LUCE: RELIGHTABLE GAUSSIANS FOR 3D ASSET GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: Our method **Luce** generates relightable 3D assets from a single image. Each asset consists of Gaussians encoding three modalities of a physically based rendering (PBR) material: albedo, metallic-roughness, and surface normals. These modalities enable standard PBR shading, rendering the Gaussians under novel illumination. From left to right, the columns show: the input condition image; the generated PBR modalities stacked vertically—albedo (top), metallic-roughness (middle, metallic in Red, roughness in Green channel), and surface normals (bottom); and the generated 3D asset rendered under three environment maps, previewed as small strips above each render (top row).

ABSTRACT

High-fidelity image-to-3D generation requires a 3D representation that captures both geometry and appearance. However, preserving fine detail across the physically based rendering (PBR) modalities needed for relighting remains challenging. To address this, we propose Luce, a 3D representation that unifies geometry and PBR materials within a voxelized multimodal Gaussian cloud, using dedicated Gaussian primitives for albedo, metallic-roughness, and surface normals. A variational autoencoder compresses this representation into a unified material-aware latent space. A rectified-flow transformer generates this latent from a single image using multi-layer features from a pretrained image encoder that preserve both semantic context and fine spatial detail. The latent is then decoded into relightable PBR Gaussians and an optional textured mesh with a tangent-space normal map. On Toys4K, Luce achieves state-of-the-art single-image-to-3D generation, improving FID by 28% over the strongest baseline. We further evaluate Luce on a benchmark of AI-generated images depicting diverse subjects and materials, where it improves the CLIP image-alignment score over the best baseline (0.8519 vs. 0.8299). Luce generates relightable, geometrically accurate, and materially faithful assets that preserve fine details such as text, logos, and inscriptions.

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.



Figure 2: **Legible text on generated 3D assets.** Luce compared with LiTo (Chang et al., 2026), TRELLIS 2 (Xiang et al., 2026), and TRELLIS (Xiang et al., 2025) on single-image-to-3D generation; each row shows the input condition image and one rendered view of each method's generated 3D asset. LiTo generates assets aligned to the input view, whereas other methods generate in a canonical orientation. Luce keeps surface text and markings legible where baselines distort them.

1 Introduction

Diffusion and flow-matching models have largely closed the realism gap in 2D image generation (Rombach et al., 2022; Black Forest Labs, 2024), where images share one representation: a dense pixel grid. 3D has many, spanning implicit fields and explicit primitives (Park et al., 2019; Mildenhall et al., 2020; Shen et al., 2023; Kerbl et al., 2023), and that choice sets a ceiling on what a generative model can produce. To render and relight assets in standard pipelines, the representation must support fine geometric detail and physically based materials together.

We introduce Luce (Fig. 1), a representation that meets these requirements: it unifies fine geometry and physically based materials in a single relightable form. In Luce, an asset is a collection of Gaussians attached to a sparse set of voxels that intersect the object's surface. Within each voxel, we encode a set of Gaussians for each PBR modality: albedo, metallic-roughness, and surface normal direction. This representation is a complete PBR material description in a 3D-native format that drops directly into standard rendering pipelines. It also compresses into a compact, diffusible latent. The per-modality encoding of geometry and appearance helps to preserve high-frequency details that prior approaches often smooth out, including legible text on generated 3D surfaces, a setting where existing image-to-3D methods typically struggle. This observation is supported by our quantitative studies, where Luce achieves the lowest FID on Toys4K (20.99), improving on the strongest baselines, TRELLIS 2 (Xiang et al., 2026) (29.22) and LiTo (29.76), by more than 8 FID.

Our contributions are:

(i) A multimodal PBR Gaussian representation (Sec. 3.1). Our representation jointly models geometry and appearance, parameterizing albedo, metallic-roughness, and surface normals directly on Gaussian primitives. It supports relighting under image-based illumination via deferred shading in a standard PBR reflectance model.

(ii) A unified latent for joint geometry and material generation (Sec. 3.2). We learn a compact latent over the PBR Gaussian representation that encodes geometry and all material modalities together, so a rectified-flow transformer generates a complete relightable asset from a single image. The latent decodes directly into relightable PBR Gaussians for rendering; optionally, we extract a mesh and bake the decoded Gaussians into its PBR texture maps, yielding a textured mesh.

(iii) Multi-layer image conditioning (Sec. 3.3). We propose combining DINOv2 (Oquab et al., 2024) features from shallow and deep encoder layers, preserving fine spatial detail (e.g., logos, labels, and inscriptions on the asset) critical for high-fidelity 3D generation (see Fig. 2 and Fig. 4).

(iv) Tangent-space normal map transfer (Sec. 3.4). Luce's normal Gaussians learn fine surface detail (e.g., engravings, fabric weave, embossed text) from the asset's authored normal maps, beyond what mesh geometry encodes. At inference, we bake these decoded normals onto the extracted mesh as tangent-space normal maps, adding high-frequency detail at zero polygon-count cost.

2 Related Work

3D representations for generation. A generative model's representation space largely determines what it can faithfully capture. Implicit level sets (Mescheder et al., 2019) admit arbitrary genus without a template, but only as closed, watertight surfaces. More recent implicit approaches scale to higher fidelity but exclusively target geometry: TripoSG (Li et al., 2025b) via SDFs and Direct3D-S2 (Wu et al., 2025) via sparse-voxel fields; both inherit the same watertight requirement. 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) renders appearance explicitly and cheaply, and a series of feed-forward and reconstruction techniques (Tang et al., 2024b;a; Zhang et al., 2024b; Szymanowicz et al., 2024; Zhang et al., 2024a) extends it to image- or text-to-3D. These approaches do not model materials; like NeRF (Mildenhall et al., 2020), they bake the lighting environment into the representation. Structured-latent methods take a different route, combining sparse voxels with learned per-voxel features: TRELLIS (Xiang et al., 2025) stores DINOv2 features (Oquab et al., 2024) and decodes to multiple formats, TRELLIS 2 (Xiang et al., 2026) replaces these with native O-Voxel geometry and per-voxel PBR attributes, and LiTo (Chang et al., 2026) tokenizes surface light fields via Perceiver IO (Jaegle et al., 2022). Luce pairs the rendering efficiency of 3DGS with per-modality PBR decomposition in a structured latent, without mesh dependence or baked lighting.

3D generative models. Latent-space approaches have become the dominant paradigm for 3D generation. Early systems such as Shap-E (Jun & Nichol, 2023) and 3DTopia-XL (Chen et al., 2025) generate implicit-function parameters or primitive-based latents; recent ones build on Diffusion Transformers (DiT) (Peebles & Xie, 2023) and flow matching (Lipman et al., 2023; Liu et al., 2023). Among DiT-based systems, TRELLIS (Xiang et al., 2025) and TRELLIS 2 (Xiang et al., 2026) use a structure-then-latent paradigm, LiTo (Chang et al., 2026) conditions a single DiT on DINOv2 (Oquab et al., 2024), and Step1X-3D (Li et al., 2025a) and Hunyuan3D 2.1 (Tencent Hunyuan3D Team, 2025) pair a DiT shape generator with a separate multi-view texture-diffusion stage. Luce also uses rectified-flow transformers, but operates in a unified PBR Gaussian latent so geometry and materials are generated jointly.

PBR and relightable 3D. Relighting requires decomposing appearance into material properties. Per-scene inverse rendering does this for a single asset (Munkberg et al., 2022; Zhang et al., 2021; Liang et al., 2024; Jiang et al., 2024; Gao et al., 2024), but does not generalize to novel objects. The GS-based ones among them tie per-Gaussian normals to the reconstructed surface, via a flattened Gaussian's shortest axis or depth-derived pseudo-normals. Luce instead predicts normals as a separate modality, so they carry detail finer than the geometry. Recent methods attach PBR attributes to Gaussians in feed-forward or generative settings, each with a constraint: TexGaussian (Xiong et al., 2025) requires an input mesh; RelitLRM (Zhang et al., 2025a) models appearance with spherical harmonics rather than explicit PBR; MGM (Ye et al., 2025) is text-conditioned and runs in multiple stages; MatSpray (Langsteiner et al., 2026) optimizes per-scene from multiple views. Among generative models, TRELLIS 2 (Xiang et al., 2026) produces PBR-complete assets via a mesh-centric representation, and LiTo (Chang et al., 2026) captures view-dependent effects through spherical harmonics that cannot be decomposed into materials. A separate line of work estimates normals from images (Bae & Davison, 2024; Ye et al., 2024); we render decoded normal Gaussians and bake them directly as tangent-space maps (Sec. 3.4). Luce learns a compact diffusible latent from large-scale PBR data that jointly generates all material modalities and decodes to PBR Gaussians and textured meshes.

3 Method

As shown in Fig. 3, Luce represents assets as multimodal PBR Gaussian clouds, compresses this representation into a compact latent with a structured latent VAE (SLatVAE), and trains an image-conditioned rectified-flow model, the structured latent flow (SLatFlow), to generate the latent.



Figure 3: **Overview of Luce. (Top) Representation and SLatVAE.** Given a 3D PBR asset, here a Toys4K (Stojanov et al., 2021) sample, we render multiview images and fit per-modality Gaussian splats (albedo, metallic-roughness, normals) on a sparse voxel grid. A structured latent VAE (SLatVAE) encodes this representation into a compact, diffusible latent and decodes it back to PBR Gaussians. **(Bottom) Generation pipeline.** Given a single input image, a sparse-structure flow first predicts the sparse voxel layout. Conditioned on multi-layer DINOv2 features, SLatFlow then generates a PBR Gaussian latent at each active voxel. The structured latent decodes into relightable PBR Gaussians and also serves as input to the mesh decoder, which yields textured meshes with tangent-space normal maps and a complete PBR material set.

3.1 Multimodal PBR Gaussian Representation

The standard PBR pipeline models surface appearance as a function of independent modalities: diffuse reflectance (albedo), material properties (metallic, roughness), and surface orientation (normals). We mirror this decomposition in 3D by giving each occupied voxel three dedicated Gaussian sets: albedo $\mathbf{G}^{\text{alb}}$ (diffuse surface color), metallic-roughness $\mathbf{G}^{\text{mr}}$ (combined metallic and roughness), and normal $\mathbf{G}^{\text{nor}}$ (per-point surface orientation). We denote this set of modalities as $\mathcal{M} = \{\text{alb}, \text{mr}, \text{nor}\}$. Rather than extending a single set of Gaussians with PBR attributes, we keep the three Gaussian sets geometrically independent: each has its own positions, scales, rotations, and opacities within the shared voxel structure. This keeps the pipeline simple, since each modality is fit and splatted with standard 3DGS against its own intrinsic images, and it lets each modality place its Gaussians where its own detail is richest. Because the normal modality is stored explicitly rather than derived from the surface shape, its normals can deviate from the underlying surface geometry, a property we leverage in Sec. 3.4 for tangent-space normal map transfer.

Rendering PBR Gaussians. Each modality is splatted independently via standard 3DGS alpha-compositing (Kerbl et al., 2023), producing per-pixel albedo $\boldsymbol{a}$, metallic κ, roughness α, surface normal $\boldsymbol{n}$, and opacity o. Normals are renormalized to unit length after compositing. We shade these splatted PBR buffers using split-sum image-based lighting (Karis, 2013) with the Cook–Torrance microfacet BRDF (Cook & Torrance, 1982), with no mesh extraction or UV unwrapping required; see Appendix A for details of this deferred shading.

Each Gaussian uses the standard 3DGS parameterization (position offset $\boldsymbol{p}$ from the voxel center, anisotropic scale $\boldsymbol{s}$, rotation quaternion $\boldsymbol{q}$, opacity o) together with a single view-independent modality value $\boldsymbol{c}$ (RGB for albedo, metallic and roughness for metallic-roughness, normal direction for normals). Unlike the original 3DGS, we store no spherical harmonics; view-dependent shading is produced analytically through PBR compositing (Appendix A). With i indexing voxels and j

indexing the K Gaussians per modality, the complete per-voxel representation at voxel i is:

$$\mathbf{F}_i = \{\mathbf{G}_i^m\}_{m\in\mathcal{M}}, \quad \mathbf{G}_i^m = \{(\boldsymbol{p}_{ij}^m,\, \boldsymbol{s}_{ij}^m,\, \boldsymbol{q}_{ij}^m,\, o_{ij}^m,\, \boldsymbol{c}_{ij}^m)\}_{j=1}^{K}, \tag{1}$$

where K is the number of Gaussians per modality per voxel, d is the voxel-grid resolution, and $N_v^{d^3}$ is the number of active voxels in the d^3 grid. The per-voxel bundles form $\mathcal{V} = \{\mathbf{F}_i\}_{i=1}^{N_v^{d^3}}$, the asset's complete voxelized multimodal Gaussian representation. We store $\boldsymbol{s}$ in log-space, $\boldsymbol{q}$ as a unit quaternion, and pass o through a sigmoid. We build each asset's $\mathcal{V}$ in a preprocessing step, fitting the K Gaussians of each modality independently against multi-view PBR renders of the asset; these fitted representations serve as input to the SLatVAE.

3.2 Variational Autoencoder

The raw multimodal Gaussian cloud $\mathcal{V}$ is high-dimensional and sparsely structured, so modeling it directly with a generative model is computationally expensive. We compress it into a compact per-voxel latent with the SLatVAE (architecture in Appendix B and Fig. 9). The key design choice is to *trade spatial resolution for per-voxel density*: the encoder downsamples the voxel grid, and the decoder compensates by predicting more Gaussians per voxel. This keeps the latent small enough to diffuse cheaply while still reconstructing fine material and geometric detail.

Encoder. $\mathcal{E}$ takes the concatenation of all three PBR Gaussian sets per voxel (the standard GS parameters across $|\mathcal{M}| = 3$ modalities, with K Gaussians per modality) and is implemented as a sparse transformer with shifted-window attention (Liu et al., 2021) that downsamples the input grid from resolution d to a coarser $d' < d$, producing a compact latent $\mathbf{z} = \mathcal{E}(\mathcal{V}) \in \mathbb{R}^{N_v^{d'^3} \times C}$, where $N_v^{d'^3}$ is the number of active voxels at the resolution d' and C is the latent channel dimension.

Decoder. $\mathcal{D}$ runs at the coarser latent resolution for efficiency and offsets the lost resolution by predicting more Gaussians per voxel per modality, recovering fine surface detail. We train the SLatVAE end-to-end with a rendering-based reconstruction loss, as in TRELLIS (Xiang et al., 2025): the decoded Gaussians are differentiably rendered per modality against the ground-truth intrinsic images, with a KL prior on the latent and light regularizers on Gaussian scale and opacity. The full objective and coefficients are in Appendix B.

3.3 Flow-Based Generation

We use rectified-flow transformers to generate in the SLatVAE's latent space, conditioned on a single image (bottom panel of Fig. 3). Generation proceeds in two stages: we first generate the object's sparse voxel structure (which voxels it intersects), then the latent features at each occupied voxel. We adopt this structure-then-latent approach from TRELLIS (Xiang et al., 2025). We use the pretrained sparse-structure VAE and sparse-structure flow from TRELLIS 2 (Xiang et al., 2026) without modification. Given a condition image, these models output the $N_v^{d'^3}$ active voxels that serve as the scaffold for the next stage. SLatFlow then generates a PBR Gaussian latent at each active voxel, producing geometry and materials jointly. It is a DiT-style transformer (Peebles & Xie, 2023) adapted for sparse 3D latents: each sample's active voxels form a variable-length token sequence. The diffusion timestep modulates each block via adaptive layer normalization; image features are injected through cross-attention. Architectural details are in Appendix B.

Multi-layer image conditioning. Dense prediction tasks have benefited from fusing features across multiple encoder layers (Long et al., 2015; Lin et al., 2017; Zhao et al., 2017; Chen et al., 2017; Ranftl et al., 2021; Cheng et al., 2022). Recent work shows the same effect for frozen ViTs: multi-layer DINOv2 features outperform single-layer ones because early layers carry fine spatial patterns while deep layers carry semantics (Karypidis et al., 2025). We apply this to 3D generation: we concatenate features from several DINOv2 layers and project them into the conditioning space with a two-layer MLP (Appendix B). Early-layer features carry fine spatial structure (text, logos, inscriptions); fusing them with deep-layer semantics carries this detail through SLatFlow to the rendered 3D surface (Fig. 4). Multi-layer conditioning outperforms single-layer on both Toys4K FID (20.99 vs. 25.21) and CLIP on our AI-generated-image benchmark (0.8519 vs. 0.8081; Appendix C).



Figure 4: **Effect of multi-layer DINOv2 conditioning.** We compare Luce trained with multi-layer DINOv2 features (layers 6, 12, 18, 24) against single-layer DINOv2 features (layer 24 only). Multi-layer conditioning preserves fine spatial detail from the condition image, including legible text and logos on the generated 3D surface. For each variant, the larger shaded render is shown next to a cascade of the per-modality decomposition (albedo, metallic-roughness, surface normals).

Dual decoding. The primary generation output of Luce is a multimodal PBR Gaussian cloud produced by the Gaussian decoder, a complete material description that splats directly under any environment map via deferred PBR shading (Appendix A), with no mesh extraction or UV unwrapping required. To enable fair comparison with baselines whose primary output is a mesh (Xiang et al., 2025; 2026; Chen et al., 2025), we decode the same latent into a textured mesh using a FlexiCubes (Shen et al., 2023) mesh decoder, following TRELLIS (Xiang et al., 2025). We keep its architecture unchanged and retrain it on our learned latent space. Baking decoded normal Gaussians as tangent-space normal maps onto the extracted mesh (Sec. 3.4) improves mesh normal PSNR from 29.5 to 33.0 dB (Sec. 4.3), recovering high-frequency surface detail at zero polygon-count cost.

3.4 Tangent-Space Normal Map Transfer

Mesh extraction from a voxel grid bounds geometric frequency by the grid resolution, leaving sub-voxel surface features (engravings, fabric weaves, embossed text) unrepresented in the extracted geometry alone. A standard remedy in real-time rendering is to include a normal map alongside the mesh: this perturbs the geometric normal for lighting calculations, restoring apparent surface detail without raising polygon count. Our representation lends itself naturally to this. Because our PBR Gaussians are decoupled from the mesh, we supervise them against the *effective* normals used for rendering (i.e., the perturbed geometric normals after applying the authored normal map), so they learn surface orientation at a frequency finer than the mesh can represent. At inference, we express the decoded normal Gaussians in the mesh's local tangent frame to obtain a tangent-space normal map on the UV-parameterized surface, preserving high-frequency details at a resolution far beyond the mesh geometry. The final mesh carries four texture maps: diffuse (albedo), metallic, roughness, and tangent-space normal, giving a complete PBR material description directly importable into production renderers. We detail the baking pipeline in Appendix B.

4 Experiments

Datasets and evaluation benchmarks. Following prior generative 3D works (Xiang et al., 2026) for fair comparison, our training set combines ∼500K PBR-filtered assets from Objaverse (Deitke et al., 2023b) and Objaverse-XL (Deitke et al., 2023a) with a 158K-asset PBR subset of TexVerse (Zhang et al., 2025b). To evaluate, we use **Toys4K** (Stojanov et al., 2021) for reconstruction, which provides ground-truth 3D assets (a 338-asset PBR subset with all three PBR textures); for generation, we test on **412 Toys4K assets** and, to gauge generalization beyond Toys4K's simple toy domain, **130 AI-generated images** from Gemini 3.1 Flash Image (Google DeepMind, 2026), prompted to be diverse and detail-rich (text, logos, mixed materials).

Baselines. We compare against TRELLIS (Xiang et al., 2025), TRELLIS 2 (Xiang et al., 2026), LiTo (Chang et al., 2026), and 3DTopia-XL (Chen et al., 2025) (parameters in Table 1). All baselines use their public-release image conditioning model and input resolution. TRELLIS and LiTo use DINOv2 ViT-L/14 (Oquab et al., 2024), and 3DTopia-XL DINOv2 ViT-B/14, all at 518×518; TRELLIS 2 uses DINOv3 (Siméoni et al., 2026) ViT-L/16 at 512×512 for its sparse-structure stage and 1024×1024 for its latent stage. Luce (ours) uses DINOv2 ViT-L/14 at 1036×1036 for its



Figure 5: **Image-to-3D generation comparison.** Luce preserves legible text and fine surface details on generated 3D assets. For each method, the larger shaded render is shown next to a cascade of the per-modality decomposition (albedo, metallic-roughness, surface normals) when available.

Table 1: **Image-to-3D generation.** We evaluate on two benchmarks: Toys4K ($N = 412$, left) and our 130 AI-generated images ($N = 130$, right). Toys4K reports FID/KID with Inception and DINO (Oquab et al., 2024) backbones (KID and KID_{dino} $\times 100$); both benchmarks report CLIP (Radford et al., 2021), SigLIP2 (Tschannen et al., 2025), ULIP (Xue et al., 2024), and Uni3D-L (Zhou et al., 2024). Time(s) is the mean warm wall-clock time per generation on one H100 (Appendix D). Luce GS renders the decoded PBR Gaussians directly via deferred shading; Luce mesh rows render a textured mesh extracted from the same latent, with and without tangent-space normal map transfer. Best in **bold**, second-best underlined; shaded rows are ours. "—" marks metrics that do not apply to that render path; ULIP and Uni3D-L are mesh-only.

Method	Params (B)	Time (s)	Toys4K ($N = 412$)								AI-generated images ($N = 130$)			
			FID ↓	KID ↓	FID_{dino} ↓	KID_{dino} ↓	CLIP ↑	SigLIP2 ↑	ULIP ↑	Uni3D-L ↑	CLIP ↑	SigLIP2 ↑	ULIP ↑	Uni3D-L ↑
TRELLIS GS (Xiang et al., 2025)	1.70	4.78	30.75	0.212	0.109	0.0013	0.8898	0.9164	—	—	0.8299	0.8339	—	—
TRELLIS mesh (Xiang et al., 2025)	1.80	19.67	32.39	0.263	0.145	0.0039	0.8829	0.9092	0.1672	0.3747	0.8000	0.7958	0.1247	0.3278
3DTopia-XL (Chen et al., 2025)	1.02	39.61	83.23	2.726	0.542	0.0306	0.7644	0.7810	0.1418	0.2519	0.6481	0.6458	0.1108	0.2197
TRELLIS 2 (Xiang et al., 2026)	7.48	199.55	29.22	0.165	0.129	0.0022	0.8895	0.9161	0.1634	0.3635	0.8110	0.8166	0.1207	0.3280
LiTo (Chang et al., 2026)	1.84	3.75	29.76	0.208	0.128	0.0025	0.8909	0.9082	—	—	0.8234	0.8240	—	—
Luce GS	4.48	7.23	**20.99**	**0.033**	0.100	0.0028	0.9062	0.9230	—	—	**0.8519**	**0.8508**	—	—
Luce mesh without tangent normal	4.58	67.14	25.04	0.102	0.112	0.0019	0.8990	0.9234	**0.1681**	0.3767	0.8505	0.8466	**0.1254**	**0.3298**
Luce mesh with tangent normal	4.58	75.69	21.10	0.042	**0.088**	**0.0006**	**0.9072**	**0.9288**	0.1679	**0.3771**	0.8508	0.8465	0.1251	0.3295

SLatFlow (Table 3); its sparse-structure stage is inherited unchanged from TRELLIS 2 (Sec. 3.3). Baselines with explicit PBR outputs (TRELLIS 2, 3DTopia-XL) are evaluated on shaded color and all three modalities; those without (TRELLIS, LiTo), on shaded color and, where available, surface normals.

Metrics. *Generation:* On Toys4K, we report FID and KID on rendered views with both Inception and DINO (Oquab et al., 2024) backbones (FID_{dino}, KID_{dino}) as distribution-level metrics. Both benchmarks report alignment metrics: CLIP score, SigLIP2 (Tschannen et al., 2025), ULIP (Xue et al., 2024), and Uni3D-L, the Large variant of Uni3D (Zhou et al., 2024), measuring input-output agreement. *Reconstruction:* We provide per-modality PSNR, SSIM, and LPIPS for albedo, metallic-roughness, normal maps, and shaded renders, measuring surface orientation and material reconstruction. We detail the evaluation rendering and per-benchmark illumination in Appendix E.

4.1 Implementation Details

Model architecture, training hyperparameters, and the texture-baking pipeline are in Appendix B, and inference details in Appendix D; we summarize preprocessing here.

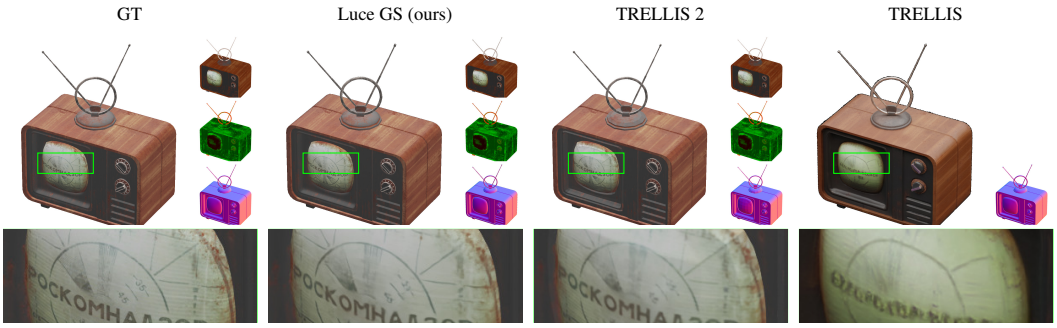

Figure 6: **Reconstruction quality on Toys4K.** Columns are ground truth and methods; each block shows the modalities that each method reconstructs (albedo, metallic-roughness, normals). Luce preserves fine detail (text, highlights, texture); the green box marks the region magnified below.

Table 2: **Reconstruction quality** on a PBR subset of Toys4K ($N = 338$). We report per-modality PSNR, SSIM, and LPIPS for color rendering (combined appearance under fixed illumination), albedo (diffuse color), metallic-roughness, and normal maps (surface orientation). For Luce, the **GS** row renders the decoded PBR Gaussians directly via deferred shading (no mesh); the **mesh** rows render a textured mesh from the same latent, with and without baked tangent-space normals. Best in **bold**, second-best underlined; shaded rows are ours. "—" indicates that the method does not produce that modality. Params is the autoencoder parameter count in billions.

Method	Color			Albedo			Metallic-Roughness			Normal			Params (B)
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
TRELLIS GS (Xiang et al., 2025)	27.0	0.912	0.124	—	—	—	—	—	—	—	—	—	0.17
TRELLIS mesh (Xiang et al., 2025)	26.3	0.907	0.118	—	—	—	—	—	—	26.9	0.914	0.113	0.26
3DTopia-XL (Chen et al., 2025)	23.9	0.877	0.188	25.4	0.893	0.191	22.6	0.910	0.191	23.3	0.876	0.165	0.03
TRELLIS 2 (Xiang et al., 2026)	34.5	0.961	0.074	**40.7**	**0.976**	**0.051**	**42.7**	**0.986**	**0.038**	32.1	0.940	0.093	1.66
LiTo (Chang et al., 2026)	31.9	0.944	0.110	—	—	—	—	—	—	27.6	0.912	0.116	0.29
Luce GS	**36.1**	**0.967**	**0.059**	38.6	0.971	0.055	39.1	0.978	0.048	**34.6**	**0.961**	**0.038**	1.82
Luce mesh without tangent normal	31.1	0.944	0.091	33.7	0.961	0.067	37.9	0.977	0.054	29.5	0.928	0.103	1.91
Luce mesh with tangent normal	32.5	0.953	0.080	33.7	0.961	0.067	37.9	0.977	0.054	33.0	0.952	0.067	1.91

Preprocessing. Each training asset is centered and scale-normalized to a unit bounding box, yielding a consistent canonical frame. We then render each from 150 cameras uniformly distributed on a sphere, producing per-view albedo, metallic-roughness, and normal images at a resolution of 1024×1024 pixels. The normal pass is rendered with the asset's authored normal maps applied, so its normals carry surface detail finer than the base geometry. Per-modality 3DGS is fit independently on each set and voxelized onto a 128^3 sparse grid with $K = 8$ Gaussians per voxel per modality. For each modality, we optimize all Gaussians via differentiable rendering against the intrinsic images, parameterizing each center as a $\tanh$-bounded offset from its parent voxel center, initialized at evenly spread quasi-random positions inside the voxel.

4.2 Generation

Table 1 reports generation quality on Toys4K (left) and our AI-generated images (right), with qualitative comparisons against baselines in Fig. 5 (per-sample baseline mosaics in Appendix G) and additional Luce examples in Appendix F. On Toys4K, Luce GS reaches the lowest FID (20.99), improving on the strongest baseline TRELLIS 2 (29.22) by more than 8 FID. On our AI-generated-image benchmark, Luce GS leads CLIP (0.8519) and SigLIP2 (0.8508) over the best baseline TRELLIS GS (0.8299 and 0.8339), and the mesh variants lead the mesh-only ULIP and Uni3D-L. Beyond these alignment wins, Luce produces per-modality material decomposition natively and supports dual decoding to both Gaussians and textured meshes. The generated assets are relightable under novel illumination (Fig. 1). Luce reproduces legible text and fine surface detail (Fig. 5) that baselines often blur or distort.

Ablations. We run two ablations. First, we ablate image conditioning by training Luce with single-layer instead of multi-layer DINOv2 features; multi-layer improves all metrics (Appendix C, Table 4). Second, for tangent-space normal map transfer, baking the decoded normal Gaussians onto the mesh (Sec. 3.4) improves normal fidelity across all metrics and enhances color reconstruction (Table 2). Figure 7 compares each example with and without normal baking using shaded color and

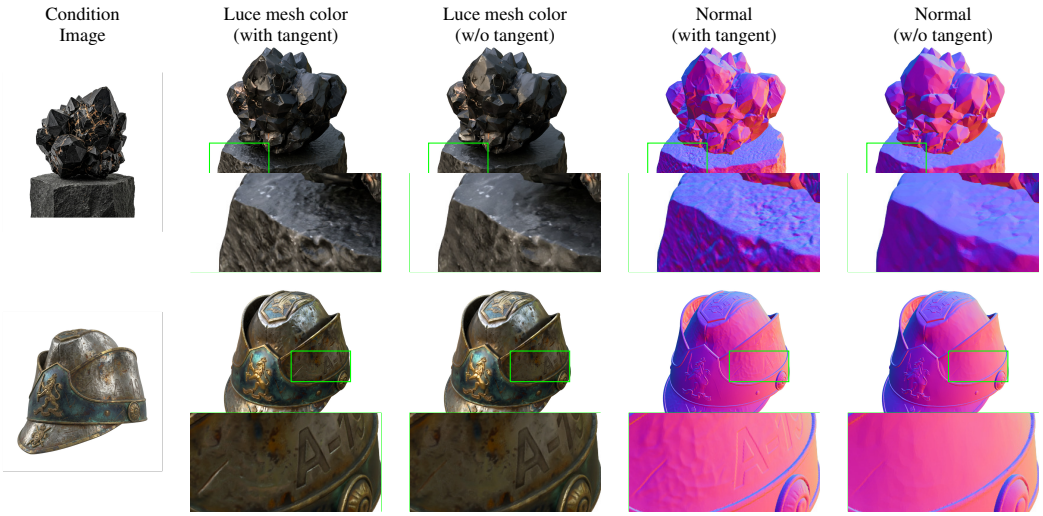

Figure 7: **Tangent-space normal map transfer.** Luce bakes tangent-space normals from decoded normal Gaussians onto the extracted mesh, recovering fine surface detail at zero polygon-count cost. Each green box marks the region magnified below.

surface-normal renderings. Baking recovers fine surface details, such as rough textures, engravings, and dents, that are smoothed out by the bare mesh.

4.3 Reconstruction

We report reconstruction as supporting evidence for our representation choices; generation (Sec. 4.2) remains our primary contribution, exercising the latent end-to-end. Table 2 reports per-modality fidelity, with qualitative examples in Fig. 6. Luce GS achieves the best color rendering (36.1 dB PSNR) and normal reconstruction (34.6 dB), surpassing TRELLIS 2, which leads on albedo and metallic-roughness, though Luce's albedo LPIPS is close (0.055 vs. 0.051).

5 Conclusion

We presented Luce, a multimodal PBR Gaussian representation for relightable image-to-3D generation. By giving each occupied voxel its own Gaussian sets for albedo, metallic-roughness, and normals, the representation makes materials explicit in a format compatible with both flow-based generation and production rendering. The learned latent is compact and diffusible, and decodes into relightable PBR Gaussians and textured meshes with tangent-space normal maps. Across standard benchmarks, these design choices yield state-of-the-art generation quality while producing relightable outputs by design. On Toys4K, Luce achieves an FID of 20.99, improving on the strongest baseline TRELLIS 2 by more than 8 FID, and leads CLIP and SigLIP2 alignment on our AI-generated-image benchmark. Together these results help bridge generative 3D modeling and production-compatible asset creation.

Limitations and future work. Luce inherits several limitations from its finite-resolution voxelized representation. Assets whose detail is fine relative to their overall extent may be under-resolved, with features spanning only a few voxels. Cascaded or adaptive-resolution decoding is a natural extension. Our current material model focuses on standard PBR attributes (albedo, metallic-roughness, and normals) under a Cook–Torrance reflectance model. While this covers many common asset types, it does not explicitly model more complex appearance effects such as subsurface scattering, anisotropy, or translucency, which additional material channels could capture. Our mesh export uses the FlexiCubes (Shen et al., 2023) decoder architecture from TRELLIS (Xiang et al., 2025); improving it with higher-resolution or learned UV-aware extraction is a direction for future work. Finally, our pipeline targets object-centric assets; extending Luce to scene-level generation with coherent lighting, material consistency, and object interactions remains an open direction.

AI USE STATEMENT

Our AI-generated-image benchmark (Sec. 4) is a synthetic data set: its 130 condition images were produced with Gemini 3.1 Flash Image (Google DeepMind, 2026), prompted for diverse, detail-rich subjects containing text, logos, and mixed materials. We inspected every image before use. They serve only as evaluation inputs and as examples in our qualitative figures. We developed the method, implemented and trained our models, prepared the training data, and designed, ran, and interpreted the experiments ourselves. Additionally, we used generative AI tools to polish the writing, help format the LaTeX layout of our figures, write parts of our evaluation scripts and visualization tools, and assist with our literature survey. We reviewed all AI-assisted work and take full responsibility for the paper's content.

ETHICS STATEMENT

As with other image-to-3D models, a misleading or harmful input image yields a matching 3D asset. Because Luce's assets are relightable and import into standard rendering pipelines, such an asset can be more convincing than the image it came from. Legible surface detail adds a further risk: a photo of a branded product yields an asset carrying the brand's logos (Fig. 2). Generated assets should be labeled as synthetic, and uses of Luce should respect the rights of the people and brands shown in input images. Luce is trained on public 3D collections (Objaverse, Objaverse-XL, TexVerse), so biases in them can carry into its outputs.

REPRODUCIBILITY STATEMENT

Luce builds on publicly released models and code: the DINOv2 image encoder, the FlexiCubes mesh decoder architecture of TRELLIS, and the pretrained sparse-structure VAE and flow of TRELLIS 2, which we use unmodified (Sec. 3.3). Section 3 defines the representation (its shading model is in Appendix A), the SLatVAE, the SLatFlow with multi-layer image conditioning, and tangent-space normal map transfer. Our training data are public 3D collections, listed in Sec. 4 with our benchmarks, baselines, and metrics. Section 4.1 summarizes how each training asset is rendered, voxelized, and fit with per-modality Gaussians. Appendix B gives the SLatVAE and SLatFlow architectures and training configurations, including training compute (Table 3). It also covers the SLatVAE objective (Eq. 3), the optimizer, and the texture-baking pipeline. Appendix C isolates the conditioning ablation. All baselines use the image conditioning and sampling steps of their public releases, and Appendix D lists the steps for every method. Appendix E specifies the evaluation renders, environment maps, and FID/KID protocol, and Appendix G gives the four views over which CLIP and SigLIP2 are averaged.

REFERENCES

Gwangbin Bae and Andrew J. Davison. Rethinking inductive biases for surface normal estimation. In *CVPR*, 2024.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024.

Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2024. URL http://www.blender.org. Version 4.1.

Jen-Hao Rick Chang, Xiaoming Zhao, Dorian Chan, and Oncel Tuzel. LiTo: Surface light field tokenization. In *ICLR*, 2026.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, Liang Pan, Dahua Lin, and Ziwei Liu. 3DTopia-XL:

Scaling high-quality 3D asset generation via primitive diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26576–26586, June 2025.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A universe of 10M+ 3D objects. *arXiv preprint arXiv:2307.05663*, 2023a.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *CVPR*, 2023b.

Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3D Gaussians: Realistic point cloud relighting with BRDF decomposition and ray tracing. In *ECCV*, 2024.

Google DeepMind. Gemini 3.1 Flash Image model card. https://deepmind.google/models/model-cards/gemini-3-1-flash-image/, February 2026. Accessed: 2026-06-03.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*, 2017.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations (ICLR)*, 2022.

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. GaussianShader: 3D Gaussian splatting with shading functions for reflective surfaces. In *CVPR*, 2024.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Heewoo Jun and Alex Nichol. Shap-E: Generating conditional 3D implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

Brian Karis. Real shading in Unreal Engine 4. In *ACM SIGGRAPH Course: Physically Based Shading in Theory and Practice*, 2013.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. DINO-Foresight: Looking into the future with DINO. *NeurIPS*, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Sébastien Lagarde and Charles de Rousiers. Moving Frostbite to physically based rendering 3.0. In *ACM SIGGRAPH Course: Physically Based Shading in Theory and Practice*, 2014.

Philipp Langsteiner, Jan-Niklas Dihlmann, and Hendrik P. A. Lensch. MatSpray: Fusing 2D material world knowledge on 3D geometry. In *CVPR*, 2026.

Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, Xiao Chen, Feipeng Tian, Jianxiong Pan, Zeming Li, Gang Yu, Xiangyu Zhang, Daxin Jiang, and Ping Tan. Step1X-3D: Towards high-fidelity and controllable generation of textured 3D assets. *arXiv preprint arXiv:2505.07747*, 2025a.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025b.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. GS-IR: 3D Gaussian splatting for inverse rendering. In *CVPR*, 2024.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019.

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.

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3D models, materials, and lighting from images. In *CVPR*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Poly Haven. HDRI environment maps. https://polyhaven.com, 2026. Licensed under Creative Commons Zero (CC0 1.0) Public Domain Dedication. Environment maps used: courtyard, ninomaru_teien, hotel_room, moonless_golf, studio_small_01, spruit_sunrise, venice_sunset.

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 *ICML*, 2021.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

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

Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics*, 42(4):37:1–37:16, 2023.

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. *Transactions on Machine Learning Research*, 2026.

Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *CVPR*, 2021.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter Image: Ultra-fast single-view 3D reconstruction. *CVPR*, 2024.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view Gaussian model for high-resolution 3D content creation. *ECCV*, 2024a.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian splatting for efficient 3D content creation. *ICLR*, 2024b.

Tencent Hunyuan3D Team. Hunyuan3D 2.1: From images to high-fidelity 3D assets with production-ready PBR material. *arXiv preprint arXiv:2506.15442*, 2025.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip H. S. Torr, and Yao Yao. Direct3D-S2: Gigascale 3D generation made easy with spatial sparse attention. In *NeurIPS*, 2025.

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 *CVPR*, 2025.

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 *CVPR*, 2026.

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 *CVPR*, 2025.

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP-2: Towards scalable multimodal pre-training for 3D understanding. *CVPR*, 2024.

Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. StableNormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics*, 2024.

Jingrui Ye, Lingting Zhu, Runze Zhang, Zeyu Hu, Yingda Yin, Lanjiong Li, Lequan Yu, and Qingmin Liao. Large material Gaussian model for relightable 3D generation. *arXiv preprint arXiv:2509.22112*, 2025.

Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. GaussianCube: A structured and explicit radiance representation for 3D generative modeling. In *NeurIPS*, 2024a.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large reconstruction model for 3D Gaussian splatting. *ECCV*, 2024b.

Tianyuan Zhang, Zhengfei Kuang, Haian Jin, Zexiang Xu, Sai Bi, Hao Tan, He Zhang, Yiwei Hu, Milos Hasan, William T. Freeman, Kai Zhang, and Fujun Luan. RelitLRM: Generative relightable radiance for large reconstruction models. In *ICLR*, 2025a.

Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. NeRFactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics*, 40(6):237:1–237:18, 2021.

Yibo Zhang, Li Zhang, Rui Ma, and Nan Cao. TexVerse: A universe of 3D objects with high-resolution textures. *arXiv preprint arXiv:2508.10868*, 2025b.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3D: Exploring unified 3D representation at scale. *ICLR*, 2024.