# NeAR: Coupled Neural Asset–Renderer Stack

Hong Li[1,2*] Chongjie Ye[3,11*] Houyuan Chen[4] Weiqing Xiao[5] Ziyang Yan[6]
Lixing Xiao[7] Zhaoxi Chen[8] Jianfeng Xiang[9] Shaocong Xu[2] Xuhui Liu[1] Yikai Wang[10]
Baochang Zhang[1†] Xiaoguang Han[11,3] Jiaolong Yang Hao Zhao[12†]
[1]BUAA [2]BAAI [3]FNii, CUHKSZ [4]HKUST [5]NJU [6]UniTn
[7]ZJU [8]NTU [9]THU [10]BNU [11]SSE, CUHKSZ [12]AIR, THU

*Project Page:* near-project.github.io

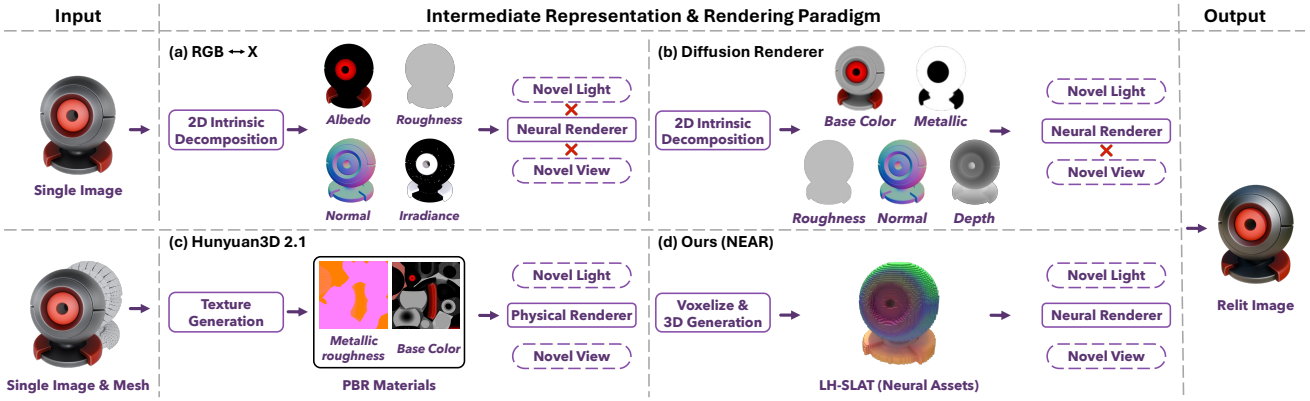


Figure 1. **Overview of NeAR vs. Existing Single Image Relighting Frameworks.** (a-b) Existing 2D methods lack explicit 3D awareness; specifically, (a) struggles to disentangle specular highlights, while both fail to guarantee multi-view consistency during relighting. (c) State-of-the-art 3D generation methods decouple asset authoring from rendering, relying on ill-posed PBR decomposition that often results in material inaccuracies and baked-in artifacts. In contrast, (d) **NeAR (Ours)** employs a **Coupled Neural Asset–Renderer Stack**. By utilizing the **LH-SLAT** representation, we simultaneously achieve photorealistic relighting and consistent novel-view synthesis.

## Abstract

*Neural asset authoring and neural rendering have traditionally evolved as disjoint paradigms: one generates digital assets for fixed graphics pipelines, while the other maps conventional assets to images. However, treating them as independent entities limits the potential for end-to-end optimization in fidelity and consistency. In this paper, we bridge this gap with **NeAR**, a **Coupled Neural Asset–Renderer Stack**. We argue that co-designing the asset representation and the renderer creates a robust "contract" for superior generation. On the **asset** side, we introduce the **Lighting-Homogenized SLAT (LH-SLAT)**. Leveraging a rectified-flow model, NeAR lifts casually lit single images into a canonical, illumination-invariant latent space, effectively suppressing baked-in shadows and highlights. On the **renderer** side, we design a **lighting-aware neural decoder** tailored to interpret these homogenized latents. Conditioned on HDR environment maps and camera views, it synthesizes relightable 3D Gaussian splats in real-time without per-object optimization. We validate NeAR on four tasks: (1) G-buffer-based forward rendering, (2) random-lit reconstruction, (3) unknown-lit relighting, and (4) novel-view relighting. Extensive experiments demonstrate that our coupled stack outperforms state-of-the-art baselines in both quantitative metrics and perceptual quality. We hope this coupled asset-renderer perspective inspires future graphics stacks that view neural assets and renderers as co-designed components instead of independent entities.*

## 1. Introduction

Images are determined by the interaction of light with scene geometry, materials, and lighting. Classical computer graphics separates this process into asset authoring, where artists define scene properties, and rendering, where a physically based renderer simulates light transport. While effective, this separation requires substantial manual effort, computationally expensive simulations, and makes inverse reconstruction from real-world images or video challenging.

*Equal contribution. †Corresponding authors.

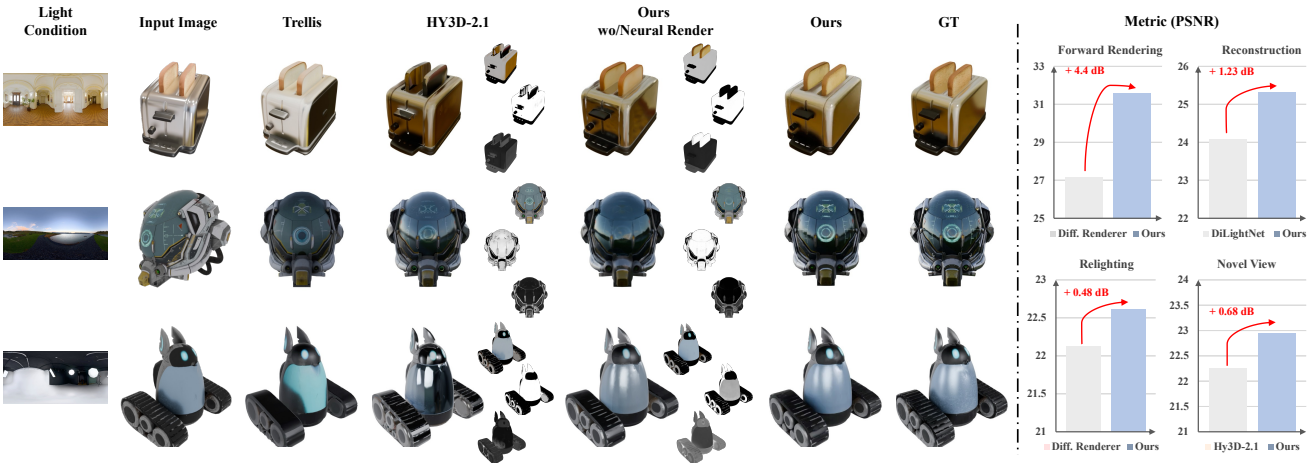


Figure 2. **Comparison of NeAR and Decoupled Paradigms**. Left: Visual results under target illumination. Cols. 3–5 are rendered via Blender to evaluate asset quality. Insets (right of cols. 4&5) display PBR maps (top-down: Base Color, Metallic, Roughness). Baselines suffer from baked-in lighting (Trellis) or material ambiguity (HY3D-2.1). Notably, HY3D-2.1 wrongly assigns high metallic values to the bread (see Metallic map, Row 1) and exhibits inconsistent highlights on the robot (Row 3). While our intermediate PBR decomposition (col. 5) corrects materials, it struggles with complex effects like transparency (Helmet, Row 2) under standard rendering. Our full Neural Renderer (col. 6) resolves this, yielding photorealistic results closest to GT. Right: Quantitative results on the Glossy Synthetic dataset. NeAR achieves the highest PSNR across all four tasks, demonstrating the superiority of our coupled stack.

Recent advances in neural graphics [3, 6, 16, 20, 28, 44, 45, 48, 52, 54, 55, 63] address these limitations from two complementary directions: *neural asset authoring* uses generative models [3, 5, 6, 20, 44, 45, 48, 55, 63] to synthesize full 3D assets for traditional pipelines, reducing manual effort, while *neural renderers* map these assets—often converted into intermediate representations such as depth, normals, or shading buffers—directly to images [23, 52, 54], providing a data-driven alternative to analytic rendering and enabling more robust inverse inference. Fig. 1 shows a comparison between our method and previous single-image relighting frameworks.

Despite recent progress in generating 3D assets with PBR materials [3, 5, 6, 20, 44, 45, 48, 55, 63], a fundamental limitation remains: asset generation and neural rendering are typically developed in isolation, with assets created assuming a fixed renderer and renderers trained on static asset distributions. This separation becomes problematic when errors in asset decomposition—such as misidentified albedo or incorrect normal maps—propagate through the rendering pipeline. Because rendering is a nonlinear process, small errors in asset decomposition compound into visible artifacts like baked-in shadows or lighting inconsistencies. Fig. 2 demonstrates this issue: existing methods rendered with traditional physically-based renderers (e.g., Blender) exhibit lighting artifacts and fail to achieve faithful relighting.

To this end, we propose **NeAR**, a *Coupled Neural Asset–Renderer Stack* for single-image relightable 3D generation. Our key insight is to co-design the asset representation and rendering process to enable relighting directly through a shared, lighting-homogenized latent space. On the **asset** side, we introduce a *Lighting-Homogenized Structured 3D Latent (LH-SLAT)*. Unlike standard assets that rely on fragile explicit decomposition, our model lifts the casually lit input into a canonical latent form. As visualized in Fig. 3, this process transforms a shadow-affected representation (Shaded-SLAT) into a clean, homogenized state, effectively suppressing baked-in shadows and unstable highlights while preserving geometric cues. On the **renderer** side, we design a *lighting-aware neural renderer*. Conditioned on a lighting tokenizer, this renderer learns to interpret the homogenized latents and synthesize view-dependent appearance under arbitrary HDR environments via differentiable 3D Gaussian splatting. By unifying the representation, NeAR generates assets that naturally support real-time, high-quality relighting and novel-view synthesis with consistent materials across views.

We validate NeAR across four downstream tasks: (1) G-buffer–based forward rendering, (2) random-lit single-image reconstruction, (3) unknown-lit single-image relighting, and (4) novel-view relighting. On benchmarks including Digital Twin Category, Aria Digital Twin, and Objaverse, NeAR achieves state-of-the-art or improved performance over recent neural relighting baselines in both quantitative metrics and perceptual quality, while running at real-time frame rates without per-object optimization.

Our contributions can be summarized as follows:

1. **Coupled neural asset–renderer stack.** We introduce NeAR, an learnable graphics stack where the neural asset representation and neural renderer are co-designed for single-image relightable 3D asset generation.
2. **Lighting-homogenized structured neural asset.** We

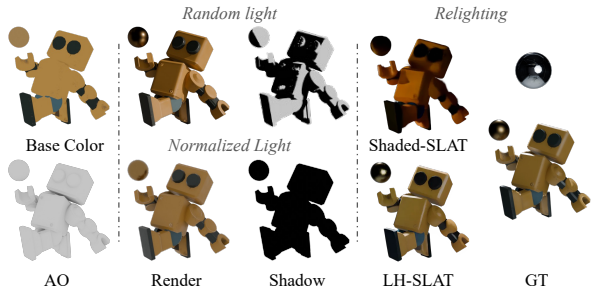


Figure 3. **Lighting homogenization as the bridge between assets and renderer.** We visualize the intrinsic components (Base Color, Ambient Occlusion), rendering results under random and uniform lighting, shadow maps, as well as relighting outputs generated respectively by Shaded SLAT and LH-SLAT. By mapping casually lit images to a canonical illumination space, LH-SLAT effectively suppresses baked-in shadows and unstable specularities while preserving geometry-consistent diffuse cues. This stable latent space serves as the robust "contract" for our lighting-aware neural renderer to enable controllable relighting.

propose a Lighting-Homogenized Structured 3D Latent (LH-SLAT) that suppresses shadows and unstable highlights while preserving geometry-consistent diffuse cues in a compact, view-agnostic 3D latent.

3. **Lighting tokenizer and lighting-aware neural 3D Gaussian renderer.** We design a lighting tokenizer and a lighting-aware neural 3D Gaussian renderer that map LH-SLAT, environment illumination, and view embeddings into a relightable 3D Gaussian field rendered via differentiable Gaussian splatting.
4. **Extensive evaluation and real-time performance.** We demonstrate on multiple datasets and tasks that NeAR delivers state-of-the-art or better quality with strong generalization and consistent multi-view rendering, while enabling real-time feed-forward inference.

## 2. Related Works

### 2.1. Image relighting and inverse rendering

Image relighting and inverse rendering lie at the intersection of geometry, material estimation, and light transport, and have been studied from both physics- and data-driven perspectives [16]. Classical methods (e.g., SIRFS) recover interpretable PBR maps (albedo, roughness, normals) via optimization with hand-crafted priors [1]. While interpretable and editable, these approaches are highly ill-posed in real scenes: shadows, inter-reflections, and view-dependent highlights bias material estimation, leading to baked-in artifacts under re-rendering.

Recent learning-based approaches fall into two categories. The first focuses on physically structured decomposition [10, 23, 63], which yields interpretable assets but often requires multi-view data or costly per-object optimization to resolve ambiguities. The second targets diffusion-based 2D relighting [11, 28, 52, 56]. Methods such as DiLightNet and IC-Light leverage diffusion priors for high-fidelity relighting with fine control, but are computationally expensive, stochastic, and limited to 2D, lacking multi-view consistency for 3D applications.

We take a middle path: rather than brittle PBR inversion or black-box diffusion, we homogenize illumination into a canonical form (LH-SLAT) and synthesize a relightable 3D field feed-forward, improving stability and controllability.

### 2.2. Generative 3D Priors and Representations

Diffusion priors and score-distillation sampling (SDS) have catalyzed rapid progress in text-to-3D and image-to-3D generation [34, 38, 40, 46, 59–61]. While SDS-based methods transfer 2D generative knowledge to 3D effectively, they suffer from slow iterative optimization. Consequently, recent works have shifted toward feed-forward 3D reconstruction models trained on large-scale 3D datasets [14, 45, 55]. Specifically, Trellis [45] utilizes Structured 3D Latents (SLAT) to compress complex geometry and appearance into sparse tokens, enabling efficient decoding.

Concurrently, 3D Gaussian Splatting (3DGS) [18] has emerged as a rasterization-friendly representation supporting real-time differentiable rendering. While current feed-forward models (like LRM or Trellis) excel at geometry, they typically bake lighting into the texture, limiting downstream utility. Our method builds upon the efficiency of SLAT and 3DGS but fundamentally redesigns the generation process. We introduce a *lighting-homogenized* variant of SLAT and a custom neural decoder, replacing static texture prediction with a relightable neural field.

### 2.3. Relightable 3D asset synthesis

Producing relightable 3D assets requires models to represent both intrinsic surface properties and lighting-dependent transport (shadows, speculars, interreflections). Prior works condition NeRFs, Gaussian splats or meshes on lighting inputs to enable relighting-aware outputs [2, 12, 16, 21, 33, 36, 47, 51, 62]. Many approaches either use volumetric neural renderers that are costly at inference, or attempt to estimate PBR maps without lighting supervision, which leads to poor disentanglement [26, 35, 39]. Some models explore large inverse-rendering architectures to predict PBR properties from sparse views, but computational cost and optimization per-object remain bottlenecks [22, 58]. Recent works [10, 41] employ diffusion models to generate multi-view material maps or multi-view relighted images, followed by 3D reconstruction. However, the absence of explicit 3D constraints in the generation stage makes it difficult to guarantee consistency across views.

In contrast, our *homogenize-then-synthesize* strategy pipeline explicitly removes unstable, scene-specific illumination before decoding. This mitigates ill-posed PBR in-

version, enabling a feed-forward decoder to produce relightable 3DGS with real-time consistency. NeAR thus combines the stability of interpretable pipelines with the fidelity of neural rendering.

# 3. Method

## 3.1. Preliminary

**3D Gaussian Splatting (3DGS).** 3DGS [18] represents scenes with anisotropic Gaussians, rendered via splatting and $\alpha$-blending: $C = \sum_{i\in\mathcal{N}} c_i\alpha_i \prod_{j<i}(1-\alpha_j)$. Crucially, standard 3DGS models color $c_i$ using Spherical Harmonics (SH), which inherently bakes static lighting into the representation. To enable relighting, we forego SH and predict color dynamically conditioned on target illumination.

**Structured 3D Latents (SLAT).** Following Trellis [45], we use SLAT to encode 3D assets efficiently. A SLAT $\mathcal{Z} = \{(\boldsymbol{z}_k, \boldsymbol{p}_k)\}_{k=1}^{K}$ consists of $K$ active feature tokens, where each token $\boldsymbol{z}_k \in \mathbb{R}^D$ is associated with a coordinate $\boldsymbol{p}_k$ in a sparse voxel grid. This representation focuses capacity on surface regions ($K \ll N^3$) and supports diverse decoding heads. However, standard SLATs blindly encode input appearance—including shadows and highlights. Our goal is to transform $\mathcal{Z}$ into a *lighting-homogenized* form, canonicalizing the appearance to a uniform illumination while preserving geometry.

## 3.2. Overview of NeAR

The challenge in single-image 3D relighting lies in disentangling lighting from intrinsic object properties, since shadows, highlights, and interreflections are inherently entangled with geometry. To avoid unstable PBR inversion and black-box neural generation, we propose a *homogenize-then-synthesize* framework that functions as a coupled stack. NeAR first extracts a Lighting-Homogenized SLAT (LH-SLAT) from the input image to neutralize lighting effects, then decodes a relightable 3DGS. Our framework consists of two stages:

**Stage 1: Light Homogenization-SLAT Generation.** We first utilize the pre-trained flow model $f_s$ to map the arbitrarily lit input $I_{\text{in}}$ into an initial shaded SLAT $Z_s$. Operating within this sparse voxel space, we employ a LoRA-adapted model $f_\theta$ to steer the latent representation from $Z_s$ toward a Lighting-Homogenized SLAT (LH-SLAT) $Z_{\text{lh}}$:

$$Z_{\text{lh}} = f_\theta(Z_s, I_{\text{in}}) = f_\theta(f_s(I_{\text{in}}), I_{\text{in}}). \tag{1}$$

Specifically, $Z_{\text{lh}}$ suppresses the baked-in shadows and highlights inherent in $Z_s$, establishing a stable light-homogenized space. This representation preserves essential geometry-material-light interactions, yielding a unified and generalizable foundation for the relighting task.

**Stage 2: Relightable Neural 3DGS Synthesis.** Leveraging the homogenized representation $Z_{\text{lh}}$, a feed-forward decoder $\mathcal{D}$ synthesizes a relightable Gaussian field $\mathcal{G}$. This process is conditioned on the target view $\mathbf{v}_{\text{target}}$ and the target illumination $L_{\text{target}}$, encoded via $\mathcal{E}_l$:

$$\mathcal{G} = \mathcal{D}\big(Z_{\text{lh}}, \mathbf{v}_{\text{target}}, \mathcal{E}_l(L_{\text{target}})\big). \tag{2}$$

Finally, the relighted image is rendered using a differentiable GS rasterizer $\mathcal{M}$:

$$I_{\text{target}} = \mathcal{M}(\mathcal{G}, \mathbf{v}_{\text{target}}). \tag{3}$$

In the following, we describe Stage 1 (Sec. 3.3) and Stage 2 (Sec. 3.5) in detail.

## 3.3. Light Homogenization & LH-SLAT Rec.

The first stage generates a Lighting-Homogenized Structured 3D Latent (LH-SLAT) $Z_{\text{lh}}$ from a single input image $I_{\text{in}}$. This representation serves as a stable, illumination-invariant substrate for downstream synthesis.

**Lighting Homogenization.** We define the homogenized light $E_h$ as a uniform, white ambient environment illumination. Extracting SLAT features under such lighting captures intrinsic geometric and material cues uncorrupted by transient lighting effects, serving as a robust basis for relighting.

**LH-SLAT Reconstruction.** To train $f_\theta$, we prepare paired data $(I_{\text{in}}, Z_{\text{lh}})$ via multi-step rendering of 3D assets under homogenized lighting. As shown in Fig. 4 top left corner, we first generate the ground-truth homogenized latents $Z_{\text{lh}}$: (1) for each 3D asset, we render $N$ views under our defined homogenized illumination; (2) we extract dense 2D visual features using a pre-trained DINOv2 model; (3) these features are back-projected into a sparse 3D voxel grid; (4) finally, this sparse grid is compressed by a pre-trained SLAT VAE encoder to obtain $Z_{\text{lh}}$. Second, to create the corresponding input $I_{\text{in}}$, we render $M$ additional images of the same asset under diverse, random lighting conditions and camera poses.

Optionally, for highly reflective materials, we extract Basecolor SLAT $Z_{\text{bc}}$ from multi-view basecolor renderings, concatenating with $Z_{\text{lh}}$ to retain base color information.

## 3.4. LH-SLAT Generation

As shown in Fig. 4 top right corner, we use a rectified flow model $f_\theta$ to generate the lighting-homogenized SLAT $Z_{\text{lh}}$ from the input image $I_{\text{in}}$. The rectified flow model is trained to learn the mapping between the arbitrarily lit image and the corresponding latent representation under our homogenized lighting conditions. Specifically, we utilize a pre-trained SLAT rectified flow model $f_s$ to generate the shadowed SLAT $Z_s$ from the input image $I_{\text{in}}$, and subsequently fine-tune $f_s$ using LoRA [15] in the sparse voxel space [45] to achieve lighting homogenization. The loss function for training is the conditional flow matching loss $\mathcal{L}_{stage1}$:

$$\mathcal{L}_{stage1} = \mathbb{E}_{t,\boldsymbol{z}_0,\boldsymbol{\epsilon}} \|\boldsymbol{v}_\theta(\boldsymbol{z}, Z_s, I_{in}, t) - (\boldsymbol{\epsilon} - \boldsymbol{z}_0)\|_2^2, \tag{4}$$

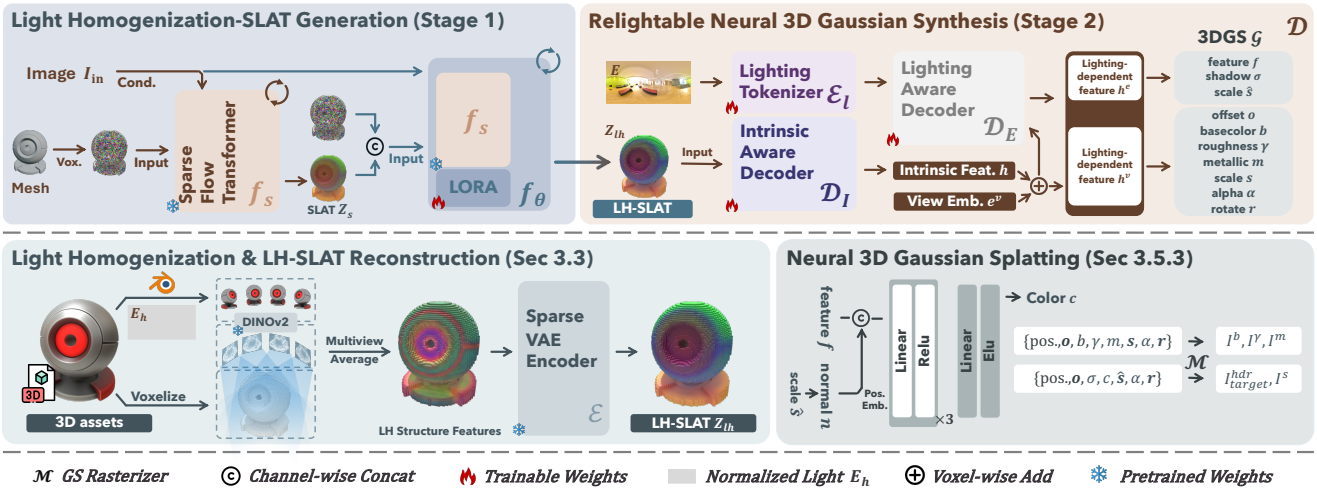


Figure 4. **Pipeline of NeAR as a coupled neural asset-renderer stack. Top (Inference Stage):** An end-to-end inference pipeline. Given a single image and a **geometry prior** (e.g., mesh from HY3D), Stage 1 utilizes a rectified-flow backbone with LoRA adaptation to predict the **Lighting-Homogenized SLAT (LH-SLAT)**. This latent acts as a bridge, which is then consumed by the Stage 2 lighting-aware neural renderer to synthesize relightable 3DGS under novel illumination and viewpoints. **Bottom-Left (Data Prep):** Offline construction of ground-truth LH-SLATs by rendering assets under homogenized illumination and encoding them via a sparse VAE. **Bottom-Right (GS Decoding & Rendering):** Detailed architecture of the 3DGS decoding head, which predicts Gaussian attributes from lighting-dependent features, followed by a differentiable rasterizer $\mathcal{M}$ that renders the final HDR image, shadow and PBR auxiliary maps.

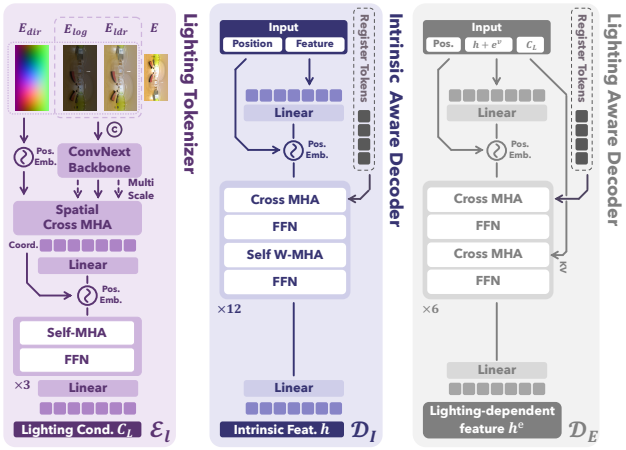


Figure 5. Architectures of Lighting Tokenizer, IAD, and LAD.

where $\boldsymbol{z}(t) = (1-t)\boldsymbol{z}_0 + t\boldsymbol{\epsilon}$ is the linear interpolation between the data sample $\boldsymbol{z}_0$ and noise $\boldsymbol{\epsilon}$, and $\boldsymbol{v}_\theta$ approximates the time-dependent vector field. If the optional basecolor SLAT $\boldsymbol{z}_{\text{bc}}$ is used, it is concatenated with $\boldsymbol{z}_{\text{lh}}$ to provide additional color information to the subsequent stage.

## 3.5. Relightable Neural 3D Gaussian Synthesis

The second stage synthesizes a relightable 3D Gaussian Splatting (3DGS) field $\mathcal{G}$ from LH-SLAT, conditioned on target illumination and viewpoint. Unlike optimization approaches [2, 12], we employ an efficient feed-forward decoder with two sequential modules: the *Intrinsic Aware Decoder (IAD)* and the *Lighting Aware Decoder (LAD)*.

### 3.5.1. Intrinsic Aware Decoder (IAD)

The IAD, denoted as $\mathcal{D}_I$, aims to process LH-SLAT $Z_{\text{lh}}$ and generate a view-independent and illumination-invariant intrinsic feature $\boldsymbol{h} = \{(\boldsymbol{h}_i, \boldsymbol{p}_i)\}_{i=1}^{L}$, where $\boldsymbol{h}_i \in \mathbb{R}^{768}$. This sparse feature field $\boldsymbol{h}$ effectively decodes the underlying geometric structure and material properties of the scene. To achieve this, IAD employs a Transformer architecture akin to TRELLIS [45], leveraging stacked self-shifted window attention blocks to exploit the inherent locality of structured 3D latent sequences. To further enhance the model's comprehension of global structural relationships and lighting context, a register cross-attention layer is incorporated into each block. Specifically, 16 learnable register tokens are appended to each SLAT sequence to capture global context and suppress high-frequency noise [7, 19]. These register tokens are injected into the decoder via global cross-attention, facilitating information exchange with all latent variable tokens and enabling a coherent and globally consistent intrinsic representation $\boldsymbol{h}$.

### 3.5.2. Lighting Aware Decoder (LAD)

The LAD, denoted as $\mathcal{D}_E$, synthesizes the final lighting-dependent features by injecting view embeddings and environmental lighting conditions, as shown in Fig. 4.

**Observe View Embedding**. To explicitly model specular highlights that vary with viewing angles, we abandon the commonly used spherical harmonics and instead inject the observed view information into the learning process of LAD from the outset to enhance the model's perception of specular highlights. Along the camera ray to each voxel

Table 1. Quantitative comparison against state-of-the-art methods across four sub-tasks.

| | ADT [32] | | | DTC [9] | | | Objaverse data [8] | | | Glossy Synthetic dataset [24] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ |
| **G-Buffers Forward Rendering** | | | | | | | | | | | | |
| DiffusionRenderer [23] | 0.0802 | 24.41 | 0.9172 | 0.0560 | 27.16 | 0.9354 | 0.0616 | 27.09 | 0.9288 | 0.0707 | 25.46 | 0.9126 |
| Ours | **0.0488** | **29.15** | **0.9484** | **0.0458** | **31.59** | **0.9586** | **0.0490** | **32.23** | **0.9627** | **0.0475** | **30.47** | **0.9594** |
| **Random-lit Single-image Reconstruction** | | | | | | | | | | | | |
| RGB↔X [54] | 0.1605 | 15.15 | 0.8445 | 0.1349 | 15.48 | 0.8624 | 0.1199 | 16.09 | 0.8801 | 0.1271 | 14.29 | 0.8612 |
| DiLightNet [52] | 0.0949 | 21.11 | 0.8947 | 0.0650 | 23.53 | 0.9147 | 0.0507 | 25.65 | 0.9300 | 0.0523 | 24.09 | 0.9213 |
| DiffusionRenderer [23] | 0.0767 | 22.50 | 0.9105 | 0.0579 | 23.70 | 0.9234 | 0.0516 | 24.81 | 0.9285 | 0.0547 | 23.40 | 0.9163 |
| Ours | **0.0754** | **22.89** | **0.9116** | **0.0532** | **24.68** | **0.9246** | **0.0394** | **26.53** | **0.9305** | **0.0368** | **25.32** | **0.9274** |
| **Unknown-lit Single-image Relighting** | | | | | | | | | | | | |
| DiLightNet [52] | 0.1037 | 20.59 | 0.8813 | 0.0729 | 22.63 | 0.8913 | 0.0657 | 23.87 | 0.9011 | 0.0622 | 22.40 | 0.9059 |
| NeuralGrafferer [16] | 0.2675 | 14.31 | 0.7839 | 0.2548 | 14.22 | 0.7943 | 0.2108 | 14.68 | 0.8238 | 0.1767 | 15.67 | 0.8200 |
| DiffusionRenderer [23] | 0.0916 | 21.91 | 0.8960 | 0.0691 | 22.99 | 0.9078 | 0.0609 | 23.75 | 0.9169 | 0.0632 | 22.13 | 0.9062 |
| Ours | **0.0915** | **21.95** | **0.8972** | **0.0642** | **23.47** | **0.9177** | **0.0557** | **24.38** | **0.9264** | **0.0465** | **22.61** | **0.9246** |
| **Novel-view Relighting** | | | | | | | | | | | | |
| 3DTopia-XL [5] | 0.1754 | 17.24 | 0.8013 | 0.1051 | 21.56 | 0.8674 | 0.0769 | 23.22 | 0.8989 | 0.0857 | 20.89 | 0.8807 |
| Stable-Fast-3D [3] | 0.1028 | 19.43 | 0.8881 | 0.0616 | 22.07 | 0.9154 | 0.0666 | 22.26 | 0.9112 | 0.0747 | 20.17 | 0.8943 |
| MeshGen [6] | 0.0939 | 20.15 | 0.8879 | 0.0661 | 22.87 | 0.9101 | 0.0509 | 24.15 | 0.9306 | 0.0637 | 21.43 | 0.9071 |
| Hunyuan3D-2.1 [63] | 0.0727 | 22.30 | 0.9017 | 0.0481 | 24.89 | 0.9255 | 0.0479 | 25.47 | 0.9328 | 0.0533 | 22.26 | 0.9119 |
| Ours | **0.0693** | **22.87** | **0.9023** | **0.0475** | **25.53** | **0.9298** | **0.0486** | **25.97** | **0.9392** | **0.0502** | **22.94** | **0.9147** |

$\boldsymbol{p}_i$ in the world coordinate system, we record the distance $x = \{(l_i, \boldsymbol{p}_i)\}_{i=1}^{L}$, where $l_i \in \mathbb{R}$, and the ray direction $\boldsymbol{d}^w = \{(\boldsymbol{d}^w{}_i, \boldsymbol{p}_i)\}_{i=1}^{L}$. We then transform $\boldsymbol{d}^w$ to the camera coordinate system using the extrinsic matrix, denoted as $\boldsymbol{d} = \{(\boldsymbol{d}_i, \boldsymbol{p}_i)\}_{i=1}^{L}$, where $\boldsymbol{d}_i \in \mathbb{R}^3$. We apply NeRF positional encoding and learnable positional encoding to $\boldsymbol{d}$ and $l$ voxel-wise, respectively, obtaining the view embedding:

$$\boldsymbol{e}^v = \{\boldsymbol{e}^d, \boldsymbol{e}^l\} = \{(\boldsymbol{e}_i^d, \boldsymbol{p}_i), (\boldsymbol{e}_i^l, \boldsymbol{p}_i)\}_{i=1}^{L}, \quad \boldsymbol{e}_i \in \mathbb{R}^{768}.$$

Then, we add $\boldsymbol{e}^d$ and $\boldsymbol{e}^l$ voxel-wise to $\boldsymbol{h}$ to obtain $\boldsymbol{h}^v$, which serves as the input to LAD.

**Lighting Tokenizer**. We encode the high dynamic range (HDR) environment map $\mathbf{E}$ into compact lighting conditions using an HDRI encoder $\mathcal{E}_l$. Following [13, 16, 23], we decompose $\mathbf{E}$ into a tone-mapped LDR image $\mathbf{E}_{\text{ldr}}$, a normalized log-intensity map $\mathbf{E}_{\text{log}} = \log(\mathbf{E}+1)/\mathbf{E}_{max}$, and a camera-space direction encoding $\mathbf{E}_{\text{dir}} \in \mathbb{R}^{H\times W\times 3}$. Unlike prior works that compress the entire map via VAE, we employ a ConvNeXt backbone to extract multi-scale visual features from $\mathbf{E}_{\text{ldr}}$ and $\mathbf{E}_{\text{log}}$. Crucially, rather than directly compressing $\mathbf{E}_{\text{dir}}$, we first encode it via NeRF-style positional embedding [30] and fuse it with visual features using **Spatial Cross Attention**. This mechanism acts as a learnable positional encoding, modulating visual features with explicit directional cues. The resulting multi-scale features are concatenated, processed with positional encoding, and passed through self-attention blocks to yield the Lighting Condition Tokens $C_L \in \mathbb{R}^{4096\times 768}$. This design explicitly embeds directional information, facilitating editable lighting directions when switching views.

**LAD Architecture**. LAD consists of stacked cross-attention blocks that inject lighting condition $C_L$ into intrinsic features $\boldsymbol{h}^v$, enabling lighting awareness. Similar to IAD, each block includes a register cross-attention layer to enhance global illumination perception. The output is the lighting-aware sparse feature $\boldsymbol{h}^e$.

### 3.5.3. Neural 3D Gaussian Splatting

We regress the 3DGS parameters using both the intrinsic feature $\boldsymbol{h}$ and the lighting-aware feature $\boldsymbol{h}^e$:

$$\begin{aligned}&\{(\boldsymbol{h}_i^v, \boldsymbol{p}_i)\}_{i=1}^{L} \rightarrow \{\{(\boldsymbol{o}_i^k, \boldsymbol{b}_i^k, \gamma_i^k, \boldsymbol{m}_i^k, \boldsymbol{s}_i^k, \alpha_i^k, \boldsymbol{r}_i^k)\}_{k=1}^{K}\}_{i=1}^{L},\\ &\{(\boldsymbol{h}_i^e, \boldsymbol{p}_i)\}_{i=1}^{L} \rightarrow \{\{(\boldsymbol{f}_i^k, \hat{s}_i^k, \sigma_i^k)\}_{k=1}^{K}\}_{i=1}^{L}\end{aligned} \tag{5}$$

the intrinsic feature $\boldsymbol{h}_i$ is decoded into $K$ Gaussian parameters: position offset $\boldsymbol{o}$, base color $\boldsymbol{b}$, roughness $\gamma$, metallic $\boldsymbol{m}$, scale $\boldsymbol{s}$, rotation $\boldsymbol{r}$, and opacity $\alpha$ (activated via $\tanh$ to support negative density [64]). Simultaneously, the lighting-dependent feature $\boldsymbol{h}_i^e$ predicts the 48-dim color feature $\boldsymbol{f}$, lighting-specific scale $\hat{s}$, and shadow $\sigma$. The Gaussian centers are defined as $\boldsymbol{x}_i^k = \boldsymbol{p}_i + \tanh(\boldsymbol{o}_i^k)$, with normals derived from the shortest axis of $\hat{s}$. Finally, we employ a simple shallow MLP network that combines the positional encoding of the normal vector and the color feature $\boldsymbol{f}$. This network uses ReLU activation functions in its intermediate layers and an ELU activation function in its final layer to predict the radiance values for each Gaussian. Through the rasterization operation $\mathcal{M}$, we obtain the 2D HDR prediction $I_{target}^{hdr}$. We also render 2D base color, roughness, metallic, shadow images $I^b, I^r, I^m, I^s$.

**Loss Function**. We supervise the training via an HDR reconstruction loss $\mathcal{L}_{hdr}$, which comprises $\mathcal{L}_1$, LPIPS [57], D-SSIM and regularization terms. Following [53], to prevent high-intensity regions from dominat-

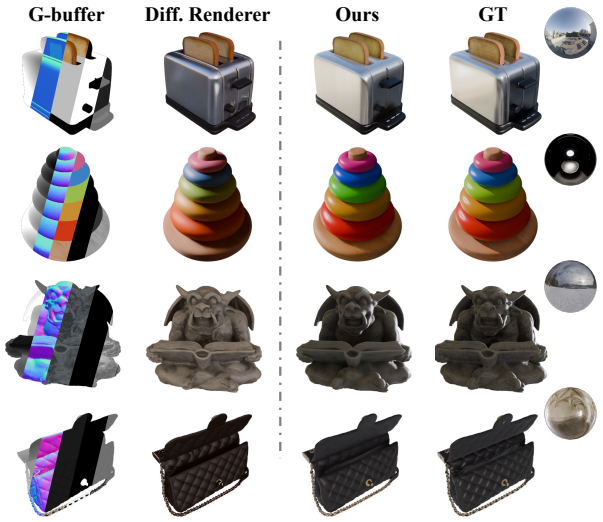


Figure 6. Visual comparison of Diffusion Renderer (with G-buffer) and our LH-SLAT method for image relighting.

ing the $\mathcal{L}_1$ optimization, we apply a logarithmic transformation to the HDR images. For perceptual metrics (LPIPS and D-SSIM), we operate on tone-mapped images using $\text{clamp}(\log_2(I), 0, 1)$. Additionally, we impose auxiliary $\mathcal{L}_1$ supervision on material properties maps (base color, roughness, metallic), denoted as $\mathcal{L}_{pbr}$, and shadows $\mathcal{L}_{shadow}$. The total objective is formulated as follows:

$$\mathcal{L}_{stage2} = \mathcal{L}_{hdr} + \lambda_{pbr}\mathcal{L}_{pbr} + \lambda_{shadow}\mathcal{L}_{shadow}. \quad (6)$$

# 4. Experiments

## 4.1. Implementation Details

Please refer to the Supplementary Material for comprehensive implementation details.

**Training Data**. Our training dataset comprises 87K 3D assets with physically-based rendering (PBR) textures, curated from the Objaverse-XL dataset. These assets are illuminated using 2K High Dynamic Range Images (HDRIs), each at 4K resolution, used as environment maps. We normalized the assets to fit within a bounding box of $[-0.5, 0.5]$. The first training stage involves rendering 150 viewpoints under normalized lighting to extract illumination-invariant structural latent representations. For input images under unknown illumination, camera poses are sampled with yaw within ±45 degrees and pitch from -10 to 45 degrees, oriented towards the object's center, and with field of view (FOV) and radius following [45]. Unknown illumination is modeled with (1) six area lights uniformly distributed on a sphere, (2) 1-3 area lights randomly sampled within the camera's hemisphere, or (3) a random, Z-axis-rotated environment map. Area light intensities are sampled uniformly between 300 and 700 (units), distances between 5 and 8 units. In the second stage, we re-light objects using randomly rotated environment maps as supervision, with a fixed FOV of $40°$. We randomly and uniformly sample 12 camera viewpoints on a sphere of radius 2.0, where each viewpoint is rendered under 16 different illumination conditions. All data generation across both stages utilizes the Blender EEVEE Next engine [42] with raytracing enabled.

**Task Definitions And Baselines**. We evaluate our method on two fundamental tasks: single-view forward rendering and novel view relighting from single-image to Relightable 3D. We evaluate the consistency between the rendered outputs and the ground truth reference images. The former involves single-view forward rendering with input G-buffers (such as normals, material, and depth information), image reconstruction from a single-image under random lighting, and relighting of a single image under unknown lighting. For single-view forward rendering, we compare against recent state-of-the-art neural rendering methods RGB↔X [54], neural-gaffer [16], DiLightNet [52], and Diffusion-render [23]. For novel view relighting, we compare against recent open-source methods that support single-image to 3D generation with PBR materials, including Huyuan3D-2.1 [63] (HY3D 2.1), MeshGen [6], 3DTopia-XL [5], and SF3D [3]. The schematic diagram for the four subtasks is illustrated in Fig. 9. We additionally present qualitative results for PBR material estimation in comparison with HY3D 2.1.

**Evaluation Metric**. We use PSNR, SSIM [43] and LPIPS [57] to measure the quality of the rendering.

**Evaluation Datasets**. We construct a test set by randomly selecting 800 unseen objects from our training data. To validate generalization capability, we evaluate on out-of-domain datasets: Aria Digital Twin (ADT) [32] and Digital Twin Catalog (DTC) [9], which feature high-fidelity photo-realistic models with sub-millimeter accuracy. We also incorporate the Glossy Synthetic dataset [24] and additional assets from BlenderKit[1], modifying rendering nodes to utilize the Principled BSDF shader[2].

## 4.2. Single-view Forward Rendering

**G-buffers Forward Rendering**. As shown in Fig. 6, we compare against Diffusion Renderer using ground truth G-buffers and LH-Slat , bypassing the single-image-to-intermediate representation step. Our method demonstrates superior accuracy in shadow and highlight distribution (e.g., the toy's specular highlight and the sculpture's shadow detail), likely due to our explicit 3D structural information. Furthermore, we accurately capture material reflections of ambient light, as illustrated by the stainless steel. Quantitatively, our method significantly outperforms baselines across four datasets in Tab. 1.

**Random-lit Single-image Reconstruction**. As shown in Figs. 14 and 15, our method achieves higher reconstruc-

[1]https://www.blenderkit.com/
[2]https://www.blender.org/

Table 2. Ablation study on the number of blocks for $\mathcal{D}_I$ and $\mathcal{D}_E$.

| Num | PSNR | SSIM | LPIPS | $\mathcal{D}_E$ Param. | FPS |
|---|---|---|---|---|---|
| 12 + 1 | 31.56 | 0.9608 | 0.0508 | 12.65M | 48 |
| 12 + 3 | 32.35 | 0.9635 | 0.0474 | 31.55M | 38 |
| 12 + 6 | 32.54 | 0.9649 | 0.0442 | 59.8M | 30 |
| 12 + 9 | 32.56 | 0.9645 | 0.0439 | 88.23M | 23 |
| 0 + 18 | 29.43 | 0.9245 | 0.0624 | 173.25M | 10 |

Table 3. Ablation study on decoder input SLAT types.

| SLAT types | PSNR | SSIM | LPIPS |
|---|---|---|---|
| shaded | 28.95 | 0.9281 | 0.0813 |
| base color | 30.38 | 0.9541 | 0.0564 |
| LH | 32.02 | 0.9631 | 0.0494 |
| LH + base color | 32.54 | 0.9649 | 0.0442 |

tion fidelity compared to baselines. Specifically, Diffusion Renderer and RGB-X misestimate materials, while DiLightNet exhibits color shifts. Quantitative evaluations in Tab. 1 confirm our method's advantage across all metrics.

**Unknown-lit Single-image Relighting**. Our method achieves more accurate highlights and color in relit images with unknown lighting, compared to other methods, as shown in Fig. 16 and 17. For example, observe the highlights on the speaker cones (first row) and the teapot color (second row). Tab. 1 quantitatively demonstrates the superiority of our method.

**Novel-view Relighting**. We benchmark our full pipeline (single-image to relightable 3D) against state-of-the-art generation methods. While other methods typically reconstruct a mesh and rely on Blender for relighting, we directly generate a relightable 3D Gaussian field. As shown in Figs. 2 and 18, our method achieves more realistic lighting–material interactions than image-based textured mesh methods [3, 5, 6, 63]. Quantitative results in Tab. 1 demonstrate improvements over existing 3D generation baselines.

**PBR Materials Estimation**. Fig. 12 demonstrates that our method surpasses the open-source SOTA model, HY3D 2.1, in material recovery. Hunyuan3D relies on multi-view diffusion, which often introduces view-inconsistent artifacts (e.g., blurred edges on the wooden cup). In contrast, our LH-SLAT preserves 3D consistency and retains crucial light-material interaction cues. For instance, HY3D 2.1 misclassifies wood as metal, resulting in erroneous metallic artifacts on the eggs, whereas our method correctly recovers the material properties.

### 4.3. Ablation Study.

We perform ablation studies on our test set, investigating the Variants of $\mathcal{D}$ and input SLAT types.

**Variants of $\mathcal{D}$**. Tab. 2 indicates that increasing the depth of $\mathcal{D}_E$ improves quality but reduces inference speed; we therefore select 6 layers to strike a balance between efficiency and performance. Relying solely on the LAD $\mathcal{D}_E$

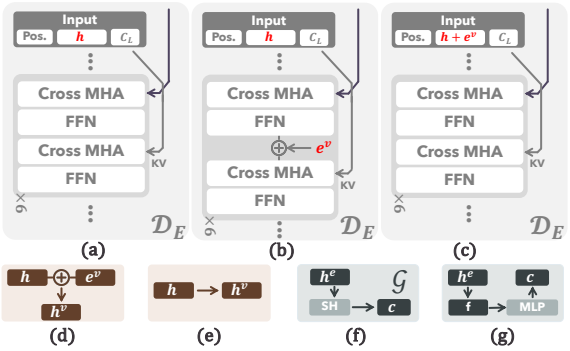


Figure 7. Different designs for the feedforward network $\mathcal{D}$.

Table 4. Performance Comparison of Different Architectures.

| Arch | PSNR | SSIM | LPIPS |
|---|---|---|---|
| a + e + f | 29.82 | 0.9472 | 0.0642 |
| a + e + g | 30.66 | 0.9524 | 0.0515 |
| a + d + g | 31.96 | 0.9597 | 0.0492 |
| b + d + g | 32.43 | 0.9628 | 0.0472 |
| c + d + g (ours) | 32.54 | 0.9649 | 0.0442 |

leads to a significant decline in relighting performance, consistent with [53]. Furthermore, Fig. 7 demonstrates that injecting camera view information to identify which lighting tokens should be attended to, *prior* to lighting baking, significantly enhances relighting results compared to baking global lighting first. This design allows for more effective capture of geometric and lighting variations, boosting the performance of $\mathcal{D}_E$ (Tab. 4).

**InputTypes**. We analyze the effect of different input latent representations on the decoder $\mathcal{D}$ in Tab. 3. LH-SLAT, which encodes rich and consistent lighting interaction information, outperforms both Base Color SLAT and Shaded SLAT ($Z_s$). The use of $Z_s$ complicates relighting due to the entanglement of unknown lighting. However, Base Color SLAT serves as a valuable complement to LH-SLAT; their combination yields the best performance.

## 5. Conclusion

We propose a compact multi-stage framework for relightable 3D generation, enabling consistent high-fidelity reconstruction and realistic relighting. Experiments show improved quantitative and perceptual results over strong baselines, and ablations confirm each component's contribution. Although evaluated on controlled captures with moderate compute, the approach suggests clear directions for in-the-wild and dynamic scenes and for efficiency and generalization improvements. We hope this work advances practical neural relighting and reconstruction.

**Acknowledgments**: This research was supported by the Beijing Natural Science Foundation (L244043), the Zhejiang Provincial Natural Science Foundation (LD24F020007).

## References


[1] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *CVPR*, 2013. 3

[2] Zoubin Bi, Yixin Zeng, Chong Zeng, Fan Pei, Xiang Feng, Kun Zhou, and Hongzhi Wu. Gs3: Efficient relighting with triple gaussian splatting. In *SIGGRAPH Asia*, 2024. 3, 5

[3] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. In *CVPR*, 2025. 2, 6, 7, 8, 3

[4] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *TVCG*, 2024. 1

[5] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *CVPR*, 2024. 2, 6, 7, 8, 1, 3

[6] Zilong Chen, Yikai Wang, Wenqiang Sun, Feng Wang, Yiwen Chen, and Huaping Liu. Meshgen: Generating pbr textured mesh with render-enhanced auto-encoder and generative data augmentation. In *CVPR*, 2025. 2, 6, 7, 8, 1, 3

[7] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024. 5

[8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 6

[9] Zhao Dong, Ka Chen, Zhaoyang Lv, Hong-Xing Yu, Yunzhi Zhang, Cheng Zhang, Yufeng Zhu, Stephen Tian, Zhengqin Li, Geordie Moffatt, et al. Digital twin catalog: A large-scale photorealistic 3d object digital twin dataset. In *CVPR*, 2025. 6, 7

[10] Andreas Engelhardt, Mark Boss, Vikram Voleti, Chun-Han Yao, Hendrik Lensch, and Varun Jampani. Svim3d: Stable video material diffusion for single image 3d generation. In *ICCV*, 2025. 3

[11] Frédéric Fortier-Chouinard, Zitian Zhang, Louis-Etienne Messier, Mathieu Garon, Anand Bhattad, and Jean-François Lalonde. Spotlight: Shadow-guided object relighting via diffusion. *arXiv:2411.18665*, 2024. 3

[12] 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. 3, 5

[13] Kai He, Ruofan Liang, Jacob Munkberg, Jon Hasselgren, Nandita Vijaykumar, Alexander Keller, Sanja Fidler, Igor Gilitschenski, Zan Gojcic, and Zian Wang. Unirelight: Learning joint decomposition and synthesis for video relighting. In *NeurIPS*, 2025. 6

[14] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *ICLR*, 2024. 3

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICCV*, 2022. 4, 1

[16] Haian Jin, Yuan Li, Fujun Luan, Yuanbo Xiangli, Sai Bi, Kai Zhang, Zexiang Xu, Jin Sun, and Noah Snavely. Neural gaffer: Relighting any object via diffusion. In *NeurIPS*, 2024. 2, 3, 6, 7

[17] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv:2312.03732*, 2023. 1

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 2023. 3, 4

[19] Hong Li, Houyuan Chen, Chongjie Ye, Zhaoxi Chen, Bohan Li, Shaocong Xu, Xianda Guo, Xuhui Liu, Yikai Wang, Baochang Zhang, Satoshi Ikehata, Boxin Shi, Anyi Rao, and Hao Zhao. Light of normals: Unified feature representation for universal photometric stereo. In *ICLR*, 2026. 5

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

[21] Zhong Li, Liangchen Song, Zhang Chen, Xiangyu Du, Lele Chen, Junsong Yuan, and Yi Xu. Relit-neulf: Efficient relighting and novel view synthesis via neural 4d light field. In *ACM MM*, 2023. 3

[22] Zhengqin Li, Dilin Wang, Ka Chen, Zhaoyang Lv, Thu Nguyen-Phuoc, Milim Lee, Jia-Bin Huang, Lei Xiao, Cheng Zhang, Yufeng Zhu, et al. Lirm: Large inverse rendering model for progressive reconstruction of shape, materials and view-dependent radiance fields. In *CVPR*, 2025. 3

[23] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. In *CVPR*, 2025. 2, 3, 6, 7, 1

[24] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *TOG*, 2023. 6, 7

[25] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 3

[26] Zexiang Liu, Yangguang Li, Youtian Lin, Xin Yu, Sida Peng, Yan-Pei Cao, Xiaojuan Qi, Xiaoshui Huang, Ding Liang, and Wanli Ouyang. Unidream: Unifying diffusion priors for relightable text-to-3d generation. In *ECCV*, 2024. 3

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[28] Nadav Magar, Amir Hertz, Eric Tabellion, Yael Pritch, Alex Rav-Acha, Ariel Shamir, and Yedid Hoshen. Lightlab: Controlling light sources in images with diffusion models. In *SIGGRAPH*, 2025. 2, 3

[29] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan.

PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022. 1

[30] 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. *CACM*, 2021. 6

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. In *TMLR*, 2023. 4

[32] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *CVPR*, 2023. 6, 7

[33] Y. Poirier-Ginter, A. Gauthier, J. Phillip, J.-F. Lalonde, and G. Drettakis. A diffusion approach to radiance field relighting using multi-illumination synthesis. *EGSR*, 2024. 3

[34] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 3

[35] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *CVPR*, 2024. 3

[36] Fabio Remondino, Ali Karami, Ziyang Yan, Gabriele Mazzacca, Simone Rigon, and Rongjun Qin. A critical analysis of nerf-based 3d reconstruction. *Remote Sensing*, 2023. 3

[37] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv:2407.08608*, 2024. 1

[38] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *ICLR*, 2024. 3

[39] Dongseok Shim, Yichun Shi, Kejie Li, H Jin Kim, and Peng Wang. Mvlight: Relightable text-to-3d generation via light-conditioned multi-view diffusion. *arXiv:2411.11475*, 2024. 3

[40] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024. 3

[41] Jiapeng Tang, Matthew Lavine, Dor Verbin, Stephan J Garbin, Matthias Nießner, Ricardo Martin Brualla, Pratul P Srinivasan, and Philipp Henzler. Rogr: Relightable 3d objects using generative relighting. In *NeurIPS*, 2025. 3

[42] Blender Development Team. Eevee release notes for blender 4.2. https://developer.blender.org/docs/release_notes/4.2/eevee/, 2025. 7

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 7

[44] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In *NeurIPS*, 2024. 2

[45] 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. 2, 3, 4, 5, 7, 1

[46] Ziyang Yan, Nazanin Padkan, Paweł Trybała, Elisa Mariarosaria Farella, and Fabio Remondino. Learning-based 3d reconstruction methods for non-collaborative surfaces—a metrological evaluation. *Metrology*, 2025. 3

[47] Ziyang Yan, Lei Li, Yihua Shao, Siyu Chen, Zongkai Wu, Jenq-Neng Hwang, Hao Zhao, and Fabio Remondino. 3dsceneeditor: Controllable 3d scene editing with gaussian splatting. In *WACV*, 2026. 3

[48] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. In *ICCV*, 2025. 2

[49] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *JMLR*, 2025. 1

[50] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *TOG*, 2024. 1

[51] Chong Zeng, Guojun Chen, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. Relighting neural radiance fields with shadow and highlight hints. In *SIGGRAPH*, 2023. 3

[52] Chong Zeng, Yue Dong, Pieter Peers, Youkang Kong, Hongzhi Wu, and Xin Tong. Dilightnet: Fine-grained lighting control for diffusion-based image generation. In *SIGGRAPH*, 2024. 2, 3, 6, 7

[53] Chong Zeng, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. Renderformer: Transformer-based neural rendering of triangle meshes with global illumination. In *SIGGRAPH*, 2025. 6, 8

[54] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. Rgb↔x: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *SIGGRAPH*, 2024. 2, 6, 7, 1, 3

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

[56] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *ICLR*, 2025. 3

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6, 7

[58] Tianyuan Zhang, Zhengfei Kuang, Haian Jin, Zexiang Xu, Sai Bi, Hao Tan, He Zhang, Yiwei Hu, Milos Hasan, William T Freeman, et al. Relitlrm: Generative relightable radiance for large reconstruction models. In *ICLR*, 2025. 3

[59] Xuying Zhang, Bo-Wen Yin, Yuming Chen, Zheng Lin, Yunheng Li, Qibin Hou, and Ming-Ming Cheng. Temo: Towards text-driven 3d stylization for multi-object meshes. In *CVPR*, 2024. 3

[60] Xuying Zhang, Yutong Liu, Yangguang Li, Renrui Zhang, Yufei Liu, Kai Wang, Wanli Ouyang, Zhiwei Xiong, Peng Gao, Qibin Hou, and Ming-Ming Cheng. Tar3d: Creating high-quality 3d assets via next-part prediction. In *ICCV*, 2025.

[61] Xuying Zhang, Yupeng Zhou, Kai Wang, Yikai Wang, Zhen Li, Shaohui Jiao, Daquan Zhou, Qibin Hou, and Ming-Ming Cheng. Ar-1-to-3: Single image to consistent 3d object via next-view prediction. In *ICCV*, 2025. 3

[62] Xiaoming Zhao, Pratul P. Srinivasan, Dor Verbin, Keunhong Park, Ricardo Martin Brualla, and Philipp Henzler. IllumiNeRF: 3D Relighting Without Inverse Rendering. In *NeurIPS*, 2024. 3

[63] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv:2501.12202*, 2025. 2, 3, 6, 7, 8, 1

[64] Jialin Zhu, Jiangbei Yue, Feixiang He, and He Wang. 3d student splatting and scooping. In *CVPR*, 2025. 6