NeuRBF: A Neural Fields Representation with Adaptive Radial Basis Functions (Supplementary Material)

Zhang Chen[1] Zhong Li[1] Liangchen Song[2] Lele Chen[1]
Jingyi Yu[3] Junsong Yuan[2] Yi Xu[1]
[1] OPPO US Research Center [2] University at Buffalo [3] ShanghaiTech University
{zhang.chen,zhong.li,lele.chen,yi.xu}@oppo.com
{lsong8,jsyuan}@buffalo.edu yujingyi@shanghaitech.edu.cn
https://oppo-us-research.github.io/NeuRBF-website/

1. Illustration of the Discontinuity in Adaptive RBFs

As shown in Fig. 1, consider a simple 1D case where $\mathbf{x}_1 \approx \mathbf{x}_2$ are two points located on the boundary where $U(\mathbf{x})$ changes. Let $U(\mathbf{x}_1) = \{1\}$, $U(\mathbf{x}_2) = \{2\}$ be the sets of their closest RBF. From Eq. (2) in the paper, the aggregated neural feature $g_b(\mathbf{x})$ is computed as $g_b(\mathbf{x}) = \sum_{i \in U(\mathbf{x})} \varphi(\mathbf{x}, \mathbf{c}_i, \Sigma_i)\mathbf{w}_i$. Generally, for adaptive RBFs, $\varphi(\mathbf{x}_1, \mathbf{c}_1, \Sigma_1) \not\approx \varphi(\mathbf{x}_2, \mathbf{c}_2, \Sigma_2)$ and $\mathbf{w}_1 \not\approx \mathbf{w}_2$. Therefore, $g_b(\mathbf{x}_1) \not\approx g_b(\mathbf{x}_2)$. This reveals a discontinuity in $g_b(\mathbf{x})$ when $\mathbf{x}$ changes from $\mathbf{x}_1$ to $\mathbf{x}_2$. On the other hand, for grid-based RBFs that use linear interpolation as kernel function, both $\varphi(\mathbf{x}_1, \mathbf{c}_1)$ and $\varphi(\mathbf{x}_2, \mathbf{c}_2)$ are close to 0, so $g_b(\mathbf{x})$ does not contain such discontinuity. We combine adaptive and grid-based RBFs through feature concatenation to balance fitting accuracy and interpolation smoothness.

2. Details on RBF Initialization

We utilize the EM-style Lloyd's K-Means algorithm to initialize RBF positions using all points. The number of RBFs is calculated based on parameter budget. The initialization is conducted only once per scene, before the start of training. We do not split or merge RBFs during training. During weighted K-Means, the initial centers are generated by weighted random sampling. We do not repeat this random sampling for multiple times because we observe it does not have major influence on final performance. The E-M steps are the following:

$$a_{ij} = \begin{cases} 1, \text{if } i = \arg\min_k \|\mathbf{x}_j - \mathbf{c}_k\|^2, \\ 0, \text{otherwise.} \end{cases} \tag{1}$$

$$\mathbf{c}_i = \frac{\sum_j a_{ij} w_j \mathbf{x}_j}{\sum_j a_{ij} w_j}. \tag{2}$$

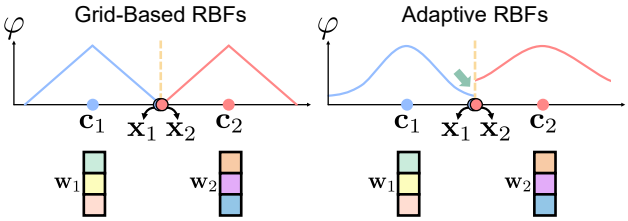

Figure 1. **Illustration of the discontinuity in adaptive RBFs.**

a_{ij} is an indicator variable: $a_{ij} = 1$ if $\mathbf{x}_j$ is assigned to cluster i and $a_{ij} = 0$ otherwise. For efficiency, we iterate Eq. (1)(2) for only 10 steps as the results are already close to convergence and sufficient for our use case. We implement the E-M steps with parallel KD Tree and vectorized centroid update.

3. More Ablation Study

3.1. RBF Initialization

To evaluate the effects of RBF initialization, we compare weighted K-Means with grid initialization, random initialization and weighted random initialization. As shown in Fig. 2, we use an image from DIV2K dataset [2, 17] and conduct 2D image fitting. To facilitate visualization, we only use 15129 RBFs in each baseline. We visualize the position and shape parameters of RBFs as yellow ellipses, and show the fitting error maps and PSNR. As demonstrated in the top two rows, weighted K-Means initialization achieves the highest fitting accuracy. Among the other three baselines, weighted random initialization has a competitve performance while random initialization leads to the worst result.

In the bottom four rows, we further evaluate the effectiveness of using gradient backpropagation to finetune RBF parameters during training. We first use a set of reasonable learning rates for position and shape parameters, which are

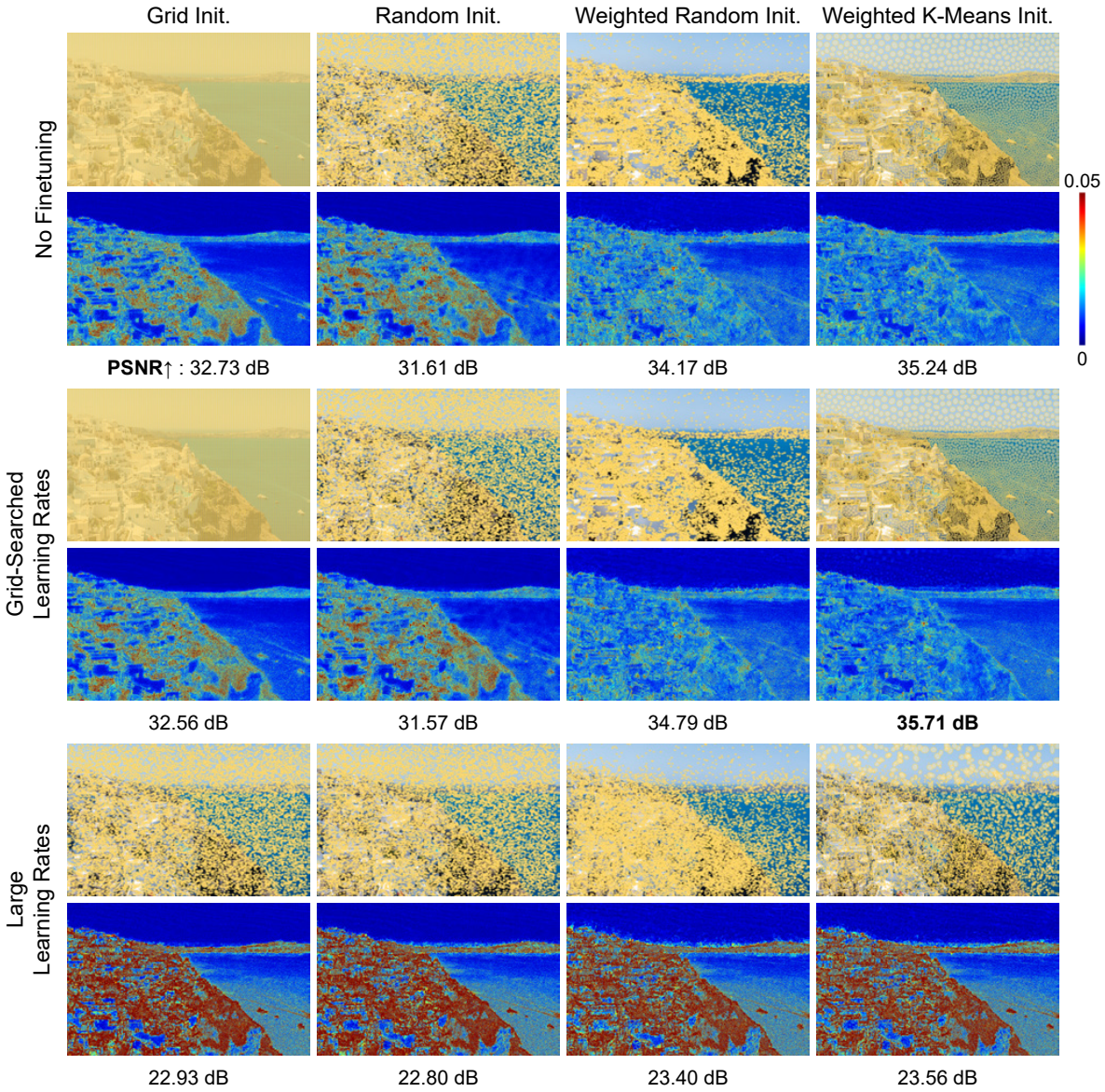

Figure 2. **Evaluation on RBF Initialization.** We compare different RBF initialization methods in columns: grid initialization, random initialization, weighted random initialization and weighted K-Means initialization. We also evaluate different RBF finetuning strategies using gradient backpropagation (with only L2 loss on pixel value) in rows: no finetuning, grid-searched learning rates, large learning rates. For each result, we visualize the RBF parameters as yellow ellipses and show fitting error maps.

obtained through grid search on the baseline with weighted random initialization. As shown in the middle two rows, gradient backpropagation (with only L2 loss on pixel value) only provides minor improvement compared to the first two rows. Besides, the update to the RBF parameters is barely noticeable for grid initialization. Then, we experiment with large learning rates in the last two rows. It can be seen that the RBF parameters can be largely changed from their initialization. However, this leads to significant performance drop for all baselines. The above results validate the benefits of RBF initialization.

3.2. Adaptive Positions and Generalized Interpolation

Here, we evaluate the effects of using adaptive positions for RBFs and generalizing N-dimensional linear interpolation to RBFs with shape parameters. We conduct this ablation study on image, SDF and NeRF tasks, and the results are shown in Fig. 3. The parameter count of each model is 567K, 856K and 17.7M respectively for the three tasks. Based on the results, both adaptive positions and generalized interpolation are beneficial to performance.

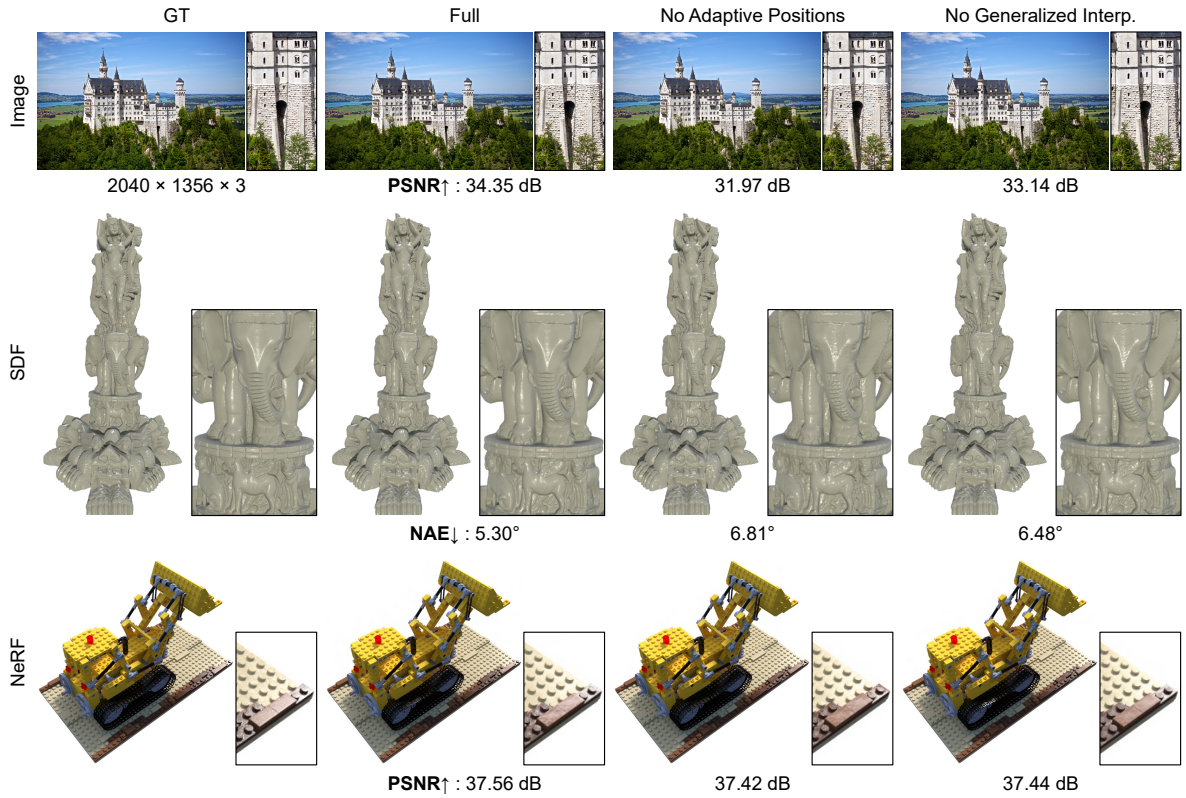

Figure 3. **Evaluation on the adaptive positions and generalized interpolation of RBFs.** "No Adaptive Positions": the positions of RBFs are fixed to a grid structure. "No Generalized Interp.": the interpolation function is N-dimensional linear interpolation.

4. More Implementation Details

Architecture. For the decoder network g_m, except NeRF task, we use a 3-layer MLP (2 hidden layers + 1 output layer) with a network width of 64 neurons, where rectified linear unit (ReLU) activation function is applied to the second hidden layer. The MLP uses a very small part of the parameters (*e.g.*, only 7K in image fitting). For the NeRF experiments on the Synthetic NeRF dataset [12], we use a single Softplus layer as density decoder (same as TensoRF [4]) and use the rendering equation encoding from NRFF [9] as color decoder. For the NeRF experiments on the real LLFF Forward-Facing dataset [11], we adopt the same network architecture as K-Planes-hybrid [7], which uses a 2-layer MLP for density decoder and a 3-layer MLP for color decoder.

Experiments on 2D Image Fitting. The neural features $\mathbf{w}_i$ of adaptive RBFs have a channel dimension of 32. The neighboring RBFs $U(\mathbf{x})$ of a point $\mathbf{x}$ is its 4 nearest neighbors. For Instant NGP [13], we use their official open-sourced codes and hyper-parameters in the comparison experiments. Note that some results in their paper use smaller hash table sizes, hence fewer parameters but also lower PSNR. For MINER [15], we use the implementation from [1]. The original MINER paper does not report their results on the DIV2K dataset [2, 17]. All methods use a batch size of 262144 and are trained for a same number of steps.

Experiments on 3D SDF Reconstruction. The neural features $\mathbf{w}_i$ of adaptive RBFs have a channel dimension of 16, and the size of neighborhood $U(\mathbf{x})$ is 8. We use a grid resolution of 1024^3 for marching cubes. IoU is evaluated on these grid points. For normal angular error (NAE), it is computed similarly as normal consistency [6], but is in unit of degree. Specifically, let P_1, P_2 be randomly sampled points on two mesh surfaces, $\text{NN}(x, P)$ be the closest point to x in point set P, and $n_f(x) \in \mathbb{R}^{3\times1}$ be the unit face normal of point x, NAE is calculated as:

$$\begin{aligned}\text{NAE}(P_1, P_2) = \frac{1}{2} \cdot \frac{180}{\pi} \cdot (\\ \frac{1}{|P_1|} \sum_{p_1 \in P_1} \arccos\big(n_f(p_1)^T n_f(\text{NN}(p_1, P_2))\big) + \\ \frac{1}{|P_2|} \sum_{p_2 \in P_2} \arccos\big(n_f(p_2)^T n_f(\text{NN}(p_2, P_1))\big)\big).\end{aligned} \tag{3}$$

	Avg.	Armadillo	Bunny	Dragon	Buddha	Lucy	XYZ Dragon	Statuette	David	Chameleon	Mechanism
NAE↓											
NGLOD5 [16]	6.58	3.60	4.81	2.85	3.28	4.73	5.66	7.53	3.43	15.91	14.00
NGLOD6 [16]	6.14	3.35	4.47	2.76	3.02	4.28	5.15	6.46	3.22	14.64	14.03
I-NGP [13]	5.70	2.89	**1.96**	2.30	2.73	3.57	4.51	6.00	2.88	11.96	18.21
Ours	**4.93**	**2.83**	2.00	**2.22**	**2.69**	**3.36**	**4.14**	**5.30**	**2.62**	**10.42**	**13.73**
I-NGP$_{400K}$ [13]	6.39	3.20	2.22	2.64	3.18	4.29	4.96	6.82	3.27	13.04	20.31
Ours$_{400K}$	5.53	2.89	2.14	2.35	2.88	3.70	4.44	6.07	2.85	11.96	16.00
IoU↑											
NGLOD5 [16]	0.9962	0.99974	0.97664	0.99964	0.99977	0.99979	0.99981	0.99969	0.99960	0.99456	0.99237
NGLOD6 [16]	0.9963	0.99979	0.97696	0.99969	0.99977	0.99986	0.99983	0.99980	0.99963	0.99528	0.99237
I-NGP [13]	0.9994	**0.99997**	0.99968	**0.99995**	**0.99996**	**0.99997**	**0.99996**	0.99993	**0.99993**	**0.99893**	0.99605
Ours	**0.9995**	0.99994	0.99943	**0.99995**	**0.99996**	0.99996	**0.99996**	**0.99995**	**0.99993**	0.99765	**0.99837**
I-NGP$_{400K}$ [13]	0.9992	0.99995	**0.99974**	0.99994	0.99994	0.99996	0.99995	0.99990	0.99990	0.99820	0.99448
Ours$_{400K}$	0.9994	0.99996	0.99964	**0.99995**	0.99995	**0.99997**	0.99995	0.99991	0.99992	0.99706	0.99767

Table 1. **3D Signed Distance Field Reconstruction.** Per-object breakdown of the quantitative metrics (NAE↓ and IoU↑) in Table 2 of the paper.

Experiments on Neural Radiance Field Reconstruction. For the Synthetic NeRF dataset [12], the channel dimension of adaptive RBFs is 32 and the size of neighborhood $U(\mathbf{x})$ is 5. We first train the grid-based part for 1000 steps, which is then used to distill scene information and conduct RBF initialization. For the real LLFF Forward-Facing dataset [11], the channel dimension of adaptive RBFs is 16 and the size of neighborhood $U(\mathbf{x})$ is 8. The grid-based model used for distillation is trained for 2000 steps while the full model is trained for 38000 steps.

5. Limitations and Future Work

In this work, we have primarily focused on local neural representation. It could be promising to explore the combination with other activation functions in MLP (*e.g.*, WIRE [14]). Besides, in our current implementation, the multipliers $\mathbf{m}, \mathbf{m}_0$ are treated as hyper-parameters and are not trainable. We tried training them along with other parameters, but observed little improvement. A possible reason is that they act as frequencies and would require tailored optimization techniques.

Our method demonstrates high representation accuracy in spatial domains; however, similar to Instant NGP [13] and Factor Fields [5], we have not explored spatial-temporal tasks such as dynamic novel view synthesis. By extending radial basis functions into higher dimensions or using dimension decomposition techniques, our method can potentially be applied to these tasks. We also observe that it is difficult to represent large-scale complicated signals with both high accuracy and small model size, which is a common challenge for local neural fields methods. An interesting future direction would be to design basis functions with more adaptive shapes and long-range support.

6. Additional Results

6.1. 2D Image Fitting

Fig. 4 compares the results on 4 ultra-high resolution images that are not displayed in the paper due to page limit. For the error maps, we calculate the mean absolute error across color channels for each pixel. To highlight the difference among methods, we set the color bar range as $0 \sim 0.01$ (the range of pixel value is $0 \sim 1$).

For the Pluto image (Fig. 4 row 2 in the paper), when fitting the 16 megapixel version of it, our method can reach 44.13 dB PSNR with 7.8M parameters and 50s training.

6.2. 3D SDF Reconstruction

Table 1 shows per-object breakdown of the quantitative metrics (NAE↓ and IoU↑) in Table 2 of the paper. Fig. 5, 6 show the qualitative results, where the numbers of trainable parameters for Instant NGP and ours are 950K and 856K.

We further compare with BACON [10] and let our method use the same training settings as them. BACON uses 531K parameters while our models only use 448K. Averaging over 4 scenes (Armadillo, Lucy, XYZ Dragon, Statuette), the normal angular errors (NAE↓) are 5.89°(BACON) vs. **4.53°(Ours)**.

6.3. Neural Radiance Field Reconstruction

Table 2 and 3 demonstrate the per-scene quantitative comparisons (PSNR↑, SSIM↑, LPIPS$_{VGG}$↓, LPIPS$_{Alex}$↓) on the Synthetic NeRF dataset [12] and the real LLFF Forward-Facing dataset [11]. Fig. 7 and Fig. 8 show more close-up and full-image comparisons on the Synthetic NeRF dataset [12]. Fig. 9 shows full-image comparisons on the real LLFF Forward-Facing dataset [11].

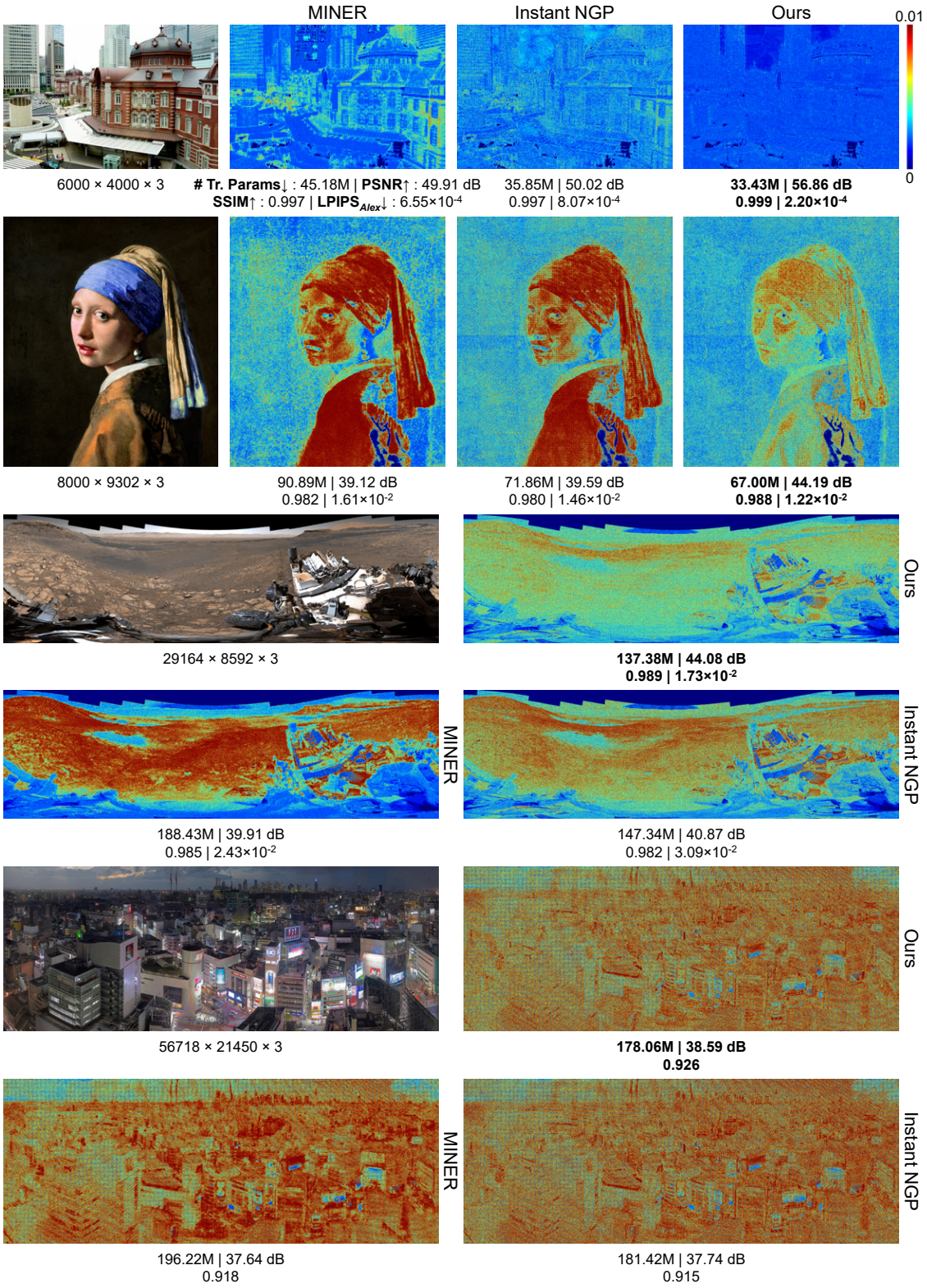

Figure 4. **2D Image Fitting.** Leftmost column or top left quarter shows the fitted images of our method and the resolution of the images. The other columns or quarters show the error maps of each method, along with the number of trainable parameters ("# Tr. Params")↓, PSNR↑, SSIM↑ and LPIPS$_{Alex}$↓. For the last image, its resolution is too high to compute LPIPS$_{Alex}$. "Girl With a Pearl Earring" renovation ©Koorosh Orooj (CC BY-SA 4.0).

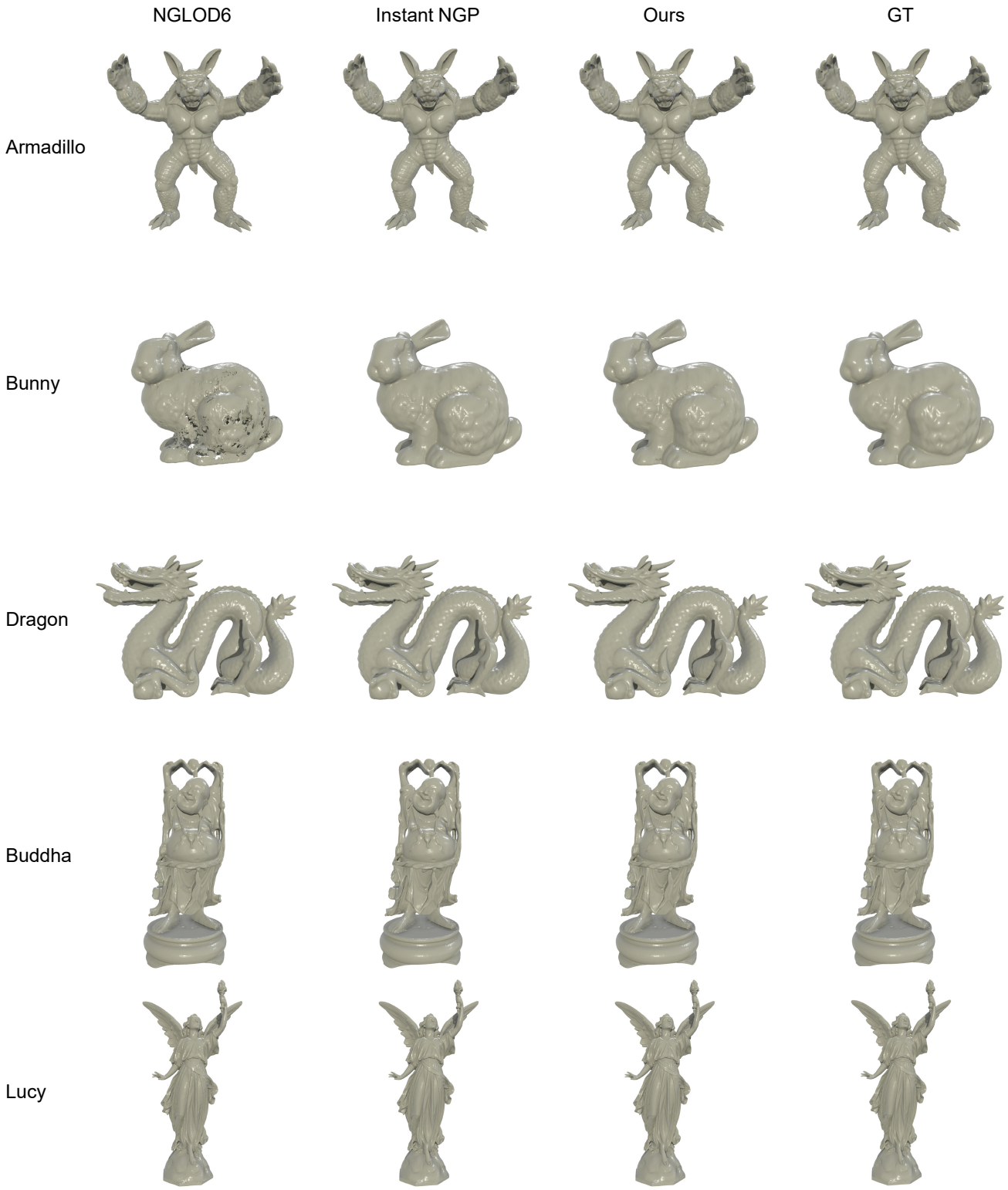

Figure 5. **3D SDF Reconstruction.** Qualitative comparisons between NGLOD6 [16], Instant NGP [13] and ours. For the results in this figure, the number of trainable parameters of Instant NGP is 950K, while that of ours is 856K. (To be continued in the next page.)

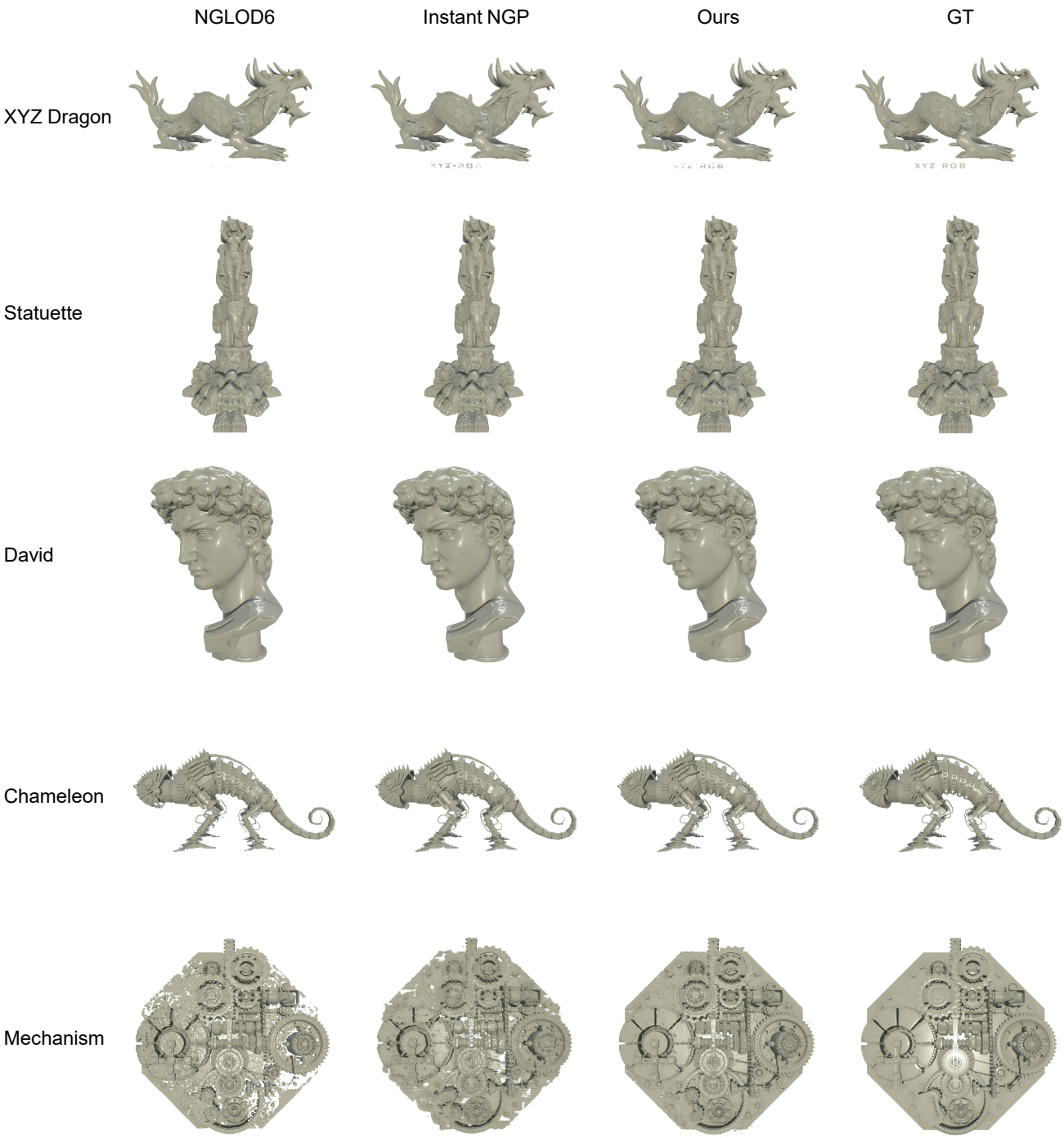

Figure 6. **3D SDF Reconstruction.** Qualitative comparisons between NGLOD6 [16], Instant NGP [13] and ours. For the results in this figure, the number of trainable parameters of Instant NGP is 950K, while that of ours is 856K.

Methods	Avg.	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
PSNR↑									
NeRF [12]	31.01	33.00	25.01	30.13	36.18	32.54	29.62	32.91	28.65
Mip-NeRF 360 [3]	33.25	-	-	-	-	-	-	-	-
Point-NeRF [18]	33.31 ●	35.40	26.06 ●	36.13 ●	37.30	35.04	29.61	35.95 ●	30.97
Plenoxels [8]	31.71	33.98	25.35	31.83	36.43	34.10	29.14	33.26	29.62
Instant NGP [13]	33.18	35.00	26.02	33.51	37.40	36.39	29.78	36.22 ●	31.10 ●
TensoRF [4]	33.14	35.76 ●	26.01	33.99	37.41 ●	36.46 ●	30.12 ●	34.61	30.77
Factor Fields [5]	33.14	-	-	-	-	-	-	-	-
K-Planes [7]	32.36	34.99	25.66	31.41	36.78	35.75	29.48	34.05	30.74
Ours	34.62 ●	36.74 ●	26.47 ●	35.14 ●	38.65 ●	37.53 ●	34.30 ●	36.17 ●	31.94 ●
Ours$_{3.66M}$	33.97 ●	35.82 ●	26.19 ●	34.08 ●	38.11 ●	36.75 ●	34.32 ●	35.49	31.03 ●
SSIM↑									
NeRF [12]	0.947	0.967	0.925	0.964	0.974	0.961	0.949	0.980	0.856
Mip-NeRF 360 [3]	0.962	-	-	-	-	-	-	-	-
Point-NeRF [18]	0.962	0.984 ●	0.935	0.987 ●	0.982 ●	0.978	0.948	0.990 ●	0.892
Plenoxels [8]	0.958	0.977	0.933	0.976	0.980	0.976	0.949	0.985	0.890
Instant NGP [13]	0.963 ●	0.985 ●	0.940 ●	0.982 ●	0.982 ●	0.982	0.949	0.989 ●	0.893
TensoRF [4]	0.963 ●	0.985 ●	0.937	0.982 ●	0.982 ●	0.983 ●	0.952 ●	0.988	0.895
Factor Fields [5]	0.961	-	-	-	-	-	-	-	-
K-Planes [7]	0.962	0.983	0.938	0.975	0.982 ●	0.982	0.950 ●	0.988	0.897 ●
Ours	0.975 ●	0.988 ●	0.946 ●	0.987 ●	0.987 ●	0.986 ●	0.980 ●	0.992 ●	0.930 ●
Ours$_{3.66M}$	0.971 ●	0.985 ●	0.942 ●	0.984 ●	0.985 ●	0.984 ●	0.980 ●	0.990 ●	0.919 ●
LPIPS$_{VGG}$↓									
NeRF [12]	0.081	0.046	0.091	0.044	0.121	0.050	0.063	0.028	0.206
Mip-NeRF 360 [3]	0.039 ●	-	-	-	-	-	-	-	-
Point-NeRF [18]	0.050	0.023	0.078	0.022 ●	0.037	0.024	0.072	0.014 ●	0.124 ●
Plenoxels [8]	0.049	0.031	0.067 ●	0.026	0.037	0.028	0.057 ●	0.015	0.134 ●
Instant NGP [13]	0.051	0.023	0.076	0.027	0.038	0.021 ●	0.065	0.020	0.137
TensoRF [4]	0.047 ●	0.022 ●	0.073	0.022 ●	0.032 ●	0.018 ●	0.058	0.015	0.138
Factor Fields [5]	-	-	-	-	-	-	-	-	-
K-Planes [7]	0.062	0.027	0.089	0.056	0.034	0.047	0.068	0.029	0.148
Ours	0.034 ●	0.015 ●	0.059 ●	0.014 ●	0.021 ●	0.015 ●	0.031 ●	0.008 ●	0.110 ●
Ours$_{3.66M}$	0.039 ●	0.019 ●	0.065 ●	0.019 ●	0.025 ●	0.018 ●	0.034 ●	0.010 ●	0.124 ●
LPIPS$_{Alex}$↓									
Point-NeRF [18]	0.028	0.010	0.055	0.009 ●	0.016	0.011	0.041	0.007 ●	0.070 ●
Instant NGP [13]	0.028	0.0097 ●	0.0540	0.0174	0.0142	0.0085 ●	0.0296	0.0072	0.0863
TensoRF [4]	0.027 ●	0.010	0.051 ●	0.012	0.013 ●	0.007 ●	0.026 ●	0.009	0.085
K-Planes [7]	0.031	0.0125	0.0527	0.0209	0.0170	0.0096	0.0303	0.0091	0.0968
Ours	0.018 ●	0.0067 ●	0.0409 ●	0.0085 ●	0.0085 ●	0.0057 ●	0.0106 ●	0.0044 ●	0.0614 ●
Ours$_{3.66M}$	0.022 ●	0.0088 ●	0.0454 ●	0.0101 ●	0.0109 ●	0.0070 ●	0.0119 ●	0.0060 ●	0.0735 ●

Table 2. **Neural Radiance Field Reconstruction.** Per-scene quantitative comparisons (PSNR↑, SSIM↑, LPIPS$_{VGG}$↓, LPIPS$_{Alex}$↓) on the Synthetic NeRF dataset [12]. Best 3 scores in each scene are marked with gold ●, silver ● and bronze ●. "-" denotes scores that are unavailable in prior work. For LPIPS$_{Alex}$, since the scores of NeRF [12], Mip-NeRF 360 [3], Plenoxels [8] and Factor Fields [5] are unavailable in prior work, we exclude these methods in this metric.

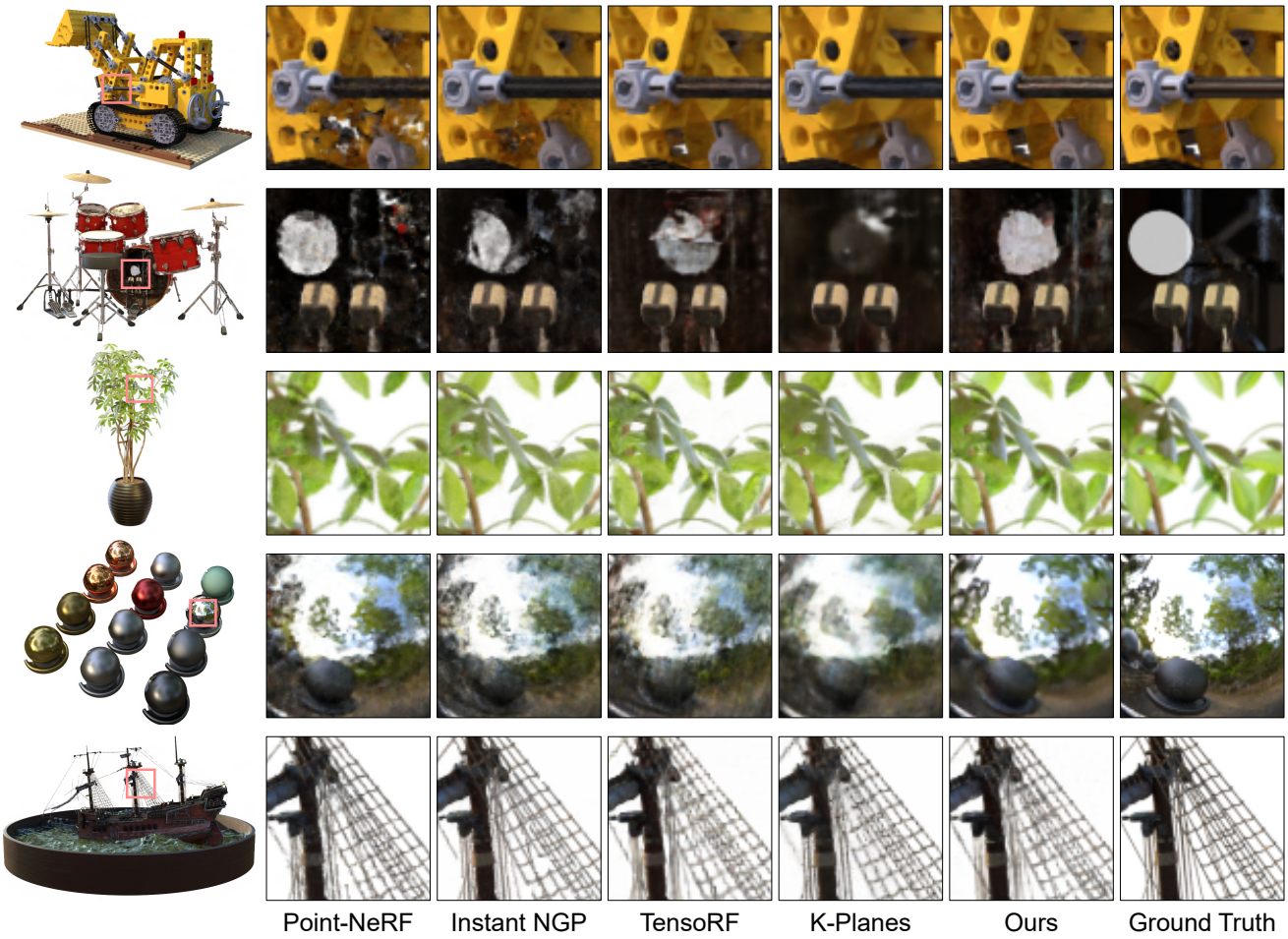

Figure 7. **Neural Radiance Field Reconstruction.** More close-up comparisons on the Synthetic NeRF Dataset [12]. Leftmost column shows the full-image results of our method.

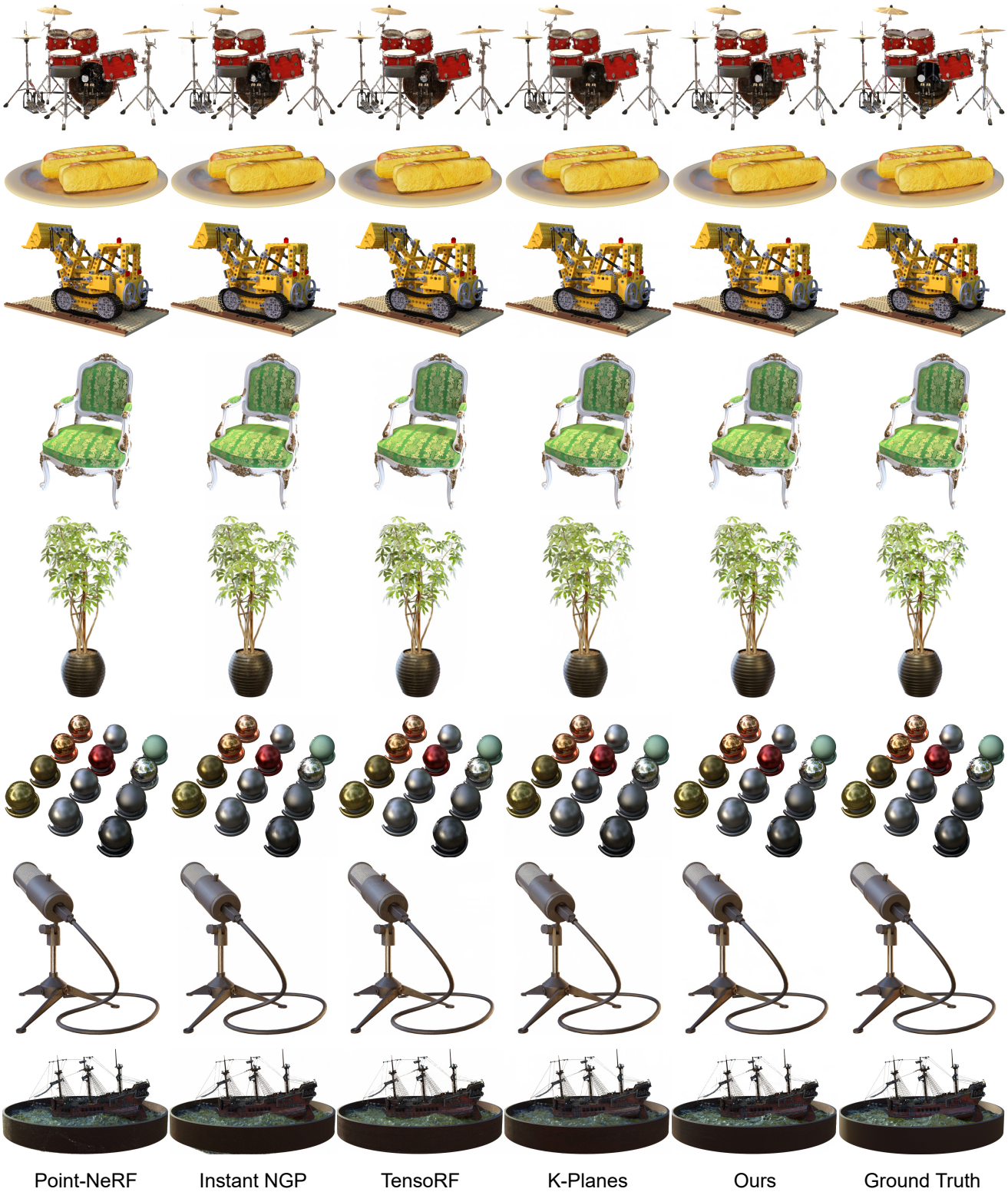

Figure 8. **Neural Radiance Field Reconstruction.** Full-image comparisons on the Synthetic NeRF Dataset [12].

Methods	Avg.	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns
PSNR↑									
NeRF [12]	26.50	32.70 ●	25.17	20.92	31.16 ●	20.36 ●	27.40	26.80	27.45
Mip-NeRF 360 [3]	26.86 ●	-	-	-	-	-	-	-	-
Plenoxels [8]	26.29	30.22	25.46 ●	21.41 ●	31.09 ●	20.24 ●	27.83 ●	26.48	27.58
TensoRF [4]	26.73	32.35	25.27	21.30 ●	31.36 ●	19.87	28.60 ●	26.97 ●	28.14 ●
K-Planes [7]	26.92 ●	32.64 ●	25.38 ●	21.30 ●	30.44	20.26 ●	28.67 ●	28.01 ●	28.64 ●
Ours	27.05 ●	32.80 ●	25.48 ●	21.81 ●	30.98	20.03	28.57	28.06 ●	28.68 ●
SSIM↑									
NeRF [12]	0.811	0.948	0.792	0.690	0.881	0.641	0.827	0.880	0.828
Mip-NeRF 360 [3]	0.858 ●	-	-	-	-	-	-	-	-
Plenoxels [8]	0.839	0.937	0.832 ●	0.760 ●	0.885	0.687 ●	0.862	0.890	0.857
TensoRF [4]	0.839	0.952 ●	0.814	0.752 ●	0.897 ●	0.649	0.871 ●	0.900 ●	0.877 ●
K-Planes [7]	0.847 ●	0.957 ●	0.828 ●	0.746	0.890 ●	0.676 ●	0.872 ●	0.915 ●	0.892 ●
Ours	0.849 ●	0.955 ●	0.822 ●	0.769 ●	0.891 ●	0.675 ●	0.868 ●	0.916 ●	0.895 ●
LPIPS$_{VGG}$↓									
NeRF [12]	0.250	0.178	0.280	0.316	0.171	0.321	0.219	0.249	0.268
Plenoxels [8]	0.210	0.192	0.224 ●	0.198 ●	0.180	0.242 ●	0.179	0.238	0.231
TensoRF [4]	0.204 ●	0.167 ●	0.237	0.217 ●	0.148 ●	0.278	0.169 ●	0.221 ●	0.196 ●
K-Planes [7]	0.194 ●	0.147 ●	0.223 ●	0.242	0.154 ●	0.250 ●	0.165 ●	0.199 ●	0.173 ●
Ours	0.179 ●	0.134 ●	0.209 ●	0.238 ●	0.128 ●	0.271 ●	0.147 ●	0.158 ●	0.149 ●
LPIPS$_{Alex}$↓									
TensoRF [4]	0.124 ●	0.082 ●	0.155 ●	0.153 ●	0.075 ●	0.201 ●	0.106 ●	0.099 ●	0.123 ●
K-Planes [7]	0.102 ●	0.066 ●	0.130 ●	0.153 ●	0.068 ●	0.151 ●	0.088 ●	0.071 ●	0.092 ●
Ours	0.090 ●	0.059 ●	0.111 ●	0.127 ●	0.056 ●	0.160 ●	0.072 ●	0.057 ●	0.075 ●

Table 3. **Neural Radiance Field Reconstruction.** Per-scene quantitative comparisons (PSNR↑, SSIM↑, LPIPS$_{VGG}$↓, LPIPS$_{Alex}$↓) on the real LLFF Forward-Facing dataset [11]. Best 3 scores in each scene are marked with gold ●, silver ● and bronze ●. "-" denotes scores that are unavailable in prior work. For LPIPS$_{Alex}$, since the scores of NeRF [12], Mip-NeRF 360 [3] and Plenoxels [8] are unavailable in prior work, we exclude these methods in this metric.

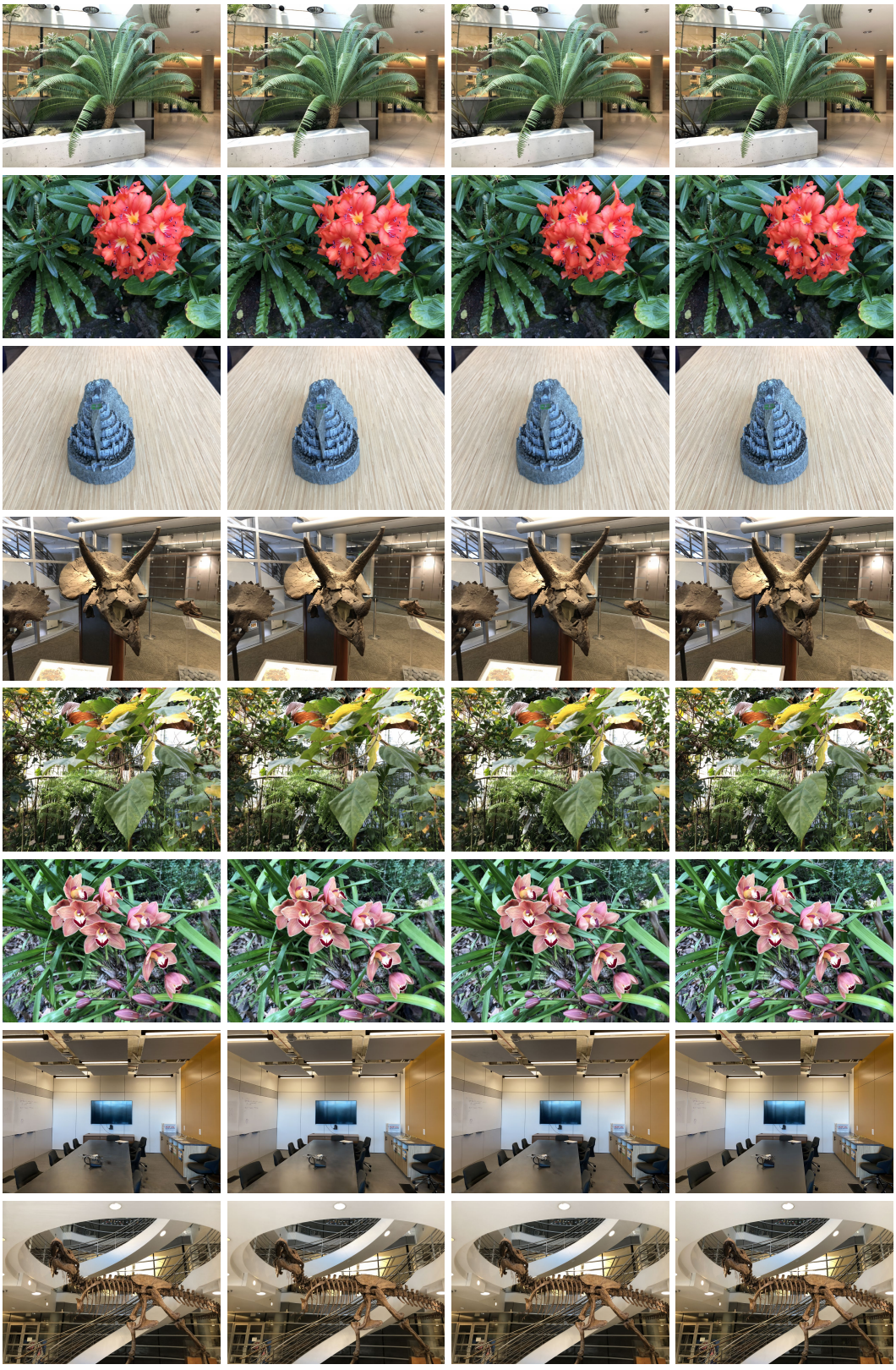

Figure 9. **Neural Radiance Field Reconstruction.** Full-image comparisons on the real LLFF Forward-Facing dataset [11].

References

[1] https://github.com/kwea123/MINER_pl. 3

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 1, 3

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 8, 11

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022. 3, 8, 11

[5] Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Factor fields: A unified framework for neural fields and beyond. *arXiv preprint arXiv:2302.01226*, 2023. 4, 8

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6970–6981, 2020. 3

[7] 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. 3, 8, 11

[8] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 8, 11

[9] Kang Han and Wei Xiang. Multiscale tensor decomposition and rendering equation encoding for view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4232–4241, 2023. 3

[10] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16252–16262, 2022. 4

[11] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3, 4, 11, 12

[12] 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, 4, 8, 9, 10, 11

[13] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 3, 4, 6, 7, 8

[14] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023. 4

[15] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pages 318–333. Springer, 2022. 3

[16] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 4, 6, 7

[17] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 1, 3

[18] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 8