

Appendix

A Datasets

We test on three synthetic scenes and seven real-world scenes from NRHints [34], as shown in Figure 7. Each image is rendered (or captured) with a unique camera pose and point-light position. For each synthetic scene, we generate a total of 600 OLAT images using Blender, with 500 for training and 100 for testing, following either out-of-distribution or colocated data patterns to align with their settings. Specifically, the camera and light source are independently sampled on the upper hemisphere with centered around the scene, with elevation angles ranging from 10° to 80° , and radial distances uniformly sampled between $4.0\times$ and $5.0\times$ the object bounding size. This setup ensures a wide diversity of incident angles while maintaining plausible visibility. In the OOD configuration, training and testing light directions are drawn from non-overlapping hemispheres (azimuth $\phi \in [0, \pi]$ for training; $\phi \in [\pi, 2\pi]$ for testing) to induce generalization difficulty. In the colocated setting, camera and light positions are identical, emulating a mobile-light capture scenario. All images are rendered at a resolution of 512×512 using Cycles path tracing in Blender, with 256 samples per pixel. For real captured scenes in the out-of-distribution experiment, we manually divide the data into training and testing sets, each with lights positioned on opposite sides of the upper hemisphere.

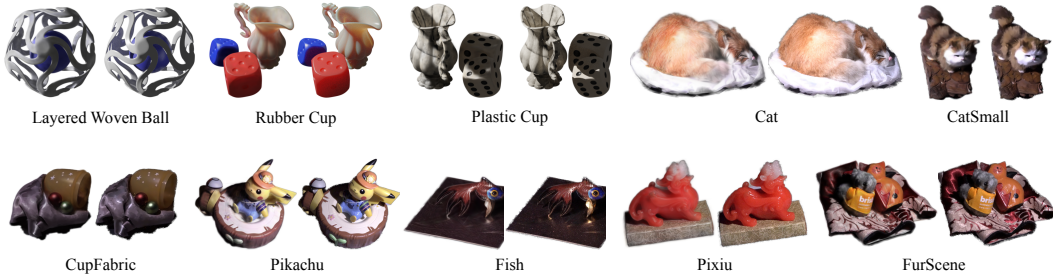


Figure 7: Synthetic and real-world scenes used in the experiment with ground-truth images (right) and rendering results of MetaGS (left).

B More Experimental Analyses

B.1 Full Comparisons for OOD Relighting

We present the OOD relighting results with all three metrics in Table 7 and Table 8.

Table 7: Full comparison of the OOD relighting comparison on NRHints synthetic datasets.

Method	Layered Woven Ball			Plastic Cup			Rubber Cup		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
NRHints [34]	17.25	0.8755	0.0872	23.92	0.9485	0.0470	27.44	0.9465	0.0691
WildLight [5]	21.73	0.9217	0.0568	20.95	0.9249	0.0629	24.02	0.9221	0.0789
GS ³ [1]	18.84	0.9063	0.0589	20.30	0.9141	0.0628	24.37	0.9349	0.0715
Ours	26.76	0.9561	0.0423	27.54	0.9580	0.0405	27.95	0.9497	0.0688

B.2 Clarification on Meta-learning

In our OLAT method, meta-learning helps mitigate overfitting to specific illuminations by explicitly simulating test conditions during training. Consider a renderer $R(\theta, l_i)$, where θ represents light-independent parameters (material/geometry), and l_i denotes the light position. Through bilevel gradients, meta-learning assesses how well the learned θ' under $L = \|R(\theta, l_i) - I_i\|^2$ adapts to minimizing $\|R(\theta', l_j) - I_j\|^2$ with the simulated test condition l_j , thereby enforcing cross-illumination consistency in geometry reconstruction. This new loss function encourages the model to learn to generalize to unseen illumination rather than to overfit to a specific training sample as direct RGB loss does. By forcing geometry parameters θ to remain consistent across $\{l'_i\}$, we prevent illumination

Table 8: Full comparison of OOD relighting results on NRHints real-world datasets.

Method	Cat			Catsmall			CupFabric			Fish		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
NRHints [34]	18.04	0.7821	0.2405	24.63	0.9289	0.1094	24.65	0.9377	0.0965	22.57	0.8658	0.1396
WildLight [5]	18.65	0.7293	0.2685	22.53	0.8820	0.1448	24.03	0.9246	0.0969	21.47	0.8419	0.1458
GS ³ [1]	17.66	0.6919	0.2431	23.34	0.9182	0.1097	25.04	0.9408	0.0954	21.12	0.8264	0.1401
Ours	26.45	0.8514	0.1759	26.44	0.9505	0.0980	27.29	0.9490	0.0918	24.68	0.8706	0.1386

Method	FurScene			Pikachu			Pixiu			Average		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
NRHints [34]	21.55	0.8568	0.1435	24.00	0.9279	0.0958	23.03	0.8970	0.1203	22.64	0.8853	0.1351
WildLight [5]	20.33	0.8304	0.1635	19.09	0.8941	0.1129	20.22	0.8330	0.1621	20.90	0.8479	0.1564
GS ³ [1]	17.34	0.7684	0.1669	24.11	0.9203	0.0927	19.63	0.8391	0.1329	21.18	0.8436	0.1387
Ours	24.82	0.8850	0.1297	25.54	0.9350	0.0928	25.65	0.9194	0.1105	25.83	0.9087	0.1196

artifacts from being baked into textures, thereby achieving more accurate highlights and smoother surfaces. As shown in Table 9, increasing batch size (by task number) improves PSNR by 2+, while meta-learning achieves an 8+ improvement, confirming that the impact of meta-learning is not due to a larger batch size.

Table 9: A comparison of meta-learning vs. Increased batch size.

	Meta-learning	w/o Meta-learning	Larger batch ($\times m$)
PSNR	27.42	19.14	20.94

B.3 Multi-light Relighting

To validate MetaGS’s capability in handling complex lighting conditions, we conduct experiments with test sets containing 2–3 light sources. As shown in Figure 8, our method demonstrates robust generalization beyond single-light scenarios.

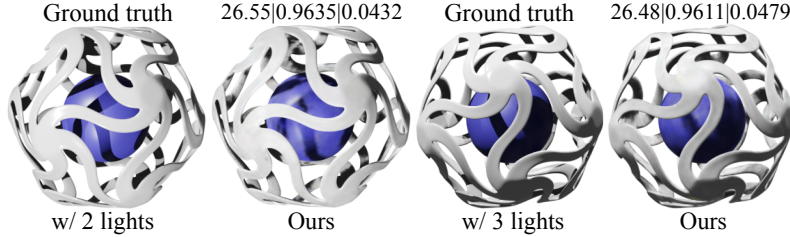


Figure 8: Results with multiple lights (PSNR/SSIM/LPIPS).

B.4 Camera-Light-Colocated Relighting

We present the colocated relighting results with all three metrics in Table 10 and the qualitative results in Figure 9. As shown in the figure, IRON fails to accurately infer the geometry of the Layered Woven Ball scene, resulting in blurry results. In the Plastic Cup and the Rubber Cup scenes, while IRON can provide reasonable geometry reconstruction, it shows more artifacts in the appearance modeling, such as the dent of the dice. On the other hand, MetaGS can provide more accurate illumination details.

B.5 In-distribution Novel View Relighting

This experiment follows the setup in NRHints [34], where both training and testing lights are randomly located in the upper hemisphere surrounding the scene). Each view is illuminated by a single point light placed at a unique position, though all light positions are distributed within the same region. During the testing phase, we evaluate on novel viewpoints under *new, in-distribution* light positions. Similar to the OOD setting, we generate 600 OLAT images per synthetic scene using Blender (500 for training and 100 for testing). For real-world scenes, we use only 600 training images—a significantly smaller subset compared to the full NRHints dataset.

We present the quantitative results for free-viewpoint relighting with in-distribution point light positions in Table 11. Notably, MetaGS outperforms other near real-time methods, including

Table 10: Novel view synthesis results in the **camera-light-colocated** relighting setup.

Method	Layered Woven Ball			Plastic Cup			Rubber Cup		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
IRON [35]	26.99	0.9231	0.0896	34.43	0.9808	0.0296	36.22	0.9677	0.0476
Ours	38.72	0.9904	0.0124	36.90	0.9893	0.0145	38.89	0.9712	0.0385

Table 11: Novel view synthesis results in the in-distribution relighting setup. We present the best and second-best results in bold and underlined format, respectively.

Method	Layered Woven Ball			Plastic Cup			Rubber Cup			Average		
	PSNR [↑]	SSIM [↑]	LPIPS [↓]	PSNR [↑]	SSIM [↑]	LPIPS [↓]	PSNR [↑]	SSIM [↑]	LPIPS [↓]	PSNR [↑]	SSIM [↑]	LPIPS [↓]
3DGS [10]	19.55	0.9130	0.0580	21.13	0.9417	0.0470	23.22	0.9248	0.0796	21.30	0.9265	0.0615
Relightable 3DGS [8]	19.62	0.9088	0.0610	21.07	0.9390	0.0502	23.20	0.9224	0.0838	21.30	0.9234	0.0650
GaussianShader [9]	19.62	0.9137	0.0586	20.92	0.9394	0.0500	23.12	0.9241	0.0833	21.22	0.9257	0.0640
NRHints [34]	18.99	0.9087	0.0731	35.81	0.9847	0.0281	36.64	0.9637	0.0481	<u>30.48</u>	<u>0.9524</u>	<u>0.0498</u>
WildLight [5]	22.71	0.9284	0.0499	23.28	0.9068	0.1018	24.65	0.9271	0.0737	23.55	0.9208	0.0751
GS ³ [1]	<u>28.20</u>	<u>0.9659</u>	<u>0.0396</u>	29.53	0.9755	0.0317	29.34	0.9520	0.0618	29.02	0.9645	0.0444
Ours	30.88	0.9690	0.0281	<u>32.51</u>	<u>0.9814</u>	<u>0.0295</u>	<u>31.75</u>	<u>0.9577</u>	<u>0.0570</u>	31.71	0.9694	0.0382

Relightable 3DGS and GaussianShader, by large margins, and shows an advantage over GS³, a Gaussian-based OLAT method. Our method underperforms NRHints on certain “not-so-difficult” datasets such as *PlasticCup* and *RubberCup*. This is partly because the implicit rendering model in NRHints provides better color fitting, resulting in superior in-distribution performance. While the Phong-based rendering model used in our work facilitates geometry-shading decoupling and is particularly effective for objects with incoherent geometries, such as the *Woven Ball*. However, does NRHints truly learn the intrinsic reflection mechanisms from the observed images? Likely not, as evidenced in the previous OOD results.

C Preliminaries

C.1 3D Gaussian Splatting

Our method builds upon 3D Gaussian Splatting (3DGS) [10], which employs explicit Gaussian points for 3D scene representation. Each Gaussian’s geometry is characterized by an opacity $o \in [0, 1]$, a center point $\mu \in \mathbb{R}^{3 \times 1}$ and a covariance matrix $\Sigma \in \mathbb{R}^{3 \times 3}$:

$$G(\mathbf{X}) = e^{-\frac{1}{2}(\mathbf{X}-\mu)^T \Sigma^{-1}(\mathbf{X}-\mu)}, \quad (4)$$

where the covariance matrix Σ can be decomposed into a scaling matrix $S \in \mathbb{R}^{3 \times 1}$ and a quaternion-parameterized rotation matrix $R \in \mathbb{R}^{3 \times 3}$ for differentiable optimization:

$$\Sigma = R S S^T R^T. \quad (5)$$

To render an image from a given viewpoint, 3DGS uses the splatting technique to cast Gaussians within the view frustum onto the camera plane. The projected 2D covariance matrix Σ' in camera coordinates can be computed as $\Sigma' = J W \Sigma W^T J^T$, where W and J denote the viewing transformation and the Jacobian of the affine approximation of the projective transformation, respectively. The color of each pixel \mathbf{p} is calculated by alpha-blending N ordered Gaussians $\{G_i | i = 1, \dots, N\}$ overlapping \mathbf{p} as:

$$\mathcal{C} = \sum_{i \in N} T_i \alpha_i \mathbf{c}_i \text{ with } T_i = \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (6)$$

where α_i denotes the alpha value calculated by multiplying the opacity o with the 2D Gaussian contribution value derived from Σ' , and T_i denotes the accumulated transmittance along the ray.

Specifically, each Gaussian’s color is defined by spherical harmonics (SH) coefficients $\mathcal{C} \in \mathbb{R}^k$ (where k represents the degrees of freedom). While high-order SH coefficients can contribute to the overall lighting of a scene under various view directions, they are not specifically designed to handle view-dependent color directly, such as specular reflections and shadows.

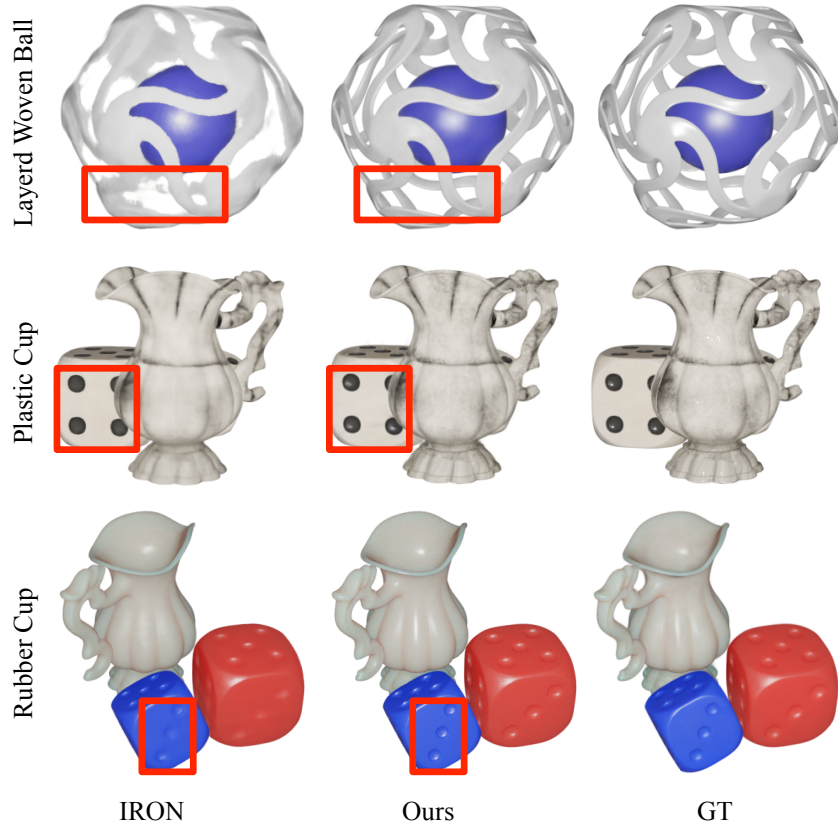


Figure 9: Novel view synthesis results under the camera-light-colocated relighting setup. Compared to IRON, our model more accurately infers object geometry in the Layered Woven Ball scene and produces fewer artifacts in the Plastic Cup scene.

C.2 Blinn-Phong Reflection Model

The Phong model is widely adopted in computer graphics for simulating the interaction of light with object surfaces [19]. In this model, the reflected light transport on a surface has three components: ambient (L_a), diffuse (L_d), and specular reflections (L_s). The ambient reflection component represents the constant illumination present in the environment, simulating how light scatters and reflects off other surfaces to establish a baseline brightness level. The diffuse reflection component, governed by Lambertian diffusion law, illustrates the scattering of light in multiple directions upon striking a rough surface, leading to a matte appearance. This component is calculated based on the angle between the light direction (\mathbf{l}) and the surface normal (\mathbf{n}), ensuring that surfaces facing the light source appear brighter. The specular term is computed based on the angle between the light direction vector and the bisector (\mathbf{h}) between the view direction (\mathbf{v}) and the light direction (\mathbf{l}). The Phong model also incorporates a shininess exponent that governs the spread of the specular highlight, allowing the model to represent the surface’s glossiness. We select the Blinn-Phong model [2] to provide the illumination decomposition priors to 3DGS for its simple yet strong physical prior and computation efficiency. With little addition to the Gaussian’s attribute, our model produces realistic lighting effects.