

# NieR: Normal-Based Lighting Scene Rendering

Hongsheng Wang<sup>\*1,2</sup>, Yang Wang<sup>\*2</sup>, Yalan Liu<sup>2</sup>, Fayuan Hu<sup>2</sup>, Shengyu Zhang<sup>†1</sup>,  
Fei Wu<sup>1</sup>, and Feng Lin<sup>2</sup>

<sup>1</sup>Zhejiang University, China

<sup>2</sup>Zhejiang Lab, China

## Abstract

*In real-world road scenes, diverse material properties lead to complex light reflection phenomena, making accurate color reproduction crucial for enhancing the realism and safety of simulated driving environments. However, existing methods often struggle to capture the full spectrum of lighting effects, particularly in dynamic scenarios where viewpoint changes induce significant material color variations. To address this challenge, we introduce NieR (Normal-Based Lighting Scene Rendering), a novel framework that takes into account the nuances of light reflection on diverse material surfaces, leading to more precise rendering. To simulate the lighting synthesis process, we present the LD (Light Decomposition) module, which captures the lighting reflection characteristics on surfaces. Furthermore, to address dynamic lighting scenes, we propose the HNGD (Hierarchical Normal Gradient Densification) module to overcome the limitations of sparse Gaussian representation. Specifically, we dynamically adjust the Gaussian density based on normal gradients. Experimental evaluations demonstrate that our method outperforms state-of-the-art (SOTA) methods in terms of visual quality and exhibits significant advantages in performance indicators. Codes are available at <https://wanghongsheng01.github.io/NieR/>.*

**Keywords:** Image rendering, 3D Gaussian, Lighting scene.

## Introduction

Recent advancements in 3D Gaussian Splatting technology (e.g., [4]) have led to significant improvements in visual quality and rendering efficiency by employing pixel-level Gaussian distributions for scene rendering. While

existing methods demonstrate excellence in numerous areas, they still struggle with dynamic environmental lighting changes [19], particularly when rendering challenging specular objects such as vehicles in autonomous driving scenarios. This often leads to distortion and blurriness, underscoring the ongoing challenge of achieving real-time and precise rendering of lifelike images in this field.

In nature, when lighting hits on the surface, both specular reflection (refs. [7, 34]) and diffuse reflection (refs. [36]) occur simultaneously. The core of 3D Gaussian rendering relies on using a continuous, anisotropic Gaussian function to process each point in an image (refs. [15, 33]). However, specular reflection is a precise process, where the direction of reflected light strictly depends on the angle at which the light hits the surface and the angle at which we observe it. The uniform smoothness of the Gaussian distribution makes it hard to accurately capture and represent this complex interaction of lighting. Additionally, specular surfaces typically exhibit high brightness contrast. Traditional 3D Gaussian rendering techniques struggle to effectively handle such high dynamic range scenes, resulting in overexposure in highlight areas or lack of details. In real-world environments, diffuse reflection is influenced not only by direct illumination but also by various factors such as ambient light and indirect light (e.g., reflected light). For diffuse reflection, accurately simulating the complex interactions between light sources and object surfaces is crucial, while 3D Gaussian rendering often focuses on enhancing visual effects through pixel-level Gaussian Splatting, failing to capture the complexity of scattered lighting.

The Normal-Based Lighting Scene Rendering model introduces an optimized Physically Based Rendering (PBR) method through lighting decomposition to enhance the color synthesis and densification strategy of 3D Gaussian, which was able to improve the rendering quality and realism of 3D Gaussian for specular objects. Firstly, we propose Light Decomposition 3D Gaussian (LD) to optimize the color synthesis method by decomposing the outgoing

<sup>\*</sup>These authors contributed equally to this work.

<sup>†</sup>Corresponding Author.

radiation into specular reflection and diffuse reflection by means of surface normals. We also introduce the specular reflection coefficient attribute to consider the degree of lighting reflection by materials, significantly enhancing the accuracy of rendering specular objects and the realism of lighting effects. However, due to the sparse Gaussian representation of scenes in 3D Gaussian, significant large displacement changes will occur during optimization. Describing bright details poses difficulty in capturing ample lighting information, particularly for highlights, making it challenging to ensure the overall quality of rendered images. To address this deficiency, we introduce the Hierarchical Normal Gradient Densification (HNGD). NieR can adaptively adjust the density of Gaussian points based on the changing gradients in the scene further improving the accuracy of rendering specular reflective objects and effectively addressing the limitations of traditional sparse Gaussian representations in capturing highlights details.

In summary, our main contributions are as follows:

- We proposed a lighting decomposition module LD to optimize the color synthesis method of 3D Gaussian.
- We introduced hierarchical normal gradient densification to address the issue of insufficient lighting information caused by sparse Gaussian points, further improving the rendering quality of specular objects.
- Extensive experiments validate that the combination of LD and HNGD methods can significantly enhance the details of scene rendering. Especially in handling specular reflections and complex lighting scenarios, our method achieves the best visual quality compared with those of the state-of-the-art (SOTA) methods.

## 1. Related Work

### 1.1. Traditional Physical Rendering Methods

Traditional physical reconstruction techniques [6, 37, 38, 42] strive to achieve highly realistic image rendering by finely modeling the complex physics of the interaction between material surfaces and light. In the initial stages, techniques such as the Lambertian [23] reflection model are based on the assumption of ideal diffuse reflection surfaces to achieve basic simulation of lighting effects. However, this model is inadequate for representing more complex phenomena such as highlights and anisotropic reflection. Therefore, the field introduced the Phong [24] reflection model and its improved version, the Blinn-Phong model, which approximate lighting effects using empirical formulas and consider both reflected light and material properties, significantly enhancing the ability to simulate lighting phenomena.

A pivotal role in this domain is held by the Bidirectional Reflectance Distribution Function [3, 35] (BRDF), quantifying the scattering or reflection behavior of spe-

cific incident and reflected light directions, thereby offering a theoretical foundation for simulating diverse surface characteristics. The Cook-Torrance model [citation needed], as an application, combines microfacet theory and the Fresnel effect to provide accurate lighting simulations for rough surfaces, while the Disney Principled BRDF [citation needed] simulates a wide range of materials with controllable parameters (*e.g.*, base color, metallicity, and roughness), which further enhances fidelity and richness of detail in image rendering.

### 1.2. implicit Rendering Methods

In contrast to traditional techniques, Neural Radiance Fields (NeRF) [21] leverage implicit Multilayer Perceptrons (MLPs) to render density and view-dependent colors from three-dimensional coordinates and viewing directions [14, 29, 39], advancing the development of differentiable volume rendering. In an extended study, a method is introduced for reconstructing high-quality geometric shapes and complex spatially varying BRDFs from a sparse image datasets, which integrating and refining geometry and BRDF by optimizing the latent space of a reflectance network, significantly reducing photometric differences. This approach also demonstrates practicality on sparse image datasets and significant progress in acquiring high-quality shapes and appearances. While NeRF exhibits unprecedented capabilities in reconstructing and rendering high-quality 3D scenes, it still faces the challenge of high computational costs in practical applications. [2, 12, 28, 30]

### 1.3. 3D Gaussian-based Methods

3D Gaussian Splatting [13] directly constructs 3D scenes by employing a three-dimensional Gaussian distribution, which significantly surpasses NeRF in rendering speed, particularly in the optimization process of light synthesis. Relightable [9] 3D Gaussian extends its functionality by combining 3D Gaussian points with normals, global and local quantities of incident light, and BRDF parameters, refined through physics-based differentiable rendering. This method also introduces an efficient visibility baking technique based on boundary volume hierarchies [20, 31], enabling real-time rendering and precise shadow effects. GaussianShader [11] introduces a simplified shading function tailored for 3D Gaussians, focusing on enhancing the rendering effects with scenes containing reflective surfaces while ensuring the efficiency of both training and rendering processes. By introducing an innovative normal estimation framework, this method tightly associates shading parameters with the true geometric form of 3D Gaussians, significantly improving the rendering quality, especially when rendering specular objects. Nevertheless, challenges still exist in accurately simulating light reflection effects, particularly in precisely capturing specular reflective objects,

occasionally leading to visual distortion and deformation of reflective objects in reconstructed scenes. Our method takes into account the differential effects brought about by different reflections in the scene and separately handles the reflective effects in the scene.

## 2. Methodology

### 2.1. Preliminary

**3D Gaussian Splatting.** Unlike traditional NeRF (Neural Radiance Fields) implicit neural networks, 3D Gaussian employs several three-dimensional Gaussian distributions to explicitly represent scenes, thereby effectively improving the rendering speed. In mathematics, a three-dimensional Gaussian distribution can be defined by the following formula:

$$G(x) = \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \quad (1)$$

Among them,  $\mu$  is the center position, and  $\Sigma$  is the 3D covariance matrix. In addition, 3D Gaussian Splitting incorporates information on opacity  $o$  and color  $c$  into each Gaussian distribution for subsequent rendering work. In our data, color  $c$  is represented by a fourth-order spherical harmonic function [5, 25]. Subsequently, we project the three-dimensional Gaussian onto a two-dimensional plane to render the image.

$$\Sigma' = JW\Sigma W^T J^T \quad (2)$$

Among them,  $\Sigma'$  is the 2D covariance matrix in the camera coordinates,  $J$  is the Jacobian matrix [40] of the affine approximation projection transformation, and  $W$  is the world-to-camera perspective transformation matrix. Given the position of pixel  $x$ , we solve for its final color by mixing  $N$  ordered points that overlap with the pixel.

$$c = \sum_{i \in N} T_i \alpha_i c_i \quad \text{with} \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (3)$$

Here,  $\alpha$  is obtained by multiplying the contribution of opacity  $o$  with the 2D covariance calculated by  $\Sigma'$  and the image's spatial pixel coordinates.

**Rendering equation** The rendering equation [32] is used to describe the interaction between light and surfaces and is widely used in lighting models in the physical world. It calculates the radiance at surface points  $x$  along the observation direction:

$$L_o(\omega_o, x) = \int_{\Omega} f(\omega_o, \omega_i, x) L_i(\omega_i, x) (\omega_i \cdot n) d\omega_i \quad (4)$$

Here,  $x$  is the surface point,  $n$  is the surface normal vector,  $f$  is the BRDF (Bidirectional Reflectance Distribution Function) that describes the spatial reflection characteristics

of the surface, and  $L_i$  is the ray incident from direction  $\omega_i$  onto point  $x$ , integrated over the hemisphere in direction  $\Omega$  towards  $\omega_i$ . When rendering 3D scenes, it is important to choose the appropriate method for  $f$  and  $L_i$  due to the complex properties of materials and the differences in lighting in different scenes.

### 2.2. LD: Light Decomposition 3D Gaussian

In the field of three-dimensional scene rendering, achieving highly realistic visual effects not only mandates accurate simulation of lighting effects but also demands a thorough consideration of the differences between diffuse [36] reflection and specular [7] reflection lighting phenomena. This requirement is particularly prominent when dealing with materials that exhibit different lighting response characteristics, such as rough wooden surfaces and smooth metallic surfaces. However, current methods often overlook these details, leading to significant discrepancies between the rendered results on complex material surfaces and the real world.

Therefore, when dealing with complex lighting scenarios, the challenge lies in how to accurately capture the different effects of diffuse reflection and specular reflection while ensuring computational efficiency, and take full account of the interaction between the surfaces, changes in lighting conditions and the diversity of material properties [41]. In this light, we propose an efficient three-dimensional scene lighting framework that integrates diffuse reflection, specular reflection, and incident light models. In this model, we employ a simplified incident lighting model  $c$  to represent the impact of light on the scene.

$$c = V \cdot L_{global} + L_{local} \quad (5)$$

To enhance the realism and naturalness of the rendering effects, we have refined the simulation of diffuse reflection and specular reflection. Firstly, by combining spherical harmonics and the cosine of the angle between the light ray and the surface normal, we carefully adjust the contribution of diffuse reflection to the lighting effects to ensure improved naturalness and realism of the lighting effects from different viewing angles. Additionally, we have further refined the application of spherical harmonics in predicting specular reflection by more precisely controlling the direction of lighting reflection and considering the influence of surface properties on reflected lighting, that leads to a more realistic specular highlights. Furthermore, considering the differences in how different materials react to lighting, we have carefully adjusted the balance between diffuse reflection and specular reflection in the rendering process. Specifically, we utilize  $c_0$  to calculate the final rendered color.

$$c_0 = (1 - a) \cdot shc \cdot \cos\theta + a \cdot shc \quad (6)$$

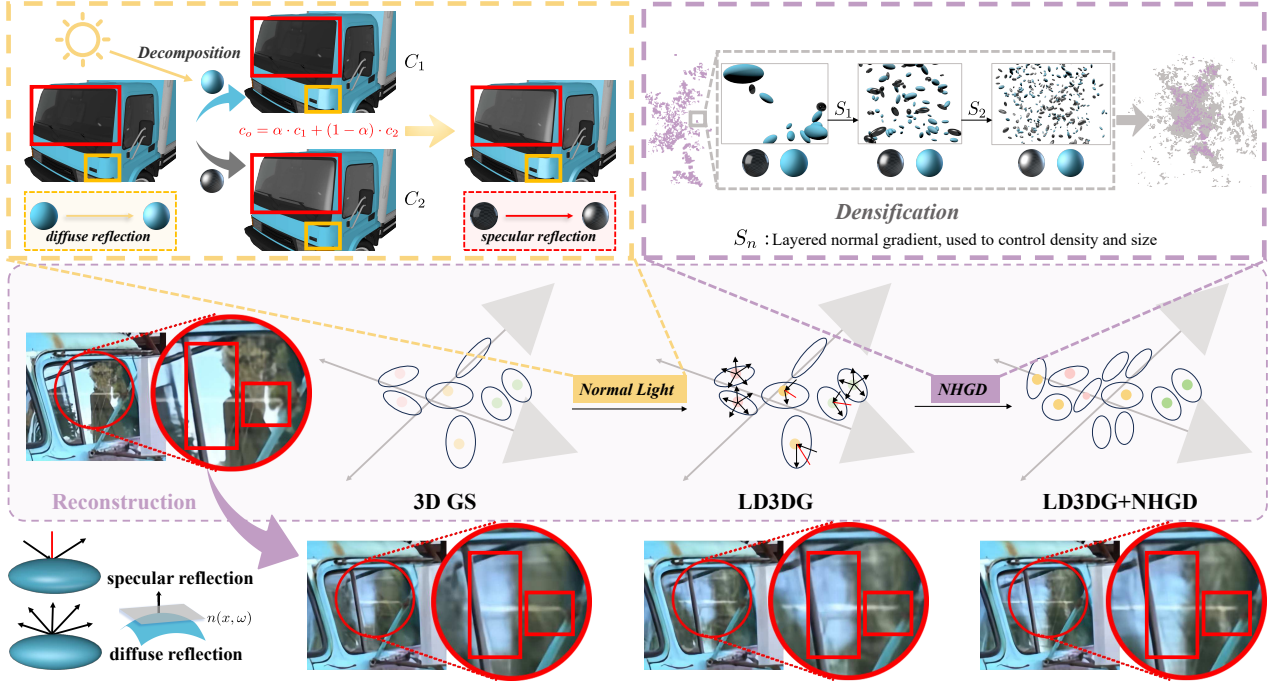


Figure 1. The pipeline of our method, with the LD module on the left and the HNGD module on the right

Where  $a$  is the weighting factor used to adjust the ratio of the two reflection effects in the final result, when  $a = 1$  means that favor towards full specular reflection and  $a = 0$  indicates the pure diffuse reflection;  $shc$  is the spherical harmonic coefficient used to calculate the colors for diffuse and specular reflections;  $\cos(\theta)$  takes into account the angle between the direction of the lighting and the surface normal, affecting the intensity of diffuse reflection.

### 2.3. HNGD : Hierarchical Normal Gradient Densification

The introduction of the spherical harmonic function allows us to quickly estimate the light intensity in any direction from precomputed coefficients, providing an effective means of approximating the lighting in complex scenes. However, when these Gaussian points are too sparse, the light prediction may produce large errors, especially in specular reflections and highlight details, due to the lack of sufficient local light information. To reduce this error and optimize the lighting prediction, we introduce HNGD (Hierarchical Normal Gradient Densification) for finer lighting control to achieve higher quality rendering.

**Hierarchical Gradient Densification.** Under empirical study, we found that simply increasing the number of Gaussian points does not necessarily improving result accuracy. On the contrary, this approach may lead to unnecessary rendering errors due to the mutual interference between Gaus-

sian sampling distributions. In order to solve this problem, we propose a hierarchical gradient densification approach, which allows us to design different sizes of gradient deflators for specific needs and features of the scene to further improve the effectiveness of the densification process.

Specifically, during the experimental process, when the fused gradient values of a certain region within the Gaussian processed by LD reach the threshold for the next level, we proceed with a splitting operation on the relevant Gaussian. This operation involves dividing each Gaussian into two, aiming to refine the distribution of Gaussian. By controlling the number of Gaussian in a manner most beneficial for the current region's lighting representation, this approach allows for a more precise capture of lighting details in complex scenes.

**Normal Gradient Fusing.** We adjust the densification of the Gaussian points by using the gradient information. However, since 3D Gaussian are more sparsely distributed in space compared to traditional point cloud data, the geometric gradient of the scene does not realistically represent the shape changes. For this reason, we perform a weighted fusion of geometric and normal vector gradients and use adjustable weighting coefficients to meet different experimental needs.

First, we use the center difference method for each volume point  $(x, y, z)$  in the scene to compute its gradient components in the three directions X, Y, and Z.



$$\begin{aligned}
G_x &= V(x+1, y, z) - V(x-1, y, z) \\
G_y &= V(x, y+1, z) - V(x, y-1, z) \\
G_z &= V(x, y, z+1) - V(x, y, z-1)
\end{aligned} \tag{7}$$

where  $V(x, y, z)$  is the density value located at  $(x, y, z)$ . Next, we normalize the gradient vector  $G_{xyz} = (G_x, G_y, G_z)$  to obtain the unit normal vector  $\hat{G}_{xyz} = \frac{G_{xyz}}{\|G_{xyz}\|}$ ,  $\|G_{xyz}\| = \sqrt{G_x^2 + G_y^2 + G_z^2}$ . Further, we compute the normal vector gradient  $G_{norm}$  to capture the gradient information induced by the change of normal vector:

$$\begin{aligned}
\Delta \hat{N}_x &= |\hat{N}(x+1, y, z) - \hat{N}(x, y, z)| \\
\Delta \hat{N}_y &= |\hat{N}(x, y+1, z) - \hat{N}(x, y, z)| \\
\Delta \hat{N}_z &= |\hat{N}(x, y, z+1) - \hat{N}(x, y, z)|
\end{aligned} \tag{8}$$

Combining the computed normal vector gradient components  $\Delta \hat{N}_x$ ,  $\Delta \hat{N}_y$ , and  $\Delta \hat{N}_z$ , we obtain the normal vector gradient vector  $\Delta \hat{N} = (\Delta \hat{N}_x, \Delta \hat{N}_y, \Delta \hat{N}_z)$ . The modulus of  $\Delta \hat{N}$  is taken to obtain a scalar representation of the normal vector gradient strength:  $G_{norm} = \|\Delta \hat{N}\|$ . We use the following formula for weighted fusion:

$$Grad = (1 - \omega) \cdot \left( \frac{G_{xyz}}{denom} \right) + \omega \cdot \left( \frac{G_{norm}}{denom} \right) \tag{9}$$

where  $denom$  is the normalization factor and  $\omega$  is the fusion weight, between 0 and 1, used to regulate the relative importance of the scene gradient (geometric gradient) (accumulation of)  $G_{xyz}$  and the normal vector gradient (accumulation of)  $G_{norm}$  in the final fusion gradient  $Grad$ .

### 3. Experimentation

#### 3.1. Training Detail

Many advanced methods exist for reconstructing and rendering real-world scenes (e.g., [16, 17, 27, 39]). We discuss related work in detail in Section 2. We implemented our method using PyTorch and CUDA, leveraging a single NVIDIA V100 GPU. Following the training protocol of 3D Gaussian Splatting (3D GS) [27], all models were trained for 30,000 iterations, with each scene reconstruction requiring less than 3 hours. We extend 3D GS by introducing additional parameters to control the synthesis of diffuse and specular reflections, enabling fine-grained control over lighting components during the rendering process.

#### 3.2. Results and Evaluation Results

To assess NieR’s ability to render scenes with realistic lighting and fine details, we compare it against Gaussian Splatting as the baseline method. We employ established quantitative metrics, including Peak Signal-to-Noise Ratio (PSNR) [10], Learned Perceptual Image Patch Similarity

(LPIPS) [26], and Structural Similarity Index (SSIM) [18]. These metrics evaluate the similarity between rendered images and ground truth data, quantifying visual fidelity and detail preservation. We compare our method with Gaussian Splatting, Mip-NeRF360 [1], InstantNGP, and Plenoxels on seven scenes from the Mip-NeRF360 dataset [1] and two scenes from the Tanks & Temples dataset [2017]. This diverse set of scenes includes both indoor environments with stable lighting (living rooms, kitchens) and outdoor scenarios characterized by complex lighting conditions (playgrounds, gardens). Quantitative comparisons are presented in Table 1 and Table 2, while Figure 2 showcases qualitative comparisons. NieR demonstrates competitive performance against baselines across various scenes, particularly in scenarios characterized by **complex lighting**. Notably, our method achieves substantial improvements in both quantitative metrics and visual quality under such challenging conditions, highlighting its effectiveness in rendering scenes with **realistic lighting and intricate details**.

**Comparisons with other Methods.** As one of the most competitive NeRF methods, the Mip-NeRF360 method estimates color and brightness by integrating across anti-aliased cone regions of lighting. However, it does not consider the reflective properties of different materials, which leads to suboptimal performance in scenes with specular objects. Because the 3D Gauss Splatting method represents the scene using 3D Gaussians and performs interleaved optimization and density control on these Gaussians, providing an accurate representation of the scene. Nevertheless, its reliance solely on spherical harmonics for lighting processing results in coarse rendering effects in complex lighting environments. Our model introduces a lighting synthesis module and a hierarchical densification module, offering a more effective rendering method for complex lighting scenes and maintaining strong competitiveness in scenes with relatively stable lighting. As shown in Table 1, our method achieves the highest level across all metrics. Compared to Mip-NeRF, our method improves SSIM by 8.7 percent, and compared to 3D Gauss Splatting, it improves SSIM by 4.7 percent. The result indicates that our method not only effectively reproduces realistic lighting variations but also reduces the interference of noise on image quality, and realistic the restoration of picture colors.

To further explore the performance of our method in different scenes, we present the comparative results based on the PSNR metric in Table 2. It can be observed that our method shows significant improvements in outdoor scenes like gardens and in scenes containing specular objects such as trains and cars. Especially in the train scene with a large number of specular objects, our method outperforms Gaussian Splatting by 25.02 percent in terms of PSNR. This indicates a substantial enhancement in rendering specular object scenes with our proposed method. Additionally, our



Figure 2. Shows the comparison between our method and previous methods, as well as the corresponding ground truth images extracted from the preserved test views. The scenes from top to bottom are the stump, bonsaimcounter, room, garden, kitchen, and cycle from the Mip NeRF360 dataset; And trucks and trains from Tanks and Temples. Non obvious quality differences are marked with arrows/embedded markings.

method surpasses our baseline method Gaussian Splatting in all scenes. While Mip-NeRF can produce high-quality results in some scenes, it requires an average training time of 48 hours on the same device. In contrast, our method has an average training time of 1 hour and 47 minutes. It shows that our method not only achieves high-quality rendering effects but also maintains **low memory consumption and rendering time**.

**Qualitative Results.** InstantNGP [22], Plenoxels [8], 3D Gaussian [13], and Mip-NeRF360 [1] rely solely on the viewing direction for rendering, resulting in the loss of

some lighting information, which will cause the potential boundary blurring and artifacts on the rendered object surfaces, as exemplified in the truck scene. While frameworks such as Mip-NeRF360 can render the outline of a building on the truck’s glass, it fails to capture details regarding the floors and boundaries of the building. To address this deficiency of lost lighting information, our framework introduces specular and diffuse reflection information present on the object surface. Our comparison shows that we can more accurately reproduce the lighting changes and the clear outlines of building on the truck’s glass. 3D Gaussian utilizes



Table 1. The average metrics. We report the results of our model and other models compared on the Mip NeRF360 dataset and Tanks & Temples dataset in all scenarios. The best number is highlighted in bold.  $\uparrow$  indicates that the larger the indicator, the better,  $\downarrow$  indicates that the smaller the indicator, the better.

Model	Mip NeRF360				Tanks & Temples			
	SSIM $\uparrow$	PSNR(dB) $\uparrow$	LPIPS $\downarrow$	Train	SSIM $\uparrow$	PSNR(dB) $\uparrow$	LPIPS $\downarrow$	Train
Plenoxels	0.686	23.067	0.386	25m49s	0.7180	21.08	0.343	25m5s
InstantNGP	0.724	25.381	0.336	5m37s	0.7225	21.71	0.329	5m26s
Mip-NeRF	0.823	27.585	0.219	48h	0.7575	22.21	0.256	48h
Gaussian-Splatting	0.863	27.455	0.181	41m33s	0.8400	23.14	0.182	26m54s
ours	<b>0.910</b>	<b>30.524</b>	<b>0.153</b>	1h36m	<b>0.9170</b>	<b>28.18</b>	<b>0.132</b>	1h59m

Table 2. Our model and other comparison models have PSNR metrics for all scenarios in the Mip NeRF360 dataset and Tanks/Temples dataset.

dataset	Stump	Room	Kitchen	Garden	Counter	Bonsai	Bicycle	truck	train	average
Plenoxels	20.677	27.574	23.433	23.503	23.636	24.710	21.897	23.231	18.944	23.067
InstantNGP	23.625	29.269	28.547	25.068	26.438	30.337	22.192	23.260	20.170	25.434
Mip-NeRF360	26.356	<b>31.467</b>	31.988	26.875	29.447	<b>33.396</b>	24.304	24.911	19.522	27.585
Gaussian Splatting	26.545	30.632	30.316	27.410	28.692	31.980	25.245	25.186	21.097	27.455
ours	<b>30.984</b>	30.886	<b>33.484</b>	<b>29.661</b>	<b>32.573</b>	33.369	<b>27.398</b>	<b>29.993</b>	<b>26.376</b>	<b>30.524</b>

Gaussian distribution for pixel smoothing during scene

rendering, leading to halo-like diffusion on surfaces under complex lighting conditions. For example, the "counter" scene in Figure 2, which smooth glare spots appear on the inner wall of the basin, lacking the expected gloss and brightness in the central points. In order to mitigate the impact of Gaussian smoothing, our framework employs densification of Gaussian in areas with stronger lighting, resulting in brighter light spots on the basin surface. In summary, these scenes demonstrate the advantages of our framework in reproducing realistic lighting environments. For instance, when generate color characteristics of materials such as metal and glass under lighting, our framework provides surfaces with more gloss and brightness, approaching closer to the ground truth.

### 3.3. Ablation Studies

We segregate the algorithm into distinct modules and further decomposed each module. Simultaneously, we design a set of experiments to investigate the significance of each module and its corresponding submodules on the model. Ablation studies will be conducted on the LD and HNGD modules.

#### 3.3.1 Light Decomposition

**Comparisons with Baseline** It would fail to accurately reconstruct details in **scenarios involving diffuse or specular reflections**, due to the limitation of 3D Gaussian Splat-

ting, which relies solely on spherical harmonics for handling surface lighting. To validate the ability of the lighting decomposition module to recreate different lighting conditions in a scene, we conducted experiments on the train and truck scenarios, as shown in Figure 3 (left column shows the rendering results of 3D Gaussian Splatting, while the right column displays our results). Firstly, Compared with images (a) and (b), our framework eliminates the shadow artifacts cast by deep green trees on the white steel frame, revealing clearer outlines of the steel structure. Secondly, for the windows of the house shown in images (c) and (d), representing reflections on specular material objects under complex lighting conditions, our module not only reconstructs the tree information with real light and shadow changes in the left window but also renders bright lighting information distinct from other objects in the right window. Finally, after multiple comparisons of rendered images from different perspectives, it was observed that our framework can **faithfully reproduce the color characteristics** of object surfaces, whether in scenarios involving diffuse reflections or more pronounced specular reflections, which proves the effectiveness of integrating diffuse reflection, specular reflection, and incident light models as input information.

**Quantitative Results** To quantitatively evaluate the rendering capability of the LD module in addressing dynamic lighting perception, we conducted an ablation experiment on the original scheme by removing the LD module to ob-



Figure 3. The top left image (a) shows the result of removing diffuse light intensity, while the top right image (b) shows the complete result. The left image (c) shows the result of removing specular reflection, while the right image (d) shows the complete result.

serve changes in metrics. As shown in Table 3, **after the LD module is removed, the PSNR and SSIM metrics decreased by 5.5 percent and 12.8 percent, while the LPIPS metric increased by 17.6 percent.** The significant changes in LPIPS and SSIM metrics resulting from the removal of the LD module, indicate that the introduction of lighting information can produce images closer to the ground truth in terms of human visual perception, and accurately simulating the color characteristics of object surfaces under complex lighting conditions.

Table 3. Ablation studies on NeRF360 datasets using our proposed methods.

Model	PSNR(dB) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Ours w/o LD and HNGD	0.878	25.186	0.147
Ours w/o HNGD	0.929	28.886	0.125
ours	<b>0.945</b>	<b>30.19</b>	<b>0.090</b>

**Analysis on the Lights** To validate the capability of the LD module in simulating real lighting scenarios, we compared the scene without the diffuse reflection effect and without the specular reflection effect by adjusting the parameters in the LD module. It was observed that when removing either the diffuse reflection effect or the specular reflection effect, the metrics of the model will decline. This further emphasizes the effectiveness and importance of the LD module in simulating lighting effects (especially specular reflection). Additionally, by comparing the changes in the LPIPS metric during the ablation process, we found that the complete LD module better preserves the contours and colors of buildings in specular reflections, making the rendered images closer to the original ground truth.

### 3.3.2 Hierarchical Densification

**Effects of HNGD module** Furthermore, we explored the role of the hierarchical densification module in addressing lighting variations. In Figure 4, we present visual comparison results of three methods: without hierarchical densification strategy (b), sparse hierarchical densification strategy (c), and dense hierarchical densification strategy (d). In the experiment, we introduced a predefined threshold range [1, 1.5, 2, 2.5, 3, 3.5] as the discriminative thresholds for the sparse splitting hierarchical strategy, with values greater than 3.5 being considered for the dense splitting hierarchical strategy. It was observed that the method without the addition of the hierarchical densification module struggled to render high-quality results, as seen in image (b). Compared with the ground truth revealed issues such as the fogging of the station skeleton in the upper left corner and the trees to the right, significant blurriness at the boundaries of mapped trees, subtle variations in brightness, unclear color patches in the darker leaf area at the bottom right of the mirror, and a loss of hierarchical relationships with shadows.

Table 4. Ablation studies on the impact of lighting

Model	PSNR(dB) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Light Synthesis w/o Diffuse Reflection	0.928	28.791	0.129
Light Synthesis w/o specular	0.905	26.771	0.161
Full Light Synthesis	<b>0.929</b>	<b>28.88</b>	<b>0.125</b>

Our framework effectively addresses this issue by employing the hierarchical densification module, which can selectively generate more Gaussian in regions with strong brightness variations. Compared with Figure 4(a), (b), and (c) reveals that even with the use of sparse hierarchical densification strategy, we can accurately render a station with a brighter skeleton and surrounding leaves, as well as bring out the tree leaves in the lower right area of the mirror from the dark shadow. Additionally, compared with Figure 4(a), (c), and (d), we observe that as the densification level of Gaussian points increases, the trees on the windows exhibit clearer brightness variations against a bright sky background, and capture more light spots projected between the leaves. Analysis of the above comparative experimental results, we have a significant observation that lighting scenes without using our HNGD module, there is often a phenomenon of objects blending together and resulting in color fog. However, when we adjust the Gaussian density using HNGD, we render scenes with realistic hierarchical details, demonstrating the effectiveness of our method in achieving realism in rendering.

To quantitatively evaluate the impact of the hierarchical densification module on enhancing the rendering capability for dynamic lighting perception, we demonstrated the gains brought by this module in terms of metrics by removing it



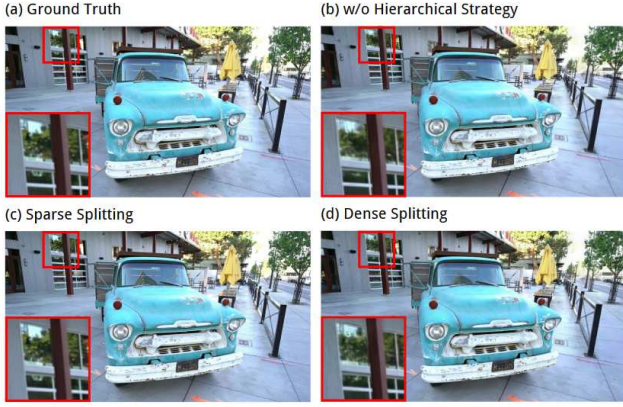


Figure 4. From left to right (a) shows Ground Truth, (b) shows no hierarchical densification result, (c) shows sparse splitting hierarchical densification result, and (d) shows dense splitting hierarchical densification result.

from the existing method. As shown in Table 5, our method shows improvements in multiple metrics compared to the method without the densification module. Specifically, the method using dense hierarchical strategy experiences a 30.9 percent decrease in LPIPS metric, demonstrate the restoration capability of this module in lighting scenes.

Table 5. Ablation studies on the effect of hierarchical strategy to densification

Model	PSNR(dB) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
ours w/o hierarchical strategy	0.928	28.791	0.129
ours with sparse splitting hierarchical strategy	0.938	29.592	0.106
ours with dense splitting hierarchical strategy	<b>0.945</b>	<b>30.195</b>	<b>0.090</b>

**Effects of Different Hierarchical Strategies** To validate the influence of our hierarchical densification module on perceiving different lighting variations through hierarchical adjustments, we dynamically adjust the densification levels in the lighting variation module to observe the effects of different settings in handling various lighting scenarios. The image in the red box in Figure 4 shows that, although dense densification has a better build-up of the structure of reflective objects such as the base station, there is no significant change for the deep shadow parts of the tree leaves compared to sparse densification. This result guides us to adaptively adjust the densification strategy based on the strength of lighting in the scene to save unnecessary time and memory overhead. In conclusion, as the level of densification perceived by the module increases in areas with significant lighting, it notably enhances the presence of stronger brightness variations in corresponding positions in the rendered images.

## 4. Conclusion

This paper presents NieR, a novel rendering method that incorporates both diffuse and specular reflection information. NieR addresses regions with significant illumination variations through a hierarchical densification strategy, leading to demonstrably improved rendering accuracy and realism. This is particularly evident for objects with specular properties and complex scenes. Our work paves the way for incorporating lighting information into 3D scenes rendered using the Gaussian field representation. Future work will focus on exploring NieR’s applicability in dynamic lighting scenarios and investigating its potential integration with other rendering techniques for further performance gains.

## References

- [1] Verbin D et al Barron J T, Mildenhall B. Mip-nerf 360: Unbounded anti-aliased neural radiance fields[c]. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 5, 6
- [2] Poiesi F Bortolon M, Del Bue A. Vm-nerf: Tackling sparsity in nerf with view morphing[j]. In *Acm Transactions on Graphics (TOG)*, page 2210.04214, 2022. 2
- [3] Nauyoks S E et al Butler S D, Ethridge J A. Experimentally validated modification to cook-torrance brdf model for improved accuracy[c]. In *Earth Observing Systems XXII. SPIE*, pages 10402: 321–328, 2017. 2
- [4] Wang W Chen G. A survey on 3d gaussian splatting[j], 2024. 1
- [5] Dai F et al Dai F, Xu Y. Spherical harmonics[j]. In *Approximation theory and harmonic analysis on spheres and balls*, pages 1–27, 2013. 3
- [6] Losasso F. Surface reflection models[j]. In *NVIDIA Corporation*, 2004. 2
- [7] Adelson E H Fleming R W, Torralba A. Specular reflections and the perception of shape[j]. In *Journal of vision*, pages 10–10, 2004,4(9). 1, 3
- [8] Tancik M et al Fridovich-Keil S, Yu A. Plenoxels: Radiance fields without neural networks[c]. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 6
- [9] Lin Y et al Gao J, Gu C. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing[j], 2023. 2
- [10] Modeling indirect illumination for inverse rendering. Yuanqing zhang, jiaming sun, xingyi he, huan fu, rongfei jia, and xiaowei zhou. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, page 18643–18652, 2022.6. 5
- [11] Liu Y et al Jiang Y, Tu J. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces[j], 2023. 2
- [12] Ye-Bin M et al Jun-Seong K, Yu-Ji K. Hdr-plenoxels: Self-calibrating high dynamic range radiance fields[c]. In *European Conference on Computer Vision*, pages 384–401, 2022. 2
- [13] Leimkühler T et al Kerbl B, Kopanas G. 3d gaussian splatting for real-time radiance field rendering[j]. In *ACM Transactions on Graphics*, page 42(4), 2023. 2, 6
- [14] Jonathan T Barron Sofien Bouaziz Dan B Goldman Steven M Seitz Keunhong Park, Utkarsh Sinha and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, page 5865–5874, 2021.2. 2
- [15] Sun X et al Lee B, Lee H. Deblurring 3d gaussian splatting[j], 2024. 1
- [16] Yang Z et al Li M, Tao J. Human101: Training 100+ fps human gaussians in 100s from 1 view[j], 2023. 5
- [17] Leibe B et al Luiten J, Kopanas G. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis[j], 2023. 5
- [18] Setiadi D R I M. Psnr vs ssim: imperceptibility quality assessment for image steganography[j]. In *Multimedia Tools and Applications*, pages 8423–8444, 2021,80(6). 5
- [19] Put J Michiels N. Interactive relighting of virtual objects under environment lighting[c]. In *SCITEPRESS*, pages 220–228, 2015,2. 1
- [20] Put J Michiels N. An adaptive granular representative volume element model with an evolutionary periodic boundary for hierarchical multiscale analysis[j]. In *International Journal for Numerical Methods in Engineering*, pages 2239–2253, 2021, 122(9). 2
- [21] Tancik M et al Mildenhall B, Srinivasan P P. Nerf: Representing scenes as neural radiance fields for view synthesis[j]. In *Communications of the ACM*, pages 65(1): 99–106, 2021. 2
- [22] Schied C et al Müller T, Evans A. Instant neural graphics primitives with a multiresolution hash encoding[j]. In *ACM Transactions on Graphics (ToG)*, pages 1–15, 2022, 41(4). 6
- [23] Nayar S K Oren M. Generalization of lambert’s reflectance model[c]. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246, 1994. 2
- [24] Tan P. Phong reflectance model[j]. In *Computer Vision: A Reference Guide*, pages 1–3, 2020. 2
- [25] Green R. Spherical harmonic lighting: The gritty details[c]. In *Archives of the game developers conference*, page 4, 2003,5. 3
- [26] Alexei A Efros Eli Shechtman Richard Zhang, Phillip Isola 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*, page 586–595, 2018.6. 5
- [27] Alexei A Efros Eli Shechtman Richard Zhang, Phillip Isola and Oliver Wang. 4d gaussian splatting for real-time dynamic scene rendering[j], 2023. 5
- [28] Stamminger M Rückert D, Franke L. Adop: Approximate differentiable one-pixel point rendering[j]. In *Acm Transactions on Graphics (TOG)*, pages 1–12, 2022,41(4). 2
- [29] Balakrishnan G et al Saragadam V, Tan J. Miner: Multiscale implicit neural representation[c]/european conference on computer vision. In *Cham: Springer Nature Switzerland*, pages 318–333, 2022. 2

- [30] Kolodiazhna O Savin V. Adapting neural radiance fields (nerf) to the 3d scene reconstruction problem under dynamic illumination conditions[j]. In *Cybernetics and Systems Analysis*, pages 910–918, 2023,59(6). 2
- [31] Bade A Sulaiman H A. Bounding volume hierarchies for collision detection[j]. In *Computer Graphics*, pages 5772–2386, 2012, 10. 2
- [32] Kajiya J T. The rendering equation[c]. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. 3
- [33] Nießner M Thies J, Zollhöfer M. Deferred neural rendering: Image synthesis using neural textures[j]. In *Acm Transactions on Graphics (TOG)*, pages 1–12, 2019, 38(4). 1
- [34] Sparrow E M Torrance K E. Theory for off-specular reflection from roughened surfaces[j]. In *Josa*, pages 1105–1114, 1967, 57(9). 1
- [35] Dong A Wang H, Zhang W. Measurement and modeling of bidirectional reflectance distribution function (brdf) on material surface[j]. In *Measurement*, pages 3654–3661, 2013, 46(9). 2
- [36] Oren M Wolff L B, Nayar S K. Improved diffuse reflection models for computer vision[j]. In *International Journal of Computer Vision*, pages 55–71, 1998, 30. 1, 3
- [37] Yoo J S et al Woo S M, Lee S H. Improving color constancy in an ambient light environment using the phong reflection model[j]. In *IEEE Transactions on Image Processing*, pages 1862–1877, 2017, 27(4). 2
- [38] Gotanda Y. Beyond a simple physically based blinn-phong model in real-time[c]. In *ACM SIGGRAPH*, page 10, 2012. 2
- [39] Lee G H Yan Z, Li C. Nerf-ds: Neural radiance fields for dynamic specular objects[c]. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 2, 5
- [40] Hsieh C J. Recurjac Zhang H, Zhang P. An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5757–5764, 2019, 33(01). 3
- [41] Wimmer M Zsolnai-Fehér K, Wonka P. Gaussian material synthesis[j], 2018. 3
- [42] Kurt M Öztürk A, Bilgili A. Polynomial approximation of blinn-phong model[c]. In *TPCG*, pages 55–61, 2006. 2