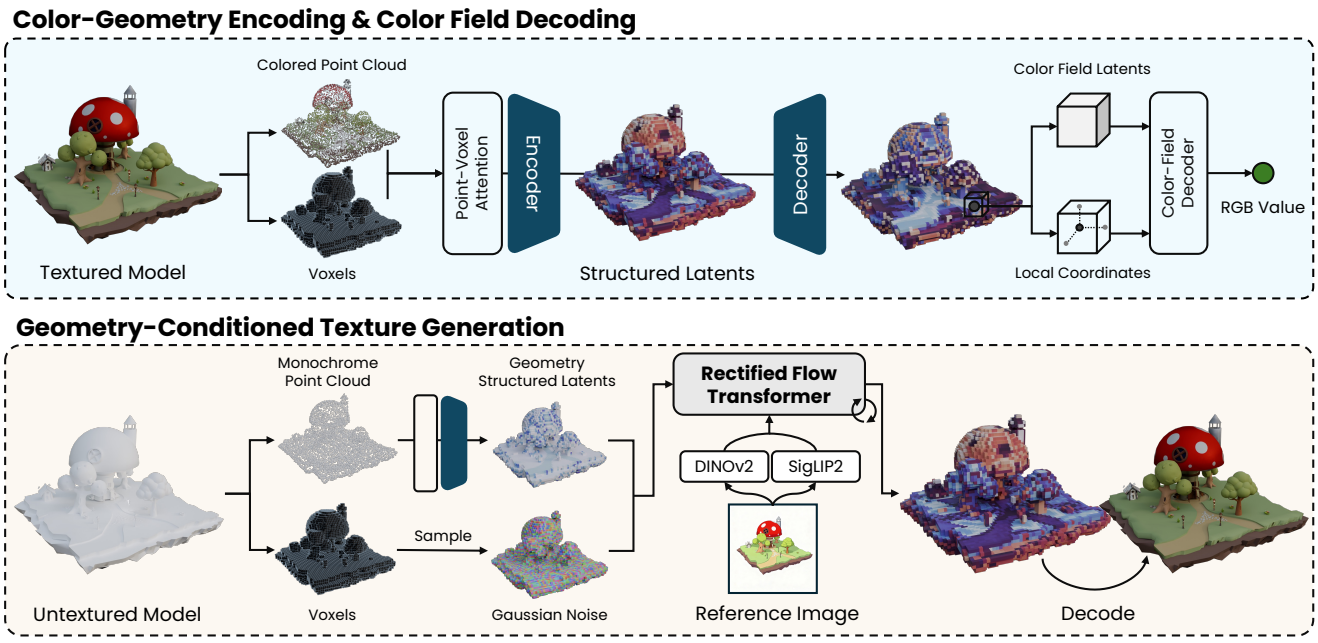

Figure 2. **The Lafite Pipeline. (Top)** A VAE autoencoder learns a sparse latent color field representation by encoding a colored point cloud from the mesh surface. **(Bottom)** For generation, the VAE encoder first extracts a pure geometry latent from an uncolored point cloud. This latent then conditions a Rectified Flow model to synthesize a new texture latent, which is decoded and baked into the final texture.

the voxel:

$$\tilde{v}_k = \sum_{i=1}^{n} \text{softmax}(\frac{Q_{v_k} K_{\tilde{x}_i}^T}{\sqrt{d}}) \cdot V_{\tilde{x}_i}. \tag{2}$$

This design robustly captures the local surface appearance rather than the average pooling in PointNet. The resulting set of voxel embeddings $\{\tilde{v}_k\}$ is then passed through a standard sparse-voxel-based VAE encoder [25, 73] with shifted window attention to produce the final latent codes $\{z_k\}$. The overall texture encoding process is as follows:

$$\mathcal{E} : \{\{x_j\}_{j=1}^{N_i}\}_{i=1}^{L} \to \{z_k\}_{k=1}^{L}. \tag{3}$$

where N_i is the number of points in the i-th voxel.

Implicit Geometry Encoding. A crucial property of our encoder is that, besides encoding texture, it naturally captures high-fidelity geometry from the point cloud positions at no extra cost. We exploit this by introducing a pure *geometry latent*: passing an uncolored point cloud – setting all input colors c_i to a canonical value like white $(1, 1, 1)$ – we effectively factor out texture information. The resulting geometry latent z_{geo} represents only the object's shape. This elegant approach provides an occlusion-free 3D geometric prior for our generative model without a separate geometry network, while ensuring perfect alignment between the shape and texture conditions during synthesis.

Texture Decoding and Supervision. The decoder's role is to reconstruct the continuous color field $\mathcal{C}(p)$ from the sparse, structured latent representation $\{z_k\}$. Given a 3D query point p located in voxel k, a latent decoder $\mathcal{D}_S$ first transforms the corresponding latent code z_k into a higher-resolution local feature grid f_k via convolutional layers. This grid represents a sparse local color field within the voxel. The color decoder $\mathcal{D}_F$ then samples the feature at location p from f_k and predicts the final RGB color c. For brevity, we denote the voxel containing p by k and directly index the latent code as z_k. The overall VAE decoder $\mathcal{D}$ is defined as:

$$\mathcal{D} : \{z_k\}_{k=1}^{L} \times p_j \to c_j. \tag{4}$$

We train the VAE end-to-end using a reconstruction loss combined with a KL divergence regularization term. The reconstruction loss is a simple L1 loss on the color values of sampled points, which we find effectively promotes sharp details. To encourage the model to learn a robust representation of high-frequency textures, we augment input colors with minor Gaussian noise during training. The overall VAE objective is given by:

$$\mathcal{L} = \mathbb{E}_{x_i \sim \mathcal{M}}[|\mathcal{D}(\mathcal{E}(\{\hat{x}_i\}), p_j) - \hat{c}_j|] + \mathcal{L}_{\text{KL}}. \tag{5}$$

where $\hat{x}_i = (p_i, n_i, \hat{c}_i)$, $\hat{c}_i = c_i + \epsilon$ is the augmented color. Importantly, we do not rely on rendering-based losses (like LPIPS or SSIM), which can introduce blurriness and bias. Instead, supervision is performed directly in the 3D volume, ensuring the learned representation faithfully models the underlying surface texture.

3.2. Texture Synthesis via Rectified Flow

Having established a powerful latent representation for 3D textures, we now detail the generative process for synthesizing novel textures within this space. We employ a condi-

tional Rectified Flow [36, 43] model, which learns to generate novel texture latents z conditioned on the geometry of the input mesh as well as on text or image prompts.

Geometry-Conditioned Albedo Generation. The cornerstone of our synthesis pipeline is the generation of a base color (albedo) texture. The quality of this generation depends critically on the fidelity of the geometric conditioning. Unlike methods that rely on 2D geometric projections (e.g., position or normal maps), which suffer from occlusions and cross-view inconsistencies, our approach conditions directly on the complete and unambiguous 3D shape.

Specifically, we use the pure geometry latent z_{geo}, derived as described in Sec. 3.1, as the primary condition. This provides the generative model with a detailed, occlusion-free 3D geometry prior that is perfectly aligned with the target latent space. The model is trained using the conditional flow matching objective:

$$\mathcal{L}_{\text{albedo}} = \mathbb{E}\|v(x_t; t, z_{\text{geo}}) - (\epsilon - x_0)\|, \tag{6}$$

where z_{geo} is the geometry condition represented by $\mathcal{E}(\{p_i, n_i, \mathbf{1}\})$, and x_t is the noisy latent in timestep t, $x_t = (1-t)x_0 + t\epsilon$. z_{geo} is concatenated with x_t as a denoising input for each step. We adopt a progressive training strategy. The model is first trained on latents at 64^3 resolution, and this pre-trained backbone is then fine-tuned on 128^3 resolution latents.

Disentangled PBR Material Generation. Our framework naturally extends to generating physically-based rendering (PBR) materials by learning to synthesize roughness and metallic (RM) maps. A key insight is that these material properties are often physically correlated with the underlying albedo; therefore, conditioning solely on geometry is suboptimal

We leverage this insight to create a disentangled, hierarchical generation process. Instead of conditioning on the pure geometry latent, we fine-tune our albedo generation model to synthesize roughness-metallic (RM) latents conditioned on the *albedo latent* z_{albedo} (either from the user input or a previously generated sample). This hierarchical approach – i.e., Geometry $\rightarrow$ Albedo, then Albedo $\rightarrow$ RM – more faithfully models the physical dependencies and leads to more plausible and detailed material generation.

For implementation, we reuse the pretrained color texture VAE by simply replacing its three color channels with roughness, metallic, and an additional zero-padding channel. We then train the model with a similar flow matching objective $\mathcal{L}_{\text{RM}}$, additionally conditioned on the albedo latent. This design enables fully disentangled control over the final material appearance.

3.3. Interfacing and Interaction with Lafite

A key advantage of our sparse latent color field is its versatility. Beyond synthesis, the learned representation serves as a powerful asset that seamlessly integrates into existing 3D workflows and manipulated for fine-grained artistic control.

Texture Baking for Standard Pipelines. To integrate with standard rendering engines, our continuous 3D color field $\mathcal{C}(p)$ is "baked" [21, 52] into a 2D UV texture map. Unlike multi-view projection methods [26, 44] that struggle to resolve inconsistent predictions, our native 3D representation makes this process straightforward and robust. Baking simply involves querying our learned function $\mathcal{C}(p)$ for each corresponding 3D point on the UV map. To produce anti-aliased textures, we supersample by averaging the colors of multiple random points within each pixel's footprint. Because our underlying field is inherently seamless, this process yields clean texture maps free of the seams and artifacts that plague projection-based methods.

Editing and Refinement. Our framework naturally supports editing tasks. Adapting techniques like RePaint [47], a user-defined target mesh region can be mapped to its corresponding latent voxels, which are then masked and selectively regenerated while keeping surrounding regions unchanged (Fig. 9). This 3D-native editing process is inherently more coherent than 2D UV inpainting, as it respects the underlying geometry to ensure seamless transitions at edit boundaries.

4. Principled Data Curation

Large-scale 3D datasets [15, 16] are notoriously uncurated and contain ambiguous signals. We therefore employ a principled data curation pipeline to normalize assets and ensure our model learns from clean surface information.

Resolving Material Ambiguity. Public datasets contain assets with non-standard materials, such as emissive channels, that create ambiguous training signals. Our pipeline canonicalizes these materials: 1) for fully emissive surfaces, we use the emission map as the base color, as environmental lighting is irrelevant; 2) for partially emissive ones, we composite the base and emission colors with a tone mapping operator to prevent value clipping and preserve the asset's visual intent. This ensures the final color fed to our VAE remains within a valid, consistent range, preserving the visual intent of the original asset.

Resolving Geometric Ambiguity. Real-world 3D assets also contain non-standard geometry, e.g., stylized "outline" shells, designed for specific rendering effects. These components can corrupt the surface sampling process, as points sampled on them may be incorrectly associated with voxels of the primary object, leading to a noisy latent representation. Our curation pipeline automatically identifies and discards such geometry using heuristics based on rendering properties (e.g., backface culling, inverted normals), ensuring the VAE learns only from the intended visible surface.

High-Fidelity Surface Sampling. The fidelity of our la-

Table 1. **Quantitative comparison for conditional texture generation.** We evaluate Lafite against leading text- and image-conditioned methods ('*' denotes methods without PBR generation, and rendered with default roughness/metallic of 1/0). Lafite achieves superior or competitive performance across nearly all metrics, demonstrating its state-of-the-art generation quality. *Lower is better for all metrics.*

Method	Condition	Unshaded					Shaded				
		FID↓	FD_{CLIP} ↓	FD_{DINO} ↓	KD_{CLIP} ↓	KD_{DINO} ↓	FID↓	FD_{CLIP} ↓	FD_{DINO} ↓	KD_{CLIP} ↓	KD_{DINO} ↓
MVPaint* [14]	Text	158.28	96.86	140.70	0.153	0.184	135.94	84.53	118.85	0.117	0.120
MaterialAnything [31]	Text	165.50	96.89	115.72	0.166	0.129	126.00	74.76	108.78	0.090	0.125
SyncMVD* [44]	Text	126.20	76.25	88.17	0.072	0.060	119.38	66.87	80.99	0.058	0.047
TEXGen* [82]	Text+Image	136.58	86.18	106.66	0.114	0.094	130.29	77.62	97.25	0.088	0.080
SeqTex* [83]	Text+Image	119.31	61.58	76.14	0.041	0.033	116.72	62.20	74.44	0.051	0.032
Paint3D [84]	Text/Image	124.73	69.72	86.47	0.084	0.059	116.61	61.70	80.28	0.063	0.050
MaterialMVP [26]	Image	113.20	54.79	67.67	0.048	0.029	101.66	48.71	66.83	0.035	0.032
UniTEX* [41]	Image	106.45	51.08	69.74	0.036	0.034	105.75	51.62	69.65	0.038	0.034
Ours	Image	107.46	53.49	67.16	0.037	0.032	101.91	46.28	64.19	0.026	0.027

tent representation depends on the density and quality of the input point cloud. To capture high-frequency textures and ensure full mesh coverage – including small triangles and UV-distorted regions – we employ a denser surface sampling strategy (2 million points uniformly sampled from each data) to train the VAE. For diffusion model training, we cache latents from an even denser cloud of 5 million points to provide the highest-quality latent encoding. This dense sampling creates a lossless ground truth, enabling our model to generate finer-grained textures.

5. Experiments

We validate Lafite through comprehensive experiments. We benchmark our core VAE representation (Sec. 5.2), evaluate the full generative pipeline against state-of-the-art methods (Sec. 5.3), ablate our supervision strategy (Sec. 5.4), and showcase downstream applications (Sec. 5.5).

5.1. Implementation Details

We train Lafite on a large-scale dataset of 1M textured 3D assets aggregated from Objaverse [16], Objaverse-XL [15], and TexVerse [90]. Our VAE is trained for 600K steps using 16 NVIDIA A100 GPUs with a batch size of 16. The rectified flow model for albedo generation is trained for 500K steps on 32 A100 GPUs. We employ a progressive curriculum, first training on latents from a 64^3 voxel grid, then fine-tuning on higher-fidelity 128^3 latents. The PBR generator is then fine-tuned from the final albedo model by switching to a PBR material target. All models are trained using the AdamW optimizer [45] with a learning rate of $1e-4$.

5.2. Analysis of the Lafite Representation

We first validate the central claim of our work by evaluating the reconstruction fidelity of our VAE representation.

Setup. We compare our VAE against TRELLIS [73], a state-of-the-art model that represents texture by projecting pre-trained 2D DINO [54] features onto the volume as input. For a fair comparison, both models are tasked with reconstructing the original ground-truth texture from their learned latent representations. We evaluate performance using standard image-space metrics (PSNR, SSIM, LPIPS) computed across multiple rendered views.

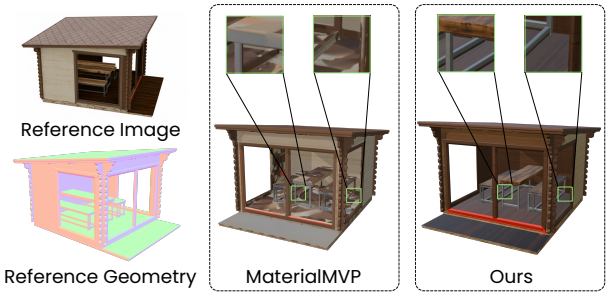

Figure 3. **Robustness to self-occlusion.** Lafite's 3D-native approach generates complete and coherent textures even in highly occluded regions where projection methods fail.

Table 2. **VAE Reconstruction Fidelity.** Our VAE, which uses a direct point cloud input, achieves significantly higher fidelity than the projection-based TRELLIS. '*' denotes the result is derived from the 64 resolution model without fine-tuning.

Method	PSNR↑	SSIM↑	LPIPS↓
TRELLIS-GS64	21.37	0.833	0.116
TRELLIS-RF64	21.83	0.864	0.108
TRELLIS-GS128*	19.28	0.787	0.195
TRELLIS-RF128*	23.07	0.880	0.127
Ours-64	33.57	0.960	0.053
Ours-128*	34.48	0.965	0.041
Ours-128	**34.62**	**0.967**	**0.039**

Results. As demonstrated in Tab. 2 and Fig. 4, Lafite achieves dramatically higher reconstruction quality across all metrics and resolutions, culminating in a *PSNR improvement of over 10 dB* at the 128^3 resolution. This result provides clear evidence for our hypothesis: directly encoding a pristine 3D colored point cloud captures high-frequency surface information far more faithfully than projecting and aggregating 2D features, which inherently suffer from resolution limits, view inconsistencies, and projection ambiguities.

Scalability with Sampling Density. We further analyze the scalability of our representation with respect to the density

Figure 4. **Visual comparison of VAE reconstruction.** Our method reconstructs sharper, more detailed textures, avoiding the blur and artifacts of the TRELLIS baseline.

Table 3. **Effect of Point Density on Reconstruction.** VAE reconstruction quality consistently improves with denser input point cloud sampling, showcasing the model's scalability.

Resolution	#Points	PSNR↑	SSIM↑	LPIPS↓
64	20,480	27.56	0.904	0.124
64	204,800	32.22	0.951	0.067
64	2,048,000	33.49	0.959	0.053
128	20,480	26.07	0.870	0.141
128	204,800	31.60	0.943	0.084
128	2,048,000	34.30	0.964	0.043
128	4,096,000	**34.45**	**0.965**	**0.041**

of the input point cloud. As shown in Tab. 3, reconstruction quality improves consistently with denser sampling. This confirms our model's capacity to leverage richer input data and validates our dense sampling strategy (Sec. 4) as a critical component for achieving maximum fidelity.

5.3. State-of-the-Art Conditional Texture Generation

Having validated our core representation, we now evaluate the full Lafite pipeline on the end-to-end task of conditional texture generation, comparing against both image-conditioned and text-conditioned state-of-the-art baselines.

Experimental Setup. We curate a challenging test set designed to reflect a practical asset creation workflow. We collect approximately 800 diverse 3D meshes generated by commercial AI tools [1–3, 27, 67], using a set of 200 prompt images from various sources [17, 71, 73]. This ensures the evaluation is performed on novel, "in-the-wild" geometries rather than familiar assets from open-source training sets. For text-based methods, we use a powerful VLM (Qwen-2.5VL [5]) to generate prompts from the reference images, ensuring a fair comparison.

Metrics and Evaluation. We render all generated texturing results from three canonical views under both unshaded and shaded settings. For methods that do not generate

Figure 5. **Qualitative comparison (Text-conditioned).** Lafite generates textures with higher fidelity and detail.

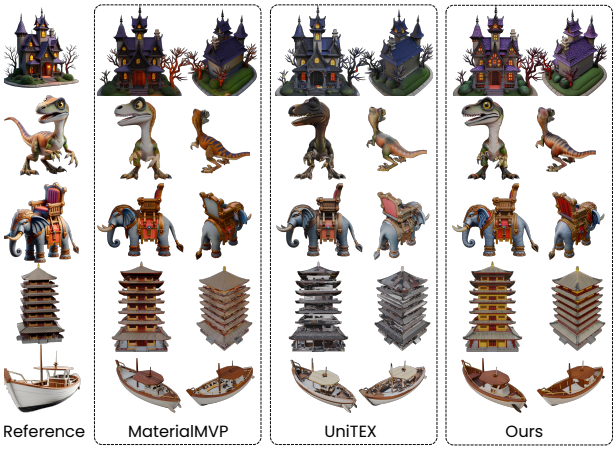

Figure 6. **Qualitative comparison (Image-conditioned).** Lafite's textures are more faithful to the reference image in color and content than prior work.

PBR channels, shaded results are rendered with a default material (metallic=0, roughness=0.5). To measure the fidelity and alignment with the input condition, we compute a suite of standard generative modeling metrics: Fréchet Distance (FD) [4] and Kernel Distance (KD) [56] using feature spaces from CLIP [60] and DINO [54] to capture semantic and stylistic similarity.

Quantitative Results. As shown in Table 1, Lafite demonstrates superior performance across nearly all metrics and conditions. In the image-conditioned task, Lafite achieves the best or second-best scores on all metrics, significantly outperforming prior work in FD_{CLIP} and KD_{DINO} under *shaded* setting. This confirms that our framework, which leverages a high-fidelity representation and precise 3D geometric conditioning, produces textures that are more faithful to the input prompt than any competing method.

Qualitative Results. Beyond metrics, Lafite exhibits clear qualitative advantages, which we analyze in terms of **Integrity**, **Continuity**, and **Versatility**. As visualized in Figures 3, 5, and 6, our method consistently generates textures that fully cover the entire mesh (**Integrity**), are free from UV seams or projection artifacts (**Continuity**), and generalize well to a wide variety of object categories (**Versatility**). In contrast, competing methods often pro-

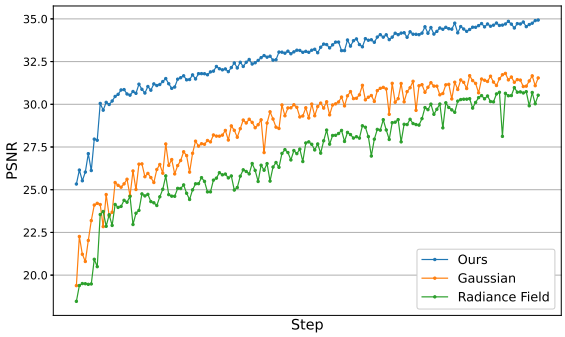

Figure 7. **Ablation on VAE Supervision.** Direct 3D color supervision (Ours) converges faster and to a higher fidelity than supervising on intermediate renderings (3DGS, Radiance Fields).

Table 4. **Ablation on VAE design.** The effectiveness of the point cloud attention and color augmentation in VAE.

	PSNR↑	SSIM↑	LPIPS↓
Ours	**32.69**	**0.962**	**0.063**
Ours w/o point attn	30.83	0.952	0.071
Ours w/o augmentation	31.62	0.958	0.067

duce visible seams, blurred details, or incomplete textures, particularly on complex or self-occluded geometry.

User Study. We conducted an additional user study with a total of 600 evaluations from 20 users – 30 randomly selected in test cases in random order for each user. We compare our method with MaterialMVP and UniTEX: Participants were asked to choose the best result based on texture neatness, consistency with the reference image, and geometric alignment. Results show that our method is preferred in 58.5% of cases, outperforming MaterialMVP (24.7%) and UniTEX (16.8%).

5.4. Ablation Studies

In our VAE design (Sec. 3.1), we supervise the decoder by directly reconstructing raw RGB colors from the input point cloud. To validate this choice, we compare against supervision from intermediate 3D representations, including renderings from 3DGS [33] and NeRF [48]. As shown in Fig. 7, direct 3D supervision yields faster convergence and higher final PSNR, indicating a more stable and unbiased learning signal without rendering artifacts. We further ablate point cloud attention and color jitter augmentation in the VAE in Table. 4.

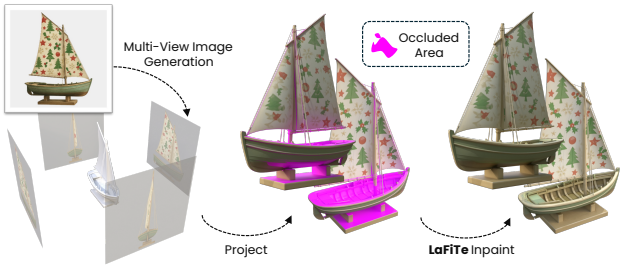

Figure 8. **3D Inpainting with Multi-View Pipelines.** Lafite can be used as a backend to robustly inpaint large occluded regions where 2D projection methods fail.

Figure 9. **Local Editing.** Our structured latent color field enables coherent local editing by regenerating masked regions (green box) while preserving the surroundings (blue box).

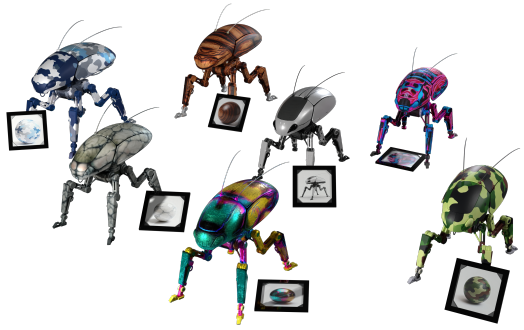

Figure 10. **Material Transfer.** Lafite synthesizes a full PBR material onto a mesh conditioned only on a 2D material image.

5.5. Downstream Applications

A key advantage of our learned latent field is its flexibility. We demonstrate its power through several applications:

- **Integration with Multi-View Pipelines:** Fig. 8 shows Lafite serving as a powerful 3D inpainting backend, robustly filling occluded regions where traditional 2D projection methods fail.
- **Local Editing:** Fig. 9 shows our method enables coherent local editing by masking and regenerating latent voxels.
- **Material Transfer:** Fig. 10 demonstrates direct material transfer, where a 2D image of a material is used to synthesize a full PBR texture on a new mesh.

6. Conclusion

We introduced Lafite, a generative framework that fundamentally advances native 3D texturing. We identified the critical bottleneck as the lack of a powerful, topology-agnostic representation and solved it by learning to generate textures as a 3D latent field. Our VAE-based representation achieves state-of-the-art fidelity, and our unified design for geometric conditioning ensures seamless, high-quality generation. By solving this core representation problem, Lafite significantly outperforms existing methods and provides a robust foundation for future 3D generative workflows.

Limitations and Future Work. Lafite's 3D-native approach lacks the semantic priors of large-scale 2D models, limiting its ability to generate specific 2D patterns like text. Future work should focus on distilling 2D knowledge into our robust 3D representation to combine the geometric integrity of our method with the semantic richness of 2D foundation models.

7. Acknowledgement

This work was supported in part by the International (Hong Kong, Macao, and Taiwan) Collaborative R&D Project, Beijing Major Science and Technology Project under Contract No.Z251100007125016, the National Key Research and Development Program of China (No. 2023YFF0905104), the Natural Science Foundation of China (No. 62132012, 62361146854) and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] Hyper3D / Rodin AI. Rodin – free ai 3d model generator for everyone. https://hyper3d.ai/, 2025. Accessed: 2025-10-30. 7

[2] Meshy AI. Meshy ai – meet the world's most popular and intuitive free ai 3d model generator. https://www.meshy.ai/, 2025. Accessed: 2025-10-30.

[3] Tripo AI. Tripo ai – create your first 3d model with text and image. https://www.tripo3d.ai/, 2025. Accessed: 2025-10-30. 7

[4] Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 327–338. Springer, 1995. 7

[5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 7

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

[7] Michael Birsak, John Femiani, Biao Zhang, and Peter Wonka. Matclip: Light-and shape-insensitive assignment of pbr material models. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–10, 2025. 3

[8] Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4169–4181, 2023. 3

[9] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 2, 3

[10] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 18558–18568, 2023. 2, 3

[11] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22246–22256, 2023. 3

[12] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 2

[13] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[14] Wei Cheng, Juncheng Mu, Xianfang Zeng, Xin Chen, Anqi Pang, Chi Zhang, Zhibin Wang, Bin Fu, Gang Yu, Ziwei Liu, et al. Mvpaint: Synchronized multi-view diffusion for painting anything 3d. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 585–594, 2025. 3, 6

[15] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023. 5, 6

[16] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023. 5, 6

[17] Dylan Ebert. 3d arena: An open platform for generative 3d evaluation. *arXiv preprint arXiv:2506.18787*, 2025. 7

[18] Ye Fang, Zeyi Sun, Tong Wu, Jiaqi Wang, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. Make-it-real: Unleashing large multimodal model's ability for painting 3d objects with realistic materials. *CoRR*, 2024. 3

[19] Simone Foti, Stefanos Zafeiriou, and Tolga Birdal. Uv-free texture generation with denoising and geodesic heat diffusion. *Advances in Neural Information Processing Systems*, 37:128053–128081, 2024. 3

[20] Chenjian Gao, Boyan Jiang, Xinghui Li, Yingpeng Zhang, and Qian Yu. Genesistex: adapting image denoising diffusion to texture space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4620–4629, 2024. 3

[21] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*, pages 213–222, New York, NY, USA, 1984. ACM. 5

[22] Benoit Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks. In *European Conference on Computer Vision*, 2022. 2

[23] Yanhui Guo, Xinxin Zuo, Peng Dai, Juwei Lu, Xiaolin Wu, Youliang Yan, Songcen Xu, Xiaofei Wu, et al. Decorate3d: text-driven high-quality texture generation for mesh decoration in the wild. *Advances in Neural Information Processing Systems*, 36:36664–36676, 2023. 3

[24] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, et al. threestudio: A unified framework for 3d content generation, 2023. 2

[25] Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. *arXiv preprint arXiv:2503.21732*, 2025. 2, 3, 4

[26] Zebin He, Mingxin Yang, Shuhui Yang, Yixuan Tang, Tao Wang, Kaihao Zhang, Guanying Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, et al. Materialmvp: Illumination-invariant material generation via multi-view pbr diffusion. *arXiv preprint arXiv:2503.10289*, 2025. 2, 3, 5, 6

[27] HiTem3D. Hitem3d: Next-gen ai 3d model generator. https://hitem3d.ai/, 2025. Accessed: 2025-10-30. 7

[28] Yijia Hong, Yuan-Chen Guo, Ran Yi, Yulong Chen, Yan-Pei Cao, and Lizhuang Ma. Supermat: Physically consistent pbr material estimation at interactive rates. *arXiv preprint arXiv:2411.17515*, 2024. 3

[29] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[30] Hongxiang Huang, Guoyuan An, Jingzhen Lan, Qi Wang, Lingfei Wang, Rui Wang, and Yuchi Huo. Ultra-high resolution facial texture reconstruction from a single image. *Computational Visual Media*, 2025. 3

[31] Xin Huang, Tengfei Wang, Ziwei Liu, and Qing Wang. Material anything: Generating materials for any 3d object via diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26556–26565, 2025. 3, 6

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

[33] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 8

[34] Hyunjin Kim, Kunho Kim, Adam Lee, and Wonkwang Lee. Goatex: Geometry & occlusion-aware texturing. *arXiv preprint arXiv:2511.23051*, 2025. 3

[35] Peter Kocsis, Lukas Höllein, and Matthias Nießner. Intrinsix: High-quality pbr generation using image priors. *arXiv preprint arXiv:2504.01008*, 2025. 3

[36] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 5

[37] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[38] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025. 2

[39] Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *arXiv preprint arXiv:2505.14521*, 2025. 3

[40] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6517–6526, 2024. 2

[41] Yixun Liang, Kunming Luo, Xiao Chen, Rui Chen, Hongyu Yan, Weiyu Li, Jiarui Liu, and Ping Tan. Unitex: Universal high fidelity generative texturing for 3d shapes. *arXiv preprint arXiv:2505.23253*, 2025. 2, 3, 6

[42] Jialun Liu, Chenming Wu, Xinqi Liu, Xing Liu, Jinbo Wu, Haotian Peng, Chen Zhao, Haocheng Feng, Jingtuo Liu, and Errui Ding. Texoct: Generating textures of 3d models with octree-based diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4284–4293, 2024. 3

[43] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 5

[44] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 2, 3, 5, 6

[45] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 6

[46] Jiawei Lu, Yingpeng Zhang, Zengjun Zhao, He Wang, Kun Zhou, and Tianjia Shao. Genesistex2: Stable, consistent and high-quality text-to-texture generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5820–5828, 2025. 3

[47] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 5

[48] 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. 8

[49] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 3

[50] Jacob Munkberg, Zian Wang, Ruofan Liang, Tianchang Shen, and Jon Hasselgren. Videomat: Extracting pbr materials from video diffusion models. In *Computer Graphics Forum*, page e70180. Wiley Online Library, 2025. 3

[51] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020. 2

[52] Tomoyuki Nishita and Eihachiro Nakamae. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*, pages 23–30, New York, NY, USA, 1985. ACM. 5

[53] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4531–4540, 2019. 2, 3

[54] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 3, 6, 7

[55] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 2

[56] Jeff M. Phillips and Suresh Venkatasubramanian. A geometric approach to comparing datasets via kernel distance. In *Proceedings of the 27th Annual Symposium on Computational Geometry (SoCG)*, pages 317–326. ACM, 2011. 7

[57] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[58] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[59] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 7

[61] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023. 3

[62] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 3

[63] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces. In *European Conference on Computer Vision*, pages 72–88. Springer, 2022. 3

[64] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19615–19625, 2024. 2

[65] Yawar Siddiqui, Tom Monnier, Filippos Kokkinos, Mahendra Kariya, Yanir Kleiman, Emilien Garreau, Oran Gafni, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, et al. Meta 3d assetgen: Text-to-mesh generation with high-quality geometry, texture, and pbr materials. *Advances in Neural Information Processing Systems*, 37:9532–9564, 2024. 3

[66] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. 3

[67] Tencent Hunyuan3D Team. Tencent hunyuan3d. https://3d.hunyuan.tencent.com/, 2025. Accessed: 2025-10-30. 2, 7

[68] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12619–12629, 2023. 2

[69] Li Wang, Yu-Tao Liu, Jie Yang, Weikai Chen, Xiaoxu Meng, Bo Yang, Jintao Li, and Lin Gao. Hsdf: Hybrid sign and distance field for neural representation of surfaces with arbitrary topologies. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):5215–5228, 2024. 2

[70] 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 distilla-

tion. *Advances in neural information processing systems*, 36: 8406–8441, 2023. 2

[71] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412*, 2025. 2, 7

[72] Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. Recent advances in 3d gaussian splatting. *Computational Visual Media*, 10(4):613–642, 2024. 2

[73] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 2, 3, 4, 6, 7

[74] Zhiyu Xie, Yuqing Zhang, Xiangjun Tang, Yiqian Wu, Dehan Chen, Gongsheng Li, and Xiaogang Jin. Styletex: Style image-guided texture generation for 3d models. *ACM Transactions on Graphics (TOG)*, 43(6):1–14, 2024. 3

[75] Bojun Xiong, Jialun Liu, Jiakui Hu, Chenming Wu, Jinbo Wu, Xing Liu, Chen Zhao, Errui Ding, and Zhouhui Lian. Texgaussian: Generating high-quality pbr material via octree-based 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 551–561, 2025. 3

[76] Jiamin Xu, Zihan Zhu, Hujun Bao, and Weiwei Xu. Hybrid mesh-neural representation for 3d transparent object reconstruction. *Computational Visual Media*, 11(1):123–140, 2025. 2

[77] Mingye Xu, Zhipeng Zhou, Yali Wang, and Yu Qiao. Towards robustness and generalization of point cloud representation: A geometry coding method and a large-scale object-level dataset. *Computational Visual Media*, 10(1):27–43, 2024. 3

[78] Dongyu Yan, Leyi Wu, Jiantao Lin, Luozhou Wang, Tianshuo Xu, Zhifei Chen, Zhen Yang, Lie Xu, Shunsi Zhang, and Yingcong Chen. Flexpainter: Flexible and multi-view consistent texture generation. *arXiv preprint arXiv:2506.02620*, 2025. 3

[79] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4304–4314, 2024. 3

[80] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. Texture generation on 3d meshes with point-uv diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4206–4216, 2023. 3

[81] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415*, 2023. 2

[82] Xin Yu, Ze Yuan, Yuan-Chen Guo, Ying-Tian Liu, Jianhui Liu, Yangguang Li, Yan-Pei Cao, Ding Liang, and Xiaojuan Qi. Texgen: a generative diffusion model for mesh textures. *ACM Transactions on Graphics (TOG)*, 43(6):1–14, 2024. 2, 3, 6

[83] Ze Yuan, Xin Yu, Yangtian Sun, Yuan-Chen Guo, Yan-Pei Cao, Ding Liang, and Xiaojuan Qi. Seqtex: Generate mesh textures in video sequence. *arXiv preprint arXiv:2507.04285*, 2025. 3, 6

[84] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4252–4262, 2024. 6

[85] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. Rgb↔x: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 3

[86] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023. 2

[87] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2024. 2

[88] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024. 2, 3

[89] Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, et al. Dreammat: High-quality pbr material generation with geometry-and light-aware diffusion models. *ACM Transactions on Graphics (TOG)*, 43(4):1–18, 2024. 3

[90] Yibo Zhang, Li Zhang, Rui Ma, and Nan Cao. Texverse: A universe of 3d objects with high-resolution textures. *arXiv preprint arXiv:2508.10868*, 2025. 6

[91] Shenhao Zhu, Lingteng Qiu, Xiaodong Gu, Zhengyi Zhao, Chao Xu, Yuxiao He, Zhe Li, Xiaoguang Han, Yao Yao, Xun Cao, et al. Mcmat: Multiview-consistent and physically accurate pbr material generation. *arXiv preprint arXiv:2412.14148*, 2024. 3

[92] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10324–10335, 2024. 2