DaReNeRF: Direction-aware Representation for Dynamic Scenes

Supplementary Material

In this supplementary material, we provide further methodological context and implementation details to facilitate reproducibility of our framework DaReNeRF. We also showcase additional quantitative and qualitative results to further highlight the contributions claimed in the paper.

A. Video Presentation

A video presentation of DaReNeRF and its results can be found online, at `https://www.youtube.com/watch?v=hYQsl6Rbxn4`.

B. Dual-Tree Complex Wavelet Transform

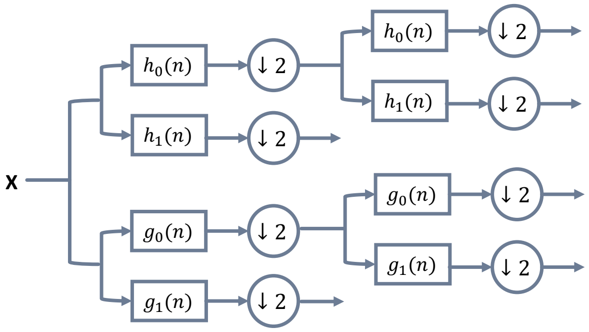

Figure S1. **Analysis Filter Bank**, for the dual tree complex wavelet transfrom.

The idea of dual-tree complex wavelet transform (DTCWT) [10] is quite straightforward. The DTCWT employs two real discrete wavelet transforms (DWTs). The first DWT gives the real part of the transform while the second DWT gives the imaginary part. The analysis filter banks used to implement the DTCWT is illustrated in Figure S1. Here $h_0(n)$, $h_1(n)$ denote the low-pass/high-pass filter pair for upper filter bank, and $g_0(n)$, $g_1(n)$ denote the low-pass/high-pass filter pair for the lower filter bank. The two real wavelets associated with each of the two real wavelet transforms as $\psi_h(t)$ and $\psi_g(t)$. And the complex wavelet can be denoted as $\psi(t) = \psi_h(t) + j\psi_g(t)$. The $\psi_g(t)$ is approximately the Hilbert transform of $\psi_h(t)$. The 2D DTCWT $\psi(x, y) = \psi(x)\psi(y)$ associated with the row-column implementation of the wavelet transform, where $\psi(x)$ is a complex wavelet given by $\psi(x) = \psi_h(x) + j\psi_g(x)$. Then we obtain for $\psi(x, y)$ the expression:

$$\begin{aligned}\psi(x, y) &= [\psi_h(x) + j\psi_g(x)][\psi_h(y) + j\psi_g(y)] \\ &= \psi_h(x)\psi_h(y) - \psi_g(x)\psi_g(y) \\ &+ j[\psi_g(x)\psi_h(y) + \psi_h(x)\psi_g(y)]\end{aligned} \tag{1}$$

The spectrum of $\psi_h(x)\psi_h(y) - \psi_g(x)\psi_g(y)$ which corresponds to the real part of $\psi(x, y)$ is supported in two quadrants of the 2D frequency plane and is oriented at $-45°$. Note that the $\psi_h(x)\psi_h(y)$ is the HH wavelet of a separable 2D real wavelet transform implemented using the filter pair $\{h_0(n), h_1(n)\}$. Similarly, $\psi_g(x)\psi_g(y)$ is the HH wavelet of a real separable wavelet transform, implemented using the filters $\{g_0(n), g_1(n)\}$. To obtain a real 2D wavelet oriented at $+45°$, we consider now the complex 2-D wavelet $\psi(x, y) = \psi(x)\overline{\psi(y)}$, where $\overline{\psi(y)}$ represents the complex conjugate of $\psi(y)$. This gives us the following expression:

$$\begin{aligned}\psi(x, y) &= [\psi_h(x) + j\psi_g(x)]\overline{[\psi_h(y) + j\psi_g(y)]} \\ &= \psi_h(x)\psi_h(y) + \psi_g(x)\psi_g(y) \\ &+ j[\psi_g(x)\psi_h(y) - \psi_h(x)\psi_g(y)]\end{aligned} \tag{2}$$

The spectrum of $\psi_h(x)\psi_h(y) + \psi_g(x)\psi_g(y)$ is supported in two quadrants of the 2D frequency plane and is oriented at $+45°$. We could obtain four more oriented real 2D wavelets by repeating the above procedure on the following complex 2-D wavelets: $\phi(x)\psi(y)$, $\psi(x)\phi(y)$, $\phi(x)\overline{\psi(y)}$ and $\psi(x)\overline{\phi(y)}$, where $\psi(x) = \psi_h(x) + j\psi_g(y)$ and $\phi(x) = \phi_h(x) + j\phi_g(y)$. By taking the real part of each of these four complex wavelets, we obtain four real oriented 2D wavelets, in additional to the two already obtain in 1 and 2:

$$\psi_i(x, y) = \frac{1}{\sqrt{2}}(\psi_{1,i}(x, y) - \psi_{2,i}(x, y)), \tag{3}$$

$$\psi_{i+3}(x, y) = \frac{1}{\sqrt{2}}(\psi_{1,i}(x, y) + \psi_{2,i}(x, y)) \tag{4}$$

for $i = 1, 2, 3$, where the two separable 2-D wavelet bases are defined in the usual manner:

$$\begin{aligned}&\psi_{1,1}(x, y) = \phi_h(x)\psi_h(y), \psi_{2,1}(x, y) = \phi_g(x)\psi_g(y), \\ &\psi_{1,2}(x, y) = \psi_h(x)\phi_h(y), \psi_{2,2}(x, y) = \psi_g(x)\phi_g(y), \\ &\psi_{1,3}(x, y) = \psi_h(x)\psi_h(y), \psi_{2,3}(x, y) = \psi_g(x)\psi_g(y),\end{aligned} \tag{5}$$

We have used the normalization $\frac{1}{\sqrt{2}}$ only so that the sum and difference operation constitutes an orthonormal operation. From the imaginary parts of $\psi(x)\psi(y)$, $\psi(x)\overline{\psi(y)}$, $\phi(x)\psi(y)$, $\psi(x)\phi(y)$, $\phi(x)\overline{\psi(y)}$ and $\psi(x)\overline{\phi(y)}$, we could obtain six oriented wavelets given by:

$$\psi_i(x, y) = \frac{1}{\sqrt{2}}(\psi_{3,i}(x, y) + \psi_{4,i}(x, y)), \tag{6}$$

$$\psi_{i+3}(x, y) = \frac{1}{\sqrt{2}}(\psi_{3,i}(x, y) - \psi_{4,i}(x, y)) \tag{7}$$

for $i = 1, 2, 3$, where the two separable 2D wavelet bases are defined as:

$$\begin{aligned}&\psi_{3,1}(x, y) = \phi_g(x)\psi_h(y), \psi_{4,1}(x, y) = \phi_h(x)\psi_g(y), \\ &\psi_{3,2}(x, y) = \psi_g(x)\phi_h(y), \psi_{4,2}(x, y) = \psi_h(x)\phi_g(y), \\ &\psi_{3,3}(x, y) = \psi_g(x)\psi_h(y), \psi_{4,3}(x, y) = \psi_h(x)\psi_g(y),\end{aligned} \tag{8}$$

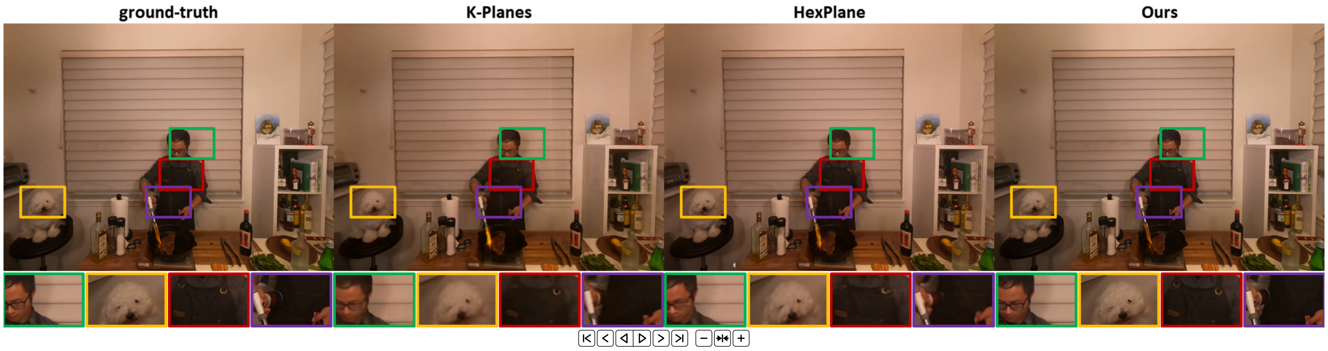

Figure S2. **Visual Comparison on Dynamic Scenes (Plenoptic Data).** K-Planes and HexPlane are concurrent decomposition-based methods. As shown in the four zoomed-in patches, our method better reconstruct fine details and captures motion. To see the figure animated, please view the document with compatible software, *e.g*., *Adobe Acrobat* or *KDE Okular.*

Table S1. Results of Plenoptic Video Dataset. We report results of each scene

Model	Flame Salmon			Cook Spinach			Cut Roasted Beef		
	PSNR ↑	D-SSIM ↓	LPIPS ↓	PSNR ↑	D-SSIM ↓	LPIPS ↓	PSNR ↑	D-SSIM ↓	LPIPS ↓
DaReNeRF-S	30.294	0.015	0.089	32.630	0.013	0.100	33.087	0.013	0.092
DaReNeRF	**30.441**	**0.012**	**0.084**	**32.836**	**0.011**	**0.090**	**33.200**	**0.011**	**0.091**
	Flame Steak			Sear Steak			Coffee Martini		
DaReNeRF-S	33.259	0.011	0.081	33.179	0.011	0.075	30.160	0.016	0.092
DaReNeRF	**33.524**	**0.009**	**0.077**	**33.351**	**0.009**	**0.072**	**30.193**	**0.014**	**0.089**

Thus we could obtain six oriented wavelets from both real and imaginary part.

C. Additional Results on Various Datasets

C.1. Plenoptic Video Dataset [6]

The quantitative results for each scene are presented in Table S1, while additional visualizations comparing DaReNeRF with current state-of-the-art methods, HexPlane [1] and K-Planes [3], are provided in Figure S3. Notably, DaReNeRF demonstrates superior recovery of texture details. We also provide an animated qualitative comparison in Figure S2. Furthermore, comprehensive visualizations of DaReNeRF on all six scenes in the Plenoptic dataset are shown in Figure S4 and Figure S5.

C.2. D-NeRF Dataset [8]

We provide quantitative results for each scene in Table S2, while additional visualizations comparing DaReNeRF with current state-of-the-art methods, HexPlane [1] and 4D-GS [12], are shared in Figure S6. We also provide further visualization in a video attached to this supplementary material. Remarkably, although 4D-GS incorporates a deformation field, DaReNeRF still outperforms it in certain cases from the D-NeRF dataset. Furthermore, comprehensive visualizations of DaReNeRF on six scenes in the Plenoptic dataset are shown in Figure S7 and the failure cases are shown in Figure S8.

C.3. NeRF Synthetic Dataset

The quantitative results for each case are presented in Table S3, while additional visualizations comparing our representation with DWT [9] based representation method, are shown in Figure S9. Furthermore, comprehensive visualizations of eight scenes in the NeRF dataset are shown in Figure S10 and in the attached video.

C.4. NSVF Synthetic Dataset

The quantitative results for each case are presented in Table S4, while additional visualizations comparing our representation with DWT [9] based representation method, are shown in Figure S11. Furthermore, comprehensive visualizations of eight scenes in the NSVF dataset are shown in Figure S12.

C.5. LLFF Dataset

The quantitative results for each case are presented in Table S5, while additional visualizations comparing our representation with DWT [9] based representation method, are shown in Figure S13. Furthermore, comprehensive visualizations of eight scenes in the NSVF dataset are shown in Figure S14 and in the video.

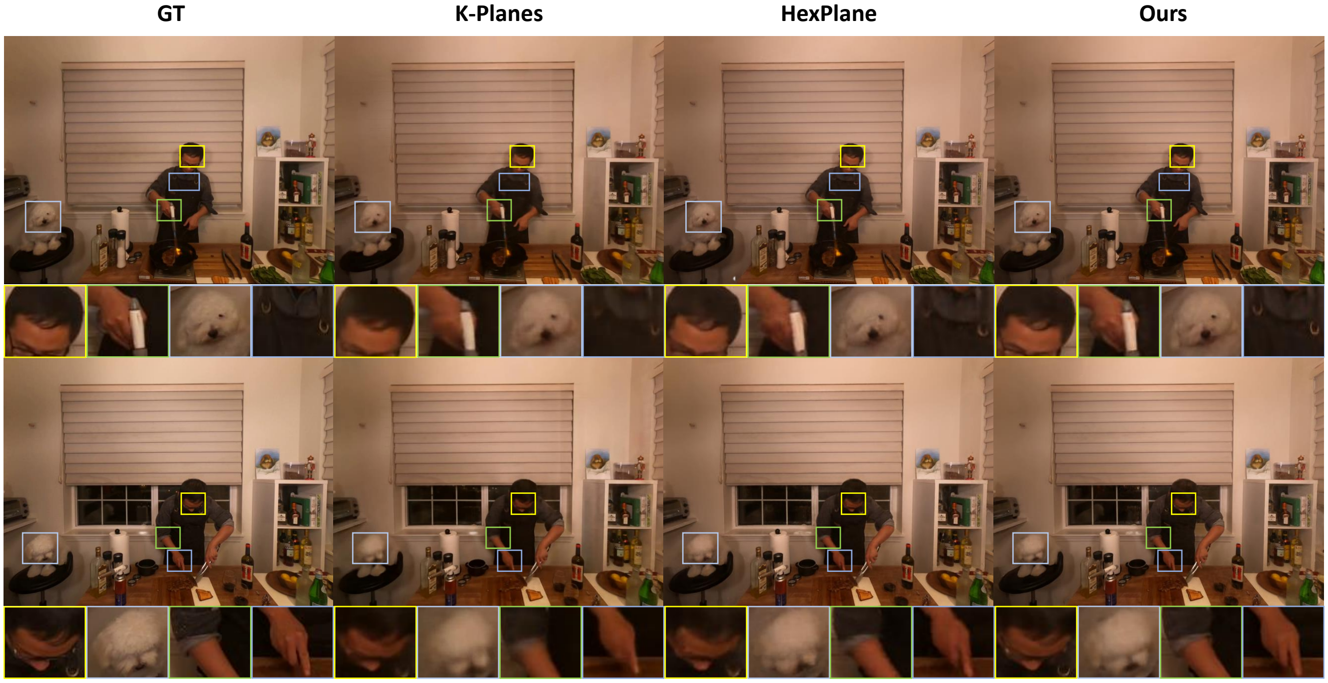

Figure S3. Visual Comparison on Dynamic Scenes (Plenoptic Data). K-Planes and HexPlane are concurrent decomposition-based methods. As shown in the four zoomed-in patches, our method better reconstructs fine details and captures motion.

D. Additional Ablation Studies

D.1. Sparsity Masks

We evaluate the performance of our direction-aware representation at various sparsity levels controlled by the mask loss weight λ_m. The quantitative and qualitative results on the NSVF dataset with different sparsity levels are presented in Table S6 and Figure S15.

D.2. Wavelet Levels

We investigated the impact of scene reconstruction performance across different wavelet levels, and the results are presented in Table S7. Interestingly, we observed that increasing the wavelet level did not lead to significant performance improvements. Conversely, we noted a substantial increase in both training time and model size with the increment of wavelet level. As a result, throughout all experiments, we consistently set the wavelet level to 1.

D.3. Training Time Analysis

To effectively demonstrate the efficiency of our proposed DaReNeRF, we conducted a comparative analysis against HexPlane under identical training durations of 2 hours (equivalent to HexPlane-100k) and 12 hours (equivalent to HexPlane-650k). The results, outlined in Table S8, reveal that across varying training periods, our proposed DaReNeRF consistently outperforms the baseline HexPlane.

D.4. Training Data Sparsity Analysis

In order to delve deeper into the few-shot capabilities of our proposed direction-aware representation, we conducted experiments with varying levels of training data sparsity. This was achieved by randomly dropping training frames while ensuring sufficient data remained to effectively learn motion on the D-NeRF dataset. The corresponding results are presented in Table S9. Remarkably, our proposed DaReNeRF consistently outperforms the baseline across different levels of training data sparsity.

E. Training Details

E.1. Plenoptic Video Dataset [6]

Plenoptic Video Dataset is a multi-view real-world video dataset, where each video is 10-second long. For training, we set $R_1 = 48$, $R_2 = 48$ and $R_3 = 48$ for appearance, where R_1, R_2 and R_3 are basis numbers for direction-aware representation of $XY - ZT$, $XZ - YT$ and $YZ - XT$ planes. For opacity, we set $R_1 = 24$, $R_2 = 24$ and $R_3 = 24$. The scene is modeled using normalized device coordinate (NDC) [7] with min boundaries $[-2.5, -2.0, -1.0]$ and max boundaries $[2.5, 2.0, 1.0]$.

Table S2. Results of D-NeRF Dataset. We report results of each scene

Model	Hell Warrior			Mutant			Hook		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
T-NeRF	23.19	0.93	0.08	30.56	0.96	0.04	27.21	0.94	0.06
D-NeRF	25.02	0.95	0.06	31.29	0.97	0.02	29.25	0.96	0.11
TiNeuVox-S	27.00	0.95	0.09	31.09	0.96	0.05	29.30	0.95	0.07
TiNeuVox-B	28.17	0.97	0.07	33.61	0.98	0.03	31.45	0.97	0.05
HexPlane	24.24	0.94	0.07	33.79	0.98	0.03	28.71	0.96	0.05
DaReNeRF-S	25.71	0.95	0.04	34.08	0.98	0.02	29.04	0.96	0.04
DaReNeRF	25.82	0.95	0.04	34.17	0.98	0.01	28.96	0.96	0.04
	Bouncing Balls			Lego			T-Rex		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
T-NeRF	37.81	0.98	0.12	23.82	0.90	0.15	30.19	0.96	0.13
D-NeRF	38.93	0.98	0.10	21.64	0.83	0.16	31.75	0.97	0.03
TiNeuVox-S	39.05	0.99	0.06	24.35	0.88	0.13	29.95	0.96	0.06
TiNeuVox-B	40.73	0.99	0.04	25.02	0.92	0.07	32.70	0.98	0.03
HexPlane	39.69	0.99	0.03	25.22	0.94	0.04	30.67	0.98	0.03
DaReNeRF-S	42.24	0.99	0.01	25.24	0.94	0.03	31.75	0.98	0.03
DaReNeRF	42.26	0.99	0.01	25.44	0.95	0.03	32.21	0.98	0.02
	Stand Up			Jumping Jacks			Average		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
T-NeRF	31.24	0.97	0.02	32.01	0.97	0.03	29.51	0.95	0.08
D-NeRF	32.79	0.98	0.02	32.80	0.98	0.03	30.50	0.95	0.07
TiNeuVox-S	32.89	0.98	0.03	32.33	0.97	0.04	30.75	0.96	0.07
TiNeuVox-B	35.43	0.99	0.02	34.23	0.98	0.03	32.64	0.97	0.04
HexPlane	34.36	0.98	0.02	31.65	0.97	0.04	31.04	0.94	0.04
DaReNeRF-S	34.47	0.98	0.02	31.99	0.97	0.03	31.82	0.97	0.03
DaReNeRF	34.58	0.98	0.02	32.21	0.97	0.03	31.95	0.97	0.03

Table S3. Results of NeRF Synthetic Dataset

Bit Precision	Method	Size(MB)	Avg	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
32-bit	KiloNeRF	≤ 100	31.00	32.91	25.25	29.76	35.56	33.02	29.20	33.06	29.23
32-bit	CCNeRF (CP)	4.4	30.55	-	-	-	-	-	-	-	-
8-bit*	NeRF	1.25	31.52	33.82	24.94	30.33	36.70	32.96	29.77	34.41	29.25
8-bit	cNeRF	0.70	30.49	32.28	24.85	30.58	34.95	31.98	29.17	32.21	28.24
8-bit*	PREF	9.88	31.56	34.55	25.15	32.17	35.73	34.59	29.09	32.64	28.58
8-bit*	VM-192	17.93	32.91	35.64	25.98	33.57	37.26	36.04	29.87	34.33	30.64
8-bit*	VM-192 (300) + DWT	0.83	31.95	34.14	25.53	32.87	36.08	34.93	29.42	33.48	29.15
8-bit*	VM-192 (300) + Ours	8.91	32.42	36.05	29.40	35.26	36.37	25.58	33.26	29.82	33.63

During the training, DaReNeRF starts with a space grid size of 64^3 and double its resolution at 20k, 40k and 70k to 512^3. The emptiness voxel is calculated at 30k, 50k and 80k. The learning rate for representation planes is 0.02 and the learning rate for V^{RF} and neural network is 0.001. All learning rates are exponentially decayed. We use Adam [4] optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We apply the total variational loss on all representation planes with loss

Table S4. Results of NSVF Synthetic Dataset

Bit Precision	Method	Size(MB)	Avg	Bike	Lifestyle	Palace	Robot	Spaceship	Steamtrain	Toad	Wineholder
32-bit	KiloNeRF	$\leq$ 100	33.77	35.49	33.15	34.42	32.93	36.48	33.36	31.41	29.72
8-bit*	VM-192	17.77	36.11	38.69	34.15	37.09	37.99	37.66	37.45	34.66	31.16
8-bit*	VM-48	4.53	34.95	37.55	33.34	35.84	36.60	36.38	36.68	32.97	30.26
8-bit*	CP-384	0.72	33.92	36.29	32.29	35.73	35.63	34.58	35.82	31.24	29.75
8-bit*	VM-192 (300) + DWT	0.87	34.67	37.06	33.44	35.18	35.74	37.01	36.65	32.23	30.08
8-bit*	VM-192 (300) + Ours	8.98	36.24	38.78	34.21	37.22	38.02	38.61	37.79	34.39	30.97

Table S5. Results of LLFF Dataset

Bit Precision	Method	Size(MB)	Avg	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
8-bit	cNeRF	0.96	26.15	25.17	27.21	31.15	27.28	20.95	20.09	30.65	26.72
8-bit*	PREF	9.34	24.50	23.32	26.37	29.71	25.24	20.21	19.02	28.45	23.67
8-bit*	VM-96	44.72	26.66	25.22	28.55	31.23	28.10	21.28	19.87	32.17	26.89
8-bit*	VM-48	22.40	26.46	25.27	28.19	31.06	27.59	21.33	20.03	31.70	26.54
8-bit*	CP-384	0.64	25.51	24.30	26.88	30.17	26.46	20.38	19.95	30.61	25.35
8-bit*	VM-192 (300) + DWT	1.34	25.88	24.98	27.19	30.28	26.96	21.21	19.93	30.03	26.45
8-bit*	VM-192 (300) + Ours	13.67	26.48	25.02	28.23	31.07	27.81	21.24	19.68	31.82	26.97

weight $\lambda = 1e-5$ for spatial planes and $\lambda = 2e-5$ for spatial-temporal planes. For DaReNeRF-S we set weight of mask loss as $1e-11$.

We follow the hierarchical training pipeline suggested in [6]. Both DaReNeRF and DaReNeRF-S use 100k iterations, with 10k stage one training, 50k stage two training and 40k stage three training. Stage one is a global-median-based weighted sampling with $\gamma = 0.02$; stage two is also a global-median based weighted sampling with $\gamma = 0.02$; stage three is a temporal-difference-based weighted sampling with $\gamma = 0.2$.

In evaluation, D-SSIM is computed as $\frac{1-MS-SSIM}{2}$ and LPIPS [13] is calculated using AlexNet [5].

E.2. D-NeRF Dataset [8]

We set $R_1 = 48$, $R_2 = 48$ and $R_3 = 48$ for appearance and $R_1 = 24$, $R_2 = 24$ and $R_3 = 24$ for opacity. The bounding box has max boundaries $[1.5, 1.5, 1.5]$ and min boundaries $[-1.5, -1.5, -1.5]$. During the training, both DaReNeRF and DaReNeRF-S starts with space grid of 32^3 and upsampling its resolution at 3k, 6k and 9k to 200^3. The emptiness voxel is calculated at 4k, 8k and 10k iterations. Total training iteration is 25k. The learning rate for representation planes are 0.02 and learning rate for V^{RF} and neural network is 0.001. All learning rates are exponentially decayed. We use Adam [4] optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. In evaluation, LPIPS [13] is calculated using VGG-Net [11] following previous works.

For **both** the Plenoptic Video dataset and the D-NeRF dataset, we set the learning rate of the masks in DaReNeRF-S same as their representation planes and we employ a compact MLP for regressing output colors. The MLP consists of 3 layers, with a hidden dimension of 128.

E.3. Static Scene

For three static scene dataset NeRF synthetic dataset, NSVF synthetic dataset and LLFF dataset, we followed the experimental settings of TensoRF [2]. We trained our model for 30000 iterations, each of which is a minibatch of 4096 rays. We used Adam [4] optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ and an exponential learning rate decay scheduler. The initial learning rates of representation-related parameters and neural network (MLP) were set to 0.02 and 0.001. For the **NeRF synthetic** and **NSVF synthetic** datasets, we adopt TensoRF-192 as the baseline and update the alpha masks at the 2k, 4k, 6k, 11k, 16k, and 26k iterations. The initial grid size is set to 128^3, and we perform upsampling at 2k, 3k, 4k, 5.5k, and 7k iterations, reaching a final resolution of 300^3. For the **LLFF** dataset, we adopt TensoRF-96 as the baseline and update the alpha masks at the 2.5k, 4k, 6k, 11k, 16k, and 21k iterations. The initial grid size is set to 128^3, and we perform upsampling at 2k, 3k, 4k and 5.5k iterations, reaching a final resolution of 640^3. The learning rates of masks are set same as learning rates of representation-related parameters. We employ a compact MLP for regressing output colors. The MLP consists of 3 layers, with a hidden dimension of 128.

Table S6. Quantitative results on NSVF dataset with different sparsity.

Sparsity	λ_m	Size(MB)	Avg	Bike	Lifestyle	Palace	Robot	Spaceship	Steamtrain	Toad	Wineholder
99.2%	1.0×10^{-10}	1.16	35.36	38.01	33.69	35.70	37.23	37.83	37.26	32.58	30.56
97.3%	5.0×10^{-11}	3.18	35.81	38.52	34.01	36.33	37.79	38.22	37.46	33.33	30.82
94.2%	2.5×10^{-11}	8.98	36.24	38.78	34.21	37.22	38.02	38.61	37.79	34.39	30.97
-	0	135	36.34	38.86	34.37	37.25	38.06	38.72	37.89	34.46	31.09

Table S7. **Wavelet Level Analysis of Direction-Aware Representation**, evaluated on NVSF data.

Level	PSNR ↑	Model Size (MB) ↓	Training Time (min) ↓
1	36.34	**135**	**23**
2	36.45	152	41
3	**36.49**	163	55

Table S8. Time eval. on Plenoptic (`FlameSteak`/`CutRoastBeef`).

Model	Eval. after training for **2hrs**			Eval. after training for **12hrs**		
	PSNR ↑	D-SSIM ↓	LPIPS ↓	PSNR ↑	D-SSIM ↓	LPIPS ↓
HexPlane	31.92 / 32.71	.012 / .015	.081 / .094	32.08 / 32.55	.011 / .013	.066 / .080
DaReNeRF	**33.01** / **32.98**	**.010** / **.013**	**.079** / **.092**	**33.62** / **33.43**	**.009** / **.010**	**.063** / **.076**

Table S9. Evaluation on D-NeRF with various training set sparsity.

Model	**75%** training set (average)			**50%** training set (average)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
HexPlane	29.85	0.95	0.05	28.03	0.94	0.06
DaReNeRF	**30.95**	**0.96**	**0.04**	**29.28**	**0.96**	**0.05**

References

[1] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 5

[3] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 2

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 5

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 5

[6] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2, 3, 5

[7] 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. *Communications of the ACM*, 65(1):99–106, 2021. 3

[8] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2, 5

[9] Daniel Rho, Byeonghyeon Lee, Seungtae Nam, Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Masked wavelet representation for compact neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20680–20690, 2023. 2

[10] Ivan W Selesnick, Richard G Baraniuk, and Nick C Kingsbury. The dual-tree complex wavelet transform. *IEEE signal processing magazine*, 22(6):123–151, 2005. 1

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[12] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 2

[13] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

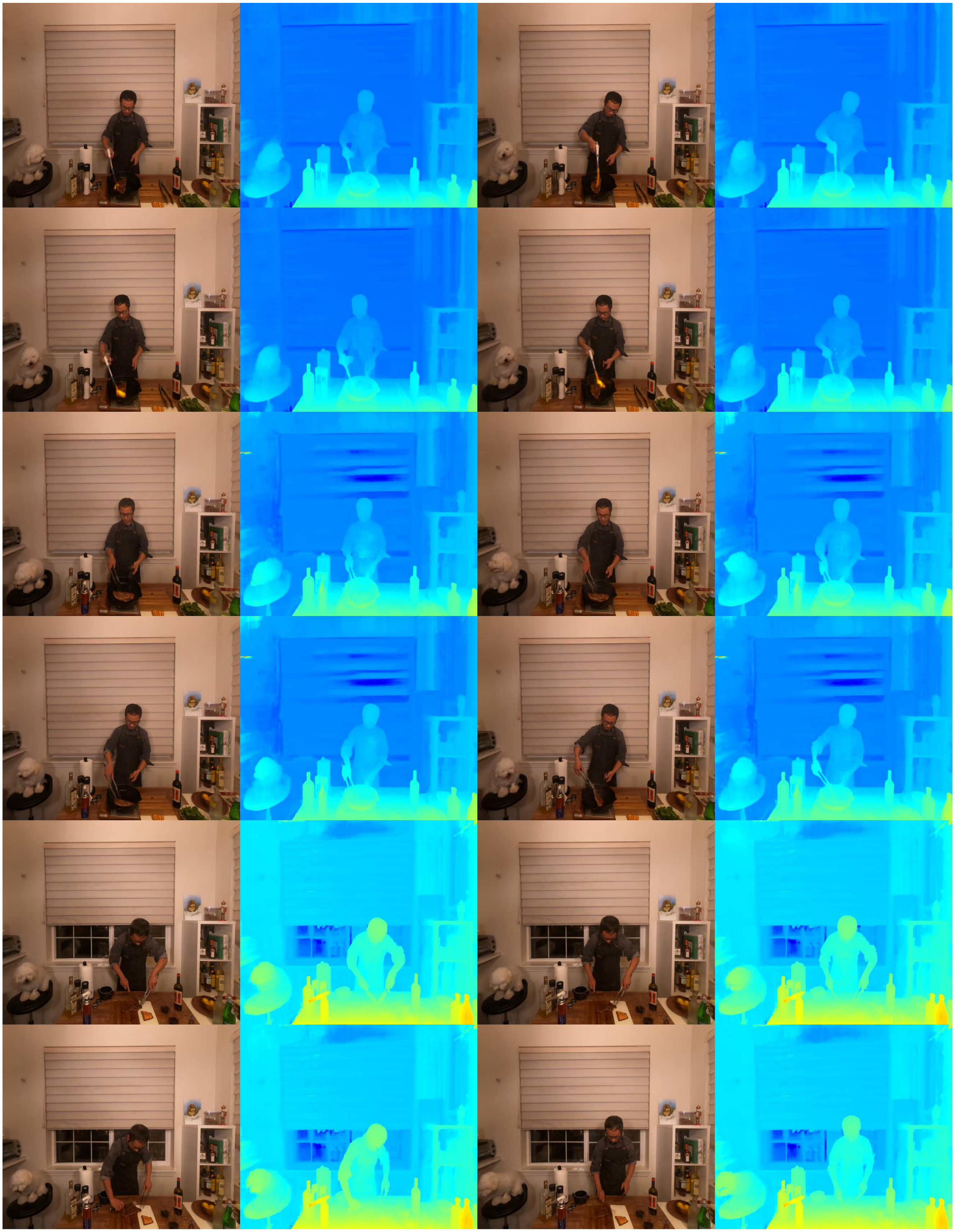

Figure S4. Visualizations on `flame steak`, `sear steak` and `cut roasted beef` scene.

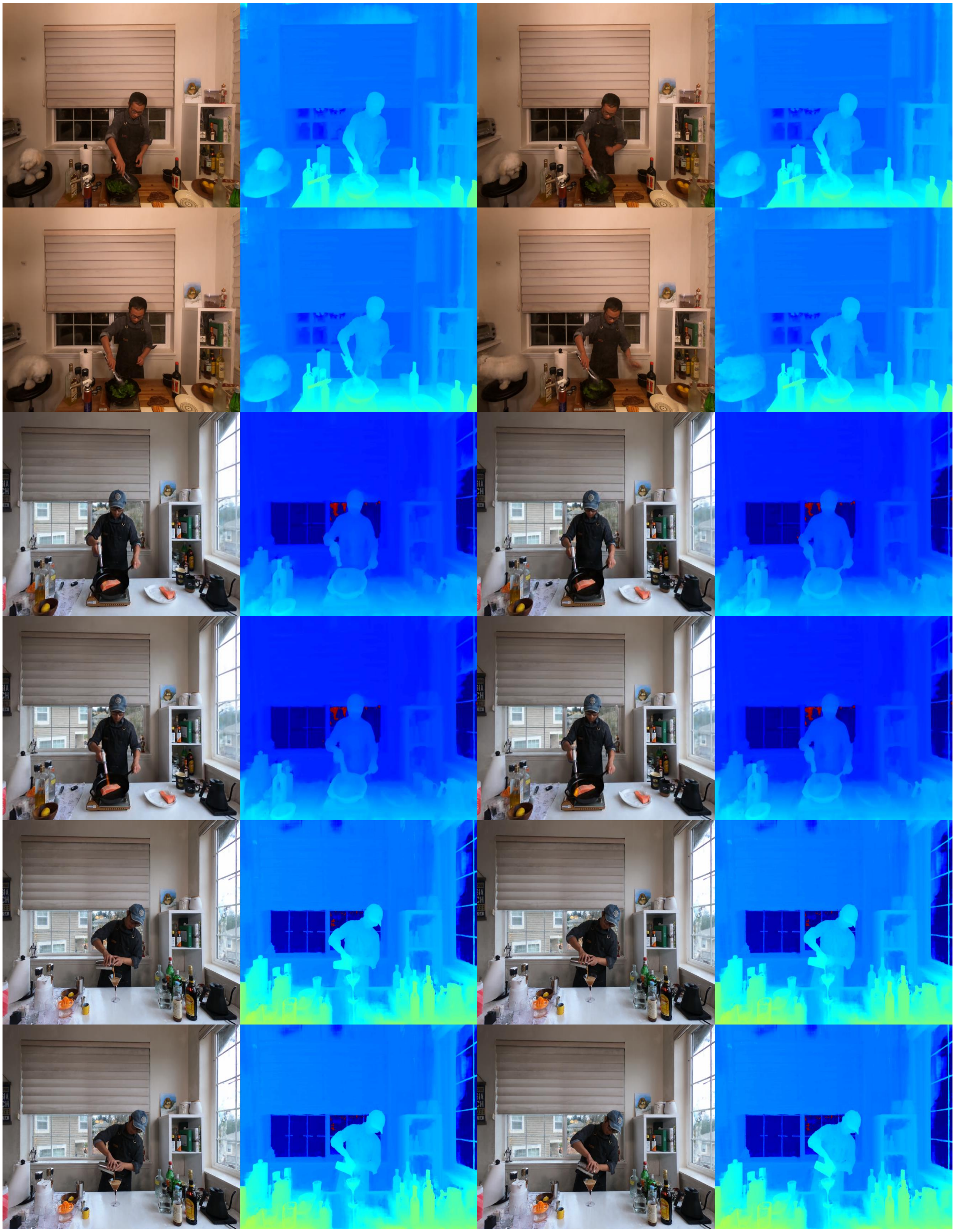

Figure S5. Visualizations on `cook spinach`, `flame salmon` and `coffee martini` scene.

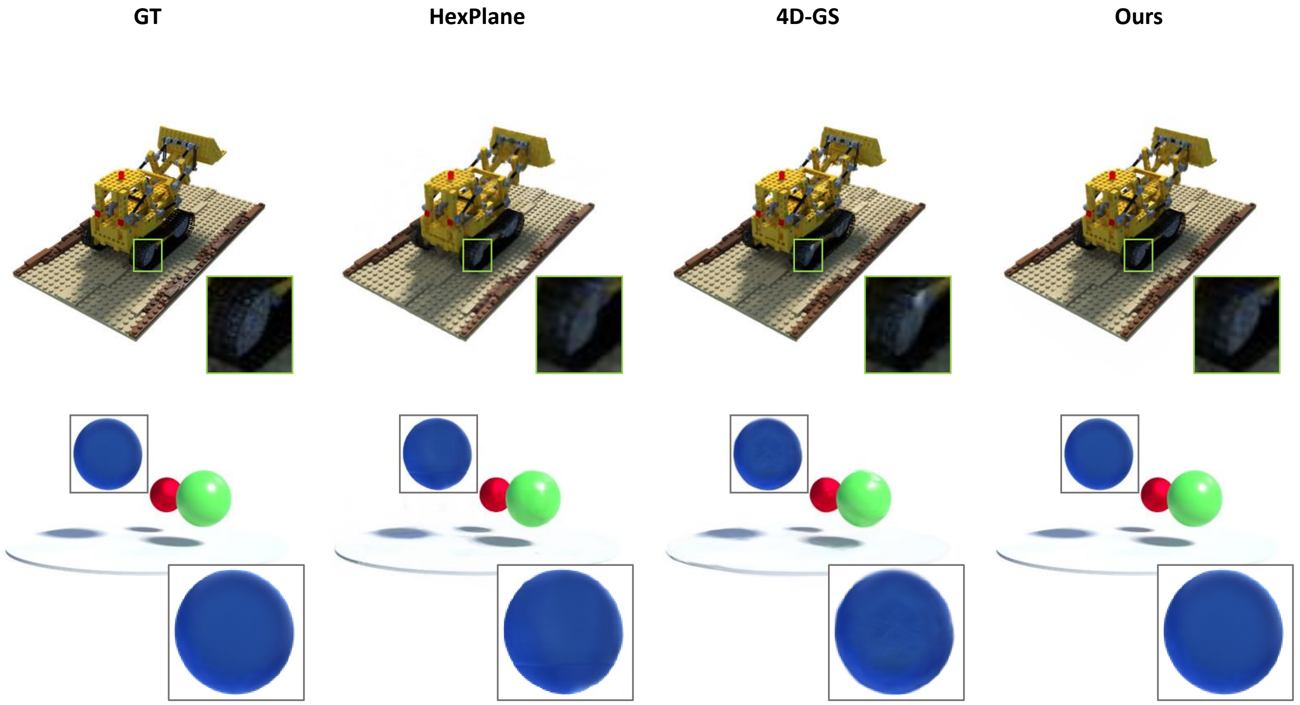

Figure S6. Visual Comparison on Dynamic Scenes (D-NeRF Data). 4D-GS and HexPlane are decomposition-based and deformation-based methods.

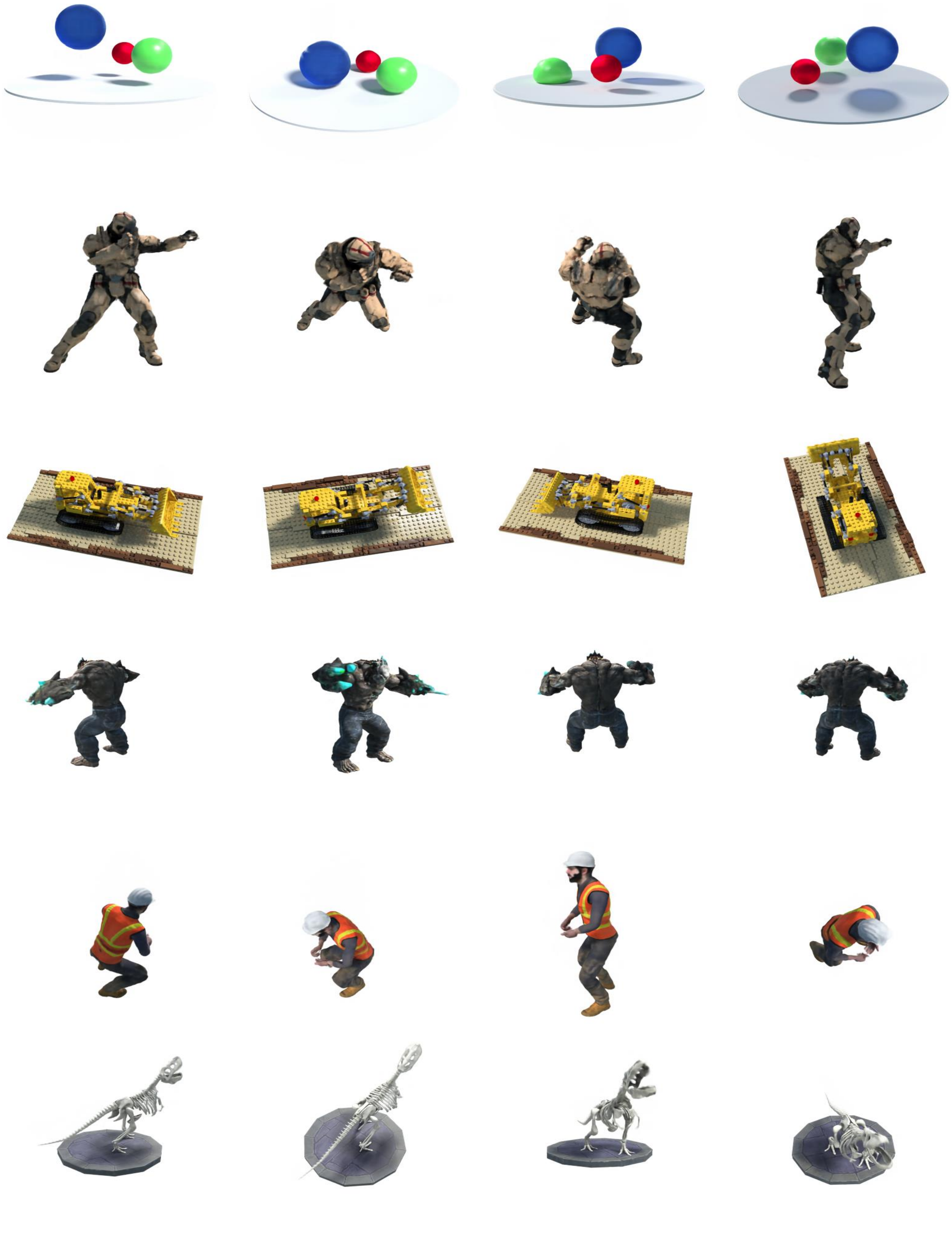

Figure S7. Visualizations on D-NeRF dataset

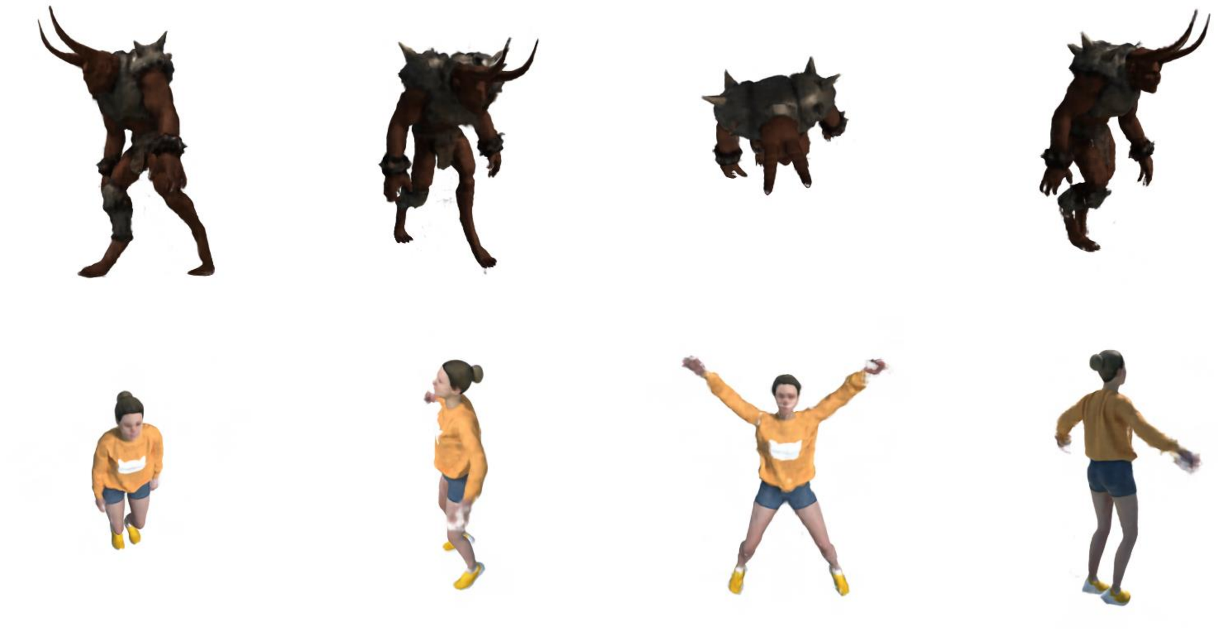

Figure S8. Visualizations on failure cases from D-NeRF dataset

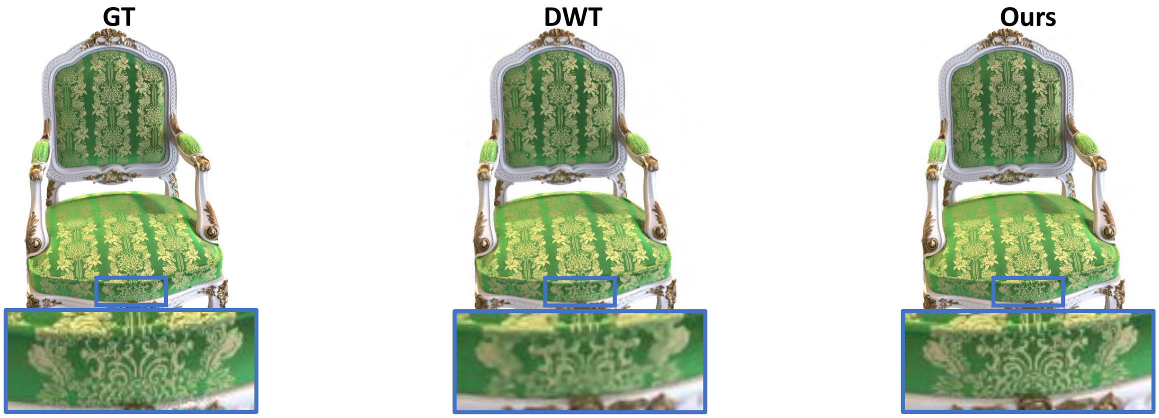

Figure S9. Visual comparison on NeRF synthetic dataset.

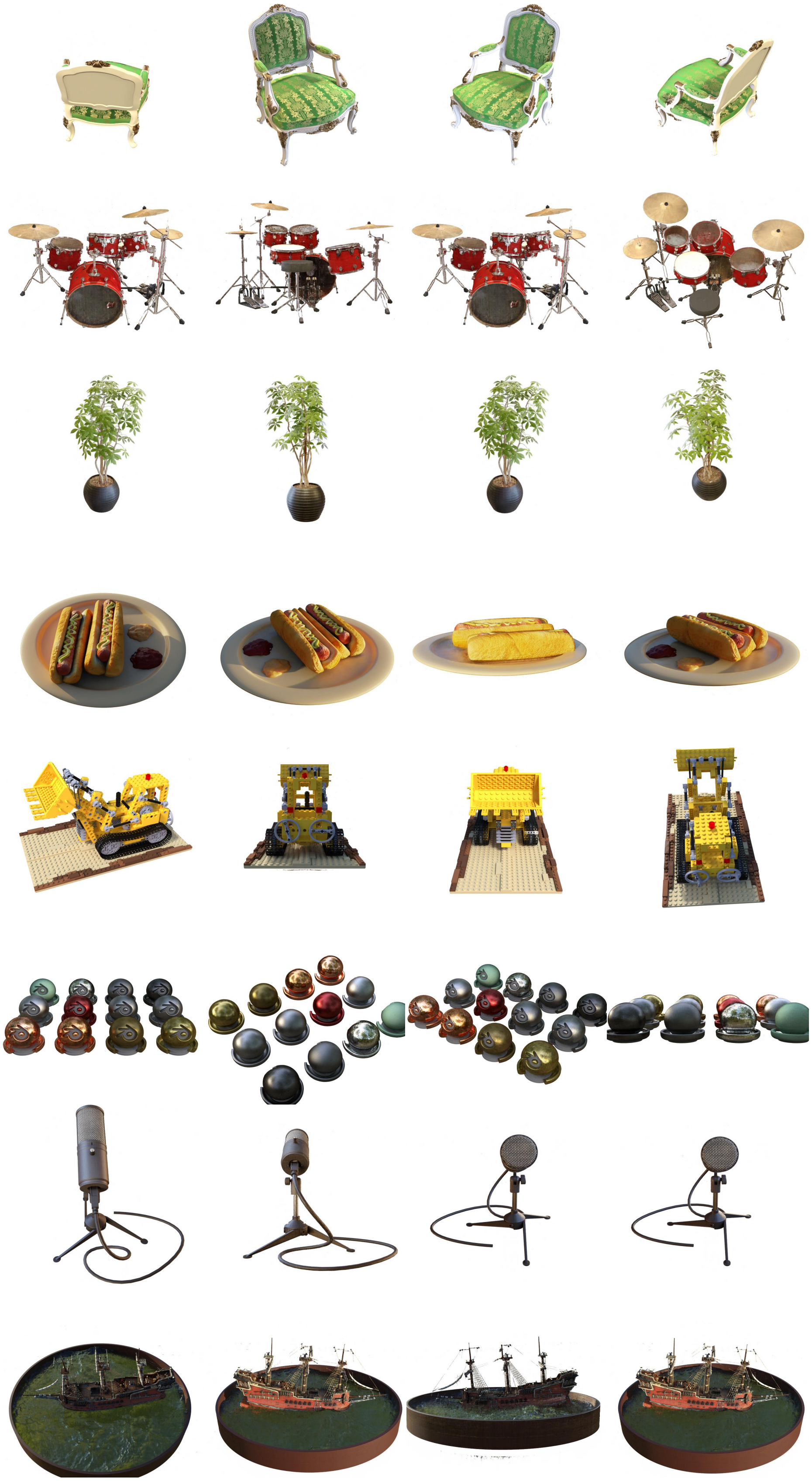

Figure S10. Visualizations on NeRF synthetic dataset.

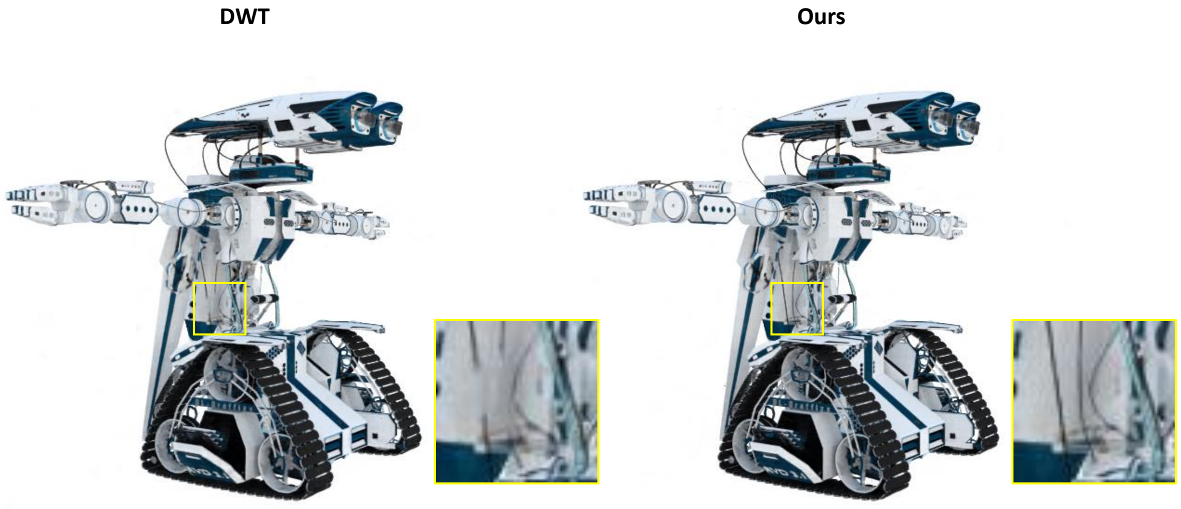

Figure S11. Visual comparison on NSVF synthetic dataset.

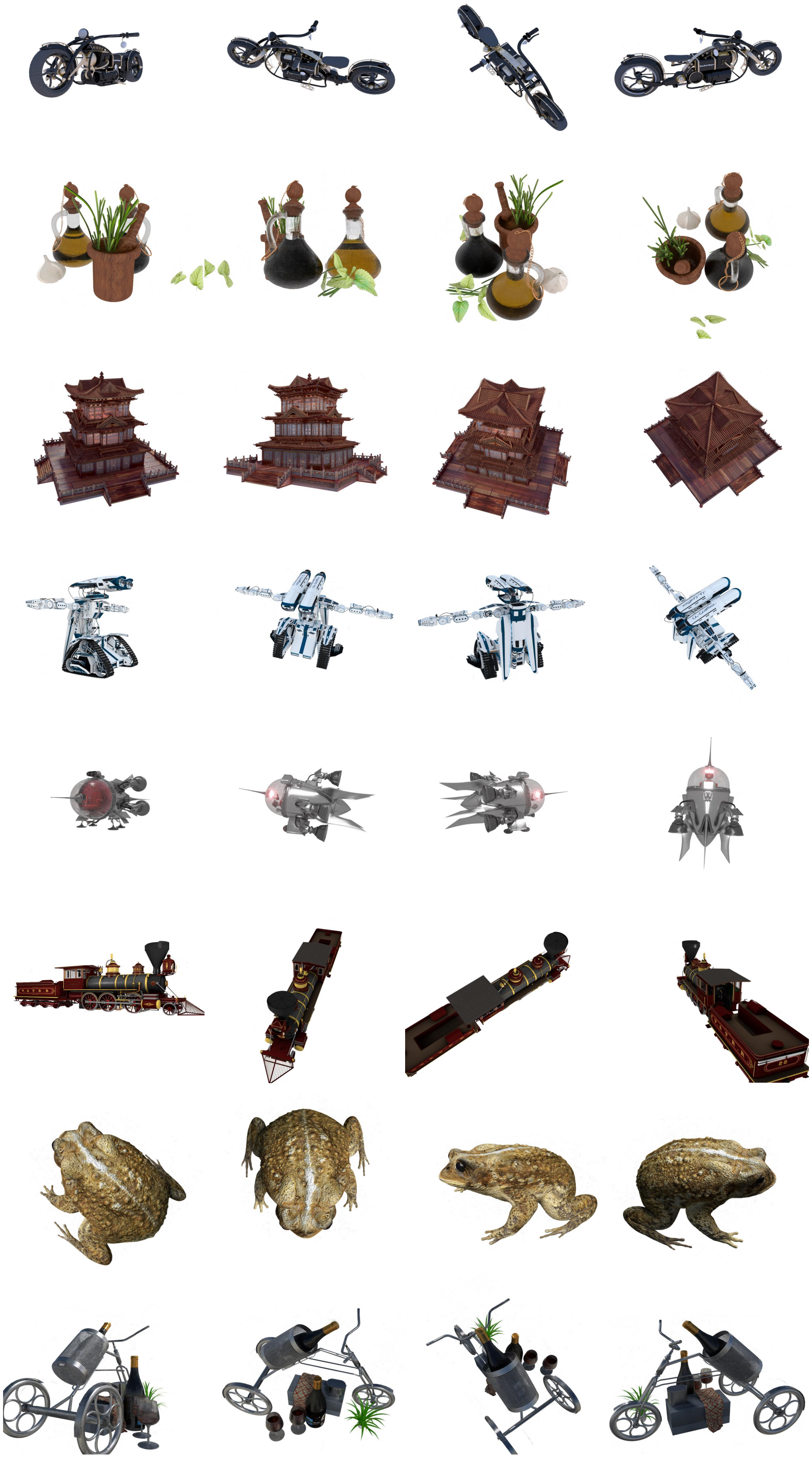

Figure S12. Visualizations on NSVF synthetic dataset.

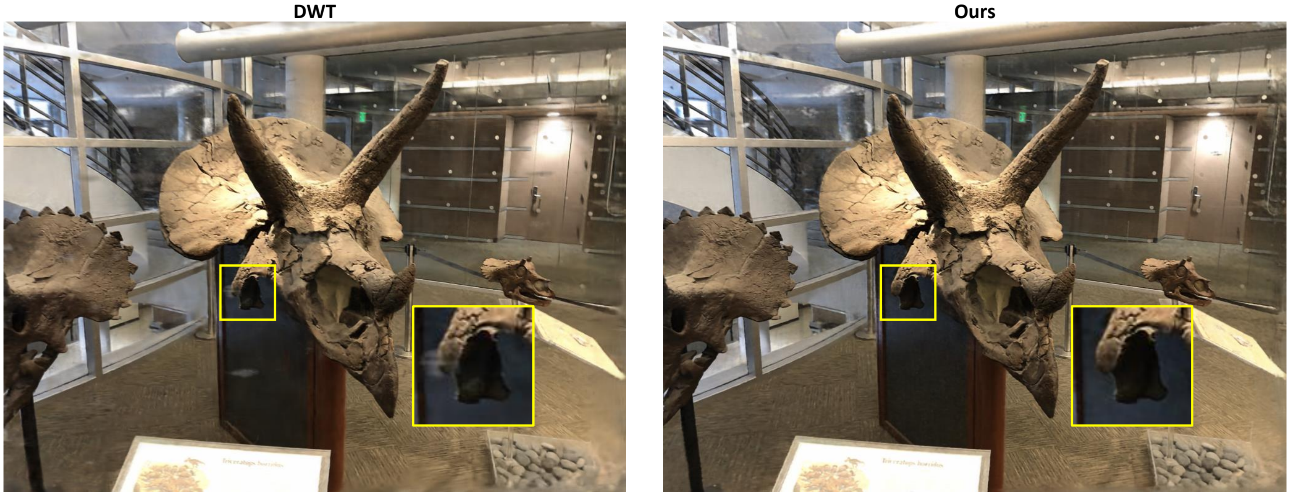

Figure S13. Visual comparison on LLFF synthetic dataset.

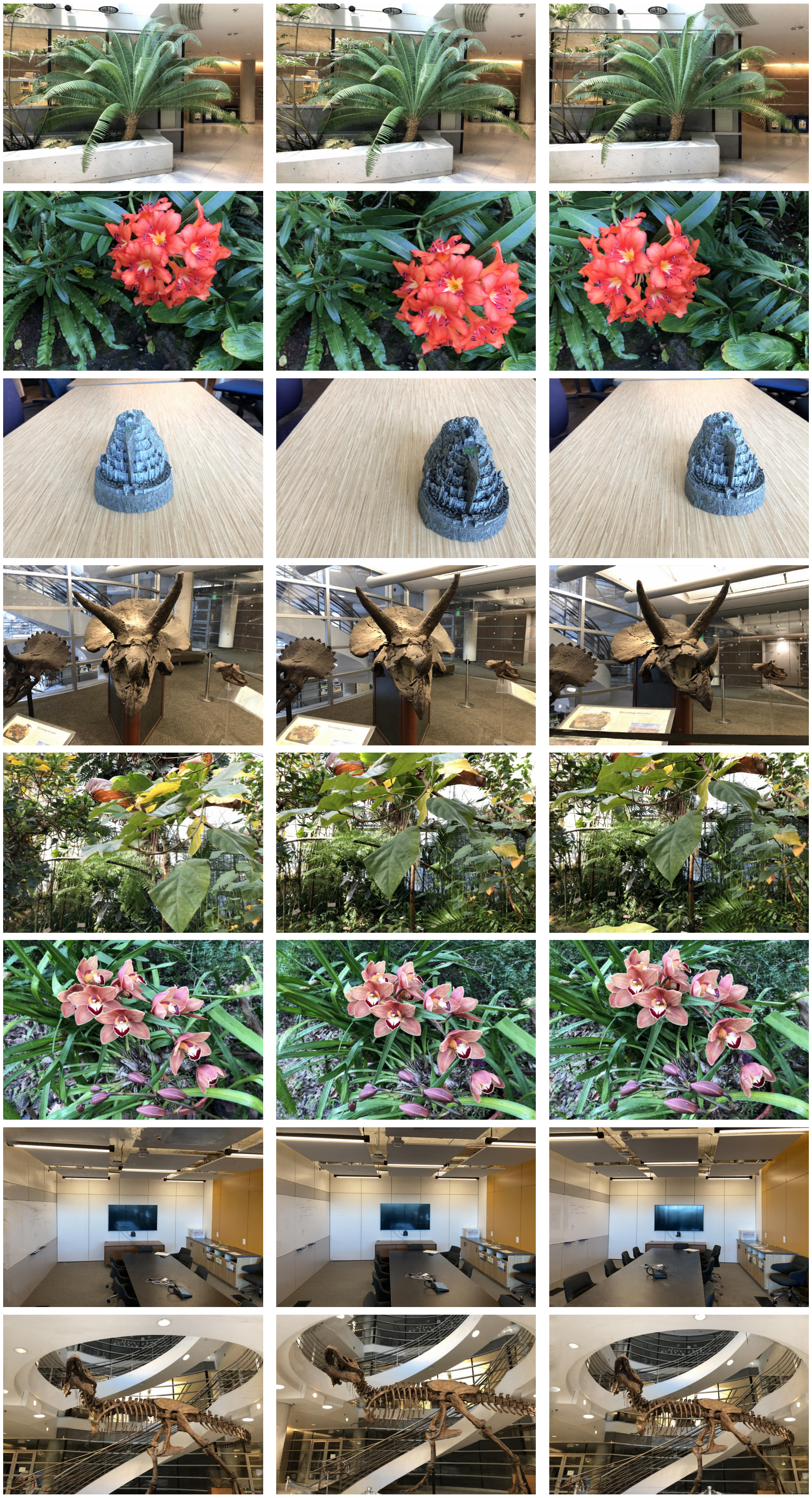

Figure S14. Visualizations on LLFF synthetic dataset.

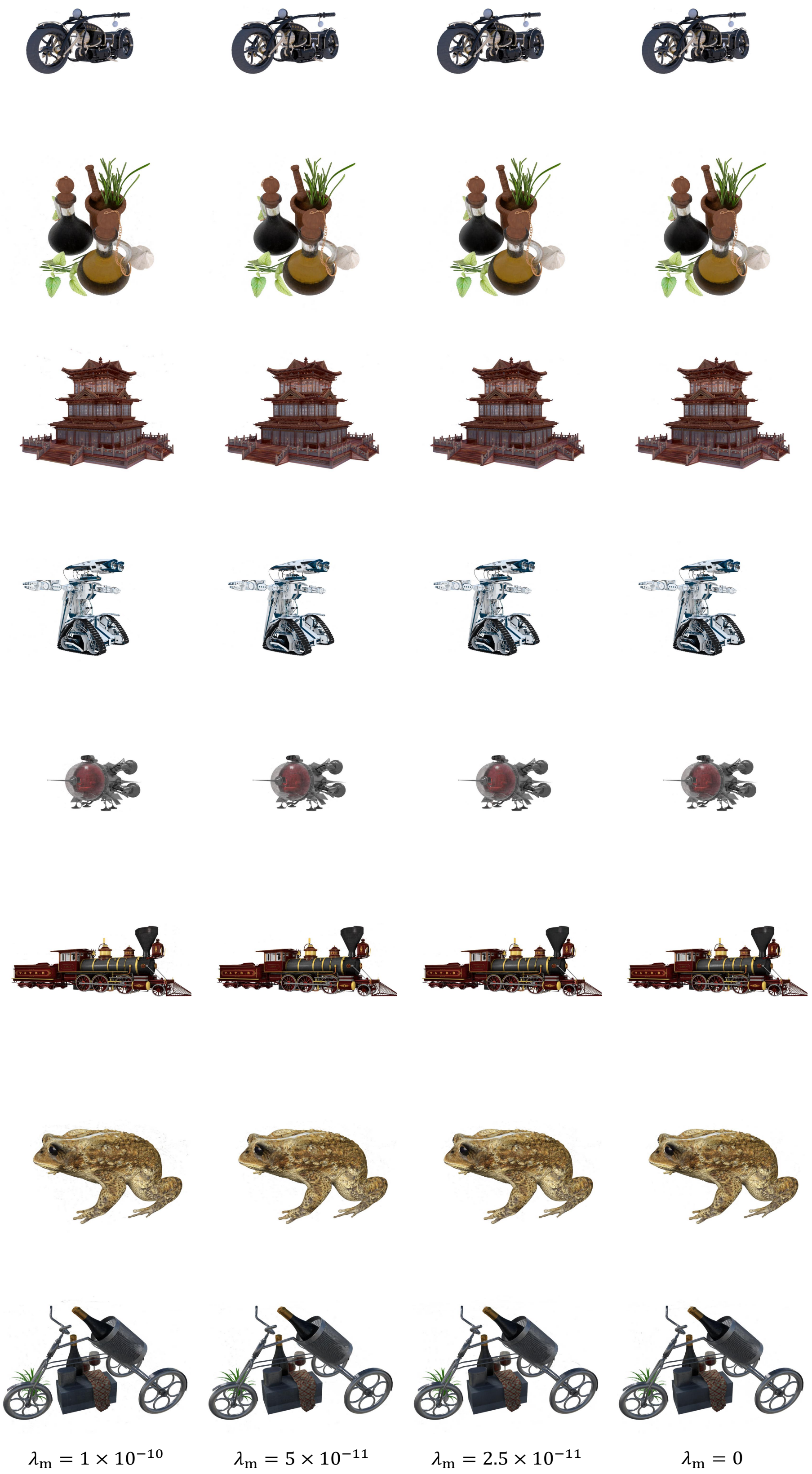
$\lambda_{\mathrm{m}} = 1 \times 10^{-10}$
$\lambda_{\mathrm{m}} = 5 \times 10^{-11}$
$\lambda_{\mathrm{m}} = 2.5 \times 10^{-11}$
$\lambda_{\mathrm{m}} = 0$

Figure S15. Qualitative results on NSVF dataset with different sparsity.