Masked Wavelet Representation for Compact Neural Radiance Fields: Supplementary Materials

Daniel Rho[1]* Byeonghyeon Lee[2]* Seungtae Nam[2] Joo Chan Lee[2] Jong Hwan Ko[2,3]†
Eunbyung Park[2,3]†
[1]AI2XL, KT
[2]Department of Artificial Intelligence, Sungkyunkwan University
[3]Department of Electrical and Computer Engineering, Sungkyunkwan University

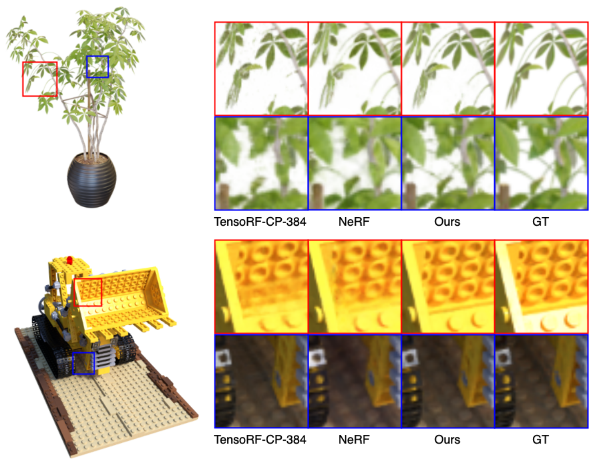

Figure 1. Qualitative analysis of similar-sized representations. Every method was quantized to 8 bits and less than 1 MB.

1. Qualitative Comparison

In this section, we visually demonstrate the qualities of neural fields with similar sizes. Fig. 1 shows the qualitative results of similar-sized representations: TensoRF-CP-384, NeRF, and Ours (λ_m=1e-10). The bit precision of each model was set to 8 bits. Even though they have similar overall sizes (less than 1 MB), ours shows the best qualitative results.

2. Computational Costs

We also present the computational costs, particularly in terms of time, because we introduced learnable masks and DWT for efficient scene representation. First of all, as we explained in the main paper, we can ignore the computational costs at inference time. However, those two components incur non-negligible training time during training time. Tab. 1 shows the approximate training time to train a network (VM-192). It was evaluated using a 40 GB memory-equipped Tesla A100. Introducing masks increases the training time, but by a relatively insignificant amount of time. On the other hand, DWT increases training time more significantly. We believe that the increased training time is due to two factors: additional computational costs caused by DWT and not fully optimized DWT codes. As a result, we believe future DWT code optimization can help close this training time gap, especially given that current codes lower the GPU utilization rate from 80% (baseline) to 50% (4-level DWT with mask). Nevertheless, the time required by our proposed method is still far less than that of neural fields that exclusively use MLP, such NeRF.

	No Mask	Mask
Spatial	$\approx$ 8 min	$\approx$ 9 min
1-level DWT	$\approx$ 13 min	$\approx$ 14 min
2-level DWT	$\approx$ 15 min	$\approx$ 17 min
3-level DWT	$\approx$ 18 min	$\approx$ 20 min
4-level DWT	$\approx$ 23 min	$\approx$ 24 min

Table 1. Approximate training time

3. Comparison on Masking Method

We also compared our masking method with a threshold-based pruning approach that removes grid coefficients whose absolute values are below a threshold τ. As Tab. 2 shows, ours is more parameter-efficient than the threshold-based pruning method. It further supports the effectiveness of our proposed masking method in improving efficiency.

*Equal contribution
†Corresponding authors

Methods	Sparsity	PSNR
Abs. Thres. (τ=0.5)	0.8554	32.33
Abs. Thres. (τ=1.0)	0.9385	28.95
Abs. Thres. (τ=1.6)	0.9747	21.40
Ours (λ_m=2.5e-11)	0.9376	32.24
Ours (λ_m=1.0e-10)	0.9769	31.95

Table 2. Pruning on NeRF Synthetic dataset. The threshold of the absolute value-based method (Abs. Thres.), is denoted as τ.

4. Ablation Studies on Mask Compression

In this section, we analyze our proposed compression pipeline in more detail by demonstrating the compression ratio at each stage and outlining the rationale for our design decisions. Tab. 3 shows the compression ratios in the NeRF dataset. By incrementing each step, we demonstrate how each step contributes to the compression ratio. Keep in mind that we do not compress non-zero coefficients; we only compress information regarding which coefficients are non-zero (mask). We keep the non-zero coefficients that are not compressed as well as the compressed mask that shows where the non-zero components are located.

Run-length encoding (RLE) The RLE is effective for compressing data with repeating numbers. Instead of encoding raw repeating numbers, RLE encodes repeating numbers as a pair of the number and its count. This is why we adopt RLE in our mask compression pipeline. Our proposed method zeros out most grid coefficients. More specifically, by adopting our proposed method, the sparsity of grids can go up to 90% or even 99%. As shown in the second row of Tab. 3, adopting RLE can reduce the mask size by a factor of 1.75, on average.

Huffman encoding As shown in the third row of Tab. 3, we can raise the compression ratio to 4.62 by including Huffman encoding in our compression pipeline. We also compared adaptive arithmetic coding [3] with Huffman encoding but did not observed meaningful improvements. As a result, we chose to use the less expensive Huffman encoding in our compression pipeline. We believe, however, that further improvements could be made by incorporating advanced compression techniques into the proposed method.

8-bit casting Packing the binarized mask values by 8 bits before applying RLE can further increase the compression ratio to 6.24 (Tab. 3). This is because the casting makes the length of the RLE code much shorter. For example, consider a thousand 0s. Without casting, we can represent a thousand 0s with three (0, 255) and one (0, 235). On the other hand, with the help of casting, we only need a pair of two numbers (0, 125). We use this method assuming most elements are zeros, and as shown in the table, it can really improve compression ratio.

Level-wise encoding We discovered that sparsity highly depends on the level of DWT, as shown in Tab. 5. More specifically, high-pass coefficients have higher sparsity. Based on the findings, we separate the mask at each level and compress separately. For the last level of DWT, we treat the upper left part (LL) and the remaining three parts (LH, HL, HH, also known as precincts) separately, as the former has much fewer zeros compared to the latter. As shown in Tab. 3, adding the level-wise encoding into the pipeline improves the compression ratio even further, resulting in an average size of 0.28 mb, which is 7.39 times smaller than the size of the original mask.

Scanning order Since scanning order before compression can affect the output size, we also tried different scanning orders and measure the output sizes. Tab. 6 shows the results on the NeRF synthetic dataset. As shown, scanning order did not affect the performance noticeably.

5. Application to Other 2D Grid-based Neural Fields

Even though we only showed results using a TensoRF [2] model (VM-192), our proposed method is not confined to TensoRF. Only TensoRF was used in the main paper because it is currently the most effective method for plane-based neural fields.

To demonstrate that our method can be used with any grid-based neural representation, we apply it to additional plane-based neural fields in this section. We exclusively employed 2D grids, not 1D grids, as plane-based neural fields. This was only intended to demonstrate that our suggested method can be used with other 2D grid-based methods in addition to TensoRF. TriPlane from EG2D [1] served as the model for this design, which uses only 2D planes. The difference is that, like TensoRF, we separate the grids for density and appearance. For implementation, we removed 1D grids from the baseline model we used in the main paper.

As shown in Tab. 4, our proposed method can be successfully applied to other 2D grid-based representations and remove most of the coefficients without causing considerable quality degradation. When λ_m was set to 5e-11, our proposed method reduced the size to less than 7% of its original size with negligible quality drops (0.17 drops measured in PSNR).

6. Color Estimation in Detail

Following TensoRF [2], we use separate grids for density and appearance (color) estimation. In this section, we describe appearance grids in detail. Similar to density grids, we use three 2D matrices and three vectors, $\phi_c = \{\mathcal{W}^x_{c,r}, \mathcal{W}^y_{c,r}, \mathcal{W}^z_{c,r}, v^x_{c,r}, v^y_{c,r}, v^z_{c,r}\}_{r=1}^{N_{c,r}}$ (c denotes color, and we will omit the subscript c for brevity from now on). N_r is the number of ranks in matrix-vector decompo-

Lv. wise.	8-bit.	RLE	Huffman	Avg	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
				2.07	1.96	2.01	2.07	2.24	1.99	2.37	1.91	2.02
		√		1.19 (1.75)	1.39 (1.41)	1.58 (1.27)	1.47 (1.41)	0.70 (3.21)	1.08 (1.84)	1.46 (1.62)	0.73 (2.62)	1.08 (1.87)
		√	√	0.45 (4.62)	0.52 (3.75)	0.59 (3.38)	0.55 (3.74)	0.26 (8.49)	0.41 (4.82)	0.55 (4.32)	0.28 (6.78)	0.41 (4.89)
	√	√	√	0.33 (6.24)	0.40 (4.88)	0.42 (4.41)	0.42 (4.93)	0.18 (12.71)	0.31 (6.48)	0.39 (6.05)	0.20 (9.50)	0.30 (6.68)
√	√	√	√	0.28 (7.39)	0.36 (5.47)	0.35 (5.60)	0.35 (5.91)	0.16 (13.92)	0.26 (7.61)	0.34 (6.97)	0.16 (12.13)	0.26 (7.88)

Table 3. The mask size and compression ratio at each stage of the compression pipeline evaluated on the NeRF synthetic dataset. Each number is in megabytes, and the number inside parenthesis indicates the compression ratio. RLE and Huffman indicate run-length and Huffman encoding, respectively. Level-wise encoding and 8-bit casting are denoted as "Lv. wise." and "8-bit.". The first row shows the original binarized mask size.

Methods	size(MB)	Avg	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
Baseline	14.41	28.52	31.83	22.81	23.06	35.50	29.67	27.31	30.87	27.10
Ours (λ_m=1e-10)	0.71	27.91	30.69	23.18	23.22	33.18	38.85	27.09	30.29	26.79
Ours (λ_m=5e-11)	0.98	28.35	30.89	23.39	23.30	34.65	29.19	27.36	30.696	27.03
Ours (λ_m=2.5e-11)	1.38	28.39	31.07	23.18	23.27	34.78	29.18	27.44	31.05	27.14

Table 4. The performance of our proposed method with the tri-planar architecture on the NeRF synthetic dataset. Reconstruction quality was measured in PSNR. By setting λ_m to 5e-11, our proposed method compresses the baseline model 14.70 times without significant PSNR loss.

	Plane	Lv. 1	Lv. 2	Lv. 3	Lv. 4 (precinct)	Lv. 4 (upper-left)
Density	YX	0.99	0.97	0.91	0.76	0.33
	ZX	0.99	0.96	0.91	0.82	0.36
	ZY	0.99	0.97	0.93	0.84	0.43
Appearance	YX	0.97	0.94	0.90	0.80	0.61
	ZX	0.98	0.96	0.93	0.87	0.63
	ZY	0.99	0.98	0.95	0.91	0.73

Table 5. Sparsity table of 4-level DWT with learnable mask (λ_m=1e-10) on Chair (from the NeRF dataset).

	no scan	zigzag	morton	spiral
Size (MB)	0.83	0.83	0.85	0.83

Table 6. Average size by scanning order on the NeRF synthetic dataset.

sition and $\mathcal{W}_r^x \in \mathbb{R}^{H\times W}$, $\mathcal{W}_r^y \in \mathbb{R}^{W\times D}$, $\mathcal{W}_r^z \in \mathbb{R}^{H\times D}$ are matrices, $v_r^x \in \mathbb{R}^D$, $v_r^y \in \mathbb{R}^H$, $v_r^z \in \mathbb{R}^W$, are vectors in x, y, z directions, respectively. H, W, D denote the resolution of the grid. We employ DWT only over matrices, just as we did over density grids. Thus, $\mathcal{W}$ are wavelet coefficients, and v are feature vectors in the spatial domain.

Density grids in TensoRF only generate a density value, while color grids generate a feature vector for each coordinate. These feature vectors are forwarded to MLP to estimate the colors. To extract a feature vector per coordinate, TensoRF uses additional vectors (f^x, f^y, f^z). More formally, a 3D grid representation G_c can be defined as follows.

$$G_c = \sum_{r=1}^{N_r} \sum_{d\in\{x,y,z\}} v_r^d \otimes \texttt{idwt}(\mathcal{W}_r^d) \otimes f_r^d, \qquad (1)$$

where $\otimes$ denotes the outer product and $\texttt{idwt}(\cdot)$ is a two-dimensional inverse discrete wavelet transform.

7. Per-Scene Results

In this section, we provide quantitative and qualitative results of each scene from NeRF synthetic, NSVF synthetic, TanksAndTemples, and LLFF dataset. Tabs. 7 to 10 show the quantitative results and Figs. 2 to 5 show the qualitative results on the four datasets.

References

[1] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16123–16133, June 2022. 2

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*. Springer, 2022. 2

[3] W. B. Pennebaker et al. An overview of the basic principles of the q-coder adaptive binary arithmetic coder. *IBM Journal of Research and Development*, 1988. 2

Methods	size(MB)	Avg	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
KiloNeRF $^{\circ}$	$\leq$ 100	31.00	32.91	25.25	29.76	35.56	33.02	29.20	33.06	29.23
CCNeRF (CP) $^{\circ}$	4.4	30.55	-	-	-	-	-	-	-	-
NeRF *	1.25	31.52	33.82	24.94	30.33	36.70	32.96	29.77	34.41	29.25
cNeRF $^{\bullet}$	0.70	30.49	32.28	24.85	30.58	34.95	31.98	29.17	32.21	28.24
PREF *	9.88	31.56	34.55	25.15	32.17	35.73	34.59	29.09	32.64	28.58
VM-192 *	17.93	32.91	35.64	25.98	33.57	37.26	36.04	29.87	34.33	30.64
VM-48 *	4.81	32.18	34.54	25.55	33.12	36.60	35.14	29.15	33.33	29.99
CP-384 *	0.72	31.18	33.49	25.11	29.86	35.97	33.26	29.56	33.56	28.59
VM-192 (300) + Ours *										
λ_m=1e-10	0.83	31.95	34.14	25.53	32.87	36.08	34.93	29.42	33.48	29.15
λ_m=5e-11	1.16	32.13	34.52	25.66	33.03	36.20	35.16	29.58	33.68	29.19
λ_m=2.5e-11	1.69	32.24	34.68	25.56	33.17	36.37	35.50	29.56	33.74	29.34
VM-192 (500) + Ours *										
λ_m=1e-10	1.02	32.14	34.64	25.55	33.04	35.85	35.15	29.54	33.91	29.41
λ_m=5e-11	1.55	32.37	34.90	25.69	33.25	36.13	35.50	29.63	34.21	29.65
λ_m=2.5e-11	2.36	32.46	35.16	25.77	33.34	36.26	35.74	29.65	34.19	29.57
VM-384 (300) + Ours *										
λ_m=1e-10	0.99	32.08	34.32	25.50	33.28	36.22	34.83	29.91	33.36	29.20
λ_m=5e-11	1.50	32.23	34.59	25.55	33.41	36.35	35.13	29.95	33.47	29.42
λ_m=2.5e-11	2.42	32.38	34.84	25.58	33.59	36.66	35.46	29.94	33.56	29.43
VM-384 (500) + Ours *										
λ_m=1e-10	1.23	32.36	34.85	25.75	33.49	36.05	35.11	29.86	34.16	29.65
λ_m=5e-11	2.03	32.66	35.35	25.75	33.71	36.55	35.69	29.91	34.46	29.89
λ_m=2.5e-11	3.36	32.77	35.49	25.78	33.78	36.79	35.85	29.94	34.56	30.01

Table 7. Performance on the NeRF synthetic dataset measured in PSNR. The performance of the 32-bit and 8-bit models described in the original paper are represented by the symbols $^{\circ}$ and $^{\bullet}$, respectively. * denotes the performance of a quantized model with 8-bit precision. The number inside the parenthesis denotes the resolution of one axis of grids.

Methods	size(MB)	Avg	Bike	Lifestyle	Palace	Robot	Spaceship	Steamtrain	Toad	Wineholder
KiloNeRF $^{\circ}$	$\leq$ 100	33.37	35.49	33.15	34.42	32.93	36.48	33.36	31.41	29.72
VM-192 *	17.77	36.11	38.69	34.15	37.09	37.99	37.66	37.45	34.66	31.16
VM-48 *	4.53	34.95	37.55	33.34	35.84	36.60	36.38	36.68	32.97	30.26
CP-384 *	0.72	33.92	36.29	32.29	35.73	35.63	34.58	35.82	31.24	29.75
VM-192 (300) + Ours *										
λ_m=1e-10	0.87	34.67	37.06	33.44	35.18	35.74	37.01	36.65	32.23	30.08
λ_m=5e-11	1.25	34.95	37.33	33.69	35.65	36.01	37.23	36.95	32.58	30.14
λ_m=2.5e-11	1.88	35.11	37.49	33.75	35.94	36.23	37.45	36.92	32.87	30.23
VM-192 (500) + Ours *										
λ_m=1e-10	1.06	35.02	37.09	33.57	35.85	36.53	37.18	36.75	32.71	30.45
λ_m=5e-11	1.66	35.41	37.53	33.77	36.43	36.99	37.37	37.14	33.35	30.71
λ_m=2.5e-11	2.63	35.63	37.70	33.96	36.86	37.15	37.60	37.26	33.77	30.71
VM-384 (300) + Ours *										
λ_m=1e-10	1.04	35.04	37.72	33.68	35.55	36.18	37.52	36.85	32.48	30.39
λ_m=5e-11	1.61	35.33	38.04	33.89	36.03	36.48	37.81	37.10	32.88	30.43
λ_m=2.5e-11	2.69	35.57	38.27	34.09	36.37	36.81	37.93	37.24	33.22	30.59
VM-384 (500) + Ours *										
λ_m=1e-10	1.27	35.45	37.89	33.80	36.28	36.89	37.70	37.13	33.03	30.91
λ_m=5e-11	2.17	35.84	38.20	34.05	36.92	37.35	37.91	37.45	33.58	31.23
λ_m=2.5e-11	3.77	36.13	38.53	34.26	37.32	37.71	38.15	37.73	34.08	31.27

Table 8. Performance on the NSVF synthetic dataset measured in PSNR. The performance of the 32-bit models described in the original paper are represented by the symbols $^{\circ}$. * denotes the performance of a quantized model with 8-bit precision. The number inside the parenthesis denotes the resolution of one axis of grids.

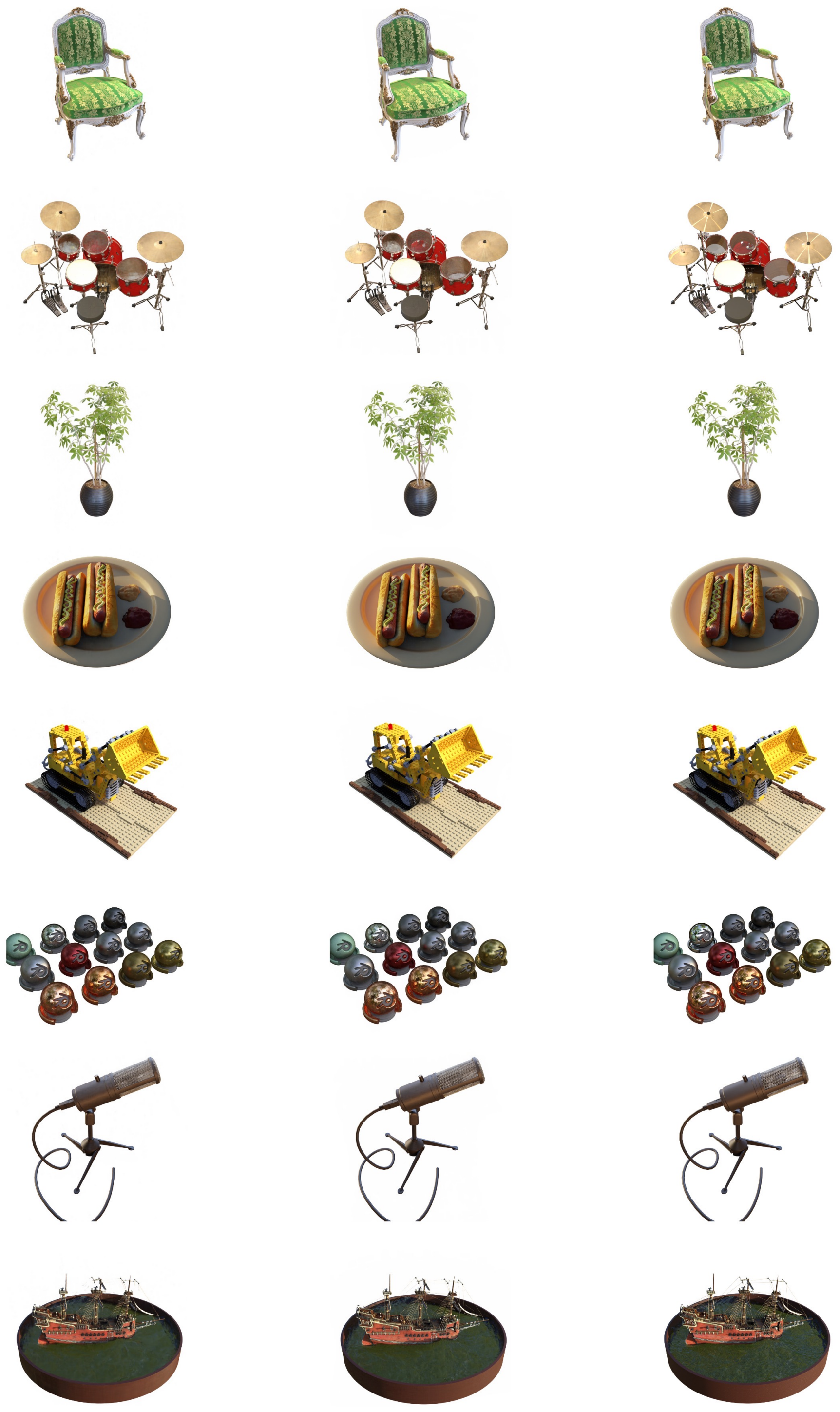

Figure 2. Qualitative results on the NeRF synthetic dataset.

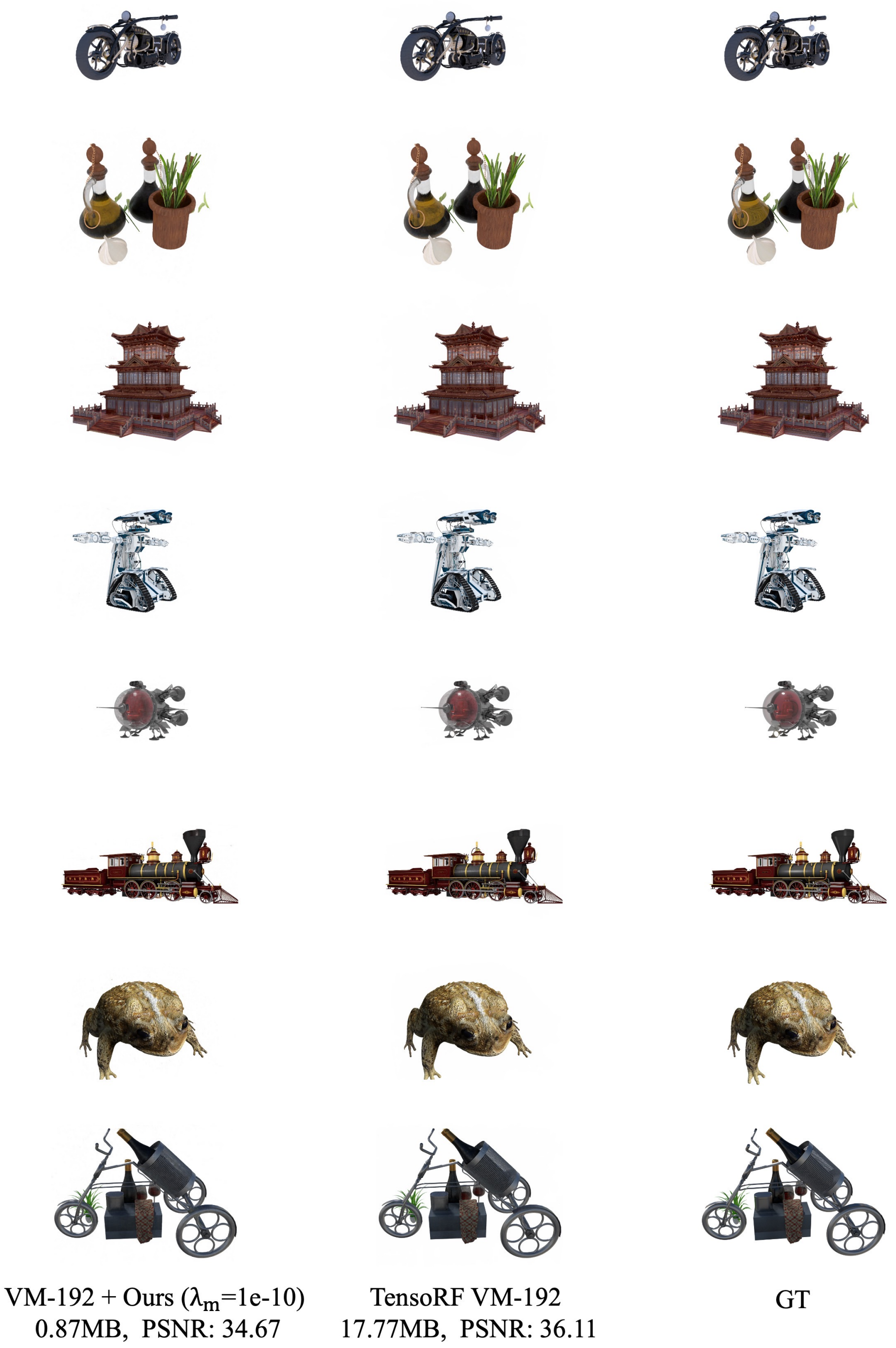

Figure 3. Qualitative results on the NSVF dataset.

Methods	size(MB)	average	Barn	Caterpillar	Family	Ignatius	Truck
KiloNeRF $^\circ$	≤ 100	28.41	27.81	25.61	33.65	27.92	27.04
CCNeRF (CP) $^\circ$	4.4	27.01	-	-	-	-	-
VM-192 *	17.82	28.55	27.25	26.18	33.86	28.37	27.11
VM-48 *	4.52	28.06	26.77	25.46	33.06	28.24	26.77
CP-384 *	0.72	27.56	26.73	24.69	32.31	27.83	26.23
VM-192 (300) + Ours *							
λ_m=1e-10	0.92	27.77	26.49	25.50	32.57	28.06	26.21
λ_m=5e-11	1.27	27.83	26.71	25.34	32.74	28.11	26.27
λ_m=2.5e-11	1.91	27.92	26.72	25.39	32.92	28.22	26.34
VM-192 (500) + Ours *							
λ_m=1e-10	1.14	27.92	26.89	25.52	32.79	28.18	26.22
λ_m=5e-11	1.76	28.01	26.97	25.40	33.03	28.23	26.42
λ_m=2.5e-11	2.77	28.04	27.05	25.34	33.18	28.21	26.43
VM-384 (300) + Ours *							
λ_m=1e-10	1.13	28.01	26.94	25.75	32.72	28.22	26.43
λ_m=5e-11	1.69	28.12	27.02	25.81	32.92	28.31	26.54
λ_m=2.5e-11	2.75	28.12	27.00	25.80	33.09	28.23	26.47
VM-384 (500) + Ours *							
λ_m=1e-10	1.42	28.14	27.41	25.78	32.91	28.11	26.48
λ_m=5e-11	2.43	28.27	27.47	25.89	33.14	28.36	26.49
λ_m=2.5e-11	4.15	28.30	27.39	25.79	33.33	28.32	26.69

Table 9. Performance on the Tanks&Temples synthetic dataset measured in PSNR. The performance of the 32-bit models described in the original paper are represented by the symbols $^\circ$. * denotes the performance of a quantized model with 8-bit precision. The number inside the parenthesis denotes the resolution of one axis of grids.

Methods	size(MB)	Avg	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
cNeRF $^\bullet$	0.96	26.15	25.17	27.21	31.15	27.28	20.95	20.09	30.65	26.72
PREF *	9.34	24.50	23.32	26.37	29.71	25.24	20.21	19.02	28.45	23.67
VM-96 *	44.72	26.66	25.22	28.55	31.23	28.10	21.28	19.87	32.17	26.89
VM-48 *	22.40	26.46	25.27	28.19	31.06	27.59	21.33	20.03	31.70	26.54
CP-384 *	0.64	25.51	24.30	26.88	30.17	26.46	20.38	19.95	30.61	25.35
VM-96 (640) + Ours *										
λ_m=1e-10	1.34	25.88	24.98	27.19	30.28	26.96	21.21	19.93	30.03	26.45
λ_m=5e-11	2.10	26.15	24.99	27.77	30.60	27.25	21.18	19.90	30.65	26.84
λ_m=2.5e-11	3.20	26.25	25.05	27.94	30.75	27.48	21.08	19.76	31.19	26.77
VM-96 (1000) + Ours *										
λ_m=1e-10	1.75	25.82	24.97	27.44	30.29	26.92	21.11	20.09	29.27	26.47
λ_m=5e-11	3.01	26.17	25.05	27.70	30.71	27.29	21.09	20.01	30.91	26.62
λ_m=2.5e-11	4.76	26.30	25.08	27.76	30.89	27.49	21.14	19.99	31.23	26.85
VM-192 (640) + Ours *										
λ_m=1e-10	1.73	25.98	25.18	27.47	29.66	27.47	21.11	19.71	30.47	26.79
λ_m=5e-11	3.01	26.46	25.12	28.16	30.81	27.88	21.07	19.77	31.61	27.28
λ_m=2.5e-11	5.04	26.53	25.05	28.15	30.99	28.09	20.97	19.75	31.85	27.40
VM-192 (1000) + Ours *										
λ_m=1e-10	2.43	26.15	25.18	27.74	30.22	27.47	21.24	19.98	30.56	26.79
λ_m=5e-11	4.57	26.43	25.22	28.06	30.78	27.71	21.23	19.92	31.61	26.93
λ_m=2.5e-11	7.41	26.54	25.27	28.20	31.01	27.88	21.17	20.02	31.73	27.07

Table 10. Performance on the LLFF dataset measured in PSNR. The performance of the 8-bit models described in the original paper are represented by the symbol $^\bullet$. * denotes the performance of a quantized model with 8-bit precision. The number inside the parenthesis denotes the resolution of one axis of grids.

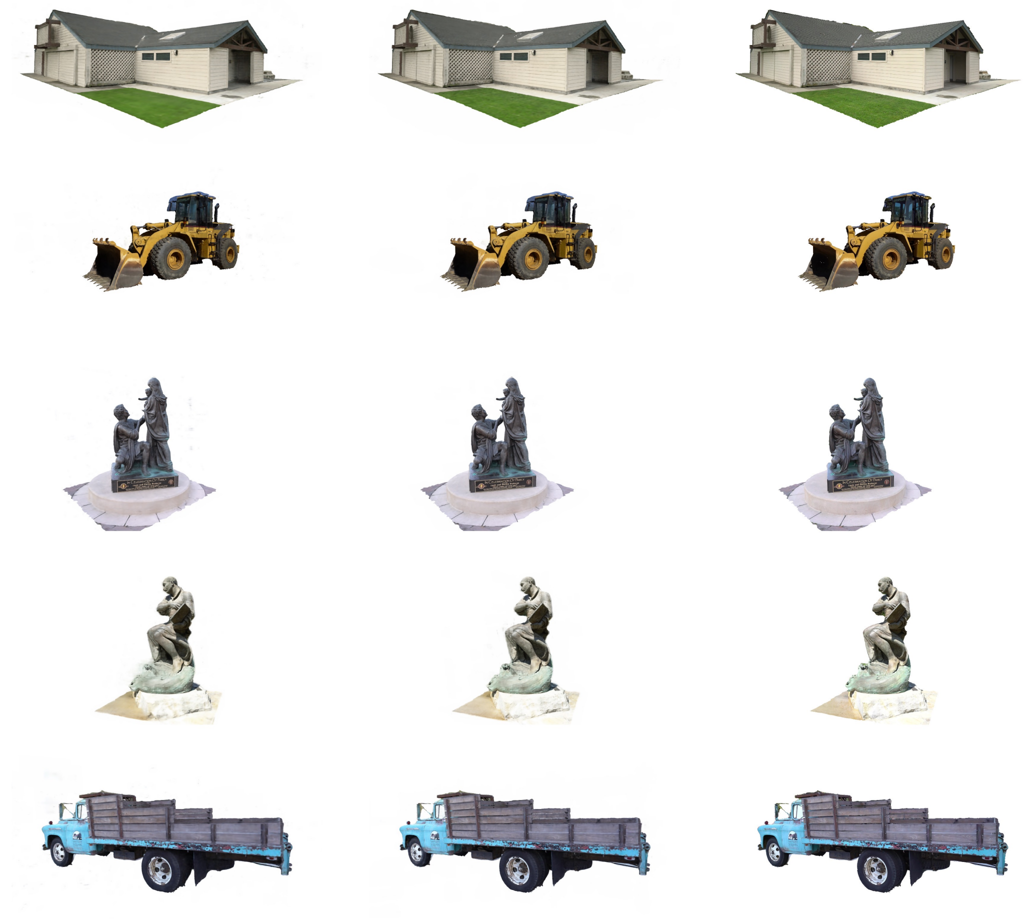

Figure 4. Qualitative results on the Tanks&Temples dataset.

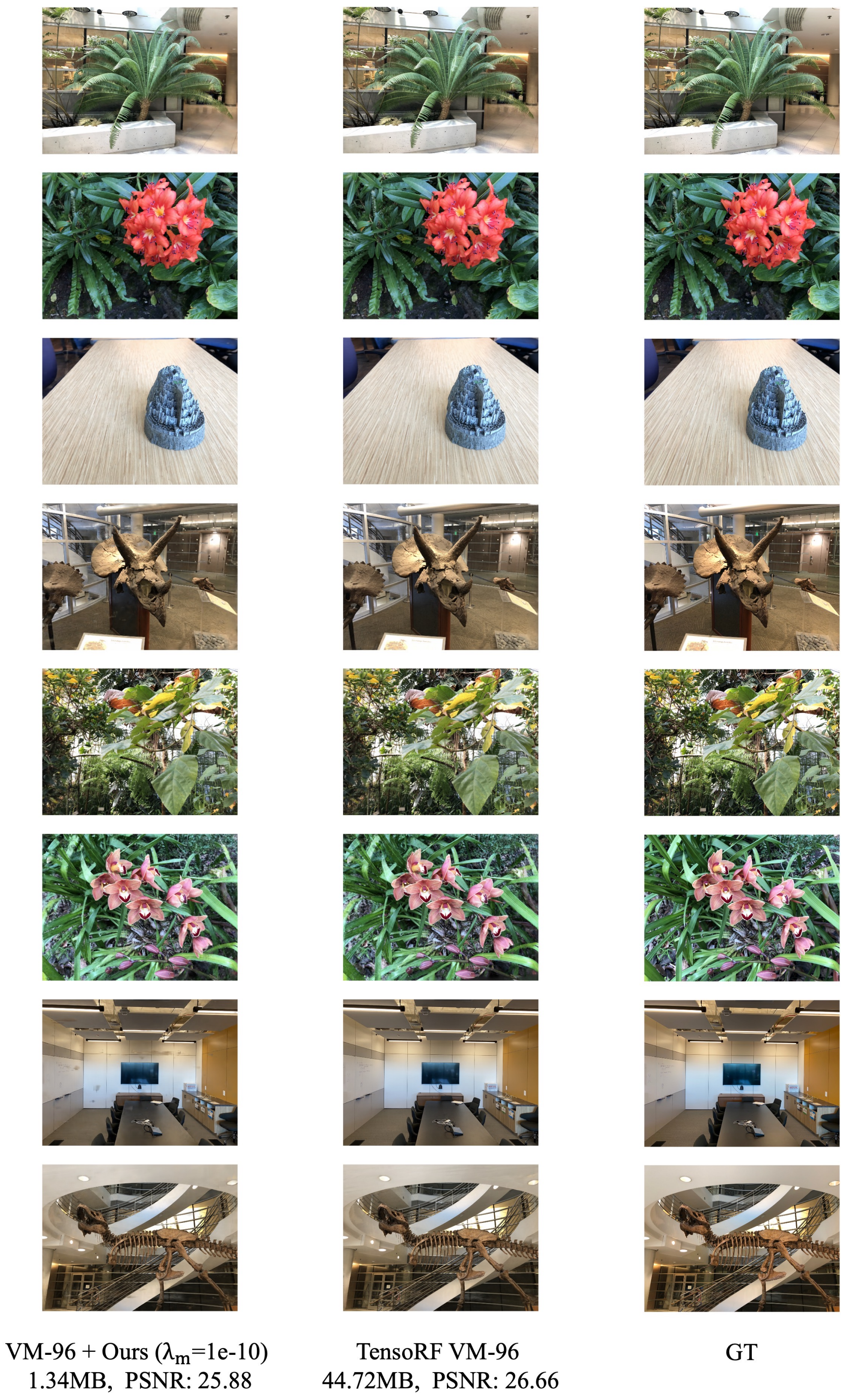

Figure 5. Qualitative results on the LLFF dataset.