

# F<sup>2</sup>-NeRF: Fast Neural Radiance Field Training with Free Camera Trajectories

## - Supplementary -

Peng Wang<sup>1,2\*</sup> Yuan Liu<sup>1\*</sup> Zhaoxi Chen<sup>2</sup> Lingjie Liu<sup>3</sup>  
 Ziwei Liu<sup>2</sup> Taku Komura<sup>1</sup> Christian Theobalt<sup>3</sup> Wenping Wang<sup>4</sup>

<sup>1</sup> The University of Hong Kong <sup>2</sup> S-Lab, Nanyang Technological University

<sup>3</sup> Max Planck Institute for Informatics <sup>4</sup> Texas A&M University

### 1. Additional Implementation Details

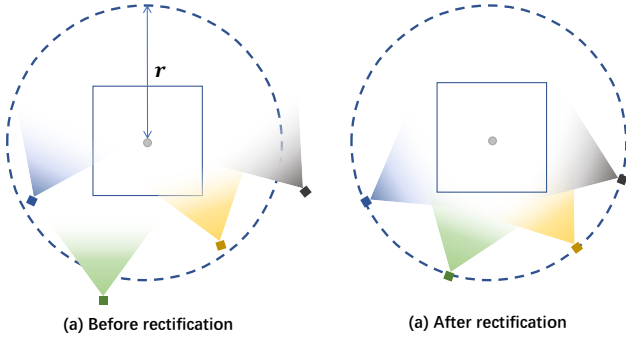


Figure 1. Camera rectification.

#### 1.1. Camera processing

**Camera rectification.** After the procedure of space-subdivision (Sec 3.3 of the main paper), for each sub-region  $S$ , we obtain a set of visible cameras whose view frustums intersect with  $S$ . We note that visible cameras are not suitable to be directly used in computing of the perspective warping, because some cameras do not fully cover the region and look at the region as shown in Fig. 1(a). Thus, we propose an empirical but effective camera rectification strategy, that we rotate the camera view directions to make them look at the center of the region  $S$ . This simple strategy helps ensure most points inside  $S$  can be warped to meaningful coordinates. Moreover, we find that aligning their distances to the center of  $S$  with the same distance  $r$  can help improve the rendering quality, as shown in Fig. 3. Here  $r$  is empirically set as the mean distance to the region center among the 1/4 nearest visible cameras.

**Camera selection.** When the number of visible cameras is larger than  $n_c = 4$ , we select a subset of the visible cameras for better computational efficiency. We select the

\*Equal contribution.

cameras based on the farthest point sampling: First, we randomly select a camera as the seed, and then we repeatedly add the farthest visible camera for  $n_c - 1$  times.



Figure 2. The effect of distance alignment.

#### 1.2. Construction of $M$

In this section, we give a detailed description of how we construct the matrix  $M$  for our perspective warping.

**Principal component analysis.** Given the region  $S$  with the selected cameras, we first uniformly sample  $n = 32^3$  points  $\{\mathbf{x}_i\}$  inside  $S$ . Then, we project the points to the selected cameras, concatenate the projected coordinates, and obtain the high-dimensional coordinates  $\{[C_1(\mathbf{x}_i), \dots, C_{n_c}(\mathbf{x}_i)] = [u_1^i, v_1^i, \dots, u_{n_c}^i, v_{n_c}^i]\}$ . These coordinates are formed in a coordinate matrix  $K \in \mathbb{R}^{2n_c \times n}$ , then we compute the covariance matrix  $Q = (K - \bar{K})(K - \bar{K})^\top$ , where  $\bar{K}$  is the mean coordinate of all projected coordinates. By eigendecomposition, we obtain the matrix  $M' \in \mathbb{R}^{3 \times 2n_c}$  formed by the eigenvectors with the first three largest eigenvalues. The matrix  $M'$  defines the directions of the projection axes.

**Computing the axis length.** After the matrix  $M' \in \mathbb{R}^{3 \times 2n_c}$  is found, we now need to perform a post normalization by scaling each axis. Specifically, we would like to find three proper scale parameters  $\{s_1, s_2, s_3\}$ , and  $M = SM'$ , where  $S \in \mathbb{R}^3$  is the diagonal matrix formed by  $\{s_1, s_2, s_3\}$ . The key idea of these scaling parameters is that we expect the unit length in the warp space can be approximately aligned with the unit length in the image space. More specifically,

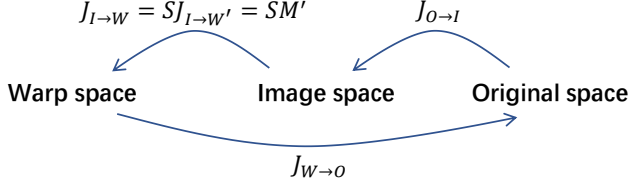


Figure 3. The Jacobian matrices among different spaces.

for each axis in the warp space, when a point moves along the axis by a unit length, we expect the maximum spatial transition of all image coordinates to be approximately a pixel length.

We take a point  $\mathbf{x}$  inside the region  $S$  for example. Let us denote the Jacobian matrix from the original space to the image space as  $J_{O \rightarrow I} \in \mathbb{R}^{3 \times 2n_c}$  derived from the image projection function, and the Jacobian matrix from the image space to the warp space as  $J_{I \rightarrow W} = M = S M'$ . Our target is to compute the Jacobian matrix  $J_{W \rightarrow I} \in \mathbb{R}^{2n_c \times 3}$  and put a constraint that the maximum value of each column vector of  $J_{W \rightarrow I}$  equals one. Note  $J_{W \rightarrow I}$  may not be directly computed by inverting  $J_{I \rightarrow W}$ , which is not a square matrix. Alternatively, we present it by  $J_{W \rightarrow I} = J_{O \rightarrow I} J_{W \rightarrow O}$ , which can be further represented by

$$\begin{aligned} J_{W \rightarrow I} &= J_{O \rightarrow I} J_{O \rightarrow W}^{-1} \\ &= J_{O \rightarrow I} (J_{I \rightarrow W} J_{O \rightarrow I})^{-1} \\ &= J_{O \rightarrow I} (S M' J_{O \rightarrow I})^{-1} \\ &= J_{O \rightarrow I} (M' J_{O \rightarrow I})^{-1} S^{-1}. \end{aligned} \quad (1)$$

What we expect is that the maximum value of each column vector of  $J_{W \rightarrow I}$  equals one. This constraint can solve the values of  $\{s_1, s_2, s_3\}$  for the example point  $\mathbf{x}$ . For all the sampled points  $\{\mathbf{x}_i\}$ , we take the average values of  $\{s_1, s_2, s_3\}$  for our final scale parameters.

### 1.3. Perspective sampling

As stated in the main paper (Sec 3.5), when sampling points in ray marching, we perform uniform sampling on the warp space, and we get a non-uniform sampling in the original space. To be specific, for the current sample point  $\mathbf{x}_i = \mathbf{o} + t_i \mathbf{d}$ , we expect to find the next sample point  $\mathbf{x}_{i+1} = \mathbf{x}_i + \delta_i \mathbf{d}$ , such that  $\|F(\mathbf{x}_{i+1}) - F(\mathbf{x}_i)\|_2 = l$ . Here  $l$  is the parameter controlling sample density and we empirically set  $l = \sqrt{3}$ , i.e., the diagonal length of the unit cube in the warp space [6]. To compute the marching step  $\delta_i$  in the original space efficiently, we perform a linear approximation that

$$F(\mathbf{x}_{i+1}) \approx F(\mathbf{x}_i) + \delta_i \cdot J_i \mathbf{d}, \quad (2)$$

Where  $J_i$  is the Jacobian matrix at  $\mathbf{x}_i$  from the original space to the warp space. Hence, the distance between  $F(\mathbf{x}_{i+1})$  and  $F(\mathbf{x}_i)$  is approximated by  $\delta_i \|J_i \mathbf{d}\|_2$ . We let it equals  $l$  and get  $\delta_i = \frac{l}{\|J_i \mathbf{d}\|_2}$ .

### 1.4. Loss functions

As described in the main paper, the loss of training is defined as

$$\mathcal{L} = \mathcal{L}_{recon}(c(r), c_{gt}) + \lambda_{Disp} \mathcal{L}_{Disp} + \lambda_{TV} \mathcal{L}_{TV}, \quad (3)$$

where the first term  $\mathcal{L}_{recon}(c(r), c_{gt}) = \sqrt{(c(r) - c_{gt})^2 + \epsilon}$  is a color reconstruction loss [2] with  $\epsilon = 10^{-4}$ , and the last two terms are the regularization losses.

The disparity loss  $\mathcal{L}_{Disp}$  of the sampled rays is defined by

$$\mathcal{L}_{Disp} = \frac{1}{n_r} \sum_k \text{disp}_k^2, \quad (4)$$

where the disparity of a ray is computed by the weighted sum of the sampled inverse distance that  $\text{disp} = \sum_i w_i \frac{1}{t_i}$ , and  $\{w_i\}$  are the weights computed by volume rendering.

The aim of total variation loss  $\mathcal{L}_{TV}$  is to encourage the border points of two neighboring octree nodes to have similar densities and colors. To achieve this goal, in each training iteration, we randomly sample  $n_b = 8192$  points on the borders of the octree nodes, then the loss is defined by

$$\mathcal{L}_{TV} = \frac{1}{n_b} \sum_k \|\text{feat}_0^k - \text{feat}_1^k\|_2^2. \quad (5)$$

Here, for each sample point  $k$ ,  $\text{feat}_0^k$  and  $\text{feat}_1^k$  are the feature vectors fetched from the hash table using two different functions conditioned on its two neighboring octree nodes.

In LLFF dataset we set  $\lambda_{Disp} = 2.5 \times 10^{-4}$ ,  $\lambda_{TV} = 10^{-1}$ , and in Free dataset and NeRF-360-V2 dataset we set  $\lambda_{Disp} = 10^{-3}$ ,  $\lambda_{TV} = 10^{-1}$ .

### 1.5. More implementation details

**Architecture details.** We follow a similar setting to Instant-NGP [6] and use the hash table with 16 levels, and each level contain  $2^{19}$  feature vectors with dimension of 2. The fetched hash feature vectors of size 32 are fed to a tiny MLP with one hidden layer of width 64, to get the scene features and the volume densities, then, the scene features are concatenated with the spherical harmonics encoding of view directions and are fed to another rendering MLP with two hidden layers of width 64 to get the RGB colors.

**Training details.** We follow Instant-NGP [6] and set the fixed batch size of point samples as 256k while the batch size of rays is dynamic, depending on the average sampled points on rays. We train the parameters with Adam optimizer [4], whose learning rate linearly grows from zero to  $1 \times 10^{-1}$  in the first 1k steps and the decay to  $10^{-2}$  at the end of training with cosine scheduling. For all the scenes in the experiments, we train F<sup>2</sup>-NeRF for 20k steps. We implement F<sup>2</sup>-NeRF using LibTorch [7]. The training time depends on the scene's complexity, and for most cases, it is between 10 minutes and 15 minutes on a single Nvidia 2080Ti GPU.

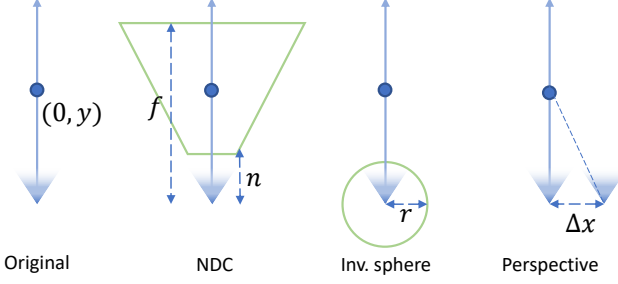


Figure 4. Different warp coordinates.

## 2. Connection to NDC warping and Inv. sphere warping

In the main paper, we intuitively show the connection of our proposed perspective warping to NDC warping and inverse sphere warping. Here we mathematically analyze the connections using two forward-facing 1D cameras that project 2D points onto their 1D camera plane as shown in Fig. 4. The proper perspective warping utilizes the image coordinates of two cameras as the warping coordinates. Thus, the point with coordinate  $(0, y)$  in the original Euclidean space will be mapped to  $(0, -\frac{\Delta x}{y})$  in the warping space. Meanwhile, the coordinates of this point in the NDC space and inverse sphere space are  $(0, \frac{f+n}{f-n} - \frac{2fn}{f-n} \cdot \frac{1}{y})$  and  $(0, 2 - \frac{r}{y})$  respectively, where  $n, f$  are preset near-far depth and  $r$  is preset sphere radius. When  $\Delta x = \frac{2fn}{f-n}$  or  $\Delta x = r$ , the perspective warping is equivalent to NDC warping or inverse sphere warping with a constant offset. However, theoretically proving such connections in general cases of the 3D space is very complex since it involves sampling points and PCA analysis on sample points.

## 3. Additional Experimental Results

### 3.1. Training for longer steps

We provide quantitative results on training F<sup>2</sup>-NeRF and instant-NGP for a longer time on the Free dataset (Table 1) and NeRF-360-V2 dataset (Table 2). When trained for a longer time, F<sup>2</sup>-NeRF and Instant-NGP can obtain better rendering quality. As shown in Table 1, on the Free dataset, training Instant-NGP for a longer time (15m, 50k steps) does not achieve better rendering quality than F<sup>2</sup>-NeRF (12m, 20k steps). Moreover, increasing the hash table size from  $2^{19}$  to  $2^{20}$  helps improve the performance of F<sup>2</sup>-NeRF on the Free dataset (F<sup>2</sup>-NeRF<sub>50k-large</sub>).

### 3.2. Compatibility with MLP-based NeRF

In this section, we provide results of applying perspective warping on MLP-based NeRF. In this experiment, for each setting, we train a neural radiance field represented by an 8-layer fully-connected MLP for 250K steps, and use different

| Method                                    | Tr. time | PSNR $\uparrow$ | SSIM $\uparrow$ | LPIPS(VGG) $\downarrow$ |
|-------------------------------------------|----------|-----------------|-----------------|-------------------------|
| Instant-NGP <sub>20k</sub>                | 6m       | 24.43           | 0.677           | 0.413                   |
| Instant-NGP <sub>50k</sub>                | 15m      | 25.07           | 0.703           | 0.376                   |
| F <sup>2</sup> -NeRF <sub>20k</sub>       | 12m      | 26.32           | 0.779           | 0.276                   |
| F <sup>2</sup> -NeRF <sub>50k</sub>       | 30m      | 26.85           | 0.811           | 0.235                   |
| F <sup>2</sup> -NeRF <sub>50k-large</sub> | 36m      | 27.19           | 0.833           | 0.204                   |

Table 1. Training Instant-NGP and F<sup>2</sup>-NeRF for longer time on the Free dataset.

| Method                              | Tr. time | PSNR $\uparrow$ | SSIM $\uparrow$ | LPIPS(VGG) $\downarrow$ |
|-------------------------------------|----------|-----------------|-----------------|-------------------------|
| Instant-NGP <sub>20k</sub>          | 6m       | 26.24           | 0.716           | 0.404                   |
| Instant-NGP <sub>50k</sub>          | 17m      | 26.55           | 0.733           | 0.382                   |
| F <sup>2</sup> -NeRF <sub>20k</sub> | 14m      | 26.39           | 0.746           | 0.361                   |
| F <sup>2</sup> -NeRF <sub>50k</sub> | 33m      | 26.92           | 0.771           | 0.333                   |

Table 2. Training Instant-NGP and F<sup>2</sup>-NeRF for longer time on the NeRF-360-V2 dataset.

warping functions before feeding the positional encoding to the MLP. For the perspective warping, we do not subdivide the spaces and also only use one single MLP. We also provide results of other warping functions using the same MLP-based NeRF, as shown in Table 3. In the forward-facing setting, our perspective warping (mean PSNR: 26.29) and NDC warping (26.31) perform better than the inverse sphere warping (26.02). The PSNR of NDC warping is slightly worse than the reported PSNR by the original paper [5] due to fewer training steps (ours: 250K steps, official: 1M steps). Fig. 5 provides qualitative results on the “Room” case of LLFF dataset, and our perspective warping method presents more visual details in the synthesized image, which demonstrates that the proposed perspective warping is compatible with MLP-based NeRF.

### 3.3. View extrapolation

Here we additionally conduct an experiment for view extrapolation on the “Lego” case from the NeRF synthetic dataset. In this experiment, we choose images with elevation angles less than  $30^\circ$  for training and the others for testing. As shown in Table 4 and Fig. 6, the result of our perspective warping method is similar to that using original Euclidean space.

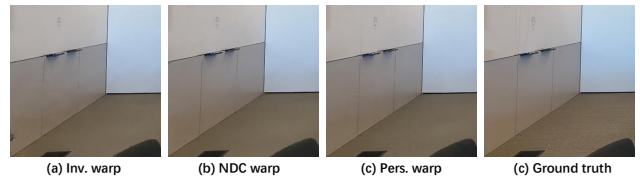


Figure 5. Visual comparisons among different warping methods on the “Room” case of LLFF dataset.

| Warping method | Fern         | Flower       | Fortress     | Horns        | Leaves       | Orchids      | Room         | Trex         | Mean         |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| NDC Warp       | <b>24.82</b> | 27.87        | <b>31.22</b> | <b>27.37</b> | 20.74        | <b>19.91</b> | <b>31.75</b> | 26.77        | <b>26.31</b> |
| Inv. Warp      | 24.40        | 27.40        | 30.96        | 27.19        | 20.59        | 19.72        | 31.23        | 26.66        | 26.02        |
| Pers. Warp     | 24.71        | <b>27.88</b> | <b>31.22</b> | 27.22        | <b>20.84</b> | 19.82        | 31.64        | <b>27.03</b> | 26.29        |

Table 3. Different warping functions on MLP-based NeRFs on the LLFF dataset.

| Elevation  | [0°, 30°) | [30°, 60°) | [60°, 90°) |
|------------|-----------|------------|------------|
| Pers. warp | 37.63     | 30.19      | 27.35      |
| w/o warp   | 37.15     | 30.35      | 27.03      |

Table 4. Results on view extrapolation in the metric of PSNR.

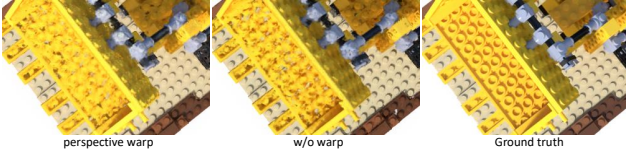


Figure 6. The NVS result of an extrapolated view.

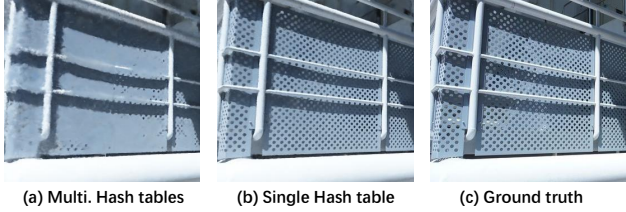


Figure 7. Visual comparison between using multiple hash tables (a) and single hash table (b) on the “Sky” case of Free dataset.

### 3.4. Additional ablations

**Single v.s. multiple hash tables.** We test F<sup>2</sup>-NeRF on the setting with multiple hash tables, i.e., one hash table for each octree node, with the same budget of parameters as the setting of a single hash table used in the paper. In this setting, the size of each hash table is  $L/n_l$ , where  $L = 2^{19}$  is the overall table size and  $n_l$  is the number of leaf octree nodes. Fig. 7 shows that when using multiple hash tables, the quality degrades clearly compared to the setting of using a single hash table with multiple hash functions. The reason is that using a global hash table has more flexibility in allocating the representation capacity to different regions.

**Effect of regularization losses.** As shown in Fig. 8, when regularization losses are not used, the foggy artifacts appear and the rendered result is not clear, especially on the regions with pure colors.

### 3.5. Per-scene results

We provide the per-scene results on the Free dataset, NeRF-360-V2 dataset, and LLFF dataset in Table 6, Table 8, and Table 7 respectively. The results are reported in the metric of PSNR. Table 5 provides per-scene results on

different warping and sampling methods on the Free dataset. We note that the longer the trajectory is (e.g., the “stair” and “grass”), the relatively better performance our perspective warping method with the perspective sampling achieves than the inverse sphere warping method.

### 3.6. More visual results

We provide more visual comparisons on the Free dataset in Fig. 9.

## 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. 5
- [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. 2, 5
- [3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. 5
- [4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2
- [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. 3, 5
- [6] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 2022. 2, 5
- [7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An



Figure 8. Visual comparison between w/o regularization losses (a) and w/ regularization losses (b) on the “Bonsai” case in NeRF-360-V2 dataset.



| Setting                   | Hydrant      | Lab          | Pillar       | Road         | Sky          | Stair        | Grass        |
|---------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| w/o warp + Disp. sample   | 23.23        | 25.06        | 26.81        | 25.46        | 25.44        | 27.64        | 21.33        |
| Inv. warp + Disp. sample  | 24.05        | 25.43        | 27.09        | 26.24        | 26.27        | 27.66        | 22.34        |
| Inv. warp + Exp. sample   | 24.15        | 25.58        | 27.86        | 26.21        | 26.27        | 28.41        | 21.94        |
| Pers. warp + Exp. sample  | 24.31        | 25.79        | 28.65        | 26.60        | 26.15        | 29.08        | <b>22.89</b> |
| Pers. warp + Pers. sample | <b>24.34</b> | <b>25.92</b> | <b>28.76</b> | <b>26.76</b> | <b>26.41</b> | <b>29.19</b> | 22.87        |

Table 5. Scene breakdown of our ablation studies on the warping and sampling methods.

| Method                   | Hydrant      | Lab          | Pillar       | Road         | Sky          | Stair        | Grass        |
|--------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| NeRF++ [10]              | 22.21        | 21.82        | 25.73        | 23.29        | 23.91        | 26.08        | 21.26        |
| mip-NeRF-360             | <b>25.03</b> | <b>26.57</b> | <b>29.22</b> | <b>27.07</b> | <b>26.99</b> | <b>29.79</b> | <b>24.39</b> |
| mip-NeRF-360 (short) [2] | 21.01        | 21.17        | 24.12        | 21.49        | 22.29        | 24.27        | 19.87        |
| Plenoxels [9]            | 19.82        | 18.12        | 18.74        | 21.31        | 18.22        | 21.41        | 16.28        |
| DVGO [8]                 | 22.10        | 23.78        | 26.22        | 23.53        | 24.26        | 26.65        | 20.75        |
| Instant-NGP [6]          | 22.30        | 23.21        | 25.88        | 24.24        | 25.80        | 27.79        | 21.82        |
| F <sup>2</sup> -NeRF     | <b>24.34</b> | <b>25.92</b> | <b>28.76</b> | <b>26.76</b> | <b>26.41</b> | <b>29.19</b> | <b>22.87</b> |

Table 6. Scene breakdown on the Free dataset.

imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 2

- [8] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 5
- [9] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 5
- [10] Kai Zhang, Gernot Riegler, Noah Snaveley, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arxiv CS.CV 2010.07492*, 2020. 5

| Method               | Fern         | Flower       | Fortress     | Horns        | Leaves       | Orchids      | Room         | Trex         |
|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| NeRF [5]             | <b>25.17</b> | 27.40        | 31.16        | 27.45        | 20.92        | <b>20.36</b> | <b>32.70</b> | 26.80        |
| mip-NeRF [1]         | 25.12        | <b>27.79</b> | <b>31.42</b> | <b>27.55</b> | <b>20.97</b> | 20.28        | 32.52        | <b>27.16</b> |
| Plenoxels [9]        | <b>25.46</b> | 27.83        | 31.09        | 27.58        | <b>21.41</b> | 20.24        | 30.22        | 26.48        |
| TensoRF [3]          | 25.27        | <b>28.60</b> | 31.36        | <b>28.14</b> | 21.30        | 19.87        | <b>32.35</b> | 26.97        |
| DVGO [8]             | 25.08        | 27.62        | 30.44        | 27.59        | 21.00        | <b>20.33</b> | 31.53        | 27.17        |
| Instant-NGP [6]      | 25.13        | 27.07        | 30.96        | 27.32        | 12.08        | 19.80        | 31.56        | 26.82        |
| F <sup>2</sup> -NeRF | 25.26        | 27.48        | <b>31.49</b> | 27.84        | 20.68        | 20.10        | 32.23        | <b>27.26</b> |

Table 7. Scene breakdown on the LLFF dataset.

| Method               | Bicycle      | Bonsai       | Counter      | Garden       | Kitchen      | Room         | Stump        |
|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| NeRF++ [10]          | 22.64        | 29.15        | 26.38        | 24.32        | 27.80        | 28.87        | 24.34        |
| mip-NeRF-360 [2]     | <b>23.99</b> | <b>33.06</b> | <b>29.51</b> | <b>26.10</b> | <b>32.13</b> | <b>31.53</b> | <b>26.27</b> |
| Plenoxels [9]        | 21.39        | 23.65        | 25.23        | 22.71        | 24.00        | 26.38        | 20.08        |
| DVGO [8]             | <b>22.12</b> | 27.80        | 25.76        | 24.34        | 26.00        | 28.33        | 23.59        |
| Instant-NGP [6]      | 22.08        | <b>29.86</b> | <b>26.37</b> | 24.26        | 28.27        | 28.90        | 23.93        |
| F <sup>2</sup> -NeRF | 22.11        | 29.65        | 25.36        | <b>24.76</b> | <b>28.97</b> | <b>29.30</b> | <b>24.60</b> |

Table 8. Scene breakdown on the NeRF-360-V2 dataset.

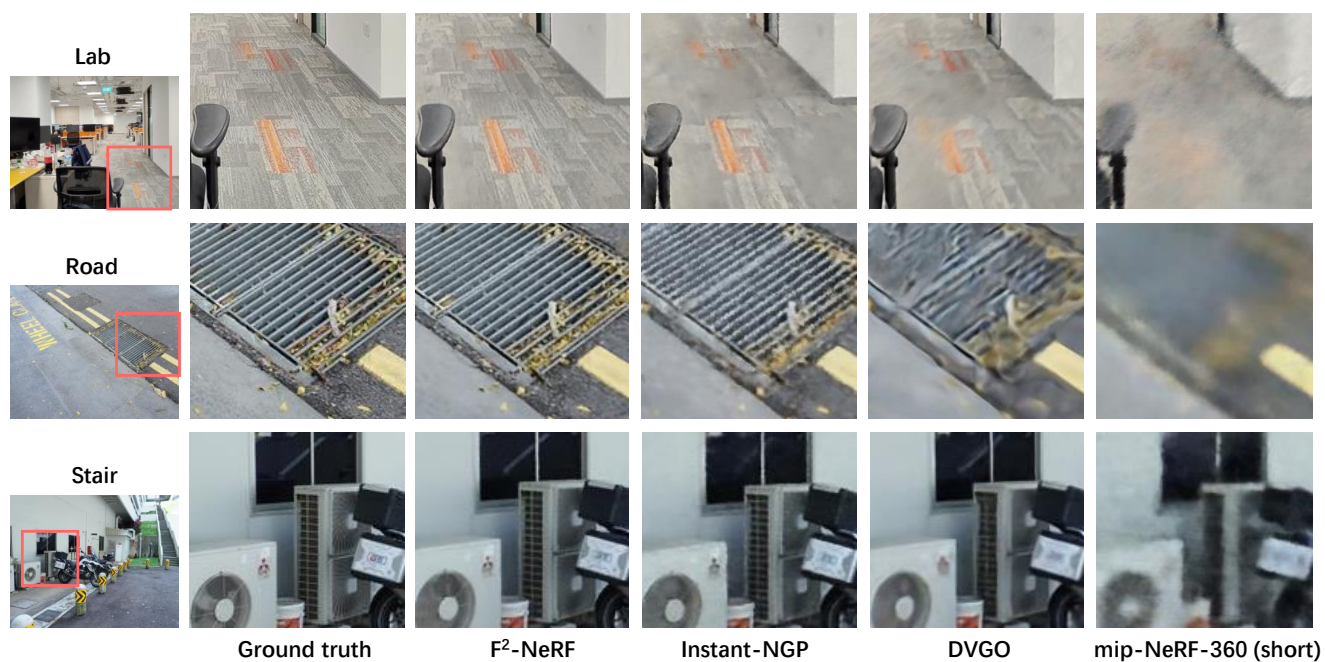


Figure 9. Additional visual comparisons on the Free dataset.