Analyzing the Internals of Neural Radiance Fields
Supplementary Material

Lukas Radl Andreas Kurz Michael Steiner Markus Steinberger
{lukas.radl, andreas.kurz, michael.steiner, steinberger}@icg.tugraz.at
Graz University of Technology, Austria

1. Additional Implementation Details for nerfacto†

Blender. For the Blender dataset [5], we set $t_n = 2.0$, $t_f = 6.0$ for all scenes. We disable the camera optimizer, the appearance embedding, the distortion loss and the scene contraction. We manually set the background color to white. For all scenes, we use a batch size of 2048 rays during training and use a uniform initial proposal network sampler. We train nerfacto† for 16.5×10^3 epochs, as in [7].

LLFF. For the LLFF dataset [4], we disable the camera optimizer, the appearance embedding and the distortion loss. We use a ℓ_∞ scene contraction as default for nerfacto [7], contrary to the use of NDC coordinates in NeRF and Mip-NeRF. We do not use our custom LLFF dataparser, we instead opt for the nerfstudio dataparser, which requires preprocessing the data. For all scenes, we use a batch size of 1024 rays during training and a batch size of 2048 during evaluation. We train nerfacto† for 30×10^3 epochs, as in [7]. For the *leaves* scenes, we use a uniform initial proposal network sampler, as the piecewise-uniform sampler did not result in stable optimization.

Mip-NeRF 360 and Deep Blending. For Mip-NeRF 360 [2] and Deep Blending [3], we use the Nerfstudio data processing pipeline [7] to extract the dataset using the given SfM output. As we use an MLP with NeRFs positional encoding [5], the hyperparameters reported in [7] do not necessarily provide the best results. We experimented with different hyperparameters and ultimately used the configuration reported in the main paper, as it resulted in the best performance for nerfacto†. Also, note that the results for Mip-NeRF 360 reported in [7] do not include the scenes *flowers* and *treehill*, for which nerfacto† exhibits the worst image metrics (c.f. Sec. 3).

Comparison with nerfacto. To investigate the effect of the MLP introduced for nerfacto†, we compare the results with nerfacto (with *HashMLPDensityField*) in Tab. 6. We use the Mip-NeRF 360 dataset [2] and use the same hyperparameter settings as reported in the main paper. As we can see, both variants achieve comparable results, although nerfacto is significantly faster.

Method	# Layers	PSNR ↑	SSIM ↑	LPIPS ↓
nerfacto†	2	24.201	0.726	0.247
	3	24.272	0.732	0.237
	4	24.201	0.732	0.237
	5	24.176	0.729	0.240
	6	24.247	0.732	0.234
nerfacto		24.682	0.735	0.229

Table 6. Comparison of nerfacto† (with MLP) and nerfacto (with *HashMLPDensityField*) for the Mip-NeRF 360 dataset [2].

2. Ablation Studies

First, we present an extensions to our method, which reduces processing time even further. In addition, we also report more detailed results for the ablation studies in Sec. **??**.

Figure 8. **Example renderings** from our our method (with ReLU) compared to without ReLU for the the Mip-NeRF 360 dataset [2]. Not using the ReLU results in artefacts for various renderings due to not being able to capture fine geometry.

Method	Blender, 800 × 800			LLFF, 1008 × 756		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
NeRF						
Base	29.05	0.93	0.07	25.12	0.74	0.24
Upsampling	28.90	0.93	0.08	24.60	0.73	0.26
NeRF	29.87	0.94	0.06	25.89	0.78	0.18
Mip-NeRF						
Base	29.04	0.93	0.07	25.44	0.75	0.22
Upsampling	28.94	0.93	0.08	24.90	0.74	0.25
Mip-NeRF	29.69	0.94	0.06	25.99	0.78	0.18

Table 7. **Ablation study**: We can further improve the performance of our implementation by obtaining the activations at a lower resolution. We use the Blender [5] and LLFF [4] datasets and use $\ell = 2$ and f_2, the configuration which performed best in our experiments for these datasets.

2.1. Upsampling Activations

As our approach builds on the intermediate activations for a proposal of density along a ray, a natural extension to our method is to obtain the activations $\mathbf{A}^{(\ell)}$ at a lower resolution. To this end, we perform an ablation study, where we record $\mathbf{A}^{(\ell)}$ at a resolution of $\frac{w}{2} \times \frac{h}{2} \times \frac{N_c}{2} \times N_h$, then perform upsampling to obtain a full-resolution activation $\mathbf{A}^{(\ell)}$. As our interpolation method, we choose nearest neighbor upsampling due to its negligible cost. Similarly to this approach, DONeRF [6] filters along the depth axis and across neighboring rays to obtain a smoothed depth target for their oracle network. For the Blender dataset [5] and the LLFF dataset [4], we use our method with $\ell = 1$ and f_2 and report results in Tab. 7. We perform this experiments for NeRF [5] and Mip-NeRF [1]: As we can see, this modifications impacts visual quality only slightly.

2.2. Impact of Capacity

We provide the full results for our capacity experiment from the main material in Tab. 8. As we can see, layer $\ell = 1, f_2$ is the best configuration for our method across various scenarios. This further indicates that capacity has minimal impact on our approach.

2.3. ReLU or no ReLU

We provide two example renderings comparing our method with the variant without the ReLU in Fig. 8. As can clearly be seen, not using the ReLU leads to more artefacts, particularly around sharp edges, causing the drop in all image metrics. This is caused by the more inconsistent histograms and the larger standard deviation.

		Reduced Capacity: 2 Layers				Reduced Capacity: 3 Layers			
		PSNR	SSIM	LPIPS	Speedup	PSNR	SSIM	LPIPS	Speedup
1	f_1	23.66	0.703	0.267		23.20	0.696	0.272	
	f_2	23.69	0.704	0.267	-	23.31	0.700	0.268	44%
	f_3	23.66	0.703	0.269		23.20	0.697	0.272	
2	f_1					22.63	0.677	0.299	
	f_2					22.56	0.676	0.299	-
	f_3					22.71	0.676	0.306	
nerfacto†		24.20	0.726	0.247	-	24.27	0.732	0.237	-

		Increased Capacity: 5 Layers				Increased Capacity: 6 Layers			
		PSNR	SSIM	LPIPS	Speedup	PSNR	SSIM	LPIPS	Speedup
1	f_1	22.83	0.689	0.279		22.73	0.686	0.282	
	f_2	22.94	0.692	0.276	101%	22.89	0.690	0.278	120%
	f_3	22.83	0.688	0.280		22.75	0.686	0.283	
2	f_1	22.82	0.688	0.280		22.67	0.683	0.287	
	f_2	22.84	0.688	0.281	50%	22.69	0.683	0.289	69%
	f_3	22.78	0.686	0.284		22.62	0.680	0.294	
3	f_1	22.70	0.684	0.287		22.64	0.683	0.289	
	f_2	22.60	0.682	0.291	20%	22.58	0.680	0.294	37%
	f_3	22.64	0.680	0.295		22.47	0.675	0.303	
4	f_1	22.54	0.677	0.299		22.52	0.678	0.298	
	f_2	22.28	0.670	0.305	-	22.31	0.672	0.304	15%
	f_3	22.62	0.673	0.308		22.41	0.671	0.310	
5	f_1					22.45	0.673	0.305	
	f_2					22.18	0.665	0.317	-
	f_3					22.55	0.673	0.312	
nerfacto†		24.18	0.729	0.240	-	24.25	0.732	0.234	-

Table 8. **Detailed Results** for our capacity ablation study with the Mip-NeRF 360 dataset [2]. In every configuration, our method performs best at $\ell = 1, f_2$. We color code best, second-best and third-best for each scene.

3. Per-Scene Results

In Tabs. 9 and 10, we show per-scene results for the Blender dataset [5] and the LLFF dataset [4]. For the *drums* scene, we manage to outperform Mip-NeRF quantitatively. In Tabs. 11 and 12, we also report the per-scene results for Mip-NeRF 360 [2] and Deep Blending [3].

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. In *ICCV*, 2021. 2

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *CVPR*, 2022. 1, 2, 4

[3] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep Blending for Free-viewpoint Image-based Rendering. *ACM TOG*, 37(6), 2018. 1, 2, 4

[4] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local Light Field Fusion: Practical View Syn-

ℓ	Fct.	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
NeRF									
1	f_1	30.10	22.33	26.21	32.41	26.79	28.05	29.10	24.86
	f_2	30.12	23.30	28.25	33.40	28.99	28.58	30.53	26.30
	f_3	30.43	23.23	28.14	33.26	28.75	28.61	30.48	26.16
2	f_1	30.66	23.57	25.83	33.17	28.54	28.87	31.27	22.04
	f_2	30.19	23.63	28.45	33.61	30.03	28.73	30.97	26.79
	f_3	30.69	23.91	29.03	33.86	30.62	29.01	31.37	27.03
3	f_1	27.93	20.26	21.74	28.10	20.56	26.13	28.15	20.48
	f_2	28.98	20.46	22.42	32.97	27.62	26.18	25.28	25.91
	f_3	27.93	19.62	21.29	32.15	26.14	24.17	23.59	25.71
NeRF		31.16	24.09	29.26	34.33	31.34	29.08	31.75	27.92
Mip-NeRF									
1	f_1	30.14	22.33	26.10	33.06	27.52	28.21	28.54	24.44
	f_2	30.01	23.36	27.60	33.64	29.83	28.60	30.25	26.69
	f_3	30.34	23.34	27.44	33.87	29.70	28.61	29.93	26.56
2	f_1	30.77	23.91	27.46	31.67	28.97	27.97	30.99	11.65
	f_2	30.13	23.66	28.07	33.57	30.38	28.62	30.97	26.89
	f_3	30.65	23.95	28.39	33.72	30.82	28.83	31.30	27.16
3	f_1	28.77	22.31	25.96	17.52	23.23	27.85	30.90	11.20
	f_2	29.19	22.68	25.16	33.60	29.64	27.75	25.06	26.66
	f_3	28.07	21.86	23.22	33.44	29.64	26.27	21.98	27.10
Mip-NeRF		31.13	23.82	28.52	34.41	31.07	29.31	31.54	27.71
nerfacto†									
1	f_1	28.30	20.80	22.06	31.48	25.89	24.19	27.49	25.14
	f_2	29.07	21.17	22.36	31.98	26.89	24.35	28.19	25.56
	f_3	29.07	21.18	22.21	32.03	26.91	24.30	28.02	25.66
2	f_1	29.16	20.89	22.08	31.19	27.26	24.61	29.18	25.57
	f_2	29.35	21.36	22.17	31.46	28.16	24.56	29.45	25.93
	f_3	28.56	21.24	22.04	31.18	27.40	24.66	29.51	25.92
3	f_1	27.96	20.59	22.02	31.00	28.50	24.64	29.52	25.04
	f_2	28.01	20.64	22.00	31.03	25.24	24.36	28.16	25.16
	f_3	27.95	20.60	22.02	31.01	22.69	23.53	25.55	24.97
nerfacto†		30.64	22.11	22.86	32.67	30.65	24.76	29.78	27.38

Table 9. **Per-Scene PSNR** results for the LLFF dataset [4]. We color code best, second-best and third-best for each scene. In addition, we highlight failures of our method.

thesis with Prescriptive Sampling Guidelines. *ACM TOG*, 38 (4), 2019. 1, 2, 3

[5] 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, 2, 3

[6] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Comput. Graph. Forum*, 40(4): 45–59, 2021. 2

[7] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *SIGGRAPH*, 2023. 1

ℓ	Fct.	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
NeRF									
1	f_1	23.83	27.61	28.12	25.11	21.38	19.35	30.32	25.16
	f_2	24.19	28.03	28.27	25.52	21.35	19.57	30.81	25.44
	f_3	24.07	27.97	28.22	25.42	21.38	19.48	30.73	25.43
2	f_1	23.76	27.68	27.95	24.84	21.08	19.48	28.66	24.32
	f_2	24.03	27.95	28.17	25.26	21.23	19.62	29.44	25.30
	f_3	23.92	27.86	28.09	25.14	21.19	19.56	29.13	25.21
3	f_1	23.61	27.65	27.74	24.34	20.97	19.50	28.18	21.73
	f_2	23.92	27.95	28.06	24.96	21.17	19.65	28.94	25.26
	f_3	23.82	27.88	27.95	24.79	21.12	19.61	28.66	25.17
NeRF		24.70	28.61	29.07	26.05	21.56	19.95	31.49	25.72
Mip-NeRF									
1	f_1	23.56	27.73	28.04	24.79	21.38	19.34	30.57	25.16
	f_2	23.93	28.26	28.50	25.51	21.37	19.59	30.89	25.56
	f_3	23.80	28.16	28.32	25.35	21.39	19.50	30.79	25.46
2	f_1	23.41	27.88	28.13	24.89	21.23	19.41	30.47	24.72
	f_2	23.72	28.33	28.52	25.49	21.34	19.60	30.92	25.58
	f_3	23.60	28.29	28.37	25.35	21.34	19.53	30.78	25.51
3	f_1	23.51	25.69	28.31	23.68	20.74	18.75	30.14	15.30
	f_2	23.83	28.37	28.72	25.52	21.32	19.67	30.90	25.61
	f_3	23.71	28.34	28.60	25.37	21.31	19.61	30.81	25.57
Mip-NeRF		24.54	28.69	29.47	26.13	21.59	20.01	31.59	25.92
nerfacto†									
1	f_1	23.52	26.60	27.59	25.18	17.98	19.03	29.81	25.95
	f_2	23.72	26.78	27.70	26.44	17.86	19.06	30.09	25.88
	f_3	23.10	26.30	27.22	24.88	18.14	18.92	29.45	25.06
2	f_1	22.32	26.06	26.84	25.97	18.61	18.57	30.20	25.07
	f_2	22.98	26.64	27.53	26.65	19.25	18.99	30.28	26.34
	f_3	21.97	25.21	26.45	23.85	18.29	18.33	29.98	24.03
3	f_1	22.44	25.83	27.23	23.65	18.34	17.88	28.03	22.28
	f_2	22.06	25.96	26.76	24.58	18.46	17.77	28.81	25.87
	f_3	22.13	25.77	27.05	23.51	18.32	17.40	27.62	22.49
nerfacto†		23.83	27.01	27.94	26.91	21.28	19.11	30.36	26.39

Table 10. **Per-Scene PSNR** results for the Blender dataset [5]. We color code best, second-best and third-best for each scene. In addition, we highlight failures of our method.

		Outdoor Scenes				
ℓ	Fct.	Bicycle	Flowers	Garden	Stump	Treehill
1	f_1	20.47	20.15	23.69	22.88	17.55
	f_2	20.70	20.17	23.75	22.98	17.55
	f_3	20.51	20.14	23.69	22.90	17.54
2	f_1	20.45	20.13	23.58	22.89	17.35
	f_2	20.59	20.03	23.45	22.97	17.33
	f_3	20.41	20.12	23.61	22.86	17.46
3	f_1	24.52	19.98	23.48	22.71	17.47
	f_2	20.03	19.94	23.20	22.56	17.42
	f_3	20.12	19.89	23.54	22.70	17.48
nerfacto†		22.19	20.81	24.87	24.03	17.88
			Indoor Scenes			
			Bonsai	Counter	Kitchen	Room
1	f_1		25.43	24.06	24.29	26.66
	f_2		25.65	24.20	24.54	26.79
	f_3		25.40	24.00	24.32	26.73
2	f_1		25.47	23.96	23.97	26.56
	f_2		25.41	23.99	23.68	26.47
	f_3		25.15	23.94	23.93	26.52
3	f_1		24.52	23.76	23.82	26.03
	f_2		24.36	23.64	23.15	25.63
	f_3		24.29	23.90	23.63	25.96
nerfacto†			27.55	25.36	26.76	28.37

Table 11. **Per-Scene PSNR** results for the Mip-NeRF 360 dataset [2]. We color code best, second-best and third-best for each scene.

		Dr Johnson			Playroom		
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
1	f_1	28.39	0.885	0.195	27.63	0.831	0.280
	f_2	28.39	0.885	0.196	27.64	0.833	0.279
	f_3	28.42	0.885	0.194	27.55	0.830	0.284
2	f_1	28.46	0.886	0.194	27.49	0.830	0.284
	f_2	28.47	0.886	0.194	27.44	0.827	0.291
	f_3	28.38	0.885	0.196	27.49	0.828	0.289
3	f_1	28.23	0.881	0.202	27.30	0.818	0.310
	f_2	28.03	0.879	0.205	26.93	0.809	0.322
	f_3	28.18	0.877	0.209	27.38	0.817	0.315
nerfacto†		29.65	0.903	0.171	28.87	0.856	0.242

Table 12. **Full Per-Scene Results** for the Deep Blending dataset [3]. We color code best, second-best and third-best for each scene.