HDRSplat: Gaussian Splatting for High Dynamic Range 3D Scene Reconstruction from Raw Images

Shreyas Singh*
itsshreyas1803@gmail.com

Aryan Garg*
aryangarg019@gmail.com

Kaushik Mitra
kmitra@ee.iitm.ac.in

Indian Institute of Technology, Madras

Abstract

The recent advent of 3D Gaussian Splatting (3DGS) has revolutionized the 3D scene reconstruction space enabling high-fidelity novel view synthesis in real-time. However, with the exception of RawNeRF, all prior 3DGS and NeRF-based methods rely on 8-bit tone-mapped Low Dynamic Range (LDR) images for scene reconstruction. Such methods struggle to achieve accurate reconstructions in scenes that require a higher dynamic range. Examples include scenes captured in nighttime or poorly lit indoor spaces having a low signal-to-noise ratio, as well as daylight scenes with shadow regions exhibiting extreme contrast. Our proposed method HDRSplat tailors 3DGS to train directly on 14-bit linear raw images in near darkness which preserves the scenes' full dynamic range and content. Our key contributions are two-fold: Firstly, we propose a linear HDR space-suited loss that effectively extracts scene information from noisy dark regions and nearly saturated bright regions simultaneously, while also handling view-dependent colors without increasing the degree of spherical harmonics. Secondly, through careful rasterization tuning, we implicitly overcome the heavy reliance and sensitivity of 3DGS on point cloud initialization. This is critical for accurate reconstruction in regions of low texture, high depth of field, and low illumination. HDRSplat is the fastest method to date that does 14-bit (HDR) 3D scene reconstruction in $\leq$15 minutes/scene ($\sim$30x faster than prior state-of-the-art RawNeRF). It also boasts the fastest inference speed at $\geq$120fps. We further demonstrate the applicability of our HDR scene reconstruction by showcasing various applications like synthetic defocus, dense depth map extraction, and post-capture control of exposure, tone-mapping and view-point. Source code available at https://github.com/shreyesss/HDRSplat.

1 Introduction

Since 2022, Kerbl *et al.*'s seminal work on real-time high-fidelity scene reconstruction: 3D Gaussian Splatting (3DGS [20]), has spawned a myriad of impactful applications, ranging from critical industries like health, urban planning, [8, 21, 25, 27] and robotic-navigation [3]
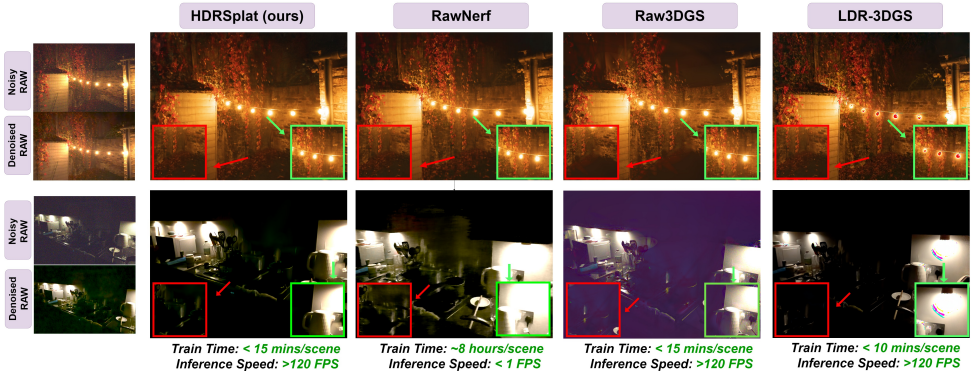
* denotes equal contribution

Figure 1: **Qualitative comparison. HDRSplat** enables the highest fidelity 3D scene reconstruction at inference speed of over 120*fps*. This is in contrast with our baselines: **Raw3DGS** (trained on demosaiced raw images) & **LDR-3DGS** (trained on 8-bit LDR images), which give poor quality renders in nighttime scenes and **RawNeRF** [34] which even though similar in fidelity to ours takes 8 hours/scene to train.

to the entertainment and fashion industry [4, 9, 24, 26, 28, 37]. Despite these advancements, nighttime 3D reconstruction remains a largely unexplored challenge. While existing 3DGS [20] methods excel in well-lit environments, addressing nighttime scenarios is vital for a variety of applications like emergency response mapping, nocturnal security, military operations, urban planning, astronomy, and archaeological documentation. However, low textures, photon-shot noise and varying local illumination in the nighttime scenes present formidable challenges, enticing the need for a real-time high-fidelity algorithm in this space.

Traditional 3D reconstruction predominantly relies on 8-bit low dynamic range (LDR) images as input, which does not effectively capture nighttime or low-exposure scenes with varying illumination. This limitation manifests in various challenges like low peak signal-to-noise ratio in darkness, heightened contrast, and saturation in regions of extreme brightness. Furthermore, the conversion process from raw to 8-bit quantized LDR images introduces loss of valuable information. Midenhall *et al*. [34] introduced an approach to generate high fidelity novel views directly in the 14-bit linear HDR space, leveraging supervision from noisy raw inputs, demonstrating that High Dynamic Range (HDR) images are needed to faithfully represent such scenes. However, RawNeRF [34]'s high memory consumption accompanied by unrealistic training time of ∼8 hours per scene on a single GPU, makes it impractical for real time use cases.

Our method, HDRSplat, presents a novel adaptation of 3DGS tailored for reconstructing scenes in the same 14-bit linear HDR space. Leveraging supervision on denoised raw images, HDRSplat achieves 3D scene reconstruction in under ∼15 minutes as shown in fig. 1. Our approach also overcomes lossy and often irreversible propriety ISP post-processing thus retaining the full dynamic range and scene information. Recent literature [22, 25] has also highlighted 3DGS's inherent limitation in reconstructing high depth of field scenes with low texture and/or low exposure. This challenge stems from the over reliance of the3DGS algorithm on the quality of point cloud obtained from COLMAP's SfM [36]. THis often results in non-uniform point density in regions of bad initialization leading to blurriness, color hallucinations, and other artifacts during reconstruction. Through a task-specific loss function design in HDR space, careful rasterization tuning, and deep Bayer space denoising, we address all these issues, thus achieving state-of-the-art HDR reconstructions while also

significantly improving the point cloud quality (fig. 4). HDRSplat further enables novel view synthesis and post-capture control for these views like focus manipulation, exposure control and custom tonemapping for dynamic range compression. Additionally, the accurate mesh extraction from the 3DGS point cloud enabled by Guédon *et al*. [14] enables further downstream creative XR-applications like 3D scene editing, virtual object insertion & deletion. To summarize, we make the following four key contributions:

- We propose HDRSplat, the first 3D Gaussian Splatting adaptation for dense scene reconstruction in linear HDR space leveraging supervision directly from raw sensor data.

- HDRSplat achieves the **fastest 3D scene reconstruction in linear HDR space** in ≤ 15 minutes ($\sim$30X faster than prior state-of-the-art RawNeRF) at an inference speed of ≥ 120 *fps*.

- Through our proposed combination of a **novel loss function**, **Bayer-space denoising, and rasterization tuning** we achieve superior fidelity ((4% LPIPS($\downarrow$), 4% SSIM($\uparrow$), 0.5 dB PSNR($\uparrow$)) over prior state-of-the-art RawNeRF.

- We showcase the advantages of rendering in linear HDR space by **demonstrating applications** such as synthetic defocus, custom tonemapping, exposure adjustment and dense depth map extraction.

2 Related Work

3D Gaussian Splatting: 3DGS [20] has found extensive usage in crucial medical applications like endoscopy [38], computed tomography [35], and X-ray imaging [2], to scalable techniques [25, 27], urban navigation [21], [8] deformable object reconstruction [7], SLAM [16, 19, 30], LiDAR processing [15], robotic navigation [3], and generative 3D applications [5, 24, 26, 29, 37] ranging from style transfer [28] to 3d-editing [4, 9]. Even applications including mirror [31] and reflection reconstruction [17], relighting [13], and mesh extraction [14] for XR have been explored. Still, nighttime reconstruction using 3DGS remains largely unexplored due to the inherent challenges and constraints of the problem. Our low-light conditions exacerbate issues such as improper point cloud initialization, limited and blurred information, ambiguous scene depth, and overall high dynamic range.

Stabilizing 3DGS: Many methods have explored regularization or stabilization for 3DGS to increase its robustness for the reconstruction of blurred and depth-ambiguous (horizon) scenes. RAIN-GS [18] uses a sparse-large-variance random initialization. Approaches such as HO-Gaussian [25] utilize grid-based volumes with MLPs to sample points in underrepresented regions, which improves the optimization of geometric information in urban scenes. Similarly, Deblurring Gaussians [22] leverage a uniform distribution over the 3D space to sample new points at fixed intervals, it also employs a simple KNN strategy to interpolate the properties of the newly sampled points for densification. Trilinear Point Splatting [10] explores a feature pyramid-based rendering pipeline. However, these 3DGS-based stabilization methods operate under the *strong assumption* of well-lit and low-dynamic range conditions. Hence, they remain largely unsuitable for nighttime 3D HDR reconstructions.

Night Time HDR 3D Reconstruction: So far, only one work, to the best of our knowledge, has tackled nighttime 3D reconstruction: RawNeRf [34]. However, this is an implicit neural radiance field [33] based method relying on accurate yet extremely time-consuming ray-marching. RawNeRF renders in the raw space using a Mip-NeRF [1] backbone unlike our

denoised-HDR renderings with 3DGS as the backbone. This allows both our methods to apply custom tonemapping and develop downstream XR, postprocessing, etc. applications but our method renders at 100+ fps (30x faster training) with similar if not superior fidelity.

Denoising: Removal of photon shot noise from raw sensor output is an integral step of our end-to-end pipeline. We empirically test a set of traditional filtering-based denoisers: BM3D [6], median, and bilateral. We also use evaluate a deep neural network based denoiser introduced by Wang *et al.* (PMRID [39]). It provides the highest fidelity results (fig. 3) and successfully recovers scene information from noisy low illumination regions.

3 HDRSplat

Our end-to-end pipeline for 3D scene reconstruction consists of 3 key stages as shown in fig. 2. First, the pre-processing step involves Bayer-space denoising to remove view-dependent photon shot noise from raw images using PMRID [39]. This is followed by a bilinear demosaicing step, which outputs a 3-channel 14-bit linear raw image with a reduced per-pixel noise level. The subsequent stage uses the denoised, demosaiced linear raw images to optimize the parameters of 3D Gaussians with our stop-gradient ($sg(.)$) scaled $\mathcal{L}_1 + DSSIM$ loss. We tailor the rasterization parameters during training to implicitly handle the initialization sparsity and non-uniformity of the generated point cloud. Our final step involves a minimalist, fully flexible post-processing pipeline. This converts the novel views from 14-bit linear raw space to 8-bit tonemapped sRGB space, enabling manipulation of focus, tone-mapping, and exposure for the rendered views.
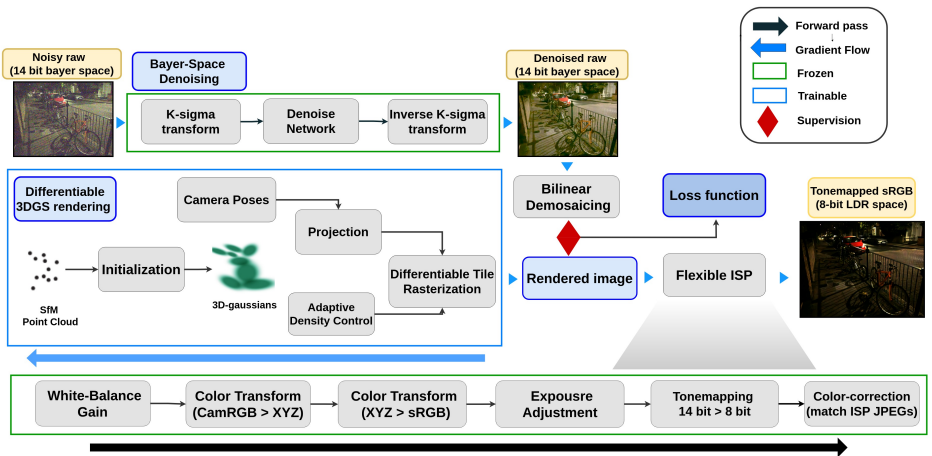

Figure 2: **Rendering Pipeline** for generating novel HDR views from noisy raw inputs. The 3 key components highlighted are: (1) **Bayer-space-denoising** (2) **Differentiable 3DGS rasterization** (3) **Flexible ISP** to convert from 14-bit linear raw to tonemapped 8-bit sRGB.

3.1 Background: 3DGS - Differentiable Rendering

Our Gaussian Splatting [20] based renderer uses point clouds to explicitly model a 3D scene. Each point in the cloud is a 3D Gaussian. According to Zwicker *et al.* [42], each 3D Gaussian (G_i) in world coordinates is centered at a unique position vector μ_i and is further defined by

an anisotropic 3D covariance matrix (Σ), and spherical harmonics (SH) to represent view-dependent appearance. The input for the proposed method consists of camera poses and point clouds, which can be obtained through the structure from motion (SfM), and a collection of images.

$$G_i(x) = e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i)}. \tag{1}$$

Zwicker *et al.* [42] also demonstrate the projection to image space from a pose represented by the viewing transformation $\mathcal{W}$, the projected covariance Σ_i^i as follows:

$$\Sigma_i^{'} = \mathcal{J}\mathcal{W}\Sigma_i\mathcal{W}^T\mathcal{J}^T. \tag{2}$$

where $\mathcal{J}$ is the Jacobian of the local affine approximation of the projective transformation. Kerbl *et al.* [20] reformulated the covariance matrix ($\Sigma = \mathcal{R}\mathcal{S}\mathcal{S}^T\mathcal{R}^T$) using scaling ($\mathcal{S}$) and quaternion matrices ($\mathcal{R}$) to satisfy the positive semi-definiteness constraint during optimization. Finally, each Gaussian (G_i) also has spherical harmonic (SH) coefficients [11] to model view-dependent colors and a learned opacity scalar α_i. After Radix-sorting [32] the Gaussians per tile, the final rendering equation at each pixel r to have a color $C(r)$ is given by:

$$C(r) = \sum_{i=1}^{N} c_i \alpha_i G_i^{'}(r) \prod_{j=1}^{i-1}(1 - \alpha_j G_j^{'}(r)) \tag{3}$$

where N is the total gaussians in the scene, c_i is the color of each gaussian and $G_i^{'}(r)$ is the ith 3D gaussian projected to the 2D image space.

3.2 Optimization

Bayer Space Denoising: Motivated by the need to enhance the supervising signal for the differentiable rasterization module, we integrate PMRID [39] ($\mathcal{G}_\theta$) as a pre-processor to remove photon shot noise in the Bayer raw space. PMRID's standout feature lies in its use of a k-sigma transform (fig. 2) to map noisy images captured under different ISO settings into an ISO-invariant signal-noise space. This approach enables a compact network trained in this ISO-invariant space to generalize and effectively denoise (varying noise) raw images.

Figure 3: Importance of bayer space denoising using PMRID: Denoised raw images successfully retrieve scene information from noisy low illumination and saturated bright regions in the raw image space due to a lowered photon shot noise level.

Loss Function: Our loss choice is inspired by Noise2Noise's (Lehtinen *et al.* [23]) stop-gradient ($sg(.)$) scaled $\mathcal{L}_2$ loss for high dynamic range (HDR) reconstruction. The scaling alleviates long-tail effects present due to unbounded luminance values in HDR samples that lead to non-convergence. $\mathcal{L}_2$ loss ensures that the expectation of noisy targets is equal to

the true pixel value when operating in the linear HDR space, which does not hold for non-linearly tonemapped LDR space ($E(\mathcal{T}(x)) \neq \mathcal{T}(E(x))$ where $\mathcal{T}$ is a non-linear tonemapping-operator). It is statistically trivial to prove that clean targets can be achieved using *only* zero-mean centered noise corrupted ground truths (or raw space) by optimizing over a squared loss over a large number of views. Multi-Layer perceptrons (MLPs) in NeRFs [1, 33, 34] have high representational power to model and remove per-scene noise implicitly, even with a low number of views. This enables Noise2Noise's [23] $\mathcal{L}_2$ formulation to work for radiance field methods. However, the explicit parameterization of 3D Gaussians lowers the representational power of 3DGS limiting its ability to learn true pixel values from noisy targets that have view-dependent photon-shot noise. Hence, we can not use the same $\mathcal{L}_2$ formulation for our method as it results in biased pixel distribution estimation and poorly learned SH coefficients. So, we incorporate a pre-trained bayer-space denoising network ($\mathcal{G}_\theta$) to eliminate this view-dependent photon-shot noise from the raw input, and a stop-gradient scaled $\mathcal{L}_1$ loss (median expectation) to ensure sharp renders from denoised raw input. Additionally, we add a D-SSIM loss term, similar to 3DGS, to improve the structural quality of the renders. Overall, our loss is given by:

$$\mathcal{L}_{total} = \lambda \left[\frac{\|\hat{y} - \mathcal{G}_\theta(y_{raw})\|_1}{(sg(\hat{y}) + \varepsilon)} \right] + (1 - \lambda) DSSIM(\hat{y}, \mathcal{G}_\theta(y_{raw})). \quad (4)$$

Rasrerization tuning: 3DGS faces inherent challenges when dealing with scenes characterized by low texture variation, low illumination, and/or high depth of field, both in LDR and HDR settings. This is attributed to the limitations of the adaptive densification and pruning module in 3DGS, which only provides localized density control over the point cloud. Specifically, splitting and pruning of Gaussians occur primarily in and around regions with dense SfM initialization [36]. Consequently, areas with poor SfM initialization suffer from severe under-reconstruction due to the absence of Gaussians that explicitly represent such regions, as shown in fig. 4:
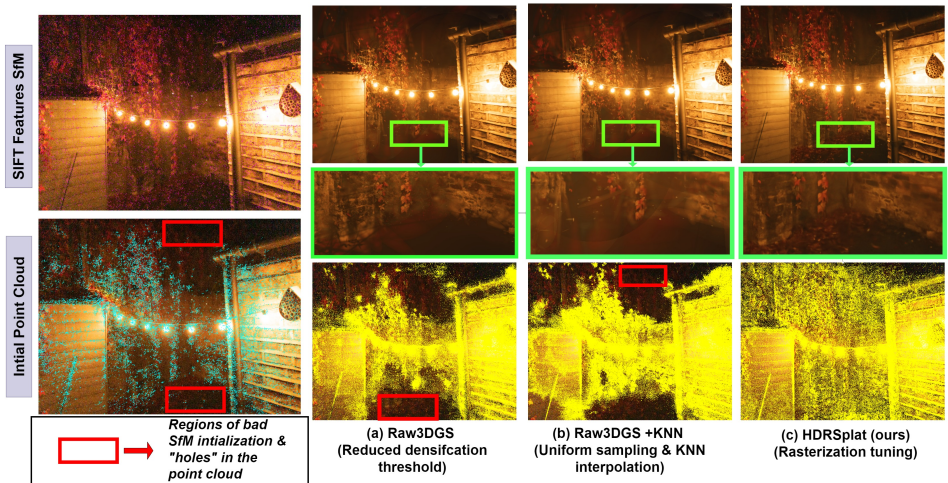

Figure 4: **Point Cloud Densification**: (a-b) highlight the non-uniformity and sparsity in regions of poor SfM initialization, leaving *holes* in the final point cloud. (c) shows improved reconstruction via our rasterization method addressing the under-reconstruction problem

By intuitively adjusting rasterization parameters, we show that avoiding Gaussian pruning

in the initial training phase effectively addresses the issue of under-reconstruction caused by the sparse and non-uniform nature of the point cloud. This simple approach contrasts with methods that forcefully augment points in the cloud, as discussed in section 2. We achieve this by halving the learning rate for scaling and setting the translation learning rate to 1/5 of the original value for all 3D Gaussians. This strategic adjustment limits the expansion and movement of Gaussians, preventing them from growing excessively large and overfitting distant sparse or uninitialized regions, or deviating significantly from their original positions, which can lead to pruning in those regions. Consequently, this *simple regularization* overcomes blurriness, floaters, and under-reconstruction in areas with poor SfM initialization.

4 Experiments & Results

Method ($\rightarrow$)	RawNeRF [34]			Raw3DGS			HDRSplat		
Scene ($\downarrow$)	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Bikes	**29.29**	0.76	**0.29**	27.10	0.65	0.38	28.49	**0.80**	**0.29**
Stove	29.42	0.92	0.11	27.72	0.82	0.13	**31.42**	**0.94**	**0.09**
Parkstatue	29.00	0.79	0.23	29.07	0.73	0.28	**30.75**	**0.85**	**0.21**
Sharpshadow	**27.94**	**0.80**	**0.21**	25.75	0.74	0.29	26.72	0.77	0.26
Candleflat	31.19	**0.86**	0.28	27.48	0.80	0.29	**31.96**	**0.86**	**0.24**
Notchbush	27.87	**0.76**	0.40	27.58	0.72	0.41	**29.18**	0.75	**0.36**
Nightstreet	28.07	0.75	0.20	28.12	0.75	0.25	**31.50**	**0.87**	**0.20**
Morningkitchen	29.55	0.77	0.28	28.77	0.77	0.30	**29.90**	**0.83**	**0.26**
Livingroom	**30.61**	**0.87**	0.19	29.39	0.85	0.22	29.12	0.86	**0.18**
Gardenlights	**24.43**	**0.53**	0.46	23.08	0.51	0.46	24.35	**0.53**	**0.42**
Scooter	35.10	**0.88**	0.28	33.22	0.79	0.33	**35.36**	**0.88**	**0.26**
Streetcorner	31.79	**0.84**	0.24	27.87	0.73	0.30	**32.16**	**0.84**	**0.23**
Average	**29.52**	**0.79**	**0.26**	**27.92**	**0.73**	**0.30**	**30.07**	**0.82**	**0.25**

Table 1: **Quantitative Comparison** of all models rendering in the 14-bit linear HDR space. We demonstrate superior fidelity of our model over RawNeRF (4% LPIPS($\downarrow$), 4% SSIM($\uparrow$), 0.5 PSNR($\uparrow$)) and our baseline Raw3DGS (17% LPIPS($\downarrow$), 12% SSIM($\uparrow$), 2.1 dB PSNR($\uparrow$)).

4.1 Implementation Details

Dataset: We use RawNeRF's [34] nighttime, 4032×3024 resolution, paired Bayer-raw and proprietary ISP LDR dataset, captured using an iPhone-X. The dataset consists of 14 scenes however we omit two scenes: Windowlegovary and Candle due to their multi-exposure captures. We undistorted the entire dataset into a *simple pinhole* camera model using COLMAP [55] since 3DGS does not provide native support for radial camera distortions.

Baselines: We compare HDRSplat with prior state-of-the-art RawNeRF [34] and our own constructed baselines: LDR-3DGS and Raw3DGS. LDR-3DGS is Kerbl *et al.*'s original implementation of 3DGS, trained on 8-bit iPhone JPEGs. Raw3DGS trains the original implementation on noisy demosaiced raw images with RawNeRF's loss (stop-gradient scaled $\mathcal{L}_2$), highlighting the limitations when directly applying RawNeRF's framework to 3DGS. Taking inspiration from [22], Raw3DGS+KNN utilizes a uniform point sampling strategy with KNN-based properties' interpolation for point cloud densification as shown in fig. 4.

Metrics: For quantitative benchmarking purposes the rendered novel views in 14-bit demosaiced space are converted to 8-bit tonemapped LDR space using our flexible post-processing pipeline as shown in fig. 2 and color-corrected using the ground truth LDRs, similar to

RawNeRF [54]. The metrics reported are SSIM, PSNR and LPIPS (VGG-16) [41]. Refer to the supplementary for exact details on the post-processing pipeline.

Experimental Setup: RawNeRF [54] follows the original setup. In Raw3DGS, we set *densify_grad_thresh* to 10^{-4} (half of original 3DGS) for maximum densification adaptability to noisy raw inputs. HDRSplat trains 3DGS [20] on PMRID [59] denoised raw inputs with our stop-gradient ($sg(.)$) scaled $\mathcal{L}_1 + DSSIM$ loss. We set *scaling_lr* to 10^{-3} (1/5 of the original implementation) and *position_lr_init* to 8×10^{-5} (1/2 of the orignal implementation). PMRID [59] denoising and bilinear demosaicing steps are pre-computed once per scene and stored, cumulatively they takes ~ 8 seconds/view to process. An exponential decay scheduling similar to [11, 20] is used for every 3DGS-based method. We use a single NVIDIA RTX 3090 with 24 GB VRAM.

4.2 Results

The quantitative comparison in table 1 and compute load table 2 highlights HDRSplat's capability to generate superior-fidelity renders extremely fast, crucial for real-time applications. Notably, HDRSplat achieves better perceptual similarity (LPIPS [41]) compared to LDR-3DGS, which is trained directly on the 8-bit iPhone JPEG ground truths. HDRSplat stands out not only for its rapid training and rendering speeds but also for its exceptional memory efficiency (table 2). On average, it utilizes only 0.35M points per scene, compared to 1.5M and 5M points in Raw3DGS and Raw3DGS+KNN, respectively.

Method	PreP* ↓	Training ↓	Total ↓	Inference	LPIPS ↓	# Points ↓
RawNeRF [54]	**3 mins**	8 hrs	8 hrs	0.1 FPS	0.30	N/A
Raw3DGS	**3 mins**	10 mins	13 mins	**120 FPS**	0.30	1.5 M
Raw3DGS+KNN	**3 mins**	20 mins	23 mins	**120 FPS**	0.28	5 M
LDR-3DGS [20]	N/A	**7 mins**	**7 mins**	**120 FPS**	0.26	**0.35 M**
HDRSplat (ours)	7 mins	**7 mins**	14 mins	**120 FPS**	**0.25**	**0.35 M**

Table 2: **Real time rendering**: We are 30x faster than RawNeRF in training and can infer at 120 fps. PreP* (Pre-processing) refers to Bayer-space denoising step and bilinear-demosaicing. All metrics have been calculated for an average of 50 views per scene.
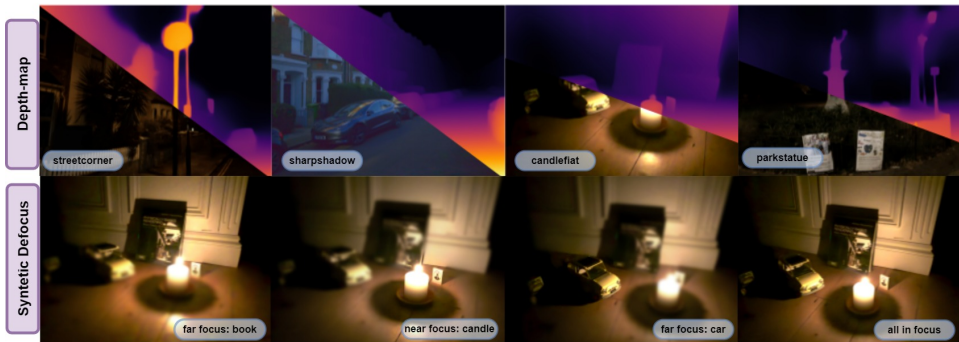

Figure 5: **Application:** Accurate Depth from Novel Views using [40] and Synthetic-Defocus

Ablation	Loss	G_θ	$\mathcal{LR}$	PSNR ↑	SSIM ↑	LPIPS ↓	# Points ↓
Raw3DGS	$\mathcal{R}$	×	×	27.92	0.73	0.30	1.5 M
w/o Rasterization Tuning	$\mathcal{H}$	✓	×	29.44	0.80	0.30	0.85 M
w/o Bayer Denoising	$\mathcal{H}$	×	✓	29.49	0.78	0.27	**0.25 M**
w/o Our Loss Function	$\mathcal{R}$	✓	✓	29.97	0.80	0.26	0.30 M
HDRSplat (Ours)	$\mathcal{H}$	✓	✓	**30.09**	**0.82**	**0.25**	0.35 M

Table 3: **Ablation Study** to demonstrate the utility of each component of our pipeline. $\mathcal{H}$ refers to our stop-gradient scaled $\mathcal{L}_1 + DSSIM$ loss, $\mathcal{R}$ refers to RawNeRF's [34] stop-gradient scaled $\mathcal{L}_2$. G_θ is Bayer-space denoising and $\mathcal{LR}$ refers to our rasterization tuning.

4.3 Ablations

In table 3 we validate the impact of each of our 3 innovations, rasterization tuning, Denoising ($\mathcal{G}_\theta$) and our loss on the rendering quality, speed & efficiency of HDRSplat. Directly training 3DGS on demosaiced raw images (Raw3DGS) consumes $\geq 4\times$ more memory and yields noticeably inferior renders (fig. 1). Our rasterization tuning method substantially reduces the memory usage by $\sim 60\%$. Bayer denoising and our proposed loss function seem to have the highest impact on perceptual quality based on LPIPS and SSIM scores.
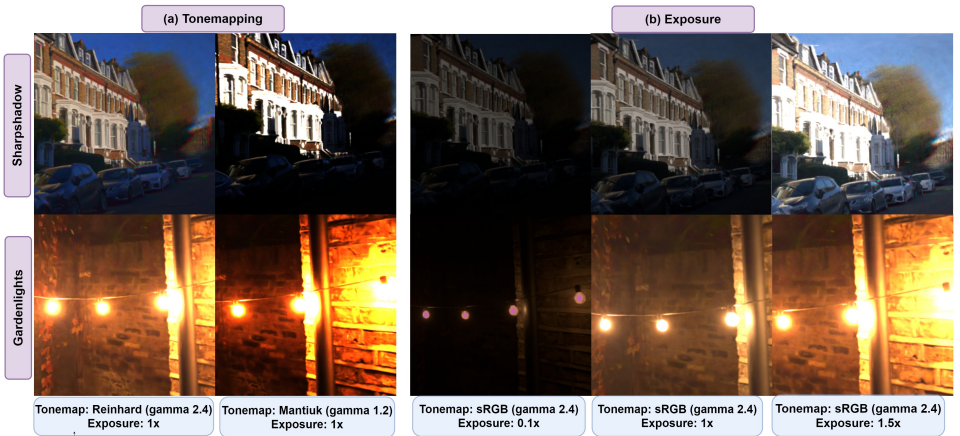

Figure 6: **Application:** HDRSplat, rendering in the linear HDR space, facilitates complete post-capture control for generating visually appealing results in downstream applications.

4.4 Applications

In fig. 5, we demonstrate applications such as novel-view synthesis, dense depth map extraction using [40] and synthetic defocus to highlight regions of interest. All enabled by our method's high-fidelity renders. fig. 6 illustrates the versatility of variable tonemappings and exposure control made possible by our linear HDR rendering space. These renderings can be retouched like any raw photograph, enabling full post capture control for the user.

5 Conclusion

Our method enables the fastest and highest fidelity HDR scene reconstruction directly from Bayer raw images, while our approach offers significant advantages, such as capturing fine

details in high dynamic range scenes and avoiding image compression artifacts, it does present some challenges. Dependency on LDR images for SfM [36] and 3DGS' high sensitivity to point cloud initialization, especially in low-light scenarios, could be alleviated in future work by using a COLMAP-free or depth projection based initialization as shown by [12]. Additionally, designing a noise model or parameterization for individual 3D Gaussians could eliminate the need for external denoising dependencies, streamlining the pipeline and improving overall compute efficiency. Despite its limitations, HDRSplat represents a significant advancement towards real-time, adaptable, and high-fidelity 3D scene reconstruction, particularly for higher dynamic range scenes.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, 2021.

[2] Yuanhao Cai, Yixun Liang, Jiahao Wang, Angtian Wang, Yulun Zhang, Xiaokang Yang, Zongwei Zhou, and Alan Yuille. Radiative gaussian splatting for efficient x-ray novel view synthesis, 2024.

[3] Timothy Chen, Ola Shorinwa, Joseph Bruno, Javier Yu, Weijia Zeng, Keiko Nagami, Philip Dames, and Mac Schwager. Splat-nav: Safe real-time robot navigation in gaussian splatting maps, 2024.

[4] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting, 2023.

[5] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting, 2024.

[6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. doi: 10.1109/TIP.2007.901238.

[7] Devikalyan Das, Christopher Wewer, Raza Yunus, Eddy Ilg, and Jan Eric Lenssen. Neural parametric gaussians for monocular non-rigid object reconstruction. *arXiv preprint arXiv:2312.01196*, 2023.

[8] Niranjan Deshpande, Dominique Vaufreydaz, and Anne Spalanzani. Navigation in urban environments amongst pedestrians using multi-objective deep reinforcement learning. *CoRR*, abs/2110.05205, 2021. URL https://arxiv.org/abs/2110.05205.

[9] Jiemin Fang, Junjie Wang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions, 2023.

[10] Linus Franke, Darius Rückert, Laura Fink, and Marc Stamminger. Trips: Trilinear point splatting for real-time radiance field rendering. 2024.

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

[12] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. 2023.

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

[14] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *arXiv preprint arXiv:2311.12775*, 2023.

[15] Sheng Hong, Junjie He, Xinhu Zheng, Hesheng Wang, Hao Fang, Kangcheng Liu, Chunran Zheng, and Shaojie Shen. Liv-gaussmap: Lidar-inertial-visual fusion for real-time 3d radiance field map rendering, 2024.

[16] Huajian Huang, Longwei Li, Cheng Hui, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[17] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces, 2023.

[18] Jaewoo Jung, Jisang Han, Honggyu An, Jiwon Kang, Seonghoon Park, and Seungryong Kim. Relaxing accurate initialization constraint for 3d gaussian splatting. *arXiv preprint arXiv:2403.09413*, 2024.

[19] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[21] Muhammed Kocabas, Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. Hugs: Human gaussian splats, 2023. URL https://arxiv.org/abs/2311.17910.

[22] Byeonghyeon Lee, Howoong Lee, Xiangyu Sun, Usman Ali, and Eunbyung Park. Deblurring 3d gaussian splatting, 2024.

[23] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In

Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2965–2974. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/lehtinen18a.html.

[24] Haoran Li, Haolin Shi, Wenli Zhang, Wenjun Wu, Yong Liao, Lin Wang, Lik-Hang Lee, and Pengyuan Zhou. Dreamscene: 3d gaussian-based text-to-3d scene generation via formation pattern sampling. *arXiv preprint arXiv:2404.03575*, 2024.

[25] Zhuopeng Li, Yilin Zhang, Chenming Wu, Jianke Zhu, and Liangjun Zhang. Hogaussian: Hybrid optimization of 3d gaussian splatting for urban scenes, 2024.

[26] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching, 2023.

[27] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *CVPR*, 2024.

[28] Kunhao Liu, Fangneng Zhan, Muyu Xu, Christian Theobalt, Ling Shao, and Shijian Lu. Stylegaussian: Instant 3d style transfer with gaussian splatting. *arXiv preprint arXiv:2403.07807*, 2024.

[29] Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. Humangaussian: Text-driven 3d human generation with gaussian splatting. *arXiv preprint arXiv:2311.17061*, 2023.

[30] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian Splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[31] Jiarui Meng, Haijie Li, Yanmin Wu, Qiankun Gao, Shuzhou Yang, Jian Zhang, and Siwei Ma. Mirror-3dgs: Incorporating mirror reflections into 3d gaussian splatting, 2024.

[32] Duane G. Merrill and Andrew S. Grimshaw. Revisiting sorting for gpgpu stream architectures. In *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, PACT '10, page 545–546, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450301787. doi: 10.1145/1854273.1854344. URL https://doi.org/10.1145/1854273.1854344.

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

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

[35] Emmanouil Nikolakakis, Utkarsh Gupta, Jonathan Vengosh, Justin Bui, and Razvan Marinescu. Gaspct: Gaussian splatting for novel ct projection view synthesis, 2024.

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] Ruizhi Shao, Jingxiang Sun, Cheng Peng, Zerong Zheng, Boyao Zhou, Hongwen Zhang, and Yebin Liu. Control4d: Efficient 4d portrait editing with text. 2024.

[38] Kailing Wang, Chen Yang, Yuehao Wang, Sikuang Li, Yan Wang, Qi Dou, Xiaokang Yang, and Wei Shen. Endogslam: Real-time dense reconstruction and tracking in endoscopic surgeries using gaussian splatting, 2024.

[39] Yuzhi Wang, Haibin Huang, Qin Xu, Jiaming Liu, Yiqun Liu, and Jue Wang. Practical deep raw image denoising on mobile devices. In *European Conference on Computer Vision (ECCV)*, pages 1–16, 2020.

[40] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[42] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. doi: 10.1109/TVCG. 2002.1021576.