

# HDRGS: High Dynamic Range Gaussian Splatting

Jiahao Wu<sup>1</sup>, Lu Xiao<sup>1</sup>, Rui Peng<sup>1</sup>, Kaiqiang Xiong<sup>1</sup>, Ronggang Wang<sup>1\*</sup>

<sup>1</sup>School of Electronic and Computer Engineering, Peking University

## Abstract

Recent years have witnessed substantial advancements in the field of 3D reconstruction from 2D images, particularly following the introduction of the neural radiance field (NeRF) technique. However, reconstructing a 3D high dynamic range (HDR) radiance field, which aligns more closely with real-world conditions, from 2D multi-exposure low dynamic range (LDR) images continues to pose significant challenges. Approaches to this issue fall into two categories: grid-based and implicit-based. Implicit methods, using multi-layer perceptrons (MLP), face inefficiencies, limited solvability, and overfitting risks. Conversely, grid-based methods require significant memory and struggle with image quality and long training times. In this paper, we introduce Gaussian Splatting—a recent, high-quality, real-time 3D reconstruction technique—into this domain. We further develop the High Dynamic Range Gaussian Splatting (HDR-GS) method, designed to address the aforementioned challenges. This method enhances color dimensionality by including luminance and uses an asymmetric grid for tone-mapping, swiftly and precisely converting pixel irradiance to color. Our approach improves HDR scene recovery accuracy and integrates a novel coarse-to-fine strategy to speed up model convergence, enhancing robustness against sparse viewpoints and exposure extremes, and preventing local optima. Extensive testing confirms that our method surpasses current state-of-the-art techniques in both synthetic and real-world scenarios. Code will be released at <https://github.com/WuJH2001/HDRGS>

## 1 Introduction

In recent years, significant progress has been made in 3D reconstruction technology [50, 31, 18], but these technologies typically assume constant exposure conditions of input images, thus restoring scenes with low dynamic range (LDR). However, high dynamic range (HDR) scenes [37, 46], which are more consistent with the physical world, offer a broader dynamic range and provide a superior visual experience for humans. Traditional HDR image reconstruction techniques [29, 47, 8] still focus on 2D images. How to reconstruct 3D HDR scenes from multi-exposure unstructured LDR images remains a question worthy of investigation.

Current methodologies in this domain can be divided into two distinct categories: explicit-based, represented by HDR-plenoxel [17], and implicit-based, represented by HDR-NeRF [16]. While these methods have achieved some impressive results, HDR-plenoxel relies on grids and spherical harmonics, complicating the construction of high-quality 3D HDR radiance fields and requiring substantial memory. HDR-NeRF models the entire HDR scene and Camera Response Function (CRF)[10] using implicit MLPs, resulting in poor interpretability and slow training and rendering speeds, which poses challenges for real-world applications. With the recent introduction of 3D Gaussian Splatting technology [18], which can efficiently and effectively complete 3D reconstruction, it appears to offer a direction for addressing the aforementioned issues.

\*Corresponding author

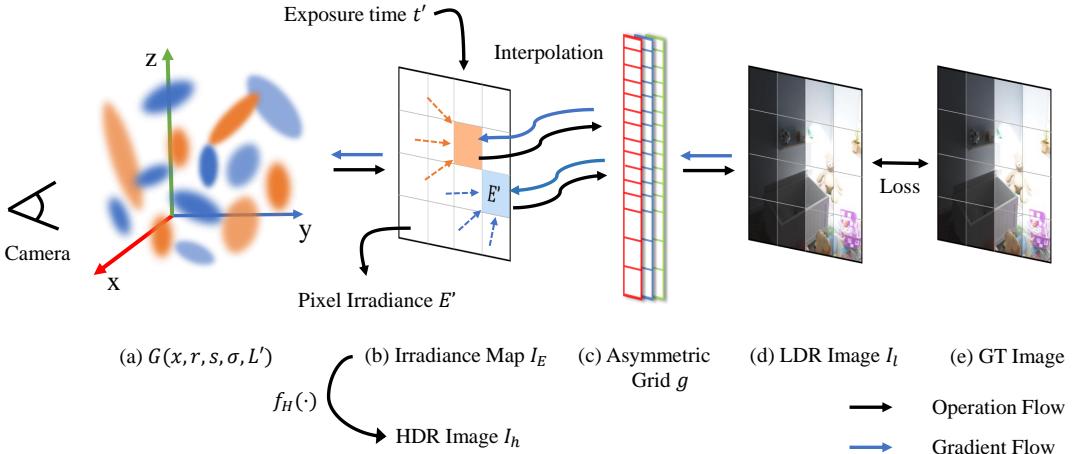


Figure 1: **Illustration of HDRGS.** We redefine the color of Gaussian points as radiance  $L$ , enabling it to meet the primary requirement for reconstructing the HDR radiance field. After splatting, pixel irradiance  $E'(\mathbf{p})$  is obtained. Then, the differentiable asymmetric grid  $g$  maps the exposure under exposure time  $t$  to LDR pixel color. The HDR image  $I_h$  can be derived by applying the function  $f_H(\cdot)$  to each  $E'$ , with detailed explanations provided in subsequent sections.

In this paper, we introduce a method based on 3D Gaussian Splatting (HDRGS) that can reconstruct a 3D HDR scene by simulating the complete physical imaging process of a camera, projecting radiance to pixel color. Each Gaussian point radiates radiance  $r$  (with potential values up to positive infinity). Through the splatting, we capture the pixel irradiance  $E$  on the HDR image plane. The exposure from irradiance  $E$  over a specific exposure time  $t$  is then transformed through a series of functions to compute pixel values  $C$ . One critical aspect of our method involves modeling these function transformations, particularly addressing the uneven distribution of irradiance  $E$ . We developed an uneven asymmetric grid  $g$  to facilitate a rapid and efficient mapping from irradiance  $E$  to pixel value  $C$ , ensuring that the entire process remains differentiable and only requires LDR images for supervision to accurately construct the HDR radiance field. To mitigate the grid’s discrete nature and the tendency to fall into local optima, we implement a novel coarse-to-fine strategy. Initially, a sigmoid function serves as the tone mapper to initialize Gaussian points during the coarse phase. In the subsequent fine phase, we employ the asymmetric grid  $g$  as the tone mapper, co-learning its parameters with the Gaussian points.

Our method’s effectiveness is demonstrated through comparative experiments with other approaches [31, 28, 17, 16, 18] and ablation studies on our primary modules. The results indicate that our model surpasses current state-of-the-art methods.

- 1) Our method can efficiently reconstructs HDR radiance fields from a series of LDR images with varied exposures within minutes, and it consistently surpasses current state-of-the-art methods across both synthetic and real datasets.
- 2) We introduce a novel coarse-to-fine strategy that enhances the model’s ability to reconstruct complex scenes, significantly improving both reconstruction speed and image quality.
- 3) We propose a differentiable, asymmetric grid to model the tone mapper, providing enhanced expressive power over symmetric grids.

## 2 Related work

**HDR Imaging.** The typical approach in traditional HDR imaging [37] is crucial for enhancing people’s immersive visual experiences. Currently, tone mapping, generating LDR from HDR, is a mature field [37, 47], but its inverse tone mapping (ITM) [38], recovering HDR from LDR, still faces many challenges. The simplest method is multi-frame LDR image synthesis with different exposures to create an HDR image. As the name suggests, this involves extracting the most visible parts from each LDR image and combining them into a single HDR image. However, such methods

based on multiple exposures [8] are typically constrained by the need for multiple exposure LDR images and are prone to artifacts caused by camera movements during LDR image capture. The emergence of deep learning in recent years has brought new solutions, with learning-based methods being categorized into indirect and direct approaches. Indirect methods [12, 21] typically predict multiple LDR images with different exposures, which are then merged into an HDR image (Indirect ITM), while direct methods [11, 27] involve neural networks directly predicting HDR images (Direct ITM). However, these methods can only synthesize HDR images with the original camera poses.

**Novel View Synthesis.** Previous methods formalized the light field [22] or Lumigraph [5] and generated novel views by interpolating from existing views. This approach required densely captured images for realistic rendering. Subsequently, some geometry-based methods were proposed. Mesh-based methods [9, 39, 43, 45, 1] supported efficient rendering but struggled with surface optimization. Volume-based methods used voxel grids [20, 35, 40] or multi-plane images (MPI) [2, 30, 41, 54], offering detailed rendering but suffering from low memory efficiency or being limited to small view variations. In recent years, the method proposed by Neural Radiance Field (NeRF) [31] to model 3D scenes using MLP has gained widespread recognition. This has led to a series of related research efforts, such as speeding up training and inference speeds [7, 14, 32], improving rendering quality [4, 44], view synthesis for dynamic scenes [36, 6, 13]. However, these methods have consistently struggled to achieve real-time rendering. Last year, 3DGS [18] emerged and addressed this issue for people. Not only that, but 3DGS also offers higher rendering quality and interpretability. More and more people are getting involved in related research [52, 15, 23].

### 3 Method

Given a series of low dynamic range (LDR) images captured under multiple exposure conditions from different viewpoints, our task is to efficiently reconstruct a high-quality high dynamic range (HDR) radiance field solely from these LDR images and obtain HDR images through rendering. The entire framework is illustrated in the Fig 1. In this section, we will provide a detailed introduction to each component of our method. In Sec.3.2, we introduce the basic process of rendering and tone mapping. In Sec.3.3, we discuss the design of our tone mapper function. Then, in Sec.3.4, we introduce our coarse-to-fine strategy, followed by the optimization process in Sec.3.5.

#### 3.1 Preliminary

3DGS initializes with a sparse point cloud generated from Structure-from-Motion (SfM) [5]. This method models geometric shapes as a set of 3D Gaussian functions defined by covariance matrices and means in world space.

$$G(x) = e^{-\frac{1}{2}(x - \mu_{3D})^T \Sigma_{3D}^{-1} (x - \mu_{3D})} \quad (1)$$

Where  $\mu_{3D}$  is the mean of the Gaussian point. To ensure the 3D Gaussian covariance matrix retains physical meaning, specifically to maintain positive semi-definiteness, it can be further decomposed into a scale matrix  $S$  and a rotation matrix  $R$ . Thus, the 3D covariance matrix can be represented as:

$$\Sigma_{3D} = RSS^T R^T \quad (2)$$

where  $\Sigma_{3D}$  is the covariance matrix of the Gaussian point. To render the image, first approximate the projection of the 3D Gaussian into 2D image space using perspective transformation. Specifically, the projection of the 3D Gaussian is approximated as a 2D Gaussian with center  $\mu_{2D}$  and covariance  $\Sigma_{2D}$ .  $\mu_{2D}$  and  $\Sigma_{2D}$  are computed as:

$$\mu_{2D} = (K((W\mu_{3D})/(W\mu_{3D})_z))_{1:2} \quad \Sigma_{2D} = (JW\Sigma_{3D}W^T J^T)_{1:2, 1:2} \quad (3)$$

$W$  and  $K$  are the viewing transformation and projection matrix. Finally, after sorting the Gaussian points by depth, the pixel  $p$  value can be computed as:

$$C_p = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad \text{where } \alpha = \sigma \exp \left( -\frac{1}{2} (\mathbf{x} - \mu_{2D})^T \Sigma_{2D}^{-1} (\mathbf{x} - \mu_{2D}) \right). \quad (4)$$

$c_i$  refers to the RGB color of Gaussian point which is represented by spherical harmonics (SH).

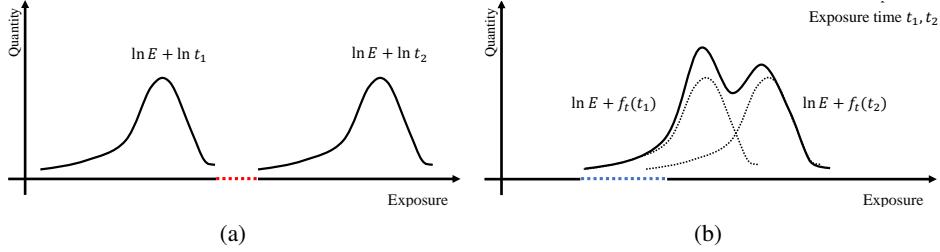


Figure 2: **The distribution of accumulated exposure.** The x-axis represents the exposure values from a given camera viewpoint, and the y-axis represents the number of pixels with the corresponding exposure values. (a) represents the original distribution of accumulated exposure, while (b) represents the distribution after applying time scaling  $f_t(\cdot)$ .

### 3.2 Basic process

**Radiance-based  $\alpha$  composition.** According to the principles of physical imaging, the radiance  $L$  emitted by objects in the scene is transformed into irradiance  $E$  on the surface of the image sensor as it passes through the camera lens. To simulate this process, we redefine the color of Gaussian points as radiance  $L$ , which results in the pixel values of the image formed by Gaussian points splatting onto the HDR plane no longer representing color  $C$ , but irradiance  $E$ . We can rewrite the rendering formula of 3DGS[18] 4 as follows:

$$E(\mathbf{p}) = \sum_{i=0}^N L_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \text{ where } E(\mathbf{p}) \in (0, +\infty) \quad (5)$$

In which  $\mathbf{p}$  means the pixel,  $E$  represents pixel irradiance of  $\mathbf{p}$ , as shown in Fig 1(b),  $L_i$  represents the radiance of  $i$ -th Gaussian point used to render pixel  $\mathbf{p}$ , as shown in Fig 1(a).

**Imaging process.** The total irradiance received by the image sensor within the exposure time  $t$  results in the accumulated exposure. After undergoing photoelectric conversion by the sensor, analog-to-digital conversion, etc. the LDR pixel value  $C$  is obtained. The entire imaging process can be represented by a function called the camera response function (CRF)  $F(\cdot)$ . Combining  $E(\cdot)$ , we represent the entire imaging process with the following formulation[42]:

$$C(\mathbf{p}, t) = F(E(\mathbf{p}) * t(\mathbf{p})) \quad (6)$$

where  $t$  represents the exposure time of the camera capturing a light ray (or pixel), which depends on the shutter speed. Following the classic non-parametric CRF calibration method by Debevec and Malik[8], we assume that the CRF  $F(\cdot)$  is monotonic and invertible. Therefore, we can rewrite the Eq. 6 as:

$$\ln F^{-1}(C(\mathbf{p}, t)) = \ln E(\mathbf{p}) + \ln t(\mathbf{p}) \quad (7)$$

After further simplification, it can be expressed as the following equation:

$$C(\mathbf{p}, t) = g(\ln E(\mathbf{p}) + \ln t(\mathbf{p})), \text{ where } g = (\ln F^{-1})^{-1} \quad (8)$$

The tone mapper function can thus be transformed into the function  $g(\cdot)$ , where we use an asymmetric grid to model. We will illustrate the details of the asymmetric grid in Sec.3.3. Here, for convenience, we disregard the logarithm and denote  $\ln E(\mathbf{p}) + \ln t(\mathbf{p})$  as exposure.

In practical operations, to streamline parameter learning and minimize the number of iterations required, we directly learn the value of  $\ln E$ , the logarithmic domain of  $E$ . Please note that the radiance of the Gaussian points learned after such transformation also change. We denote it as  $L'$ .

### 3.3 Grid-based tone mapper

To faithfully simulate the physical imaging process, in this section, we introduce our asymmetrical grid-based tone mapper  $g$  to model the process from pixel irradiance  $E'$  to color  $c$ , as shown in fig.1(b). Recently, there have been several grid-based methods for modeling 3D scenes, such as [51,

6, 13, 14, 32], etc. These grid-based methods accelerate model training while ensuring rendering quality. Influenced by their work, we designed an asymmetric grid to model our tone mapper. The difference is that we use an asymmetric grid, while they use a symmetric one. We only need one tone mapping module to map irradiance to color for any seen or unseen viewpoint.

**Learned asymmetric grid.** We have empirically found that in some scenarios, the distribution of irradiance values is highly uneven. For instance, the majority of irradiance values might fall within the range (a, b), but a small portion may also lie within the range (b, c). If a uniform symmetric grid is used, modeling these two regions presents the following drawbacks: using a sparse grid in the range with dense irradiance distribution can lead to inaccurate mapping, while using a dense grid in the range with sparse irradiance distribution can cause overfitting. Therefore, as shown in Fig.1(c), we introduce an asymmetric grid, which refers to a grid that lacks both central and density symmetry. In the range of values with a dense irradiance distribution (a,b), we use a dense grid (128 nodes per unit) to model the tonemapping function. Conversely, we use a sparse grid (64 nodes per unit) for regions with a sparse irradiance distribution (b,c). It offers greater flexibility and expressiveness compared to the symmetric grid [51, 6, 13, 14, 17, 32], especially in the scene with uneven irradiance distribution. Considering generality, we usually set  $b=0$ , but in some extreme cases,  $b$  needs to be adjusted according to the scenario.

In the training phase, gradients would not propagate backward if boundary values were merely assigned to the small percentage of exposure values that fall beyond the grid range. Therefore, we design the following function to deal with these exposure values:

$$g_{\text{leaky}}(x) = \begin{cases} \beta(x - a), & x < a \\ g(x), & a \leq x \leq b \\ -\frac{\beta}{\sqrt{(x-b+1)}} + \beta + 1, & b < x \end{cases} \quad (9)$$

In this function, we usually set  $\beta = 0.01$ . According to Debevec and Malik[8], the asymmetric grid  $g$  also needs to satisfy  $g(a) = 0$  and  $g(b) = 1$ . Rewriting the above Eq.8, we have:

$$C(\mathbf{p}, t) = g_{\text{leaky}}(E_{ln}(\mathbf{p}) + t_{ln}(\mathbf{p})) \quad (10)$$

Where  $E_{ln}$  and  $t_{ln}$  represent the logarithmic domains of  $E$  and  $t$  respectively.

**Exposure time scaling.** During experiments, directly optimizing the asymmetric grid  $g$  as described in Eq. 10 presents considerable challenges. If the exposure time of training dataset is  $\{t_1, t_2\}$ . Given a camera viewpoint, the exposure distribution received by its camera plane may resemble that in Fig.2(a). There is a significant gap (marked in red) between the two curves is evident. This results in unoptimized regions in the grid  $g$  during the training phase. Therefore, when inferring LDR images with an exposure time of  $t_{1.5}$  (where  $t_1 < t_{1.5} < t_2$ ), errors occur because the grid region (marked in red) for exposure  $t_{1.5}$  has not been trained. Consequently, we apply scaling and an offset to the logarithm of the exposure time  $t$  to mitigate these issues:

$$t' = f_t(t) = r \cdot t_{ln} + s \quad (11)$$

We determine  $r$  based on the exposure time, and if the exposure levels in the training data are too high, we need to scale them. Given a training set with monotonically increasing exposure times  $\{t_1, t_2, t_3, \dots, t_n\}$ , the formulas for determining  $r$  and  $s$  is:

$$r = \min_i \left\{ \left( \frac{2t_i}{t_{i+1}} \right) \right\} \quad s = -\frac{r \times (\ln t_{\max} + \ln t_{\min})}{2}. \quad (12)$$

where  $t_{\max}$  and  $t_{\min}$  are the maximum and minimum values of exposure time  $t_i$ . After the above processing, the distribution of exposure changes from Fig.2(a) to Fig.2(b). In Fig.2(b), the variance of exposure time is reduced, and the two independent areas converge and merge into one, resolving the issues of separate optimization and eliminating unoptimized zone.

Another noteworthy point is that applying the scaling and offset to  $t_{ln}$  will result in a change in the originally expected  $E_{ln}$ . Instead of learning  $E_{ln}$ , the model learns another pixel irradiance value, denoted as  $E'$ . The relationship between  $E'$  and  $E$  will be explained shortly. Rewriting the Eq.10, the relationship between pixel irradiance  $E$  and LDR color  $C$  can be expressed as:

$$C(\mathbf{p}, t) = g_{\text{leaky}}(E'(\mathbf{p}) + t'(\mathbf{p})) \quad (13)$$

This equation is straightforward yet powerful, capturing both our implementation approach and the underlying physical imaging process. The success of our method is largely attributed to the ability to

mathematically describe each module’s design. As illustrated in Fig.1(d), this equation enables our model to render LDR images across a range of exposure times.

**HDR mapping function.** As illustrated in Fig.1(b), the model iteratively updates and learns  $E'$  via gradient backpropagation. However,  $E'$  is not the desired HDR value  $E$ . They are related through a mapping function defined as follows:

$$E = f_H(E') = e^{((E'+s)/r)} \quad (14)$$

Please refer to the Supplement for more proof details. Through training, we obtain the corresponding HDR image  $I_h$ , where each pixel value represents HDR value  $E$ .

### 3.4 Coarse to fine optimization strategy

The above sections describe the main process of our training pipe. However, we empirically found that directly using a grid as the tonemapping function and jointly training it with the attributes of Gaussian points leads to severe coupling and overfitting to the training dataset, as shown in Fig.7. Therefore, we designed a coarse-to-fine strategy: during the coarse phase, we use a fixed function  $g_s$  as the tonemapper and train only the attributes of the Gaussian points. In the fine phase, we then use the grid as the tonemapper and jointly train it with the attributes of the Gaussian points. Now, the next question is which function  $g_s$  we should choose as the tonemapper during the coarse phase. According to Debevec [8], we can opt for a monotonically increasing, smooth function bounded between 0 and 1 as our tone mapper for pre-training the Gaussian points. For simplicity, we opt for the sigmoid function. Therefore, in the coarse phase, the function relationship between pixel irradiance  $E'$  and LDR color  $C$  can be expressed as:

$$C(\mathbf{p}, t) = g_s(E'(\mathbf{p}) + t'(\mathbf{p})) \quad (15)$$

During the coarse phase, we typically set the number of training iterations to 6000, and the training takes approximately 50 seconds. We also noticed that [49] solely employs a sigmoid function as their tone mapper. Due to the limited expressive power of the sigmoid function, this can lead to their HDR images experiencing floating-point issues and biasing towards white. Lastly, we will highlight the significance of the coarse stage in ablation research Tab.2.

### 3.5 Loss function

**Reconstruction Loss.** We adopt the same loss function as in 3DGS [18] to constrain our Gaussian points:

$$\mathcal{L}_{rec} = (1 - \lambda_1)\mathcal{L}_1 + \lambda_1\mathcal{L}_{D-SSIM}, \quad (16)$$

**Smooth Loss:** To ensure that our grid  $g$  conforms to the CRF properties proposed by Debevec [8] that CRFs increase smoothly, we employ the loss function [17]:

$$\mathcal{L}_{smooth} = \sum_{i=1}^N \sum_{e \in [a,b]} g_i''(e)^2, \quad (17)$$

Where the  $g_i''(e)$  denotes the second order derivative of grid w.r.t.  $e$  in the domain  $(a, b)$  of the grid.

**Unit Exposure Loss.** Learning HDR radiance fields solely from LDR images may result in radiance fields with various scales  $\alpha E$ . To ensure consistency between the HDR radiance fields we construct and those generated by Blender, facilitating HDR quality evaluation, we employ the loss function introduced by HDRNeRF[16]:

$$\mathcal{L}_u = \|g(0) - C_0\|_2^2. \quad (18)$$

Where  $C_0 = 0.73$ . The ablation studies shown in Table. 2 and Fig. 3.

**Total loss.** Finally, by combining the aforementioned loss functions, we derive the total loss function as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \lambda_2\mathcal{L}_{smooth} + \lambda_3\mathcal{L}_u \quad (19)$$

Table 1: Quality Comparison. LDR-OE represents the use of LDR images with exposures  $\{t_1, t_3, t_5\}$  as the training set, while LDR-NE represents the use of LDR images with exposures  $\{t_2, t_4\}$  as the training set. The exposure times of the test dataset are  $\{t_1, t_2, t_3, t_4, t_5\}$ . HDR denotes the HDR results. The **best** and the **second best** results are denoted by red and blue.

		LDR-OE			LDR-NE			HDR					
		PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	HDR	VDP-Q $\uparrow$	PUPSNR $\uparrow$	PUSSIM $\uparrow$	FPS $\uparrow$	Time $\downarrow$
NeRF	syn.	13.97	0.555	0.376	—	—	—	—	—	—	—	—	—
	real.	14.95	0.661	0.308	—	—	—	—	—	—	—	—	—
NeRF-W	syn.	29.83	0.936	0.047	29.22	0.927	0.050	—	—	—	—	—	—
	real.	28.55	0.927	0.094	28.64	0.923	0.089	—	—	—	—	—	—
HDR-NeRF	syn.	36.72	0.951	0.044	34.43	0.948	0.064	6.56	20.33	0.527	<1	9.4 hours	
	real.	31.88	0.950	0.067	30.43	0.946	0.078	—	—	—	<1	8.2 hours	
3DGS	syn.	11.24	0.395	0.534	11.97	0.431	0.489	—	—	—	352.5	12.7 mins	
	real.	11.87	0.611	0.355	12.99	0.682	0.339	—	—	—	391.7	11.3 mins	
Ours	syn.	39.16	0.974	0.012	38.84	0.975	0.012	9.70	23.67	0.835	248.8	6.8 mins	
	real.	33.34	0.967	0.023	32.77	0.960	0.030	—	—	—	210.9	8.5 mins	

## 4 Experiments

In this section, we will present our experimental setup and the dataset used in Sec.4.1 and 4.2 respectively. We will go into more detail about the experiment results in Sec.4.3. Our experiments on synthetic and real datasets demonstrate that our method achieves state-of-the-art performance.

### 4.1 Implementation details

To avoid the explosion of point numbers during training in complex scenes, we employ the pruning strategy proposed by [34], as detailed in the supplementary material. The Gaussian point parameters are set in the same way as in 3DGS[18]. The loss function parameters for synthetic datasets are  $\lambda_2 = 0.3$ ,  $\lambda_3 = 0.5$ , but for real datasets they are  $\lambda_1 = 0.2$ ,  $\lambda_2 = 1e - 3$ ,  $\lambda_3 = 0$ . We utilize the Adam optimizer[19] for training. As for the learning rate of the asymmetric grid, we initially set it to 0.02, and then decayed to 5e-6. Our usual setting in sparsely interpolated areas is less than 64 nodes. The entire model is trained on an NVIDIA A100, with 6000-14000 epochs for the coarse phase, completing in under 1 minute, and 17000-30000 epochs for the fine phase, totaling approximately 4-8 minutes of training time. We also test our code on the Tesla V100, and the training time does not exceed 10 minutes. Our model runs consistently with a GPU memory use of less than 5GB during training.

### 4.2 Dataset

The dataset we utilize is provided by HDRNeRF[16], consisting of 8 synthetic scenes rendered by Blender and 4 real scenes captured by digital cameras. Each dataset comprises 35 different viewpoints captured by a forward-facing camera, with each viewpoint having 5 exposure levels  $\{t_1, t_2, t_3, t_4, t_5\}$ , ranging from -4EV to 5EV. We strictly followed HDRNeRF’s training and measurement protocols, employing one image per exposure level for each view, resulting in a total of 18 training images. The test images consist of 85 images, covering 5 exposure levels across 17 views. The resolution of each view in synthetic scenes is  $400 \times 400$  pixels, whereas in real scenes, it is  $804 \times 534$  pixels.

### 4.3 Evaluation

We will compare our method with the following baseline methods: NeRF [31], NeRF-w [28], HDRNeRF [16], HDR-Plenoxel [17], and 3DGS [18]. Additionally, we conducted ablation studies on some of our key modules, as presented in Table. 2. The metrics measured include training time, PSNR, SSIM [48], LPIPS [53], High Dynamic Range-Visual Difference Predictor (HDR-VDP) [24, 25, 26, 33], PUPSNR [3], PUSSIM [3], and FPS. All the HDR results are tone-mapped using Photomatix. These measurement results collectively indicate the efficiency of our proposed method.

**HDR metrics.** We employ two Authoritative HDR image measurement methods: High Dynamic Range-Visual Difference Predictor (HDR-VDP) [24, 25, 26, 33] and Perceptually Uniform (PU21) [3]. HDR-VDP simulates the human eye’s perception of high dynamic range images under various lighting conditions, offering a means to quantify perceptual differences in images. We have opted for

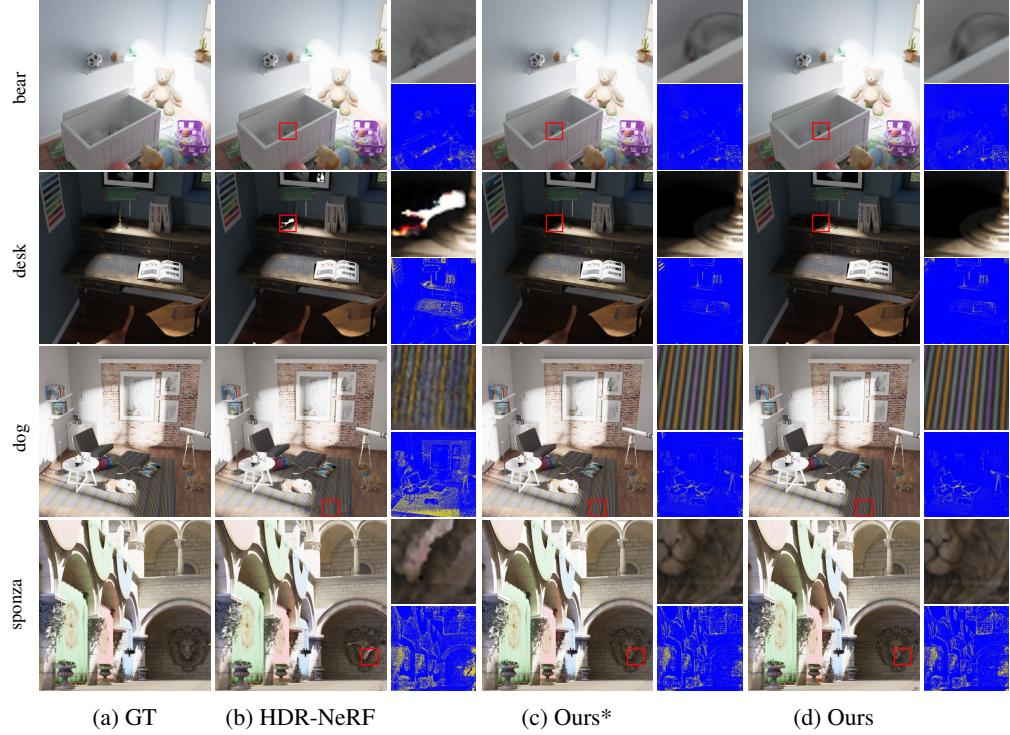


Figure 3: Comparison of LDR Image Quality for Novel View Renderings. The image in the top right of each rendered image is a zoomed-in section selected by the red rectangle. The mse error heatmap for each rendered image is shown in the bottom right. "Ours\*" represents our method without  $\mathcal{L}_u$

version 3.0.7 of the HDR-VDP measurement tool, employing the Side by Side assessment approach and the Q-score measurement metric. In this metric, a score of 10 indicates the highest quality, which diminishes for lower quality. PU21 [3] propose introducing a PU21 encoder to map the HDR pixel value range to the LDR pixel value range, thereby enabling the utilization of conventional LDR measurement techniques (such as PSNR, SSIM, LPIPS) for assessment. During PU21 testing, the maximum luminance of the HDR display is set to 1000 nits.

**Quality comparison.** The comparison results of our method with other methods on synthetic and real datasets are presented in Table 1. It is evident that our method exhibits even faster training speeds compared to 3DGS [18]. This is attributed to the tendency of the 3DGS method to overfit to the LDR images from each viewpoint, leading to an excessive number of points and consequently slowing down the training speed. Moreover, we adopt a pruning strategy [34] to reduce the number of points, thereby enhancing the training speed. Furthermore, it can be observed that our method exhibits significantly better rendering quality compared to the second best—HDRNeRF, especially in LDR-NE (using only two exposure times as the training set). Unlike HDRNeRF, our method achieves high rendering quality in both LDR-NE and LDR-OE (using three exposure times as the training set), demonstrating greater stability. More importantly, our method can achieve real-time rendering, and the training time only takes several minutes.

In Fig. 3, we compare our method with HDRNeRF in rendering images under novel viewpoints and exposures. Zoom-in insets reveal that our approach recovers details more accurately than HDRNeRF [16]. Moreover, in the error map, it is evident that our method exhibits fewer errors overall. The novel HDR views of real and synthetic scenes are respectively presented in Fig. 4 and Fig. 5. In Fig. 5(c) and (e), it can be observed that our method demonstrates smaller errors in HDR-VDP measurements, suggesting closer proximity to the ground truth. Finally, in the supplementary material,

Table 2: Some ablation studies of our method

$\mathcal{L}_u$	Coarse	Time scaling	Asymmetric	PSNR ↑	SSIM ↑	LPIPS ↓
✓	✓	✓	✓	39.00	0.975	0.012
✗	✓	✓	✓	38.76	0.976	0.012
✓	✗	✓	✓	19.32	0.618	0.308
✓	✓	✗	✓	26.00	0.785	0.180
✓	✓	✓	✗	38.60	0.975	0.013

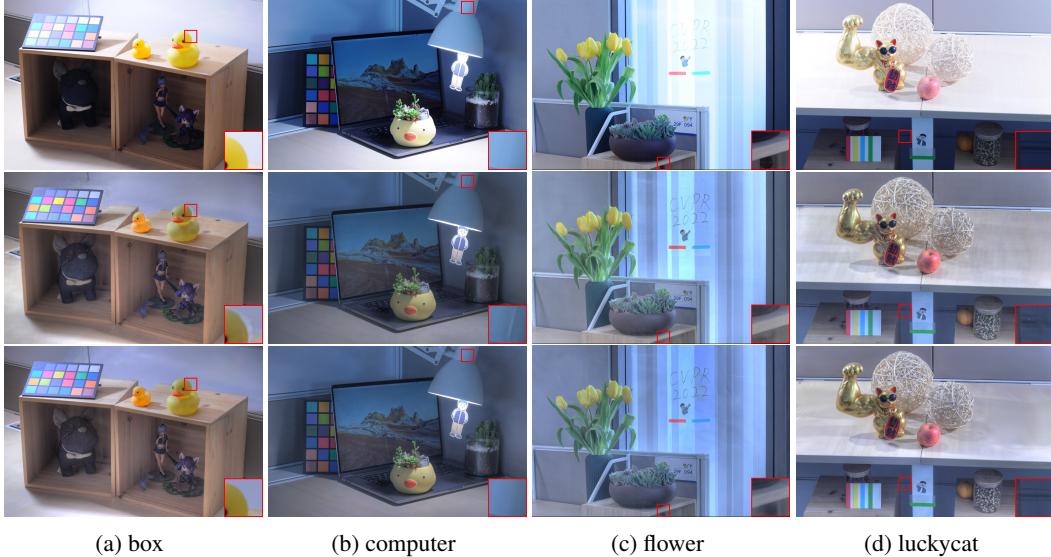


Figure 4: Comparison of HDR image quality for novel viewpoints on real scenes. The first row is GT LDR images, the second row HDR images are rendered by HDRNeRF[16], and the third row is ours.

we will discuss how our method exhibits greater stability compared to HDRNeRF, as the implicit representation of MLPs may fail to accurately decouple nonlinear components.

**Ablation studies.** In Table. 2, we conducted ablation experiments on several key modules of our method (with or without unit loss, with or without coarse stage, with or without time scaling, and symmetric or asymmetric grid). The left side of the table presents the methods under investigation, while the right side displays the average test results for LDR-OE and LDR-NE. The second row illustrates that, unlike HDRNeRF [16], our method can still learn a high quality HDR radiance field and produce high-quality images even without utilizing  $\mathcal{L}_u$  to constrain the value at zero (with the goal of learning a specific tone mapper). The third row highlights that our designed coarse-to-fine strategy significantly enhances the learning of tone mapper based on discrete grids. Without coarse-to-fine, the model is susceptible to falling into local optima or requiring extended convergence times. In the fourth row, we conduct ablation experiments on our proposed time scaling. Due to the discrete exposure range of the dataset, without time scaling, the model tends to overfit to each exposure time, resulting in the generation of low-quality LDR images at novel exposure times. Lastly, we verify the effectiveness of our suggested asymmetric grid, particularly in scenarios with high radiance value ranges and irregular density distributions. The results in Table 2 demonstrate the success of our approach in enhancing quality.

## 5 Conclusion

To recover 3D HDR radiance fields from 2D multi-exposure LDR images, we propose HDRGS, a real-time rendering-supported method with highly interpretable HDR radiance field reconstruction. We redefine the color of Gaussian points as radiance, implying that each Gaussian point radiates radiance rather than color. Then we construct an uneven asymmetric grid as our tone mapper, mapping irradiance received by each pixel to LDR color values. Additionally, to mitigate the grid’s discrete nature and the tendency to fall into local optima, we introduce a novel coarse-to-fine strategy, effectively accelerating model convergence and enhancing reconstruction quality, making it more robust for various complex exposure scenes. Each of our module designs can be mathematically explained. Experimental results on real and synthetic datasets demonstrate that our method achieves state-of-the-art efficiency and quality in reconstructing both LDR images from arbitrary viewpoints and HDR images from arbitrary viewpoints and exposures. To our knowledge, our method marks the pioneering exploration into the potential of Gaussian splatting for 3D HDR reconstruction. Our model and data code will be fully provided for future research endeavors.

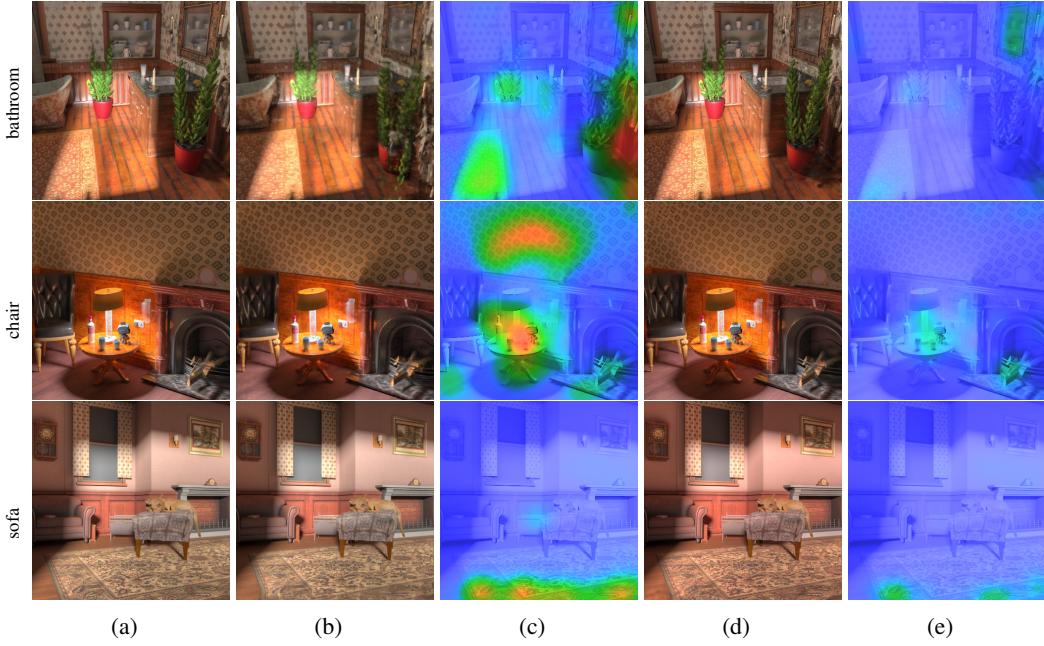


Figure 5: Comparison of HDR image quality for novel viewpoints on syn scenes. (a) represents the GT HDR images, (b) depicts the HDR images rendered by HDRNeRF[16] , while the right side (c) shows their error maps drawn by HDR-VDP. (d) presents the HDR images rendered by our method, and (e) displays the error maps of ours.

**Limitation:** Our method crudely simulates the physical imaging process, simplifying some details. For instance, we have not taken into account the influence of aperture size and ISO gain. Due to the limitations of the 3DGS[18] method, our method currently exhibits subpar reconstruction results for scenes containing transparent objects.

## References

- [1] Daniel Wood Daniel Azuma Wyvern Aldinger, B Curless T Duchamp DH Salesin, and W Stuetzle. “Surface light fields for 3D photography”. In: *Computer Graphics, SIGGRAPH 2000 Proc.* 2000, pp. 287–296.
- [2] Nantheera Anantrasirichai and David Bull. “Artificial intelligence in the creative industries: a review”. In: *Artificial intelligence review* 55.1 (2022), pp. 589–656.
- [3] Maryam Azimi et al. “PU21: A novel perceptually uniform encoding for adapting existing quality metrics for HDR”. In: *2021 Picture Coding Symposium (PCS)*. IEEE. 2021, pp. 1–5.
- [4] Jonathan T Barron et al. “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5855–5864.
- [5] Chris Buehler et al. “Unstructured lumigraph rendering”. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001, pp. 425–432.
- [6] Ang Cao and Justin Johnson. “Hexplane: A fast representation for dynamic scenes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 130–141.
- [7] Anpei Chen et al. “Tensorf: Tensorial radiance fields”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 333–350.
- [8] Paul E Debevec and Jitendra Malik. “Recovering high dynamic range radiance maps from photographs”. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 643–652.
- [9] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. “Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach”. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 465–474.
- [10] Frederic Dufaux et al. *High dynamic range video: from acquisition, to display and applications*. Academic Press, 2016.
- [11] Gabriel Eilertsen et al. “HDR image reconstruction from a single exposure using deep CNNs”. In: *ACM transactions on graphics (TOG)* 36.6 (2017), pp. 1–15.
- [12] Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. “Deep reverse tone mapping.” In: *ACM Trans. Graph.* 36.6 (2017), pp. 177–1.
- [13] Sara Fridovich-Keil et al. “K-planes: Explicit radiance fields in space, time, and appearance”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 12479–12488.
- [14] Sara Fridovich-Keil et al. “Plenoxels: Radiance fields without neural networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5501–5510.
- [15] Binbin Huang et al. “2d gaussian splatting for geometrically accurate radiance fields”. In: *arXiv preprint arXiv:2403.17888* (2024).
- [16] Xin Huang et al. “Hdr-nerf: High dynamic range neural radiance fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18398–18408.
- [17] Kim Jun-Seong et al. “Hdr-plenoxels: Self-calibrating high dynamic range radiance fields”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 384–401.
- [18] Bernhard Kerbl et al. “3d gaussian splatting for real-time radiance field rendering”. In: *ACM Transactions on Graphics* 42.4 (2023), pp. 1–14.
- [19] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [20] Kiriakos N Kutulakos and Steven M Seitz. “A theory of shape by space carving”. In: *International journal of computer vision* 38 (2000), pp. 199–218.
- [21] Siyeong Lee, Gwon Hwan An, and Suk-Ju Kang. “Deep chain hdri: Reconstructing a high dynamic range image from a single low dynamic range image”. In: *IEEE Access* 6 (2018), pp. 49913–49924.
- [22] Marc Levoy and Pat Hanrahan. “Light field rendering”. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 441–452.
- [23] Tao Lu et al. “Scaffold-gs: Structured 3d gaussians for view-adaptive rendering”. In: *arXiv preprint arXiv:2312.00109* (2023).

- [24] Rafal Mantiuk, Karol Myszkowski, and H-P Seidel. “Visible difference predictor for high dynamic range images”. In: *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*. Vol. 3. IEEE. 2004, pp. 2763–2769.
- [25] Rafał Mantiuk et al. “HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions”. In: *ACM Transactions on graphics (TOG)* 30.4 (2011), pp. 1–14.
- [26] Rafal K Mantiuk, Dounia Hammou, and Param Hanji. “HDR-VDP-3: A multi-metric for predicting image differences, quality and contrast distortions in high dynamic range and regular content”. In: *arXiv preprint arXiv:2304.13625* (2023).
- [27] Demetris Marnerides et al. “Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content”. In: *Computer Graphics Forum*. Vol. 37. 2. Wiley Online Library. 2018, pp. 37–49.
- [28] Ricardo Martin-Brualla et al. “Nerf in the wild: Neural radiance fields for unconstrained photo collections”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7210–7219.
- [29] Tom Mertens, Jan Kautz, and Frank Van Reeth. “Exposure fusion: A simple and practical alternative to high dynamic range photography”. In: *Computer graphics forum*. Vol. 28. 1. Wiley Online Library. 2009, pp. 161–171.
- [30] Ben Mildenhall et al. “Local light field fusion: Practical view synthesis with prescriptive sampling guidelines”. In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–14.
- [31] Ben Mildenhall et al. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.
- [32] Thomas Müller et al. “Instant neural graphics primitives with a multiresolution hash encoding”. In: *ACM transactions on graphics (TOG)* 41.4 (2022), pp. 1–15.
- [33] Manish Narwaria et al. “HDR-VDP-2.2: a calibrated method for objective quality prediction of high-dynamic range and standard images”. In: *Journal of Electronic Imaging* 24.1 (2015), pp. 010501–010501.
- [34] Michael Niemeyer et al. “RadSplat: Radiance Field-Informed Gaussian Splatting for Robust Real-Time Rendering with 900+ FPS”. In: *arXiv preprint arXiv:2403.13806* (2024).
- [35] Eric Penner and Li Zhang. “Soft 3d reconstruction for view synthesis”. In: *ACM Transactions on Graphics (TOG)* 36.6 (2017), pp. 1–11.
- [36] Albert Pumarola et al. “D-nerf: Neural radiance fields for dynamic scenes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10318–10327.
- [37] Erik Reinhard. “High dynamic range imaging”. In: *Computer Vision: A Reference Guide*. Springer, 2020, pp. 1–6.
- [38] Allan G Rempel et al. “Ldr2hdr: on-the-fly reverse tone mapping of legacy video and photographs”. In: *ACM transactions on graphics (TOG)* 26.3 (2007), 39–es.
- [39] Gernot Riegler and Vladlen Koltun. “Free view synthesis”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*. Springer. 2020, pp. 623–640.
- [40] Steven M Seitz and Charles R Dyer. “Photorealistic scene reconstruction by voxel coloring”. In: *International journal of computer vision* 35 (1999), pp. 151–173.
- [41] Pratul P Srinivasan et al. “Pushing the boundaries of view extrapolation with multiplane images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 175–184.
- [42] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.
- [43] Justus Thies, Michael Zollhöfer, and Matthias Nießner. “Deferred neural rendering: Image synthesis using neural textures”. In: *Acm Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–12.
- [44] Dor Verbin et al. “Ref-nerf: Structured view-dependent appearance for neural radiance fields”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2022, pp. 5481–5490.

- [45] Michael Waechter, Nils Moehrle, and Michael Goesele. “Let there be color! Large-scale texturing of 3D reconstructions”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 836–850.
- [46] Chao Wang et al. “GlowGAN: Unsupervised Learning of HDR Images from LDR Images in the Wild”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 10509–10519.
- [47] Chao Wang et al. “Learning a self-supervised tone mapping operator via feature contrast masking loss”. In: *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library. 2022, pp. 71–84.
- [48] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. “Multiscale structural similarity for image quality assessment”. In: *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. Ieee. 2003, pp. 1398–1402.
- [49] Guanjun Wu et al. “Fast High Dynamic Range Radiance Fields for Dynamic Scenes”. In: *arXiv preprint arXiv:2401.06052* (2024).
- [50] Yao Yao et al. “Mvsnet: Depth inference for unstructured multi-view stereo”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 767–783.
- [51] Alex Yu et al. “Plenoctrees for real-time rendering of neural radiance fields”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5752–5761.
- [52] Zehao Yu et al. “Mip-splatting: Alias-free 3d gaussian splatting”. In: *arXiv preprint arXiv:2311.16493* (2023).
- [53] Richard Zhang et al. “The unreasonable effectiveness of deep features as a perceptual metric”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 586–595.
- [54] Tinghui Zhou et al. “Stereo magnification: Learning view synthesis using multiplane images”. In: *arXiv preprint arXiv:1805.09817* (2018).

## A Appendix / supplemental material

We will provide more detailed information about our experimental procedures in the supplemental material. In Section A.1, we will explore additional implementation details. Section A.2 will outline the derivation process of  $f_H$ . Section A.3 will discuss some of the inherent issues in the HDRNeRF model. Finally, Section A.4 will present additional experimental results.

### A.1 Additional Implementation Details

**Pruning Strategy.** We utilize the Ray Contribution-Based Pruning technique described in [34] as our pruning technique, giving each Gaussian point a score:

$$h(\mathbf{p}_i) = \max_{I_f \in \mathcal{I}_f, r \in I_f} \alpha_i^r \tau_i^r \quad (20)$$

where  $\alpha_i^r \tau_i^r$  represents the contribution of Gaussian point  $\mathbf{p}_i$  to pixel  $r$ . A Gaussian point  $\mathbf{p}_i$  can be retained as long as its contribution to any pixel point during a pruning interval exceeds the threshold; otherwise, it must be pruned. This strategy allows for the removal of points that make relatively low contributions to rendering the image. In this case, the threshold is set at 0.02, and pruning begins after 500 rounds, with intervals of 200 rounds each.

It is worth noting that this pruning technique must be staggered with transparency reset, as it will cause all Gaussian scores  $h(\mathbf{p}_i)$  to drop below the cutoff value.

### A.2 Proof

The functional relationship  $f_H(\cdot)$  between the learnable parameters  $E'$  and the HDR image pixel values  $E$  will be determined here. If no time scaling is performed, then  $E$  satisfies the following equation:

$$C_1 = g_1(\ln E + \ln t) \quad (21)$$

If time scaling is performed, then  $E'$  satisfies the following equation:

$$C_2 = g_2(r \cdot \ln t + s + E') \quad (22)$$

Where  $g_1$  and  $g_2$  represent different CRF functions, both of which are monotonic and differentiable. Since the same LDR image is used for supervision regardless of whether time scaling is performed or not,  $C_1 = C_2$ . Therefore,  $E'$  and  $E$  satisfy:

$$g_1(\ln t + \ln E) = g_2(r \cdot \ln t + f(\ln E)) \quad \text{where } f(\ln E) = E' + s \quad (23)$$

Fixing  $E$ , differentiate both sides w.r.t.  $\ln t$ , we obtain:

$$g'_1 = g'_2 \cdot r \quad (24)$$

Fixing  $t$ , differentiate both sides w.r.t.  $\ln E$ , we obtain:

$$g'_1 = g'_2 \cdot f' \quad (25)$$

From the eq 24 25, we can infer that  $f' = r$ , which implies:

$$\frac{d(E' + s)}{d(\ln E)} = r \quad (26)$$

If the unit exposure loss is adopted, then there exists only a scaling relationship between  $g_1$  and  $g_2$ , without any offset. Therefore, we can easily derive the final equation:

$$E = e^{\frac{(E' + s)}{r}} \quad (27)$$

From Eq.27, we can render HDR images after the model training is completed.

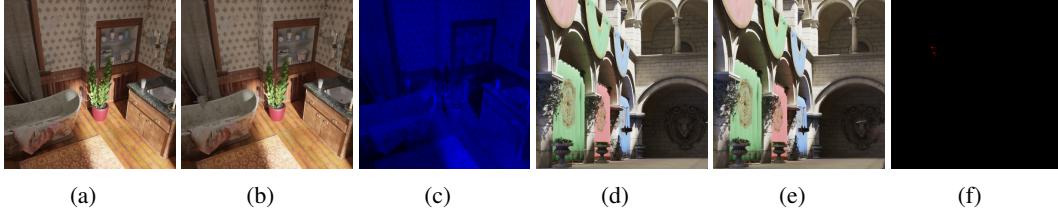


Figure 6: **Failure cases.** (a) and (d) are the GT images, (b) and (e) are LDR images rendered by HDRNeRF[16], while (c) and (f) are HDR images.

### A.3 Discussion

During our experiments, we observed instances where the tone mapper of HDRNeRF [16] occasionally overfits to LDR, leading to an inaccurate HDR radiance field. As depicted in Fig. 6, it's evident that directly utilizing MLP in HDRNeRF to model the tone mapper is unstable. Employing three separate MLPs in HDRNeRF [16] to handle the RGB channels independently might induce crosstalk problems, where interference between channels could potentially lead to being trapped in local optimal solutions. While HDRNeRF [16] learns the correct LDR, errors arise in HDR radiance field construction. Furthermore, in [17, 49], the authors mention that constructing a tone mapping module based on MLPs may fail to correctly decouple nonlinear components.

### A.4 Additional Results

In this section, we present more experimental results. Table. 5 shows the HDR measurement results compared to HDRNeRF on the synthetic dataset. Tab3 and Tab 4 demonstrate the LDR measurement results of our method compared to other baseline methods across various scenes. In Fig7, we present LDR images of our various ablation methods (with/without coarse stage, with/without unit exposure loss, with/without t scaling , asymmetric or symmetric grid ) and other methods under the same viewpoint but different exposure time. In the ablation experiments of the coarse stage, we can clearly see that without the coarse stage, the asymmetric grid would easily overfit to the exposure time used in the training dataset. In Fig10, we show the distribution of irradiance  $E'$  for each scene in the first frame of the test dataset. It can be observed that the radiance distribution of most scenes is highly uneven, especially scenes like "diningroom", which easily lead the grid into local optimization problems. In Fig.9, we present additional results, where (e) and (f) respectively show HDR images rendered by HDR-NeRF and our method. The upper right triangles of these images are the rendered HDR images, while the lower left triangles are error maps drawn by HDR-VDP.



Figure 7: Comparison of LDR image quality under exposure time  $\{t_1, t_2, t_3, t_4, t_5\}$  for novel viewpoints on "sofa". The exposure time of training set is randomly selected from  $\{t_1, t_3, t_5\}$ .

Table 3: LDR image quality assessment, LDR-OE represents the use of LDR images with exposures  $\{t_1, t_3, t_5\}$  as the training set, while LDR-NE represents the use of LDR images with exposures  $\{t_2, t_4\}$  as the training set. The exposure times of the test dataset are  $\{t_1, t_2, t_3, t_4, t_5\}$ . The **best** and the **second best** results are denoted by red and blue.

		Bathroom			Bear			Chair			Diningroom		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LDR-OE	NeRF *	14.59	0.429	0.424	11.97	0.560	0.515	12.23	0.422	0.492	12.50	0.378	0.600
	HDR-P +	6.73	0.348	0.669	10.77	0.637	0.553	10.29	0.404	0.861	11.34	0.439	0.670
	NeRF-W *	29.64	0.900	<b>0.055</b>	32.24	0.978	<b>0.021</b>	28.01	0.840	0.161	32.25	<b>0.979</b>	<b>0.016</b>
	HDR-NeRF	<b>32.81</b>	<b>0.887</b>	0.121	<b>42.50</b>	<b>0.989</b>	0.097	<b>32.33</b>	<b>0.901</b>	<b>0.095</b>	<b>41.34</b>	<b>0.984</b>	<b>0.012</b>
	3DGS	11.78	0.353	0.555	9.72	0.473	0.551	10.94	0.380	0.550	10.58	0.316	0.600
	Ours	<b>39.64</b>	<b>0.970</b>	<b>0.013</b>	<b>43.16</b>	<b>0.992</b>	<b>0.003</b>	<b>32.84</b>	<b>0.922</b>	<b>0.034</b>	<b>37.66</b>	0.972	0.023
LDR-NE	NeRF *	-	-	-	-	-	-	-	-	-	-	-	-
	HDR-P +	6.64	0.349	0.662	10.70	0.636	0.551	10.27	0.403	0.862	11.35	0.439	0.669
	NeRF-W *	26.98	0.881	<b>0.066</b>	32.67	0.976	<b>0.022</b>	26.96	0.815	0.157	32.53	0.972	0.019
	HDR-NeRF	<b>32.49</b>	<b>0.942</b>	0.076	<b>38.48</b>	<b>0.987</b>	0.031	<b>32.50</b>	<b>0.909</b>	<b>0.091</b>	<b>41.71</b>	<b>0.988</b>	<b>0.009</b>
	3DGS	12.94	0.464	0.420	10.83	0.521	0.500	12.05	0.422	0.484	11.05	0.322	0.594
	Ours	<b>39.61</b>	<b>0.973</b>	<b>0.014</b>	<b>42.56</b>	<b>0.991</b>	<b>0.005</b>	<b>32.49</b>	<b>0.923</b>	<b>0.037</b>	<b>38.01</b>	<b>0.980</b>	<b>0.013</b>
LDR-OE		Dog			Desk			Sofa			Sponza		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
	NeRF *	13.69	0.619	0.279	15.29	0.645	0.249	15.06	0.718	0.229	16.39	0.664	0.219
	HDR-P +	10.48	0.395	0.737	8.69	0.455	0.660	10.34	0.444	0.783	10.33	0.358	0.778
	NeRF-W *	31.01	<b>0.967</b>	<b>0.022</b>	30.21	0.958	0.030	30.76	0.955	0.029	24.50	0.908	0.037
	HDR-NeRF	<b>34.46</b>	0.948	0.040	<b>37.62</b>	<b>0.965</b>	<b>0.026</b>	<b>37.88</b>	<b>0.973</b>	<b>0.017</b>	<b>34.85</b>	<b>0.959</b>	<b>0.030</b>
LDR-NE	3DGS	8.51	0.193	0.651	11.54	0.429	0.484	10.37	0.370	0.562	16.46	0.645	0.317
	Ours	<b>40.80</b>	<b>0.990</b>	<b>0.005</b>	<b>40.43</b>	<b>0.981</b>	<b>0.007</b>	<b>40.32</b>	<b>0.986</b>	<b>0.007</b>	<b>38.42</b>	<b>0.980</b>	<b>0.008</b>
LDR-NE	NeRF *	-	-	-	-	-	-	-	-	-	-	-	-
	HDR-P +	10.48	0.395	0.737	8.62	0.455	0.660	10.32	0.444	0.789	10.33	0.358	0.778
	NeRF-W *	30.41	<b>0.964</b>	<b>0.026</b>	29.60	<b>0.950</b>	<b>0.034</b>	30.31	0.952	<b>0.031</b>	24.32	0.904	<b>0.042</b>
	HDR-NeRF	<b>31.66</b>	0.939	0.088	<b>31.86</b>	0.937	0.092	<b>36.05</b>	<b>0.962</b>	0.036	<b>30.72</b>	<b>0.924</b>	0.092
	3DGS	9.09	0.197	0.648	11.23	0.414	0.502	11.89	0.462	0.465	16.71	0.649	0.298
	Ours	<b>40.71</b>	<b>0.990</b>	<b>0.005</b>	<b>39.83</b>	<b>0.979</b>	<b>0.008</b>	<b>39.07</b>	<b>0.983</b>	<b>0.009</b>	<b>38.47</b>	<b>0.979</b>	<b>0.009</b>

+ HDR-Plenoxel[17]. We directly utilize the grid parameters provided by them for training, which may not be suitable for HDRNeRF datasets.

\* The results are derived from HDRNeRF[16].

Table 4: LDR image quality assessment, LDR-OE represents the use of LDR images with exposures  $\{t_1, t_3, t_5\}$  as the training set, while LDR-NE represents the use of LDR images with exposures  $\{t_2, t_4\}$  as the training set. The exposure times of the test dataset are  $\{t_1, t_2, t_3, t_4, t_5\}$ . The **best** and the **second best** results are denoted by red and blue.

		box			computer			flower			luckycat		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LDR-OE	NeRF *	17.06	0.770	0.233	14.68	0.697	0.281	14.60	0.504	0.524	13.67	0.706	0.262
	HDR-P +	9.47	0.587	0.693	8.91	0.469	0.697	9.30	0.599	0.711	9.39	0.610	0.696
	NeRF-W *	29.21	0.927	0.097	28.91	0.919	0.112	26.23	0.933	0.094	30.00	0.927	0.076
	HDR-NeRF	<b>31.48</b>	<b>0.957</b>	<b>0.062</b>	<b>32.55</b>	<b>0.945</b>	<b>0.079</b>	<b>31.01</b>	<b>0.952</b>	<b>0.061</b>	<b>32.47</b>	<b>0.945</b>	<b>0.063</b>
	3DGS	12.25	0.628	0.360	11.15	0.511	0.367	12.15	0.651	0.342	11.91	0.654	0.352
	Ours	<b>33.65</b>	<b>0.972</b>	<b>0.021</b>	<b>32.79</b>	<b>0.961</b>	<b>0.026</b>	<b>31.28</b>	<b>0.964</b>	<b>0.028</b>	<b>35.64</b>	<b>0.972</b>	<b>0.017</b>
LDR-NE	NeRF *	-	-	-	-	-	-	-	-	-	-	-	-
	HDR-P +	9.41	0.588	0.694	8.88	0.471	0.700	9.25	0.600	0.710	9.33	0.613	0.696
	NeRF-W *	29.59	0.923	0.104	27.54	0.892	0.136	26.84	0.939	0.078	<b>30.78</b>	<b>0.940</b>	<b>0.058</b>
	HDR-NeRF	<b>30.21</b>	<b>0.955</b>	<b>0.073</b>	<b>32.48</b>	<b>0.945</b>	<b>0.083</b>	<b>30.15</b>	<b>0.948</b>	<b>0.063</b>	28.90	0.936	0.094
	3DGS	12.76	0.694	0.342	13.40	0.594	0.353	12.93	0.717	0.332	12.85	0.724	0.330
	Ours	<b>32.94</b>	<b>0.971</b>	<b>0.020</b>	<b>32.30</b>	<b>0.932</b>	<b>0.056</b>	<b>30.79</b>	<b>0.964</b>	<b>0.022</b>	<b>35.05</b>	<b>0.972</b>	<b>0.019</b>

+ HDR-Plenoxel[17]. We directly utilize the grid parameters provided by them for training, which may not be suitable for HDRNeRF datasets.

\* The results are derived from HDRNeRF[16].

Table 5: HDR image quality assessment, LDR-OE represents the use of LDR images with exposures  $\{t_1, t_3, t_5\}$  as the training set, while LDR-NE represents the use of LDR images with exposures  $\{t_2, t_4\}$  as the training set. The best results are denoted by red.

		Bathroom			Bear			Chair		
		HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑
LDR-OE	HDR-NeRF	9.50	22.42	0.88	9.94	<b>37.65</b>	<b>0.99</b>	9.11	17.91	0.40
	Ours	<b>9.85</b>	<b>26.92</b>	<b>0.92</b>	<b>9.95</b>	28.78	0.97	<b>9.40</b>	<b>18.37</b>	<b>0.49</b>
LDR-NE	HDR-NeRF	1.85	7.57	0.02	-0.22	5.64	0.01	5.05	14.63	0.21
	Ours	<b>9.91</b>	<b>27.92</b>	<b>0.94</b>	<b>9.94</b>	<b>19.18</b>	<b>0.83</b>	<b>9.37</b>	<b>15.48</b>	<b>0.22</b>
LDR-OE	Diningroom	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑
	HDR-NeRF	<b>9.94</b>	<b>35.64</b>	<b>1.00</b>	9.57	18.64	0.94	9.65	<b>33.60</b>	<b>0.92</b>
LDR-NE	Ours	9.83	25.13	0.93	<b>9.87</b>	<b>26.62</b>	<b>0.97</b>	<b>9.73</b>	31.77	0.87
	HDR-NeRF	<b>9.87</b>	<b>28.60</b>	0.86	9.46	11.69	-0.12	-0.82	19.94	0.36
LDR-OE	Ours	9.25	24.27	<b>0.96</b>	<b>9.79</b>	<b>25.14</b>	<b>0.94</b>	<b>9.63</b>	<b>32.14</b>	<b>0.91</b>
LDR-NE	Dog	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑
	HDR-NeRF	<b>9.75</b>	<b>31.24</b>	<b>0.99</b>	2.14	6.11	0.00	8.70	<b>25.40</b>	0.76
	Ours	9.52	25.37	0.88	<b>9.89</b>	<b>17.37</b>	<b>0.83</b>	<b>9.75</b>	25.04	<b>0.86</b>
LDR-OE	sofa	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑	HDR VDP-Q ↑	PUPSNR ↑	PUSSIM ↑
	HDR-NeRF	<b>9.70</b>	<b>29.24</b>	<b>0.99</b>	0.42	4.82	0.00	4.41	15.27	0.29
LDR-NE	Ours	9.62	17.63	0.87	<b>9.70</b>	<b>16.65</b>	<b>0.82</b>	<b>9.65</b>	<b>22.30</b>	<b>0.81</b>

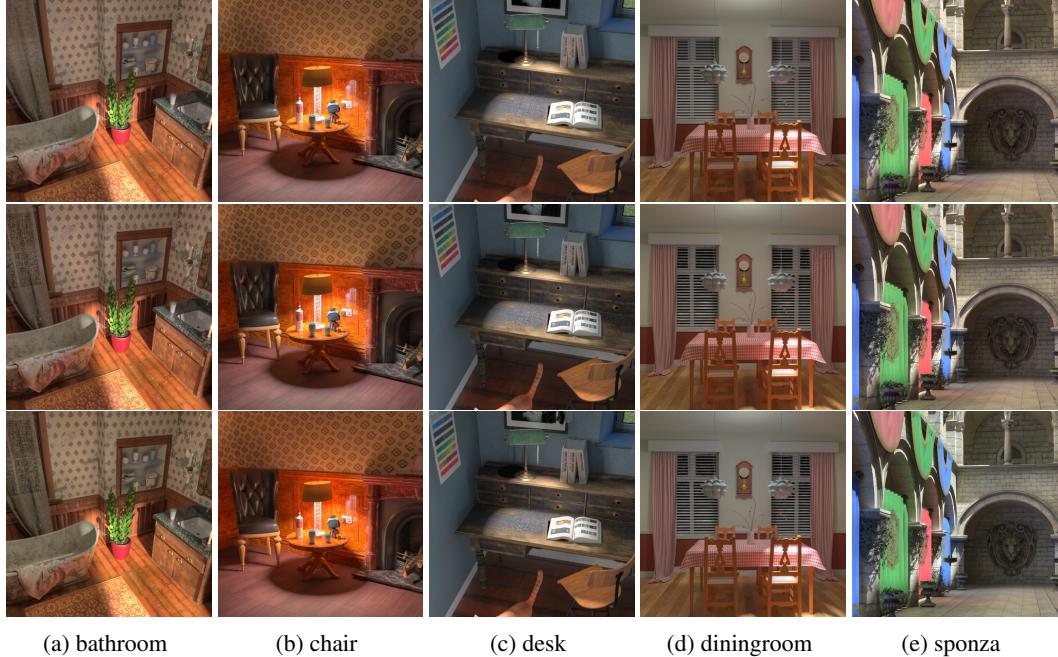


Figure 8: HDR image quality comparison: the first row shows the ground truth (GT). The second and third rows respectively show HDR images rendered by our method with  $\mathcal{L}_u$  and without  $\mathcal{L}_u$ .

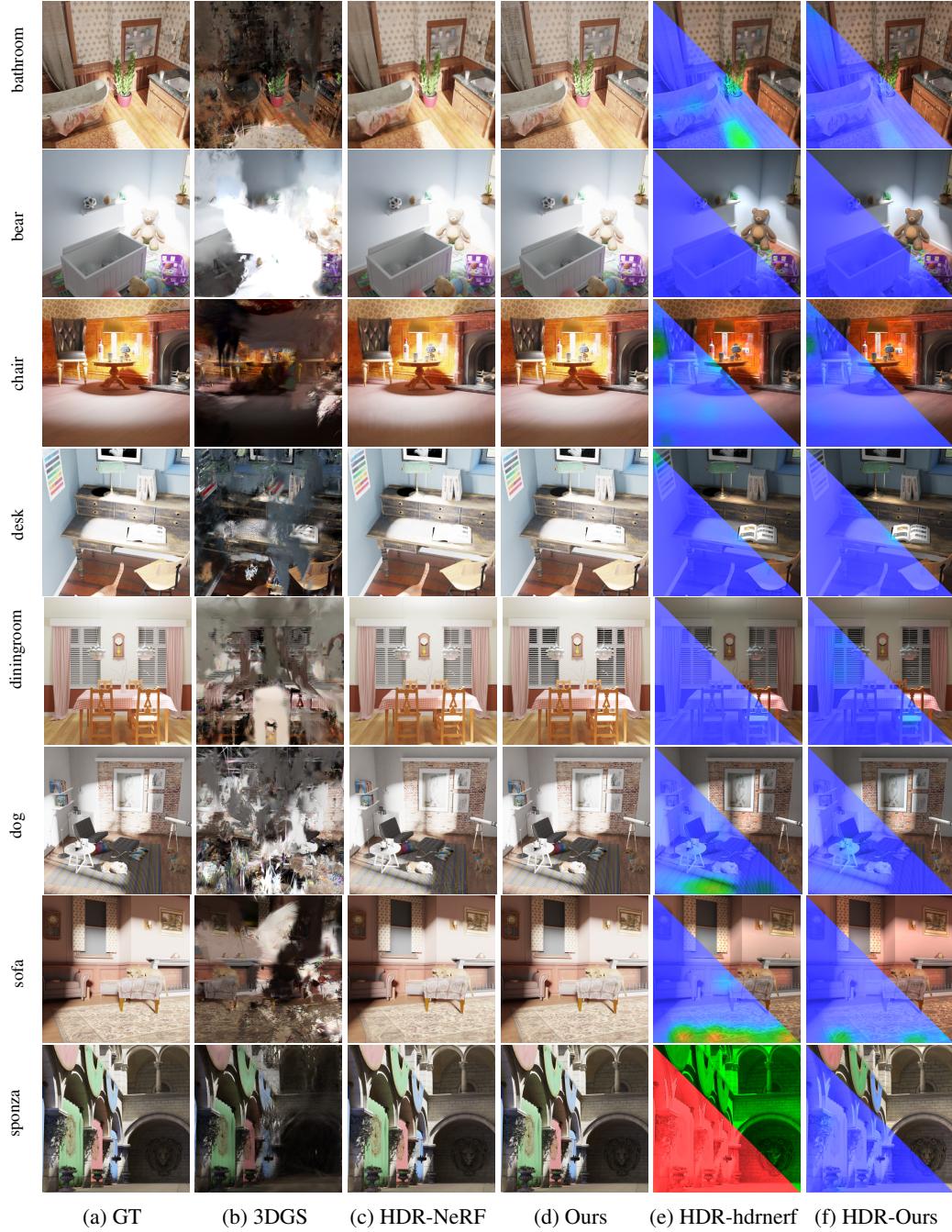


Figure 9: Results of the synthesis datasets. The first four columns represent LDR images, while the last two columns depict HDR images and the error maps generated by HDR-VDP.

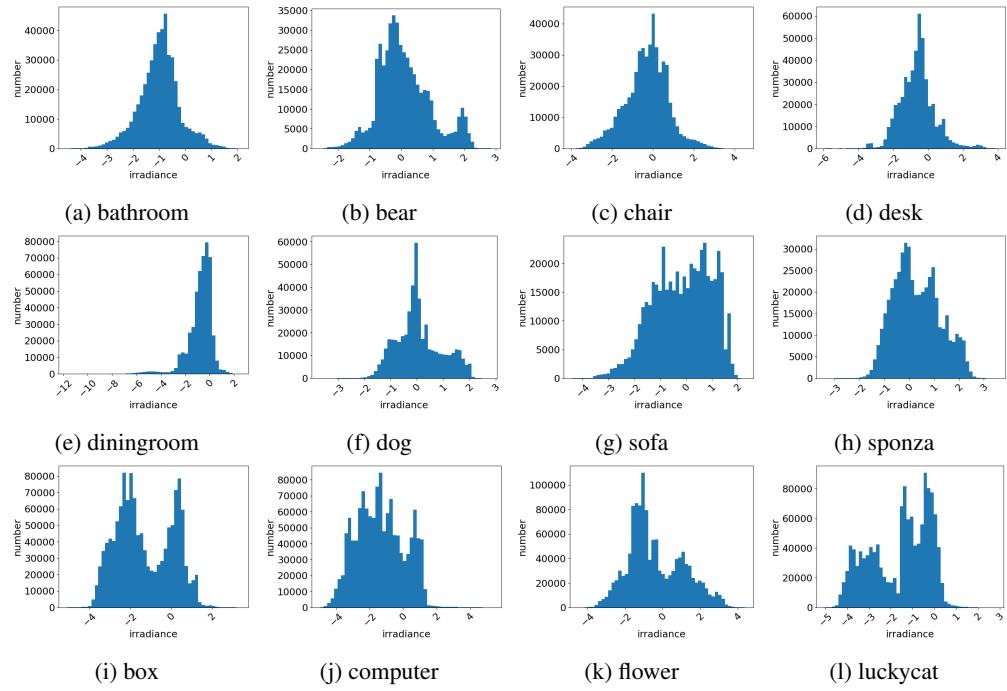


Figure 10: The distribution of irradiance.