Lighting Every Darkness with 3DGS: Fast Training and Real-Time Rendering for HDR View Synthesis

Xin Jin[1,2*] **Pengyi Jiao**[1*] **Zheng-Peng Duan**[1] **Xingchao Yang**[2]
Chongyi Li[1] **Chun-Le Guo**[1†] **Bo Ren**[1†]
[1] VCIP, CS, Nankai University [2] MEGVII Technology
https://srameo.github.io/projects/le3d

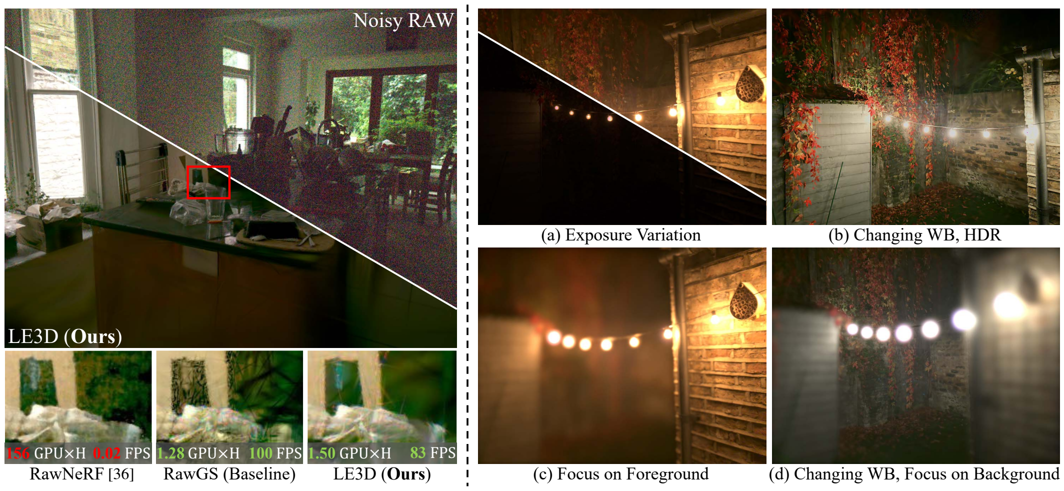


Figure 1: LE3D reconstructs a 3DGS representation of a scene from a set of multi-view noisy RAW images. As shown on the left, LE3D features *fast training and real-time rendering* capabilities compared to RawNeRF [36]. Moreover, compared to RawGS (a 3DGS [25] we trained with RawNeRF's strategy), LE3D demonstrates superior noise resistance and the ability to represent HDR linear colors. The right side highlights the variety of *real-time downstream tasks* LE3D can perform, including (a) exposure variation, (b, d) changing White Balance (WB), (b) HDR rendering, and (c, d) refocus.

Abstract

Volumetric rendering-based methods, like NeRF, excel in HDR view synthesis from RAW images, especially for nighttime scenes. They suffer from long training times and cannot perform real-time rendering due to dense sampling requirements. The advent of 3D Gaussian Splatting (3DGS) enables real-time rendering and faster training. However, implementing RAW image-based view synthesis directly using 3DGS is challenging due to its inherent drawbacks: 1) in nighttime scenes, extremely low SNR leads to poor structure-from-motion (SfM) estimation in distant views; 2) the limited representation capacity of the spherical harmonics (SH) function is unsuitable for RAW linear color space; and 3) inaccurate scene structure hampers downstream tasks such as refocusing. To address these issues, we propose **LE3D** (**L**ighting **E**very darkness with **3D**GS). Our method proposes Cone Scatter Initialization to enrich the estimation of SfM and replaces SH with a Color MLP to represent the RAW linear color space. Additionally, we introduce depth distortion and near-far regularizations to improve the accuracy of scene structure for downstream tasks. These designs enable LE3D to perform real-time novel view synthesis, HDR rendering, refocusing, and tone-mapping changes. Compared to previous volumetric rendering-based methods, LE3D reduces training time to 1% and improves rendering speed by up to 4,000 times for 2K resolution images in terms of FPS. Code and viewer can be found in https://srameo.github.io/projects/le3d.

[*] Equal Contribution. This project is done during Xin Jin's internship at MEGVII Technology.
[†] C. L. Guo and B. Ren are corresponding authors.

1 Introduction

Since the advent of Neural Radiance Fields (NeRF) [37], novel view synthesis (NVS) has entered a period of vigorous development, thereby advancing the progress of related applications in augmented and virtual reality (AR/VR). Existing NVS technologies predominantly utilize multiple well-exposed and noise-free low dynamic range (LDR) RGB images as inputs to reconstruct 3D scenes. This significantly impacts the capability to capture high-quality images in environments with low light or high contrast, such as nighttime settings or areas with stark lighting differences. These scenarios typically necessitate the use of high dynamic range (HDR) scene reconstruction techniques.

Existing HDR scene reconstruction techniques primarily fall into two categories and all are based on volumetric rendering: 1) using multiple-exposure LDR RGB images for supervised training [23], and 2) conducting training directly on noisy RAW data [36]. The first type of method is typically highly effective in well-lit scenes. However, in nighttime scenarios, its reconstruction performance is constrained due to the impact of the limited dynamic range in sRGB data [6]. While RAW data preserves more details in nighttime scenes, it is also more susceptible to noise. Therefore, RawNeRF [36] is proposed to address the issue of vanilla NeRF's lack of noise resistance. However, RawNeRF suffers from prolonged training times and an inability to render in real-time (a common drawback of volumetric rendering-based methods). This significantly limits the application of current scene reconstruction techniques and HDR view synthesis. Enabling real-time rendering for HDR view synthesis is a key step towards bringing computational photography to 3D world.

Recently, 3D Gaussian Splatting (3DGS) has garnered significant attention based on its powerful capabilities in real-time rendering and photorealistic reconstruction. 3DGS utilizes Structure from Motion (SfM) [41] for initialization and employs a set of 3D gaussian primitives to represent the scene. Each gaussian represents direction-aware colors using spherical harmonics (SH) functions and can be updated in terms of color, position, scale, rotation, and opacities through gradient descent optimization. Although 3DGS demonstrates its strong capabilities in reconstruction and real-time rendering, it is not suitable for direct training using nighttime RAW data. This is primarily due to 1) the SfM estimations based on nighttime images are often inaccurate, leading to blurred distant views or the potential emergence of floaters; 2) the SH does not adequately represent the HDR color information of RAW images due to its limited representation capacity; and 3) the finally reconstructed structure, such as depth map, is suboptimal, leading to unsatisfactory performance in downstream tasks like refocus.

To make 3DGS suitable for HDR scene reconstruction, we introduce LE3D that stands for **L**ighting **E**very darkness with **3D**-GS, addressing the three aforementioned issues. First, to address the issue of inaccurate SfM distant view estimation in low-light scenes, we proposed Cone Scatter Initialization to enrich the COLMAP-estimated SfM. It performs random point sampling within a cone using the estimated camera poses. Second, instead of the SH, we use a tiny MLP to represent the color in RAW linear color space. To better initialize the color of each gaussian, different color biases are used for various gaussian primitives. Thirdly, we propose to use depth distortion and near-far regularization to achieve better scene structure for downstream tasks like refocusing. As shown in Fig. 1 (left), our LE3D can perform real-time rendering with only about 1.5 GPU hours (99% less than RawNeRF [36]) of training time and at a rate of 100 FPS (about 4000$\times$ faster than RawNeRF [36]) with comparable quality. Additionally, it is capable of supporting downstream tasks such as HDR rendering, refocusing, and exposure variation, as shown in Fig. 1 (right).

In summary, we make the following contributions:

- We propose LE3D, which can reconstruct HDR 3D scenes from noisy raw images and perform real-time rendering. Compared with the NeRF-based methods, LE3D requires only 1% of the training time and has 4000$\times$ render speed.
- To address the deficiencies in color representation by vanilla 3DGS and the inadequacies of SfM estimation in nighttime scenes, we leverage a Color MLP with primitive-aware bias, and introduce Cone Scatter Initialization to enrich the point cloud initialized by COLMAP.
- To improve the scene structure in the final results for achieving better downstream task performance, we introduce depth distortion and near-far regularizations.

2 Related work

Novel view synthesis Since the emergence of NeRF [37], NVS has gained significant advancement. NeRF employs an MLP to represent both the geometry of the scene and the view-dependent color. It utilizes the differentiable volume rendering [28], thereby enabling gradient-descent training through a multi-view set of 2D images. Subsequent variants of NeRF [2, 3, 22] have extended NeRF's capabilities with anti-aliasing features. To overcome the deficiencies of vanilla NeRF in geometry reconstruction, strategies such as depth supervision [10, 12] and distortion loss [3] have been introduced into NeRF. Several methods [9, 46] work on low-light image enhancement with NeRF, and extended its application scenarios. Some methods [5, 7, 16, 39] have conducted experiments with feature-grid based approaches to enhance the training and rendering speeds of NeRF. Although these methods have achieved relatively promising results in novel view synthesis, the training and rendering speeds are still significant bottlenecks. This issue is primarily due to the dense sampling inherently required by volume rendering.

Recently, the advent of 3D Gaussian Splatting [25] has marked a significant advancement in real-time NVS methods. 3DGS represents a scene using a collection of 3D gaussian primitives, each endowed with distinct attributes. Some subsequent works have added anti-aliasing capabilities to 3DGS representations [32, 43, 52]; others have enhanced 3DGS representation capabilities through supervision in the frequency domain [53]. DNGaussian [29] proposed a depth-regularized framework to optimize sparse-view 3DGS, and other works also relying on depth supervision [8, 27]. Additionally, some works [30, 34, 49] have focused on applying 3DGS to dynamic scene representation. However, these methods only accept LDR sRGB data for training, and thus cannot reconstruct the scene's HDR radiance. This limitation suggests they cannot perform downstream tasks such as HDR tone mapping and exposure variation. In contrast, LE3D is specifically designed to reconstruct the HDR representation of scenes from noisy RAW images.

HDR view synthesis and its applications HDR typically refers to a concept in computational photography that focuses on preserving as much dynamic range as possible to facilitate more post-processing options [11, 14, 20, 24, 33]. The existing HDR view synthesis techniques bear a striking resemblance to the two main approaches in 2D image HDR synthesis: 1) Direct use of multiple-exposure LDR images to compute the camera response function (CRF) and synthesize an HDR image [11]. This corresponds to HDR-NeRF [23] which employs an MLP to learn the CRF. 2) Acquisition of noise-free underexposed RAW images, utilizing the characteristics of the linear color space in RAW to manually simulate multiple-exposure images, and synthesize an HDR image. This corresponds to RawNeRF [36], which learns a NeRF representation of RAW linear color space with noisy RAW images to perform both denoising and NVS. VR-NeRF[50] performs a perceptual color space that enables perceptual optimization of high dynamic range images using. Although both methods achieve impressive visual results, the dense sampling required by volume rendering still poses a bottleneck for both training time and rendering efficiency.

LE3D follows the same technical approach as RawNeRF, reconstructing scene representations from noisy RAW images. This means that LE3D does not necessarily require training data with multiple exposures, significantly broadening its range of applications. However, a key distinction of LE3D is its use of differentiable rasterization techniques [25, 26, 44], which enable *fast training and real-time rendering*. Based on a 3DGS-like representation of the reconstructed scene, LE3D can perform real-time HDR view synthesis. This is a novel attempt to introduce computational photography into the 3D world, as it enables *real-time reframing and post-processing* (changing white balance, HDR rendering, etc. as shown in Fig. 1).

3 Preliminaries

3D Gaussian Splatting 3D Gaussian Splatting renders detailed scenes by computing the color and depth of pixels through the blending of many 3D gaussian primitives. Each gaussian is defined by its center in 3D space $\mu_i \in \mathbb{R}^3$, a scaling factor $s_i \in \mathbb{R}^3$, a rotation quaternion $q_i \in \mathbb{R}^4$, and additional attributes such as opacity o_i and color features f_i. The basis function of a gaussian primitive is given by Eqn. (1) that incorporates the covariance matrix derived from the scaling and rotation parameters.

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

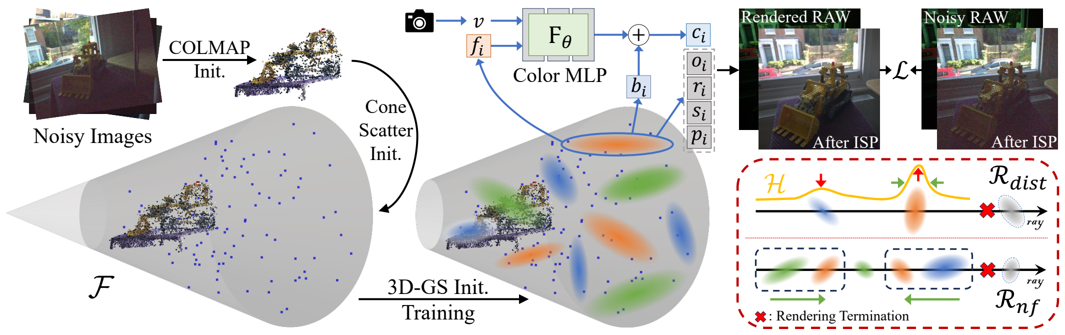


Figure 2: Pipeline of our proposed LE3D. 1) Using COLMAP to obtain the initial point cloud and camera poses. 2) Employing Cone Scatter Initialization to enrich the point clouds of distant scenes. 3) The standard 3DGS training, where we replace the original SH with our tiny Color MLP to represent the RAW linear color space. 4) We use RawNeRF's weighted L2 loss $\mathcal{L}$ (Eqn. (3)) as image-level supervision, and our proposed $\mathcal{R}_{dist}$ (Eqn. (8)) as well as $\mathcal{R}_{nf}$ (Eqn. (9)) as scene structure regularizations. In this context, f_i, b_i, and c_i respectively represent the color feature, bias, and final rendered color of each gaussian i. Similarly, o_i, r_i, s_i, and p_i denote the opacity, rotation, scale, and position of them.

During rendering, the color of a pixel is determined by blending the contributions of multiple gaussians that overlap the pixel's location. This process involves decoding the color features f_i to color c_i by the SH, and calculating α_i of each primitive by multiplying its opacity o_i with its projected 2D gaussian G_i^{2D} on the image plane. Unlike traditional ray sampling strategies, 3D Gaussian Splatting employs an optimized rasterizer to gather the relevant gaussians for rendering. Specifically, the color C is computed by blending N ordered gaussians overlapping the pixel:

$$C = \sum_{i \in N} c_i \cdot \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \text{where } \alpha_i = G_i^{2D} o_i. \tag{2}$$

HDR view synthesis with noisy RAW images RawNeRF [36], as a powerful extension of NeRF, specifically addresses the challenge of high dynamic range (HDR) view synthesis with noisy images. Different from LDR images, the dynamic range in HDR images can span several orders of magnitude between bright and dark regions, resulting in the NeRF's standard L2 loss being inadequate. To address this challenge, RawNeRF introduces a weighted L2 loss that enhances supervision in the dark regions. RawNeRF applies gradient supervision on tone curve $\psi = \log(y + \epsilon)$ with $\epsilon = 10^{-3}$, and uses $\psi' = (y + \epsilon)^{-1}$ as the weighting term on the L2 loss between rendered color $\hat{y}_i$ and noisy reference color y_i. Applying the stop-gradient function $sg(\cdot)$ on y, the final loss can be expressed as:

$$L_\psi(\hat{y}, y) = \sum_i \left(\frac{\hat{y}_i - y_i}{sg(\hat{y}_i) + \epsilon} \right)^2. \tag{3}$$

Moreover, RawNeRF employs a variable exposure training method to take advantage of images with varying shutter speeds. The method scales the output color in linear RGB space by a learned factor $\beta_{t_i}^c$ for each color channel c with each unique shutter speed t_i. In particular, the c-th channel of the output color $\hat{y}_i^c$ will be mapped into $\min(\hat{y}_i^c \cdot t_i \cdot \beta_{t_i}^c, 1)$ as the final output for rendering.

4 Proposed method

The pipeline of our LE3D is shown in Fig. 2. Our main motivations and solutions are as follows: 1) To address the issue of COLMAP's inadequacy in capturing distant scenes during nighttime, we utilize the proposed Cone Scatter Initialization to enrich the point cloud obtained from COLMAP. 2) Experiments show that the original SH in 3DGS is inadequate for representing the RAW linear color space (as shown in Fig. 4 (e) and Fig. 7). Therefore, we replace it with a tiny color MLP. 3) To enhance the scene structure and optimize the performance of downstream tasks, we propose the depth distortion $\mathcal{R}_{dist}$ and near-far $\mathcal{R}_{nf}$ regularizations.

4.1 Improvements to the vanilla 3DGS representation

Directly applying 3DGS on noisy RAW images set faces the aforementioned two challenges, lack of distant points and inadequate representation of RAW linear color space. To address them, we propose the following improvements to the vanilla 3DGS representation.

Cone Scatter Initialization To enrich the COLMAP-initialized point cloud $\mathcal{S} = \{\mathbf{s}_i\}$ with distant scenes, we estimate the position and orientation of all cameras. Based on this, we randomly scatter points within a predefined viewing frustum $\mathcal{F}$. To define $\mathcal{F}$, we need to determine: 1) The viewpoint $\mathbf{p}$; 2) The viewing direction $\vec{\mathbf{n}}$; 3) The field of view Θ; and 4) The near and far planes, $z = z_n$ and $z = z_f$, respectively. For forward-facing scenes, the viewing direction can be easily determined by averaging the orientations of all cameras, represented as $\vec{\mathbf{n}} = \mathrm{avg}\{\vec{\mathbf{n}}_i^{\mathrm{c}}\}$. To encompass all visible areas in space from the training viewpoints, we use the maximum value of FOV from all cameras, denoted as $\Theta = \max\{\theta_i^{\mathrm{c}}\}$. Additionally, $\mathcal{F}$ needs to include all the camera origins $\{\mathbf{p}_i^{\mathrm{c}}\}$ to ensure complete coverage of the scene from all perspectives. It means that $\mathcal{F}$ should encompass a circle with its center at $\overline{\mathbf{p}}^{\mathrm{c}} = \mathrm{avg}\{\mathbf{p}_i^{\mathrm{c}}\}$, radius $r = \max\{\|\mathbf{p}_i^{\mathrm{c}} - \overline{\mathbf{p}}^{\mathrm{c}}\|_2\}$, and perpendicular to $\vec{\mathbf{n}}$. Therefore, we can establish $\mathcal{F}$:

$$\begin{aligned} &\mathbf{p} = \overline{\mathbf{p}}^{\mathrm{c}} - \frac{r}{\tan(\Theta/2)} \cdot \frac{\vec{\mathbf{n}}}{\|\vec{\mathbf{n}}\|_2}, \quad \vec{\mathbf{n}} = \mathrm{avg}\{\vec{\mathbf{n}}_i^{\mathrm{c}}\}, \quad \Theta = \max\{\theta_i^{\mathrm{c}}\}, \\ &z_n = \min\{\|\mathbf{s}_i - \mathbf{p}\|_2\}, \quad z_f = \lambda_{\mathcal{F}} \cdot \max\{\|\mathbf{s}_i - \mathbf{p}\|_2\}. \end{aligned} \tag{4}$$

For near z_n and far z_f, we use the distance from the nearest and $\lambda_{\mathcal{F}}$ times the distance from farthest points in the COLMAP-initialized point cloud $\mathcal{S}$ to $\mathbf{p}$ to represent them, respectively. Subsequently, we randomly scatter points within our viewing frustum $\mathcal{F} = \{\mathbf{p}, \vec{\mathbf{n}}, \Theta, z_n, z_f\}$ to obtain our enriched point cloud $\mathcal{S}' = \mathcal{S} \cup \mathcal{S}^{\mathcal{F}}$, where $\mathcal{S}^{\mathcal{F}}$ is the scattered point set. Then $\mathcal{S}'$ is used for initialization of the gaussians, instead of $\mathcal{S}$.

Color MLP with primitive-aware bias To address the issue that SH could not adequately represent the RAW linear color space, we replace it with a tiny color MLP $\mathbf{F}_\theta$. Each gaussian is initialized with a random color feature f_i and a color bias b_i. To initialize b_i, we project each $\mathbf{s}_i' \in \mathcal{S}'$ onto every training image, obtaining a set of all pixels $\{c_{\mathrm{pix}}\}_i$ for each point i,. The color features f_i is concatenated with the camera pose v, and then it is fed into the tiny color MLP $\mathbf{F}_\theta$ to obtain the view-dependent color. Since the HDR color space theoretically has no upper bound on color values, we use the exponent function as the activation function for $\mathbf{F}_\theta$ to simulate it. The final color c_i is:

$$c_i = \exp\left(\mathbf{F}_\theta(f_i, v) + b_i\right), \text{where } b_i^{(0)} = \log(\mathrm{avg}(\{c_{\mathrm{pix}}\}_i)), f_i^{(0)} \leftarrow \mathcal{N}(0, \sigma_f). \tag{5}$$

where $f_i^{(0)}$ is sampled from a gaussian distribution $\mathcal{N}(0, \sigma_f)$ and $b_i^{(0)}$ is setted by the log value of the average of $\{c_{\mathrm{pix}}\}_i$. This initialization makes $c_i^{(0)}$ close to $\mathrm{avg}\{c_{\mathrm{pix}}\}_i$. Both f_i and b_i are learnable parameters and during cloning and splitting, they are copied and assigned to new gaussians.

4.2 Depth distortion & near-far regularizations

Scene structure is crucially important for the downstream applications of our framework, particularly the tasks such as refocusing. Therefore, we propose depth distortion and near-far regularizations to enhance the ability of 3DGS to optimize scene structure. Borrowing from NeRF-based methods [3], we use a depth map and weight map to regularize the scene structure.

Depth and weight map rendering Recently, several 3DGS-based works [29, 8] employ some form of supervision on depth. Also, depth maps are crucial for downstream tasks such as refocus (Sec. 6), mech extraction [19] and relighting [17, 54]. They are achieved by obtaining the rendered average depth map d in the following manner:

$$d = \frac{\sum_i z_i^{\mathrm{c}} \omega_i}{\sum_i \omega_i}, \text{ where } [x_i^{\mathrm{c}}, y_i^{\mathrm{c}}, z_i^{\mathrm{c}}]^T = W[x_i, y_i, z_i]^T + t, \text{ and } \omega_i = \alpha_i \prod_{j=1}^{i-1}(1 - \alpha_j). \tag{6}$$

where d denotes the depth map, ω_i represents the blending weight of the i-th gaussian, $[x_i, y_i, z_i]^T$ and $[x_i^{\mathrm{c}}, y_i^{\mathrm{c}}, z_i^{\mathrm{c}}]^T$ represent the position in the world and the camera coordinate system, respectively, and $[W, t]$ corresponds to the camera extrinsics. The pixel values in the Weight Map each describe a histogram $\mathcal{H}$ of the distribution on the ray passing through this pixel. Similar to Mip-NeRF 360 [3], we can optimize the scene structure by constraining the gaussian primitives on each ray to be more concentrated. To obtain the Weight Map, we first need to determine the distances to the nearest and farthest gaussian primitives from the current camera pose p^{c}, represented as $z_n^{\mathrm{c}}, z_f^{\mathrm{c}}$, respectively. Subsequently, we transform the interval $[z_n^{\mathrm{c}}, z_f^{\mathrm{c}})$ to obtain K intersections, where the k-th intersection is denoted as $[t_k, t_{k+1})$. Thus, the k-th value in the histogram $\mathcal{H}(k)$ can be obtained through rendering in the following manner:

$$\mathcal{H}(k) = \sum_i \mathbb{1}(z_i^{\mathrm{c}}, k)\omega_i, \text{ where } \mathbb{1}(z_i^{\mathrm{c}}, k) = \begin{cases} 1 & \text{if } z_i^{\mathrm{c}} \in [t_k, t_{k+1}) \\ 0 & \text{else} \end{cases}. \tag{7}$$

Rendering $\mathcal{H}$ is essential, as it is effective not only in regularization but also plays a role in the refocusing application.

Proposed regularizations Inspired by Mip-NeRF 360 [3], we proposed similar depth distortion regularization $\mathcal{R}_{dist}$ to concentrate the gaussians on each ray:

$$\mathcal{R}_{dist} = \sum_{u,v}^{K} \mathcal{H}(u)\mathcal{H}(v) \left| \frac{t_u + t_{u+1}}{2} - \frac{t_v + t_{v+1}}{2} \right|. \tag{8}$$

$\mathcal{R}_{dist}$ constrains the weights along the entire ray to either approach zero or be concentrated at the same intersection. However, in unbounded scenes of the real world, the distances $(z_f^{\mathrm{c}} - z_n^{\mathrm{c}})/K$ between each intersection are vast. Forcibly increasing the size of K to reduce the length of each intersection also significantly increases the computational burden. This means that our $\mathcal{R}_{dist}$ can only provide a relatively coarse supervision for the gaussians on each ray, primarily by constraining them as much as possible within the same intersection.

To further constrain the concentration of gaussians, we propose the Near-Far Regularization $\mathcal{R}_{nf}$. $\mathcal{R}_{nf}$ enhances the optimization of scene structure by narrowing the distance between the weighted depth of the nearest and farthest M gaussians on each ray, where the farthest refers to the last M gaussians when the blending weight approaches 1. First, we extract two subsets of gaussians, $\mathbf{N}$ and $\mathbf{F}$, which respectively contain the nearest and farthest M gaussians on each ray. Subsequently, we render the depth maps for both subsets ($d^{\mathbf{N}}$, $d^{\mathbf{F}}$), as well as the final blending weight map ($T^{\mathbf{N}}$, $T^{\mathbf{F}}$). The blending weight map T is the sum of each ω_i. And here comes $\mathcal{R}_{nf}$:

$$\mathcal{R}_{nf} = T^{\mathbf{N}} \cdot T^{\mathbf{F}} \cdot \left| d^{\mathbf{N}} - d^{\mathbf{F}} \right|. \tag{9}$$

It not only can prune the gaussians at the front or back of each ray through opacity supervision when there is a significant disparity between them (relying on the $T^{\mathbf{N}} \cdot T^{\mathbf{F}}$ term). Compared to $\mathcal{R}_{dist}$, $\mathcal{R}_{nf}$ can also supervise the position of the first and last M gaussians on each ray to be as close as possible (relying on the $\left| d^{\mathbf{N}} - d^{\mathbf{F}} \right|$ term). Besides the weighted L2 loss $\mathcal{L}$ and proposed regularizations $\mathcal{R}_{dist}$ and $\mathcal{R}_{nf}$, we also introduce constraints on the final blending weights T. Given that the LE3D is tested in real-world scenarios, T should ideally approach 1, meaning that all pixels should be rendered. Thus, we propose $\mathcal{R}_T = -\log(T + \epsilon)$ to penalize the pixels where T is less than 1.

5 Experiments

5.1 Implementation details

Loss functions and regularizations In our implementation, the final loss function is:

$$L = \mathcal{L} + \lambda_T \mathcal{R}_T + \lambda_{dist} \mathcal{R}_{dist} + \lambda_{nf} \mathcal{R}_{nf}, \tag{10}$$

where $\mathcal{L}$ is the weighted L2 loss, and $\mathcal{R}_T$, $\mathcal{R}_{dist}$, and $\mathcal{R}_{nf}$ are the proposed T, depth distortion, and near-far regularizations, respectively.

Optimization We set $\lambda_{\mathcal{F}}$ to 10 to enrich the COLMAP-initialized point cloud in distant views. $\lambda_T, \lambda_{dist}, \lambda_{nf}$ in the loss function are set to $0.01, 0.1, 0.01$ respectively. For our color MLP $\mathbf{F}_\theta$, we use the Adam optimizer with an initial learning rate of $1.0e-4$. The initial learning rates for color features and biases for each gaussians are set to $2.0e-3$ and $1.0e-4$, respectively. The learning rates for all three decrease according to a cosine decay strategy to a final learning rate of $1.0e-5$. Besides the color MLP, primitive-aware color bias, and the color features for each gaussians, other settings are the same as those of 3DGS [25]. For scenes captured with multiple exposures, we employ the same multiple-exposure training strategy as RawNeRF [36].

Table 1: Quantitative results on the test scenes of the RawNeRF [36] dataset. The best result is in **bold** whereas the second best one is in underlined. TM indicates whether the tone-mapping function can be replaced for HDR rendering. For methods where the tone-mapping function can be replaced, the metrics on sRGB are calculated using LDR tone-mapping for a fair comparison. The FPS measurement is conducted at a 2K (2016×1512) resolution. Train denotes the training time of the method, measured in GPU×H. LE3D achieves comparable performance with previous volumetric rendering based methods (RawNeRF [36]), but with 4000× faster rendering speed.

Method	TM	FPS↑	Train↓	RAW		sRGB		
				PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
LDR-NeRF [37]	✗	0.007	13.66	–	–	20.0391	0.5541	0.5669
LDR-3DGS [25]	✗	153	0.75	–	–	20.2936	0.5477	0.5344
HDR-3DGS [25]	✓	238	0.73	56.4960	0.9926	20.3320	0.5286	0.6563
RawNeRF [36]	✓	0.022	129.54	58.6920	0.9969	24.0836	**0.6100**	**0.4952**
RawGS (Baseline)	✓	176	1.05	59.2834	0.9971	23.3485	0.5843	0.5472
LE3D (Ours)	✓	103	1.53	**61.0812**	**0.9983**	**24.6984**	0.6076	0.5071

5.2 Datasets and comparisons

Datasets We evaluated LE3D's performance on the benchmark dataset of RawNeRF. It includes fourteen scenes for qualitative testing and three test scenes for quantitative testing. The three test scenes contains 101 noisy images and a clean reference image merged from stabilized long exposures. However, the training data are captured with short exposures, leading to exposure inconsistencies. Therefore, we apply the same affine alignment operation as RawNeRF before testing (detailed in Sec. A.1 of the supplementary materials). All images are 4032×3024 Bayer RAW images captured by an iPhone X, saved in DNG format.

Baseline and comparative methods We compare two categories of methods, 3DGS-based methods, and NeRF-based methods. The baseline we compare against is RawGS which uses vanilla 3DGS for scene representation and employs the weighted L2 loss and multiple exposure training proposed in RawNeRF [36]. Besides, we remove the SSIM loss, as its computation over local neighborhoods disrupts the noise model present in the RAW data. Additionally, we compare LDR-GS and HDR-GS, both of which are vanilla 3DGS trained on post-processed LDR images and unprocessed RAW images, respectively. The NeRF-based methods include RawNeRF [36] and LDR-NeRF. LDR-NeRF is a vanilla NeRF [37] trained on the post-processed LDR images with L2 loss.

Quantitative evaluation Tab. 1 has shown the quantitative comparisons on the RawNeRF [36] dataset. Although NeRF-based methods have long training times and slow rendering speeds, they exhibit good metrics on sRGB. This indicates that the volume rendering they rely on has strong noise resistance (mainly due to the dense sampling on each ray). In contrast, 3DGS-based methods have inferior metrics compared to RawNeRF, due to their sparse scene representation and poor noise resistance. Additionally, the splitting of gaussians depends on gradient strength, and supervision using noisy raw images affects this process, leading to incomplete structure recovery. LE3D achieves better structure reconstruction suitable for downstream tasks through supervision on structure, depth distortion, and near-far regularizations, as detailed in Sec. 6. Note that the results of LE3D are comparable to the previous volumetric rendering-based method, RawNeRF [36], in both quantitative and qualitative aspects. However, it requires only 1% of the training time and achieves a 3000×-6000× rendering speed improvement.

Qualitative evaluation Fig. 3 has shown the qualitative comparisons on the RawNeRF [36] dataset. We selected four scenes for comparison, including two indoor scenes and two outdoor scenes. The data for the first two scenes were collected with a single exposure, while the data for the latter two scenes included multiple exposures. Compared to 3DGS [25]-based methods, LE3D demonstrates stronger noise resistance, particularly in the first two scenes. Additionally, LE3D achieves better results in distant scene reconstruction. For example, in the second scene, LE3D produces a smoother sky compared to RawGS, and in the fourth scene, LE3D recovers distant details more sharply. Compared to RawNeRF, LE3D typically produces smoother results while still effectively preserving details. Most importantly, LE3D offers faster training times and rendering speeds.

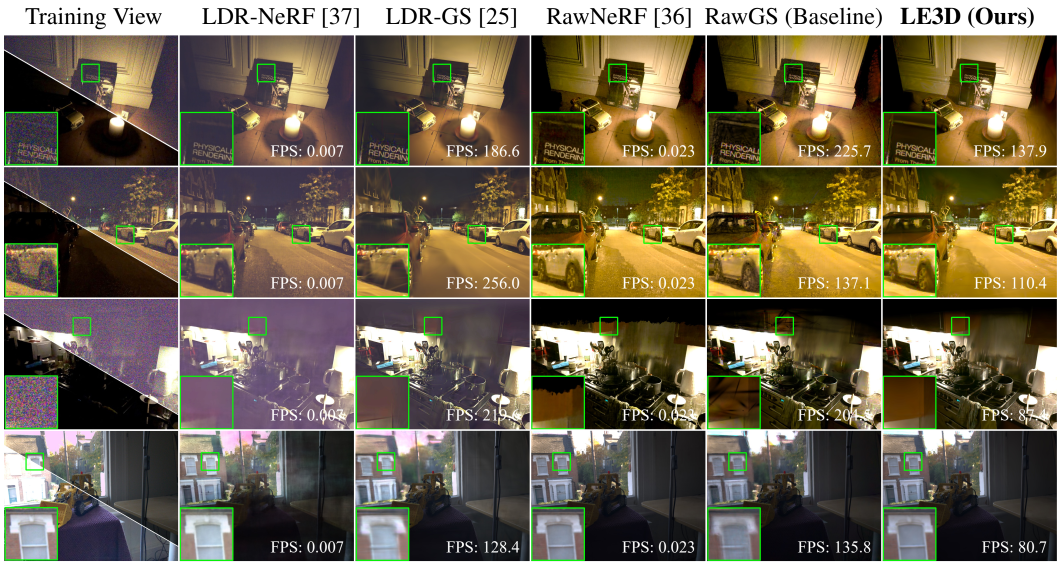


Figure 3: Visual comparison between LE3D and other reconstruction methods (*Zoom-in for best view*). The training view contains two parts: the post-processed RAW image with linear brightness enhancement (up) and the image directly output by the device (down). By comparison to the 3DGS-based method, LE3D recovers sharper details in the distant scene and is more resistant to noise. Additionally, compared to NeRF-based methods, LE3D achieves comparable results with 3000×-6000× improvement in rendering speed.

5.3 Ablation studies

Cone Scatter Initialization (CSI) In low-light environments, COLMAP struggles to obtain a high-quality sparse point cloud. Although 3DGS demonstrates its robustness to the quality of the initial point cloud, it still encounters difficulties in achieving optimal geometric reconstruction within insufficient initialized areas. It can be observed from Fig. 4 (b) that the methods without CSI tend to generate gaussians at incorrect depths and lack fine details. Conversely, CSI extends the depth coverage of the scene, enabling 3DGS to generate gaussians at relatively accurate depths and exhibit superior detail representation. A comparative analysis between Fig. 4 (a) and Fig. 4 (b) suggests that our initialization technique plays a pivotal role in achieving accurate and detailed 3D reconstruction.

Color MLP Replacing SH with Color MLP not only enhances the expressiveness of our model but also introduces greater stability during the optimization process. Fig. 4 (e) reveals that the method employing SH rather than Color MLP exhibits strange color representations early in the training phase, due to the inability of SH to adequately represent the RAW linear color space. Although the rendered image may appear similar to those produced by the LE3D, the underlying issues have significantly affected the final structural reconstruction, as depicted in Fig. 4 (c).

Regularizations Superior visual effects in 3D are contingent upon a robust 3D structure reconstruction, which in turn significantly enhances the performance of downstream tasks such as refocusing. To this end, we implement depth distortion regularization $\mathcal{R}_{dist}$ and near-far regularization $\mathcal{R}_{nf}$ to constrain the gaussians, ensuring their aggregation at the surfaces of objects and thereby improving the quality of structural reconstruction. Fig. 4 (d) underscores the substantial enhancement our proposed regularizations provide in reconstructing the 3D structure of scenes.

6 More applications

Refocus Structural information is crucial for tasks like refocusing. As discussed in Sec. 5.3, LE3D benefits from the inclusion of depth distortion and near-far regularizations, which enhances its ability to learn structural details. As shown in Fig. 5 (b, d), LE3D achieves more realistic refocusing effects due to its superior structural information, as reflected in the depth shown in (c). Conversely, RawGS suffers from foreground and background ambiguity in refocusing due to the lack of structural information. The detailed refocusing algorithm will be released in the supplementary materials.

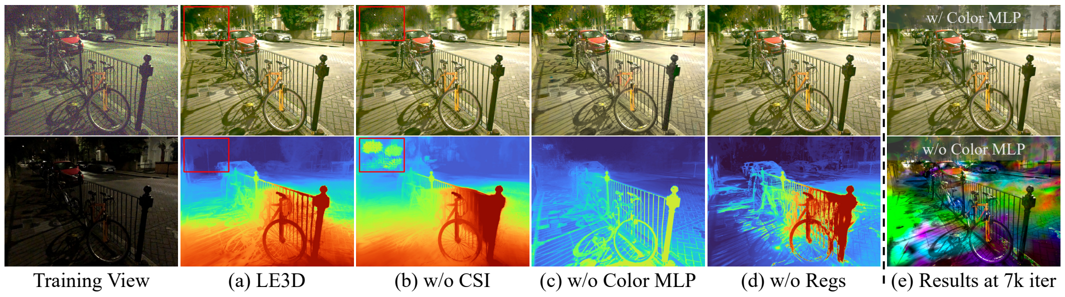


Figure 4: Ablation studies on our purposed methods (*Zoom-in for best view*). CSI in (b) and Regs in (d) denote Cone Scatter Initialization and Regularizations, respectively. (e) shows the rendering result of LE3D w/ or w/o Color MLP in the early stages of training.

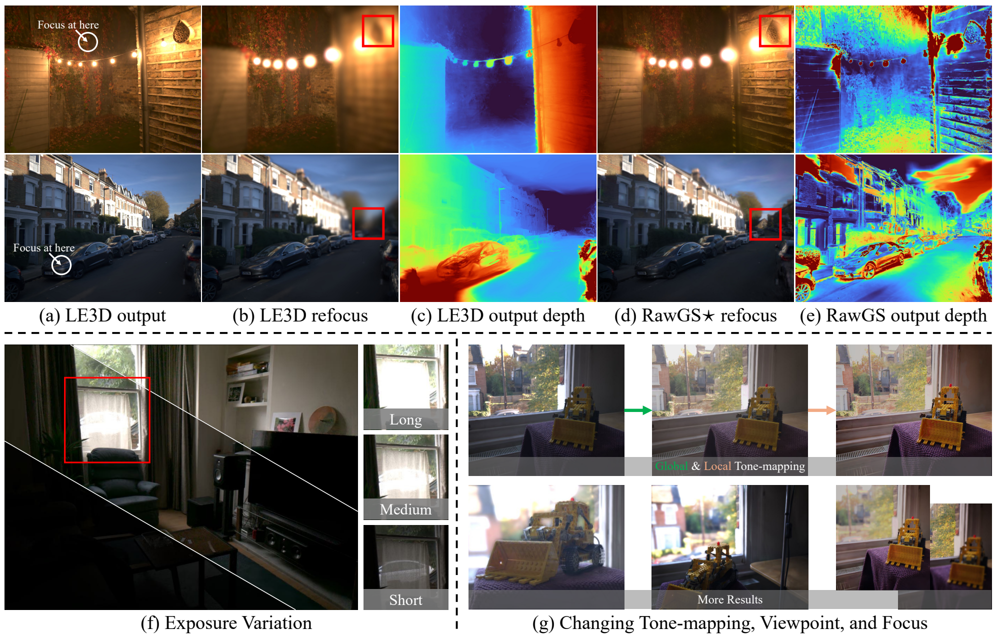


Figure 5: LE3D supports various applications. RawGS⋆ in (d) denotes using LE3D's rendered image and RawGS's structure information as input for refocusing. (c, e) are the weighted depth rendered by LE3D and RawGS, respectively. (f) shows the same scene rendered by LE3D with different exposure settings. In (g), the "$\rightarrow$" denotes global tone-mapping, while the "$\rightarrow$" represents local tone-mapping.

Exposure variation and HDR tone-mapping LE3D can easily achieve exposure variation and recover details from overexposed data, as shown in Fig. 5 (f). Fig. 5 (g) showcases the various tone-mapping methods LE3D can implement, including global tone-mapping, such as color temperature and curve adjustments, and local tone-mapping using our re-implemented HDR+ [20] (implementation details will be released in the supplementary materials).

Although RawNeRF [36] can also perform similar applications, its inability to achieve **real-time** rendering significantly limits its use cases, such as real-time editing described in Sec. C.

7 Conclusion

To address the long training times and slow rendering speeds of previous volumetric rendering-based methods, we propose LE3D based on 3DGS. Additionally, we introduce Cone Scatter Initialization and a tiny MLP to represent color in the linear color space. This addresses the issue of missing distant points in nighttime scenes with COLMAP initialization. It also replaces spherical harmonics with the tiny color MLP, effectively representing the linear color space. Finally, we enhance the structural reconstruction with the proposed depth distortion and near-far regularization, enabling more effective and realistic downstream tasks. Benefiting from the rendering images in the linear color space, LE3D can achieve more realistic exposure variation and HDR tone-mapping in real time, expanding the possibilities for subsequent HDR view synthesis processing.

Acknowledgement

This work is funded by National Natural Science Foundation of China (No. 62225604, No. 62272245, No. 62306153), the Fundamental Research Funds for the Central Universities (Nankai University, No. 63243147), and Tianjin science and technology projects (22JCYBJC01270). The computational devices of this work is supported by the Supercomputing Center of Nankai University (NKSC).

References

[1] Jonathan T Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. Fast bilateral-space stereo for synthetic defocus. In *CVPR*, 2015.

[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.

[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] Yuanhao Cai, Zihao Xiao, Yixun Liang, Yulun Zhang, Xiaokang Yang, Yaoyao Liu, and Alan Yuille. Hdr-gs: Efficient high dynamic range novel view synthesis at 1000x speed via gaussian splatting. *arXiv:2405.15125*, 2024.

[5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022.

[6] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, 2018.

[7] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *ICCV*, 2023.

[8] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. *arXiv:2311.13398*, 2023.

[9] Ziteng Cui, Lin Gu, Xiao Sun, Xianzheng Ma, Yu Qiao, and Tatsuya Harada. Aleth-nerf: Illumination adaptive nerf with concealing field assumption. In *AAAI*, 2024.

[10] David Dadon, Ohad Fried, and Yacov Hel-Or. Ddnerf: Depth distribution neural radiance fields. In *WACV*, 2023.

[11] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. *Siggraph*, 1997.

[12] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022.

[13] Paul HC Eilers and Brian D Marx. Flexible smoothing with b-splines and penalties. *Statistical science*, 1996.

[14] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K Mantiuk, and Jonas Unger. Hdr image reconstruction from a single exposure using deep cnns. *ACM TOG*, 2017.

[15] Zeev Farbman, Raanan Fattal, and Dani Lischinski. Convolution pyramids. *ACM TOG*, 2011.

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

[17] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv:2311.16043*, 2023.

[18] NeRF Studio Group. Viser. `https://github.com/nerfstudio-project/viser`, 2023.

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

[20] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM TOG*, 2016.

[21] Tanner Helland. How to convert temperature (k) to rgb: Algorithm and sample code. `https://tannerhelland.com/2012/09/18/convert-temperature-rgb-algorithm-code.html`, 2012.

[22] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, 2023.

[23] Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, Xuan Wang, and Qing Wang. Hdr-nerf: High dynamic range neural radiance fields. In *CVPR*, 2022.

[24] Nima Khademi Kalantari, Ravi Ramamoorthi, et al. Deep high dynamic range imaging of dynamic scenes. *ACM TOG*, 2017.

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

[26] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, 2021.

[27] Pou-Chun Kung, Seth Isaacson, Ram Vasudevan, and Katherine A Skinner. Sad-gs: Shape-aligned depth-supervised gaussian splatting. In *ICRA Workshops*, 2024.

[28] Marc Levoy. Efficient ray tracing of volume data. *ACM TOG*, 1990.

[29] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. *arXiv:2403.06912*, 2024.

[30] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. *CVPR*, 2024.

[31] Zhihao Li, Yufei Wang, Alex Kot, and Bihan Wen. From chaos to clarity: 3dgs in the dark. *arXiv:2406.08300*, 2024.

[32] Zhihao Liang, Qi Zhang, Wenbo Hu, Ying Feng, Lei Zhu, and Kui Jia. Analytic-splatting: Anti-aliased 3d gaussian splatting via analytic integration. *arXiv:2403.11056*, 2024.

[33] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Single-image hdr reconstruction by learning to reverse the camera pipeline. In *CVPR*, 2020.

[34] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv:2308.09713*, 2023.

[35] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *PG*, 2007.

[36] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *CVPR*, 2022.

[37] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[38] Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman, Ricardo Martin-Brualla, and Jonathan T. Barron. MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF. `https://github.com/google-research/multinerf`, 2022.

[39] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022.

[40] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023.

[41] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[42] Shreyas Singh, Aryan Garg, and Kaushik Mitra. Hdrsplat: Gaussian splatting for high dynamic range 3d scene reconstruction from raw images. *BMVC*, 2024.

[43] Xiaowei Song, Jv Zheng, Shiran Yuan, Huan-ang Gao, Jingwei Zhao, Xiang He, Weihao Gu, and Hao Zhao. Sa-gs: Scale-adaptive gaussian splatting for training-free anti-aliasing. *arXiv:2403.19615*, 2024.

[44] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022.

[45] Neal Wadhwa, Rahul Garg, David E Jacobs, Bryan E Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM TOG*, 2018.

[46] Haoyuan Wang, Xiaogang Xu, Ke Xu, and Rynson WH Lau. Lighting up nerf via unsupervised decomposition and enhancement. In *ICCV*, 2023.

[47] Xintao Wang, Liangbin Xie, Ke Yu, Kelvin C.K. Chan, Chen Change Loy, and Chao Dong. BasicSR: Open source image and video restoration toolbox. `https://github.com/XPixelGroup/BasicSR`, 2022.

[48] Kaixuan Wei, Ying Fu, Yinqiang Zheng, and Jiaolong Yang. Physics-based noise modeling for extreme low-light photography. *IEEE TPAMI*, 2021.

[49] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv:2310.08528*, 2023.

[50] Linning Xu, Vasu Agrawal, William Laney, Tony Garcia, Aayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Aljaž Božič, et al. Vr-nerf: High-fidelity virtualized walkable spaces. In *Siggraph Asia*, 2023.

[51] Chongjie Ye, Yinyu Nie, Jiahao Chang, Yuantao Chen, Yihao Zhi, and Xiaoguang Han. Gaustudio: A modular framework for 3d gaussian splatting and beyond. *arXiv:2403.19632*, 2024.

[52] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. *CVPR*, 2024.

[53] Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. Fregs: 3d gaussian splatting with progressive frequency regularization. *CVPR*, 2024.

[54] Tianyi Zhang, Kaining Huang, Weiming Zhi, and Matthew Johnson-Roberson. Darkgs: Learning neural illumination and 3d gaussians relighting for robotic exploration in the dark. *arXiv:2403.10814*, 2024.

[55] Xuaner Zhang, Kevin Matzen, Vivien Nguyen, Dillon Yao, You Zhang, and Ren Ng. Synthetic defocus and look-ahead autofocus for casual videography. *arXiv:1905.06326*, 2019.

[56] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv:1805.09817*, 2018.

Appendix

A Implementation details and more ablation studies

A.1 More detailed implementation

Detailed network architecture of the Color MLP To prevent our color MLP from occupying too much time during rendering, we used a very compact MLP. It consists of only one 16-dimension hidden layer and takes a 19-dimensional input (16 for the color feature of each gaussian and 3 for the viewing direction). With activation function `ReLU`, the output of the MLP is a 3-dimensional vector as the final color of the gaussian primitive.

Sepcific hyperparameters The $\lambda_{\mathcal{F}}$ in (4) to determine the farthest plane of the training frustum is set to 10. The $\mathbf{K}$ intersections of our weight map $\mathcal{H}$ is set to 32. The $\mathbf{M}$ gaussian number to calculate the near, far subsets $\mathbf{N}$ and $\mathbf{F}$ if set to 5. The $\lambda_T, \lambda_{dist}, \lambda_{nf}$ are set to $0.01, 0.1, 0.01$, respectively. And our training and testing resolution is set to 2016×1512.

Experimental environments In our implementation, we use a single Nvidia GeForce 3090 GPU (24 GB VRAM) for both training and testing along with a 20-core CPU and a 64GB RAM. The Pytorch version is 1.12.1, CUDA version is 11.8. Due to the limitation of 24 GB VRAM, RawNeRF cannot be trained on a single Nvidia GeForce 3090 GPU. For the test scenes, we train RawNeRF on 4 GPUs, while for the other scenes, RawNeRF is trained with 2 GPUs. Apart from the difference in the number of GPUs, we did not make any modifications to RawNeRF.

Affine alignment Since all training views in the RawNeRF [36] dataset are captured with a fast shutter, while the test views (ground truth) are captured with a slow shutter, linear enhancement is needed during testing for alignment. However, due to color bias (non-zero-mean noise for high ISO, [48]), direct linear enhancement does not achieve perfect alignment. Therefore, affine alignment is performed on both the output and ground truth during testing. In RawNeRF [36], this process is done as the following procedure:

$$a = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\overline{x^2} - \overline{x}^2} = \frac{\mathrm{Cov}(x, y)}{\mathrm{Var}(x)}, b = \overline{y} - a\overline{x}. \tag{11}$$

where $\overline{x}$ is the mean of x. And x, y are the ground truth and the final output, respectively. This is the least-squares fit of an affine transform $ax + b \approx y$. At test time, we first process y with $(y - b)/a$, then calculate the metric. For those methods whose output is in RAW linear color space, the affine alignment process is only done *once* in the RAW color space. While for other methods (LDR-NeRF [37], LDR-3DGS [25]) which can only output in RGB color space, the affine alignment process is done in the RGB color space.

A.2 More ablation studies

Ablation on number of gaussian with/without CSI While our Cone Scatter Initialization is used to add sparse points for distant scenes, some may be concerned it will influence the total number of gaussians, which is relevant for rendering speed. In fact, due to the optimization capabilities of 3DGS, in all scenarios, CSI not only did not increase the number of points but actually reduced them. Besides, CSI is particularly effective in the outdoor scene (-24.86% for 'bikes' and -13.10% for 'windowlegovary'). Details can be found in Tab. 2.

Ablation on Color Representation In practice, we have observed that employing SH directly can lead to negative color output values, an anomaly in the context of the RAW linear RGB space where such values are not permissible. Noticing that we attempted to directly apply the exponential function after the SH function, turning these negative values to positive. However, this approach did not resolve all the issues encountered. For example, in the 'windowlegovary' scenario, this approach showed a maximum value above 10^3 on one channel while staying below 10 on others, highlighting SH's expressive limitations. Additionally, we observed color issues similar to those in the 'yuccatest' scenario, as shown in Fig. 7. Therefore, using Color MLP instead of SH to represent color is necessary in our method.

Table 2: The final number of gaussians at convergence: w/o CSI indicates no cone scatter initialization, while LE3D uses full CSI. The change represents the variation in point count between LE3D and w/o CSI. Outdoor indicates if the scene is outdoors. CSI reduces the final point count in all scenarios and is particularly effective in outdoor scenes. This aligns with our motivation to improve distant view reconstruction using CSI.

Scene	bikes	candle	candlefiat	gardenlights	livingroom	morningkitchen
w/o CSI	532734	55784	110624	282499	495330	279090
LE3D	400310	49402	105924	246332	428997	260249
Change (%)	-24.86%	-11.44%	-4.25%	-12.80%	-13.39%	-6.75%
Outdoor	✓	✗	✗	✓	✓	✗

Scene	nightstreet	notchbush	officetest	parkstatue	pianotest	scooter
w/o CSI	377224	161850	418443	420570	300112	183443
LE3D	328822	148823	404602	336241	294808	148167
Change (%)	-12.83%	-8.05%	-3.31%	-20.05%	-1.77%	-19.23%
Outdoor	✓	✓	✗	✓	✗	✓

Scene	sharpshadow	stove	streetcorner	windowlegovary	yuccatest
w/o CSI	722148	378176	313398	802336	233571
LE3D	589047	363461	299916	697204	217325
Change (%)	-18.43%	-3.89%	-4.30%	-13.10%	-6.96%
Outdoor	✓	✗	✓	✓	✗

Ablation on each of the regularizations Fig. 6 (a,b) presents the visualization of $d^{\mathbf{N}}$ and $d^{\mathbf{F}}$ adjacent to the depth map. In an ideal scenario, both $d^{\mathbf{N}}$ and $d^{\mathbf{F}}$ should align with d, ensuring that the weights along each ray are concentrated at the surface. The comparison between Fig. 6 (a,b) demonstrates that the incorporation of near-far regularization indeed encourages $d^{\mathbf{N}}$ and $d^{\mathbf{F}}$ to progressively align with d. This alignment results in a more refined representation of the three-dimensional structure, capturing better details of the scene's geometry. The comparison between Fig. 6 (a,c) elucidates the adverse effects of omitting distortion regularization. Without such constraints, the model would fail to produce depth maps with a natural depth progression, with artifacts such as abrupt depth discontinuities or voids on planes. Such anomalies are indicative of significant issues in the reconstruction of the scene's geometry.

Ablation studies on test scenes As shown in Tab. 3, LE3D has shown superior performance over all ablated methods. The results without Color MLP are the worst because the SH used in vanilla 3DGS is not suitable for representing colors in the RAW linear color space. As shown in Fig. 7, the results without Color MLP are noticeably desaturated and appear gray. Both w/o $\mathcal{R}nf$ and w/o $\mathcal{R}dist$ show degraded depth, as seen in Fig. 7. Additionally, it can be observed that w/o Color MLP also has poor structural information. This is mainly due to instability during the early stages of training, leading to suboptimal depth map reconstruction, as discussed in Sec. 5.3.

The stability of LE3D Due to the random initialization of our Color MLP, we trained 9 versions of LE3D using 9 different random seeds to test the stability of LE3D. We then compared their metrics on the test set, as shown in Fig. 8. It can be observed that the stability of LE3D is remarkably high, and the overall fluctuations do not impact the experimental conclusions.

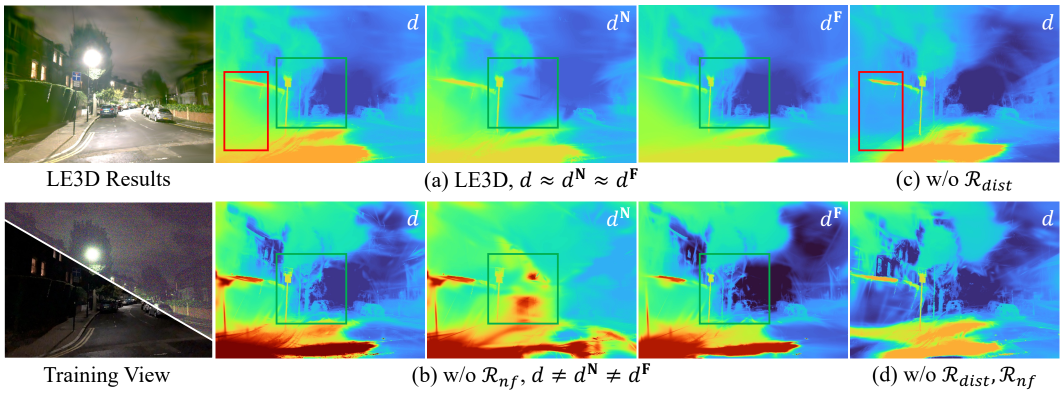
B Detailed postprocessing

B.1 Refocus

To implement the refocusing algorithm in LE3D, we adopted a refocusing approach similar to previous methods [1, 45, 55]. Most of these methods require pre-calculating a MultiPlane Image (MPI) representation [56] from the depth to synthesize refocus data. The MPI representation consists of multiple planes arranged from front to back, with each plane being an RGBA image (the RGBA image's color is still in the linear color space for our LE3D). RawNeRF [36] pre-calculates the

Table 3: Quantitative results of the ablation studies. Notice that, since Cone Scatter Initialization (CSI) is used to supplement the point cloud in distant scenes, and the test scenes do not contain distant views (all being indoor scenes), LE3D *does not apply CSI* in this context. The ablation study of CSI can be found in Fig. 4 (a, b), which shows significant differences in distant views. Best results is denoted in **bold**. The rank is indicated in the lower right corner of each metric.

Method	RAW		sRGB		
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
w/o Color MLP	59.4483$_{(4)}$	0.9969$_{(4)}$	23.1884$_{(4)}$	0.5862$_{(4)}$	0.5635$_{(4)}$
w/o $\mathcal{R}_{dist}$	60.5202$_{(3)}$	0.9981$_{(3)}$	24.3615$_{(3)}$	0.6007$_{(3)}$	0.5087$_{(2)}$
w/o $\mathcal{R}_{nf}$	60.7144$_{(2)}$	0.9982$_{(2)}$	24.5705$_{(2)}$	0.6043$_{(2)}$	0.5096$_{(3)}$
LE3D (Ours)	**61.0812**$_{(1)}$	**0.9983**$_{(1)}$	**24.6984**$_{(1)}$	**0.6077**$_{(1)}$	**0.5071**$_{(1)}$



d
$d^{\mathbf{N}}$
$d^{\mathbf{F}}$
d
LE3D Results
(a) LE3D, $d \approx d^{\mathbf{N}} \approx d^{\mathbf{F}}$
(c) w/o $\mathcal{R}_{dist}$
d
$d^{\mathbf{N}}$
$d^{\mathbf{F}}$
d
Training View
(b) w/o $\mathcal{R}_{nf}$, $d \neq d^{\mathbf{N}} \neq d^{\mathbf{F}}$
(d) w/o $\mathcal{R}_{dist}, \mathcal{R}_{nf}$

Figure 6: Ablation on each of the regularizations. Since both $\mathcal{R}_{dist}$ and $\mathcal{R}_{nf}$ are regularization terms intended to strengthen the structural representation, we have elected to display only the depth map for the sake of clarity. In addition, to demonstrate the effect of $\mathcal{R}_{nf}$ in aligning $d, d^{\mathbf{N}}, d^{\mathbf{F}}$, we have also visualized $d^{\mathbf{N}}$ and $d^{\mathbf{F}}$ as mentioned in Sec. 4.2.

MPI representation from its rendered results and then uses it for refocusing. However, due to its extremely slow rendering speed (about 50 seconds per image), real-time refocusing with RawNeRF is nearly impossible. To obtain our MPI representations in real time, we use the weight distortion map ($\mathcal{H} \in \mathbb{R}^{\mathbf{K}\times H\times W}$, where $\mathbf{K}$ denotes the plane number) rendered by LE3D as described in Eqn. (7) to directly render the final opacity for each plane in the MPI. The color image ($I \in \mathbb{R}^{3\times H\times W}$) used for synthesis can also be directly rendered by LE3D. Thanks to the rendering speed of 3DGS, both I and $\mathcal{H}$ can be rendered in real time, enabling rapid refocusing.

After obtaining the color image I and the weight distortion map $\mathcal{H}$, we can perform refocusing using the algorithm described in Algorithm 1, similar to the method in [36, 55]. Here, i_{focus} represents the index of the focus plane, and Δ_r controls the simulated aperture size (i.e., the blur intensity). $\mathrm{blur(c,r)}$ represents performing a gaussian blur operation on c with a radius of r. This operation is accelerated using 2D Fourier space convolution.

Algorithm 1 Refocus Synthesis in LE3D

```
Require: $I$, $\mathcal{H}$, $i_{focus}$, $\Delta_r$
  $\Phi \leftarrow 0$
  for $i = 1$ to K do
    $r \leftarrow \Delta_r \cdot |i - i_{refocus}|$
    $\alpha_i \leftarrow \mathcal{H}_i$
    $\phi_i \leftarrow I \times \alpha_i$
    $\alpha_{blur} \leftarrow \mathrm{blur}(\alpha_i, r)$
    $\phi_{blur} \leftarrow \mathrm{blur}(\phi_i, r)$
    $\Phi \leftarrow \phi_{blur} + (1 - \alpha_{blur}) \cdot \Phi$
  end for
  return $\Phi$
```

In Fig. 9, we show the refocus result and multiplane image representations of LE3D and RawGS. A more specific and global result of these two methods can be found in Fig. 5 (c) and (e), their depth maps. It can be observed that the lack of structural information leads to inconsistencies in the foreground-background relationship in Fig. 9 (b), resulting in poor refocusing effects. In contrast, LE3D, with its superior structural information, achieves more realistic refocusing effects, accurately depicting the scene's depth relationships.

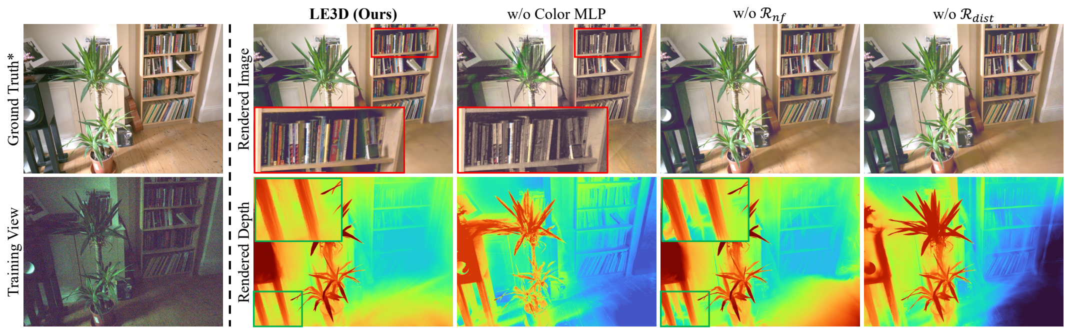


Figure 7: Visualization results for ablation studies on the test scene. The Ground Truth denotes the raw image averaged from a burst set with a slow shutter to perform denoising. It can be observed that the results of w/o Color MLP show significant color degradation, while the results of w/o $\mathcal{R}_{nf}$ and w/o $\mathcal{R}_{dist}$ exhibit structural degradation.

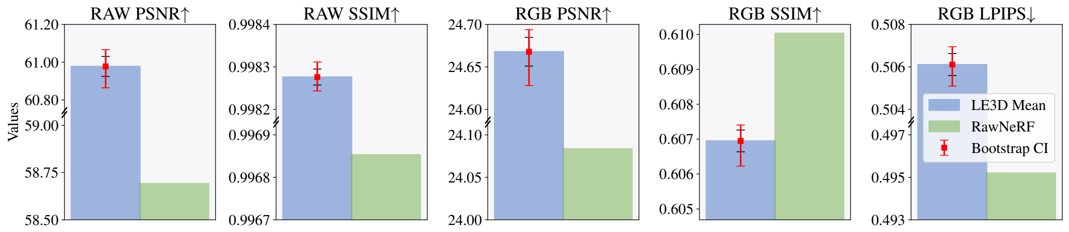


Figure 8: The error bars of our proposed LE3D and comparison with RawNeRF [36].

B.2 Tone-mapping

Default LDR tone-mapping We use the following simplified tone-mapping function to map the RAW image to an LDR sRGB image, which is similar to the ISP pipeline used in RawNeRF [36]:

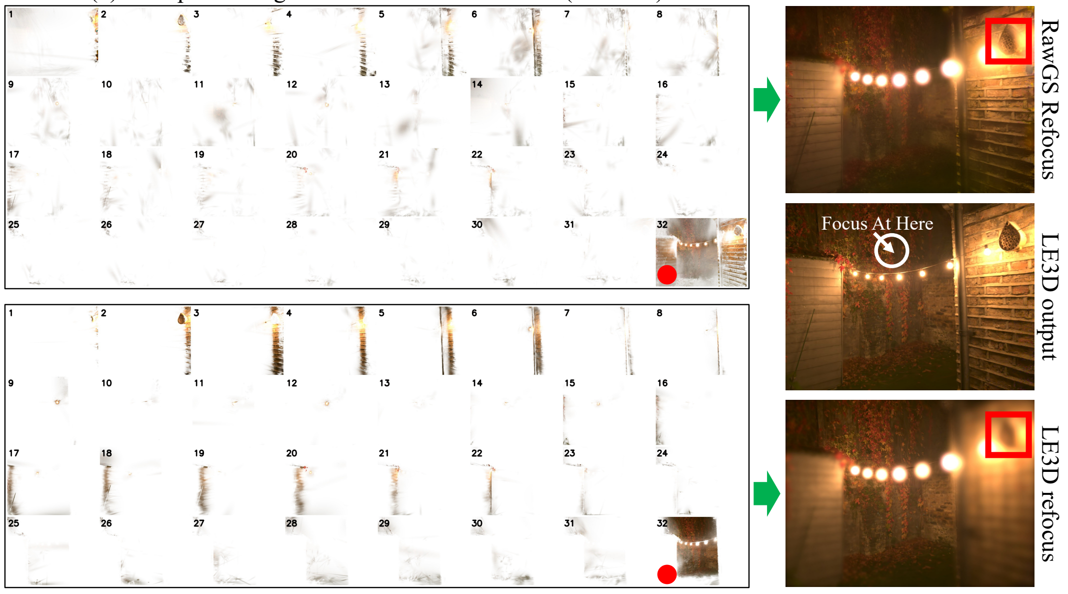
1. Normalizing the value of the input RAW image to $[0, 1]$.
2. Apply bilinear demosaicking to the normalized RAW images.
3. Apply white balance gains for each of the three channels.
4. Perform the color correction operation with the color matrix from camera RGB-to-sRGB (combined camera RGB-to-XYZ and XYZ-to-sRGB).
5. (Optional) Exposure adjustment by setting the white level to the 97 percentile.
6. Clip the value to $[0, 1]$, and apply the sRGB gamme correction.

LE3D is trained using the RAW images from step 2, resulting in outputs in the unmodified RAW linear color space. This means that LE3D's outputs are more suitable for various downstream tone-mapping applications.

Global tone-mapping In our implementation, we use two kinds of global tone-mapping techniques: color temperature and curve adjustment. For color temperature adjustment, we refer to Tanner Helland's algorithm [21]. For the curve adjustment algorithm, we adopted a control point strategy. Users first determine several control points, and then we use the "B-splines" interpolation algorithm [13] to generate a smooth curve from these control points. Since the output of LE3D is a 32-bit float, we quantize it to a 16-bit unsigned integer. The curve obtained earlier is converted into a look-up table (LUT) over the range $[0, 65535]$. The LUT is then applied to the output of LE3D to achieve the curve adjustment results. The results of our global tone-mapping can be found in Fig. 10. The color temperature adjustment is directly applied to LE3D's output, while the curve adjustment is performed after applying white balance and the color correction matrix.

Local HDR+ tone-mapping Our local tone-mapping is based on the HDR+ [20] algorithm. However, since LE3D's output is noise-free, we have simplified some of the steps. Our simplified HDR+ algorithm is as follows:

(b) Multiplane Images with Structure of RawGS (Baseline)

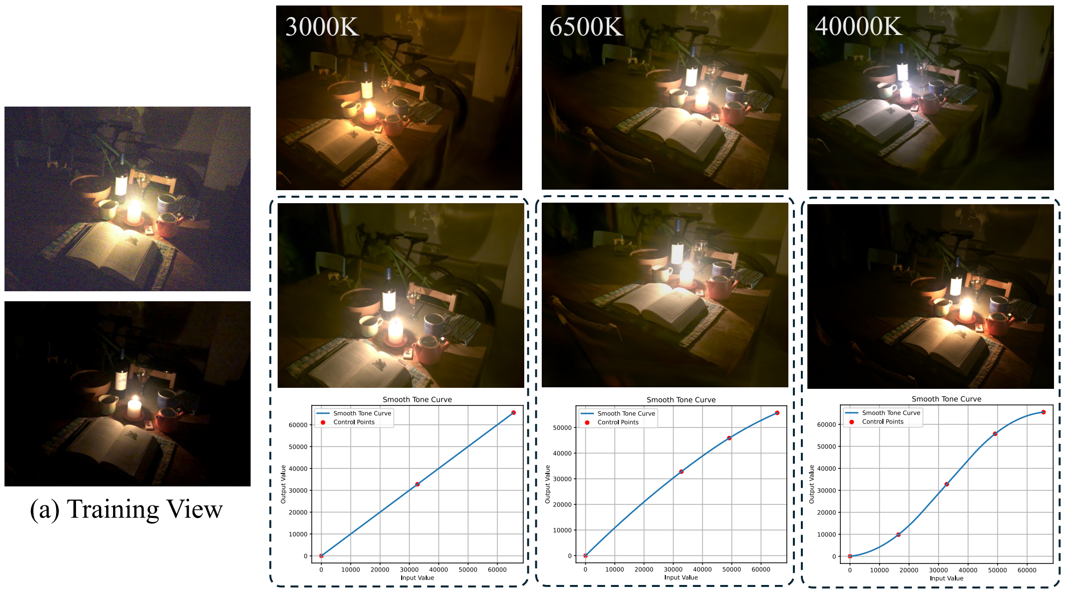


(a) Multiplane Images with Structure of **LE3D (Ours)**

Figure 9: Multiplane image representations of (a) LE3D and (b) RawGS, and their corresponding refocused images (*Zoom-in for best view*). The "●" in (a) and (b) denotes the selected focus plane. The "$\rightarrow$" represents the refocus algorithm in Algorithm 1. It can be observed that the multiplane image representations generated using RawGS structure are noticeably more "noisy" with incorrect depth relationships. This results in ambiguity in the final refocusing result regarding foreground and background relationships.

1. Apply white balance gains for each of the three channels of the LE3D 's output.
2. Perform the color correction operation with the color matrix from camera RGB-to-sRGB (combined camera RGB-to-XYZ and XYZ-to-sRGB).
3. (Optional) Apply the global tone-mapping curve if the curve exists.
4. Apply the additional digital gain for exposure adjustment, resulting in z.
5. Based on z, simulate multiple exposures $z_{ev-}, z_{ev0}, z_{ev+}$ with different exposure times (in our implementation, we use $[0.25, 1.0, 4.0]$).
6. Postporcess $z_{ev-}, z_{ev0}, z_{ev+}$ to the sRGB color space, then using ExposureFusion [35] to compress the HDR results into LDR. Then we need to re-map the merged sRGB result to RAW linear color space for further postprocess. This would not cause information loss, due to all the processes are in 32-bit float format.
7. Perform "dehazing" like HDR+ [20], which aims at reduceing the effect of veiling glare. This is done by allowing up to 0.1% of pixels to be clamped to zero, but only adjusting pixels below 7% of the white level.
8. Then we apply the Reinhard tone-mapping [40] and gamma compression to get the sRGB results.
9. We apply the unsharp masking technique to sharpen the image with a 3-level gaussian pyramid [15].
10. Convert the result into the HSV color space. Then shifting bluish cyans and purples towards light blue, and increasing the saturation of blues and greens generally. Finally, re-map the result to sRGB to get the final result.

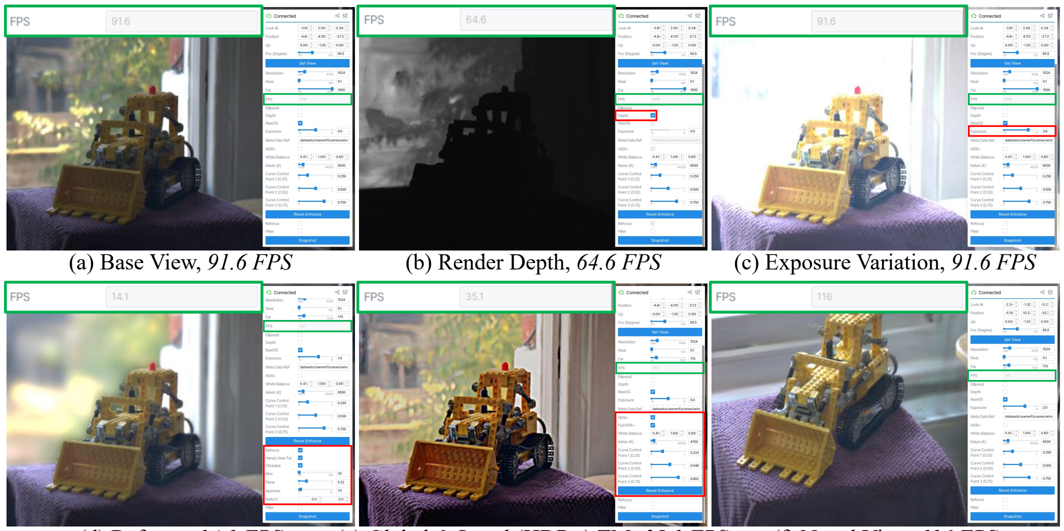
Some sampled local HDR+ tone-mapping results can be found in Fig. 1, Fig. 5 (g) and Fig. 11 (e).



(a) Training View

(b) Color Temperature / Curve Adjustment & Viewpoint Changing

Figure 10: Global tone-mapping and viewpoint changing in a single scene. The first row in (b) shows the color temperature adjustment result and the corresponding Kelvin color temperature. The second row shows the curve adjustment result the corresponding control points and the interpolated curve.



(a) Base View, *91.6 FPS* (b) Render Depth, *64.6 FPS* (c) Exposure Variation, *91.6 FPS*

(d) Refocus, *14.1 FPS* (e) Global & Local (HDR+) TM, *35.1 FPS* (f) Novel View, *116 FPS*

Figure 11: Some screenshots of our interactive viewer, which can perform (b) depth rendering, (c) exposure variation, (d) refocus, (e) global & local tone-mapping, and (f) novel view rendering. The FPS is emphasized by the green bounding box, and the changed rendering parameters are emphasized in red bounding box.

C Interactive viewer

Fig. 11 has shown some screenshots and downstream tasks results of our interactive viewer which is built upon Viser [18]. Besides refocusing, most of the downstream tasks can be performed in real time. For refocusing, most of the time is spent on the gaussian blur according to Algorithm 1 (due to the large gaussian blur kernel size) rather than on rendering.

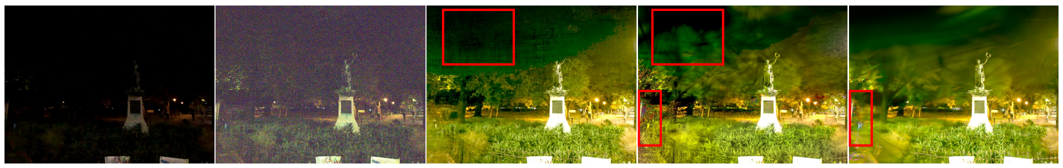

(a) Training View (b) RawNeRF [36] (c) RawGS (d) **LE3D (Ours)**

Figure 12: Failure cases for both volumetric rendering-based and differentiable rasterization (3DGS) based methods (*Zoom-in for best view*). Red bounding box highlights the failure cases.

D Limitation

Although LE3D can achieve real-time HDR view synthesis, it, like volumetric rendering-based methods (such as RawNeRF [36]), encounters reconstruction failures in extremely dark scenes. For example, in the training view shown in Fig. 12 (a), RawNeRF [36] failed to reconstruct the upper left corner, leaving large artifacts. RawGS, lacking structural supervision, produced many floater gaussians, significantly degrading the reconstruction quality. Although LE3D can reduce these floaters, it, like the other methods, fails to recover the details of the leaves in the upper left corner.

The main cause of this issue is the low scene brightness, resulting in a low signal-to-noise ratio. Potential solutions include: 1) Collecting as much data as possible during scene capture to enable multi-frame fusion "3D" denoising; 2) Using neural networks to denoise images before reconstruction, then using the denoised images to rebuild the scene. However, ensuring consistent denoising results across frames (consistent color space) is challenging, making it difficult to guarantee effectiveness.

E Broader impacts

Potential positive impacts The development of the LE3D method could lead to improvements in virtual and augmented reality applications, providing users with more realistic and immersive experiences. 3D reconstruction capabilities may be utilized for the digital preservation of cultural heritage sites, museums, and art, allowing for greater access and study by a broader audience. The capability of LE3D to reconstruct scenes in low-light environments also facilitates data collection, broadening the application scenarios for 3DGS. This advancement makes it viable to employ 3DGS in situations where establishing adequate lighting conditions is challenging. *Most importantly*, LE3D enables real-time editing of HDR view synthesis tasks (as shown in Sec. C). This further advances the development of 3D video editors.

Potential negative impacts The fast training reconstruction method like 3DGS could potentially facilitate the implementation of illegal surveillance activities. Enhancing the capabilities of 3DGS in low-light environments may render illicit surveillance practices more insidious, thereby posing a threat to individual privacy and the security of confidential information.

F Licenses for existing assets

Our method is implemented in the style of BasicSR[47], and we encourage further work based on this codebase. The dataset we utilized is sourced from RawNeRF[36]. For rendering the depth map, we employed the rasterizer by GauStudio[51], with modifications made to the rasterizer by 3DGS to implement the $\mathcal{R}_{nf}$ regularization. Our real-time viewer is based on Viser[18]. For comparative analysis and reference code, we used RawNeRF from MultiNeRF[38].

G More results

G.1 Detailed quantitative results

Since the metrics in Tab. 1 are calculated after affine alignment, we also provide the metrics without affine alignment in Tab. 5. It can be observed that even without affine alignment, LE3D still ranks

Table 4: Licenses for Existing Assets.

Asset	License	Cite
Rasterizer by 3DGS	Gaussian-Splatting License [a]	[25]
Rasterizer by GauStudio	MIT license	[51]
BasicSR	Apache-2.0 license	[47]
Viser	Apache-2.0 license	[18]
MultiNeRF	Apache-2.0 license	[38]
RawNeRF datasets [b]	Apache-2.0 license [c]	[36]

[a]License can be found at `https://github.com/graphdeco-inria/gaussian-splatting/blob/main/LICENSE.md`. This license mentions that licensors grant non-exclusive rights to use the Software for research purposes to research users (both academic and industrial), free of charge, without the right to sublicense.

[b]`http://storage.googleapis.com/gresearch/refraw360/raw.zip`

[c]Follow MultiNeRF's license.

Table 5: Quantitative results on the test scenes in the RawNeRF [36] dataset **w/o affine alignment**. The best result is in **bold** whereas the second best one is in underlined. TM indicates whether the tone-mapping function can be replaced for HDR rendering. For methods where the tone-mapping function can be replaced, the metrics on sRGB are calculated using LDR tone-mapping for a fair comparison. The FPS measurement is conducted at a 2K (2016×1512) resolution. Train denotes the training time of the method, measured in GPU×H.

Method	TM	FPS↑	Train↓	RAW		sRGB		
				PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
LDR-NeRF [37]	✗	0.007	13.66	–	–	14.4014	0.5541	0.7552
LDR-3DGS [25]	✗	153	0.75	–	–	14.7270	0.5477	0.6977
HDR-3DGS [25]	✓	238	0.73	56.3537	0.9924	20.4687	0.5286	0.6622
RawNeRF [36]	✓	0.022	129.54	58.1442	0.9956	22.3398	**0.5992**	**0.5363**
RawGS (Baseline)	✓	176	1.05	<u>58.7603</u>	**0.9963**	**22.6751**	0.5810	0.5603
LE3D (Ours)	✓	103	1.53	**59.5099**	<u>0.9959</u>	<u>22.4391</u>	<u>0.5924</u>	<u>0.5483</u>

among the top in overall metrics and achieves real-time rendering speeds. Compared to other 3DGS-based methods (all capable of real-time rendering), LE3D demonstrates significantly superior structural information, as shown in Fig. 5 and Fig. 9, enabling more downstream applications.

G.2 More qualitative results

Detailed comparisons between 3DGS-based methods and LE3D As shown in Fig. 13, LE3D achieves better structure reconstruction than our baseline RawGS (3DGS trained with RawNeRF's loss and multiple exposure training strategy). Compared with LDR-GS (trained on the LDR images) and HDR-GS (trained directly on the RAW data), LE3D achieves better color reconstruction results as well as better denoising ability. We also found that LDR-GS and HDR-GS have fewer reconstructed gaussians, resulting in faster rendering speeds but poor overall reconstruction quality. Additionally, LDR-GS, trained on linear brightened LDR images, shows weaker resistance to color bias [48], resulting in severe color shifts in the final output. We also found that the generally low values of RAW images lead to insufficient gradients, reducing the number of gaussian splitting. RawNeRF's weighted L2 loss (Eqn. (3)) strengthens supervision in dark areas but at the cost of structural information. LE3D incorporates both the weighted L2 loss and depth distortion and near-far regularizations to constrain the structure, ultimately achieving the best structural and visual results.

More qualitative results Fig. 7, Fig. 13, Fig. 14, Fig. 15 has shown more qualitative comparisons between LE3D and 3DGS [25]-based methods. From the figures above, it is evident that LE3D demonstrates superior noise resistance and color representation capabilities. Additionally, LE3D produces smoother and more accurate depth maps, which are essential for downstream tasks like refocusing. It is worth noting that volumetric rendering-based methods, such as RawNeRF [36],

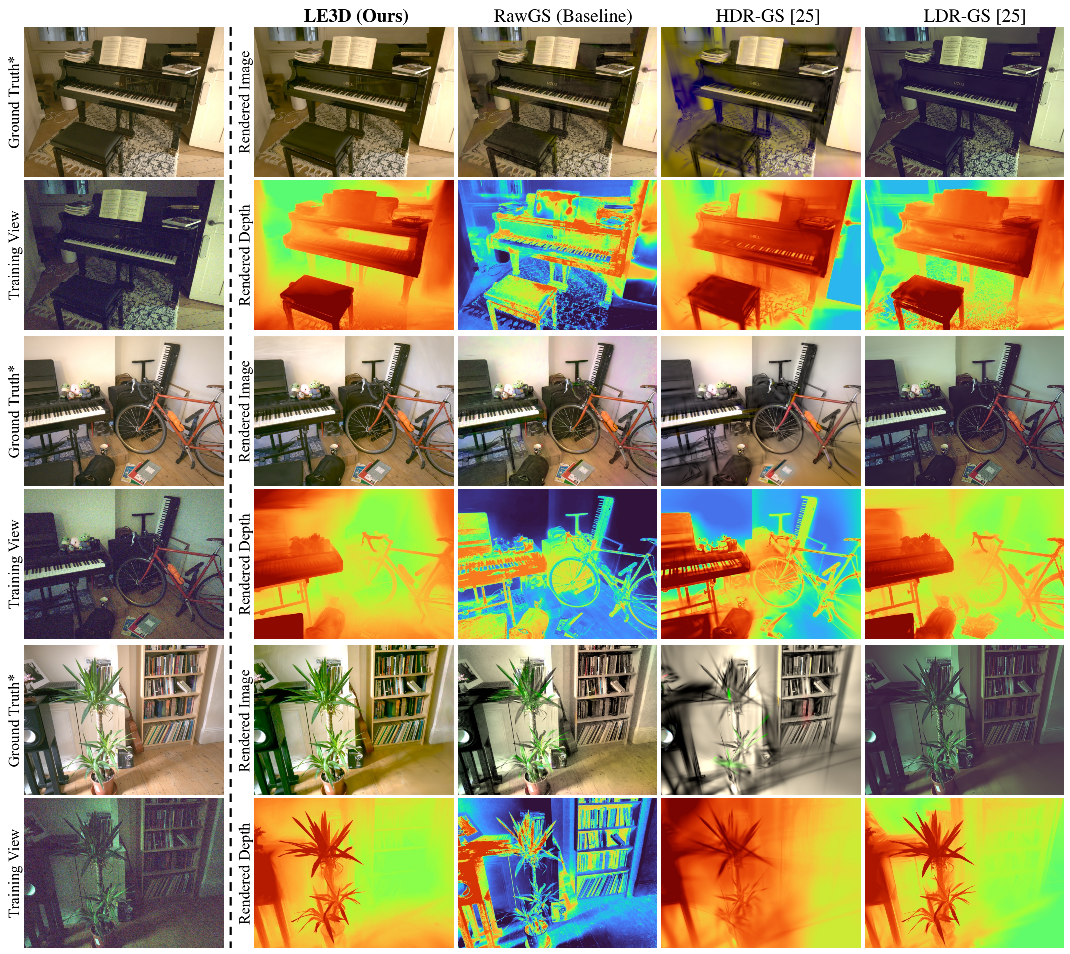


Figure 13: Comparison between LE3D and other 3DGS-based methods (*Zoom-in for best view*). All the results are the direct output of each model, not being applied by affine alignment. The Ground Truth denotes the raw image averaged from a burst set with a slow shutter to perform denoising.

cannot achieve real-time rendering, which significantly limits their applications (including real-time scene editing). Therefore, we do not compare them here. For comparisons between LE3D and volumetric rendering based methods, please refer to Tab. 1, Tab. 5, and Fig. 3.

H Concurrent work

Here list of some concurrent works that focus on HDR view synthesis based on the 3DGS-like method.

1) Training with RAW images: HDRSplat [42] uses an extra Bayer-space-denoising step to handle noise. 3DGS in the Dark [31] analysis how noise impacts the optimization of 3DGS, and purposes Noise robust reconstruction loss to handle noise.

2) Training with multi-exposure RGB images: HDR-GS [4] purposes Dual Dynamic Range (DDR) Gaussian point cloud model to jointly fit the HDR and LDR colors, and employs three independent MLPs to learn the tone-mapping operation in RGB channels.

Compared to those methods that need a trained denoising network, our LE3D could just denoise a set of noisy RAW images from a single-blind device. It means we do not need to train the denoising network from scratch for every device like HDRSplat [42] and 3DGS in the Dark [31]. And since our method uses high dynamic range data (RAW images) directly, so we do not need multi-exposure data for training like HDR-GS [4], which simplifies the data capture pipeline.

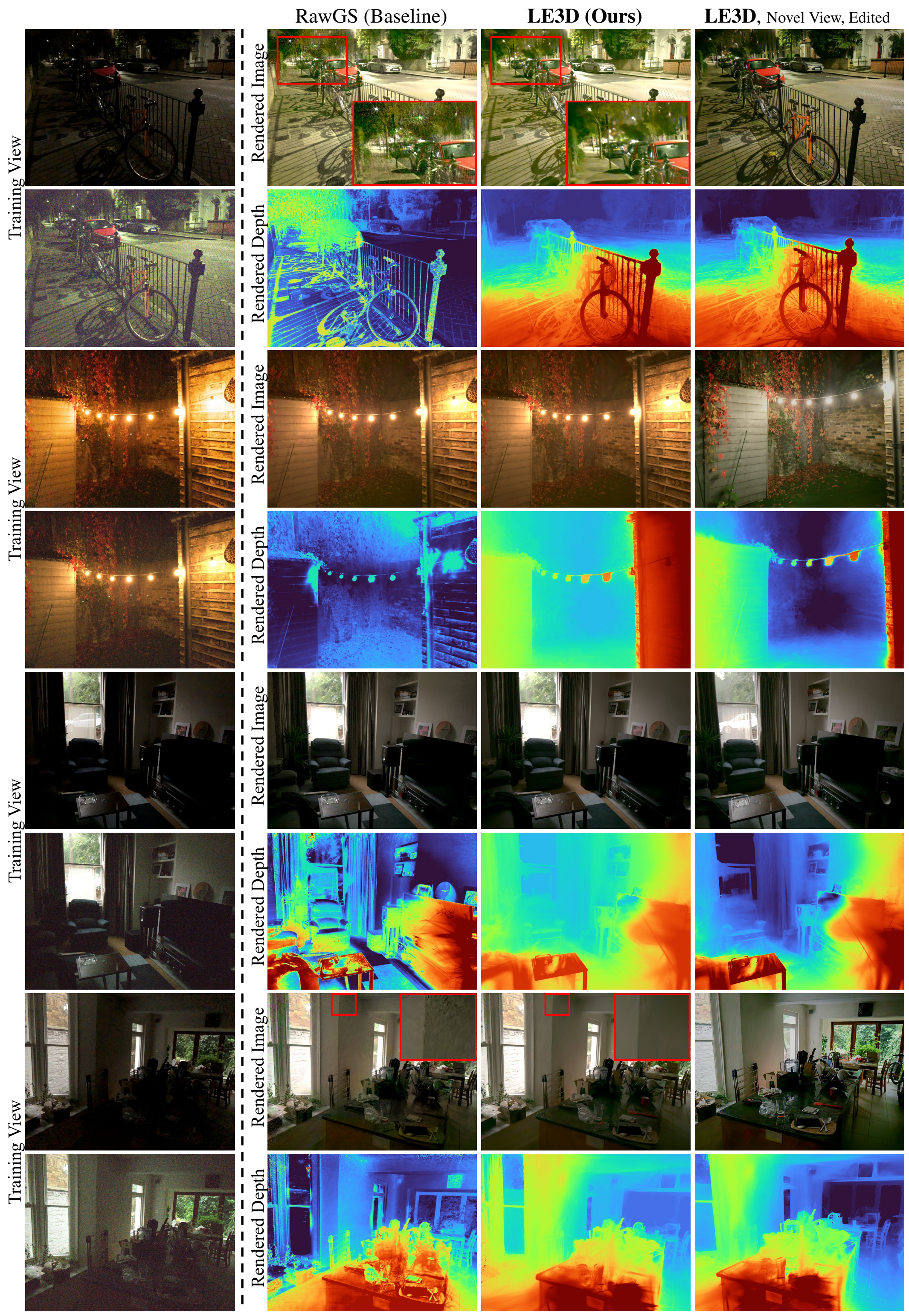


Figure 14: Comparison between LE3D and RawGS (baseline, 3DGS trained with weighted L2 loss in Eqn. (3) and multiple exposure strategy). It can be observed that LE3Dexhibits stronger noise resistance and color representation in low-light scenes. Additionally, it produces smoother and more accurate depth maps across all scenes.

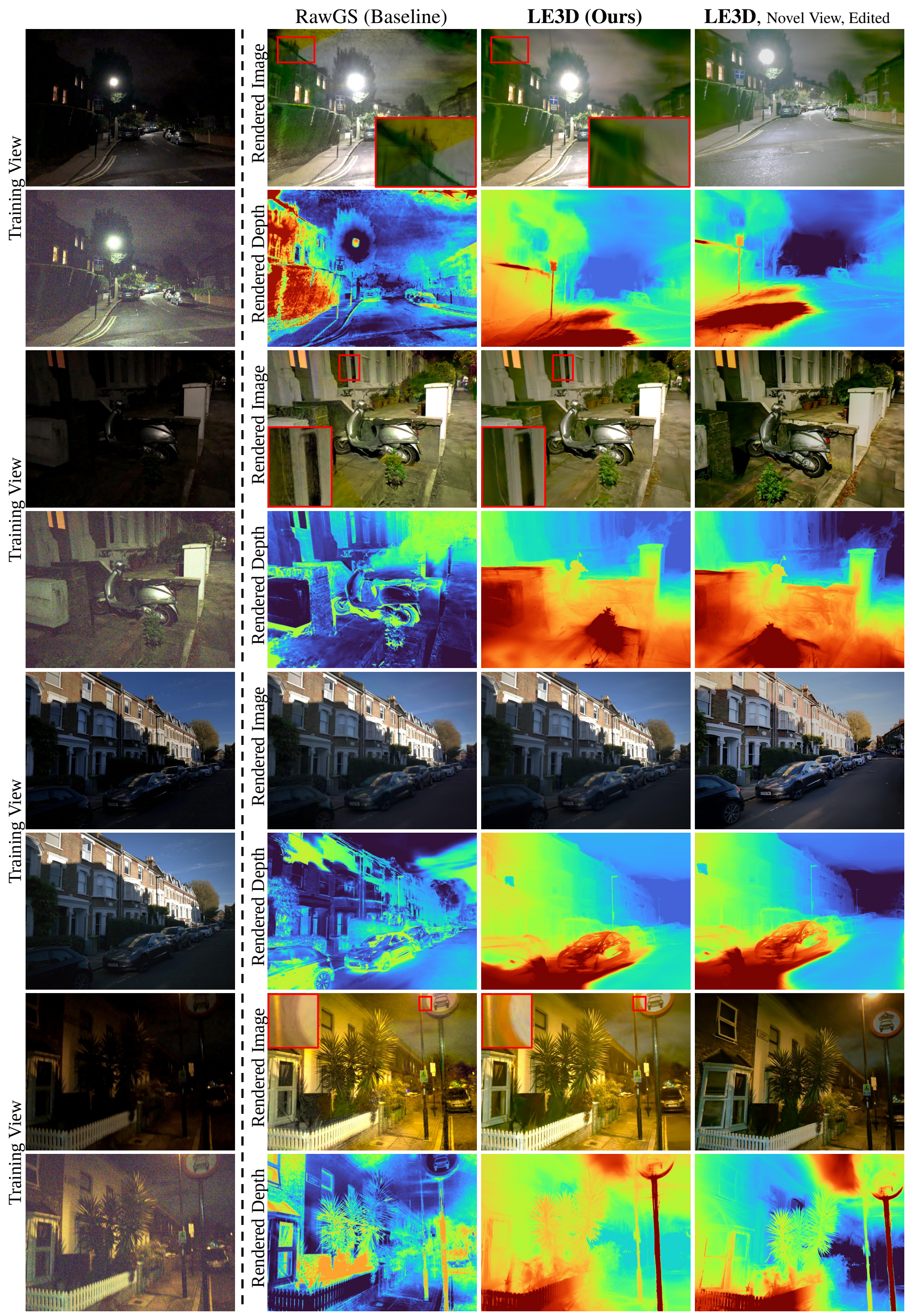


Figure 15: Comparison between LE3D and RawGS (baseline, 3DGS trained with weighted L2 loss in Eqn. (3) and multiple exposure strategy). It can be observed that LE3Dexhibits stronger noise resistance and color representation in low-light scenes. Additionally, it produces smoother and more accurate depth maps across all scenes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have described our motivation and contributions both in abstract and Sec. 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have described our limitation in Sec. D

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We do not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the implementation details can be found in Sec. 5.1 and Sec. A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: No reproducible code is provided in the submission. However, all the codes, both to reproduce the results and the interactive viewer, will be released.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the implementation details can be found in Sec. 5.1 and Sec. A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The error bar can be found in Sec. A.2 and Fig. 8.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The necessary computer resources LE3D needed can be found in Sec. A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: 1) Our experiments and research are not with human subjects; 2) the RawNeRF [36] dataset is public (`http://storage.googleapis.com/gresearch/refraw360/raw.zip`); 3) we have discuss our potential broader impacts.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Sec. E.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Methods and models for reconstruction task is hardly been misused.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Licenses for existing assets are listed in Sec. F.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: No new assets is submitted along with the submission. However, all the codes, both to reproduce the results and the interactive viewer, will be released.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.