Supplementary Materials for GeoNeRF: Generalizing NeRF with Geometry Priors

Mohammad Mahdi Johari
Idiap Research Institute, EPFL
mohammad.johari@idiap.ch

Yann Lepoittevin
ams OSRAM
yann.lepoittevin@ams.com

François Fleuret
University of Geneva, EPFL
francois.fleuret@unige.ch

1. Additional Technical Details

As stated in the main article, we borrow the architecture of our geometry reasoner from CasMVSNet [2]. We construct $D^{(2)} = 48$ depth planes for the coarsest cost volume, $D^{(1)} = 32$ for the intermediate one, and $D^{(0)} = 8$ for the finest full-resolution cost volume. We use channel size $C = 8$ in group-wise correlation similarity calculations. When training the generalizable model, we create a set of 3–5 nearby source views for constructing each cost volume, whereas for fine-tuning and evaluating, we always use a set of 5 nearby views. Also, we scale input images with a factor uniformly sampled from $\{1.0, 0.75, 0.5\}$ when we train our generalizable model.

The network architectures of Feature Pyramid Network (FPN), 3D regularizer ($R_{3D}^{(l)}$), and the auto-encoder (AE) are provided in Tables 1, 2, and 3 respectively.

2. Additional Qualitative Analysis

Full-size examples of rendered images for novel views by our GeoNeRF model are presented in Figures 1 and 2. Figure 1 includes samples from the real forward-facing LLFF dataset [3], and Figure 2 contains samples from the NeRF realistic synthetic dataset [4]. In addition to the rendered images, we also show the rendered depth maps for each novel view. Images

Input	Layer	Output
Input	ConvBnReLU(3, 8, 3, 1)	conv0_0
conv0_0	ConvBnReLU(8, 8, 3, 1)	conv0
conv0	ConvBnReLU(8, 16, 5, 2)	conv1_0
conv1_0	ConvBnReLU(16, 16, 3, 1)	conv1_1
conv1_1	ConvBnReLU(16, 16, 3, 1)	conv1
conv1	ConvBnReLU(16, 32, 5, 2)	conv2_0
conv2_0	ConvBnReLU(32, 32, 3, 1)	conv2_1
conv2_1	ConvBnReLU(32, 32, 3, 1)	conv2
conv2	Conv(32, 32, 1, 1)	feat2
conv1	Conv(16, 32, 1, 1)	f1_0
conv0	Conv(8, 32, 1, 1)	f0_0
(feat2, f1_0)	Upsample_and_Add(x, y)	f1_1
(f1_1, f0_0)	Upsample_and_Add(x, y)	f0_1
f1_1	Conv(32, 16, 3, 1)	feat1
f0_1	Conv(32, 8, 3, 1)	feat0

Table 1. Network architecture of Feature Pyramid Network (FPN), where *feat2*, *feat1*, and *feat0* are output feature pyramids. Conv(c_{in}, c_{out}, k, s) stands for a 2D convolution with input channels c_{in}, output channels c_{out}, kernel size of k, and stride of s. ConvBnReLU represents a Conv layer followed by Batch Normalization and ReLU nonlinearity. Upsample_and_Add(x, y) adds y to the bilinearly upsampled of x.

Input	Layer	Output
Input	ConvBnReLU(8, 8, 3, 1)	conv0
conv0	ConvBnReLU(8, 16, 3, 2)	conv1
conv1	ConvBnReLU(16, 16, 3, 1)	conv2
conv2	ConvBnReLU(16, 32, 3, 2)	conv3
conv3	ConvBnReLU(32, 32, 3, 1)	conv4
conv4	ConvBnReLU(32, 64, 3, 2)	conv5
conv5	ConvBnReLU(64, 64, 3, 1)	conv6
conv6	TrpsConvBnReLU(64, 32, 3, 2)	x_0
(conv4, x_0)	Add(x, y)	x_1
x_1	TrpsConvBnReLU(32, 16, 3, 2)	x_2
(conv2, x_2)	Add(x, y)	x_3
x_3	TrpsConvBnReLU(16, 8, 3, 2)	x_4
(conv0, x_4)	Add(x, y)	x_5
x_5	ConvBnReLU(8, 8, 3, 1)	prob_0
prob_0	Conv(8, 1, 3, 1)	prob
x_5	ConvBnReLU(8, 8, 3, 1)	feat

Table 2. Network architecture of the 3D regularizer ($R_{3D}^{(l)}$), where *feat* is the output 3D feature map $\mathbf{\Phi}^{(l)}$ and *prob* is the output probability which is used to regress the depth map $\hat{\boldsymbol{D}}^{(l)}$. Conv($c_{in}$, c_{out}, k, s) stands for a 3D convolution with input channels c_{in}, output channels c_{out}, kernel size of k, and stride of s. ConvBnReLU represents a Conv layer followed by Batch Normalization and ReLU nonlinearity, and TrpsConv stands for transposed 3D convolution. Add(x, y) simply adds y to x.

Input	Layer	Output
Input	ConvLnELU(32, 64, 3, 1)	conv1_0
conv1_0	MaxPool	conv1
conv1	ConvLnELU(64, 128, 3, 1)	conv2_0
conv2_0	MaxPool	conv2
conv2	ConvLnELU(128, 128, 3, 1)	conv3_0
conv3_0	MaxPool	conv3
conv3	TrpsConvLnELU(128, 128, 4, 2)	x_0
[conv2 ; x_0]	TrpsConvLnELU(256, 64, 4, 2)	x_1
[conv1 ; x_1]	TrpsConvLnELU(128, 32, 4, 2)	x_2
[Input ; x_2]	ConvLnELU(64, 32, 3, 1)	Output

Table 3. Network architecture of the auto-encoder network (AE). Conv(c_{in}, c_{out}, k, s) stands for a 1D convolution with input channels c_{in}, output channels c_{out}, kernel size of k, and stride of s. ConvLnELU represents a Conv layer followed by Layer Normalization and ELU nonlinearity, and TrpsConv stands for transposed 1D convolution. MaxPool is a 1D max pooling layer with a stride of 2, and $[\cdot\,;\,\cdot]$ denotes concatenation.

indicated by GeoNeRF are rendered by our generalizable model, while images indicated by GeoNeRF$_{10k}$ are rendered after fine-tuning our model on each scene for 10k iterations.

3. Per-Scene Breakdown

Tables 4, 5, 6, and 7 break down the quantitative results presented in the main paper into per-scene metrics. The results are consistent with the aggregate results in the main paper. Tables 4 and 5 include the scenes from the real forward-facing LLFF dataset [3], and Tables 6 and 7 contain the scenes from NeRF realistic synthetic dataset [4]. As it was already shown in the main paper, our generalizable GeoNeRF model outperforms all existing generalizable methods on average, and after fine-tuning, it is on par with per-scene optimized vanilla NeRF [4].

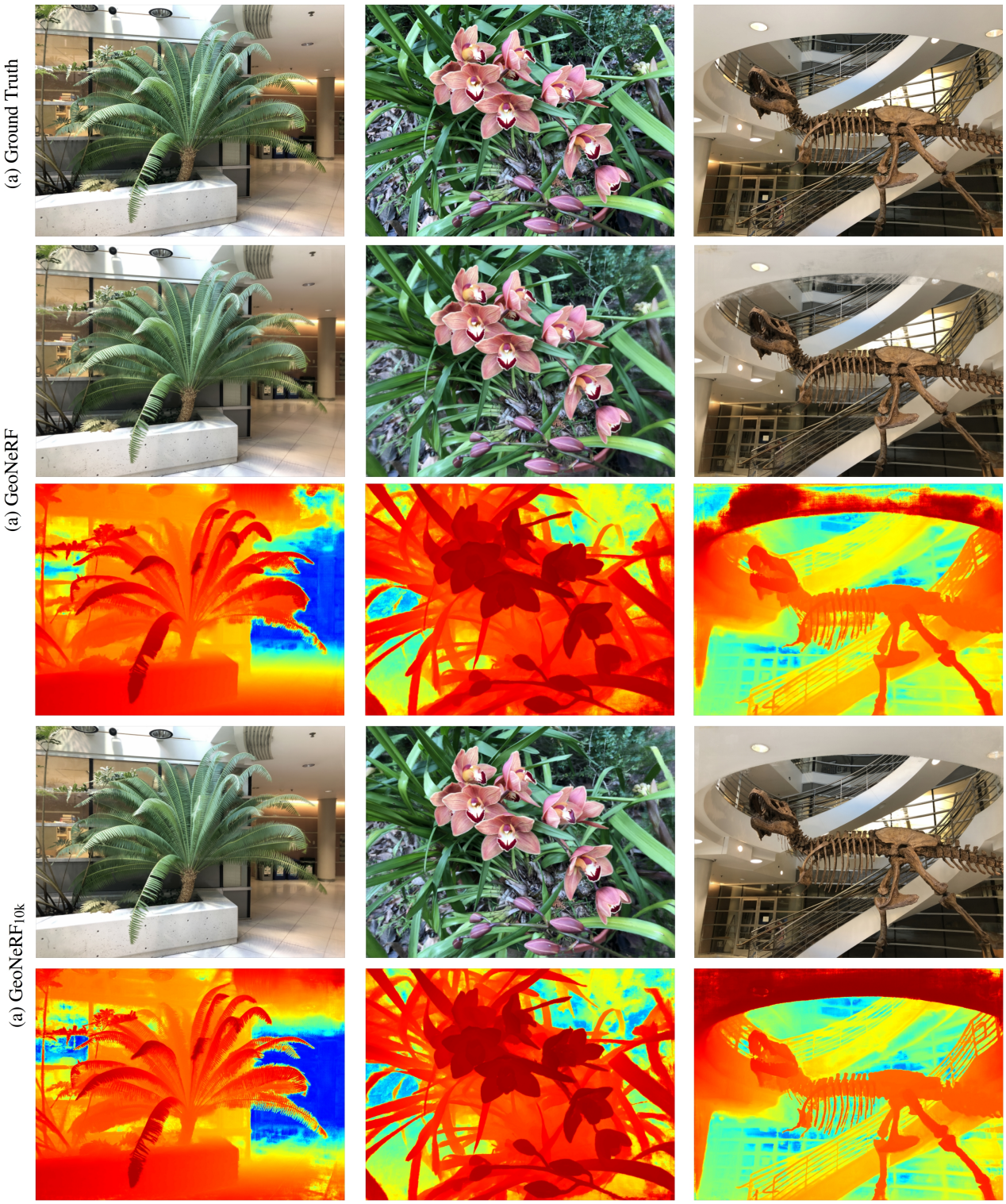


Figure 1. Full-size examples of novel images and their depth map rendered by our generalizable (GeoNeRF) and fine-tuned (GeoNeRF$_{10k}$) models. The images are from test scenes of the real forward-facing LLFF dataset [3].

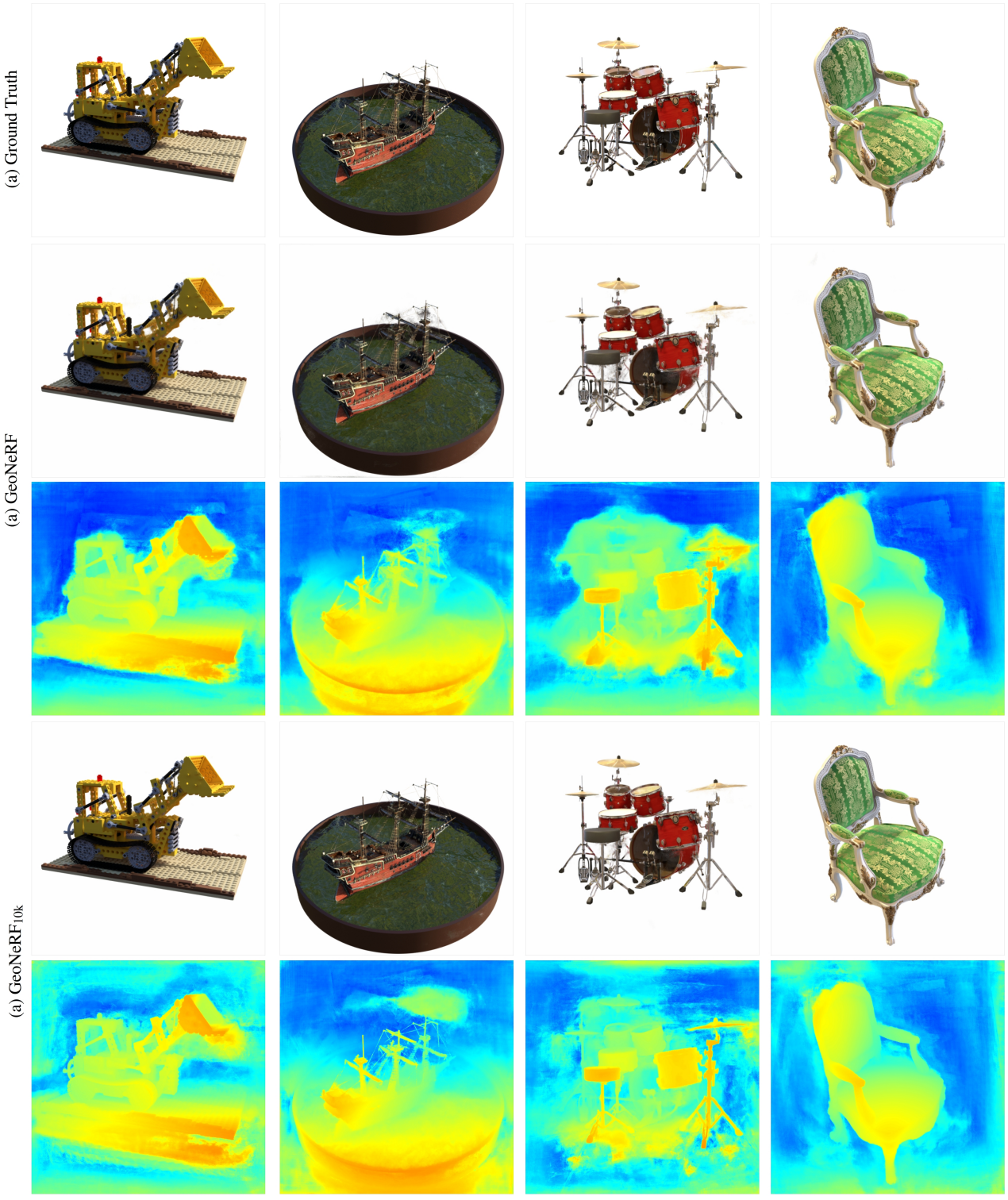


Figure 2. Full-size examples of novel images and their depth map rendered by our generalizable (GeoNeRF) and fine-tuned (GeoNeRF$_{10k}$) models. The images are from test scenes of the NeRF realistic synthetic dataset [4].

	PSNR↑							
	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
pixelNeRF [7]	12.40	10.00	14.07	11.07	9.85	9.62	11.75	10.55
IBRNet [5]	23.84	26.67	30.00	26.48	20.19	19.34	**29.94**	**24.57**
MVSNeRF [1]	21.15	24.74	26.03	23.57	17.51	17.85	26.95	23.20
GeoNeRF	**24.61**	**28.12**	**30.49**	**26.96**	**20.58**	**20.24**	28.74	23.75

	SSIM↑							
	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
pixelNeRF [7]	0.531	0.433	0.674	0.516	0.268	0.317	0.691	0.458
IBRNet [5]	0.772	0.856	0.883	0.869	0.719	0.633	0.946	0.861
MVSNeRF [1]	0.638	**0.888**	0.872	0.868	0.667	0.657	**0.951**	0.868
GeoNeRF	**0.811**	0.885	**0.898**	**0.901**	**0.741**	**0.666**	0.935	**0.877**

	LPIPS↓							
	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
pixelNeRF [7]	0.650	0.708	0.608	0.705	0.695	0.721	0.611	0.667
IBRNet [5]	0.246	0.164	0.153	0.177	0.230	0.287	0.153	0.230
MVSNeRF [1]	0.238	0.196	0.208	0.237	0.313	0.274	0.172	**0.184**
GeoNeRF	**0.202**	**0.133**	**0.123**	**0.140**	**0.222**	**0.256**	**0.150**	0.212

Table 4. Per-scene Quantitative comparison of our proposed GeoNeRF with existing generalizable NeRF models on real forward-facing LLFF dataset [3] in terms of PSNR (higher is better), SSIM [6] (higher is better), and LPIPS [8] (lower is better) metrics.

	PSNR↑							
	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
NeRF [4]	25.17	27.40	**31.16**	27.45	20.92	20.36	**32.70**	**26.80**
GeoNeRF$_{10k}$	**25.24**	28.57	30.75	**28.12**	**21.40**	20.39	31.51	26.63
GeoNeRF$_{1k}$	25.08	**28.74**	30.83	27.66	21.16	**20.41**	30.52	26.07

	SSIM↑							
	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
NeRF [4]	0.792	0.827	0.881	0.828	0.690	0.641	0.948	0.880
GeoNeRF$_{10k}$	**0.829**	0.890	0.900	**0.912**	**0.781**	**0.674**	**0.956**	**0.910**
GeoNeRF$_{1k}$	0.824	**0.892**	**0.905**	0.908	0.769	0.673	0.946	0.901

	LPIPS↓							
	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex
NeRF [4]	0.280	0.219	0.171	0.268	0.316	0.321	0.178	0.249
GeoNeRF$_{10k}$	**0.185**	0.120	0.125	**0.126**	**0.183**	**0.247**	**0.126**	**0.181**
GeoNeRF$_{1k}$	0.189	**0.114**	**0.117**	0.130	0.198	0.248	0.135	0.188

Table 5. Per-scene Quantitative comparison of our fine-tuned GeoNeRF with per-scene optimized vanilla NeRF [4] on real forward-facing LLFF dataset [3] in terms of PSNR (higher is better), SSIM [6] (higher is better), and LPIPS [8] (lower is better) metrics. Our model is fine-tuned on each scene for 10k iterations (GeoNeRF$_{10k}$) and 1k iterations (GeoNeRF$_{1k}$), and NeRF [4] is optimized for 200k iterations.

	PSNR↑							
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
pixelNeRF [7]	7.18	8.15	6.61	6.80	7.74	7.61	7.71	7.30
IBRNet [5]	28.54	21.22	24.23	31.72	24.59	22.20	27.97	23.64
MVSNeRF [1]	23.35	20.71	21.98	28.44	23.18	20.05	22.62	23.35
GeoNeRF	**31.84**	**24.00**	**25.28**	**34.33**	**28.80**	**26.16**	**31.15**	**25.08**

	SSIM↑							
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
pixelNeRF [7]	0.624	0.670	0.669	0.669	0.671	0.644	0.729	0.584
IBRNet [5]	0.948	0.896	0.915	0.952	0.918	0.905	0.962	0.834
MVSNeRF [1]	0.876	0.886	0.898	0.962	0.902	0.893	0.923	**0.886**
GeoNeRF	**0.973**	**0.921**	**0.931**	**0.975**	**0.956**	**0.926**	**0.978**	0.844

	LPIPS↓							
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
pixelNeRF [7]	0.386	0.421	0.335	0.433	0.427	0.432	0.329	0.526
IBRNet [5]	0.066	**0.091**	0.097	0.067	0.095	**0.115**	0.051	0.219
MVSNeRF [1]	0.282	0.187	0.211	0.173	0.204	0.216	0.177	0.244
GeoNeRF	**0.040**	0.098	**0.092**	**0.056**	**0.059**	0.116	**0.037**	**0.200**

Table 6. Per-scene Quantitative comparison of our proposed GeoNeRF with existing generalizable NeRF models on NeRF realistic synthetic dataset [4] in terms of PSNR (higher is better), SSIM [6] (higher is better), and LPIPS [8] (lower is better) metrics.

	PSNR↑							
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
NeRF [4]	33.00	25.01	**30.13**	36.18	**32.54**	**29.62**	32.91	28.65
GeoNeRF$_{10k}$	**33.54**	**25.13**	27.79	**36.26**	30.32	28.19	**33.41**	**28.76**
GeoNeRF$_{1k}$	32.76	24.74	27.06	35.71	29.79	27.69	32.83	28.11

	SSIM↑							
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
NeRF [4]	0.967	0.925	**0.964**	0.974	0.961	0.949	0.980	0.856
GeoNeRF$_{10k}$	**0.980**	**0.935**	0.955	**0.983**	**0.965**	**0.953**	**0.987**	**0.890**
GeoNeRF$_{1k}$	0.977	0.930	0.948	0.982	0.961	0.948	0.985	0.883

	LPIPS↓							
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
NeRF [4]	0.046	0.091	**0.044**	0.121	0.050	0.063	0.028	0.206
GeoNeRF$_{10k}$	**0.024**	**0.073**	0.061	**0.032**	**0.041**	**0.058**	**0.016**	**0.137**
GeoNeRF$_{1k}$	0.030	0.081	0.069	0.034	0.046	0.069	0.020	0.145

Table 7. Per-scene Quantitative comparison of our fine-tuned GeoNeRF with per-scene optimized vanilla NeRF [4] on NeRF realistic synthetic dataset [4] in terms of PSNR (higher is better), SSIM [6] (higher is better), and LPIPS [8] (lower is better) metrics. Our model is fine-tuned on each scene for 10k iterations (GeoNeRF$_{10k}$) and 1k iterations (GeoNeRF$_{1k}$), and NeRF [4] is optimized for 500k iterations.

4. Ablation Study

An ablation study of our generalizable model on the NeRF synthetic dataset [4] and the real forward-facing dataset [3] is presented in Table 8, contrasting the effectiveness of individual components of our proposed model. We evaluated GeoNeRF in the cases where (a) no self-supervision loss is used, (b) no positional encoding is employed, (c) points on a ray are merely sampled uniformly, (d) occluded views are not excluded, (e) attention mechanism is removed from the renderer, (f) view-independent tokens are not regularized with the AE network before predicting volume densities, and (g) only a single cost

Experiment	Realistic Synthetic NeRF [4]			Real Forward Facing LLFF [3]			Examples
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
a. Without self-supervision	28.10	0.935	0.098	25.37	0.836	0.184	Figure 3.a
b. Without positional encoding	27.19	0.927	0.116	25.02	0.836	0.189	Figure 3.b
c. Uniform sampling along a ray	28.04	0.934	0.089	25.31	0.835	0.184	Figure 3.c
d. Without occlusion masks	27.92	0.932	0.097	25.22	0.834	0.185	Figure 3.d
e. Without attention mechanism	27.69	0.929	0.135	24.95	0.828	0.194	Figure 3.e
f. Without the AE network	23.53	0.884	0.182	24.92	0.821	0.199	Figure 3.f
g. Single cost volume	26.60	0.915	0.132	24.60	0.814	0.211	Figure 3.g
h. Full GeoNeRF	**28.33**	**0.938**	**0.087**	**25.44**	**0.839**	**0.180**	Figure 3.h

Table 8. Ablation study of the key components of GeoNeRF. The evaluation is performed on the NeRF synthetic [4] and the real forward-facing LLFF [3] test scenes. See Section 4 for the details of these experiments, and see Figure 3 for qualitative analysis.

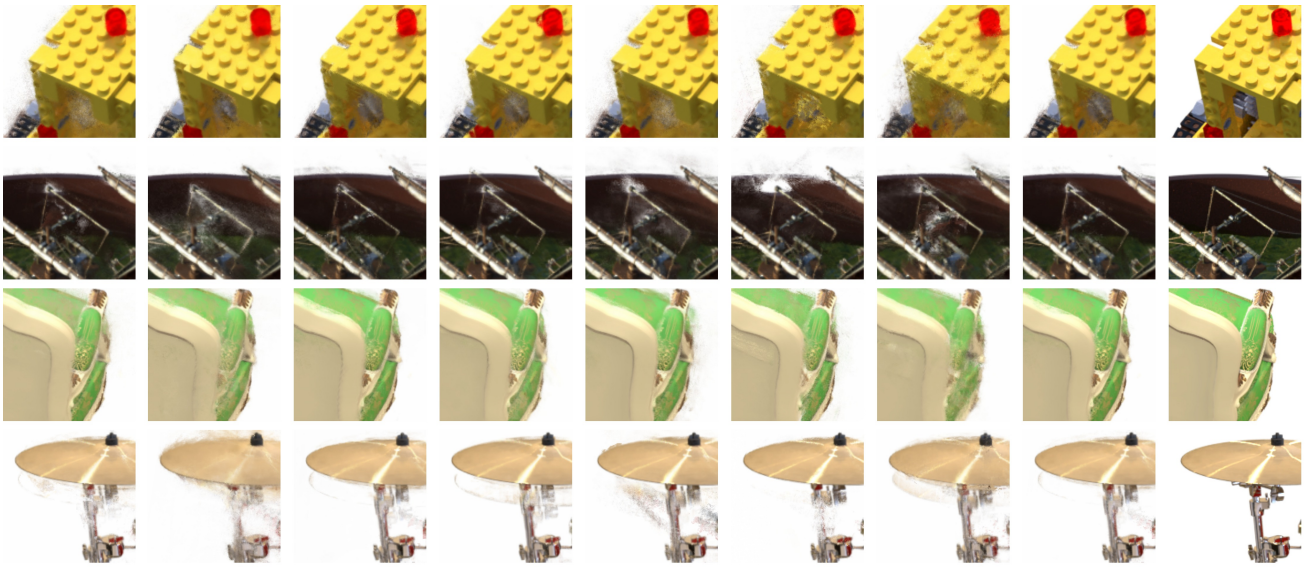


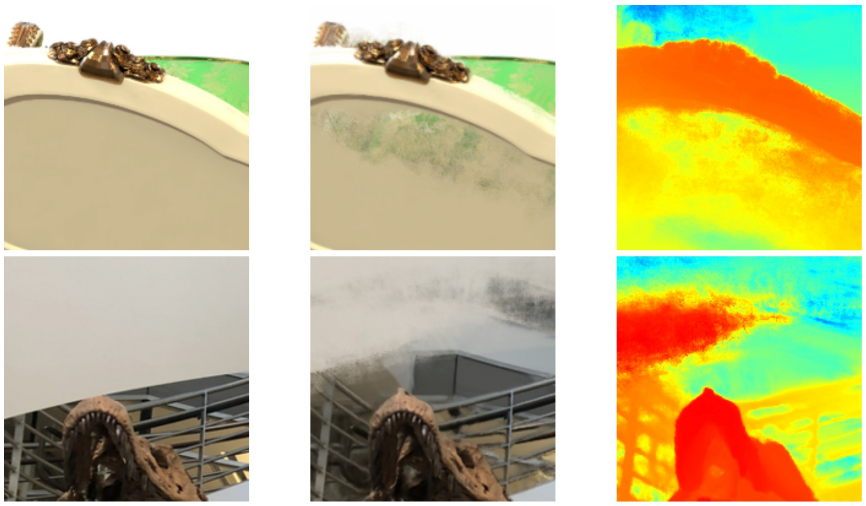
Figure 3. Qualitative ablation study of the key components of GeoNeRF. The examples are selected from challenging views of the NeRF synthetic dataset [4]. Columns correspond to the experiments in Table 8.

volume is constructed per-view instead of cascaded multi-level cost volumes.

Figure 3 contains examples from the NeRF synthetic dataset [4] for qualitative analysis corresponding to the experiments in Table 8. The examples focus on challenging views of the scenes in order to contrast the behavior of the models properly.

5. Limitations

Our model with the experimental settings in the main article can be trained and evaluated on a single GPU with 16 GB of memory. Failure cases in our model could occur when the stereo reconstruction fails in the geometry reasoner, and the renderer is misled by incorrect geometry priors. Since the architecture of the geometry reasoner is inspired by multi-view stereo models, it is prone to failure in textureless areas similarly. Such failure examples are shown in Fig. 4.



Ground truth | Rendered image | Rendered depth

Figure 4. Failure examples in our method where stereo reconstruction fails in the geometry reasoner for textureless areas.

References

[1] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14124–14133, October 2021. 5, 6

[2] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2020. 1

[3] 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)*, 2019. 1, 2, 3, 5, 6, 7

[4] 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 *European conference on computer vision*, pages 405–421. Springer, 2020. 1, 2, 4, 5, 6, 7

[5] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 5, 6

[6] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5, 6

[7] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 5, 6

[8] 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, 6