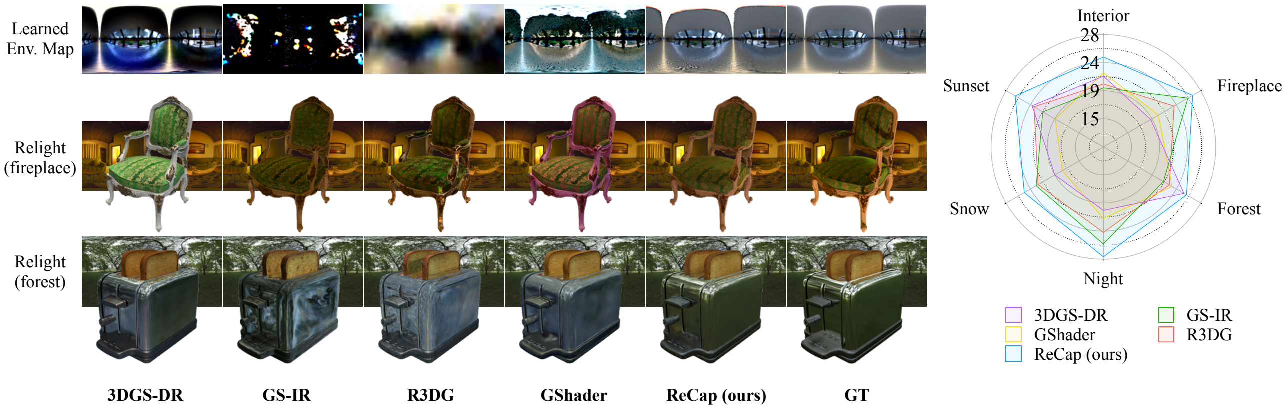
ReCap: Better Gaussian Relighting with Cross-Environment Captures

Jingzhi Li   Zongwei Wu[*]   Eduard Zamfir   Radu Timofte
Computer Vision Lab, CAIDAS & IFI, University of Würzburg, Germany



(a) Visual comparison of learned environment maps and relighting quality.

(b) Average relighting PSNR across unseen environment maps.

Figure 1. ReCap shows a significant advantage in reconstructing environment maps with accurate tones and color fidelity. Both qualitative and quantitative assessments show that ReCap achieves more realistic and consistent relighting results under a range of unseen lighting conditions.

Abstract

Accurate 3D objects relighting in diverse unseen environments is crucial for realistic virtual object placement. Due to the albedo-lighting ambiguity, existing methods often fall short in producing faithful relights. Without proper constraints, observed training views can be explained by numerous combinations of lighting and material attributes, lacking physical correspondence with the actual environment maps used for relighting. In this work, we present ReCap, treating cross-environment captures as multi-task target to provide the missing supervision that cuts through the entanglement. Specifically, ReCap jointly optimizes multiple lighting representations that share a common set of material attributes. This naturally harmonizes a coherent set of lighting representations around the mutual material attributes, exploiting commonalities and differences across varied object appearances. Such coherence enables physically sound lighting reconstruction and robust material estimation — both essential for accurate relighting. Together with a streamlined shading function and effective post-processing, ReCap outperforms all leading competitors on an expanded relighting benchmark.

1. Introduction

For a realistic and immersive augmented reality experience, virtually placed objects must convincingly reflect light, cast shadows, and adapt naturally to different lighting conditions. Achieving the desired realism requires a physically accurate response to environment lighting, driving a line of research at enabling relighting capabilities in popular neural representation models.

In recent years, Neural Radiance Field (NeRF) [28] gained prevailing popularity as an *implicit* scene representation. Subsequent NeRF-based relighting methods [19, 39, 50] produced impressive relighting results, but their computational demands make them impractical for interactive applications [18, 46]. Lately, 3D Gaussian Splatting (3DGS) [22] is widely acclaimed as an *explicit* 3D representation model for its high rendering quality and interactive frame rates, naturally suited for applications requiring real-time performance. Building on this, follow-up works [15, 18, 26, 46] have enabled relighting of Gaussians using explicit shading functions and learnable lighting representations, often in the form of environment maps.

While standard HDR maps could theoretically replace these learned environment maps for relighting, directly substituting them, as current methods do, remains questionable due to the unclear physical meaning of the learned

[*] Corresponding author

values. Because of the *albedo-lighting ambiguity* [1, 5], where changes in surface albedo are indistinguishable from changes in lighting intensity, existing supervision from the reconstruction loss alone is not enough for a truthful lighting reconstruction. As shown in Fig. 1a), the learned environments are often observed to be tinted with object colors, shifted in tone, scaled in intensity or filled with noise. Without proper constraints, these maps act as sinks for unmodeled residual terms during optimization, becoming indispensable for producing high-quality novel view synthesis results. Replacing them with the ground truth HDR maps significantly degrades the output quality, much less when attempting to relight with novel environment maps.

Inspired by photometric appearance modeling [25, 33, 45] which estimates surface properties from object appearances under varied lighting, we introduce additional photometric supervision to address the albedo-lighting ambiguity. While traditional approaches rely on controlled lighting setups like light stages [11] and collocated lights [44] to provide supervision through known light directions and/or intensities, they require dedicated hardwares. Instead, we propose ReCap to leverage object captures across unknown lighting conditions, modeling light-dependent appearances with multiple environment maps that share a common Gaussian model. Conceptually, this resembles multi-task learning, where the learned environment maps act as task heads querying a shared material representation for varied object appearances. In this case, the "querying" is done by the physically-based shading function, which facilitates the separation of material and lighting. This joint optimization promotes internal consistency when accounting for varied object appearances across diverse environments, simultaneously supporting more accurate lighting reconstruction.

To streamline the joint optimization, we introduce a generalized shading function based on the split-sum approximation [21], which eliminates a material parameter that introduces ambiguity in inverse rendering. Additionally, we ensure compatibility with standard HDR maps by encouraging the learned values to remain in a linear HDR space through appropriate post-processing. This enables the direct application of novel environment maps without the need for image normalization [27] or map adjustments [26].

Current relighting evaluations are limited in scope, focusing primarily on diffuse surfaces [19, 51]. For a more comprehensive assessment, we re-rendered 13 objects from NeRF [28] and RefNeRF [40] featuring both diffuse and specular surfaces. As shown in Fig. 1, experiments on the expanded benchmark confirm the effectiveness of ReCap in producing robust and more realistic relighting results. Codes and datasets are available here.

To summarize, the contribution of this work includes

- Treating cross-environment captures as multi-task targets, we address the albedo-lighting ambiguity via the joint optimization of shared material properties and independent lighting representations.
- We propose a novel shading function with physically appropriate post-processing, providing more flexible material representation that eases optimization and allows the direct application of standard HDR maps.
- ReCap achieves state-of-the-art relighting performance on a more comprehensive benchmark, showing robustness across diverse lighting and object types.

2. Related Work

Novel View Synthesis (NVS). Pure NVS techniques focus on reproducing the scene appearance under its original environment. NeRF [28] and its follow-ups [2–4] achieve remarkable NVS quality with implicit radiance fields and volume rendering. Although great progress has been made in NeRF acceleration [14, 29, 43] leveraging voxel grids or hash tables, their rendering speeds ($\sim$10 fps) are still far from interactive. Building on progress in differentiable point-based rendering [23, 47], 3DGS [22] recently introduced an explicit Gaussian representation with an efficient tile-based rasterizer, delivering photo-realistic rendering at impressive frame rates ($\sim$100 fps). This makes 3DGS well-suited for high-quality real-time relighting.

Relighting. The relighting task traditionally involves editing lighting within fixed views [12, 36, 44]. The term has since broadened to encompass novel view relighting [20, 45], which aims to generate images from both new viewpoints and under new lighting conditions. With the advent of differentiable neural rendering, reconstruction-based methods [7, 39, 48, 50] have found a promising direction to combine physically-based rendering (PBR) with neural representations for editable and realistic rendering. Closely related to our work, PBR is also incorporated by 3D Gaussians, which enable Gaussian relighting. 3DGS-DR [46] and GShader [18] both specialize in specular objects and propose customized shading functions. R3DG [15] further incorporates ray tracing for indirect illumination, while GS-IR [26] relies on baked occlusion maps. With only single environment captures as input, these Gaussian relighting methods face challenges in correctly decoupling lighting from material properties.

Albedo-lighting Disambiguation. Recovering reliable illumination and albedo from object appearances is an ill-posed problem [35] in inverse rendering. It is often simplified under the assumption of controllable lighting [6, 31], known geometry [32], or when regularized with strong priors [13, 37] in domain-specific tasks such as relighting of human faces. For more general cases, object captures from varied lighting, known as photometric images, can provide important visual cues. Several NeRF-based relighting methods have taken advantage of such captures. NeRV [39] requires multiple known lighting environments. NeRD [7]

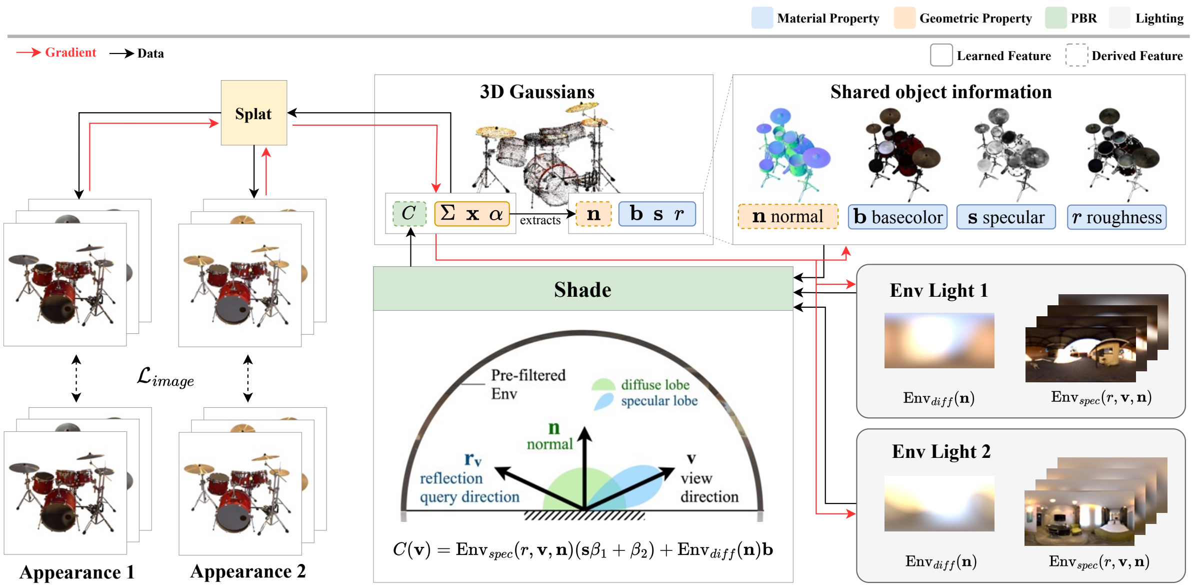

Material Property
Geometric Property
PBR
Lighting
Gradient
Data
Learned Feature
Derived Feature
Splat
3D Gaussians
Shared object information
C
Σ $\mathbf{x}$ α
extracts
$\mathbf{n}$
$\mathbf{b}$ $\mathbf{s}$ r
$\mathbf{n}$ normal
$\mathbf{b}$ basecolor
$\mathbf{s}$ specular
r roughness
Shade
Env Light 1
$\text{Env}_{diff}(\mathbf{n})$
$\text{Env}_{spec}(r, \mathbf{v}, \mathbf{n})$
Env Light 2
$\text{Env}_{diff}(\mathbf{n})$
$\text{Env}_{spec}(r, \mathbf{v}, \mathbf{n})$
$\mathcal{L}_{image}$
Pre-filtered Env
diffuse lobe
specular lobe
$\mathbf{n}$ normal
$\mathbf{r_v}$ reflection query direction
$\mathbf{v}$ view direction
$C(\mathbf{v}) = \text{Env}_{spec}(r, \mathbf{v}, \mathbf{n})(\mathbf{s}\beta_1 + \beta_2) + \text{Env}_{diff}(\mathbf{n})\mathbf{b}$
Appearance 1
Appearance 2

Figure 2. **The proposed ReCap training framework**. Compared to original 3DGS [22], each Gaussian is augmented with 3 extra material attributes. Given k sets of object appearances from unknown lighting conditions as input, k learnable environment maps are instantiated. Gaussian color is computed according to the shading function in the world space based on environment queries and material properties. 2D images are rasterized with standard Gaussian splatting and used for loss computation. $\mathcal{L}_{\text{image}}$: image reconstruction loss from [22]. Additional loss terms for material and geometry are not shown.

models lighting as spherical Gaussians and optimizes separate lighting for every input image. TensoIR [19] encodes lighting as an additional dimension in its factorized tensor representation and relies on ground-truth albedo scaling for accurate relighting. In contrast, we use object captures under unknown lighting, optimizing a learnable light map separately for each scene and independently from the Gaussian model, without requiring ground-truth albedo for relighting.

3. Method

In this section, we introduce ReCap, a robust Gaussian relighting method that leverages cross-environment object captures. The overall framework is illustrated in Fig. 2. In Sec. 3.1, we explain how relighting is enabled in 3DGS with an explicit shading function. In Sec. 3.2, an optimization-friendly variant of the split-sum-approximation is introduced for shading. Lighting representation and post-processing are detailed in Sec. 3.3 and Sec. 3.4, with normal estimation covered in Sec. 3.5.

3.1. Relighting Gaussians with shading function

3DGS represents objects as explicit point clouds. A point at location $\mathbf{x}$ in the world space is represented as a 3D Gaussian function defined by covariance matrix Σ and mean μ:

$$\mathcal{G}(\mathbf{x}|\mu, \Sigma) = e^{\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)} \tag{1}$$

Additionally, each point holds an opacity attribute α and a color attribute $\mathbf{c}$ for point-based α-blending. From a viewing direction $\mathbf{v}$, the pixel color C can be computed by blending ordered points overlapping the pixel:

$$C(\mathbf{v}) = \sum_i \mathbf{c}_i(\mathbf{v})\alpha_i \prod_{j=1}^{j-1}(1-\alpha_j) \tag{2}$$

The original 3DGS models the view-dependent color $\mathbf{c}(\mathbf{v})$ with spherical harmonics. This simplification abstracts the complex view-dependent interactions between material, lighting and geometry into a composite representation. Relighting Gaussians thus requires an alternative representation of $\mathbf{c}(\mathbf{v})$ that factors out the influence of lighting. A natural choice is to use various well-established shading functions from graphics. Ignoring any emission, let the classic rendering equation represent the outgoing radiance at Gaussian point $\mathbf{x}$ viewed from direction $\mathbf{v}$ as

$$L_{out}(\mathbf{x}, \mathbf{v}) = \int_\Omega f_r(\mathbf{x}, \mathbf{v}, \mathbf{l}) L_{in}(\mathbf{x}, \mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\mathbf{l}, \tag{3}$$

where Ω is the hemisphere above the surface, f_r is the bidirectional reflection function (BRDF), $\mathbf{l}$ is the incident light direction and L_{in} is the incoming radiance, all defined in the local coordinate system centered at $\mathbf{x}$. The Gaussian color $\mathbf{c}$ can be computed from L_{out} with proper post-processing such as tone mapping and gamma correction as discussed in Sec. 3.4. The pixel color C becomes

$$C(\mathbf{v}) = \sum_i \mathbf{c}_i(\mathbf{v}|f_r, L_{in}, \mathbf{n})\alpha_i \prod_{j=1}^{j-1}(1-\alpha_j). \tag{4}$$

The rendering equation is commonly simplified for implementation as shading functions. GShader [18] and GS-DR [46] handcrafted their shading functions with a light-independent diffuse component and a light-dependent specular component querying lighting information from a learned environment map. R3DG [15] and GS-IR [26] both leverage microfacet based model as a more expressive alternative. In our work, we start with the split-sum approximation of the microfacet model and propose a more optimization-friendly variant.

3.2. Disambiguate Split Sum Approximation

Disney Principled BRDF [8] provides user-friendly parameters building on the Cook-Torrance microfacet BRDF [9]. We adopt simplifications from Epic Game [21] and consider the following parameters: 1) *basecolor*, $\mathbf{b} \in [0,1]^3$; 2) *roughness*, $r \in [0,1]$; 3) *metallic*, $m \in [0,1]$; and 4) *specular*, $s = 0.04$, assumed to be constant for non-metals.

The BRDF of interest is given as

$$\begin{aligned} f_r(\mathbf{x},\mathbf{v},\mathbf{l}) =&(1-m)\frac{\mathbf{b}}{\pi} \\ &+ \frac{D(r,\mathbf{n},\mathbf{l},\mathbf{v})F(\mathbf{b},m,s,\mathbf{l},\mathbf{v})G(r,\mathbf{n},\mathbf{l},\mathbf{v})}{4(\mathbf{l}\cdot\mathbf{n})(\mathbf{v}\cdot\mathbf{n})}, \end{aligned} \tag{5}$$

where D, F and G are the normal distribution function, the Fresnel term and the geometry term respectively.

Substitute Eq. (5) into Eq. (3) and apply the split-sum approximation as described in [21], the shading function can be written as

$$\begin{aligned} L_{out}(\mathbf{x},\mathbf{v}) = &\underbrace{E_d(\mathbf{n})(1-m)\mathbf{b}}_{diffuse} \\ &+ \underbrace{E_s(\mathbf{n},\mathbf{v})\left[F_0\beta_1(r,\mathbf{n},\mathbf{v}) + \beta_2(r,\mathbf{n},\mathbf{v})\right]}_{specular}, \end{aligned} \tag{6}$$

where E_d and E_s are the pre-filtered environment maps for diffuse and specular reflectance, β_1 and β_2 are pre-calculated BRDF look-ups, $F_0 = m\mathbf{b} + (1-m)s$ is the effective reflectance.

This is a linear blend of two different models controlled by the metallic parameter. For metals, there is no diffuse component and the specular part is colored. For non-metals, the diffuse part is colored but not the specular part. Dropping arguments for conciseness, the two models are

$$\begin{aligned} L_{\text{metal}} &= E_s\mathbf{b}\beta_1 + E_s\beta_2 \\ L_{\text{non-metal}} &= E_s s\beta_1 + E_s\beta_2 + E_d\mathbf{b}. \end{aligned} \tag{7}$$

Without prior knowledge on the distribution of material parameters of real metal and non-metal, optimization of the metallic parameter is problematic especially when the lighting is also learned. First, s and $\mathbf{b}$ have overlaps. Certain specular highlights of dark surfaces may be misinterpreted and assigned to the wrong metallic value, as illustrated by the example in Fig. 3. Furthermore, when the environment maps and roughness values are optimized such that $E_d \sim E_s\beta_1$, the two equations becomes interchangeable with some value of $\mathbf{b}$ leading to ambiguous optimization.

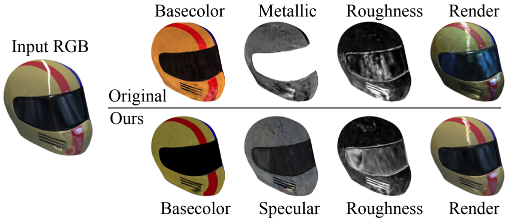


Figure 3. With the original shading model, the shield of the helmet is falsely identified as being metallic during optimization.

Instead, we discard the metallic parameter and expand the specular parameter to propose a general expression as

$$L_{\text{out}} = E_s\mathbf{s}\beta_1 + E_s\beta_2 + E_d\mathbf{b}, \tag{8}$$

where $\mathbf{s} \in [0,1]^3$ is now a vector representing *specular tint*. For $\mathbf{s} = [s,s,s]^T$, it becomes $L_{\text{non-metal}}$; for $\mathbf{s} = \mathbf{b}_{\text{metal}}$ and $\mathbf{b} = 0$, it becomes L_{metal}. On the downside, the expanded range of $\mathbf{s}$ encompasses a significant portion of unnatural specular colors. To avoid overly saturated specular tint, we apply saturation penalty on $\mathbf{s}$ as

$$\mathcal{L}_{\text{sat}} = \lambda_{\text{sat}} \cdot \|\mathbf{s} - \mathbf{s}_{\text{mean}}\|. \tag{9}$$

To account for energy conservation, we further include a regularizer to encourage $\|\mathbf{s}\| + \|\mathbf{b}\| \leq 1$.

3.3. Lighting Representation

While spherical functions [16, 41, 48] are popular choices for efficient lighting representations, we adopt an image based lighting [10] model to provide high frequency details necessary for specular reflections. Specifically, each environment lighting is represented by a $6 \times 256 \times 256$ learnable cube map, which is pre-filtered into a diffuse map, E_d, for diffuse reflection [34] and a set of specular mipmaps across different roughness levels, E_s, for specular reflection [21]. In practice, we use the efficient approximation provided by NVDIFFRAST [30] for differentiable pre-filtering and querying of the learnable environment maps, which are performed in every forward pass for shading.

To leverage photometric supervision from cross-environment captures, k sets of learnable environment maps will be instantiated to explain k sets for object appearances. As illustrated by Fig. 4, the same object position can display a range of pixel colors depending on viewing direction and lighting. With only single-environment captures, the view-dependent variations can be explained by a multitude of material-lighting combinations, manifesting the albedo-lighting ambiguity. By introducing additional sets of object appearances under new, unknown environments, the light-dependent variations are attributed to corresponding learnable environment maps as multi-task learning targets. The joint optimization encourages physically sound decoupling of material properties and lighting to explain all observed appearances as guided by the shading function.

While increasing the number of environments can enhance the decoupling as later shown in Sec. 4.3, it also re-

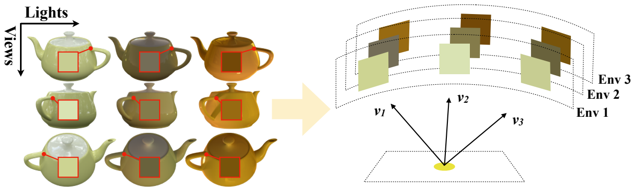


Figure 4. The same object position exhibit view-dependent and light-dependent pixel color, the later is accounted for by querying corresponding learnable environments.

quires additional effort to capture real objects across multiple scenes. As a proof of concept, we limit our investigation to a dual-environment setup unless mentioned otherwise.

3.4. Post-shading Processing

Standard HDR maps represent radiance in linear RGB space when used as lighting sources in rendering [10]. To apply them effectively for relighting, the learned lighting values need a clear physical interpretation. Current methods adopt diverse post-shading strategies during training, influencing both the interpretation of learned environment maps and the processing of new HDR maps. For example, if the learned values are constrained to $[0, 1]$ and gamma correction is applied post-shading, the environment map is effectively treated as being in a linear low dynamic range (LDR) space.

In Tab. 1, we analyze three major factors that affects relighting. Gamma correction is essential for relighting, as expected, since the shading function models linear light transport. However, it is often overlooked [15, 46], because omitting it still yields good NVS results. While previous work also consider complex tone-mappers [39] from graphics, they perform worse than simple clipping, likely due to the introduced non-linearity hindering optimization.

To balance relighting and NVS, we constrain environment maps to be positive and apply simple clipping followed by gamma correction. This implies a linear HDR lighting space where new environment maps can be used without modification, removing the need for normalizing relit images [27] or learned albedo [19] to match the ground-truth (unavailable in practice) when when evaluating relighting. Combined with cross-environment decoupling, this essentially improves relighting by (i) allowing Gaussians to push light-dependent appearances into learnable "sinks" as pseudo light maps, and (ii) regulating these "sinks" with HDR processing and PBR, enabling them to approximate real ambient light in value distribution.

3.5. Geometry Estimation

Table 1. Relighting and NVS performance (PSNR) of various design choices. The → symbol denotes implication on the learned environment map. During relighting, the input HDR map are pre-processed to match the interpretation. See the supplementary for more details.

Env. Map Range	Tonemap	Gamma	Relight	NVS
$[0, 1] \rightarrow$ LDR	✗	✗→ non-linear	23.55	29.97
$[0, 1] \rightarrow$ LDR	✗	✓→ linear	24.07	30.09
$[0, \infty) \rightarrow$ HDR	clip	✗→ non-linear	22.69	**32.36**
$[0, \infty) \rightarrow$ HDR	clip	✓→ linear	**25.82**	32.23
$[0, \infty) \rightarrow$ HDR	reinhard	✓→ linear	23.13	29.79
$[0, \infty) \rightarrow$ HDR	aces	✓→ linear	23.70	30.64

Accurate normal estimation is essential for precise querying of the high frequency light maps. As discrete sparse point clouds, 3D Gaussians have no native support for normal computation. Following the common observation that converged Gaussians are often flat and align with the object surface [17, 18, 26, 46], we adopt the shortest axis of each Gaussian as an estimation of the normal vector. This shortest-axis normal approximation, $\mathbf{n}$, is simply constrained using depth-derived normal, $\hat{\mathbf{n}}$, where the depth image is rendered from Gaussian opacities. This depth-normal consistency loss is given as

$$\mathcal{L}_{\text{dn}} = \lambda_{\text{dn}} \cdot \|\mathbf{n} - \hat{\mathbf{n}}\|^2 . \qquad (10)$$

While existing works commonly augment normal estimation with additional residuals [18] or learn per-Gaussian normal vectors directly [15, 26], we observe that such learnable normals often overfit to explain specularities or shadowing effects. This leads to improved NVS but compromises geometric accuracy. As shown in Fig. 5, the cross-lighting consistency imposed by ReCap naturally improves normal estimation by preventing overfitting to a single lighting condition. When only single-environment captures are used, specular highlights become embedded in the surface normals, preserving the specific highlight shapes seen in training even when relighted with new environment maps. In contrast, our ReCap with dual lighting produce accurate highlight shapes with better surface normal.

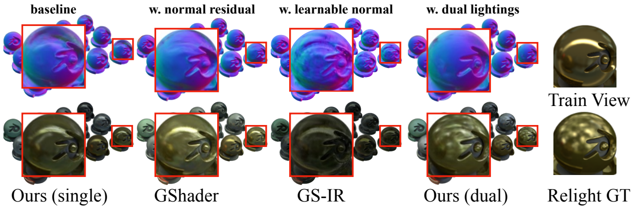


Figure 5. The comparison of estimated normal and corresponding relighting results. With single-environment captures, the highlights from the train view are falsely attributed to object property instead of lighting, passing down to relighting views. Cross-environment supervision provides more robust normal estimation and correct highlight shapes.

4. Experiments

4.1. Implementation Details

Datasets. We construct a more comprehensive *RelightObj* dataset by relighting 8 general objects from NeRF Synthetic Dataset [28] and 5 shiny objects from the Shiny Blender Dataset [40] under 8 different HDR maps. Each scene includes 200 training views and 200 test views, with identical camera poses across scenes. To eliminate biases from training view poses, the training views are divided into two

Table 2. Relighting and NVS results. Methods optimized over single-env (✦) and dual-env (✧) inputs differ in the choice of training split B. "GT scaling" refers to the use of ground truth albedo to re-scale learned albedo for relighting. Best and second best results are highlighted in **bold** and *italic* respectively.

	use GT	*Training Envs*		*Unseen Relights*							*Seen Relights*		*NVS*	
	scaling	**split A**	**split B**	**night**	**snow**	**sunset**	**interior**	**fireplace**	**forest**	**Avg.**	**bridge**	**courtyard**	**bridge***	**courtyard***
PSNR↑														
3DGS-DR✦[46]	-	bridge	bridge	20.18	18.99	23.19	21.33	18.76	*24.90*	21.22	**33.18**	-	**35.59**	-
GS-IR✦[26]	-	bridge	bridge	25.56	*22.44*	21.22	19.46	25.64	21.10	22.57	22.01	-	31.22	-
R3DG✦[15]	-	bridge	bridge	23.71	22.09	22.84	20.09	23.08	22.19	22.33	22.32	-	30.61	-
GShader✦[18]	-	bridge	bridge	21.48	17.61	18.93	21.89	20.18	22.70	20.47	26.03	-	33.05	-
ReCap✦ (ours)	-	bridge	bridge	25.31	21.76	23.20	21.66	24.52	23.15	23.27	25.30	-	*33.40*	-
3DGS-DR✧[46]	-	bridge	courtyard	20.53	20.25	*24.56*	21.71	19.41	24.25	21.78	24.89	23.92	24.84	23.57
GS-IR✧[26]	-	bridge	courtyard	26.22	22.03	20.85	19.18	*25.90*	20.52	22.45	20.78	20.96	24.92	23.16
R3DG✧[15]	-	bridge	courtyard	23.91	21.78	22.55	19.86	23.38	21.74	22.21	20.98	20.84	25.26	22.68
GShader✧[18]	-	bridge	courtyard	22.73	18.20	19.96	22.11	20.97	23.02	21.17	23.34	21.75	24.96	22.84
TensoIR✧[19]	✓	bridge	courtyard	*27.22*	22.15	24.31	23.12	25.35	24.80	*24.49*	24.50	23.82	29.13	27.46
TensoIR✧$_{\text{no scale}}$[19]	-	bridge	courtyard	26.24	20.11	22.75	21.79	24.45	23.29	23.11	23.50	22.47	29.13	27.46
GShader✧[18] + ours	-	bridge	courtyard	23.52	17.69	19.95	*23.22*	21.75	24.50	21.77	24.45	*24.32*	31.13	*29.30*
ReCap✧ (ours)	-	bridge	courtyard	**27.62**	**24.64**	**26.33**	**24.40**	**26.52**	**25.38**	**25.82**	*26.95*	**25.52**	32.23	**30.76**
SSIM↑														
3DGS-DR✦[46]	-	bridge	bridge	0.882	0.894	0.925	0.907	0.843	0.906	0.893	**0.971**	-	**0.978**	-
GS-IR✦[26]	-	bridge	bridge	0.902	0.904	0.890	0.861	0.884	0.867	0.885	0.898	-	0.952	-
R3DG✦[15]	-	bridge	bridge	0.889	0.913	0.918	0.866	0.866	0.877	0.888	0.879	-	0.959	-
GShader✦[18]	-	bridge	bridge	0.889	0.887	0.896	0.905	0.854	0.904	0.889	0.949	-	0.968	-
ReCap✦ (ours)	-	bridge	bridge	0.911	*0.918*	0.926	0.907	0.883	0.906	*0.909*	0.946	-	*0.970*	-
3DGS-DR✧[46]	-	bridge	courtyard	0.885	0.902	*0.934*	0.907	0.846	0.903	0.896	0.930	0.927	0.926	0.924
GS-IR✧[26]	-	bridge	courtyard	0.906	0.893	0.887	0.858	0.885	0.862	0.882	0.881	0.879	0.915	0.905
R3DG✧[15]	-	bridge	courtyard	0.896	0.912	0.922	0.866	0.870	0.873	0.890	0.859	0.888	0.932	0.919
GShader✧[18]	-	bridge	courtyard	0.904	0.891	0.908	0.906	0.862	0.905	0.896	0.925	0.913	0.928	0.922
TensoIR✧[19]	✓	bridge	courtyard	0.908	0.861	0.891	0.870	0.886	0.888	0.884	0.893	0.883	0.962	*0.957*
TensoIR✧$_{\text{no scale}}$[19]	-	bridge	courtyard	0.910	0.861	0.893	0.871	*0.889*	0.889	0.885	0.895	0.884	0.962	0.957
GShader✧[18] + ours	-	bridge	courtyard	*0.915*	0.893	0.915	*0.919*	0.874	*0.922*	0.906	0.937	*0.938*	0.959	0.956
ReCap✧ (ours)	-	bridge	courtyard	**0.935**	**0.938**	**0.951**	**0.929**	**0.903**	**0.926**	**0.930**	*0.951*	**0.943**	0.966	**0.963**
LPIPS↓														
3DGS-DR✦[46]	-	bridge	bridge	0.081	0.091	0.073	*0.085*	0.108	0.083	0.087	**0.041**	-	**0.034**	-
GS-IR✦[26]	-	bridge	bridge	0.099	0.094	0.101	0.119	0.108	0.109	0.105	0.099	-	0.065	-
R3DG✦[15]	-	bridge	bridge	0.103	0.086	0.082	0.124	0.114	0.104	0.102	0.113	-	0.053	-
GShader✦[18]	-	bridge	bridge	0.091	0.116	0.098	0.094	0.113	0.091	0.100	0.062	-	0.044	-
ReCap✦ (ours)	-	bridge	bridge	0.077	*0.080*	*0.073*	0.089	*0.092*	0.084	*0.083*	0.061	-	*0.042*	-
3DGS-DR✧[46]	-	bridge	courtyard	0.084	0.096	0.078	0.096	0.115	0.094	0.094	0.078	0.080	0.082	0.085
GS-IR✧[26]	-	bridge	courtyard	0.101	0.107	0.107	0.127	0.112	0.116	0.112	0.115	0.119	0.099	0.104
R3DG✧[15]	-	bridge	courtyard	0.101	0.092	0.083	0.129	0.117	0.110	0.105	0.127	0.106	0.078	0.089
GShader✧[18]	-	bridge	courtyard	0.087	0.115	0.095	0.101	0.114	0.097	0.101	0.086	0.094	0.080	0.086
TensoIR✧[19]	✓	bridge	courtyard	0.131	0.155	0.134	0.138	0.138	0.127	0.137	0.128	0.133	0.093	0.098
TensoIR✧$_{\text{no scale}}$[19]	-	bridge	courtyard	0.133	0.159	0.136	0.141	0.138	0.131	0.140	0.130	0.137	0.093	0.098
GShader✧[18] + ours	-	bridge	courtyard	*0.074*	0.113	0.084	0.089	0.102	*0.080*	0.090	0.073	*0.073*	0.055	*0.059*
ReCap✧ (ours)	-	bridge	courtyard	**0.059**	**0.069**	**0.057**	**0.077**	**0.075**	**0.071**	**0.068**	*0.058*	**0.064**	0.047	**0.051**

splits, A and B, each containing 100 views. In the single-environment (single-env) setup, both splits come from the same environment, whereas in the dual-environment (dual-env) setup, split B is taken from a different environment. Relighting performance is judged on unseen environments.

Baseline and Metrics. We compare with the following baselines: (a) **GShader** [18]: a Gaussian rendering method utilizing a customized shading function with a residual color term; (b) **GS-IR** [26]: an inverse rendering approach using view-space shading with a microfacet BRDF model and a baked occlusion map; (c) **3DGS-DR** [46]: a Gasussian rendering method targeting reflective surfaces with a customized shading function and normal propagation; (d) **R3DG** [15]: a Gaussian relighting technique that incorporates ray tracing and a microfacet BRDF model; and (e) **TensoIR** [19]: top-performing NeRF-based method in novel view synthesis and relighting. Following these works, rendering and relighting quality are evaluated using PSNR, SSIM [42], and LPIPS [49].

Training and Testing. All experiments are conducted on an Nvidia RTX 3090 graphics card. Our models are optimized using Adam for 30,000 iterations. All other methods are retrained on our dataset using their official repositories and recommended settings, with results reported from the retrained models. As GShader [18] and 3DGS-DR [46] lack official relighting code, we pre-processed HDR maps following their design choices (details in the supplementary). For TensoIR [19], which uses per-object hyperparameters, we tested all provided settings and report the best results.

4.2. Comparison with Previous Methods

Single-env Setup. We compare single-env performance in the first group of Tab. 2. The substantial difference between NVS using the learned environment map and relighting us-

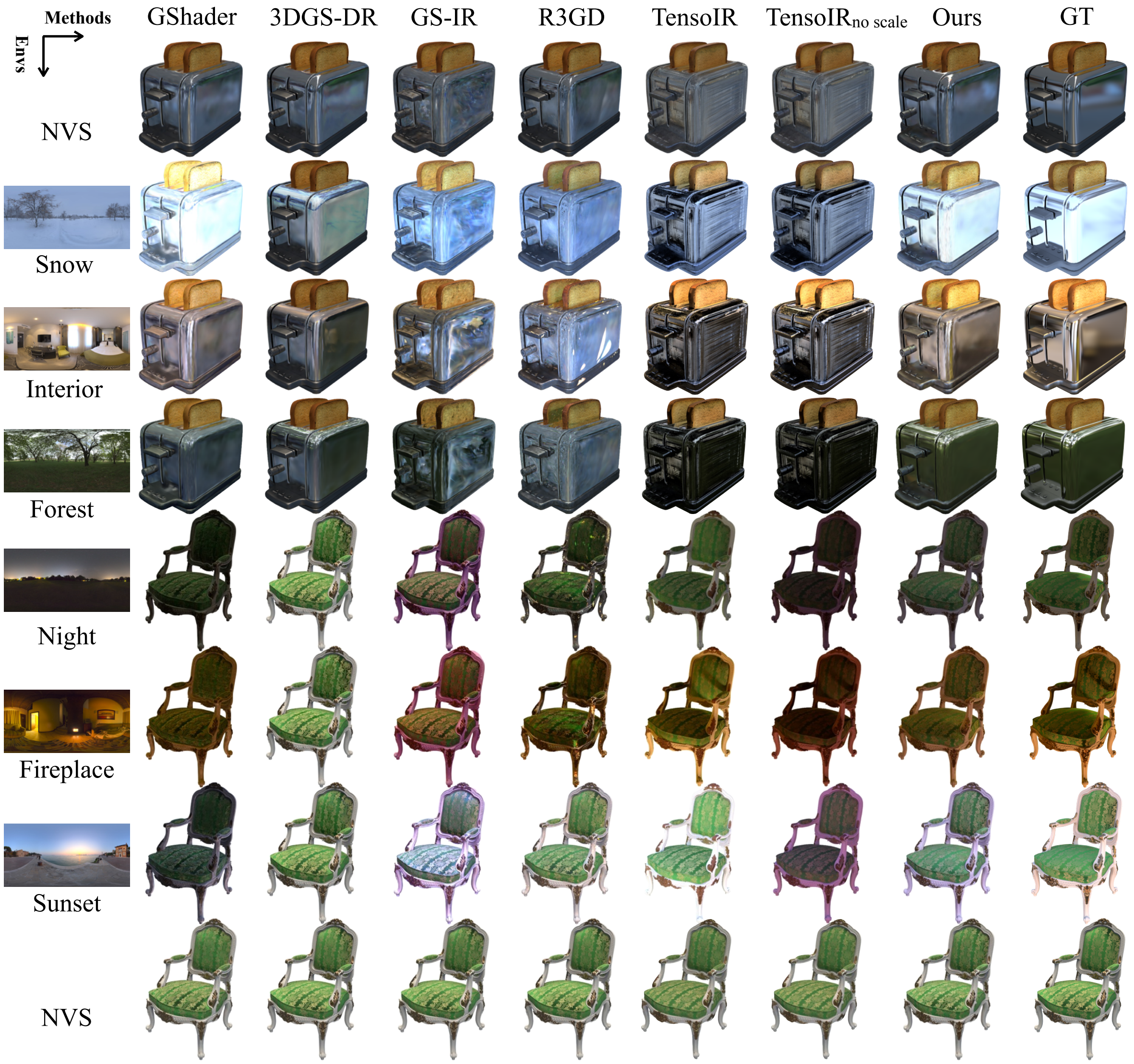


Figure 6. Qualitative comparison of NVS and relighting results. Upper: *toaster*, lower: *chair*. TensoIR produces accurate relighting with GT albedo scaling for *chair*, but suffers from tone-shift without it. For our method, specular surfaces in *toaster* provide more information for lighting reconstruction, leading to better albedo-lighting decoupling hence more truthful relighting results than mostly diffuse *chair*. Refer to the supplementary for more visuals.

ing the ground truth HDR map (bridge* vs. bridge) highlights the albedo-lighting ambiguity: the learned lighting representation diverges from ground truth but still works effectively with the learned materials for NVS. While 3DGS-DR✦ [46] has the smallest gap, its ability to generalize to unseen relighting conditions remains limited. As shown in the second column of Fig. 6, this is because it tends to reproduce training views, preserving the overall tone and adapting only the specular reflections to new lighting.

Dual-env Setup. The second group of Tab. 2 presents results from directly optimizing existing Gaussian models over dual-env inputs while maintaining a single learned lighting representation. This leads to mixed relighting performance and significant NVS artifacts since the learned environment is no longer well-defined. Therefore, we continue our analysis with their single-env results for the remainder of the discussion. In the third group, we compare against TensoIR [19] with their multi-light setup and further augment GShader [18] with the proposed multi-light querying. Since TensoIR [19] uses GT albedo scaling in their official code, we additionally report their scale-free performance for clearer comparison.

Qualitative Comparison. Fig. 6 compares NVS and relighting performance across single-env Gaussian relighting methods, two dual-env TensoIR [19] variants, and dual-env ReCap. Our ReCap achieves noticeable perceptual improvements over methods that do not rely on GT for relighting. TensoIR [19] produces accurate relighting with GT albedo scaling, but suffers from obvious tone-shift without it (e.g., chair) and struggles with metallic objects (e.g., toaster) due to their dielectric assumption.

Reconstructed Environments. While specular objects

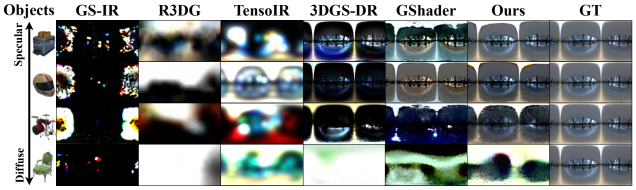


Figure 7. Learned environment maps for different objects. As objects become more diffuse, reconstructions are less accurate.

pose significant challenges for NVS due to their complex view-dependent appearance[18, 46], they offer distinct advantages for relighting tasks. Specular surfaces act as natural light probes, providing abundant high-frequency, view-dependent details that facilitate more accurate environment map reconstruction. In Fig. 7, we display the reconstructed environment maps for objects with varying level of specularity. For predominantly diffuse objects, disentangling lighting from intrinsic properties is challenging due to limited lighting cues. Notice how our learned light map for *chair* retains object color, highlighting this challenge.

4.3. Ablation Study

Shading Function. To demonstrate the effectiveness of our proposed shading function in Eq. (8), we compare the average relighting performance across unseen scenes in Tab. 3. The new shading function yields the most substantial improvement by offering a more flexible material representation, with the two additional regularization terms also contributing positively to the final results. Additional visual examples are provided in the supplementary material.

Table 3. Ablation of shading function and regularization terms. $\mathcal{L}_{sat}$: specular saturation loss. $\mathcal{L}_{ec}$: energy conservation loss.

Proposed Shading	$\mathcal{L}_{sat}$	$\mathcal{L}_{ec}$	PSNR	SSIM	LPIPS
-	-	-	24.62	0.923	0.076
✓	-	-	25.29	0.929	0.070
✓	✓	-	25.38	0.929	0.069
✓	✓	✓	**25.58**	**0.930**	**0.069**

Extra training environments. So far, we have focused on dual-env setups. Here, we examine how extra photometric supervision from additional training environments affects the relighting. Due to the constraint of 200 training views per scene, we allocate 100 identical views per environment as we include more environments for training. To align with existing results, we also compare the use of identical views versus extra views. Results in Tab. 4 reveal three key findings: (i) Adding identical views from an extra environment is better than adding extra unique views within the same environment (row 2 vs. 3); (ii) Although identical views across environments theoretically provides stronger decoupling, the benefit of extra unique views is more pronounced in practice (row 3 vs. 4). It is likely that extra views already provide sufficient decoupling, especially with perfect pose calibration in synthetic datasets; and (iii) Increasing the number of environments consistently enhances relighting performance, with no observed plateau up to five.

Table 4. Ablation on the number of environments used in training. Since up to 5 environments are used, relighting results are reported on the remaining 3 unseen scenes: sunset, fireplace, night.

# Envs	# Views	# Unique Views	Cam Poses	PSNR	SSIM	LPIPS
1	100	100	identical	23.98	0.906	0.084
1	200	200	extra	24.07	0.908	0.081
2	200	100	identical	26.14	0.927	0.066
2	200	200	extra	26.25	0.928	0.065
3	300	100	identical	26.35	0.930	0.064
4	400	100	identical	26.68	0.931	0.062
5	500	100	identical	27.36	0.936	0.060

Application on Real Captures. ReCap requires multiple sets of camera poses to align to a common coordinate. In Fig. 8, we demonstrate its practicality with two real-life examples from StanfordORB [24], which captures the same object in different environments and provides COLMAP-estimated poses for each environment. They also provide sparse cross-environment alignment pairs computed by SuperGlue [38]. We use these pairs to estimate the view transform matrix and account for scale ambiguity. While exact calibration is non-trivial, sufficiently accurate pose calibration proves to be feasible for the proposed application.

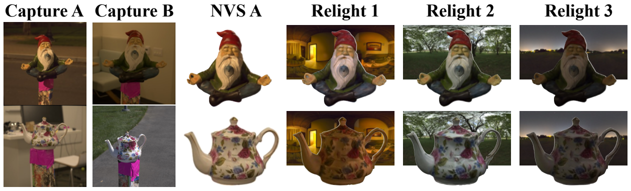


Figure 8. Relighting and NVS examples on StanfordORB: *gnome*, *teapot*.

4.4. Limitation and discussion

While we achieve compelling relighting results, indirect illumination and subsurface scattering effects are not considered. Over-exposed regions also present inherent challenges for opacity estimation in all Gaussian-based relighting methods, where clipped highlights are incorrectly interpreted as transparent against white splatting backgrounds. For easier pose calibration in practice, semi-controlled capturing, such as rotating the object within the same environment at a known angle, may be considered.

5. Conclusion

In this paper, we proposed ReCap, leveraging internal photometric consistency to address the albedo-lighting ambiguity limiting existing Gaussian relighting methods. With cross-environment captures, we explicitly model light-dependent appearance with independent learnable lighting representations that share a common set of material attributes. Combined with an optimization-friendly shading function and physically appropriate post-processing during training, ReCap demonstrates realistic relighting quality with truthful tones across diverse scenes.

Acknowledgment. This research was supported by the Alexander von Humboldt Foundation.

References

[1] Jonathan T Barron and Jitendra Malik. Shape, albedo, and illumination from a single image of an unknown object. In *CVPR*, 2012. 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023. 2

[5] Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. Recovering intrinsic scene characteristics. *Computer Vision Systems*, 2(3-26):2, 1978. 2

[6] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *ECCV*, 2020. 2

[7] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *ICCV*, pages 12684–12694, 2021. 2

[8] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, pages 1–7. vol. 2012, 2012. 4

[9] Robert L Cook and Kenneth E Torrance. A reflectance model for computer graphics. *ACM SIGGRAPH*, 15(3):307–316, 1981. 4

[10] Paul Debevec. Rendering with natural light. In *ACM SIGGRAPH Electronic art and animation catalog*, page 166. 1998. 4, 5

[11] Paul Debevec. The light stages and their applications to photoreal digital actors. *ACM SIGGRAPH Asia*, 2(4):1–6, 2012. 2

[12] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *PACMCGIT*, pages 145–156, 2000. 2

[13] Haiwen Feng, Timo Bolkart, Joachim Tesch, Michael J Black, and Victoria Abrevaya. Towards racially unbiased skin tone estimation via scene disambiguation. In *ECCV*, 2022. 2

[14] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[15] Jian Gao, Chun Gu, Youtian Lin, 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. 1, 2, 4, 5, 6

[16] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In *CVPR*, 2019. 4

[17] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *CVPR*, 2024. 5

[18] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *CVPR*, 2024. 1, 2, 3, 5, 6, 7, 8

[19] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *CVPR*, 2023. 1, 2, 3, 5, 6, 7

[20] Kaizhang Kang, Cihui Xie, Chengan He, Mingqi Yi, Minyi Gu, Zimin Chen, Kun Zhou, and Hongzhi Wu. Learning efficient illumination multiplexing for joint capture of reflectance and shape. *ACM TOG*, 38(6):165–1, 2019. 2

[21] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013. 2, 4

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

[23] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *CGF*, pages 29–43. Wiley Online Library, 2021. 2

[24] 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. *NeurIPS*, 36:46938–46957, 2023. 8

[25] Min Li, Zhenglong Zhou, Zhe Wu, Boxin Shi, Changyu Diao, and Ping Tan. Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials. *IEEE TIP*, 29:4159–4173, 2020. 2

[26] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *CVPR*, 2024. 1, 2, 4, 5, 6

[27] 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. *ACM TOG*, 42(4):1–22, 2023. 2, 5

[28] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 5

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

[30] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *CVPR*, 2022. 4

[31] Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM TOG*, 37(6):1–12, 2018. 2

[32] Jeong Joon Park, Aleksander Holynski, and Steven M Seitz. Seeing the world in a bag of chips. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1417–1427, 2020. 2

[33] Ruggero Pintus, Tinsae Gebrechristos Dulecha, Irina Ciortan, Enrico Gobbetti, and Andrea Giachetti. State-of-the-art in multi-light image collections for surface visualization and analysis. In *CGF*, pages 909–934. Wiley Online Library, 2019. 2

[34] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *ACM SIGGRAPH*, 2001. 4

[35] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *ACM SIGGRAPH*, pages 117–128, 2001. 2

[36] Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. Image based relighting using neural networks. *ACM TOG*, 34(4):1–12, 2015. 2

[37] Xingyu Ren, Jiankang Deng, Chao Ma, Yichao Yan, and Xiaokang Yang. Improving fairness in facial albedo estimation via visual-textual cues. In *CVPR*, 2023. 2

[38] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 8

[39] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021. 1, 2, 5

[40] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 2, 5

[41] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. In *ACM SIGGRAPH Asia*, 2009. 4

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 6

[43] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *CVPR*, 2022. 2

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

[45] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM TOG*, 38(4):1–13, 2019. 2

[46] Keyang Ye, Qiming Hou, and Kun Zhou. 3d gaussian splatting with deferred reflection. In *ACM SIGGRAPH*, 2024. 1, 2, 3, 5, 6, 7, 8

[47] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM TOG*, 38(6):1–14, 2019. 2

[48] 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 *CVPR*, 2021. 2, 4

[49] 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

[50] 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 TOG*, 40(6):1–18, 2021. 1, 2

[51] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022. 2