From Chaos to Clarity: 3DGS in the Dark

Zhihao Li* **Yufei Wang*** **Alex Kot** **Bihan Wen†**
Department of EEE, Nanyang Technology University, Singapore
https://lizhihao6.github.io/Raw3DGS

Abstract

Novel view synthesis from raw images provides superior high dynamic range (HDR) information compared to reconstructions from low dynamic range RGB images. However, the inherent noise in unprocessed raw images compromises the accuracy of 3D scene representation. Our study reveals that 3D Gaussian Splatting (3DGS) is particularly susceptible to this noise, leading to numerous elongated Gaussian shapes that overfit the noise, thereby significantly degrading reconstruction quality and reducing inference speed, especially in scenarios with limited views. To address these issues, we introduce a novel self-supervised learning framework designed to reconstruct HDR 3DGS from a limited number of noisy raw images. This framework enhances 3DGS by integrating a noise extractor and employing a noise-robust reconstruction loss that leverages a noise distribution prior. Experimental results show that our method outperforms LDR/HDR 3DGS and previous state-of-the-art (SOTA) self-supervised and supervised pre-trained models in both reconstruction quality and inference speed on the RawNeRF dataset across a broad range of training views. Code can be found in `https://lizhihao6.github.io/Raw3DGS`.

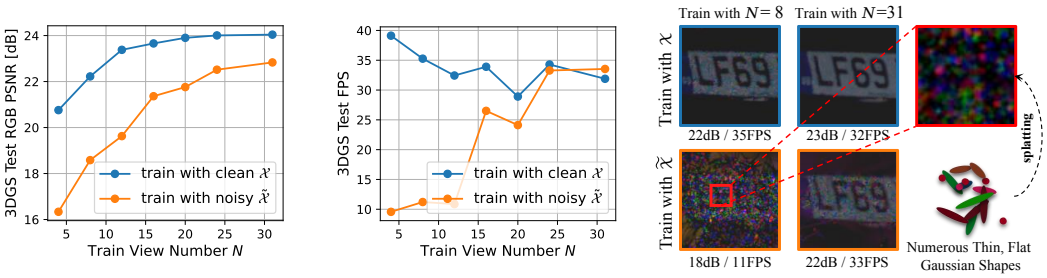
1 Introduction

Novel view synthesis (NVS) is fundamental to 3D vision, with extensive applications in virtual and augmented reality (VR/AR) [16; 17], autonomous driving [18; 34], and 3D asset creation [22; 4; 20]. Neural Radiance Fields (NeRF) [24] have revolutionized this field by rendering colors through the accumulation of RGB values along sampling rays, employing an implicit MultiLayer Perceptron (MLP) representation. Typically, this method uses low dynamic range (LDR) RGB images processed by image signal processing (ISP) modules, leading to a significant loss of crucial scene details, especially in high-contrast areas like highlights and shadows, which can degrade performance in high dynamic range (HDR) environments such as tunnels, sunsets, or dimly lit scenes. Moreover, reliance on ISP-processed RGB images restricts post-capture color and tone adjustments, presenting significant challenges for photographers and modelers during post-production.

In contrast, raw images before ISP offer a higher dynamic range and preserve more scene information. Recent research has indicated that utilizing raw images can significantly enhance the performance of downstream computer vision tasks in complex lighting conditions [19] and offer greater flexibility in post-production adjustments [37; 5]. Building on this advantage, RawNeRF [23] first employed raw images as the optimization target in NeRF, achieving marked improvements over traditional RGB-based LDR NeRF approaches. However, RawNeRF's reliance on implicit 3D representation is computationally demanding, requiring up to 48 hours to train a single scene and about one minute to render a single view, which limits its practicality for real-time applications.

*Both authors contributed equally to this research.
†Corresponding author. Email: `bihan.wen@ntu.edu.sg`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Impact of noise on PSNR (b) Impact of noise on FPS (c) Visualization of test views

Figure 1: Comparative analysis of 3DGS trained with clean raw images, denoted $\mathcal{X}$, versus noisy raw images, denoted $\tilde{\mathcal{X}}$, across various training view counts N. The clean raw images, captured in daylight, are selected from the RawNeRF dataset [23]. The noisy raw images are generated from these clean images using the noise model from PMN [6] with calibrated camera noise parameters. (a) Training with noisy raw images results in decreased PSNR in the test views, with a widening performance gap as the number of training views is reduced. (b) The rendering speed (FPS) shows a similar trend to PSNR. (c) Test view visualizations show that training with noisy images causes 3DGS to produce numerous thin, flat Gaussian shapes, leading to visual artifacts and reduced FPS, especially with fewer training views.

Very recently, the advance on 3D Gaussian Splatting (3DGS) [13] offers an explicit 3D representation that employs a set of learnable 3D Gaussian points to depict color, shape, and opacity, thereby enabling real-time novel view rendering. However, utilizing raw images directly as the optimization target in 3DGS can introduce significant artifacts and adversely affect rendering speed, especially in scenarios with limited training views—a common challenge in real-world applications, as depicted in Fig. 1. These challenges arise primarily from the inherent noise in raw images, which is more pronounced than in RGB images due to the absence of noise reduction and smoothing processes typically performed by ISPs. As demonstrated in Fig. 2a, the presence of noise in raw images is an inevitable consequence of physical phenomena such as the photoelectric effect and hardware limitations like dark current leakage. Unlike NeRF, where the Multilayer Perceptron (MLP) acts as a low-frequency filter [38] to mitigate high-frequency noise, 3DGS tends to produce numerous thin, elongated Gaussians to represent noise, as seen in Fig. 1c.

To address these challenges, we propose a novel self-supervised framework that jointly denoises and constructs HDR 3DGS representations within noisy raw images using a noise robust reconstruction loss. Specifically, our method utilizes a noise extractor to predict the underlying noise in raw images, constrained by a predefined noise model. Simultaneously, the 3DGS branch is optimized towards pseudo noise-free raw images. This dual-branch approach allows the noise extractor to accurately predict and separate noise from the raw images. Compared to using the standard RawNeRF loss function with conventional 3DGS, our approach significantly reduces noise artifacts in rendered views and enhances the rendering speed. Our contributions can be summarized as follows:

- We discover that noisy raw images significantly degrade the rendering quality and speed of 3DGS, especially in scenarios with limited training views. We provide a detailed analysis of how noise impacts the optimization of 3DGS, and we model its relationship with the number of training views and the noise distribution.
- We propose a novel self-supervised framework that incorporates a noise robust reconstruction loss. This framework leverages a physical-based noise model to jointly denoise and enhance the HDR representation of 3DGS within noisy raw images.
- Our method substantially outperforms conventional 3DGS methods that employ RawNeRF loss, along with both self-supervised and pre-trained supervised denoisers, in terms of rendering quality and processing speed across various training view counts.

2 Related work

2.1 Neural Radiance Fields and 3D Gaussian Splatting

Recently, Neural Radiance Fields (NeRF) [24] have spearheaded a significant advancement in novel-view synthesis by reconstructing 3D scenes. NeRF models achieve scene representation by optimizing coordinate-based multi-layer perceptrons (MLPs) to estimate color and density values through differentiable volume rendering. Typically, NeRF utilizes tone-mapped low dynamic range (LDR) images as inputs. These images are processed by image signal processing (ISP), which, while reducing noise and smoothing details, also maps high dynamic range (HDR) to LDR, often clipping highlights and shadows. To counter these limitations, RawNeRF [23] modified NeRF with a scaling loss to directly train on linear raw images, thus preserving the scene's full dynamic range. However, like most previous NeRF-based methods, RawNeRF is computationally demanding, requiring up to 48 hours to train a single scene and about one minute to render a single view, which constrains its usability in real-time applications. To accelerate rendering, various methods have been explored, including space discretization [7; 8; 10; 26; 29; 33; 35], codebooks [28], and hash tables [25]. Yet, these approaches still require a substantial number of queries to render a single pixel, rendering them unsuitable for real-time applications. More recently, 3D Gaussian Splatting (3DGS) [13] introduced a novel approach by explicitly representing scenes with learnable 3D Gaussians. Utilizing differentiable splatting and tile-based rasterization, 3DGS enables real-time novel view rendering. However, the initial 3DGS models did not achieve rendering quality comparable to NeRF. To address this, Scaffold-GS [13] introduces an anchor-level feature to capture correlations between adjacent Gaussian points, which significantly enhances the rendering quality of 3DGS, achieving comparable results to NeRF while maintaining real-time rendering speeds.

2.2 Raw image and video denoising methods

Raw image denoising is critical in both photography [9] and scientific imaging [15; 12]. Traditional ISPs typically employ self-similarity-based methods such as BM3D [3] to reduce noise. Recent advancements such as SID [2] and SIDD [1] demonstrate that data-driven, deep-learning approaches can surpass these traditional techniques. Innovatively, noise model-based methods like the ELD [32] have extended denoising capabilities into extreme low-light scenarios. It challenges conventional Gaussian noise models by introducing a row-based noise model that uses the Tukey Lambda distribution, which more accurately captures long-tail noise patterns in low-light conditions. The PMN [6] further investigates Fixed Pattern Noise in dark frames, enhancing alignment with the Poisson-Gaussian (P-G) distribution. These methods, however, often rely on large datasets of paired or clean images for training, as addressed by Neighbor2Neighbor [11], which trains denoising models solely on noisy images without clean counterparts. Building on the Neighbor2Neighbor approach, LGBPN [31] introduces more diverse masks to denoise spatially relevant noise. Nevertheless, Neighbor2Neighbor and LGBPN lack integration of a precise noise model or a physical prior of the 3D world, aspects our framework incorporates by merging a noise model with 3DGS to effectively separate noise from noisy images, eliminating the need for clean images. Complementing image-based methods, Yue et al. [36] expanded the scope to raw video. They introduced a dataset specifically curated for raw video denoising, along with multi-frame-based networks that leverage temporal information across frames to enhance the denoising process.

3 Preliminaries

In this section, we describe the noise model that we have adopted to regularize the extracted noise from raw images, provide an overview of the background of 3D Gaussian Splatting (3DGS)[13], and review the RawNeRF scaling loss[23] for handling the wide dynamic range in raw images.

Noise model. In the camera imaging process, depicted in Fig. 2a, the raw image $\tilde{\mathbf{x}}$ derived from the ideal clean raw image $\mathbf{x}$ inevitably contains noise $\mathbf{n}$. This relationship is expressed as:

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}. \tag{1}$$

Drawing on existing noise models [32; 6], we decompose $\mathbf{n}$ into various noise components as:

$$\mathbf{n} = \mathbf{n}_{shot} + \mathbf{n}_{read} + \mathbf{n}_{fp}, \tag{2}$$

where $\mathbf{n}_{shot}$, $\mathbf{n}_{read}$, and $\mathbf{n}_{fp}$ denote shot noise, read noise, and fixed pattern noise, respectively. A comprehensive description of each noise component is provided in the Supplemental Material. Given that $\mathbf{n}_{shot}$ can be approximated from the Poisson distribution $\mathcal{P}(\frac{\mathbf{x}}{k}) \cdot k - \mathbf{x}$ to a Gaussian distribution $\mathcal{N}(0, \mathbf{x} \cdot k)$, where k represents the camera system gain, it can be seamlessly combined with $\mathbf{n}_{read}$, which also follows a Gaussian distribution $\mathcal{N}(0, \sigma^2_{read})$. This integration yields a heteroscedastic Gaussian noise model:

$$\mathbf{n}_{hg} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{hg}), \quad \boldsymbol{\sigma}^2_{hg} = \sigma^2_{read} + \mathbf{x} \cdot k. \tag{3}$$

Thus, the overall noise model simplifies to:

$$\mathbf{n} = \mathbf{n}_{hg} + \mathbf{n}_{fp}. \tag{4}$$

Utilizing Eq. (4) as the prior for noise modeling allows for the potential separation of the clean image $\mathbf{x}$ from noise corruption, enabling more precise 3DGS reconstructions.

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [13] models a 3D scene using a collection of 3D Gaussians, *i.e.*, $\mathcal{G} = \{\mathbf{g}_1, \ldots, \mathbf{g}_M\}$, which are rendered from various viewpoints via a differentiable splatting and tile-based rasterization process. Each Gaussian $\mathbf{g}_i$ is initialized from Structure-from-Motion (SfM) and characterized by a 3D covariance matrix $\boldsymbol{\Sigma}_i = \mathbb{R}^{3\times3}$, position vectors $\boldsymbol{\mu}_i \in \mathbb{R}^3$, a color vector $\mathbf{c}_i \in \mathbb{R}^3$, and opacity o_i. To maintain the positive semi-definiteness of $\boldsymbol{\Sigma}_i$, it is parameterized as $\boldsymbol{\Sigma}_i = \mathbf{R}_i\mathbf{S}_i\mathbf{S}_i^T\mathbf{R}_i^T$, where $\mathbf{R}_i \in \mathbb{R}^{3\times3}$ and diagonal matrix $\mathbf{S}_i \in \mathbb{R}^{3\times3}$ represent rotation and scaling matrices, respectively. To render a pixel $\hat{\mathbf{x}}(\mathbf{p})$ at a pixel position $\mathbf{p}$ in the reconstructed image $\hat{\mathbf{x}}$ from a selected viewpoint within the camera projection $\mathcal{J}$, 3DGS utilizes a differentiable splatting operation to splat 3D Gaussians onto the 2D surface as follows:

$$\begin{aligned} \hat{\mathbf{x}}(\mathbf{p}) &= \sum_{i=1}^{M} \mathbf{c}_i \alpha_i(\mathbf{p}) \prod_{j=1}^{i-1} (1 - \alpha_j(\mathbf{p})), \\ \alpha_i(\mathbf{p}) &= o_i \cdot \exp\left(-\frac{1}{2} (\mathbf{p} - \mathcal{J}(\boldsymbol{\mu}_i))^T \mathcal{J}(\boldsymbol{\Sigma}_i^{-1}) (\mathbf{p} - \mathcal{J}(\boldsymbol{\mu}_i)) \right). \end{aligned} \tag{5}$$

All attributes, *i.e.*, $\mathbf{g}_i = \{\boldsymbol{\mu}_i, \mathbf{R}_i, \mathbf{S}_i, \mathbf{c}_i, o_i\}$, are learnable and optimized through the loss function:

$$\mathcal{L}_{\text{3DGS}}(\hat{\mathbf{x}}, \mathbf{x}) = (1 - \lambda)\mathcal{L}_1(\hat{\mathbf{x}}, \mathbf{x}) + \lambda L_{\text{D-SSIM}}(\hat{\mathbf{x}}, \mathbf{x}). \tag{6}$$

RawNeRF scaling loss. As discussed in RawNeRF [23], the dynamic range of colors in a raw image is vast, which complicates the application of standard $\mathcal{L}_1$ or $\mathcal{L}_2$ loss. These losses tend to overemphasize errors in brighter areas, leading to tone-mapped images with poorly rendered, low-contrast dark regions. To address this, RawNeRF proposes a scaling loss as:

$$\mathcal{L}_{\text{RawNeRF}}(\hat{\mathbf{x}}, \mathbf{x}) = \left(\frac{\hat{\mathbf{x}} - \mathbf{x}}{\text{sg}(\hat{\mathbf{x}}) + \epsilon} \right)^2, \tag{7}$$

where $\text{sg}(\cdot)$ is the stop gradient function, and ϵ is a small constant to prevent division by zero.

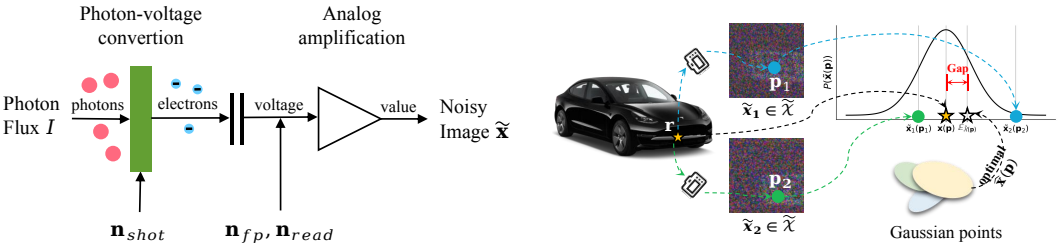
4 Methodology

4.1 Motivation

Consider a set of noisy raw images, denoted as $\tilde{\mathcal{X}} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N\}$, which are captured from different viewpoints using the same camera settings. For any real-world point, *e.g.*, $\mathbf{r}$, that is captured by all cameras, its intensity is recorded at a specific pixel coordinate $\mathbf{p}_i$ in each raw image $\tilde{\mathbf{x}}_i$, with the corresponding pixel value $\tilde{\mathbf{x}}_i(\mathbf{p}_i)$. As discussed in Sec. 3, the noisy pixel value is the sum of the noise-free value and the noise at that location, *i.e.*, $\tilde{\mathbf{x}}_i(\mathbf{p}_i) = \mathbf{x}(\mathbf{p}) + \mathbf{n}_i(\mathbf{p}_i)$. Given the same camera settings and real-world point $\mathbf{r}$, the clean value $\mathbf{x}(\mathbf{p})$ remains constant across images.

To further explore the noise impact on 3D Gaussian Splatting (3DGS) [13] optimization, we simplify our analysis by utilizing an $\mathcal{L}_2$ loss function. The optimal target for the 3DGS output $\hat{\mathbf{x}}(\mathbf{p})$, which corresponds to different positions $\mathbf{p}_i$ in each image $\tilde{\mathbf{x}}_i$ after the splatting process with camera projection $\mathcal{J}_i$, is the following:

$$\hat{\mathbf{x}}(\mathbf{p}) = \arg\min_{\hat{\mathbf{x}}(\mathbf{p})} \sum_{i=1}^{N} \mathcal{L}_2(\hat{\mathbf{x}}(\mathbf{p}), \tilde{\mathbf{x}}_i(\mathbf{p}_i)) = \mathbf{x}(\mathbf{p}) + \frac{1}{N} \cdot \sum_{i=1}^{N} \mathbf{n}_i(\mathbf{p}_i). \tag{8}$$



(a) Noise is inevitable in the imaging process

(b) Impact of noise on the 3DGS optimal target

Figure 2: An illustration of how prevalent noise in raw images impacts the 3DGS optimization. (a) The imaging process inherently introduces additive noise at various stages due to physical principles and hardware limitations, represented as $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}$, where $\tilde{\mathbf{x}}$ and $\mathbf{x}$ denote the noisy and clean images, respectively. (b) For a real-world point $\mathbf{r}$, a collection of raw images $\tilde{\mathcal{X}} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2\}$ records its intensity at pixel coordinates $\{\mathbf{p}_1, \mathbf{p}_2\}$, influenced by noise. The optimal target of 3DGS for this point, denoted as $\hat{\mathbf{x}}(\mathbf{p}) = \mathbb{E}_{\tilde{\mathbf{x}}(\mathbf{p}) \sim \tilde{\mathcal{X}}}$, has a discrepancy from the clean pixel intensity $\mathbf{x}(\mathbf{p})$. The variance of this discrepancy is detailed in Eq. (9).

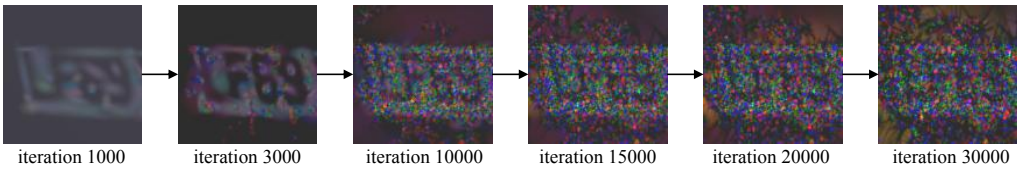


Figure 3: Visualization of the 3DGS test view changes across optimization iterations. Initially, the 3DGS model fits the clean signal (at 1,000 and 3,000 iterations). However, as the iterations progress (from 10,000 to 30,000), the model starts to overfit the noise.

Considering a real-world flat surface around point $\mathbf{r}$ where intensities remain consistent, independent noise will cause variations in the optimal target across the surface as:

$$\mathrm{Var}(\hat{\mathbf{x}}(\mathbf{p})) = \frac{1}{N^2} \cdot \mathrm{Var}(\mathbf{n}_i(\mathbf{p}_i)) = \frac{1}{N^2} \cdot \boldsymbol{\sigma}_{hg}^2. \tag{9}$$

With the same camera settings, it is evident that the variance of the optimal target $\hat{\mathbf{x}}(\mathbf{p})$ is inversely proportional to the number of training views N. Since 3D Gaussians are splatted according to Eq. (5), the splatting process tends to produce numerous thin, flat Gaussian shapes to compensate for this variance. With fewer viewpoints, which is a common setting in real-world applications, these elongated Gaussians dominate, as illustrated in Fig. 1. This dominance not only degrades the reconstruction quality but also affects rendering speed, as the performance is directly linked to the number of Gaussian points M.

Although 3DGS decomposes the variance on a flat surface into numerous elongated Gaussians, we observed that these Gaussian points tend to collapse and are subsequently divided during the later stages of the optimization process. In this process, the low-frequency signals are the first to be approximated before noise, as depicted in Fig. 3, which aligns with Deep Image Prior (DIP) [30], in which the low-frequency image components are typically reconstructed before noise. Based on such an observation, we propose incorporating a noise model as a prior in Sec. 4.3 to relax the constraints in the 3DGS optimization framework.

4.2 Lens distortion correction

Unlike NeRF [24], which employs ray-tracing directly from image data, 3DGS maps 3D Gaussians onto a 2D image plane through a splatting operation. To enhance processing efficiency, 3DGS adopts the straightforward pinhole camera model. However, this model does not account for the nonlinear distortions—primarily radial—that are typical with real-world camera lenses. These distortions are more significant in raw images, where the typical lens corrections provided by ISP are absent.

To effectively correct for both radial and tangential lens distortions, we employ a distortion mapping function $\mathcal{D}(\cdot)$. This function transforms the undistorted image coordinates (x, y) into their

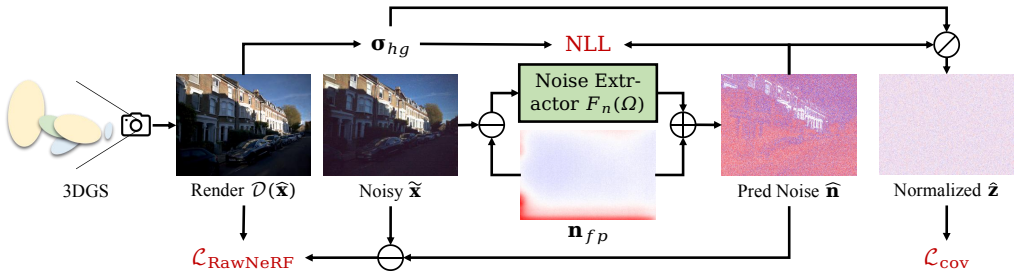


Figure 4: Illustration of the noise-robust reconstruction loss, $\mathcal{L}_{\text{nrr}}$, which comprises three components: the reconstruction loss $\mathcal{L}_{\text{RawNeRF}}$, the negative likelihood loss (NLL), and the covariance loss $\mathcal{L}_{\text{cov}}$. A noisy raw image, $\tilde{\mathbf{x}}$, is first input to the noise extractor $F_n(\cdot;\Omega)$ to estimate the noise, $\hat{\mathbf{n}}$. The estimated noise $\hat{\mathbf{n}}$ is then used to calculate the NLL loss relative to the noise distribution. After that, the normalized noise, $\hat{\mathbf{z}}$, undergoes a covariance loss, $\mathcal{L}_{\text{cov}}$, to minimize spatial dependencies among noise components. Finally, the reconstruction loss, $\mathcal{L}_{\text{RawNeRF}}$, is computed between the rendered distorted image $\mathcal{D}(\hat{\mathbf{x}})$ and the pseudo clean image $\tilde{\mathbf{x}} - \hat{\mathbf{n}}$.

corresponding distorted coordinates (x_d, y_d), based on the distortion model used in COLMAP [27]. The distortion map is derived through an iterative Newton-Raphson method. This computation is performed once prior to the training phase, thereby not affecting the training duration. Detailed explanations of the distortion mapping process are provided in the Supplemental Material.

4.3 Noise robust reconstruction loss

To mitigate overfitting to noise in 3DGS, we have revised the reconstruction loss function, now termed the noise-robust reconstruction loss function $\mathcal{L}_{\text{nrr}}$, as depicted in Fig. 4. This function is defined as follows:

$$\mathcal{L}_{\text{nrr}}(\mathcal{D}(\hat{\mathbf{x}}), \tilde{\mathbf{x}}) = \mathcal{L}_{\text{RawNeRF}}(\mathcal{D}(\hat{\mathbf{x}}), \tilde{\mathbf{x}} - \hat{\mathbf{n}}) + \lambda_{\text{nd}} \cdot \mathcal{L}_{\text{nd}}(\hat{\mathbf{n}}, \mathbf{n}), \tag{10}$$

where $\mathcal{D}(\cdot)$ denotes the lens distortion mapping as described in Sec. 4.2. The function $\mathcal{L}_{\text{nd}}(\hat{\mathbf{n}}, \mathbf{n})$ measures the noise divergence between the estimated noise $\hat{\mathbf{n}} \sim q_{\hat{\mathbf{n}}}$ and the physical noise model $\mathbf{n} \sim p_{\mathbf{n}}$. The term λ_{nd} is a Lagrangian relaxation parameter that balances fidelity and noise reduction. To estimate the noise, we introduce a noise extractor $F_n(\cdot;\Omega)$, a neural network parameterized by Ω, which predicts the noise $\hat{\mathbf{n}}$ from the input raw image $\tilde{\mathbf{x}}$ as:

$$\hat{\mathbf{n}} = F_n(\tilde{\mathbf{x}} - \mathbf{n}_{fp}; \Omega) + \mathbf{n}_{fp}. \tag{11}$$

Here, we subtract the input by the fixed pattern noise $\mathbf{n}_{fp}$ before feeding it into the noise extractor $F_n(\cdot;\Omega)$, due to the consistent nature of fixed pattern noise across different captures.

For the noise divergence loss $\mathcal{L}_{\text{nd}}(\hat{\mathbf{n}}, \mathbf{n})$, we first use the NLL Loss to measure the divergence between the estimated noise and the physical noise in corresponding pixel coordinates. The NLL loss is defined as:

$$\text{NLL}(\hat{\mathbf{n}}, \mathbf{n}) = \mathbb{E}_{q_{\hat{\mathbf{n}}}}[-\log p_{\mathbf{n}}(\hat{\mathbf{n}})], \quad p_{\mathbf{n}} = \mathcal{N}(0, \boldsymbol{\sigma}_{hg}^2) + \mathbf{n}_{fp}, \tag{12}$$

where $\boldsymbol{\sigma}_{hg}^2 = \sigma_{read}^2 + \hat{\mathbf{x}} \cdot k$ is calculated according to Eq. (3).

The NLL loss models the distribution of predicted noise across each pixel, but it does not address the inter-pixel correlations, which ideally should be minimal. To tackle this, we then standardize the predicted noise $\hat{\mathbf{n}}$ to a standard Gaussian distribution $\mathcal{N}(0, 1)$ and introduce a covariance loss $\mathcal{L}_{\text{cov}}$ to minimize spatial dependencies:

$$\mathcal{L}_{\text{cov}}(\hat{\mathbf{n}}, \mathbf{n}) = \mathbb{E}_{q_{\hat{Z}|\hat{X}}}[I - \hat{\mathbf{z}}^T \hat{\mathbf{z}}], \quad \hat{\mathbf{z}} = (\hat{\mathbf{n}} - \mathbf{n}_{fp})/\boldsymbol{\sigma}_{hg}, \tag{13}$$

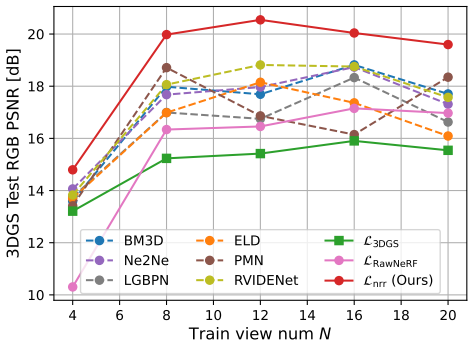
where I is the identity matrix, ensuring the decorrelation of noise across pixels.

Consequently, the comprehensive noise divergence loss $\mathcal{L}_{\text{nd}}(\hat{\mathbf{n}}, \mathbf{n})$ is formulated as:

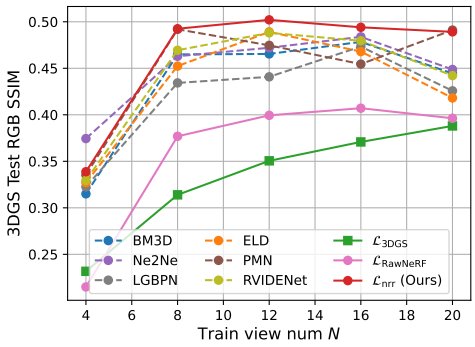
$$\mathcal{L}_{\text{nd}}(\hat{\mathbf{n}}, \mathbf{n}) = \text{NLL}(\hat{\mathbf{n}}, \mathbf{n}) + \lambda_{\text{cov}}/\lambda_{\text{nd}} \cdot \mathcal{L}_{\text{cov}}(\hat{\mathbf{n}}, \mathbf{n}). \tag{14}$$

Finally, the complete noise-robust reconstruction loss can be formulated as:

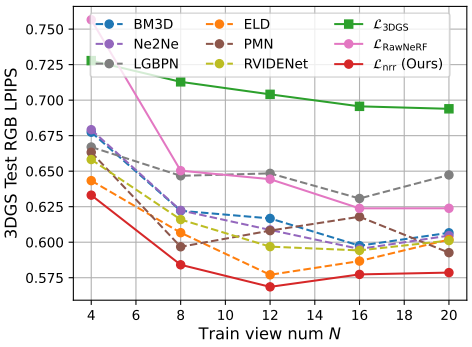
$$\mathcal{L}_{\text{nrr}}(\mathcal{D}(\hat{\mathbf{x}}), \tilde{\mathbf{x}}) = \mathcal{L}_{\text{RawNeRF}}(\mathcal{D}(\hat{\mathbf{x}}), \tilde{\mathbf{x}} - \hat{\mathbf{n}}) + \lambda_{\text{nd}} \cdot \text{NLL}(\hat{\mathbf{n}}, \mathbf{n}) + \lambda_{\text{cov}} \cdot \mathcal{L}_{\text{cov}}(\hat{\mathbf{n}}, \mathbf{n}). \tag{15}$$



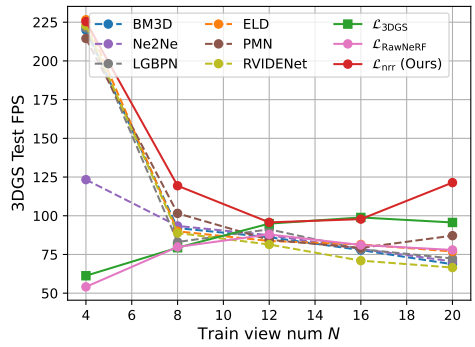
(a) PSNR over the number of views in training.



(b) SSIM over the number of views in training.



(c) LPIPS over the number of views in training.



(d) FPS over the number of views in training.

Figure 5: Comparative evaluation of various baselines and our method on rendering quality and speed in limited views training settings. The two-stage denoiser + 3DGS methods are represented by dotted lines, while training on RGB images is indicated by square markers. All metrics are evaluated on test views within the RGB domain.

5 Experiments

In this section, we introduce the experimental setup, followed by comparing our methods with existing baselines. Finally, we conduct ablation studies to facilitate a thorough discussion of our approach.

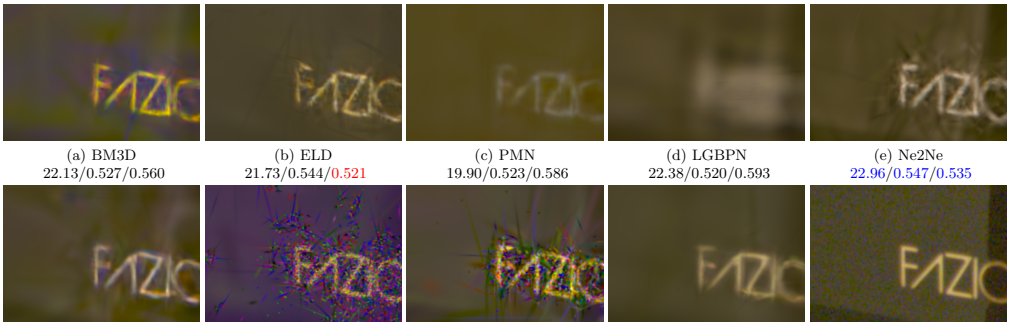
5.1 Implementation details

Our framework is built upon Scaffold-GS [21]. Considering the challenges associated with training $F_n(\cdot, \Omega)$ from scratch within only 30,000 iterations, we opted to pretrain it on the SID dataset [2], which features paired images of noise and noise-free scenes. Notably, the SID dataset is captured using a Sony DSLR camera, distinctly different from the iPhone X used in the RawNeRF dataset [23] for evaluation (e.g., bit depth differences: 14 vs 12), thereby mitigating the risk of data leakage.

Noise extractor $F_n(\cdot, \Omega)$ is optimized using the Adam optimizer with an initial learning rate of 1×10^{-4}, reduced to 1×10^{-5} after 25,000 iterations. The hyperparameters λ_{nd} and λ_{cov}, as defined in Eq. (15) and Eq. (14), are set to 5 and 20 for full-views settings, and 3 and 20 for limited views settings, respectively. All 3DGS models are trained on a single NVIDIA RTX 3090 GPU, following the default parameter settings of Scaffold-GS.

5.2 Comparison with baselines

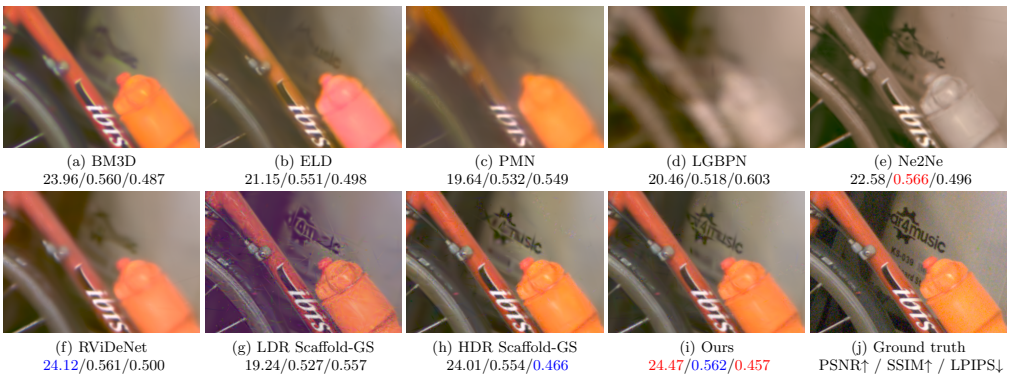
Baselines. We first compare our method with LDR Scaffold-GS [21], which uses RGB images processed by an ISP as the optimal target, to demonstrate the benefits of training on raw images. The ISP settings are the same as those used in RawNeRF. We also benchmark our method against HDR Scaffold-GS optimized with the $\mathcal{L}_{\text{RawNeRF}}$ loss using raw images as a fair baseline. Additionally, we evaluate our approach against two-stage denoiser+3DGS pipelines optimized using the $\mathcal{L}_{\text{RawNeRF}}$ loss, where 3DGS is optimized on denoised raw images produced by the denoiser. These comparisons



(a) BM3D 22.13/0.527/0.560 | (b) ELD 21.73/0.544/0.521 | (c) PMN 19.90/0.523/0.586 | (d) LGBPN 22.38/0.520/0.593 | (e) Ne2Ne 22.96/0.547/0.535

(f) RViDeNet 22.96/0.545/0.548 | (g) LDR Scaffold-GS 17.20/0.366/0.701 | (h) HDR Scaffold-GS 20.18/0.457/0.605 | (i) Ours 23.63/0.551/0.521 | (j) Ground truth PSNR↑ / SSIM↑ / LPIPS↓

Figure 6: Visual comparison using ours and competing methods under the 12-view training settings.



(a) BM3D 23.96/0.560/0.487 | (b) ELD 21.15/0.551/0.498 | (c) PMN 19.64/0.532/0.549 | (d) LGBPN 20.46/0.518/0.603 | (e) Ne2Ne 22.58/0.566/0.496

(f) RViDeNet 24.12/0.561/0.500 | (g) LDR Scaffold-GS 19.24/0.527/0.557 | (h) HDR Scaffold-GS 24.01/0.554/0.466 | (i) Ours 24.47/0.562/0.457 | (j) Ground truth PSNR↑ / SSIM↑ / LPIPS↓

Figure 7: Visual comparison across various methods in full views (100-view) training settings.

include traditional denoiser BM3D [3], state-of-the-art pretrained image denoisers such as ELD [32] and PMN [6], the pretrained video denoiser RViDeNet [36], and advanced self-supervised methods like Neighbor2Neighbor [11] and LGBPN [31] for thorough evaluations. For the pretrained supervised denoisers, we utilized the checkpoints provided by the authors. For the self-supervised methods, we initially pretrained the denoisers on the SID dataset [2] and subsequently fine-tuned them on the training views. All compared methods apply the lens distortion function $\mathcal{D}(\cdot)$ to the rendered images to align with distorted raw images, ensuring a fair comparison.

Datasets and metrics. We utilize the RawNeRF dataset [3], which comprises raw images captured in dark scenes using an iPhone X with various ISO settings. For the full-views training setting, we adhere to the same train-test view splits as in RawNeRF. For limited views settings, we randomly select subsets (4, 8, 12, 16, 20 views) from the training views, while maintaining a consistent test view set across all experiments to ensure fair comparison. Following RawNeRF, we assess the rendering quality of various methods using PSNR, SSIM, and LPIPS metrics in the RGB domain. Additionally, we report the frames per second (FPS) to evaluate the real-time rendering capabilities of each method.

Limited views training results. In Fig. 5, we present the quantitative results of our approach compared to other baselines in limited views training settings. At first glance, it is evident that our method (indicated by red lines) significantly surpasses both the LDR Scaffold-GS, HDR Scaffold-GS, and two-stage methods across all view numbers. Compared to LDR Scaffold-GS (green lines), which utilizes RGB images processed by an ISP, our method achieves approximately a 4 dB improvement in PSNR, demonstrating the benefits of training with raw images. Additionally, when both are trained with raw images, our approach outperforms HDR Scaffold-GS (pink lines), optimized by $\mathcal{L}_{\text{RawNeRF}}$, by approximately 3 dB in PSNR across different training views, and about 4 times faster in FPS with limited 4-view settings, highlighting the efficiency of our noise robust reconstruction loss $\mathcal{L}_{\text{nrr}}$. Furthermore, our method also exceeds the two-stage denoiser + 3DGS pipelines by 1 dB, which may lose high-frequency details during the denoising process. Fig. 6 provides a visualization comparison. Notably, with 12 views, both LDR- and HDR- Scaffold-GS exhibit significant elongated Gaussian artifacts, while our method produces more accurate and detailed reconstructions.

[3] `https://bmild.github.io/rawnerf/` © google-research. Licensed under the Apache License, v2.0.

Table 1: Quantitative results of our approach and other baselines in full views training settings.

Method	Loss	Raw PSNR↑	Affine-aligned RGB			FPS↑
			PSNR↑	SSIM↑	LPIPS↓	
BM3D [3]	$\mathcal{L}_{\text{RawNeRF}}$	58.46	22.70	0.526	0.521	76
ELD [32]		54.70	19.82	0.511	0.544	80
PMN [6]		53.69	19.00	0.498	0.584	94
Neighbor2Neighbor [11]		58.40	22.22	0.531	0.518	78
LGBPN [31]		53.62	19.09	0.481	0.624	84
RViDeNet [36]		57.59	22.05	0.518	0.548	74
LDR Scaffold-GS [21]	$\mathcal{L}_{\text{3DGS}}$	-	17.34	0.486	0.622	56
HDR Scaffold-GS [21]	$\mathcal{L}_{\text{RawNeRF}}$	58.08	22.69	0.521	0.513	73
Ours	$\mathcal{L}_{\text{nrr}}$	59.49	23.53	0.535	0.499	80

Table 2: Ablation study on the lens distortion $\mathcal{D}(\cdot)$. All models are trained using full views.

$\mathcal{D}(\cdot)$	Raw PSNR	RGB PSNR	RGB SSIM	RGB LPIPS
w/o	59.33	23.42	0.531	0.509
w/	**59.49**	**23.53**	**0.535**	**0.499**

Table 3: Ablation study on the training time.

Method	Ne2Ne [11] Scaffold-GS [21]	LGBPN [31] Ours
Time cost	2.5h 1.6h	5.3h 3.1h

Full-views training results. We also benchmarked our method against other baselines in full-views training settings. As shown in Table. 1 and Fig. 7, our method consistently outperforms all other baselines across PSNR, SSIM, and LPIPS, even with a large number of training views. Besides, visualizations demonstrate that our method recovers images with better detail and color accuracy.

Table 4: Ablation study on the selection of λ_{nd} and λ_{cov}. All models are trained using full views.

λ_{nd}	λ_{cov}	Raw PSNR	RGB PSNR	RGB SSIM	RGB LPIPS
0.5	20	56.36	21.95	0.508	0.591
10	20	56.84	22.18	0.509	0.539
5	0	55.38	22.41	0.515	0.523
5	2	57.10	22.52	0.516	0.520
0.5	50	56.72	21.98	0.514	0.576
5	**20**	**59.49**	**23.53**	**0.535**	**0.499**

5.3 Ablation study and discussions

Impact of lens distortion function $\mathcal{D}(\cdot)$. As detailed in Table 2, the absence of lens distortion correction results in a marginal decrease in reconstruction quality. In RawNeRF datasets, this effect is negligible due to the minimal lens distortion inherent in the iPhone X. However, for cameras with more pronounced distortion characteristics, the impact could be significantly more substantial.

Impact of hyperparameters. The hyperparameters λ_{nd} and λ_{cov} within the $\mathcal{L}_{\text{nrr}}$ play a pivotal role in striking the right balance between denoising efficacy and gradient accumulation. As delineated in Table 4, an undersized λ_{nd} leads to inadequate constraints on the noise model, culminating in imprecise noise extraction. On the flip side, an over-amplified λ_{nd} engenders excessively strong gradients emanating from the noise regularization loss, which can detrimentally impact the optimization process of 3DGS. The parameter λ_{cov}, when assigned values that are either too low or too high, can incite similar challenges.

Limitations. Our method has several limitations. The noise extractor and covariance loss computation require more training time compared to standard Scaffold-GS, as shown in Table 3. Additionally, some high-frequency signal distributions are similar to noise, making it difficult to differentiate them using the distribution divergence loss. This can result in over-smoothing in full-views training settings, as discussed in the Supplemental Material.

6 Conclusion

In this work, we identify the unavoidable noise in raw images as a significant detriment to the rendering quality and speed of HDR 3DGS, particularly in scenarios with limited training views. We provide a comprehensive analysis of how noise affects the optimization of 3DGS, modeling its relationship with both the number of training views and the noise distribution. To tackle these challenges, we

propose a novel self-supervised framework that incorporates a noise-robust reconstruction loss. This framework utilizes a physically-based noise model to simultaneously denoise and enhance the HDR representation within noisy raw images. Our approach markedly outperforms LDR/HDR 3DGS that employs 3DGS/RawNeRF loss, as well as both self-supervised and supervised pre-trained two-stage methods, in terms of rendering quality and speed across various training view counts.

Acknowledgements

This work was partially supported by the National Research Foundation Singapore Competitive Research Program (award number CRP29-2022-0003).

References

[1] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1692–1700, 2018.

[2] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[4] Y. Deng, J. Yang, J. Xiang, and X. Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10673–10683, 2022.

[5] G. Eilertsen, R. K. Mantiuk, and J. Unger. Real-time noise-aware tone mapping. *ACM Transactions on Graphics (TOG)*, 34(6):1–15, 2015.

[6] H. Feng, L. Wang, Y. Wang, and H. Huang. Learnability enhancement for low-light raw denoising: Where paired real data meets noise modeling. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1436–1444, 2022.

[7] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

[8] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14346–14355, 2021.

[9] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

[10] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021.

[11] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14781–14790, 2021.

[12] M. S. Joens, C. Huynh, J. M. Kasuboski, D. Ferranti, Y. J. Sigal, F. Zeitvogel, M. Obst, C. J. Burkhardt, K. P. Curran, S. H. Chalasani, et al. Helium ion microscopy (him) for the imaging of biological samples at sub-nanometer resolution. *Scientific reports*, 3(1):3514, 2013.

[13] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[14] S. Koskinen, D. Yang, and J.-K. Kämäräinen. Reverse imaging pipeline for raw rgb image augmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2896–2900. IEEE, 2019.

[15] N. Levin, C. C. Kyba, Q. Zhang, A. S. de Miguel, M. O. Román, X. Li, B. A. Portnov, A. L. Molthan, A. Jechow, S. D. Miller, et al. Remote sensing of night lights: A review and an outlook for the future. *Remote Sensing of Environment*, 237:111443, 2020.

[16] C. Li, S. Li, Y. Zhao, W. Zhu, and Y. Lin. Rt-nerf: Real-time on-device neural radiance fields towards immersive ar/vr rendering. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9, 2022.

[17] S. Li, C. Li, W. Zhu, B. Yu, Y. Zhao, C. Wan, H. You, H. Shi, and Y. Lin. Instant-3d: Instant neural radiance field training towards on-device ar/vr 3d reconstruction. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–13, 2023.

[18] Z. Li, L. Li, and J. Zhu. Read: Large-scale neural scene rendering for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1522–1529, 2023.

[19] Z. Li, M. Lu, X. Zhang, X. Feng, M. S. Asif, and Z. Ma. Efficient visual computing with camera raw snapshots. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[20] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[21] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. *arXiv preprint arXiv:2312.00109*, 2023.

[22] G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023.

[23] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022.

[24] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[25] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[26] C. Reiser, S. Peng, Y. Liao, and A. Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14335–14345, 2021.

[27] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[28] T. Takikawa, A. Evans, J. Tremblay, T. Müller, M. McGuire, A. Jacobson, and S. Fidler. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.

[29] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021.

[30] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[31] Z. Wang, Y. Fu, J. Liu, and Y. Zhang. Lg-bpn: Local and global blind-patch network for self-supervised real-world denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18156–18165, 2023.

[32] K. Wei, Y. Fu, J. Yang, and H. Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2758–2767, 2020.

[33] X. Wu, J. Xu, Z. Zhu, H. Bao, Q. Huang, J. Tompkin, and W. Xu. Scalable neural indoor scene rendering. *ACM transactions on graphics*, 41(4), 2022.

[34] J. Xu, L. Peng, H. Cheng, L. Xia, Q. Zhou, D. Deng, W. Qian, W. Wang, and D. Cai. Regulating intermediate 3d features for vision-centric autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6306–6314, 2024.

[35] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.

[36] H. Yue, C. Cao, L. Liao, R. Chu, and J. Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2301–2310, 2020.

[37] X. Zhang, K. Matzen, V. Nguyen, D. Yao, Y. Zhang, and R. Ng. Synthetic defocus and look-ahead autofocus for casual videography. *arXiv preprint arXiv:1905.06326*, 2019.

[38] H. Zhu, S. Xie, Z. Liu, F. Liu, Q. Zhang, Y. Zhou, Y. Lin, Z. Ma, and X. Cao. Disorder-invariant implicit neural representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[39] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pages 145–163. Springer, 2025.

From Chaos to Clarity: 3DGS in the Dark — Supplemental Material

In this supplement, we begin by discussing the broader impacts of our method. Next, we outline the distribution and calibration of the camera noise model. Following that, we elaborate on the lens distortion function $\mathcal{D}(\cdot)$. Finally, we address the failure cases of our method as mentioned in the limitations section.

A Broader Impacts

The broader social impacts of our work can be summarized as follows:

1. **Enhanced Applications:** By significantly improving 3DGS performance in low-light conditions, our work has the potential to broaden the scope of its applications across various fields such as medical imaging, autonomous driving, and surveillance systems. These advancements could lead to better outcomes in scenarios where lighting is a critical factor.
2. **Open Access and Collaboration:** In the spirit of fostering innovation and collaboration within the research community, we will release our code as open-source. This will allow other researchers and developers to build upon our work, potentially leading to further advancements and new applications.

We have carefully considered the potential impacts of our work and do not foresee any serious negative consequences. Our contributions are intended to promote positive developments and collaboration in the field of 3DGS technology.

B Distribution of the camera noise model

Building on the work of ELD [32] and PMN [6], the noise components in Eq.(4) adhere to specific distributions:

$$\begin{aligned} \mathbf{n}_{shot} &\sim \mathcal{P}\left(\frac{\mathbf{x}}{k}\right) \cdot k - \mathbf{x}, \\ \mathbf{n}_{read} &\sim \mathcal{N}(0, \sigma_{read}^2), \\ \mathbf{n}_{fp} &= \text{ISO} \cdot \mathbf{n}_{fp_k} + \mathbf{n}_{fp_b}, \end{aligned} \tag{16}$$

where k is the overall system gain associated with the ISO setting, and $\mathcal{P}$ and $\mathcal{N}$ represent Poisson and Gaussian distributions, respectively. The terms n_{fp_k} and $n_{fp_b} \in \mathbb{R}^{H \times W}$ are pixel-wise dark frame noise components. Consistent with ELD[32] and PMN [6], the relationships among k, σ_{read}, and ISO are expressed as:

$$\begin{aligned} k &= a_k \cdot \text{ISO} + b_k, \\ \log\left(\sigma_{read}\right) &= a_{read} \cdot \log(k) + b_{read}, \end{aligned} \tag{17}$$

Parameters n_{fp_k}, n_{fp_b}, a_k, b_k, a_{read}, and b_{read} are calibrated for each camera using a series of flat-frame and dark-frame images captured at various ISO levels.

C Calibration of the camera noise model parameters

The calibration process comprises three steps:

1. Calibrating k_i for each ISO level:

- Capture 25 flat frames under consistent lighting for each exposure time Exp_j.
- Calculate the mean and variance for each color block, yielding 24 mean-variance pairs per exposure time.
- With three exposure times per ISO, gather 72 mean-variance pairs per ISO.
- Model $\mathbf{n}_{shot}$ as $\mathcal{N}(\mathbf{x}, \mathbf{x} \cdot k)$, where $\mathbf{x}$ is the mean and $\mathbf{x} \cdot k$ the variance, and calibrate k from the mean-variance relationship. Exclude points with mean values beyond 1/4 saturation due to clipping effects.

2. Calibrating $\mathbf{n}_{fp_i}$ and σ_{read_i}:

(a) HDR Scaffold-GS (PSNR: 24.59 dB) (b) Ours (PSNR: 24.85 dB)

Figure 8: Visualization of our failure cases on the RawNeRF dataset [23]. Despite achieving a higher PSNR than HDR Scaffold-GS by eliminating noise-induced artifacts, our approach resulted in overly smooth outcomes in areas with reflections.

- Capture 100 dark frames at each ISO in a dark room.
- The mean of these dark frames gives $\mathbf{n}_{fp_i}$, representing fixed pattern noise.
- Subtract $\mathbf{n}_{fp_i}$ from all dark frames and calculate variance across the total frame for $\mathbf{n}_{read}$.

3. Fitting ISO-related parameters:

- Repeat the above steps for different ISO levels to obtain a set of parameters $\mathbf{n}_{fp_i}$, σ_{read_i}.
- Fit a_k, b_k, n_{fpk}, nfp_b, a_{read}, b_{read} based on these parameters and equations Eq.16.

D Lens distortion mapping process

To align the distorted coordinates with their accurate positions, we compute a reverse mapping from (x_d, y_d) to (x, y) using a Newton-Raphson iterative method. This method minimizes the residuals between the distorted and true coordinates until the adjustments are beneath a predefined accuracy threshold. Crucially, this distortion mapping is computed once before the training process, thereby minimizing its impact on computational efficiency.

using the following equations:

$$\begin{aligned} x_d &= x\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + k_4 r^8\right) + 2p_1 xy + p_2\left(r^2 + 2x^2\right), \\ y_d &= y\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + k_4 r^8\right) + p_1\left(r^2 + 2y^2\right) + 2p_2 xy, \end{aligned} \tag{18}$$

where $r = \sqrt{x^2 + y^2}$ represents the radial distance from the center, and $k_1, k_2, k_3, k_4, p_1, p_2$ are the coefficients for radial and tangential distortions, respectively.

Since x and y are not integers, we use bilinear interpolation to obtain the values of x_d and y_d.

E Failure case

Upon examination, we have observed that certain high-frequency signals exhibit a distribution pattern that closely resembles noise, making it difficult for the noise divergence loss to differentiate between them effectively. Consequently, this has resulted in overly smooth outputs in some areas rich in detail, particularly when training with full views.

Differentiating high-frequency signals from noise is an intricate challenge. We acknowledge the complexity of this issue and are committed to further exploring potential solutions in our future research endeavors.

Table 5: Quantitative Comparison on LLFF Datasets (3 Training Views)

Method	Loss	1/4*			1/8*		
		Raw PSNR↑	RGB PSNR↑	FPS↑	Raw PSNR↑	RGB PSNR↑	FPS↑
BM3D	L_{RawNeRF}	42.902	18.032	163	22.596	17.242	272
PMN	L_{RawNeRF}	42.585	18.064	201	23.208	18.572	**389**
Neighbor2Neighbor	L_{RawNeRF}	42.774	17.977	152	23.008	18.297	372
FSGS	L_{RawNeRF}	41.197	15.496	65	21.183	15.166	132
Ours	L_{nrr}	**43.418**	**18.753**	**216**	**23.799**	**18.902**	370

*denotes the downsample ratio of the resolution, as same as [39].

F Extended evaluation on LLFF dataset

Given the limitations of the RawNeRF dataset, we conducted additional quantitative comparisons using the LLFF dataset with simulated noisy raw images and provided qualitative comparisons for the training scenes in the RawNeRF dataset. Specifically, we used the inverse ISP proposed in [14] to convert LLFF RGB images into RAW images, added synthetic noise following the iPhone X noise model, and adhered to baseline methods' settings . We used FSGS [39] as the 3D representation baseline, modifying only the loss function to our proposed $\mathcal{L}_{\text{nll}}$. The results are listed in Table 1.

G Visual comparison between NeRF-based and 3DGS-based methods

(a) RawNeRF (b) HDR Scaffold-GS (c) Ours

Figure 9: Visual comparison between NeRF-based and 3DGS-based methods in full views training settings (*Zoom in for best view*). **Noise impact is unique to 3DGS.** Compared to RawNeRF, HDR Scaffold-GS is more vulnerable to noise, generating many thin-flat 3D Gaussian points to overfit it. The MLP in RawNeRF, acting as a low-pass filter, does not have this issue. Additionally, the rendering speed of 3DGS is affected by the number of Gaussian points, degrading due to noise points. Our method avoids these noisy Gaussian points, boosting rendering speed and quality.

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: The abstract and introduction outline the primary contributions and scope of our paper, providing an accurate representation of the methodologies and findings discussed in the subsequent sections.

   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: The limitations of our work are discussed in the "Limitations" section of the paper.

   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: [Yes]

Justification: We provide a detailed list of assumptions and proofs for all theoretical results in the paper.

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: We detail all the necessary information including experiment settings and hyperparameters to reproduce the main experimental results in the paper.

   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: Our code is under internal review and will be released upon paper acceptance.

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: We detail all the data splits, hyperparameters, and optimizer settings in the experimental section of the paper.

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: [No]

Justification: Considering we test our method across a broad range of training views and provide a detailed comparison with existing baselines, the extensive computational resources required to run the entire set of experiments several times make it infeasible to report error bars.

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: We report the GPU type, and time of execution for each experiment in the paper.

   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: All data and experiments comply with ethical guidelines.

   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: We discuss the broader impacts of our paper in the "Broader impacts" section.

    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: This paper poses no risks that require safeguards.

    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: We label the original datasets used in the paper and provide proper credit to the creators.

    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: [NA]

    Justification: We do not introduce new assets in the paper.

    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 in the paper.

    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 in the paper.

    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.