Supplementary Material for MVSNeRF

1. Per-scene optimization.

More details. As described in the paper Sec. 3.4, we optimize the predicted neural encoding volume with the MLP decoder for each scene to achieve final high-quality fine-tuning results. The neural encoding volume with the MLP is an effective neural representation of a radiance field. All fine-tuning results (and also NeRF's optimization comparison results) are generated using a single NVIDIA RTX 2080Ti GPU. On this hardware, our 15min fine-tuning corresponds to 10k training iterations and NeRF's 10.2h optimization corresponds to 200k training iterations.

Our neural encoding volume is reconstructed in the frustrum of the reference view; it thus only covers the scene content in the frustrum. As a result, for a large scene, artifacts may appear when some parts that are not located in the frustrum show up in the novel view. Therefore, we extend the neural encoding volume by padding its boundary voxels when fine-tuning on some large scenes. This can address the out-of-frustrum artifacts, though the padding voxels are not well reconstructed initially by the network and may require longer fine-tuning to achieve high quality.

As described in the paper, we do not apply two-stage (coarse and fine) ray sampling as done in NeRF [2]. We uniformly sample points (with per-step small randomness) along each marching ray. We find that 128 points are enough for most scenes and keep using 128 point for our across-scene training on the DTU dataset. When fine-tuning, we increase the number of points to 256 for some challenging scenes.

Optimization progress. We have demonstrated in the paper that our 15min fine-tuning results of DTU and LLFF dataset are comparable with the 10.2h optimization results of NeRF [2]. We now show comparisons of the per-scene optimization progress between our fine-tuning and NeRF's from-scratch optimization. Note that, thanks to the strong initial reconstruction predicted by our network, our fine-tuning is consistently better than NeRF's optimization through 200k training iterations. As mentioned, our each 18min result corresponds to the result at the 12k-th training iteration, which is at the very early stage in the curves; however, as demonstrated, it can be already better than the NeRF's result after 48k iterations, corresponding to the 10.2h optimization result shown in the paper. Moreover, while our 15min results are already very good, our results can be further improved over more iterations, if continuing optimizing the radiance fields.

Layer	**k**	**s**	**d**	**chns**	input
CBR2D_0	3	1	1	$3/8$	I
CBR2D_1	3	1	1	$8/8$	CBR2D_0
CBR2D_2	5	2	2	$8/16$	CBR2D_1
CBR2D_3	3	1	1	$16/16$	CBR2D_2
CBR2D_4	3	1	1	$16/16$	CBR2D_3
CBR2D_5	5	2	2	$16/32$	CBR2D_4
CBR2D_6	3	1	1	$32/32$	CBR2D_5
T	3	1	1	$32/32$	CBR2D_6
CBR3D_0	3	1	1	$32+9/8$	T, I
CBR3D_1	3	2	1	$8/16$	CBR3D_0
CBR3D_2	3	1	1	$16/16$	CBR3D_1
CBR3D_3	3	2	1	$16/32$	CBR3D_2
CBR3D_4	3	1	1	$32/32$	CBR3D_3
CBR3D_5	3	2	1	$32/64$	CBR3D_4
CBR3D_6	3	1	1	$64/64$	CBR3D_5
CTB3D_0	3	2	1	$64/32$	CTB3D_0 + CBR3D_4
CTB3D_1	3	2	1	32/16	CTB3D_1 + CBR3D_2
CTB3D_2	3	2	1	$16/8$	CTB3D_2 + CBR3D_0
PE_0	-	-	-	3/63	x
LR_0	-	-	-	8+12/256	f, c
LR_1	-	-	-	63/256	PE
LR_{i+1}	-	-	-	256/256	$\text{LR}_i \cdot \text{LR}_0$
σ	-	-	-	256/1	LR_6
PE_1	-	-	-	3/27	d
LR_7	-	-	-	27+256/256	PE_1, LR_6
c	-	-	-	256/3	LR_7

Table 1. From top to bottom: 2D CNN based feature extraction model, 3D CNN based neural encoding volume prediction model and MLP based volume properties regression model ($i \in [1, ..., 5]$). **k** is the kernel size, **s** is the stride, **d** is the kernel dilation, and **chns** shows the number of input and output channels for each layer. We denote CBR2D/CBR3D/CTB3D/LR to be ConvBnReLU2D, ConvBnReLU3D, ConvTransposeBn3D and LinearRelu layer structure respectively. PE refers to the positional encoding as used in [2].

2. Network Architectures

We show detailed network architecture specifications of our 2D CNN (that extracts 2D image features), 3D CNN (that outputs a neural encoding volume), and MLP decoder (that regresses volume properties) in Tab 1.

3. Limitations.

Our approach generally achieves fast radiance field reconstruction for view synthesis on diverse real scenes. However, for highly challenging scenes with high glossiness/specularities, the strong view-dependent shading effects can be hard to directly recovered via network inference and a longer fine-tuning process can be required to fully reconstruct such effects. Our radiance field representation is reconstructed within the frustrum of the reference view. As a result, only the scene content seen by the reference view is well reconstructed and initialized for the following fine-tuning stage. Padding the volume (as discussed earlier) can incorporate content out of the original frustrum; however, the unseen parts (including those that are in the frustrucm but are occluded and invisible in the view) are not directly recovered by the network. Therefore, it is challenging to use a single neural encoding volume to achieve rendering in a wide viewing range around a scene (like 360° rendering). Note that, a long per-scene fine-tuning process with dense images covering around the scene can still achieve 360° rendering, though it can be as slow as training a standard NeRF [2] (or Sparse Voxel Fields [1] that is similar to our representation) to recover those uninitialized regions in the encoding volume. Combining multiple neural encoding volumes at multiple views can be an interesting future direction to achieve fast radiance field reconstruction with larger viewing ranges.

4. Per-scene breakdown.

We show the pre-scene breakdown analysis of the quantitative results presented in the main paper for the three dataset (*Realistic Synthetic*, *DTU* and *LLFF*).

These results are consistent with the averaged results shown in the paper. In general, since the training set consists of DTU scenes, all three methods can work reasonably well on the DTU testing set. Our approach can outperform PixelNeRF [4], when using the same three-image input, and achieve higher PSNR and SSIM and lower LPIPS. Note that, as mentioned in the paper, the implementation of IBRNet [3] is trained and tested with 10 input images to achieve its best performance as used in their paper. Nonetheless, our results with three input images are still quantitatively comparable to the results of IBRNet with 10 input images on the DTU testing set; IBRNet often achieves better PSNRs while we often achieve better SSIMs and LPIPSs.

DTU Dataset

Scan	#1	#8	#21	#103	#114
			PSNR↑		
PixelNeRF	21.64	23.70	16.04	16.76	18.40
IBRNet	25.97	**27.45**	20.94	27.91	27.91
Ours	**26.96**	27.43	**21.55**	**29.25**	**27.99**
$\text{NeRF}_{10.2h}$	26.62	28.33	23.24	30.40	26.47
IBRNet_{ft-1h}	**31.00**	**32.46**	**27.88**	**34.40**	**31.00**
$\text{Ours}_{ft-15min}$	28.05	28.88	24.87	32.23	28.47
			SSIM↑		
PixelNeRF	0.827	0.829	0.691	0.836	0.763
IBRNet	0.918	0.903	0.873	0.950	0.943
Ours	**0.937**	**0.922**	**0.890**	**0.962**	**0.949**
$\text{NeRF}_{10.2h}$	0.902	0.876	0.874	0.944	0.913
IBRNet_{ft-1h}	**0.955**	0.945	**0.947**	0.968	**0.964**
$\text{Ours}_{ft-15min}$	**0.934**	0.900	**0.922**	**0.964**	0.945
			LPIPS ↓		
PixelNeRF	0.373	0.384	0.407	0.376	0.372
IBRNet	0.190	0.252	0.179	0.195	0.136
Ours	**0.155**	**0.220**	**0.166**	**0.165**	**0.135**
$\text{NeRF}_{10.2h}$	0.265	0.321	0.246	0.256	0.225
IBRNet_{ft-1h}	**0.129**	**0.170**	**0.104**	**0.156**	**0.099**
$\text{Ours}_{ft-15min}$	0.171	0.261	0.142	0.170	0.153

Table 2. **Quantity comparison on five sample scenes in the DTU testing set**.

More importantly, as already shown in paper, when testing on novel datasets, our approach generalizes significantly better than PixelNeRF and IBRNet, leading to much better quantitative results on the Synthetic Data and the Forward-Facing dataset. We also provide detailed per-scene quantitative results for the three testing datasets in Tab. 3-10. Please also refer to the supplementary video for video comparisons.

References

[1] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020. 2

[2] 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

[3] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. *arXiv preprint arXiv:2102.13090*, 2021. 2

[4] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, 2020. 2

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
				PSNR↑				
PixelNeRF	7.18	8.15	6.61	6.80	7.74	7.61	7.71	7.30
IBRNet	**24.20**	18.63	21.59	27.70	22.01	**20.91**	22.10	22.36
Ours	23.35	**20.71**	**21.98**	**28.44**	**23.18**	20.05	**22.62**	**23.35**
NeRF	**31.07**	**25.46**	**29.73**	**34.63**	**32.66**	**30.22**	**31.81**	29.49
IBRNet$_{ft-1h}$	28.18	21.93	25.01	31.48	25.34	24.27	27.29	21.48
Ours$_{ft-15min}$	26.80	22.48	26.24	32.65	26.62	25.28	29.78	26.73
				SSIM↑				
PixelNeRF	0.624	0.670	0.669	0.669	0.671	0.644	0.729	0.584
IBRNet	0.888	0.836	0.881	0.923	0.874	0.872	0.927	0.794
Ours	**0.876**	**0.886**	**0.898**	**0.962**	**0.902**	**0.893**	**0.923**	**0.886**
NeRF	**0.971**	**0.943**	**0.969**	**0.980**	**0.975**	**0.968**	**0.981**	**0.908**
IBRNet$_{ft-1h}$	0.955	0.913	0.940	0.978	0.940	0.937	0.974	0.877
Ours$_{ft-15min}$	0.934	0.898	0.944	0.971	0.924	0.927	0.970	0.879
				LPIPS ↓				
PixelNeRF	0.386	0.421	0.335	0.433	0.427	0.432	0.329	0.526
IBRNet	**0.144**	0.241	**0.159**	0.175	**0.202**	**0.164**	**0.103**	0.369
Ours	0.282	**0.187**	0.211	**0.173**	0.204	0.216	0.177	**0.244**
NeRF	**0.055**	**0.101**	**0.047**	**0.089**	**0.054**	0.105	**0.033**	0.263
IBRNet$_{ft-1h}$	0.079	0.133	0.082	0.093	0.105	**0.093**	0.040	**0.257**
Ours$_{ft-15min}$	0.129	0.197	0.171	0.094	0.176	0.167	0.117	0.294

Table 3. **Quantity comparison on the Realistic Synthetic dataset**.

	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	Trex
				PSNR↑				
PixelNeRF	12.40	10.00	14.07	11.07	9.85	9.62	11.75	10.55
IBRNet	20.83	22.38	**27.67**	22.06	**18.75**	15.29	**27.26**	20.06
Ours	**21.15**	**24.74**	26.03	**23.57**	17.51	**17.85**	26.95	**23.20**
NeRF$_{10.2h}$	**23.87**	26.84	**31.37**	25.96	21.21	19.81	**33.54**	**25.19**
IBRNet$_{ft-1h}$	22.64	26.55	30.34	25.01	**22.07**	19.01	31.05	22.34
Ours$_{ft-15min}$	23.10	**27.23**	30.43	**26.35**	21.54	**20.51**	30.12	24.32
				SSIM↑				
PixelNeRF	0.531	0.433	0.674	0.516	0.268	0.317	0.691	0.458
IBRNet	**0.710**	0.854	**0.894**	0.840	**0.705**	0.571	0.950	0.768
Ours	0.638	**0.888**	0.872	**0.868**	0.667	**0.657**	**0.951**	**0.868**
NeRF$_{10.2h}$	**0.828**	0.897	**0.945**	0.900	0.792	0.721	**0.978**	**0.899**
IBRNet$_{ft-1h}$	0.774	0.909	0.937	0.904	**0.843**	0.705	0.972	0.842
Ours$_{ft-15min}$	0.795	**0.912**	0.943	**0.917**	0.826	**0.732**	0.966	0.895
				LPIPS ↓				
	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	Trex
PixelNeRF	0.650	0.708	0.608	0.705	0.695	0.721	0.611	0.667
IBRNet	0.349	0.224	**0.196**	0.285	**0.292**	0.413	**0.161**	0.314
Ours	**0.238**	**0.196**	0.208	**0.237**	0.313	**0.274**	0.172	**0.184**
NeRF$_{10.2h}$	0.291	0.176	0.147	0.247	0.301	0.321	0.157	0.245
IBRNet$_{ft-1h}$	0.266	0.146	**0.133**	0.190	**0.180**	0.286	**0.089**	0.222
Ours$_{ft-15min}$	**0.253**	**0.143**	0.134	**0.188**	0.222	**0.258**	0.149	**0.187**

Table 4. **Quantity comparison on the Forward Facing dataset**.