Supplementary

1 Implementation Details

1.1 Hyyperparameters

In table 1, we provide the full set of hyper-paramters used to train our model HDRSplat and our baselines Raw3DGS and LDR-3DGS for code replication purpose. Apart from the reduction of *scaling_lr* and *position_lr_init* as mentioned in the *main_text*, we have also reduced the *feature_lr* to 5×10^{-4} (1/5 of the original) value to prevent convergence at local minima, though in some scenes it leads to faster convergence and qualitatively better results, but also introduces floater artifacts. Note add image to showcase this

Params.	**HDRSPlat (ours)**	Raw3DGS	LDR-3DGS
Input-space	demosaiced	demosaiced	tonemapped-LDR
Bit-depth	14 bit	14-bit	8-bit
Resolution	756×1008	756×1008	756×1008
iterations	*30K*	*30K*	*30K*
densify_untill_iter	*10K*	*15K*	*15K*
densify_grad_thresh	2×10^{-4}	$10^{-4} - 1.2 \times 10^{-4}$	1.2×10^{-4}
scaling_lr	10^{-3}	5×10^{-3}	5×10^{-3}
rotation_lr	10^{-3}	10^{-3}	10^{-3}
opacity_lr	5×10^{-3}	5×10^{-3}	5×10^{-3}
feature_lr	5×10^{-4}	2.5×10^{-3}	2.5×10^{-3}
position_lr_init	8×10^{-5}	1.6×10^{-4}	1.6×10^{-4}
position_lr_final	1.6×10^{-6}	1.6×10^{-6}	1.6×10^{-6}
sh_degree	3	3	3

Table 1: Hyperparamter Detail

1.2 Dataset Processing

Terminology: In the main text *14-bit linear raw image* can be deconstructed as follows, 14-bit refers to the maximum dynamic range possible supported by our model and loss function. (Typical raw camera captures range anywhere b/w (10 -14 bits)). Linear HDR space has been leniently used to refer to the images at any stage of the pre-processing pipeline prior to the tonemapping which compresses the dynamic range of the image to 8-bits. These images are stored as 32 bit floating point tensors for computation.

Camera Undistortion: Since 3DGS does not natively support camera models with distortion unlike RawNeRF, we had to use COLMAP to undistort the train images from a *fish_eye* lens model to *simple_pinhole*.

Metadata extraction: All the raw images were saved as Adobe DNG files. We use exiftool to extract the following (table 2) metadata from the raw DNG files for post processing. The spectral response curves for each color filter element vary between different cameras, and a color correction matrix is used to convert the image from this camera-specific color space to a standardized color space. The color correction matrix C_{ccm} is an XYZ to camera RGB

Variable	EXIF Field Name	Values
w	WhiteLevel	1
b	BlackLevel	1
g_{wb}	AsShotNeutral	3
C_{ccm}	ColorMatrix2	3×3
t	ShutterSpeed	1
ISO	ISO	1

Table 2: Variable EXIF Field Names and Values

transform under the D-65 illuminant. We create C_{all} matrix to convert directly camera RGB to standard sRGB color space.

$$C_{\text{all}} = rownorm((C_{\text{rgb-xyz}} C_{\text{ccm}})^{-1})$$

where $C_{\text{rgb-xyz}}$ trasforms from from sRGB to cannonical XYZ color space.

$$C_{\text{rgb-xyz}} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix}$$

Tonemapping: We use the standard sRGB gamma curve for tonemapping during the evaluation process.

$$\gamma_{\text{sRGB}}(z) = \begin{cases} 12.92z & \text{if } z \leq 0.0031308 \\ 1.055z^{1/2.4} - 0.055 & \text{if } z > 0.0031308 \end{cases}$$

Color Correction: Each raw data source imparts different color tints to the output, in order for this to not affect metrics, we calculate a per-color-channel affine transform that best matches each method's output to the ground truth image. This is done using a *color_correction* function that aims to warp the colors in the rendered image to resemble those in a reference image (ground truth). It achieves this by iteratively solving a system of linear equations for each color channel and learning a transform matrix $C_{\text{color-correct}}$ for each view. Unlike RawNeRF we have implemented the color correction transform in the tonemapped sRGB space instead of the linear demosaiced space for all models including RawNeRF, as this gives better results qualitatively and quantitatively.

$$C_{\text{color-correct}}(img) = Iterative\_Solver(img, ref\_img)$$

Synthetic Defocus: To showcase the fidelity of our novel view and the dense depth map extracted for those novel views, we demonstrate the application of applying synthetic depth-wise varying defocus blur in the LDR space. This is different from RawNeRF where they showcase the advantage of applying defocus blur in the linear HDR space to accurately render the *bokeh_effect*.

1.3 Image Processing Pipeline

1. Load raw data using rawpy and cast to 32-bit floating point.

2. Rescale so that the black level is 0 and the white level is 1, preserving values below zero, b is the black level and w is the white level.

$$z \leftarrow \frac{z - b}{w - b}$$

3. Apply bilinear demosaicing (when necessary), the output at this stage is used to supervise RawNeRF, RaW3DGS and HDRSplat (ours) models.

4. Apply elementwise white balance gains. g_{wb}:

$$z \leftarrow \frac{z}{g_{wb}}$$

5. Apply color correction matrix C_{all} to convert from camera RGB to sRGB color space.

$$z \leftarrow C_{all} z$$

6. Adjust the exposure to set the white level to the p-th percentile (default: $p = 97$).

$$z \leftarrow \frac{z}{\text{percentile}(z, p)}$$

7. Clip pixel values to the range [0, 1]

$$z \leftarrow \text{clip}(z, 0, 1)$$

8. Apply the sRGB gamma curve to each color channel.

$$z \leftarrow \gamma_{\text{sRGB}}(z)$$

9. **Optional** Apply the per-image color correction matrix $C_{\text{color-correct}}$ for evaluation.

$$z \leftarrow C_{\text{color-correct}} z$$

2 Discussion

Denoising: As discussed in the *main_text* denoising the bayer-raw is an integral part of our pipeline as a pre-processing step due to limited ability of 3DGS algorithm to implicitly denoise the raw images. In fig. 2 we qualitatively and quantitatively compare different rule based denoisers such as BM3D[cite], bilateral-filtering and median-filtering against our adopted solution PMRID[cite].

Parameter sensitivity: In fig. 1 we study the impact of different hyper-parameters on the final render quality. We observe that lowering the *scaling_lr* causes significant improvement in the rendering quality of 3DGS in the linear HDR space. We also observe that increasing the degree of spherical harmonics beyond a certain threshold (3) does not imrpove the performance of 3DGS but only adds additional computational load. We can also see that our modifications enable 3DGS to converge to high quality results in less than 30K iterations.

Rasterization tuning for 8-bit tonemapped space: As elucidated in the *main text* , our *rasterization tuning method* significantly enhances the adaptability of 3DGS to varying depth of field and low-texture scenes in the raw image (linear HDR) space. Illustrated in Table 3, we demonstrate the efficacy of our *rasterization tuning module* in yielding superior fidelity outcomes in scenes afflicted with the aforementioned challenges, even when working in the 8-bit tonemapped (LDR) space. Thus, our findings corroborate the robustness and versatility of our tuning module as a comprehensive enhancement to the 3DGS algorithm.

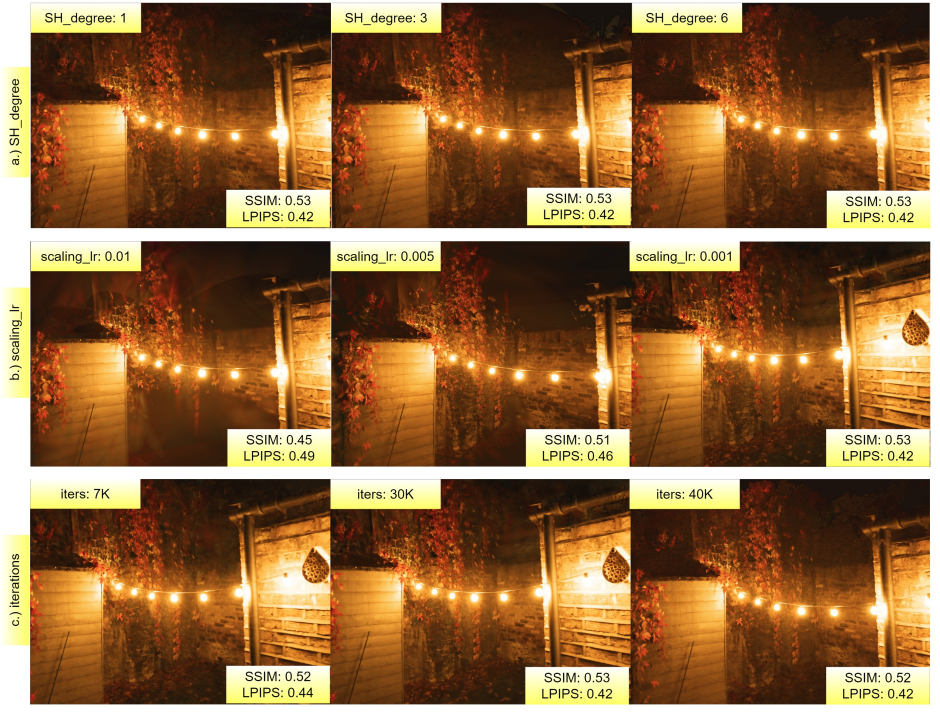

Figure 1: (a.) demonstrates that increasing the degree of Spherical Harmonics does not cause performance increase, and hence proves HDRSplat can adapt to HDR and noisy input without needing to increasing the degree of spherical harmonics. (b.) shows that decreasing the scaling_lr improves the performance of HDRSplat in overcoming the under-reconstruction issue (c.) demonstrates that HDRSplat converges within 30K iterations akin to LDR-3DGS in inder 12 minutes

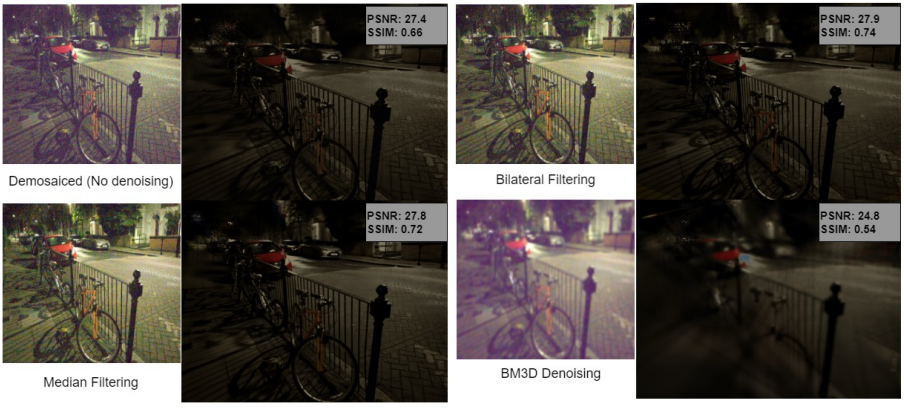

Figure 2: Comparison of different rule-based denoisers in their ability to remove photon shot noise from bayer-raw input.(For each pair Left: raw input, Right: Tonemapped LDR view generated by 3DGS

LDR-3DGS (→)	w/ rasterization tuning			w/o rasterization tuning		
Scene (↓)	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS
Bikes	30.87	0.83	0.28	29.95	0.82	0.32
Stove	33.34	0.95	0.08	28.23	0.9	0.12
Parkstatue	31.82	0.87	0.22	32.28	0.88	0.22
Sharpshadow	28.94	0.87	0.22	28.27	0.86	0.26
Candlefiat	36.05	0.9	0.28	36.05	0.9	0.27
Notchbush	30.35	0.81	0.37	29.25	0.8	0.38
Nightstreet	33.08	0.89	0.2	31.57	0.89	0.22
Morningkitchen	32.82	0.88	0.26	32.28	0.87	0.27
Livingroom	30.43	0.91	0.18	30.5	0.91	0.19
Gardenlights	24.38	0.56	0.43	25	0.56	0.43
Scooter	36.96	0.9	0.31	36.96	0.9	0.32
Streetcorner	30.94	0.88	0.24	31.63	0.88	0.25
Average	**31.66**	**0.85**	**0.26**	**30.99**	**0.84**	**0.27**

Table 3: **Quantitative Comparison** of training 3DGS w/ and w/o our *rasterization tuning method*.It clearly indicates the importance of the module in helping 3DGS render high fidelity results in low texture and variable depth of field scenes.