CRAFT: REFERENCE-GUIDED SURFACE-ADDRESSED TEXTURE GENERATION FOR COMPOSABLE 3D ASSETS

Anonymous authors
Paper under double-blind review

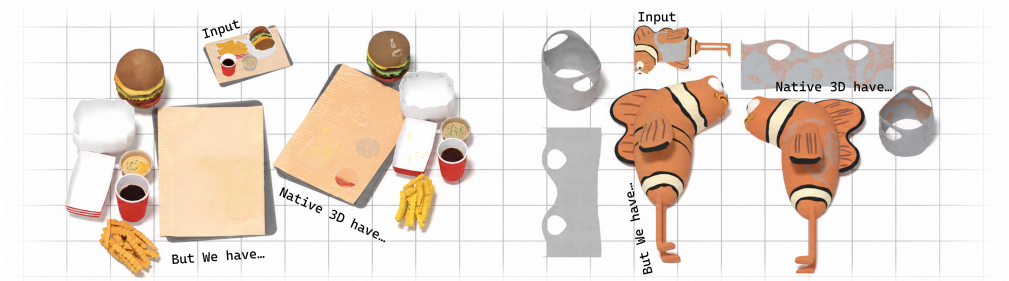

Figure 1: CRAFT generates textures that remain associated with their surfaces when contacting or layered components are separated. The reference images guide visible surfaces, while the native 3D prior completes unobserved regions.

ABSTRACT

While progress in native 3D texture generation can complete appearance beyond an observed view, existing texture generators often suffer from appearance bleeding when distinct geometric parts touch or intersect. This is because spatial voxelization can aggregate texture samples from multiple surfaces into a single texture state. Reference-image conditioning introduces a second ambiguity: features observed on a visible surface may be incorrectly transferred to nearby or occluded surfaces that share the same voxel. We introduce CRAFT, a native 3D texture-generation framework that augments each voxel coordinate with a discrete surface identity derived from UV-chart connectivity. This surface address preserves independent texture features for different surfaces throughout encoding, generation, and decoding. We further introduce visibility-routed reference conditioning, to establish accurate correspondence between visible surface tokens and reference-image features according to surface address and visibility. These designs preserve surface-level texture correspondence while retaining the efficiency of a sparse voxel-based representation. As a result, CRAFT reproduces the reference appearance more faithfully while reducing cross-surface color leakage and keeping distinct surface regions more clearly separated for component-level editing, separation, and recombination.

1 INTRODUCTION

Recent advances (Lai et al., 2026a; He et al., 2025a) in 3D generative modeling have brought generated assets closer to real-world production workflows. However, gaps remain that prevent existing generative texturing methods from being used in practice. One major limitation is their ability to properly texture multi-part geometries that may touch, intersect, or overlap with one another. For example, for a character wearing a white T-shirt, properly and separately texturing the character's body and the T-shirt remains challenging.

Current generative texturing approaches can be broadly grouped into multi-view texturing, UV-space texturing, and Native 3D generation. Multi-view texturing methods (Liu et al., 2024b; Liang et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2026; He et al., 2025b) often suffer from texture bleeding at the junctions or intersections between geometry parts due to multi-view inconsistency. Occlusion is also challenging, as occluded surfaces are not observable from the camera views. UV-space methods perform texture completion in 2D. Hybrid approach (Yu et al., 2024) additionally aggregate features in 3D space to capture geometric relationships. However, spatial pooling can mix texture features from distinct geometry parts that are spatially close. Native 3D generation methods (Xiang et al., 2026; Lai et al., 2026b; Chen et al., 2026; Zeng et al., 2026; Liu et al., 2026) instead model texture directly in 3D, but may still exhibit texture bleeding between touching, intersecting, or overlapping surfaces. We focus on voxel-based representations, where texture features from distinct surfaces may be merged within the same voxel, as shown in Fig. 2. We refer to this problem as *intra-voxel texture entanglement*. Reference-image conditioning introduces a second ambiguity. Routing image features by spatial position without resolving surface identity and visibility can transfer appearance to the unintended surfaces. For example, a white T-shirt may incorrectly colorize the body it occludes or touches. We refer to this problem as *surface-ambiguous reference injection*. Partitioning the space into even smaller voxels only defers, rather than solves, these problems.

To address this problem, the model must distinguish surfaces that touch or intersect in 3D but should remain texturally separate. Our key observation is that artist-created meshes typically come with their original UV atlases, whose connectivity often reveals this distinction. For example, although a character's body and white T-shirt may touch in 3D, they are semantically distinct and are likely disconnected in the UV domain. We therefore use UV connectivity to partition the mesh into surface groups and assign each group a surface index k. This effectively maps the address of each voxel token from the original 3D coordinates (x, y, z) to an augmented surface address (k, x, y, z) in a 4D space. This augmented address enables unambiguous attribute aggregation, VAE encoding and decoding, and token processing within the generative model. Information can be exchanged within each surface group, while features from different groups are prevented from being forcibly merged through spatial aggregation. Our design derives surface indices from the mesh's UV connectivity without additional semantic part annotations and remains compatible with standard UV texture-editing and material-authoring workflows.

While surface groups retain separate texture features, coordinating textures across groups remains challenging because separating local feature aggregation limits cross-group information exchange. Global cross-attention to DINOv3 features (Siméoni et al., 2026) provides shared image guidance, but we observe that this guidance alone does not reliably associate reference texture features with the intended surfaces, resulting in misplaced colors. We therefore extend the pixel-to-voxel back-projection scheme of Pixal3D (Li et al., 2026) with surface address and First-Hit Visibility Cache, introducing Visibility-Routed Reference Conditioning. The First-Hit Visibility Cache records the first surface index k reached by each reference pixel, establishing a pixel-to-surface correspondence in the 4D space. This allows reference features to be injected at corresponding 4D surface tokens during learning and inference. Our routing provides each surface group with corresponding reference features, helping coordinate textures across groups while keeping distinct surfaces texturally separate. During inference, we construct the cache deterministically from the input mesh and calibrated reference camera, enabling the same 4D conditioning mechanism used during training.

Together, our approach transforms a partial image observation into a complete 3D texture. Surface addresses preserve surface-specific texture states, visibility routing assigns reference texture features to their corresponding visible surfaces, and the native 3D prior completes unobserved regions. The practical benefits extend beyond matching the reference appearance in rendered views. Our experiments demonstrate surface separation and recombination, as well as image-guided texture editing, with edited textures remaining consistent across the shown viewpoints and retaining surface-specific texture states after separation. Our main contributions are as follows:

- We propose a surface-addressed representation that preserves distinct texture states within the same voxel throughout encoding, generation, and decoding.
- We introduce *Visibility-Routed Reference Conditioning*, using a *First-Hit Visibility Cache* to route reference texture features by surface address and visibility.
- By fine-tuning a pretrained 3D generative model, we reduce cross-surface texture bleeding and enable surface separation, recombination, and image-guided texture editing.

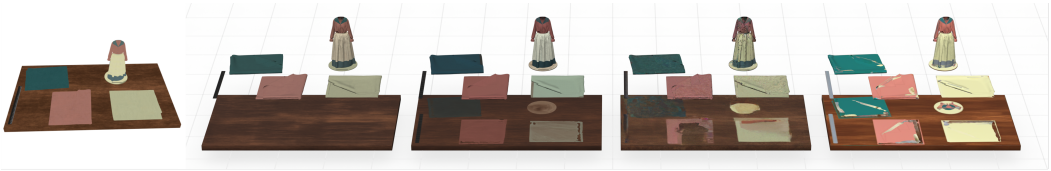

Figure 2: **Cross-surface texture bleeding revealed by surface separation.** Given the input reference (left), lifting the objects exposes texture contamination on the underlying tabletop in native 3D, UV-based, and multiview back-projection results. CRAFT preserves surface-specific textures, reducing the transfer of object colors and patterns onto the tabletop.

2 Related Work

2.1 View- and UV-Space Texture Generation

A common paradigm for 3D texturing synthesizes appearance in 2D and transfers it onto a 3D surface. Early methods use pretrained 2D diffusion priors (Rombach et al., 2022) to sequentially paint and inpaint rendered views (Richardson et al., 2023; Chen et al., 2023a), making them sensitive to view ordering and accumulated cross-view inconsistencies. Related methods optimize appearance through per-object SDS objectives (Poole et al., 2023; Chen et al., 2023b; Lin et al., 2023), primarily targeting broader 3D asset generation.

Recent approaches improve cross-view coherence by jointly synthesizing camera-conditioned views (Shi et al., 2024; 2023; Long et al., 2024; Li et al., 2024; Wang & Shi, 2023) or coordinating denoising across views through geometric or surface correspondences (Liu et al., 2024b; Cheng et al., 2025; Gao et al., 2024; Lu et al., 2025). Depth and normal maps provide geometric guidance (Zeng et al., 2024; Huang et al., 2025), while related methods extend synthesis to relightable PBR materials (Zhang et al., 2024b; He et al., 2025b).

UV-space methods instead generate or refine texture maps directly on surface parameterizations (Yu et al., 2023; 2024; Liu et al., 2025). These paradigms face different correspondence challenges: view-space methods must reconcile partial, potentially inconsistent observations, whereas UV-space methods must maintain consistency across chart boundaries that separate neighboring 3D surface regions.

2.2 Native 3D Appearance Generation

Native 3D appearance representations include mesh-face features (Siddiqui et al., 2022), continuous texture fields (Oechsle et al., 2019), octree-organized point colors (Liu et al., 2024a), octree-aligned Gaussians (Xiong et al., 2025), and hybrid triplane–cube features (Liang et al., 2026). TRELLIS (Xiang et al., 2025) jointly represents geometry and appearance through voxel-indexed structured latents.

Recent native texturing methods learn appearance representations from 3D surface data, broadly adopting set-based latents (Zhang et al., 2023; 2024a; Hao et al., 2026) or spatially indexed sparse latents (Ren et al., 2024; He et al., 2025a). NaTex (Lai et al., 2026b) compresses colored surface samples into a fixed-size latent set through a geometry-aware color VAE. Spatially indexed methods instead attach appearance latents to explicit 3D coordinates. LaFiTe (Chen et al., 2026) encodes samples within sparse spatial cells for local continuous decoding, while TEXTRIX (Zeng et al., 2026) learns compact attribute latents over a sparse voxel grid. TRELLIS.2 (Xiang et al., 2026) jointly models geometry and PBR materials using O-Voxel-aligned sparse representations.

These spatially indexed formulations provides localized appearance states, but a spatial address alone does not distinguish multiple surfaces within the same cell. We augment it with a discrete surface identity, allowing surface-specific appearance states to coexist at the same voxel coordinate and remain distinct throughout encoding and generation.

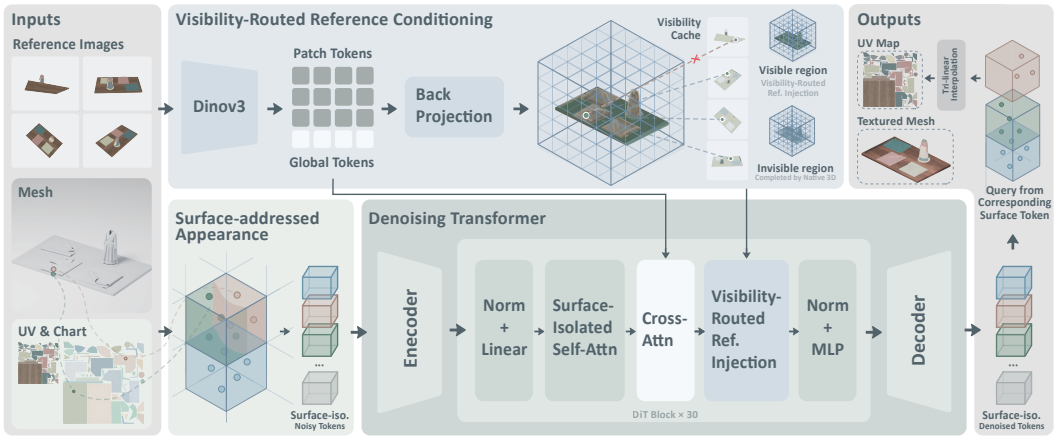

Figure 3: **CRAFT overview.** Surface groups are defined by UV-chart connectivity. Surface-address packing preserves these groups through the frozen VAE and surface- restricted DiT self-attention. Global image features provide shared context, while the first- hit visibility cache routes local reference features to visible surface tokens. Decoded textures are associated with their source charts.

2.3 Spatially Aligned Conditioning for Native 3D Appearance Generation

Recent methods guide native 3D texturing with spatially aligned image evidence. Hitem3D 2.0 (He et al., 2026) projects synthesized multiview appearance onto the surface for native 3D completion. TEXTRIX (Zeng et al., 2026) projects an aligned canonical view into its attribute grid, while Pixal3D (Li et al., 2026) further establishes explicit pixel-to-3D correspondence by back-projecting pixel-level image features into a 3D feature volume.

However, spatial correspondence alone can remain ambiguous in voxel-based representations. When multiple distinct surfaces occupy the same spatial cell, a voxel coordinate does not specify which surface should receive the observed appearance. Our method addresses this ambiguity by augmenting spatial correspondence with surface identity and visibility. Reference features are routed to the corresponding visible surface records, while unobserved surface records are completed by the native 3D appearance prior.

3 Method

Given an input mesh $\mathcal{M}$ with a UV atlas and calibrated reference images $\mathcal{I} = \{(I_v, \Pi_v)\}_{v=1}^{V}$, CRAFT generates a texture that follows the visible reference appearance and completes unobserved surfaces. We build on TRELLIS.2 and retain its PBR latent parameterization; our reference conditioning and evaluation concern base color (albedo). Fig. 3 summarizes the representation and conditioning paths.

3.1 Voxel Representation and Surface Ambiguity

Given a textured mesh $\mathcal{M}$ with surface $\mathcal{S}$, its texture can be described as a continuous surface-color field $\boldsymbol{c} : \mathcal{S} \to \mathbb{R}^3$, which associates each surface position $\boldsymbol{x} \in \mathcal{S}$ with an RGB color $\boldsymbol{c}(\boldsymbol{x})$. Existing native 3D methods (Xiang et al., 2026; 2025) discretize this continuous surface-color field using a sparse voxel grid. At resolution N, the resulting representation can be abstracted as a set of voxel-color feature pairs

$$\mathcal{C} = \{(\boldsymbol{p}_i, \boldsymbol{c}_i)\}_{i=1}^{L}, \qquad \boldsymbol{p}_i \in \{0, \dots, N-1\}^3, \quad \boldsymbol{c}_i \in \mathbb{R}^C, \tag{1}$$

where $\boldsymbol{p}_i$ is the quantized coordinate of the i-th active voxel and $\boldsymbol{c}_i$ is its associated color feature. Given these pairs, the color feature at voxel coordinate $\boldsymbol{p}$ can be retrieved as $\boldsymbol{c}[\boldsymbol{p}]$. The retrieved color feature is then encoded into a compact sparse latent volume using a sparse-convolutional VAE. Existing native 3D methods employ VAEs with spatial downsampling factors of up to 16,

reducing high-resolution color volumes from 1024^3 to latent grids as compact as 64^3. At the 64^3 resolution, the VAE compresses the L active color voxels into L_z active latent tokens, where the j-th token, $j \in \{1, \ldots, L_z\}$, summarizes the color information within a local region of the original color volume. During generation, a diffusion transformer (DiT) denoises the latent tokens. At each $\mathbf{q}_j$, the corresponding geometry feature $\mathbf{g}_j$ is concatenated with the noisy latent $\mathbf{z}_j$, while reference-image features guide denoising through global cross-attention (Lai et al., 2026b) or spatial back-projection (Li et al., 2026; Zeng et al., 2026). The VAE decodes the denoised tokens $\{\hat{\mathbf{z}}_j\}_{j=1}^{L_z}$ into a high-resolution sparse color volume $\{(\mathbf{p}_i, \hat{\mathbf{c}}_i)\}_{i=1}^{L}$, from which surface colors are queried at mesh-surface positions. This pipeline combines image evidence with the native 3D prior learned from complete textured assets. However, two ambiguities remain: voxelization aggregating by $\mathbf{p}$ can merge different surfaces, while conditioning by $\mathbf{q}$ can assign one reference feature to multiple surfaces at the same latent coordinate. Addressing them requires separate texture states and visible-surface correspondence, respectively. CRAFT follows the TRELLIS.2 backbone and solves the ambiguities by adding surface-addressed representation and visibility-routed spatial back-projection.

3.2 Surface-Addressed Appearance Generation

To preserve surface-level color independence, we assign each sampled point $\mathbf{x} \in \mathcal{S}$ a surface index $s(\mathbf{x}) \in \{1, \ldots, K\}$. After voxelization, samples are aggregated only if they share both the same surface index and the same voxel coordinate. The resulting representation is

$$\mathcal{C}_{\text{surf}} = \{(k_i, \boldsymbol{p}_i, \boldsymbol{c}_i)\}_{i=1}^{L_{\text{surf}}}, \qquad k_i \in \{1, \ldots, K\}, \quad \boldsymbol{p}_i \in \{0, \ldots, N-1\}^3, \quad \boldsymbol{c}_i \in \mathbb{R}^C, \tag{2}$$

where $k_i = s(\mathbf{x}_i)$ denotes the surface index associated with the i-th token. Multiple tokens may share the same voxel coordinate $\mathbf{p}_i$ while remaining distinct through their surface indices. After aggregation, each valid address $(k, \mathbf{p})$ corresponds to a unique color feature. Thus, the conventional single-valued voxel field $\mathbf{c}[\mathbf{p}]$ is replaced by a surface-indexed field $\mathbf{c}[k, \mathbf{p}]$. We preserve this surface address through latent encoding, analogously lifting the latent field from $\mathbf{z}[\mathbf{q}]$ to $\mathbf{z}[k, \mathbf{q}]$.

Surface-address construction from UV charts. In principle, our surface-addressed representation is compatible with different choices of discrete surface partition. In this work, we use the UV charts of the input mesh to define these identities, motivated by the fact that artist-created meshes commonly retain their original UV atlases, whose chart boundaries often align with meaningful surface discontinuities. Importantly, UV connectivity is used only to provide a discrete surface address; color representation and generation remain entirely in the 3D voxel domain, where spatial neighborhoods are defined by voxel-space proximity. Since a raw 3D asset may be associated with multiple texture maps, we first repack them into a unified texture canvas while preserving the original UV-chart identity of each texel. Then, we densely sample the resulting 2K texture and map each sample to its corresponding 3D surface position, following (Zeng et al., 2026), while retaining its UV-chart identity. After voxelization, samples within each voxel are grouped by surface identity and averaged independently, yielding the voxelized surface-indexed field $\mathbf{c}[k, \mathbf{p}]$, where a single voxel may retain multiple color features from different surfaces.

Surface-addressed feature packing and latent modeling. The surface-indexed field is packed into a sparse tensor by placing different surfaces along the batch dimension and encoded with the frozen TRELLIS.2 texture VAE. Since sparse convolutions do not mix features across batch entries, surface identities remain isolated throughout encoding and decoding while sharing the same VAE weights. The resulting surface-indexed latents are then modeled by the Texture DiT using the native flow-matching objective, with sparse self-attention restricted within each surface and object-level geometry features gathered at the corresponding spatial coordinates. After decoding, the features are reassociated with their original surfaces and baked back to the source UV atlas. Reference-image conditioning is described separately in Sec. 3.3.

3.3 Visibility-routed reference conditioning.

Although our surface-addressed representation preserves distinct appearance states within the same voxel, reference features must still be routed to the correct surface token. To this end, we introduce

visibility-routed reference conditioning. Our conditioning design builds on the pixel-to-voxel back-projection introduced by Pixal3D (Li et al., 2026). Pixal3D extracts DINOv2 patch features, upsamples them to the image resolution using NAF, and aggregates the resulting pixel-level features in a voxel grid. We retain the pixel-aligned back-projection principle but use a frozen DINOv3 (Siméoni et al., 2026) encoder to extract the reference features. The patch features are upsampled using NAF (Chambon et al., 2026) before being back-projected into the latent grid, thereby preserving fine-grained feature details. Pixal3D aggregates reference features according to voxel coordinates. However, voxel-level back-projection remains ambiguous when multiple surfaces share the same latent coordinate $\mathbf{q}$, as a feature observed from one visible surface may also be assigned to another surface at the same location. We therefore introduce visibility-routed reference conditioning, which augments pixel-to-voxel correspondence with surface identity and visibility, routing each reference feature only to its first-hit surface token $\mathbf{z}[k, \mathbf{q}]$.

Specifically, for each reference image I_v, we construct a first-hit visibility cache C_v. Each valid pixel (u, w) is mapped to a packed token index $C_v(u, w) = j$, where j denotes the j-th surface token identified by the first-hit triangle and its quantized latent coordinate $(k, \mathbf{q})$ at the 64^3 latent resolution. Thus, the cache identifies not only the voxel intersected by each camera ray, but also the specific surface token visible at that pixel. During training, a surface token receives reference conditioning only if its surface ID and voxel coordinate appear in the first-hit cache of the current view. When multiple pixels are mapped to the same token, their reference features are averaged. Tokens without matched pixels are treated as invisible and receive no reference conditioning. Including the surface ID in the correspondence prevents feature mixing between different surfaces within the same voxel, ensuring that each reference feature is injected only into its observed surface. Appendix B illustrates the detailed algorithms.

3.4 Training and Inference

We fine-tune the pretrained TRELLIS.2 material DiT on surface-addressed latents using its rectified-flow objective and noise schedule. We optimize all DiT parameters and the zero-initialized reference projections while keeping the material VAE, shape VAE, and DINOv3 encoder frozen. Geometry latents are concatenated with noisy material latents at the corresponding spatial coordinates.

Training uses 16 calibrated views with local-view dropout and a fixed four-view subset for global context. Image-derived conditions are jointly dropped with probability 0.1 for classifier-free guidance. Appendix C specifies the local and global dropout behavior and optimization settings. At inference, we construct surface addresses and first-hit caches from the input mesh and supplied reference cameras, then sample with 25 Euler steps and a guidance scale of 3.0. The frozen VAE decodes each surface group independently, and chart-aware baking produces a 2048×2048 base-color atlas (Appendix D).

4 Experiments

We evaluate the two mechanisms identified in the introduction: preserving separate texture states and assigning reference features to the visible surface. We first compare reference-view fidelity and occluded-region texture quality, then use generation ablations and VAE reconstruction to examine each mechanism. Component separation exposes contact regions, and image-guided editing illustrates the use of the resulting surface correspondence.

4.1 Setup

We train on 20k curated assets from TexVerse (Zhang et al., 2025) and evaluate on 105 Objaverse-XL (Deitke et al., 2023) objects with composable components. UV textures and chart-index maps have resolution 2048×2048, and the 16 calibrated training views have resolution 1024×1024. We fine-tune for 160K iterations on eight 80 GB GPUs with batch size one per GPU. The reported generation comparison uses a single reference view.

We compare against MaterialMVP (He et al., 2025b), TexGen (Yu et al., 2024), TRELLIS.2 (Xiang et al., 2026), and UniTEX (Liang et al., 2026). SSIM, PSNR, and LPIPS evaluate visible-view reconstruction. LPIPS, CLIP score, CLIP FID, and CMMD evaluate occluded-region generation, as

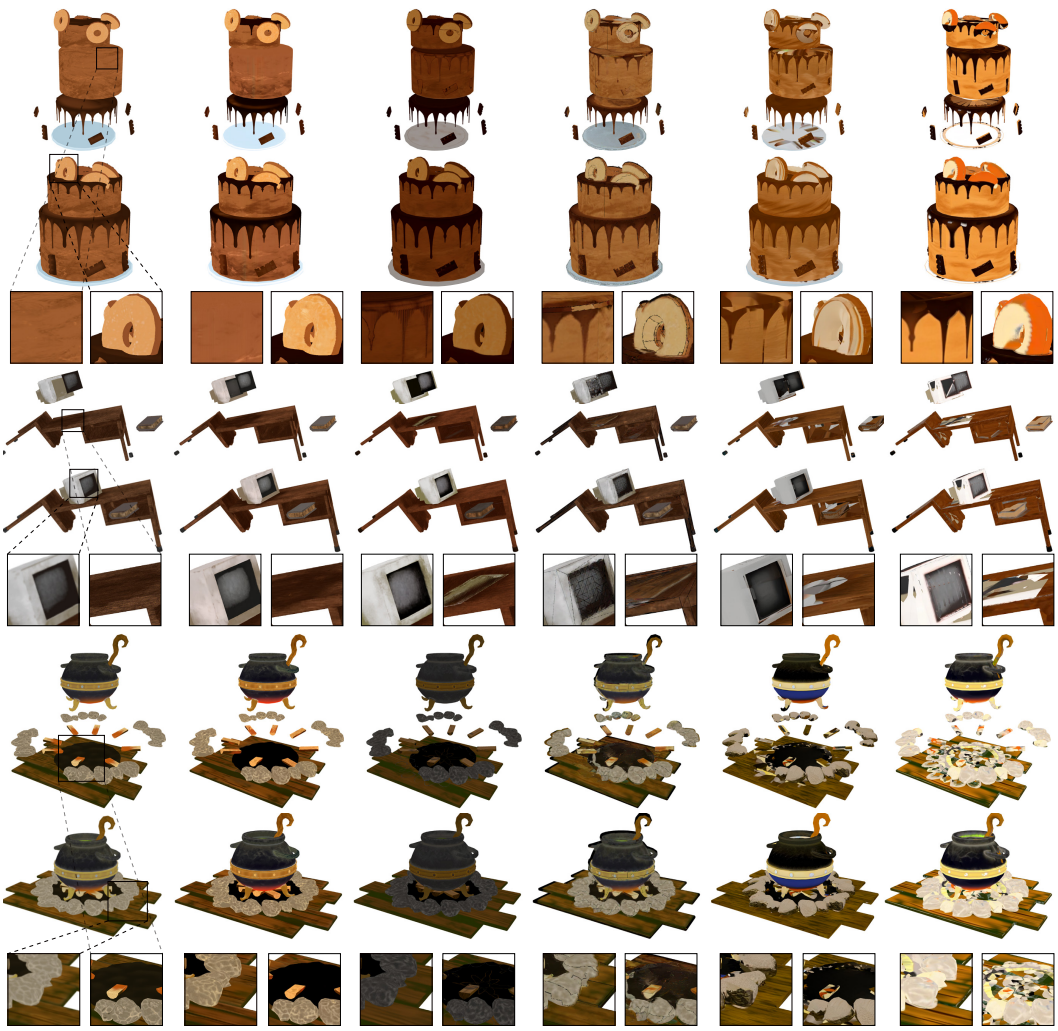

Figure 4: **Single-view texture generation.** For each example, the upper row separates components to expose surfaces occluded in the reference view; the lower row uses the reference viewpoint. The separated views reveal whether colors and patterns remain associated with their intended surfaces.

reported in Table 1. To expose hidden contacts, we separate each asset's source components using the same displacement for ground truth and all methods, then render them with the same camera. Appendix E.1 details the setting of occluded-region evaluation.

4.2 Reference Fidelity and Occluded-Region Completion

Table 1 reports the strongest scores for CRAFT on all listed metrics. Visible-view PSNR reaches 27.176 dB, compared with 25.796 dB for TexGen, and LPIPS decreases from 0.0668 to 0.0554. These results support improved recovery of the observed reference appearance. On occluded regions, CRAFT obtains LPIPS 0.0503, compared with 0.0755 for TRELLIS.2, and also improves CLIP score, CLIP FID, and CMMD. The visible and occluded evaluations therefore support both reference alignment and completion quality.

Fig. 4 shows why component separation matters for this comparison. The assembled view can conceal texture transferred between contacting surfaces. Separating them reveals transferred patterns and misplaced colors in baseline results, whereas CRAFT better retains the underlying texture at the contacts identified in Fig. 2. These renderings provide spatial evidence of reduced cross-surface bleeding, complementing the aggregate image metrics.

Table 1: **Quantitative comparison on visible and occluded region generation.**

	Visible-View Reconstruction			Occluded-Region Generation			
Method	SSIM ↑	PSNR ↑	LPIPS ↓	LPIPS ↓	CLIP ↑	CLIP FID ↓	CMMD ↓
MaterialMVP	0.9447	23.322	0.0685	0.0909	0.7870	33.500	29.919
TexGen	0.9535	25.796	0.0668	0.0858	0.7970	33.056	32.399
TRELLIS.2	0.9347	21.313	0.0713	0.0755	0.7645	36.891	38.839
UniTEX	0.9430	22.816	0.0682	0.0914	0.7879	34.292	32.947
Ours	**0.9567**	**27.176**	**0.0554**	**0.0503**	**0.8113**	**30.158**	**24.903**

Table 2: **Ablation study.** Effect of each component on reconstruction quality.

Method	PSNR↑	SSIM↑	LPIPS↓
Ours	**24.454**	**0.9429**	**0.0503**
w/o surface-address packing	23.453	0.9320	0.0578
w/o BP. Ref. Injection	22.572	0.9338	0.0638
w/o visibility	22.912	0.9325	0.0649

Table 3: **VAE Reconstruction w/ and w/o surface-address packing.**

Layout	PSNR↑	SSIM↑	LPIPS↓
Unpack	26.466	0.9554	0.0414
Pack	**26.868**	**0.9588**	**0.0357**

4.3 Ablation Study

We evaluate the contribution of each component by comparing the full model with three variants in Table 2. The first removes surface-address packing and collapses the surface-indexed state $\mathbf{c}[k, \mathbf{p}]$ into a conventional voxel state $\mathbf{c}[\mathbf{p}]$, with reference features injected at the corresponding voxel coordinate $\mathbf{q}$ (w/o surface-address packing). The second retains the surface-addressed representation but replaces first-hit routing with voxel-level injection, broadcasting each reference feature to all surface tokens sharing $\mathbf{q}$ (w/o visibility). The third retains surface addressing and global image conditioning but removes the back-projected local reference features (w/o back-projection reference injection). These variants examine state separation, surface-specific correspondence, and the value of local reference evidence. Fig. 5 illustrates the changes in local color placement and detail.

Without surface-address packing, PSNR drops from 24.454 to 23.453 dB and LPIPS rises from 0.0503 to 0.0578. Without first-hit routing, PSNR falls to 22.912 dB and LPIPS rises to 0.0649. Thus, retaining separate states alone is insufficient when reference features are still broadcast across co-located surfaces. Removing local reference injection yields 22.572 dB PSNR and 0.0638 LPIPS, showing the benefit of spatially aligned features beyond global context.

To isolate the representation path, Table 3 compares reconstruction with the frozen VAE under the two packing layouts. Surface-address packing increases PSNR from 26.466 to 26.868 dB and decreases LPIPS from 0.0414 to 0.0357. Because this comparison reconstructs supplied textures rather than sampling new ones, it provides evidence for preserving surface-specific information during encoding and decoding. Fig. 6 exposes the corresponding contact-region artifacts: surface addressing reduces transferred outlines in VAE reconstruction, and retaining it during generation reduces the analogous contamination. Together, the reconstruction and generation comparisons connect the observed bleeding to both stages of the pipeline.

4.4 Surface Separation and Image-Guided Editing

The separated renderings in Figs. 1, 2, and 4 test whether texture remains associated with the intended surface when components are moved. They illustrate the relevance of surface-specific texture states to separation and recombination. For image-guided editing, we modify a rendering from a selected viewpoint and route the edited reference features to its first-hit surface addresses. Surface-addressed token exchange associates updates with the same surface across views. Fig. 7 shows edited textures from multiple viewpoints and after component separation, illustrating the retention of surface-specific texture assignments.



Figure 5: **Generation ablations corresponding to Table 2.** The figure's "w/o visibility" variant removes first-hit routing, and "w/o back projection reference injection" removes local reference injection. Insets compare texture details near adjacent surfaces.

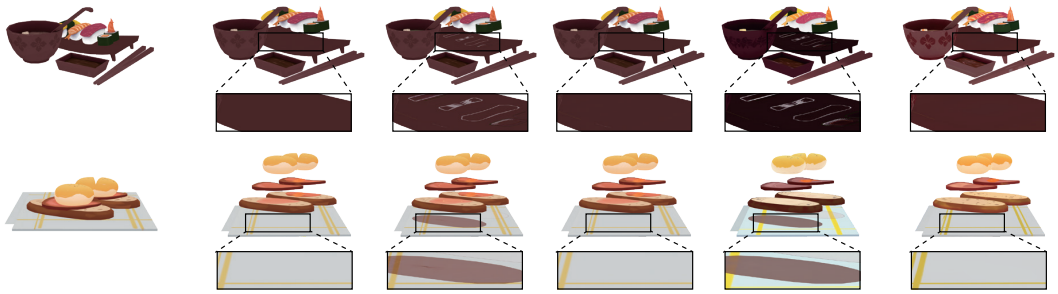

Figure 6: **Surface addressing (SA) in reconstruction and generation.** "VAE w/o SA" and "VAE w/ SA" reconstruct supplied textures with the frozen VAE; "Gen. w/o SA" removes surface addressing during generation. Separated components expose transferred outlines on underlying surfaces. * denotes the input image.

5 Conclusion

CRAFT addresses intra-voxel texture entanglement and surface-ambiguous reference injection by preserving surface addresses and routing local image features through first-hit visibility. The reported comparisons support improved visible-view fidelity and occluded-region texture quality, while VAE reconstruction and generation ablations support the roles of state separation and feature assignment. The resulting textures retain surface-specific detail in the demonstrated separation, recombination, and editing examples.

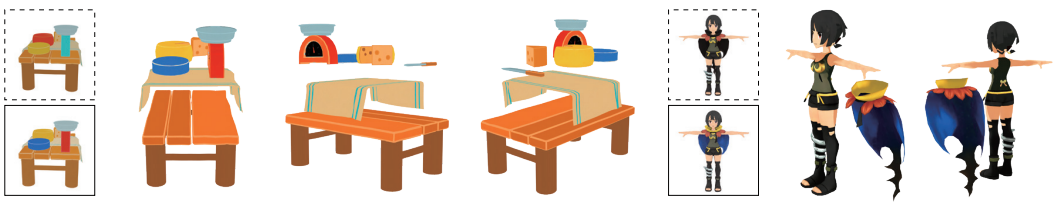

Figure 7: **Image-guided texture editing.** An edited reference guides the corresponding surface addresses. The shown viewpoints and separated components illustrate how the edited texture remains associated with its target surface.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the conceptual development and refinement of the proposed method, provide feedback on research methodology and experimental design, support method implementation and data preparation, assist with translation, and support qualitative analysis and interpretation of experimental results. Other required-disclosure tasks that are not relevant to this work were not applicable.

Additionally, we used generative AI tools to summarize and analyze existing literature, identify relevant prior work and research gaps, and assist with manuscript preparation, including improving readability, organizing the paper structure, formatting references, and suggesting the paper title and keywords.

We have reviewed all AI-assisted work. Methodological and experimental suggestions were assessed by the authors before adoption, AI-assisted implementation and data-processing procedures were reviewed and tested as part of the research workflow, and literature-related content and references were checked against the original sources. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Loïck Chambon, Paul Couairon, Eloi Zablocki, Alexandre Boulch, Nicolas Thome, and Matthieu Cord. Naf: Zero-shot feature upsampling via neighborhood attention filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26604–26613, 2026.

Chia-Hao Chen, Yuan-Chen Guo, Zi-Xin Zou, Ze Yuan, Guan Luo, Xiaojuan Qi, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. LaFiTe: A generative latent field for 3d native texturing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19960–19971, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Chen_Lafite_A_Generative_Latent_Field_for_3D_Native_Texturing_CVPR_2026_paper.html.

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*, pp. 18558–18568, 2023a. URL https://openaccess.thecvf.com/content/ICCV2023/html/Chen_Text2Tex_Text-Driven_Texture_Synthesis_via_Diffusion_Models_ICCV_2023_paper.html.

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*, pp. 22246–22256, 2023b. URL https://openaccess.thecvf.com/content/ICCV2023/html/Chen_Fantasia3D_Disentangling_Geometry_and_Appearance_for_High-quality_Text-to-3D_Content_Creation_ICCV_2023_paper.html.

Wei Cheng, Juncheng Mu, Xianfang Zeng, Xin Chen, Anqi Pang, Chi Zhang, Zhibin Wang, Bin Fu, Gang Yu, Ziwei Liu, and Liang Pan. MVPaint: Synchronized multi-view diffusion for painting anything 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 585–594, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Cheng_MVPaint_Synchronized_Multi-View_Diffusion_for_Painting_Anything_3D_CVPR_2025_paper.html.

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.

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*, pp. 4620–4629, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Gao_GenesisTex_Adapting_Image_Denoising_Diffusion_to_Texture_Space_CVPR_2024_paper.html.

Dehao Hao, Kaiyi Zhang, Tanghui Jia, Xiangjun Gao, Dongyu Yan, Weikai Chen, Zeyu Hu, Lingting Zhu, Yingda Yin, Runze Zhang, et al. Msvs-vae: Multi-scale anchored vecset for high-fidelity 3d reconstruction. In *European Conference on Computer Vision*, pp. 332–349. Springer, 2026.

Huiang He, Shengchu Zhao, Jianwen Huang, Jie Li, Jiaqi Wu, Hu Zhang, Pei Tang, Heliang Zheng, Yukun Li, and Rongfei Jia. Hitem3D 2.0: Multi-view guided native 3d texture generation. *arXiv preprint arXiv:2604.09231*, 2026. URL https://arxiv.org/abs/2604.09231.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14822–14833, 2025a.

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. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26294–26305. IEEE, 2025b.

Zehuan Huang, Yuan-Chen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. MV-Adapter: Multi-view consistent image generation made easy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16377–16387, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Huang_MV-Adapter_Multi-View_Consistent_Image_Generation_Made_Easy_ICCV_2025_paper.html.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. LATTICE: Democratize high-fidelity 3d generation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19982–19992, 2026a.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Xin Yang, Xin Huang, Jingwei Huang, Xiangyu Yue, and Chunchao Guo. NaTex: Seamless texture generation as latent color diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18619–18629, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lai_NaTex_Seamless_Texture_Generation_as_Latent_Color_Diffusion_CVPR_2026_paper.html.

Dong-Yang Li, Wang Zhao, Yuxin Chen, Wenbo Hu, Meng-Hao Guo, Fang-Lue Zhang, Ying Shan, and Shi-Min Hu. Pixal3d: Pixel-aligned 3d generation from images. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026.

Peng Li, Yuan Liu, Xiaoxiao Long, Feihu Zhang, Cheng Lin, Mengfei Li, Xingqun Qi, Shanghang Zhang, Wenhan Luo, Ping Tan, Wenping Wang, Qifeng Liu, and Yike Guo. Era3D: High-resolution multiview diffusion using efficient row-wise attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/65a723bf7d8dad838c09178270d30e80-Abstract-Conference.html.

Yixun Liang, Kunming Luo, Xiao Chen, Rui Chen, Hongyu Yan, Weiyu Li, Jiarui Liu, Fei-Peng Tian, and Ping Tan. UniTEX: Universal high fidelity generative texturing for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19917–19927, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liang_UniTEX_Universal_High_Fidelity_Generative_Texturing_for_3D_Shapes_CVPR_2026_paper.html.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 300–309, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Lin_Magic3D_High-Resolution_Text-to-3D_Content_Creation_CVPR_2023_paper.html.

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 (CVPR)*, pp. 4284–4293, 2024a. doi: 10.1109/CVPR52733.2024.00410. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liu_TexOct_Generating_Textures_of_3D_Models_with_Octree-based_Diffusion_CVPR_2024_paper.html.

Jialun Liu, Jinbo Wu, Xiaobo Gao, Jiakui Hu, Bojun Xiong, Xing Liu, Chen Zhao, Hongbin Pei, Haocheng Feng, Yingying Li, Errui Ding, and Jingdong Wang. TexGarment: Consistent garment UV texture generation via efficient 3d structure-guided diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26566–26575, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liu_TexGarment_Consistent_Garment_UV_Texture_Generation_via_Efficient_3D_Structure-Guided_CVPR_2025_paper.html.

Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. In *ACM SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024b. doi: 10.1145/3680528.3687621. URL https://doi.org/10.1145/3680528.3687621.

Yuxin Liu, Minshan Xie, Jiawen Liang, Runsong Zhu, Chi-Wing Fu, and Tien-Tsin Wong. PLSR: Progressive and localized super-resolution of 3d objects via localized latent voxel diffusion. In *European Conference on Computer Vision (ECCV)*, 2026.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3D: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9970–9980, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Long_Wonder3D_Single_Image_to_3D_using_Cross-Domain_Diffusion_CVPR_2024_paper.html.

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*, volume 39, pp. 5820–5828, 2025. doi: 10.1609/aaai.v39i6.32621. URL https://ojs.aaai.org/index.php/AAAI/article/view/32621.

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 (ICCV)*, pp. 4531–4540, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Oechsle_Texture_Fields_Learning_Texture_Representations_in_Function_Space_ICCV_2019_paper.html.

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

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. XCube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4209–4219, 2024.

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*, 2023. doi: 10.1145/3588432.3591503. URL https://doi.org/10.1145/3588432.3591503.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: A single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. URL https://arxiv.org/abs/2310.15110.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International conference on learning representations*, volume 2024, pp. 39838–39859, 2024.

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*, pp. 72–88. Springer, 2022.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael 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, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. URL https://mlanthology.org/tmlr/2026/simeoni2026tmlr-dinov3/.

Peng Wang and Yichun Shi. ImageDream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023. URL https://arxiv.org/abs/2312.02201.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xiang_Structured_3D_Latents_for_Scalable_and_Versatile_3D_Generation_CVPR_2025_paper.html.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14419–14429, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xiang_Native_and_Compact_Structured_Latents_for_3D_Generation_CVPR_2026_paper.html.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 551–561, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xiong_TexGaussian_Generating_High-quality_PBR_Material_via_Octree-based_3D_Gaussian_Splatting_CVPR_2025_paper.html.

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*, pp. 4206–4216, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Yu_Texture_Generation_on_3D_Meshes_with_Point-UV_Diffusion_ICCV_2023_paper.html.

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*, 43(6), December 2024. doi: 10.1145/3687909. URL https://doi.org/10.1145/3687909.

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*, pp. 4252–4262, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Zeng_Paint3D_Paint_Anything_3D_with_Lighting-Less_Texture_Diffusion_Models_CVPR_2024_paper.html.

Yifei Zeng, Yajie Bao, Jiachen Qian, Shuang Wu, Youtian Lin, Hao Zhu, Buyu Li, Feihu Zhang, Xun Cao, and Yao Yao. TEXTRIX: Latent attribute grid for native texture generation and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27104–27113, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zeng_TEXTRIX_Latent_Attribute_Grid_for_Native_Texture_Generation_and_Beyond_CVPR_2026_paper.html.

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*, 42(4), 2023. doi: 10.1145/3592442.

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*, 43(4), 2024a. doi: 10.1145/3658146.

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.

Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, and Xiaogang Jin. DreamMat: High-quality pbr material generation with geometry- and light-aware diffusion models. *ACM Transactions on Graphics*, 43 (4), 2024b. doi: 10.1145/3658170. URL https://doi.org/10.1145/3658170.

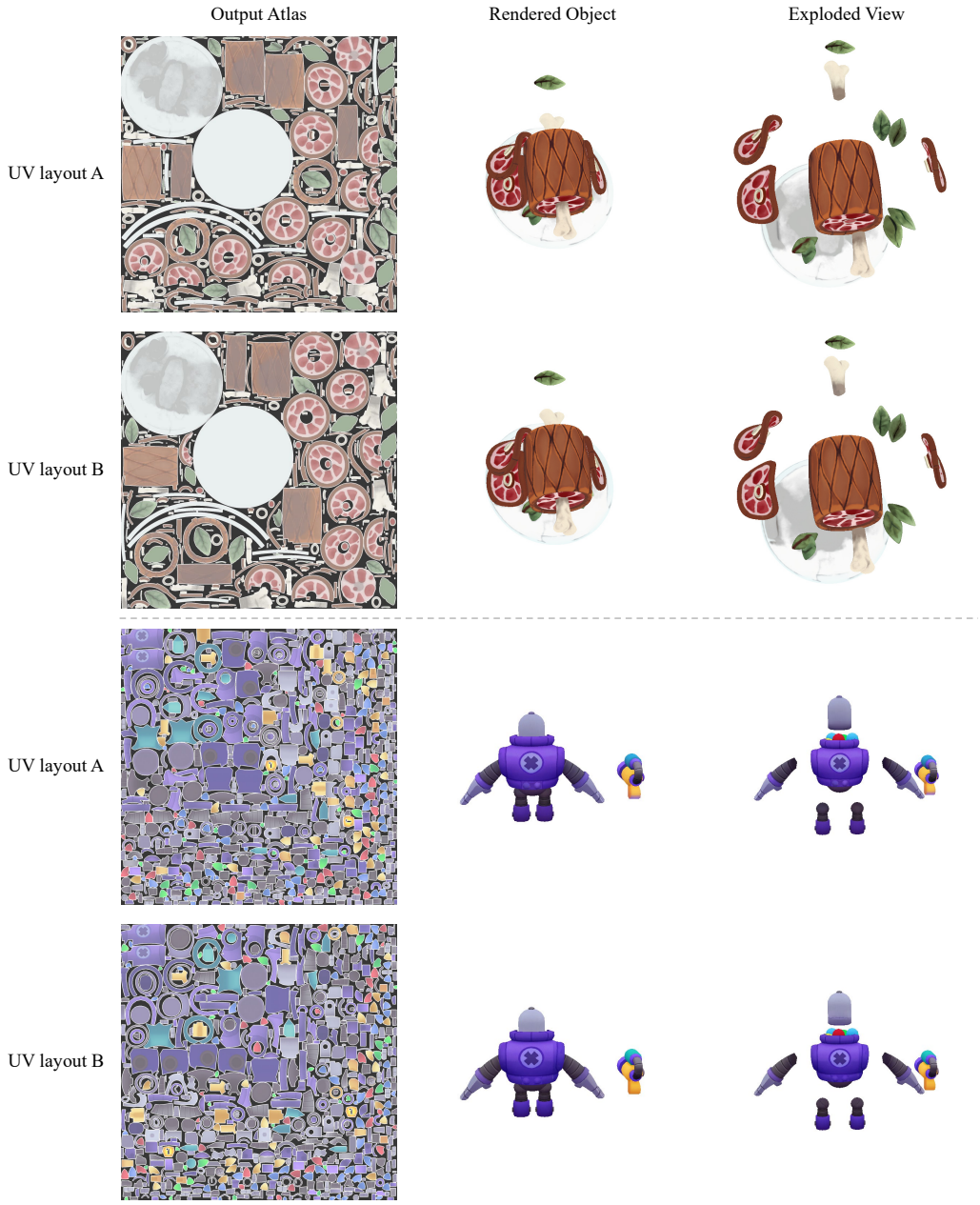

Figure 8: **Robustness to UV layout variation.** Different arrangements of UV islands lead to largely consistent generated appearance, as shown by both the rendered objects and exploded views, suggesting limited sensitivity to UV-space organization.

A UV PARAMETERIZATION ROBUSTNESS

Since CRAFT leverages surface information derived from the UV atlas, we also examine how variations in UV layout affect generation results. As shown in Fig. 9, for the same geometry, different arrangements of UV islands still lead to largely consistent generated appearances, both on the assembled object and in the exploded views of individual components. We further quantify this consistency in Table 4, reporting PSNR, MSE, and LPIPS between results produced under different UV layouts. These results suggest that CRAFT is relatively robust to changes in UV-space organization and does not rely strongly on a particular atlas layout.

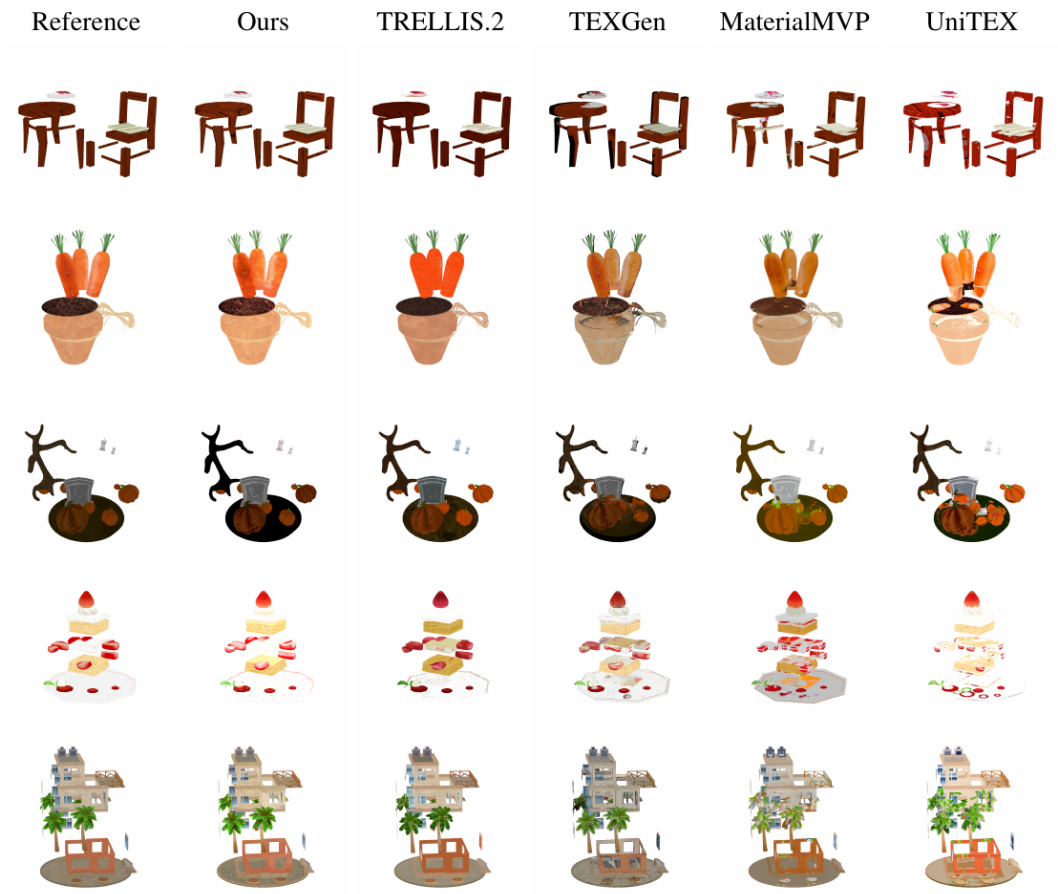

Figure 9: **More Comparison results.**

Table 4: **Quantitative Comparison for UV parameterization robustness**

Setting	PSNR↑	MSE↓	LPIPS↓
Normal	**36.54**	**0.000694**	**0.0088**
Explosion	30.44	0.001982	0.0200

B Visibility Cache Build

For a token τ, its visibility in a conditioning view is defined by

$$\text{proj_valid}_\tau = \mathbb{1}[\{(u,v) : ((u,v),\tau) \in \mathcal{C}\} \neq \emptyset] \,.$$

The visibility cache is constructed from first-hit rasterization at 1024×1024 resolution. The 64^3 grid is used only to quantize the interpolated 3D hit point and perform the lookup $(\text{surface_id}, \text{xyz}_{64}) \mapsto \tau$. Multiple image pixels may therefore contribute to the same token through a scatter-mean operation. Details are list in 1 and 2

C Training

The pretrained TRELLIS.2 material DiT initializes the original transformer layers. Each block adds a zero-initialized affine projection for the routed local reference feature, whose output is masked after projection and added to the global cross-attention output before the residual update. The material VAE, shape VAE, and DINOv3 encoder remain frozen; all DiT parameters and new projections are optimized.

Algorithm 1 Construction of the First-Hit Visibility Cache

Require: Atlas mesh (V, F), face-part labels π_f, packed token table $\mathcal{T}$, conditioning camera C, image resolution $R = 1024$, voxel resolution $K = 64$
Ensure: Visibility cache $\mathcal{C}$

$\mathcal{C} \leftarrow \emptyset$
$\mathbf{G} \leftarrow \mathrm{RasterizeZBuffer}(V, F, C, R)$ ▷ $\mathbf{G}_{uv,3} > 0$ iff pixel (u, v) hits a triangle
$\mathbf{B} \leftarrow \mathrm{RasterizeBarycentrics}(V, F, C, R)$
for all pixels (u, v) **do**
  if $\mathbf{G}_{uv,3} \leq 0$ **then**
    continue ▷ no first-hit triangle
  end if
  $f \leftarrow \mathbf{G}_{uv,1}$
  $\boldsymbol{\beta} \leftarrow \mathbf{B}_{uv}$ ▷ barycentric coordinates on face f
  $\mathbf{p} \leftarrow \sum_{j=1}^{3} \beta_j V_{f,j}$ ▷ 3D first-hit point
  $\mathbf{x} \leftarrow \mathrm{floor}\big((\mathrm{NormalizeAABB}(\mathbf{p}) + \frac{1}{2})K\big)$ ▷ $\mathbf{x} \in \{0, \ldots, K-1\}^3$
  $s \leftarrow \pi_f$ ▷ surface/island identifier of face f
  $\tau \leftarrow \mathcal{T}.\mathrm{lookup}(s, \mathbf{x})$
  if $\tau \neq \varnothing$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{((u, v), \tau)\}$ ▷ store pixel–token correspondence
  end if
end for
return $\mathcal{C}$

We sample timesteps from $\mathrm{LogitNormal}(1, 1)$, perturb clean surface-addressed material latents using the backbone's flow path, and minimize mean-squared error against the target velocity over valid surface tokens. Geometry latents, routed local features, and global image tokens provide the conditions. Training runs for 160K iterations with AdamW, learning rate 2×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay 0.01. We use eight 80 GB GPUs with batch size one per GPU and bfloat16 mixed precision.

For classifier-free guidance, all image-derived conditions are dropped with probability 0.1: global image tokens, routed reference projections, and visibility masks are zeroed while geometry conditioning is retained. Separately, local-view dropout retains all 16 views with probability 0.5; otherwise, the number retained is sampled uniformly from $\{1, \ldots, 15\}$ and those views are selected without replacement.

Global image conditioning uses a fixed subset of four views to control memory. If a view in this subset is dropped from local conditioning, its spatial patch features are removed but its CLS and register tokens remain as global context. The subset is fixed throughout training and is not chosen using a view-specific criterion. View dropout is disabled during validation.

For the reported single-view evaluation, the same supplied reference view provides both the routed local features and the global image tokens. No additional view is used for either condition at inference.

D INFERENCE

At inference, the input mesh provides UV-chart surface indices and sparse latent support. Calibrated reference views provide local features, global image features, and first-hit visibility caches. We initialize material latents with independent standard Gaussian noise in normalized training space using a fixed seed. Shape latents are normalized with training statistics and concatenated as geometry conditions without added noise. We integrate the learned velocity field from $t = 1$ to $t = 0$ with a 25-step Euler solver and classifier-free guidance scale 3.0. Guidance is applied over the full trajectory without guidance rescaling. Generated material latents are denormalized with the training mean and standard deviation.

Algorithm 2 First-Hit Feature Gathering and Token Aggregation

Require: Conditioning image I, visibility cache $\mathcal{C}$, packed token set $\mathcal{T}$, pixel encoders E_{DINO} and E_{NAF}
Ensure: Token features $\{\mathbf{h}_\tau\}_{\tau\in\mathcal{T}}$, token visibility indicators $\{\mathrm{proj_valid}_\tau\}$

$\mathbf{D} \leftarrow E_{\mathrm{DINO}}(I)$
$\mathbf{N} \leftarrow E_{\mathrm{NAF}}(I)$ ▷ features are evaluated on the 1024^2 image grid
for all $\tau \in \mathcal{T}$ **do**
  $\mathcal{P}_\tau \leftarrow \{(u,v) \mid ((u,v),\tau) \in \mathcal{C}\}$
  if $|\mathcal{P}_\tau| = 0$ **then**
    $\mathrm{proj_valid}_\tau \leftarrow 0$
    $\mathbf{h}_\tau \leftarrow \mathbf{0}$
  else
    $\mathrm{proj_valid}_\tau \leftarrow 1$
    $\mathbf{d}_\tau \leftarrow \mathrm{Mean}_{(u,v)\in\mathcal{P}_\tau}\, \mathbf{D}_{uv}$
    $\mathbf{n}_\tau \leftarrow \mathrm{Mean}_{(u,v)\in\mathcal{P}_\tau}\, \mathbf{N}_{uv}$
    $\mathbf{h}_\tau \leftarrow \mathrm{Concat}(\mathbf{d}_\tau, \mathbf{n}_\tau)$
  end if
end for
return $\{\mathbf{h}_\tau\}_{\tau\in\mathcal{T}}, \{\mathrm{proj_valid}_\tau\}_{\tau\in\mathcal{T}}$

The frozen material VAE decodes the packed groups independently. Each decoded sample is assigned to its nearest texel in a 2048×2048 atlas while respecting its source UV-chart identity. Assignments to a different chart are discarded; multiple samples assigned to one texel are averaged. Chart-constrained neighbor propagation fills uncovered texels, followed by 16-pixel dilation into empty atlas regions. The decoder produces base color, metallic, roughness, and opacity. The reported experiments bake and evaluate base color; the remaining material channels are not evaluated.

E Evaluation

Recent native 3D texturing methods (Lai et al., 2026b; Zeng et al., 2026; Chen et al., 2026) are omitted because reproducible public code or pretrained weights were unavailable at submission time.

E.1 Evaluation of occluded region

To evaluate texture on the occluded surfaces, we split each mesh into the source sub-parts and translate every part outward from the object center, along the ray from the center to the part centroid, by 0.45 times the bounding-box radius. Ground truth and all methods share this decomposition and displacement and are rendered with the same camera, so the comparison isolates appearance on the newly exposed surfaces.

F Data Preparation

We curate the training set from TexVerse (Zhang et al., 2025) using its asset captions and appearance metadata. We first filter captions by keywords to exclude scanned objects and Gaussian-generated assets. We then remove uniformly white or black assets, assets without textures, and objects without submesh structure. For the test set from Objaverse-XL (Deitke et al., 2023), we retain assets with aesthetic scores above 6.5 and between 4 and 15 submeshes, inclusive, then randomly sample 105 objects.

G Data Preparation

We curate the training set from TexVerse (Zhang et al., 2025) using its asset captions and appearance metadata. We first filter captions by keywords to exclude scanned objects and Gaussian-generated assets. We then remove uniformly white or black assets, assets without textures, and objects without

submesh structure. For the test set from Objaverse-XL (Deitke et al., 2023), we retain assets with aesthetic scores above 6.5 and between 4 and 15 submeshes, inclusive, then randomly sample 105 objects.