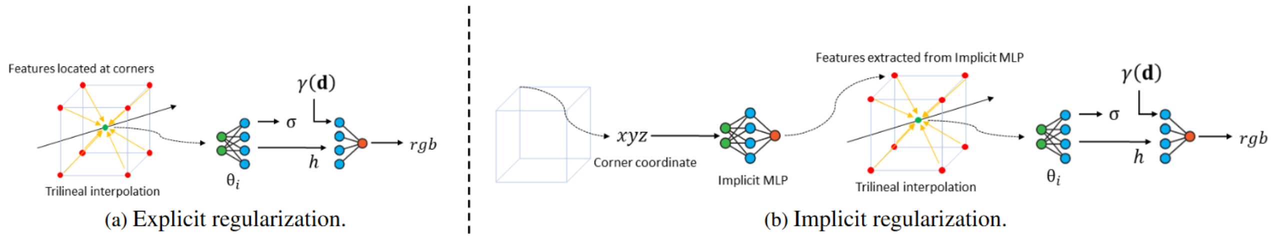
NeRFLight Supplemental Materials



(a) Explicit regularization.

(b) Implicit regularization.

Figure 7. **Grid regularizations** Features can be obtained either using an auto-decoder architecture (a) where the features are directly optimized or using a auto-encoder architecture (b) where an implicit MLP is optimized to obtained the features depending on their spatial position.

In this section, we provide additional details about the architecture of the MLPs used in our work together with a more detailed description of the feature optimization methods applied (explicit and implicit regularization). We also report more detailed per-scene quantitative results and additional qualitative results.

A. Additional MLP architecture details

Fig. 7 shows the two different training strategies that are used to train NeRFLight. In the coarse stage, features are initialized randomly and optimized using explicit regularization. However, at the fine stage, the features are first optimized using an auto-encoder architecture, where an implicit MLP is used to obtain a feature value depending on its spatial location. After 1000 epochs, the features for each of the corners in the grid are extracted. These values are used as initialization for an auto-decoder architecture, where features are, once again, directly optimized.

Fig. 8 shows the architecture of the implicit MLP used for implicit regularization. It is composed of one input layer and eight hidden layers of 512 neurons activated using ReLU and an output layer of 32 neurons without activation. Positional encoding is applied to the corner coordinates. We use ten frequencies for this positional encoding.

Fig. 9 shows the architecture of a density decoder and a color decoder. Each of the density decoders is composed of one input layer, one hidden layer, and one output layer. They take the integrated feature (dimension 32) as input and output the density value σ and an intermediate representation h. Each of the layers is activated with ReLU activation except for the neurons of the last layer that correspond to the intermediate representation, where no activation is used. This intermediate representation is concatenated with the positional encoding of the by direction to produce the input of the color decoder. Finally, the color decoder, composed of a ReLU-activated input layer of 32 neurons and a sigmoid-activated output layer of 3 neurons, obtains the color of the given interval. Four frequencies are used for the direction positional encoding.

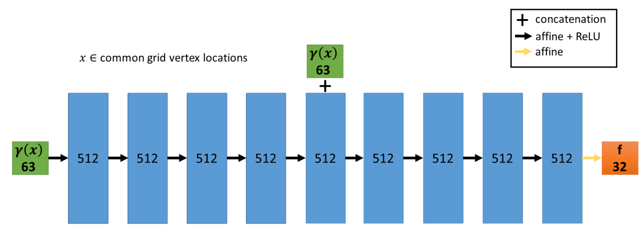


Figure 8. **Implicit MLP architecture.** The implicit MLP is used to obtain a robust initialization of the feture grid. It is evaluated on the corners of the common feature grid.

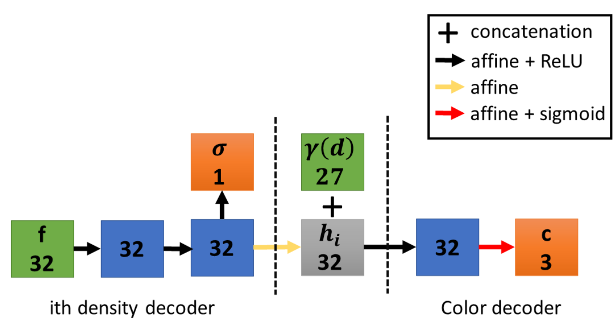
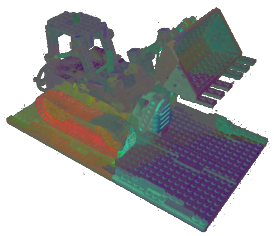


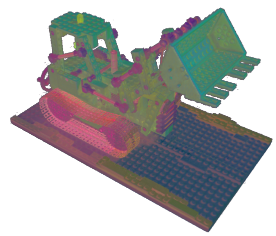
Figure 9. **Density and color decoders.** Depending on the location of the voxel, one of the N density decoders obtains the density value at the given voxel and an intermediate representation from the integrated feature. Then, the color decoder used the intermediate representation and the direction positional encoding to obtain the color value.

B. Additional Results

Fig. 10 shows the same experiment as Fig. 5 but using a Lego scene rotated $(14°, 28°, 64°)$ on the X-Y-Z axes. The goal of this experiment is to show that our approach is also



(a) Rotated Linear grid



(b) Rotated Symmetric grid

Figure 10. **PCA decomposition** of density decoder's outputs of the rotated Lego Scene. The symmetric grid shows to be more stable.

Orientation	PSNR ↑	SSIM ↑	LPIPS ↓
Axis aligned	31.41	0.968	0.039
Rotated	31.32	0.967	0.039

Table 3. Results on Lego scene with and w/o rotation.

robust for the case where the symmetry axes of the scenes are not aligned with the axes of the scene (as occurs in some scenes of the NeRF Synthetic dataset. Tab. 3 shows that the metrics obtained by our model in both scenes (original and rotated) are similar. Tab. 4 shows the results on Lego Scene from the NeRF Synthetic dataset of NeRFLight using different regularizations. It can be seen how all of them (explicit, implicit, and hybrid) obtain high-quality results. While explicit regularization obtains the lowest quality metrics, it requires less time and GPU memory. For the lego scene, the training time was 16 hours and the peak GPU memory usage was 22GB, a significant reduction compared with the 60 hours and the 40GB needed by the hybrid regularization.

In Tab. 5, we show a per-scene comparison of the rendering quality between NeRFLight and all the baselines we compared with in NeRF synthetic dataset. In Tab. 6, we do the same for the Tanks and Temples dataset. We also provide the model size and frame rate for NeRF synthetic dataset (Tab. 7) and the model size for the Tanks and Temples dataset Tab. 8. In these results, we also add the DIVeR-64 model [48] for NeRF Synthetic dataset, while the tables shown in the paper only reports the results from DIVeR-32 since it is the model of [48] with the best FPS/MB ratio, while the quality metrics are really similar between both models.

As explained in the paper, in Tab. 1 we provide the results of Instant-NGP for Tanks and Temples dataset tuning its hyperparameters so that the model size matches NeRFlight. We use a hash table size of 2^{22} and the finest resolution is set to 256. In Tab. 6 also shows the results of a larger model of Instant-NGP using the same size for the hash table and the finest resolution of 512. It is marked as (l) while the Instant-NGP shown in the paper is marked as (s).

Reg.	PSNR ↑	SSIM ↑	LPIPS ↓
Ex	33.02	0.975	0.016
Im	33.24	0.979	0.013
Im-Ex	**33.84**	**0.985**	**0.011**

Table 4. Results on Lego scene using different regularizations

Finally, we also provide additional qualitative results in Fig. 12 and Fig. 11.

PSNR ↑									
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean
NeRF*	33.00	25.01	30.13	36.18	32.54	29.62	32.91	28.65	31.00
JAXNeRF+*	35.35	25.65	32.77	37.58	35.35	30.29	36.52	30.48	33.00
JAXNeRF*	33.88	25.08	30.51	36.91	33.24	30.03	34.52	29.07	31.65
AutoInt*	25.60	20.78	22.47	32.33	25.09	25.90	28.10	24.15	25.55
SNeRG(PNG)	33.24	24.57	29.32	34.33	33.82	27.21	32.60	27.97	30.38
SNeRG(H264)	-	-	-	-	-	-	-	-	29.86
Eff-NeRF	-	-	-	-	-	-	-	-	31.68
KiloNeRF	32.91	25.25	29.76	35.56	33.02	29.20	33.06	29.23	31.00
Plenoctrees	34.66	25.31	30.79	36.79	32.95	29.76	33.97	29.42	31.71
TensorRF-CP	33.60	25.17	30.72	36.24	34.05	30.10	33.77	28.84	31.56
TensorRF-VM	35.76	26.01	33.99	37.41	36.46	30.12	34.51	30.77	*33.14*
Instant-NGP	35.00	26.02	33.51	37.40	36.39	29.78	36.22	31.10	**33.18**
DIVeR-64	34.35	25.38	31.76	36.76	35.49	29.61	34.57	30.48	32.30
DIVeR-32	34.09	25.40	32.02	36.35	35.17	29.24	34.53	30.14	32.12
NeRFLight	32.76	25.60	31.65	35.96	33.84	29.00	33.1	29.38	31.41

SSIM ↑									
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean
NeRF*	0.967	0.925	0.964	0.974	0.961	0.952	0.987	0.865	0.947
JAXNeRF+*	0.982	0.936	0.980	0.983	0.979	0.956	0.991	0.887	0.962
JAXNeRF*	0.974	0.927	0.967	0.979	0.968	0.952	0.987	0.865	0.952
AutoInt*	0.928	0.861	0.898	0.974	0.900	0.930	0.948	0.852	0.911
SNeRG(PNG)	0.975	0.929	0.967	0.971	0.973	0.938	0.982	0.865	0.950
SNeRG(H264)	-	-	-	-	-	-	-	-	0.938
Eff-NeRF	-	-	-	-	-	-	-	-	0.954
KiloNeRF	0.970	0.930	0.970	0.980	0.9700	0.950	0.980	0.880	0.950
Plenoctrees	0.981	0.933	0.970	0.982	0.971	0.955	0.987	0.884	0.958
TensorRF-CP	0.973	0.921	0.965	0.975	0.971	0.950	0.983	0.857	0.949
TensorRF-VM	0.985	0.937	0.982	0.982	0.983	0.952	0.988	0.895	0.963
Instant-NGP	0.987	0.950	0.987	0.989	0.988	0.964	0.991	0.938	**0.974**
DIVeR-64	-	-	-	-	-	-	-	-	-
DIVeR-32	0.977	0.932	0.977	0.978	0.978	0.946	0.987	0.885	0.958
NeRFLight	0.978	0.954	0.983	0.990	0.985	0.949	0.989	0.927	*0.968*

LPIPS ↓									
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean
NeRF*	0.046	0.091	0.044	0.121	0.050	0.063	0.028	0.206	0.081
JAXNeRF+*	0.017	0.057	0.018	0.022	0.017	0.041	0.008	0.123	0.038
JAXNeRF*	0.027	0.070	0.033	0.030	0.030	0.048	0.013	0.156	0.051
AutoInt*	0.141	0.224	0.148	0.080	0.175	0.136	0.131	0.323	0.170
SNeRG(PNG)	0.025	0.061	0.028	0.043	0.022	0.052	0.016	0.156	0.050
SNeRG(H264)	-	-	-	-	-	-	-	-	0.065
Eff-NeRF	-	-	-	-	-	-	-	-	**0.020**
KiloNeRF	0.020	0.050	0.020	0.020	0.020	0.020	0.010	0.080	*0.030*
Plenoctrees	0.022	0.076	0.038	0.032	0.034	0.059	0.017	0.144	0.053
TensorRF-CP	0.044	0.114	0.058	0.052	0.038	0.068	0.035	0.196	0.076
TensorRF-VM	0.022	0.073	0.022	0.032	0.018	0.058	0.015	0.138	0.047
Instant-NGP	0.012	0.08	0.053	0.013	0.014	0.063	0.023	0.086	0.043
DIVeR-64	-	-	-	-	-	-	-	-	-
DIVeR-32	0.014	0.058	0.020	0.019	0.010	0.034	0.011	0.100	0.033
NeRFLight	0.025	0.073	0.036	0.010	0.011	0.045	0.030	0.08	0.039

Table 5. **Rendering quality on NeRF synthetic dataset**

	PSNR ↑					
	Barn	Caterpillar	Family	Ignatius	Truck	Mean
NeRF	27.71	25.87	33.33	27.79	26.92	28.32
KiloNeRF	27.81	25.61	33.65	27.92	27.04	28.41
Plenoctrees	26.80	25.29	32.85	28.19	26.83	27.99
TensoRF-CP	26.74	24.73	32.39	27.86	26.25	27.59
TensoRF-VM	27.22	26.19	33.92	28.34	27.14	*28.56*
Instant-NGP(s)	26.51	24.9	32.98	27.63	25.49	27.50
Instant-NGP(l)	27.41	26.61	34.68	28.23	27.92	**28.97**
DIVeR-32	26.7	25.8	33.34	27.42	27.81	28.21
NeRFLight	26.42	25.3	33.23	27.24	27.22	27.85

	SSIM ↑					
	Barn	Caterpillar	Family	Ignatius	Truck	Mean
NeRF	0.85	0.89	0.95	0.94	0.89	0.90
KiloNeRF	0.85	0.90	0.96	0.94	0.90	0.910
Plenoctrees	0.856	0.907	0.962	0.948	0.914	0.917
TensoRF-CP	0.839	0.879	0.948	0.934	0.885	0.897
TensoRF-VM	0.864	0.912	0.965	0.948	0.914	0.920
Instant-NGP(s)	0.897	0.924	0.968	0.963	0.921	0.935
Instant-NGP(l)	0.910	0.940	0.980	0.970	0.942	**0.948**
DIVeR-32	0.843	0.900	0.954	0.932	0.900	0.906
NeRFLight	0.891	0.927	0.976	0.958	0.941	*0.939*

	LPIPS ↓					
	Barn	Caterpillar	Family	Ignatius	Truck	Mean
NeRF	0.19	0.12	0.05	0.08	0.11	0.11
KiloNeRF	0.16	0.10	0.04	0.06	0.10	0.090
Plenoctrees	0.226	0.148	0.069	0.080	0.130	0.131
TensoRF-CP	0.237	0.176	0.063	0.089	0.154	0.144
TensoRF-VM	0.217	0.139	0.057	0.081	0.129	0.125
Instant-NGP(s)	0.151	0.056	0.036	0.075	0.060	*0.076*
Instant-NGP(l)	0.142	0.051	0.034	0.064	0.053	**0.069**
DIVeR-32	0.142	0.095	0.041	0.054	0.079	0.082
NeRFLight	0.139	0.100	0.043	0.055	0.083	0.084

Table 6. **Rendering quality on the Tanks and Temples dataset**

	MB↓								
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean
NeRF*	5	5	5	5	5	5	5	5	5
JAXNeRF+*	18	18	18	18	18	18	18	18	18
JAXNeRF*	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	*4.8*
AutoInt*	5	5	5	5	5	5	5	5	5
SNeRG(PNG)	-	-	-	-	-	-	-	-	84
SNeRG(H264)	-	-	-	-	-	-	-	-	30.2
Eff-NeRF	-	-	-	-	-	-	-	-	648
KiloNeRF	204	-	-	-	108	-	-	173	161
Plenoctrees	832	1239	1792	2683	2068	3686	443	2693	1930
TensorRF-CP	3.71	3.78	3.73	4.23	3.74	4.49	3.83	3.87	**3.9**
TensorRF-VM	66.7	69.9	68.6	83.8	81.5	67.8	70.2	68.1	71.8
Instant-NGP	27	27	27	27	27	27	27	27	27
DIVeR-64	55	42	49	80	64	62	24	118	62
DIVeR-32	55	56	47	84	64	62	24	151	68
NeRFLight	9	15	9	16	13	11	10	30	14

	FPS↑								
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean
NeRF*	-	-	-	-	-	-	-	-	0.03
JAXNeRF+*	-	-	-	-	-	-	-	-	0.01
JAXNeRF*	-	-	-	-	-	-	-	-	0.05
AutoInt*	-	-	-	-	-	-	-	-	0.38
SNeRG(PNG)	-	-	-	-	-	-	-	-	*202*
SNeRG(H264)	-	-	-	-	-	-	-	-	*202*
Eff-NeRF	-	-	-	-	-	-	-	-	**238**
KiloNeRF	103	-	-	-	88	-	-	45	79
Plenoctrees	312	302	116	89	280	70	320	58	168
TensorRF-CP	0.66	0.752	0.15	0.21	0.24	0.42	0.69	0.49	0.45
TensorRF-VM	0.60	0.747	0.08	0.11	0.17	0.36	0.61	0.42	0.38
Instant-NGP	85	61	68	63	66	37	99	17	62
DIVeR-64	101	87	78	82	168	70	150	78	102
DIVeR-32	160	117	108	117	200	114	168	79	133
NeRFLight	223	252	149	130	215	121	201	153	181

	FPS/MB↑								
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean
NeRF*	-	-	-	-	-	-	-	-	0.006
JAXNeRF+*	-	-	-	-	-	-	-	-	0.0006
JAXNeRF*	-	-	-	-	-	-	-	-	0.0104
AutoInt*	-	-	-	-	-	-	-	-	0.076
SNeRG(PNG)	-	-	-	-	-	-	-	-	2.404
SNeRG(H264)	-	-	-	-	-	-	-	-	*6.689*
Eff-NeRF	-	-	-	-	-	-	-	-	0.367
KiloNeRF	0.505	-	-	-	0.815	-	-	0.260	0.490
Plenoctrees	0.375	0.244	0.065	0.033	0.135	0.019	0.722	0.022	0.085
TensorRF-CP	0.178	0.199	0.040	0.050	0.064	0.094	0.180	0.123	0.115
TensorRF-VM	0.009	0.011	0.0012	0.0013	0.002	0.005	0.009	0.006	0.0053
Instant-NGP	3.15	2.26	2.52	2.33	2.44	1.37	3.67	0.63	2.296
DIVeR-64	1.84	2.07	1.59	1.025	2.625	1.129	6.25	0.661	1.645
DIVeR-32	2.91	2.09	2.30	1.393	3.125	1.84	7.00	0.52	1.956
NeRFLight	24.78	16.8	16.56	8.125	16.54	11	20.1	5.1	**12.929**

Table 7. **Model size and rendering speed on NeRF synthetic dataset**

	MB↓					
	Barn	Caterpillar	Family	Ignatius	Truck	Mean
NeRF	5	5	5	5	5	_5_
KiloNeRF	-	-	-	-	-	-
Plenoctrees	2509	2365	2928	3031	2314	2629
TensorRF-CP	4.2	3.93	4.07	4.32	4.64	**4.23**
TensorRF-VM	71.5	67.4	66.7	80.7	70.6	71.4
Instant-NGP(s)	40	40	40	40	40	40
Instant-NGP(l)	116	116	116	116	116	116
DIVeR-32	85	92	133	154	117	116
NeRFLight	31	50	24	43	52	40

	FPS↑					
	Barn	Caterpillar	Family	Ignatius	Truck	Mean
NeRF*	-	-	-	-	-	0.005
KiloNeRF	-	-	-	-	-	41.3
Plenoctrees	23	37	58	51	42	42
TensorRF-CP	0.15	0.35	0.35	0.28	0.21	0.27
TensorRF-VM	0.02	0.24	0.22	0.18	0.16	0.16
Instant-NGP(s)	16	30	38	34	27	29
Instant-NGP(l)	16	30	38	34	27	29
DIVeR-32	31	56	70	66	48	_54_
NeRFLight	45	97	90	75	82	**78**

	FPS/MB↑					
	Barn	Caterpillar	Family	Ignatius	Truck	Mean
NeRF*	-	-	-	-	-	0.001
KiloNeRF	-	-	-	-	-	-
Plenoctrees	0.009	0.016	0.020	0.017	0.018	0.016
TensorRF-CP	0.036	0.089	0.086	0.065	0.045	0.064
TensorRF-VM	0.0003	0.004	0.003	0.002	0.002	0.0022
Instant-NGP(s)	0.4	0.75	0.95	0.85	0.675	_0.725_
Instant-NGP(l)	0.14	0.26	0.33	0.293	0.232	0.25
DIVeR-32	0.36	0.61	0.53	0.43	0.41	0.47
NeRFLight	1.45	1.94	3.75	1.74	1.58	**1.95**

Table 8. **Model size on Tanks and Temples dataset.**

Figure 11. **Additional qualitative results on Tanks and Temples**.

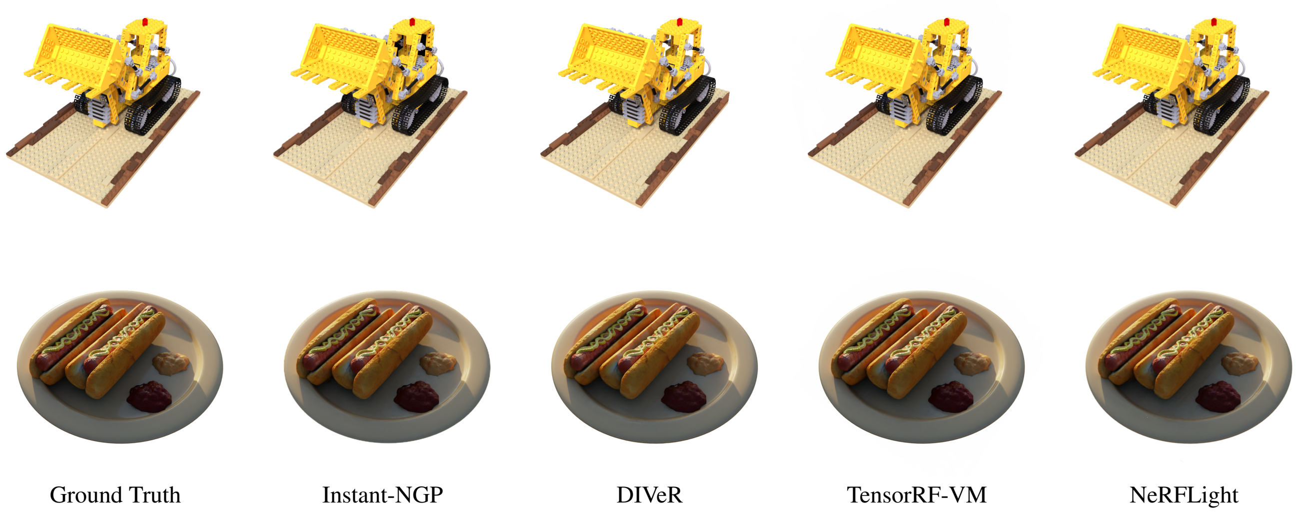


Figure 12. **Additional qualitative results on NeRF Synthetic**.