LIRM: Large Inverse Rendering Model for Progressive Reconstruction of Shape, Materials and View-dependent Radiance Fields

Zhengqin Li[1] Dilin Wang[1] Ka Chen[1] Zhaoyang Lv[1] Thu Nguyen-Phuoc[1] Milim Lee[1] Jia-Bin Huang[12]
Lei Xiao[1] Yufeng Zhu[1] Carl S. Marshall[1] Yuheng Ren[1] Richard Newcombe[1] Zhao Dong[1]
[1]Meta Reality Labs [2]University of Maryland, College Park

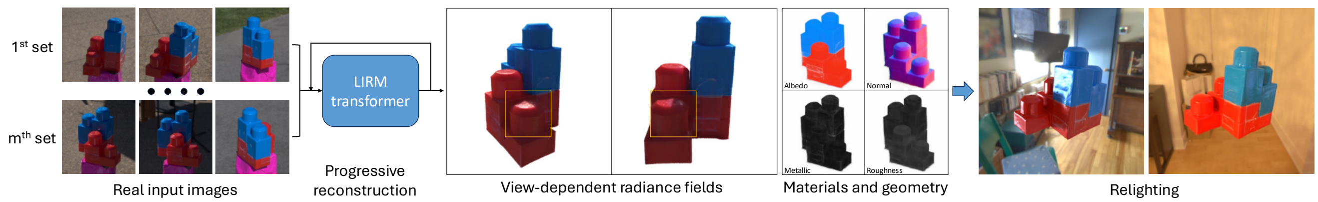

Figure 1. Given small sets of images (e.g., 4 to 8), LIRM progressively reconstructs view-dependent radiance fields, geometry and material reflectance in less than a second through a feed-forward transformer, enabling realistic rendering under novel lighting conditions.

Abstract

We present Large Inverse Rendering Model (LIRM), a transformer architecture that jointly reconstructs high-quality shape, materials, and radiance fields with view-dependent effects in less than a second. Our model builds upon the recent Large Reconstruction Models (LRMs) that achieve state-of-the-art sparse-view reconstruction quality. However, existing LRMs struggle to reconstruct unseen parts accurately and cannot recover glossy appearance or generate relightable 3D contents that can be consumed by standard Graphics engines. To address these limitations, we make three key technical contributions to build a more practical multi-view 3D reconstruction framework. First, we introduce an update model that allows us to progressively add more input views to improve our reconstruction. Second, we propose a hexa-plane neural SDF representation to better recover detailed textures, geometry and material parameters. Third, we develop a novel neural directional-embedding mechanism to handle view-dependent effects. Trained on a large-scale shape and material dataset with a tailored coarse-to-fine training scheme, our model achieves compelling results. It compares favorably to optimization-based dense-view inverse rendering methods in terms of geometry and relighting accuracy, while requiring only a fraction of the inference time.

1. Introduction

High-quality reconstruction of shape and materials from multi-view images, often referred to as inverse rendering, is a fundamental challenge in computer vision research. It has numerous important industrial applications in gaming, film, architecture, robotics, and AR/VR. Recent advancements in neural 3D representations [32, 53, 59, 60] and generative modeling [26, 70] have led to the development of more robust and efficient inverse rendering methods. These advancements help democratize 3D content creation. Modern inverse rendering methods [5–7, 19, 24, 29, 30, 46, 61, 68, 72, 91, 97, 98, 102–104] can jointly reconstruct shape, materials and even lighting from multiple images captured in arbitrary natural illuminations with reasonable accuracy. This is often achieved by minimizing a rendering loss through a carefully designed optimization process. This is a great step forward from traditional measurement-based methods [14, 56, 57] that require specialized devices and highly constrained environments. However, these optimization-based methods still suffer from long reconstruction time and require densely captured images. They also lack inductive bias to disambiguate lighting, materials and geometry, which often causes shadows and highlights to be baked into reconstructed materials.

We present Large Inverse Rendering Model (LIRM), the first feed-forward transformer that takes less than 1 second to jointly reconstruct high-quality shape, materials, and view-dependent radiance fields of a full 3D object. It can achieve this from as few as 3 to 6 posed images captured in arbitrary, unconstrained environments. Without increasing GPU memory consumption, LIRM can progressively refine its prediction results by incorporating additional input views, allowing for the addition of missing textures and specular highlights in previously unseen regions or from novel view angles. Inverse rendering from sparse inputs is an extremely ill-posed problem. State-of-the-art

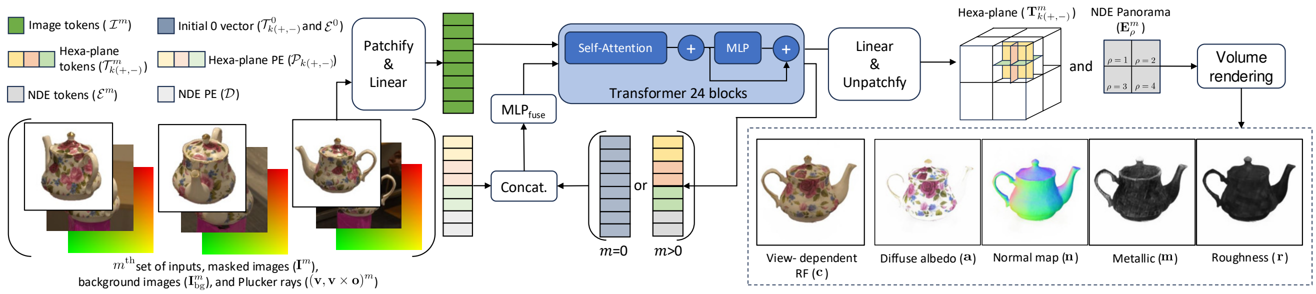

Figure 2. The network architecture of LIRM. The inputs are masked images $\mathbf{I}^m$, background images to provide more lighting information $\mathbf{I}^m_{\text{bg}}$ and Plücker rays $(\mathbf{v}, \mathbf{v} \times \mathbf{o})^m$ that encodes camera intrinsics and extrinsics. These 3 images are concatenated together and turned into tokens through a simple linear layer. These tokens are sent to a self-attention transformer to update hexa-plane tokens ($\mathcal{T}^m_{k(+,-)}$, $k \in \{xy, xz, yz\}$) and NDE tokens ($\mathcal{E}^m$). We decode the 2 kinds of tokens into hexa-plane representation and NDE panoramas through linear layers, which can be used to render view dependent radiance fields and BRDF parameters through neural volume rendering.

optimization-based methods often fail to fully decompose materials, lighting, and geometry with dense observations. Our work draws inspiration from recent Large Reconstruction Models (LRM) [27], which are trained on large-scale 3D datasets [15] and have achieved unprecedented high-quality sparse-view reconstruction results.

However, several drawbacks hinder their practical use in image-based 3D content creation frameworks. First, existing LRMs [25, 35, 75, 79, 84, 88, 89, 99] output radiance fields without any view-dependent effects, which fails to correctly model glossy appearances. Secondly, they face difficulties in reconstructing unseen parts of objects. Unfortunately, naively adding more input views poses challenges due to GPU memory limitations and model capacity constraints during both training and inference. More importantly, they lack the ability to fully decompose geometry, materials, and lighting, which prevents them from generating relightable 3D contents that can be consumed by standard graphic pipelines.

In this work, we present a more practical model for efficient and robust reconstruction of high-quality relightable 3D contents. It fully supports downstream applications such as rendering, simulation and editing using standard graphics pipelines. Our approach introduces several important novel technical components. Firstly, we develop a novel network module that enables progressive updates of inverse rendering results. This is achieved by comparing predicted 3D tokens with new input image tokens using self-attention modules. As a result, our model can refine its predictions as more input views are added without increasing GPU memory consumption. Secondly, we propose a novel hexa-plane neural SDF representation. Our hexa-plane representation better reconstructs texture details and memorizes prior reconstruction results without significantly increasing computational cost. Thirdly, we adopt neural directional encoding [85] into our feed-forward transformer architecture to recover view-dependent effects, which is essential for creating photorealistic appearances. Fourthly, we build a new large-scale 3D dataset with ground-truth materials and realistic appearances. This dataset carefully mimics real-world capturing settings to minimize domain gaps. We also made several improvements to further enhance reconstruction quality, including a higher-capacity model and an elaborate coarse-to-fine training paradigm that balances computational cost and quality. Experiments show that our model achieves competitive reconstruction results on real object benchmarks [34, 76], and even outperforms several recent dense-view optimization-based methods in terms of geometry and relighting accuracy.

2. Related Works

Inverse rendering Built on recent advancements in neural 3D representations [32, 53, 59, 60] and differentiable rendering [21, 37, 50, 96], latest optimization-based inverse rendering methods [5–7, 19, 24, 29, 30, 46, 61, 68, 72, 91, 97, 98, 102–104] can reconstruct geometry, materials, and lighting from images densely captured in natural and unknown illumination. This is achieved by minimizing a rendering loss, a process takes from several minutes to hours. However, due to its ill-posed nature and the lack of effective inductive priors, even state-of-the-art inverse rendering methods are prone to generate artifacts under challenging scenarios, such as the presence of shadows, strong specular highlights, and interreflections. Various priors have been designed to improve inverse rendering accuracy. Earlier methods [3, 4, 52] rely on hand-crafted regularizations. Recent learning-based methods increasingly rely on deep priors learned from large-scale real or synthetic datasets to solve challenging inverse rendering problems. These include single image intrinsic decomposition [9, 39, 58], SVBRDF estimation [17, 38, 40, 41], lighting estimation [22, 23, 42, 43, 71, 82], and relighting [44, 90]. However, none of the above methods can reconstruct fully relightable 3D objects from arbitrarily posed sparse inputs in a feed-forward manner.

Sparse-view reconstruction Numerous attempts have been made to inject regularization [28, 33, 62, 66] or learned priors [11, 31, 54, 81, 94] into sparse-view neural radiance field reconstruction. However, these methods generally rely on sophisticated optimization paradigms with various regularization terms, which can take several minutes or even hours to reconstruct a 3D object with quality much lower than dense-view reconstruction methods.

Recent research has facilitated the use of diffusion priors [26, 70] learned from billions of 2D images to significantly improve sparse-view reconstruction. These priors are usually applied either through score distillation sampling, which is usually time-consuming and unstable, [18, 47, 64, 78, 80, 83, 86] or by fine-tuning pre-trained diffusion models on 3D object datasets to generate multi-view consistent images directly [48, 49, 51, 55, 65, 67, 74, 87]. While these methods demonstrate an impressive ability to hallucinate visually appealing appearances, the reconstructed unseen parts do not always align with real objects. They also lack the knowledge to faithfully recover material reflectance. In contrast, LIRM offers a more explicit approach to controlling the reconstruction process by adding additional input views. It achieves inverse rendering accuracy comparable to state-of-the-art methods optimization-baesd, making it a highly effective solution.

Large reconstruction models LRM [27] and its variants [8, 25, 35, 73, 75, 79, 84, 88, 89, 99] demonstrate transformer architecture's exceptional capability for sparse reconstruction. Trained on large-scale 3D datasets [15, 16] and multi-view image datasets [95], they can reconstruct realistic geometry and texture details from very few or even a single image in a feed-forward manner within a second.

Most LRM variants focus on reconstructing radiance fields, which are incompatible with standard graphics pipeline for editing and visualization. While MeshLRM [84] and InstantMesh [88] extract mesh and texture maps, it remains a challenging problem to reconstruct relightable 3D objects with realistic specular highlights. Two concurrent works aim to solve this problem. RelitLRM [101] repurposes the LRM transformer as a neural renderer to directly predict view-dependent appearance under a new lighting condition. However, the output 3D Gaussian points have limitations, including a lack of support for near-field effects like area lighting and interreflections, as well as material and geometry editing applications. Similar to our method, SF3D [8] and AssetGen[69] target fully decomposing materials, geometry and lighting. Compared to concurrent works, LIRM features a simpler network design with larger capacity, supporting multi-view progressive reconstruction and achieving lower relighting errors on a popular benchmark [34].

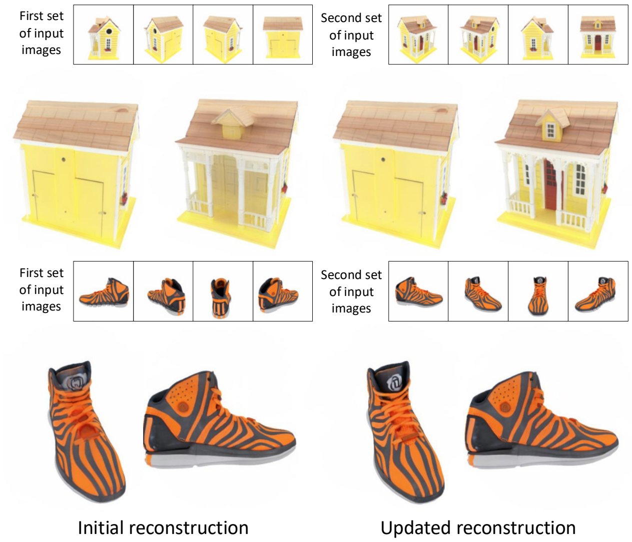

Figure 3. Visualization of the initial and updated reconstruction. Our simple update strategy enables our model to memorize prior reconstruction while progressively improve results.

3. Method

Our LIRM network architecture is demonstrated in Fig. 2. We start by introducing notations and preliminary knowledge of LRM. Then, we present our update module for progressive reconstruction and an improved neural SDF representation for more detailed shape and material reconstruction. Next, we explain how we handle view-dependent radiance fields. Finally, we summarize our tailored coarse-to-fine training scheme and all the implementation details.

3.1. Preliminaries

LIRM transformer architecture is built on MeshLRM [84] but with larger capacity. Compared to the original LRM method [27], MeshLRM makes two major improvements in its network design. First, it uses Plückers-rays representation instead of adaLN [63]. This enables better generalization to diverse camera settings and allows for cropping images to focus on the object, which is crucial for real-world applications with challenging camera settings (see Sec. 4). Second, it removes the DinoViT [10] encoder, which is difficult to train and results in reduced texture detail, likely due to its pre-training on semantic tasks. Instead, it uses a simple linear layer to tokenize image patches. More specifically, let $\{\mathbf{I}\}$ be a set of non-overlapping 8×8 image patches from multiple input views. Let $\{(\mathbf{v}, \mathbf{v}\times\mathbf{o})\}$ be the corresponding Plücker representation, where $\mathbf{v}$ is the ray direction and $\mathbf{o}$ is the camera original point. The image tokens are computed as

$$\{\mathcal{I}\} = \mathrm{Linear}(\{\mathbf{I}, (\mathbf{v}, \mathbf{v}\times\mathbf{o})\}). \quad (1)$$

MeshLRM adopts tri-plane as its 3D representation. Tri-plane is first tokenized with learned positional encoding

Figure 4. Comparisons between our tri-plane and hexa-plane reconstruction. Here we show diffuse texture reconstruction results without considering material reflection and view-dependent effects. Hexa-plane clearly recovers better texture details.

$\{\mathcal{P}_k\}, k \in \{xy, xz, yz\}$. These tri-plane tokens together with image tokens $\{\mathcal{I}\}$ are sent to a simple transformer architecture that consists of a series of pre-LN self-attention blocks. Both types of tokens are updated by every self-attention block but we only keep the final output tri-plane tokens,

$$\{\mathcal{T}_k\} = \text{Transformer}(\{\mathcal{P}_k\}, \{\mathcal{I}\}). \tag{2}$$

The predicted tri-plane tokens are then decoded by a simple linear layer. Each token is decoded to an 8×8 non-overlapping feature patch on the tri-plane,

$$\{\mathbf{T}_k\} = \text{Linear}(\{\mathcal{T}_k\}). \tag{3}$$

The decoded $\{\mathbf{T}_k\}, k \in \{xy, xz, yz\}$ is a standard tri-plane 3D representation, which can be used to render images either through volume ray marching or differentiable rasterization. In MeshLRM [84], the whole transformer is primarily trained with a rendering loss between rendered and ground-truth images, with several regularization terms to enforce geometry quality and training stability.

3.2. Update Module for Progressive Reconstruction

One unsolved challenge for existing LRMs is to reconstruct unseen parts of objects. Several attempts have been made to combine LRMs with multi-view diffusion models to hallucinate unseen appearance [35, 88]. However, the hallucinated appearance can not be guaranteed to align with the real object. Therefore, we argue that a more explicit way to control the reconstruction process is to enable users to interactively select more input views according to the current reconstruction result. Naively adding more images as inputs to the transformer will significantly increase the number of tokens, causing at least a linear increase of GPU memory consumption and close to quadratic increase of computation time, even with an advanced attention module [13]. On the contrary, our LIRM aims to support progressively adding an arbitrary number of images without increasing GPU memory consumption during both training and inference.

The key to achieve our goal is to feed the predicted tri-plane tokens $\{\mathcal{T}_k\}$ back into the transformer and update these tokens through self-attention with new input image tokens $\{\mathcal{I}\}$. In this process, the learnable positional encoding $\{\mathcal{P}_k\}$ are kept unchanged. We denote $\{\mathcal{T}_k^{m-1}\}$ as the $(m-1)^{\text{th}}$ set of output tri-plane tokens predicted by the transformer. We compute the m^{th} input tri-plane tokens $\{\mathcal{P}_k^m\}$ by concatenating the $(m-1)^{\text{th}}$ output tokens with the learned positional encoding $\{\mathcal{P}_k\}$ and pass it through a two-layer MLP. Eq. (1) and (2) are re-written as:

$$\{\mathcal{I}^m\} = \text{Linear}(\{\mathbf{I}^m, \mathbf{I}_{\text{bg}}^m, (\mathbf{v}, \mathbf{v} \times \mathbf{o})^m\}), \tag{4}$$

$$\{\mathcal{P}_k^m\} = \text{MLP}_{\text{fuse}}(\text{Concat}(\{\mathcal{T}_k^{m-1}\}, \{\mathcal{P}_k\})), \tag{5}$$

$$\{\mathcal{T}_k^m\} = \text{Transformer}(\{\mathcal{P}_k^m\}, \{\mathcal{I}^m\}), \tag{6}$$

where $\{\mathcal{T}_k^0\} = \{\mathbf{0}\}$. Since LIRM targets decomposing materials and lighting, we add images with background $\mathbf{I}_{\text{bg}}$ as an extra input to help the network figure out the lighting condition of surrounding environments. We observe that this simple modification is sufficient to enable us to effectively update the tri-plane prediction with new input images without "forgetting" the prior observations. Fig. 3 shows an example of our reconstruction results where the first set of 4 input images only cover the front side of the object and the second set of 4 images only cover the back side. With the first set of 4 input images, our LIRM only reconstructs the front side of the object accurately. After taking the second set of inputs, our network updates the tri-plane prediction to obtain high-quality reconstruction of the full 3D object.

3.3. Hexa-plane for Detailed Shape and Materials

While tri-plane-based 3D representation can achieve highly detailed appearance and geometry in most scenarios, we observe that it struggles when both sides of an object contain complex but different textures, as shown in Fig. 4. This limitation arises from using a single feature plane to represent both sides of textures in tri-plane representations. Therefore, we adopt a hexa-plane representation where we use 6 planes to divide the bounding box into 8 volumes, each with its own tri-plane. This representation utilizes the prior that the target object is likely to be roughly convex and located in the center of the 3D volume. K-plane [20] also uses multiple planes to represent 3D objects and scenes, but its primary goal is to model the temporal axis for dynamic scene reconstruction.

We now present our neural SDF representation based on hexa-plane for joint shape and materials reconstruction. We denote our hexa-plane as $\{\mathbf{T}_{k(+,-)}\}$. The corresponding output tokens and positional encoding are defined as $\{\mathcal{T}_{k(+,-)}\}$ and $\{\mathcal{P}_{k(+,-)}\}$, which can be replaced into Eq. (3), (5) and (6). To render images and material maps from our hexa-plane representation, we use the SDF-based volume ray marching method proposed in [92] for its simplicity and its ability to obtain high-quality geometry. We also considered [77] but it causes much higher computational cost, which will be discussed in the supplementary material. MeshLRM [84] and InstantMesh [88] use differentiable marching cube to reconstruct geometry and textures. However, that requires pre-training and special regularization terms to overcome training stability issues. Our training pipeline is much simpler while still showing promising geometry reconstruction and view synthesis results, as will be discussed in Sec. 4. Formally, let $\mathbf{x} = (x, y, z)$ be a 3D point. We can query feature $\mathbf{f}_{xy}(\mathbf{x})$ from plane $\{\mathbf{T}_{xy(+,-)}\}$ following,

$$\mathbf{f}_{xy}(\mathbf{x}) = \begin{cases} \text{Bilinear}(\mathbf{T}_{xy+}; x, y) & z \geq 0 \\ \text{Bilinear}(\mathbf{T}_{xy-}; x, y) & z < 0. \end{cases} \tag{7}$$

$\mathbf{f}_{xz}$ and $\mathbf{f}_{yz}$ can be queried similarly. $\mathbf{f}_{xy}$, $\mathbf{f}_{xz}$ and $\mathbf{f}_{yz}$ are concatenated as feature vector $\mathbf{f}$, which is sent to small MLPs with 2 to 3 hidden layers and 32 hidden dimensions for decoding, following [84]. We use separate MLPs to predict SDF value, RGB color ($\mathbf{c}$), normal ($\mathbf{n}$), albedo ($\mathbf{a}$), metallic ($\mathbf{m}$) and roughness ($\mathbf{r}$), written as,

$$\mathbf{s} = \text{MLP}_{\mathbf{s}}(\mathbf{f}) + \mathbf{s}_{\text{bias}}(\mathbf{x}) \tag{8}$$

$$\sigma = \begin{cases} \frac{1}{2}\exp(-\frac{\mathbf{s}}{\beta}) & s \geq 0 \\ 1 - \frac{1}{2}\exp(\frac{\mathbf{s}}{\beta}) & s < 0 \end{cases} \tag{9}$$

$$\mathbf{z} = \text{sigmoid}(\text{MLP}_{\mathbf{z}}(\mathbf{f})), \quad \mathbf{z} \in \{\mathbf{a}.\mathbf{c}, \mathbf{r}, \mathbf{m}\} \tag{10}$$

$$\mathbf{n} = \text{normalize}(\text{MLP}_{\mathbf{n}}(\mathbf{f})), \tag{11}$$

where σ is density and β is the standard deviation that controls sharpness of the reconstructed surface. We choose to gradually decrease β as will be discussed in Sec. 3.5. Meanwhile, $\mathbf{s}_{\text{bias}}$ is a prior that we find important to ensure fast convergence of the training loss. We set

$$\mathbf{s}_{\text{bias}}(\mathbf{x}) = ||\mathbf{x}|| - 0.1R, \tag{12}$$

where R is the radius of the bounding sphere.

Fig. 4 compares our hexa-plane and tri-plane reconstruction results. We reduced the number of tokens per-plane (from 64×64 to 48×48) so that both representations have similar computational cost. We can clearly see that hexa-plane recovers much better texture details, while tri-plane causes texture patterns to "leak" from one side to another, indicating a capacity limitation of tri-plane.

3.4. View-dependent Radiance Fields

View-dependent effect is important for modeling realistic appearance as glossy materials and specular highlights are commonly seen in daily objects. Moreover, neglecting view-dependent effects can negatively impact geometry quality, as the model will learn to create concave geometry to fake view-dependent appearance, as shown in [99]. It is challenging to recover view-dependent radiance fields from sparse observations, especially when the 3D object is highly glossy. This requires a holistic understanding of not only geometry and materials but also lighting and the surrounding environment. Although the primary focus of LIRM is on reconstructing materials and geometry, we also investigate methods for reconstructing view-dependent radiance fields.

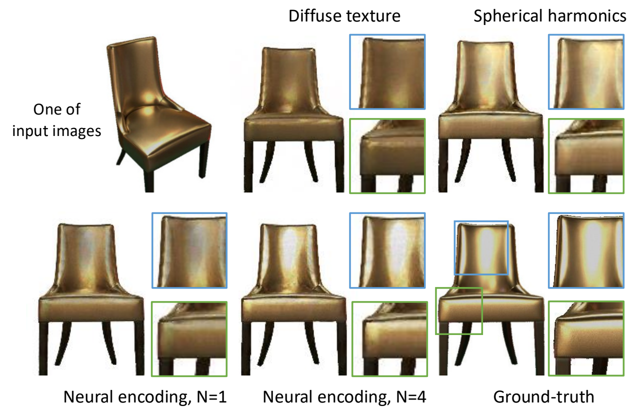

Figure 5. Comparisons of different strategies to model view-dependent effect in a feed-forward network module.

Simply adding the view direction as an extra input to the $\text{MLP}_{\mathbf{c}}$ cannot work because of its limited capacity. We therefore explore two solutions. We first consider predicting three orders of spherical harmonic coefficients for every query point, following [32] and [93]. However, this representation cannot handle high frequency angular signals, causing blurry specular highlights. To further improve the quality, we adopt the recent neural directional encoding (NDE) method [85] with modifications to make it compatible with a feed-forward transformer. NDE transfers the concept of feature volume into angular domain. It uses the reflection direction to query directional feature from a feature cubemap to model high frequency details in the angular domain. However, this feature cubemap alone cannot model near-field reflections, which makes a large impact when the target object is not fully convex. Wu et al. [85] solve this problem by computing second bounce features, which is too expensive for training a feed-forward model. We solve the problem by predicting multiple NDE panoramas and let the model learn which panorama to query feature from. We predict those NDE panoramas progressively through our transformer architecture. Let $\{\mathcal{D}\}$ be the learnable positional encoding for the NDE panorama and $\{\mathcal{E}^{m-1}\}$ be the output NDE panorama tokens from $(m-1)^{\text{th}}$ update. We fuse them together similar to Eq. (5) and send them to our self-attention transformer and decode the output tokens with a linear layer similar to Eq. (3):

$$\{\mathcal{D}^m\} = \text{MLP}_{\text{fuse}}(\text{Concat}(\{\mathcal{E}^{m-1}\}, \{\mathcal{D}\})), \tag{13}$$

$$\{\mathcal{E}^m\}, \{\mathcal{T}_k^m\} = \text{Transformer}(\{\mathcal{P}_k^m\}, \{\mathcal{D}^m\}, \{\mathcal{I}^m\}), \tag{14}$$

$$\{\mathbf{E}_\rho^m\} = \text{Linear}(\{\mathcal{E}^m\}) \tag{15}$$

where $\rho \in [1, N]$ is the index of NDE panoramas and N is the total number of NDE panoramas we use to approximate occlusion and near-field reflections.

To utilize these NDE panoramas for rendering, we add a small MLP to predict NDE panorama index ρ. Let $\mathbf{v}$ be the pixel ray direction and $\mathbf{l}$ be the reflection direction. RGB color $\mathbf{c}$ at a 3D point $\mathbf{x}$ can be written as:

$$\mathbf{n} = \text{normalize}(\text{MLP}_{\mathbf{n}}(\mathbf{f})), \tag{16}$$

$$\mathbf{l} = -2(\mathbf{v} \cdot \mathbf{n})\mathbf{n} + \mathbf{v} \tag{17}$$

$$\phi, \theta = \text{arctan2}(\mathbf{l}[0], \mathbf{l}[1]), \text{arccos}(\mathbf{l}[2]) \tag{18}$$

$$\rho = \text{sigmoid}(\text{MLP}_{\rho}(\mathbf{f})) \tag{19}$$

$$\mathbf{f}_d = \text{Trilinear}(\{\mathbf{E}^m_{\rho}\}_{\rho=1}^{N}; \rho, \theta, \phi) \tag{20}$$

$$\mathbf{c} = \text{MLP}_{\mathbf{c}}(\text{Concat}(\mathbf{f}, \mathbf{f}_d)) \tag{21}$$

In Fig. 5, we compare the representation power of spherical harmonics and neural directional encoding in a feed-forward setting by overfitting LIRM to a single 3D object. For neural directional encoding experiments, we test $N = 1$ and $N = 4$. Spherical harmonics solution misses sharp specular highlights. NDE solution fails at concave regions when $N = 1$ because we only model a single bounce. On the contrary, multiple panoramas ($N = 4$) achieves the lowest reconstruction error and reconstructs accurate specular highlights for the whole object.

3.5. Training Scheme

Coarse-to-fine training and acceleration We adopt a coarse-to-fine training scheme to ensure fast convergence and reduce computational cost. In the first stage, we use a larger batch size and learning rate, but smaller resolutions and fewer samples per ray to obtain a coarse reconstruction result. In the second and third stages, we fine-tune the model by decreasing learning rates and batch sizes, while increasing resolutions and samples per ray to refine reconstruction details. We also gradually reduce the standard deviation β by increasing $\frac{1}{\beta}$ following a linear schedule. Moreover, we increase the number of tokens for our tri-plane and hexa-plane representations compared to the prior state-of-the-art [84], which we rise from $32 \times 32 \times 3$ to $64 \times 64 \times 3$ and $48 \times 48 \times 6$ respectively. This makes training significantly slower at the third stage. Therefore, we incorporate occupancy grid acceleration from NerfAcc [36] into our differentiable renderer. Our final output plane resolutions are 512×512 and 384×384. For NDE panoramas, we add another 32×32 tokens into our transfomer and decode them into 4 feature panoramas of resolution 128×128. During training, we provide the update model with 2 sets of 3-6 images per iteration in the first stage, and 3 sets of images per iteration in the second and third stages. More details are in the supplementary material.

Loss functions LIRM is trained with image losses computed from direct supervision of ground-truth 2D RGB image, normal and material maps. We use L_2 loss for all rendered 2D maps and add LPIPS [100] loss for RGB image

Table 1. Quantitative comparisons for view synthesis under uniform lighting on the **GSO** dataset

Radiance fields	PSNR (↑)	SSIM (↑)	LPIPS (↓)
MeshLRM [84]	28.13	0.923	0.093
GS-LRM [99]	30.52	0.952	0.050
LIRM-hexa 1st	29.27	0.941	0.061
LIRM-hexa 2nd	30.48	0.947	0.056
LIRM-hexa 3rd	30.65	0.949	0.054
LIRM-hexa 4th	30.56	0.948	0.054
LIRM-tri 4th	29.61	0.941	0.063
Mesh	**PSNR (↑)**	**SSIM (↑)**	**LPIPS (↓)**
MeshLRM [84]	27.93	0.925	0.081
LIRM-hexa 4th	29.22	0.942	0.059

Table 2. Quantitative comparisons for view synthesis under uniform lighting on the **ABO** dataset.

Radiance fields	PSNR (↑)	SSIM (↑)	LPIPS (↓)
MeshLRM [84]	28.31	0.906	0.108
GS-LRM [99]	29.59	0.944	0.051
LIRM-hexa 1st	32.69	0.957	0.056
LIRM-hexa 2nd	33.08	0.959	0.050
LIRM-hexa 3rd	32.98	0.958	0.055
LIRM-hexa 4th	32.83	0.958	0.054
LIRM-tri 4th	32.58	0.956	0.055
Mesh	**PSNR (↑)**	**SSIM (↑)**	**LPIPS**
LIRM-hexa 4th	29.79	0.951	0.069

and albedo, which we find essential to recover texture details. We also find that an L_2 loss on numerical normal computed from SDF gradients can help improve geometry accuracy. To compute the numerical normal, we perturb every 3D point in axis-aligned directions (i.e., along the x, y, and z axes) and then calculate the normalized gradients at each point, which serve as the numerical normal. We set the size of perturbation to be twice the size of a voxel in hexa-plane, following [45]. However, this would require 3 times more feature query to compute numerical normal loss, which is expensive. We therefore only use it for the third stage of training, which we find to be sufficient.

More implementation details LIRM's transformer consists of 24 self-attention blocks, each with 16 heads and a feature dimension of 1024, where each head has a separate feature dimension of 64. Since LIRM's hexa-plane output both 2D images and material parameters maps, we increase the number of feature channels of $\{\mathbf{T}_k\}$ from 32 to 64 compared to MeshLRM [84]. We use AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.95)$. The whole training takes around 2 weeks on 64 H100 GPUs. The inference time for one step is around 0.3 seconds on an A100 GPU.

4. Experiments

Training data We create a new dataset that carefully mimics real capturing environments to reduce domain gaps.

Table 3. Quantitative comparisons for view synthesis and inverse rendering under environment lighting on the **ABO** dataset.

	PSNR (↑)				LPIPS (↓)		CD (↓)
LIRM	c	a	r	m	c	a	-
NDE 1st	29.01	32.34	23.20	27.28	0.064	0.070	0.123
NDE 2nd	29.70	32.82	23.30	28.16	0.060	0.067	0.122
NDE 3rd	29.82	32.85	23.36	28.29	0.060	0.067	0.121
NDE 4th	29.92	32.76	23.32	28.34	0.060	0.067	0.121
Diff 4th	29.50	32.95	23.05	28.14	0.061	0.067	0.121

Table 4. Quantitative comparisons for view synthesis and inverse rendering under environment lighting on the **DTC** dataset.

	PSNR (↑)				LPIPS (↓)		CD (↓)
LIRM	c	a	r	m	c	a	-
NDE 1st	27..59	30.19	18.71	26.97	0.092	0.096	0.119
NDE 2nd	28.97	31.23	18.79	28.85	0.082	0.088	0.118
NDE 3rd	29.23	31.50	18.88	29.24	0.081	0.086	0.117
NDE 4th	29.35	31.54	18.84	29.33	0.080	0.085	0.117
Diff 4th	28.98	31.60	18.44	29.12	0.081	0.086	0.117

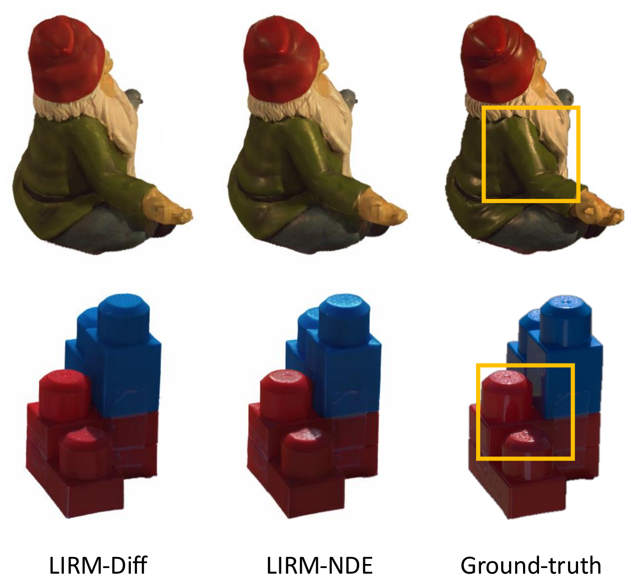

Figure 6. LIRM-NDE can generalize to real images to recover view-dependent radiance fields.

We select 600k 3D objects from the Shutterstock dataset [1] with GT PBR materials. To render these 3D objects, we use 2.5K HDR environments collected from Laval Dataset and Polyhaven. Each HDR environment map is randomly rotated before rendering. We also randomly perturb the exposure time, white balance, camera intrinsics, and material parameters for data augmentation. Our final training set includes 38 million images with GT material maps.

View synthesis under uniform lighting We first train and test the LIRM model on datasets rendered with uniform lighting. This enables us to validate the effectiveness of our update module and hexa-plane representation. It also allows us to directly compare with state-of-the-art LRM variants. We tested on GSO and ABO datasets. Quantitative comparisons are summarized in Tab. 1 and Tab. 2. Qualitative results can be seen in Fig. 3 and Fig. 4. We tested 4 stages of update. The input and output views are selected following [84]. We select 16 input views and 12 output views. For each stage, we randomly select 4 views from 16 input views. This is a more challenging setting compared to [84] as it always uses canonical views as inputs. Nevertheless, both LIRM tri-plane and LIRM hexa-plane outperforms the prior stage-of-the-art volume-based LRM method [84]. LIRM hexa-plane even outperforms the baseline with the first 4 input views, thanks to our larger number of plane tokens and the novel representation. We observe that our update model is the most effective within the first 3 sets of input views. LIRM hexa-plane achieves lower error compared to LIRM tri-plane but the gap is much smaller on ABO datasets, possibly because it has simpler textures.

Inverse rendering and view synthesis under natural lighting We first evaluate our LIRM on synthetic datasets. We use ABO [12] and newly released DTC [2] datasets. Both datasets contain 3D models with high-quality material properties. We test two variants, LIRM-Diff only predict diffuse texture and material parameters. LIRM-NDE also predicts NDE panoramas to model view-dependent effects. The view selection is the same as previous experiments. Quantitative results are summarized in Tab. 3 and Tab. 4. From quantitative results, we observe that LIRM-NDE consistently achieve lower view synthesis errors compared to LIRM-Diff, indicating that our NDE module can successfully model view-dependent effects. Geometry and material quality of the 2 models are similar. Similarly, LIRM-Diff and LIRM-NDE can improve reconstruction quality with more inputs. Fig. 7 shows that our material prediction can accurately match the ground-truth, even for spatially varying roughness values (first row), leading to highly realistic object relighting results (last 2 columns).

We then test both two variants on a real dataset, Stanford-ORB [34]. Without any fine-tuning, LIRM generalizes impressively to real data. It achieves reconstruction quality on par and even better than state-of-the-art optimization-based methods, which takes dense views as inputs and several hours to run. Even with one set of inputs, LIRM achieves the second best relighting quality. With 3 sets of inputs, it achieves the highest geometry reconstruction quality. Fig. 8 shows LIRM can better handle specular materials compared to optimization-based methods. Fig. 6 shows that with real inputs, LIRM can also effectively handle view-dependent effects. We also compare to concurrent LRM-based inverse rendering method [69], and show that LIRM outperforms by a large margin both qualitatively and quantitatively.

Table 5. Quantitative comparisons with methods on Stanford-ORB dataset [34] for relighting, view synthesis and geometry reconstruction. We separate methods into predictive and optimization-based methods. We select 3 top optimization-based methods from the leaderboard.

Methods	Relighting				View Synthesis				Shape
	PSNR-H (↑)	PSNR-L (↑)	SSIM (↑)	LPIPS (↓)	PSNR-H (↑)	PSNR-L (↑)	SSIM (↑)	LPIPS (↓)	CD (↓)
InvRender [103]	23.76	30.83	0.970	0.046	25.91	34.01	0.977	0.042	0.44
NVDiffrecMc [24]	24.43	31.60	0.972	0.036	28.03	36.40	0.982	0.028	0.51
Neural-PBIR [72]	26.01	33.26	0.979	0.023	28.82	36.80	0.986	0.019	0.43
MetaLRM [69]	21.46	28.00	0.956	0.045	19.93	26.20	0.956	0.042	-
LIRM-diff 1st	24.76	32.11	0.971	0.027	25.82	34.01	0.977	0.021	0.48
LIRM-diff 3rd	25.09	32.45	0.972	0.025	26.66	34.88	0.979	0.018	0.38
LIRM-NDE 1st	24.25	31.63	0.969	0.028	25.84	34.00	0.976	0.021	0.33
LIRM-NDE 3nd	24.60	32.05	0.971	0.025	27.03	35.26	0.979	0.018	0.31

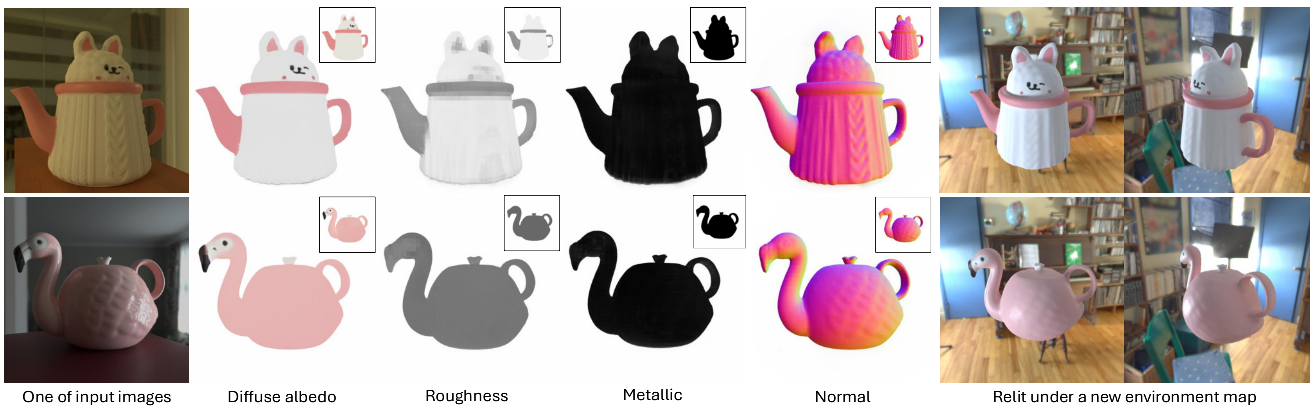

Figure 7. Inverse rendering results on DTC dataset. Material ground-truth are included in insets.

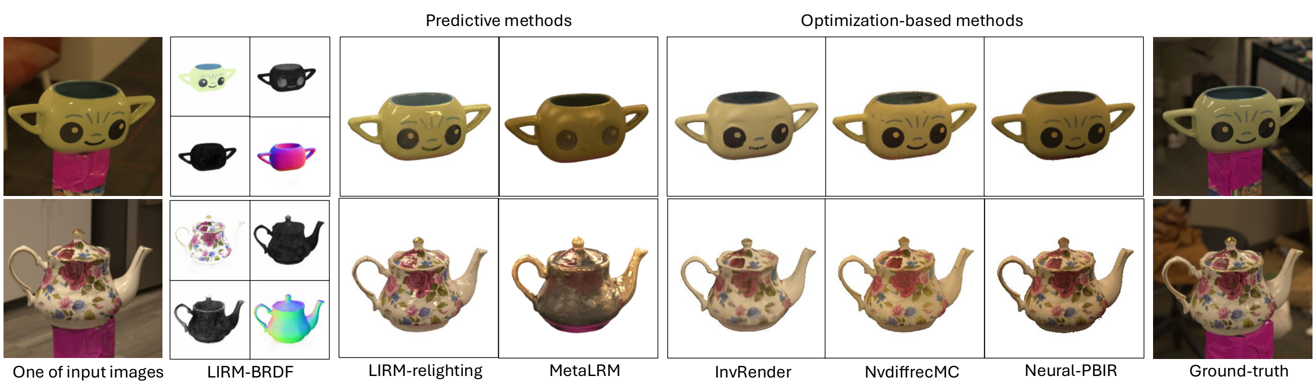

Figure 8. Comparisons with prior works on Stanford-ORB dataset.

5. Conclusion

We present LIRM, a Large Inverse Rendering Model that rapidly reconstructs high-quality shape, materials, and radiance fields with view-dependent effects from sparse inputs in under one second. LIRM overcomes the limitations of existing LRMs by introducing three key technical contributions: an update model for progressive reconstruction, a hexa-plane neural SDF representation for detailed texture recovery, and a novel neural directional encoding mechanism for view-dependent effects. Trained on a large-scale dataset in a coarse-to-fine manner, LIRM delivers results comparable to optimization-based methods, while significantly reducing inference time.

Supplementary: We will include more qualitative results and quantitative comparisons with a MeshLRM [84] baseline and optimization-based methods. We will also add implementation details, discuss limitations and advantages of LIRM under more challenging settings.

Acknowledgements We thank Yawar Siddiqui, Jesus Zarzar and David Novotny for providing baseline comparisons with MetaLRM [69]. We thank Yunzhi Zhang, Hong-Xing Yu, Guangyan Cai and Chen Sun for supporting experiments on the Stanford-ORB dataset [34].

References

[1] Shutterstock. https://www.shutterstock.com/search/3d. 7

[2] Digital twin catalog. https://www.projectaria.com/datasets/dtc/, 2024. 7

[3] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013. 2

[4] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2014. 2

[5] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 1, 2

[6] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34: 10691–10704, 2021.

[7] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan Barron, Hendrik Lensch, and Varun Jampani. Samurai: Shape and material from unconstrained real-world arbitrary image collections. *Advances in Neural Information Processing Systems*, 35:26389–26403, 2022. 1, 2

[8] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. *arXiv preprint*, 2024. 3

[9] Chris Careaga and Yağız Aksoy. Intrinsic image decomposition via ordinal shading. *ACM Transactions on Graphics*, 43(1):1–24, 2023. 2

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[11] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 3

[12] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022. 7

[13] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 4

[14] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156, 2000. 1

[15] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2, 3

[16] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[17] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (ToG)*, 37(4):1–15, 2018. 2

[18] Sam Earle, Filippos Kokkinos, Yuhe Nie, Julian Togelius, and Roberta Raileanu. Dreamcraft: Text-guided generation of functional 3d environments in minecraft. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, pages 1–15, 2024. 3

[19] Andreas Engelhardt, Amit Raj, Mark Boss, Yunzhi Zhang, Abhishek Kar, Yuanzhen Li, Deqing Sun, Ricardo Martin Brualla, Jonathan T Barron, Hendrik Lensch, et al. Shinobi: Shape and illumination using neural object decomposition via brdf optimization in-the-wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19636–19646, 2024. 1, 2

[20] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 4

[21] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A pytorch library for accelerating 3d deep learning research. https://github.com/NVIDIAGameWorks/kaolin, 2022. 2

[22] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090*, 2017. 2

[23] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In *Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition, pages 6908–6917, 2019. 2

[24] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light, and material decomposition from images using monte carlo rendering and denoising. *Advances in Neural Information Processing Systems*, 35:22856–22869, 2022. 1, 2, 8

[25] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models. `https://github.com/3DTopia/OpenLRM`, 2023. 2, 3

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3

[27] 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. *arXiv preprint arXiv:2311.04400*, 2023. 2, 3

[28] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 3

[29] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024. 1, 2

[30] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023. 1, 2

[31] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18365–18375, 2022. 3

[32] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 5

[33] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022. 3

[34] Zhengfei Kuang, Yunzhi Zhang, Hong-Xing Yu, Samir Agarwala, Elliott Wu, Jiajun Wu, et al. Stanford-orb: a real-world 3d object inverse rendering benchmark. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3, 7, 8

[35] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023. 2, 3, 4

[36] Ruilong Li, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: A general nerf acceleration toolbox. *arXiv preprint arXiv:2210.04847*, 2022. 6

[37] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37 (6):1–11, 2018. 2

[38] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Transactions on Graphics (ToG)*, 36(4):1–11, 2017. 2

[39] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *Proceedings of the European conference on computer vision (ECCV)*, pages 371–387, 2018. 2

[40] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: Svbrdf acquisition with a single mobile phone image. In *Proceedings of the European conference on computer vision (ECCV)*, pages 72–87, 2018. 2

[41] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 2

[42] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020. 2

[43] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, et al. Openrooms: An end-to-end open framework for photorealistic indoor scene datasets. *arXiv preprint arXiv:2007.12868*, 2020. 2

[44] Zhengqin Li, Jia Shi, Sai Bi, Rui Zhu, Kalyan Sunkavalli, Miloš Hašan, Zexiang Xu, Ravi Ramamoorthi, and Manmohan Chandraker. Physically-based editing of indoor scene lighting from a single image. In *European Conference on Computer Vision*, pages 555–572. Springer, 2022. 2

[45] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 6

[46] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21644–21653, 2024. 1, 2

[47] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 3

[48] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10083, 2024. 3

[49] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 3

[50] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7708–7717, 2019. 2

[51] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 3

[52] Stephen Lombardi and Ko Nishino. Reflectance and illumination recovery in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):129–141, 2015. 2

[53] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 1, 2

[54] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, pages 210–227. Springer, 2022. 3

[55] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9970–9980, 2024. 3

[56] Stephen R Marschner, Stephen H Westin, Eric PF Lafortune, Kenneth E Torrance, and Donald P Greenberg. Image-based brdf measurement including human skin. In *Rendering Techniques' 99: Proceedings of the Eurographics Workshop in Granada, Spain, June 21–23, 1999 10*, pages 131–144. Springer, 1999. 1

[57] Wojciech Matusik. *A data-driven reflectance model*. PhD thesis, Massachusetts Institute of Technology, 2003. 1

[58] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6315–6324, 2018. 2

[59] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[60] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 1, 2

[61] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 1, 2

[62] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 3

[63] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 3

[64] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 3

[65] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 3

[66] Ruoxi Shi, Xinyue Wei, Cheng Wang, and Hao Su. Zerorf: Fast sparse view 360deg reconstruction with zero pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21114–21124, 2024. 3

[67] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 3

[68] Yahao Shi, Yanmin Wu, Chenming Wu, Xing Liu, Chen Zhao, Haocheng Feng, Jingtuo Liu, Liangjun Zhang, Jian Zhang, Bin Zhou, et al. Gir: 3d gaussian inverse rendering for relightable scene factorization. *arXiv preprint arXiv:2312.05133*, 2023. 1, 2

[69] Yawar Siddiqui, Tom Monnier, Filippos Kokkinos, Mahendra Kariya, Yanir Kleiman, Emilien Garreau, Oran Gafni, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, et al. Meta 3d assetgen: Text-to-mesh generation with high-quality geometry, texture, and pbr materials. *arXiv preprint arXiv:2407.02445*, 2024. 3, 7, 8

[70] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 3

[71] Pratul P Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T Barron, Richard Tucker, and Noah Snavely. Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8080–8089, 2020. 2

[72] Cheng Sun, Guangyan Cai, Zhengqin Li, Kai Yan, Cheng Zhang, Carl Marshall, Jia-Bin Huang, Shuang Zhao, and Zhao Dong. Neural-pbir reconstruction of shape, material, and illumination. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18046–18056, 2023. 1, 2, 8

[73] Shitao Tang, Jiacheng Chen, Dilin Wang, Chengzhou Tang, Fuyang Zhang, Yuchen Fan, Vikas Chandra, Yasutaka Furukawa, and Rakesh Ranjan. Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 175–191. Springer, 2025. 3

[74] Shitao Tang, Jiacheng Chen, Dilin Wang, Chengzhou Tang, Fuyang Zhang, Yuchen Fan, Vikas Chandra, Yasutaka Furukawa, and Rakesh Ranjan. Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 175–191. Springer, 2025. 3

[75] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 2, 3

[76] Benjamin Ummenhofer, Sanskar Agrawal, Rene Sepulveda, Yixing Lao, Kai Zhang, Tianhang Cheng, Stephan Richter, Shenlong Wang, and German Ros. Objects with lighting: A real-world dataset for evaluating reconstruction and rendering for object relighting. In *2024 International Conference on 3D Vision (3DV)*, pages 137–147. IEEE, 2024. 2

[77] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 4

[78] Peihao Wang, Zhiwen Fan, Dejia Xu, Dilin Wang, Sreyas Mohan, Forrest Iandola, Rakesh Ranjan, Yilei Li, Qiang Liu, Zhangyang Wang, et al. Steindreamer: Variance reduction for text-to-3d score distillation via stein identity. *arXiv preprint arXiv:2401.00604*, 2023. 3

[79] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023. 2, 3

[80] Peihao Wang, Dejia Xu, Zhiwen Fan, Dilin Wang, Sreyas Mohan, Forrest Iandola, Rakesh Ranjan, Yilei Li, Qiang Liu, Zhangyang Wang, et al. Taming mode collapse in score distillation for text-to-3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9037–9047, 2024. 3

[81] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021. 3

[82] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning indoor inverse rendering with 3d spatially-varying lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12538–12547, 2021. 2

[83] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[84] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality mesh. *arXiv preprint arXiv:2404.12385*, 2024. 2, 3, 4, 5, 6, 7, 8

[85] Liwen Wu, Sai Bi, Zexiang Xu, Fujun Luan, Kai Zhang, Iliyan Georgiev, Kalyan Sunkavalli, and Ravi Ramamoorthi. Neural directional encoding for efficient and accurate view-dependent appearance modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21157–21166, 2024. 2, 5

[86] Tong Wu, Zhibing Li, Shuai Yang, Pan Zhang, Xingang Pan, Jiaqi Wang, Dahua Lin, and Ziwei Liu. Hyperdreamer: Hyper-realistic 3d content generation and editing from a single image. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023. 3

[87] Chao Xu, Ang Li, Linghao Chen, Yulin Liu, Ruoxi Shi, Hao Su, and Minghua Liu. Sparp: Fast 3d object reconstruction and pose estimation from sparse views. *arXiv preprint arXiv:2408.10195*, 2024. 3

[88] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 2, 3, 4, 5

[89] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023. 2, 3

[90] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)*, 37 (4):1–13, 2018. 2

[91] Ziyi Yang, Yanzhen Chen, Xinyu Gao, Yazhen Yuan, Yu Wu, Xiaowei Zhou, and Xiaogang Jin. Sire-ir: Inverse rendering for brdf reconstruction with shadow and illumination removal in high-illuminance scenes. *arXiv preprint arXiv:2310.13030*, 2023. 1, 2

[92] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 4

[93] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 5

[94] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or

few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 3

[95] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023. 3

[96] Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. A differential theory of radiative transfer. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019. 2

[97] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 1, 2

[98] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5565–5574, 2022. 1, 2

[99] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2025. 2, 3, 5, 6

[100] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[101] 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. *arXiv preprint arXiv:2410.06231*, 2024. 3

[102] 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 (ToG)*, 40(6):1–18, 2021. 1, 2

[103] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022. 8

[104] Youjia Zhang, Teng Xu, Junqing Yu, Yuteng Ye, Yanqing Jing, Junle Wang, Jingyi Yu, and Wei Yang. Nemf: Inverse volume rendering with neural microflake field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22919–22929, 2023. 1, 2