

# Mipmap-GS: Let Gaussians Deform with Scale-specific Mipmap for Anti-aliasing Rendering

## Supplementary Material

In supplementary material, we first present a user study of image quality in Section 1. Next, we report per-scene quantitative and quality results in Section 2, and give more analysis on initialization, scales, zoom-in mipmap and dilation regularization in Section 3. Finally, we provide zoom-in and zoom-out videos to compare the rendering quality of 3DGS [4] and ours at varying observation settings.

### 1. User Study

We conduct a user study on image quality of 3DGS [4], Mip-Splatting [10] and ours. This study aims to assess the fidelity and overall rendering quality in novel view synthesis. Given 30 zoom-out images and 30 zoom-in images separately, users rate them from 1 (worst) to 5 (best). Unknown of generation methods and GT, the score is entirely based on human perceptual judgments. The zoom-out series consist of shuffled images from  $\times 1/2$ ,  $\times 1/4$ ,  $\times 1/8$  scales. And the scales of zoom-in images vary among  $\times 2$ ,  $\times 4$ ,  $\times 8$ . As shown in Table 1, 3DGS [4] is graded lowest for both zoom-out and zoom-in. Compared with Mip-Splatting [10], our method demonstrates better visual effects.

	3DGS [4]	Mip-Splatting[10]	Ours
Fidelity	2.85	4.23	4.65
Overall Quality	1.95	3.67	4.02
(a) Zoom-out			
	3DGS [4]	Mip-Splatting [10]	Ours
Fidelity	2.34	4.11	4.39
Overall Quality	1.66	3.54	3.89
(b) Zoom-in			

Table 1. User study on NeRF Synthetic dataset [8]. The rating is of scale 1-5, the higher the better.

### 2. Per-Scene Results

In this section, we report per-scene quantitative and quality results of our adaptation and different baseline models, namely 3DGS [4], Scaffold-GS [7], Pixel-GS [12], Mip-Splatting [10], Octree-GS [9], Analytic-Splatting [6].

We optimize both 3DGS [4] and Scaffold-GS [7] to show the compatibility and adaptation capacity of our method. As a subsequent work of 3DGS, Scaffold-GS [7] leverages anchor points to distribute local Gaussians for view-adaptive rendering. However, the scale adaptation of Scaffold-GS is still limited, leading to even worse performance for zoom-in compared with 3DGS.

In experiments, we train baseline models at  $\times 1/8$  downsampled images and render into three high resolutions to

simulate zoom-in effect. For zoom-out, we re-train baseline models at full resolution and render novel views at  $\times 1/2$ ,  $\times 1/4$ ,  $\times 1/8$  resolutions. The qualitative zoom-out and zoom-in comparisons on NeRF Synthetic dataset are shown in Table 4 and 5, evaluated by SSIM, PSNR and LPIPS [11]. We report the arithmetic mean of each metric over three different scales for each scene. In Figure 4 and 6, the renderings of baseline models exhibit dilation artifacts or erosion aliasing respectively. Our optimization successfully alleviates the aliasing artifacts for both 3DGS and Scaffold-GS.

### 3. Additional Analysis

#### 3.1. Initialize with Sparse Points

[3, 7, 12] indicate that the performance of 3DGS is highly dependent on initial points and inadequate initialized areas show poor texture structure. When starting from sparse points extracted from LR images, 3DGS suffers from not only aliasing artifacts but also missing structures when zooming in.

To explore our scale adaptation capacity at coarse initialization, we extract the sparse point cloud from  $\times 1/8$  downsampled bicycle images, with the resolution of only  $411 \times 618$ . When zooming in with factor 8, the Gaussians are rendered into image resolution of  $3286 \times 4946$ .

The memory size of trained Gaussians is illustrated in Table 2, the Gaussians initialized from more sparse points generally grow less than those initialized from default. In Figure 1, we visualize the optimized Gaussians by setting the scale modifier to 0.1 (left part of each result). Through proportional reduction, it can be observed that our method adapts finer Gaussians for a higher zoom factor to render aliasing-free images without increased storage consumption. The sparse initialization raises more challenges for the optimization, since the small or texture-less areas are insufficiently initialized (e.g. the accessories cropped at the corner in Figure 1). Nevertheless, our method is still capable of achieving high-quality rendering.

	Default			Sparse		
	$\times 2$	$\times 4$	$\times 8$	$\times 2$	$\times 4$	$\times 8$
3DGS [4]	1145			1110		
Mip-Splatting [10]	1141			1137		
Ours	958	963	970	902	909	916

Table 2. Comparison of memory size (MB) on bicycle scene. The default initial SfM points given by dataset occupy 1.38 MB, while the sparse points extracted from 1/8 downsampled images take only 0.52 MB.

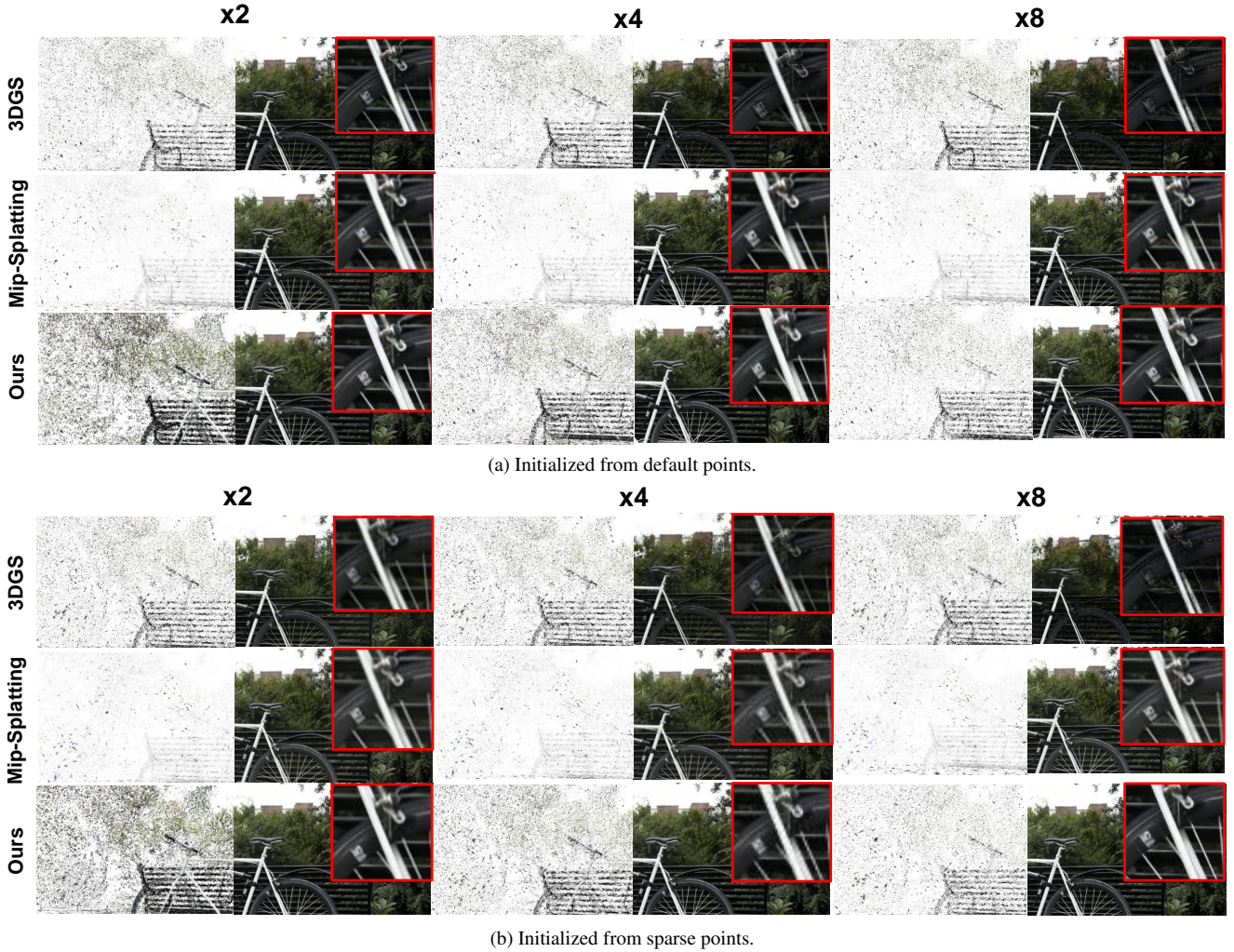


Figure 1. Comparison of zoom-in effects on Mip-NeRF 360 [1] when initialized from sparse points.

### 3.2. Rendering on More Scales

We test the baseline models and our method on extreme zoom factors to show our adaptation capacity towards varying observation settings. In Figure 2, 3DGS exhibits severe zoom-out blurriness at  $\times 1/16$ ,  $\times 1/32$ ,  $\times 1/64$  (thick buds at the branch tip) due to the invariant fine Gaussians. Mip-Splatting leverages the Mip filter for less dilated 2D Gaussians, which is less robust towards extreme observation conditions. Our adaptation merges Gaussians for more concentrated scene representation through density control and adjusts the primitive properties by backward propagation, which mitigates the dilation blurriness efficiently even in harsh conditions.

### 3.3. Different mipmap resizing methods.

**Zoom-out Mipmap.** We use the bilinear interpolation method to generate downsampled novel-view rendering. Here we report the quantitative results at 0.5 K iterations using different interpolation methods in Table 3. As observed,

bilinear and cubic interpolation achieve comparable results, while nearest interpolation performs poorly with 3.56 dB PSNR reduction at  $\times 1/8$ .

Method	SSIM			PSNR		
	$\times 1/2$	$\times 1/4$	$\times 1/8$	$\times 1/2$	$\times 1/4$	$\times 1/8$
Nearest	0.726	0.737	0.752	27.31	26.21	24.67
Cubic	0.810	0.811	0.845	27.76	28.19	28.25
Bilinear	0.811	0.825	0.850	27.75	28.22	28.23

Table 3. Comparison of different downsampling methods for zoom-out on Mip-NeRF 360 [1].

**Zoom-in Mipmap.** Except for taking SwinIR [5] to generate zoom-in mipmap, we report the optimization results when adopting bilinear interpolation, downsampling loss and video SR model [2]. Generating HR mipmap via bilinear interpolation improves the zooming results slightly, since the mipmap is blurry with insufficient details. To leverage downsampling loss, the 3D Gaussians are splatted into zoom-in scales ( $\times N$ ) and then downsampled into  $\times 1$



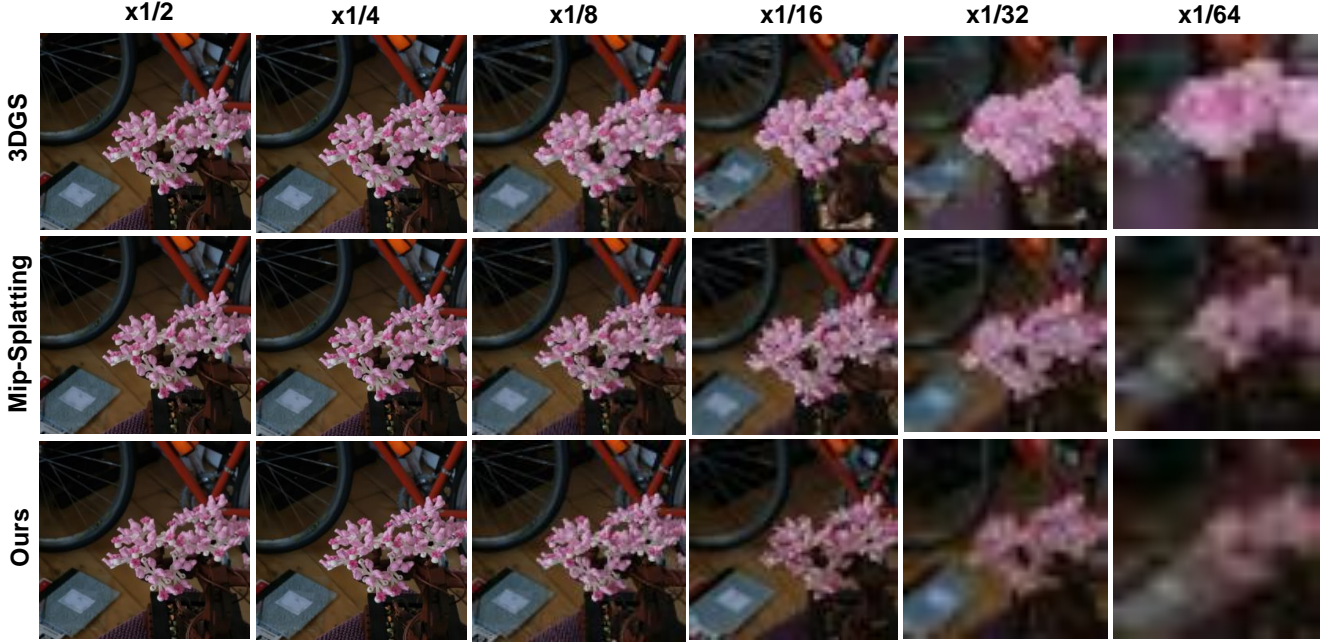


Figure 2. Comparison of zoom-out effects on Mip-NeRF 360 [1] at more scales.

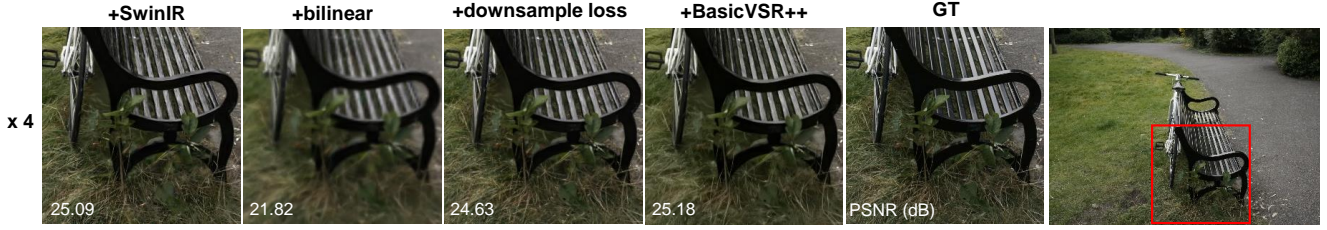


Figure 3. Comparison of different SR methods for zoom-in on bicycle scene.

to calculate pixel loss with basic-scale novel views. Here the resizing function works for current rendering, not for pseudo-GT. The final method inserts the pretrained video SR model [2] as the mipmap resizing function. To provide consistent input sequences, we first concatenate the training views and test views for a dense collection. Then we conduct frame interpolation, where an intermediate frame between two input frames is rendered. The numerically doubled images go through [2] to obtain  $\times 4$  SR mipmaps. Consistent with previous settings, the test view SR images are selected as pseudo-GT to deform the base Gaussians. From Fig. 3, bilinear mipmap and basic scale mipmap with downsampling loss exhibit sub-optimal rendering results compared with SwinIR mipmap. Video-based mipmap improves the rendering quality while introducing extra time consumption during frame interpolation and SR inference.

### 3.4. Dilation Regularization

The original dilation operation in 3DGS [4] involves passing the 2D projection through a low-pass filter, which rectifies the diagonal elements of the  $2 \times 2$  covariance matrix by a hyper-parameter  $s$  (set to be 0.3), see Eqn. 1. When zoom-

ing out, more Gaussians attend to one-pixel shading, while the dilation hyper-parameter  $s$  is invariant towards varying zoom factors. The mismatched dilation scale causes extreme brightness and thickness for zoom-out. Mip-Splatting [10] adjusts the 2D Gaussian by a Mip-filter, illustrated in Eqn. 2, which is versatile for any zoom factors. Considering the lack of scale-aware guidance in previous methods [4, 10], we regularize the dilated covariance matrix by introducing the scale factor  $1/N$  in Eqn. 3 to mitigate the extreme thickness.

$$G_k^{2D}(x)_{3DGS} = e^{-\frac{1}{2}(x-\mu_k)^\top (\Sigma_k^{2D} + sI)^{-1}(x-\mu_k)}, \quad (1)$$

$$G_k^{2D}(x)_{\text{mip}} = \sqrt{\frac{|\Sigma_k^{2D}|}{|\Sigma_k^{2D} + sI|}} e^{-\frac{1}{2}(x-\mu_k)^\top (\Sigma_k^{2D} + sI)^{-1}(x-\mu_k)}, \quad (2)$$

$$G_k^{2D}(x)_{\text{reg}} = e^{-\frac{1}{2}(x-\mu_k)^\top (\Sigma_k^{2D} + (\frac{1}{N})^2 sI)^{-1}(x-\mu_k)}. \quad (3)$$

To compare the zoom-out effects of the scale factor, we conduct training at full resolution and rendering at  $\times 1/8$  in Figure 5. The results illustrate the role of scale information in achieving aliasing-free rendering.

SSIM								
Method	ship	drums	figus	hotdog	lego	materials	mic	chair
3DGS [4]	0.827	0.851	0.921	0.930	0.882	0.882	0.991	0.915
Scaffold-GS [7]	0.806	0.821	0.909	0.929	0.857	0.868	0.892	0.899
Pixel-GS [12]	0.806	0.794	0.889	0.894	0.829	0.871	0.884	0.861
Mip-Splatting [10]	0.930	0.958	0.988	0.987	0.982	0.974	0.981	0.988
Scaffold-Ours	0.918	0.951	0.981	0.984	0.980	0.976	0.992	0.987
3DGS-Ours	0.940	0.960	0.984	0.989	0.983	0.979	0.989	0.989

---

PSNR								
Method	ship	drums	figus	hotdog	lego	materials	mic	chair
3DGS [4]	20.50	19.49	23.01	25.82	21.90	20.86	20.79	23.92
Scaffold-GS [7]	20.63	19.69	23.28	27.16	22.13	21.06	21.20	24.39
Pixel-GS [12]	23.64	19.98	23.37	25.20	22.32	24.49	25.73	21.15
Mip-Splatting [10]	29.27	26.07	31.74	34.70	31.34	29.56	29.86	35.23
Scaffold-Ours	30.77	26.56	33.57	36.34	33.46	30.37	35.21	35.38
3DGS-Ours	30.83	27.05	31.86	36.89	32.81	30.65	33.75	35.69

---

LPIPS								
Method	ship	drums	figus	hotdog	lego	materials	mic	chair
3DGS [4]	0.113	0.108	0.055	0.034	0.064	0.056	0.046	0.057
Scaffold-GS [7]	0.114	0.110	0.075	0.050	0.057	0.057	0.063	0.055
Pixel-GS [12]	0.114	0.110	0.075	0.050	0.100	0.057	0.057	0.047
Mip-Splatting [10]	0.066	0.038	0.014	0.014	0.019	0.017	0.022	0.011
Scaffold-Ours	0.078	0.038	0.010	0.020	0.018	0.024	0.008	0.013
3DGS-Ours	0.048	0.036	0.015	0.011	0.017	0.016	0.016	0.014

Table 4. Per-scene quantitative zoom-out results on NeRF Synthetic dataset [8].

SSIM								
Method	ship	drums	figus	hotdog	lego	materials	mic	chair
3DGS [4]	0.783	0.842	0.901	0.912	0.801	0.838	0.911	0.884
Scaffold-GS [7]	0.714	0.801	0.598	0.894	0.779	0.792	0.888	0.871
Pixel-GS [12]	0.769	0.821	0.878	0.896	0.792	0.843	0.905	0.864
Mip-Splatting [10]	0.852	0.899	0.929	0.958	0.906	0.924	0.958	0.938
Scaffold-Ours	0.853	0.908	0.926	0.965	0.917	0.935	0.969	0.944
3DGS-Ours	0.860	0.912	0.933	0.966	0.916	0.929	0.970	0.944

---

PSNR								
Method	ship	drums	figus	hotdog	lego	materials	mic	chair
3DGS [4]	20.23	18.72	19.97	24.90	19.00	18.75	20.24	23.25
Scaffold-GS [7]	17.03	16.44	17.81	23.03	16.24	15.78	17.47	21.51
Pixel-GS [12]	22.97	19.57	21.04	23.89	20.20	22.01	25.53	20.51
Mip-Splatting [10]	26.37	23.79	26.36	32.01	27.11	26.12	29.15	30.15
Scaffold-Ours	26.62	23.82	25.91	33.22	27.75	27.20	31.04	30.71
3DGS-Ours	27.41	24.24	26.52	33.69	27.87	26.80	31.79	30.90

---

LPIPS								
Method	ship	drums	figus	hotdog	lego	materials	mic	chair
3DGS [4]	0.203	0.131	0.081	0.095	0.155	0.125	0.085	0.093
Scaffold-GS [7]	0.222	0.179	0.121	0.099	0.171	0.164	0.100	0.106
Pixel-GS [12]	0.194	0.131	0.086	0.100	0.164	0.123	0.079	0.093
Mip-Splatting [10]	0.173	0.110	0.080	0.073	0.116	0.082	0.061	0.079
Scaffold-Ours	0.174	0.104	0.088	0.055	0.100	0.077	0.043	0.066
3DGS-Ours	0.154	0.090	0.077	0.049	0.098	0.072	0.036	0.063

Table 5. Per-scene quantitative zoom-in results on NeRF Synthetic dataset [8].



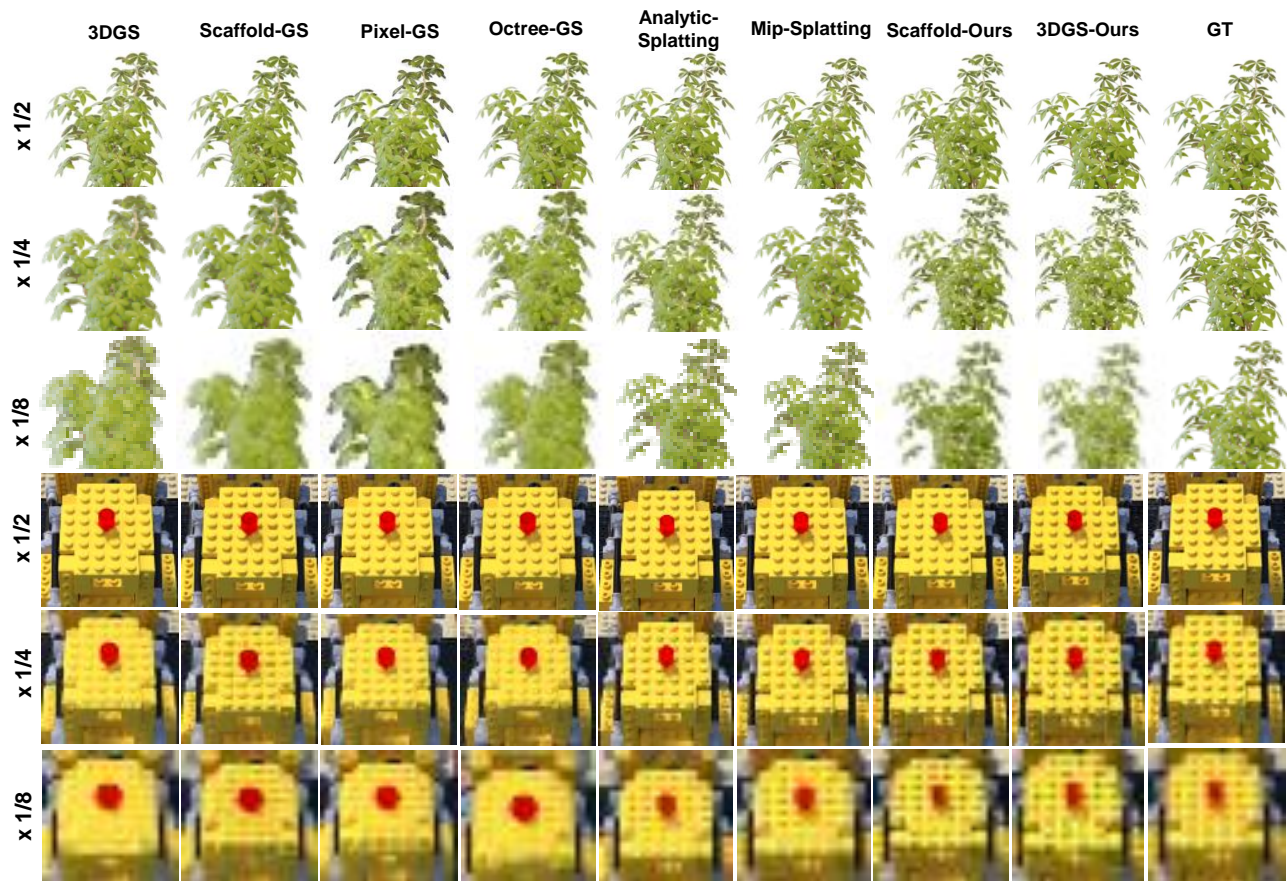


Figure 4. Comparison of zoom-out results on NeRF Synthetic dataset [8].



Figure 5. Comparison of simple dilation regularization on bicycle scene.

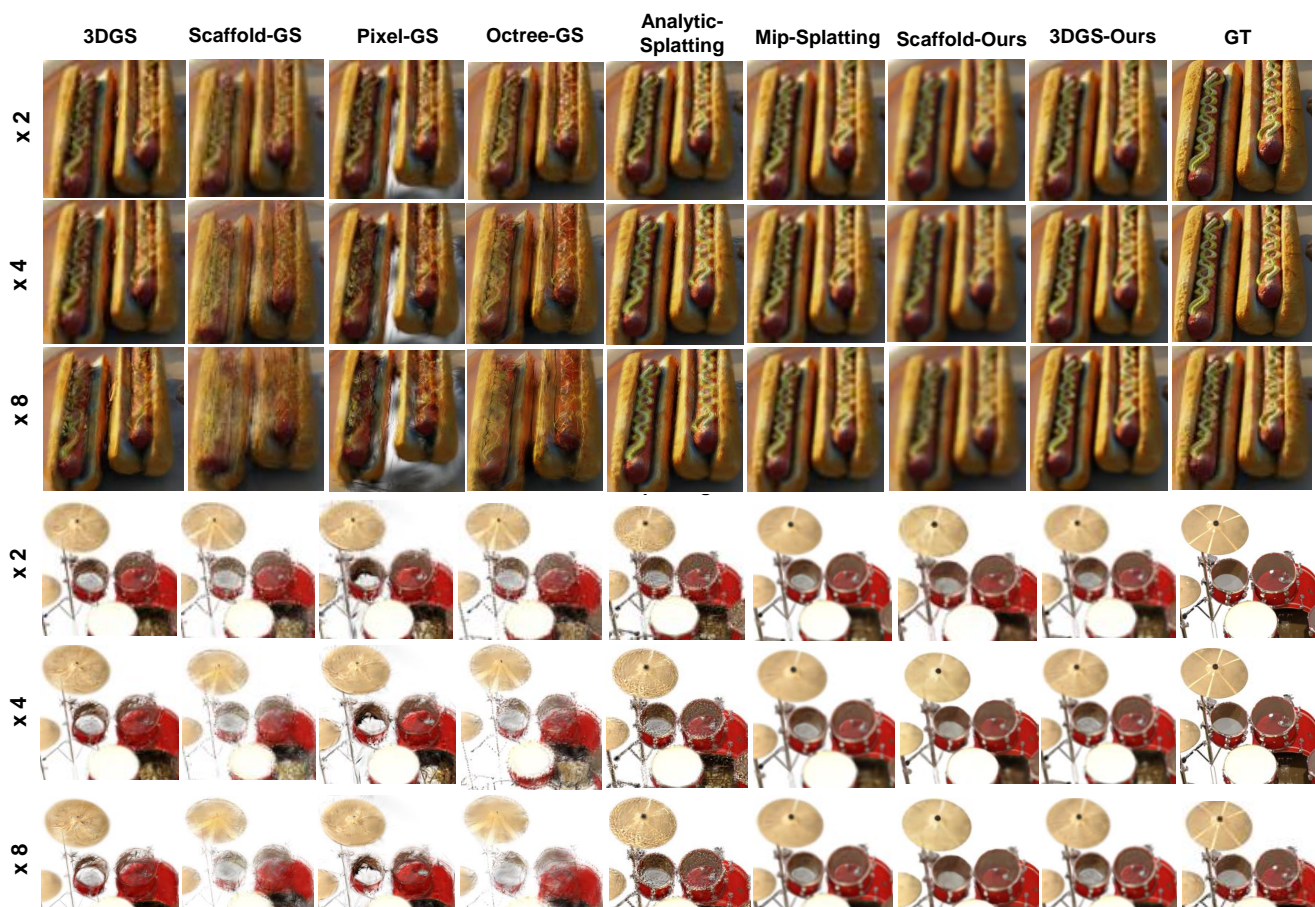


Figure 6. Comparison of zoom-in results on NeRF Synthetic dataset [8].

## References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022. 2, 3
- [2] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, pages 5972–5981, 2022. 2, 3
- [3] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *ICML*, 2024. 1
- [4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM TOG*, 2023. 1, 3, 4
- [5] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *CVPR*, pages 1833–1844, 2021. 2
- [6] Zhihao Liang, Qi Zhang, Wenbo Hu, Ying Feng, Lei Zhu, and Kui Jia. Analytic-splatting: Anti-aliased 3d gaussian splatting via analytic integration. In *ECCV*, 2024. 1
- [7] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *CVPR*, 2024. 1, 4
- [8] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 4, 5, 6
- [9] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 1
- [10] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *CVPR*, 2024. 1, 3, 4
- [11] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 1
- [12] Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *ECCV*, 2024. 1, 4