

Joint Optimization of Neural Radiance Fields and Continuous Camera Motion from a Monocular Video

—Supplementary Material—

Hoang Chuong Nguyen¹ Wei Mao Jose M. Alvarez² Miaomiao Liu¹
¹Australian National University ²NVIDIA
{hoangchuong.nguyen, miaomiao.liu}@anu.edu.au josea@nvidia.com

A. Implementation Details

In this section, we provide more details about the datasets and our training setting. For each scene used in our experiments, we summarize the statistics and training hyper-parameters in Tab. 1.

A1. Datasets

We evaluate our method using 4 scenes from the ScanNet [2] dataset, 5 scenes from the Co3D [9] dataset, and 8 scenes from the Tanks & Temples (TNT) [6] dataset. Tab. 1 shows that scenes in the TNT dataset tend to have smaller rotations compared to those in the other two datasets. We adopt the same train/test splits as in previous works [1, 4]. Specifically, for most scenes, we reserve every 8th image for depth and novel-view synthesis evaluation, while the remaining images are used for training. For the Family scene in the TNT dataset, we follow [1, 4] and use a test sampling rate of 2 to analyze the influence of different sampling rates on our training pipeline. Regarding pose evaluation, the learned poses of images in the training set are evaluated against the provided ground-truth poses. For Co3D and Scannet dataset, we utilize the provided poses as ground-truth, whereas the ground-truth poses for TNT dataset are generated using COLMAP [10], as in [1].

A2. Training Details

Tab. 1 lists the hyper-parameters used to train our method on different scenes. At each training iteration, we sample 1024 rays (i.e., pixels) and 128 3D points per ray within a pre-defined depth range. We use a sampled depth range of 5 for the Co3D and Scannet datasets. For the TNT dataset, we set a higher depth range of 10 as this dataset mainly includes outdoor scenes. The parameters such as weights for losses and intervals for neighboring frames for training the time-dependent NeRF, are validated for each dataset.

Our pipeline starts with the joint optimization of the time-dependent NeRF and the camera motions until convergence. Then, we use the learned camera poses to fine-

tune our NeRF model for another 5000 epochs. During the fine-tuning, we set the loss weights of $\mathcal{L}_{\text{flow}}$, $\mathcal{L}_{\text{photo}}$, \mathcal{L}_{sdf} to 0 and reduce the learning rate by a factor of 0.9954 every 10 epochs. During training, we also leverage the standard edge-aware smoothness loss [5] to encourage local smoothness of the rendered depth maps. In Tab. 2, we present additional experimental results where we train our model for the same training time as NoPe-NeRF. Given the similar training time, our method outperforms NoPe-NeRF across almost all metrics.

A3. Evaluation Metrics

Novel View Synthesis. We evaluate the novel-view synthesis results using the standard metrics which include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [12] and Learned Perceptual Image Patch Similarity (LPIPS) [16]. Following [1, 4], LPIPS is computed using VGG model [11].

Pose. Given the learned camera pose $\hat{\mathbf{P}}$ and the corresponding ground-truth pose \mathbf{P} , the Absolute Trajectory Error (ATE) is computed as follows,

$$\text{ATE} = \|\hat{\mathbf{t}} - \mathbf{t}\|_2 \quad (1)$$

where $\hat{\mathbf{t}}$ and \mathbf{t} are the translation vectors extracted from the learned and ground-truth poses, respectively.

Given the learned relative pose $\hat{\mathbf{P}}_{\text{rel}}$ between two consecutive frames in a video and its corresponding ground-truth \mathbf{P}_{rel} , the Relative Translation Error (RPE_t) and Relative Rotation Error (RPE_r) are defined as,

$$\mathbf{P}_{\text{error}} = (\mathbf{P}_{\text{rel}})^{-1} \hat{\mathbf{P}}_{\text{rel}} \quad (2)$$

$$\text{RPE}_t = \|\mathbf{t}_{\text{error}}\|_2 \quad (3)$$

$$\text{RPE}_r = \cos^{-1} \left(\frac{\text{trace}(\mathbf{R}_{\text{error}}) - 1}{2} \right) \quad (4)$$

with $\mathbf{R}_{\text{error}}$ and $\mathbf{t}_{\text{error}}$ being the rotation matrix and translation vector in $\mathbf{P}_{\text{error}}$, respectively. The function $\text{trace}(\cdot)$ computes the trace of a matrix.

Statistics								Training hyper-parameters					
	Scene	Type	Length	Resolution	Max rotation angle (degree) between any two frames	Max rotation angle (degree) between two consecutive frames	Test sampling rate	Loss weights				Neighboring frame intervals \mathcal{N}	Sampled depth range
								λ_1 (\mathcal{L}_{eik})	λ_2 ($\mathcal{L}_{\text{flow}}$)	λ_3 ($\mathcal{L}_{\text{photo}}$)	λ_4 (\mathcal{L}_{sdr})		
Scannet	0079_00	Indoor	90	484×648	54.4	2.04	8	0.1	0.1	7.5	1.0	[1,2,3]	[0.01,5]
	0431_00	Indoor	100		27.5	2.78	8					[1,2,3]	
	0418_00	Indoor	80		45.8	3.68	8					[1,2,3]	
	0301_00	Indoor	100		43.7	2.21	8					[1,5,10]	
Co3D	Bench	Outdoor	202	712×1266	180	5.70	8	0.1	0.1	7.5	1.0	[1,2,3]	[0.01,5]
	Teddy	Indoor	202	858×481	180	5.03	8						
	Plant	Indoor	202	1895×1065	180	5.35	8						
	Hydrant	Outdoor	202	1267×712	180	4.96	8						
	Skateboard	Indoor	202	717×1275	180	7.70	8						
Tanks Temples	Church	Indoor	400	540×960	37.3	0.60	8	0.1	0.1	5	1.0	[1,3,5]	[0.01,10]
	Horse	Outdoor	120		39.0	0.97	8					[1,3,5]	
	Francis	Outdoor	150		47.5	1.39	8					[1,3,5]	
	Barn	Outdoor	150		47.5	1.99	8					[1,5,10]	
	Museum	Indoor	100		76.2	1.96	8					[1,5,10]	
	Ignatius	Outdoor	120		26.0	1.09	8					[1,5,10]	
	Ballroom	Indoor	150		30.3	1.05	8					[1,5,10]	
	Family	Outdoor	200		35.4	0.63	2					[1,5,10]	

Table 1. Details of each video and the training hyper-parameters used in our experiments.

	Scene	Ours						Ours (fast)						NoPe-NeRF (D)					
		Image (PSNR)	Depth (AbRel)	Pose RPE _t	Pose RPE _r	Pose ATE	Training time (hours)	Image (PSNR)	Depth (AbRel)	Pose RPE _t	Pose RPE _r	Pose ATE	Training time (hours)	Image (PSNR)	Depth (AbRel)	Pose RPE _t	Pose RPE _r	Pose ATE	Training time (hours)
Scannet	0079.00	35.52	<u>0.006</u>	<u>0.664</u>	0.182	0.013	~ 42	<u>34.00</u>	0.005	0.623	<u>0.188</u>	<u>0.014</u>	~ 12	32.47	0.047	0.673	0.190	0.015	~ 12
	0418.00	34.53	0.067	<u>0.401</u>	<u>0.119</u>	0.015	~ 37	<u>33.06</u>	<u>0.088</u>	0.399	0.118	<u>0.016</u>	~ 10	31.33	0.137	0.455	<u>0.119</u>	0.015	~ 10
	0301.00	32.06	0.012	0.367	0.117	0.009	~ 46	<u>31.05</u>	<u>0.014</u>	<u>0.387</u>	<u>0.120</u>	<u>0.011</u>	~ 13	29.83	0.252	0.393	<u>0.118</u>	0.015	~ 13
	0431.00	34.25	0.016	<u>1.097</u>	<u>0.223</u>	<u>0.042</u>	~ 46	<u>33.85</u>	<u>0.018</u>	1.045	0.203	0.039	~ 13	33.83	0.111	1.301	0.269	0.064	~ 13
Co3D	Bench	26.36	0.085	0.013	<u>0.037</u>	0.001	~ 110	<u>25.68</u>	<u>0.229</u>	0.013	0.035	0.001	~ 26	24.32	0.945	0.301	1.925	0.053	~ 26
	Skateboard	30.93	0.013	0.024	0.063	0.001	~ 110	<u>27.23</u>	<u>0.021</u>	<u>0.031</u>	<u>0.125</u>	<u>0.005</u>	~ 26	26.22	0.527	0.421	1.883	0.048	~ 26
	Plant	26.53	0.526	0.014	0.062	0.002	~ 110	<u>25.82</u>	<u>0.811</u>	<u>0.019</u>	<u>0.108</u>	<u>0.004</u>	~ 26	23.79	1.816	0.305	1.587	0.047	~ 26
	Hydrant	20.60	0.064	0.010	0.027	0.001	~ 110	<u>20.54</u>	<u>0.081</u>	<u>0.011</u>	0.027	0.001	~ 26	19.82	0.677	0.337	1.557	0.060	~ 26
	Teddy	33.04	<u>0.175</u>	0.053	0.129	0.004	~ 110	<u>31.68</u>	0.147	<u>0.100</u>	<u>0.308</u>	<u>0.013</u>	~ 26	29.40	0.823	0.286	1.295	0.040	~ 26

Table 2. Comparison between our method and NoPe-NeRF [1] in terms of training time. With similar training time, our method still outperforms NoPe-NeRF on most metrics.

Depth. In terms of depth evaluation, with $\hat{\mathbf{D}}$ and \mathbf{D} denoting the rendered and ground-truth depth maps, we report our results using the following metrics [3],

$$\text{AbRel} = \frac{1}{|\mathcal{V}|} \sum_{(u,v) \in \mathcal{V}} \frac{|\hat{\mathbf{D}}(u,v) - \mathbf{D}(u,v)|}{\mathbf{D}(u,v)}, \quad (5)$$

$$\text{SqRel} = \frac{1}{|\mathcal{V}|} \sum_{(u,v) \in \mathcal{V}} \frac{(\hat{\mathbf{D}}(u,v) - \mathbf{D}(u,v))^2}{\mathbf{D}(u,v)}, \quad (6)$$

$$\delta_1 = \frac{\left| \left\{ (u,v) \mid \max \left(\frac{\mathbf{D}(u,v)}{\hat{\mathbf{D}}(u,v)}, \frac{\hat{\mathbf{D}}(u,v)}{\mathbf{D}(u,v)} \right) < 1.25, (u,v) \in \mathcal{V} \right\} \right|}{|\mathcal{V}|}, \quad (7)$$

where \mathcal{V} is a set of pixel coordinates corresponding to valid ground-truth depth values.

B. Additional Results

B1. Comparison with CF-NeRF on NeRFBuster dataset

Tab. 3 shows the comparison between our method and CF-NeRF [15]. Following [15], we use the same evaluation metrics and train our method on the similar scenes in the NeRFBuster dataset [14]. On average, our rotation error and translation error are **4.80°** and **2.85**, respectively. On the other hand, CF-NeRF produces an average rotation error of 11.27° and an average translation error of 3.53. The result shows that our method outperforms CF-NeRF in estimating both the camera rotation and translation.

		aloe	art	car	century	flowers	garbage	picnic	pikachu	pipe	plant	roses	table	mean
$\Delta R \downarrow$	CF-NeRF	12.127	19.250	17.557	9.6811	8.2556	9.7658	12.650	11.307	19.993	4.8968	5.1229	4.5837	11.265
	Ours	3.9385	0.9963	3.4539	11.739	2.0427	2.9629	2.7478	8.2785	4.1264	7.8968	4.5565	4.8696	4.8007
$\Delta T \downarrow$	CF-NeRF	3.3788	2.2821	6.5452	2.7383	2.7026	4.0535	1.2833	4.0586	9.4491	3.4346	1.1945	1.2127	3.5278
	Ours	3.8158	2.2038	2.4453	2.2434	1.5863	2.9602	1.7982	4.5326	3.1821	6.4346	1.5598	1.3870	2.8458

Table 3. Comparison with CF-NeRF [15] on the NeRFBuster dataset.

Scenes	Ours			COLMAP [10]		
	RPE _t ↓	RPE _r ↓	ATE ↓	RPE _t	RPE _r	ATE
0079	0.664	0.182	0.013	0.655	0.221	0.012
0418	0.401	0.119	0.015	0.491	0.124	0.016
0301	0.367	0.117	0.009	0.414	0.136	0.009
0431	1.097	0.223	0.042	1.292	0.249	0.051

Table 4. Poses comparison with COLMAP [10] on Scannet.

Scenes	DPT [8]	NoPe-NeRF [1]		Ours	
	AbRel ↓	AbRel ↓	PSNR ↑	AbRel	PSNR
0079	0.149	0.099	32.47	0.036	35.52
0418	0.190	0.152	31.33	0.144	34.53
0301	0.317	0.185	29.83	0.037	32.06
0431	0.132	0.127	33.83	0.038	34.25

Table 5. Comparison between our method, NoPe-NeRF [1] and its depth prior DPT [8] in depth estimation (AbRel) and image synthesis (PSNR) on ScanNet dataset.

B2. Comparison with COLMAP poses

Tab. 4 shows the pose error produced by our method and COLMAP [10] on Scannet dataset. It can be seen that our method yields more accurate poses in almost all cases. Notably, we consistently outperform COLMAP in estimating the relative camera rotation in all scenes. On average, our method reduces the relative rotation error RPE_r by 11%. This result motivates the joint optimization of NeRF and camera poses, rather than relying on COLMAP poses.

B3. Comparison with NoPe-NeRF’s depth prior.

We show the comparison between our method, NoPe-NeRF [1], and its depth prior, DPT [8], in depth estimation and image synthesis on the ScanNet dataset in Tab. 5. The results reveal that NoPe-NeRF is heavily dependent on the quality of the depth priors it uses. For example, poor depth quality produced by DPT in scene ‘0301’ significantly degrades NoPe-NeRF’s performance. However, our method is not constrained by the quality of a depth prior and is able to render more accurate depth maps compared to both NoPe-NeRF and its prior.

Scenes	Ours			MonoGS [7]		
	AbRel _t ↓	RPE _r ↓	ATE ↓	AbRel _t	RPE _r	ATE
0079	0.033	0.363	0.013	0.128	1.180	0.035
0418	0.147	0.481	0.017	0.272	0.978	0.021
0301	0.039	0.309	0.008	0.084	0.308	0.010
0431	0.038	0.902	0.045	0.163	1.080	0.059

Table 6. Comparison with MonoGS [7] on estimating the poses and depth maps at MonoGS’s keyframes.

	Depth		Pose			Image	
	AbRel ↓	$\delta_1 \uparrow$	RPE _t ↓	RPE _r ↓	ATE ↓	PSNR ↑	SSIM ↑
Full	0.064	0.951	<u>0.632</u>	0.161	0.020	<u>34.089</u>	<u>0.913</u>
w/o $\mathcal{L}_{\text{flow}}$	<u>0.079</u>	<u>0.907</u>	<u>0.637</u>	<u>0.163</u>	0.020	34.024	0.911
w/o \mathcal{L}_{sdf}	0.095	0.857	0.631	0.161	0.020	34.014	0.911
w/o $\mathcal{L}_{\text{photo}}$	0.399	0.508	2.798	0.655	0.089	24.385	0.780
w/o NeRF _t	0.274	0.621	5.709	0.676	0.239	25.439	0.797
w/o NeRF _{t→c}	0.113	0.838	—	—	—	35.176	0.928
w/o motion network	0.125	0.856	0.702	0.198	<u>0.025</u>	34.047	0.909

Table 7. Ablation study on the Scannet dataset.

B4. Comparison with MonoGS

Tab. 6 shows the comparison between our method and a GS-based method, MonoGS [7]. Similar to our method, MonoGS does not require a geometric prior and can jointly optimize camera poses and scene geometry from an RGB video with random initialization. Since MonoGS only estimates camera poses at keyframes, we compare the accuracy of the poses and depth maps estimated at these keyframes. The results show that our method significantly outperforms MonoGS in both estimating camera poses and depth maps.

B5. Ablation study on Scannet dataset

In Tab. 7, we show the ablation study on Scannet dataset. The results show that removing each of the loss terms leads to a performance degradation. However, training without the time-dependent NeRF (w/o NeRF_t) or not fine-tuning the time-dependent NeRF in a later training stage (w/o NeRF_{t→c}) reduces the geometric accuracy of the model. Lastly, our full model also outperforms an instance in which we remove the motion network while optimizing camera poses as variables (w/o motion net).

B6. Results on the Tanks & Temples dataset

Here, we show the comparison between our method, NeRFmm [13], NoPe-NeRF [1], and CF3DGS [4] on the TNT dataset. As there is no ground-truth depth map provided, we only show the quantitative comparison in terms of pose estimation (Tab. 8) and novel-view synthesis (Tab. 9). We further show the qualitative comparison for depth estimation in Fig. 1 and Fig. 2

Camera Poses Our method performs on par with NoPe-NeRF and CF3DGS in camera pose estimation. As scenes in the TNT dataset tend to have smaller camera rotations (see Tab. 1) and smoother camera trajectories (see Fig. 1, Fig. 2) compared to the other two datasets, NoPe-NeRF does not struggle in learning accurate camera poses for most of the scenes. However, in the Museum scene which has the largest camera rotation in the TNT dataset, the pose errors of NoPe-NeRF are at least 300% higher than ours.

Novel-view Synthesis. Regarding novel-view synthesis, our method significantly outperforms the other NeRF-based methods [1, 12], although it achieves lower image quality compared to CF3DGS. This can be explained as, given camera poses of equal quality, this 3DGS-based approach excels at synthesizing high-frequency details in the images more effectively than our NeRF-based method. However, when CF3DGS encounters difficulties in learning accurate camera poses, its rendered images appear less photo-realistic compared to ours, as evidenced by the novel-view synthesis results on the Co3D and Scannet datasets (see Tab. 2 and Fig. 3 in the main paper). We conjecture that the pre-trained depth network utilized by CF3DGS performs well on the TNT dataset; however, its predictions are less accurate on Scannet and Co3D, thus leading to the inconsistent performance of CF3DGS. In contrast, the performance of our method is more stable across different scenes, as we do not rely on any priors. Additionally, due to the high geometric errors of CF3DGS, its rendered images may contain clear artifacts as shown the first sample in Fig. 1

Geometry. The visualizations in Fig. 1 and Fig. 2 demonstrate that our rendered depth maps are significantly more accurate than those produced by NoPe-NeRF [1] and CF3DGS [4] in all scenes.

B7. More Qualitative Results

In this section, we present more qualitative results for the Co3D (Fig. 3) and Scannet datasets (Fig. 4). We also provide supplemental videos showcasing the learned poses, rendered images and depth maps to further illustrate the superiority of our approach compared to the other methods.

References

- [1] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023. 1, 2, 3, 4, 5, 6, 7, 9
- [2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1
- [3] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 2
- [4] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and XiaoLong Wang. Colmap-free 3d gaussian splatting. 2023. 1, 4, 5, 6, 7, 9
- [5] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2360–2367, 2013. 1
- [6] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 1
- [7] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian Splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3
- [8] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 3
- [9] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021. 1
- [10] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 3
- [11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1
- [12] 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. 1, 4
- [13] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 4, 5, 9
- [14] Frederik Warburg, Ethan Weber, Matthew Tancik, Aleksander Holynski, and Angjoo Kanazawa. Nerfbusters: Removing ghostly artifacts from casually captured nerfs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18120–18130, 2023. 2
- [15] Qingsong Yan, Qiang Wang, Kaiyong Zhao, Jie Chen, Bo Li, Xiaowen Chu, and Fei Deng. Cf-nerf: Camera parameter

Scenes	Ours			NeRFmm [13]			NoPe-NeRF (D) [1]			CF3DGS (D) [4]			
	RPE _t ↓	RPE _r ↓	ATE ↓	RPE _t	RPE _r	ATE	RPE _t	RPE _r	ATE	RPE _t	RPE _r	ATE	
Tanks and Temples	Church	0.028	<u>0.010</u>	<u>0.005</u>	0.626	0.127	0.065	<u>0.034</u>	0.008	0.008	0.008	0.018	0.002
	Barn	0.032	0.013	<u>0.004</u>	1.629	0.494	0.159	0.046	<u>0.032</u>	<u>0.004</u>	<u>0.034</u>	0.034	0.003
	Museum	<u>0.064</u>	0.018	0.002	4.134	1.051	0.346	0.207	0.202	0.020	0.052	<u>0.215</u>	<u>0.005</u>
	Family	<u>0.036</u>	<u>0.026</u>	0.003	2.743	0.537	0.120	0.047	0.015	0.001	0.022	0.024	<u>0.002</u>
	Horse	0.185	<u>0.044</u>	<u>0.004</u>	1.349	0.434	0.018	<u>0.179</u>	0.017	0.003	0.112	0.057	0.003
	Ballroom	0.025	0.010	0.001	0.449	0.177	0.031	0.041	<u>0.018</u>	<u>0.002</u>	<u>0.037</u>	0.024	0.003
	Francis	0.027	<u>0.033</u>	0.003	1.647	0.618	0.207	0.057	0.009	0.005	<u>0.029</u>	0.154	0.006
	Ignatius	0.024	<u>0.013</u>	<u>0.003</u>	1.302	0.379	0.041	<u>0.026</u>	0.005	0.002	0.033	0.032	0.005

Table 8. Pose evaluation on the Tanks & Temples dataset

Scenes	Ours			NeRFmm [13]			NoPe-NeRF (D) [1]			CF3DGS (D) [4]			
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	
Tanks and Temples	Church	<u>26.94</u>	<u>0.83</u>	<u>0.15</u>	21.640	0.580	0.540	25.17	0.73	0.39	30.23	0.93	0.11
	Barn	<u>27.09</u>	<u>0.76</u>	<u>0.32</u>	23.210	0.610	0.530	26.35	0.69	0.44	31.23	0.90	0.10
	Museum	<u>29.11</u>	<u>0.86</u>	<u>0.19</u>	22.370	0.610	0.530	26.77	0.76	0.35	29.91	0.91	0.11
	Family	<u>28.03</u>	<u>0.84</u>	<u>0.26</u>	23.040	0.580	0.560	26.01	0.74	0.41	31.27	0.94	0.07
	Horse	<u>28.32</u>	<u>0.87</u>	<u>0.19</u>	23.120	0.700	0.430	27.64	0.84	0.26	33.94	0.96	0.05
	Ballroom	<u>29.31</u>	<u>0.90</u>	<u>0.15</u>	20.030	0.480	0.570	25.33	0.72	0.38	32.47	0.96	0.07
	Francis	<u>30.57</u>	<u>0.85</u>	<u>0.26</u>	25.400	0.690	0.520	29.48	0.80	0.38	32.72	0.91	0.14
	Ignatius	<u>25.36</u>	<u>0.71</u>	<u>0.36</u>	21.160	0.450	0.600	23.96	0.61	0.47	28.43	0.90	0.09

Table 9. Image evaluation on the Tanks & Temples dataset

free neural radiance fields with incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6440–6448, 2024. 2, 3

- [16] 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. 1

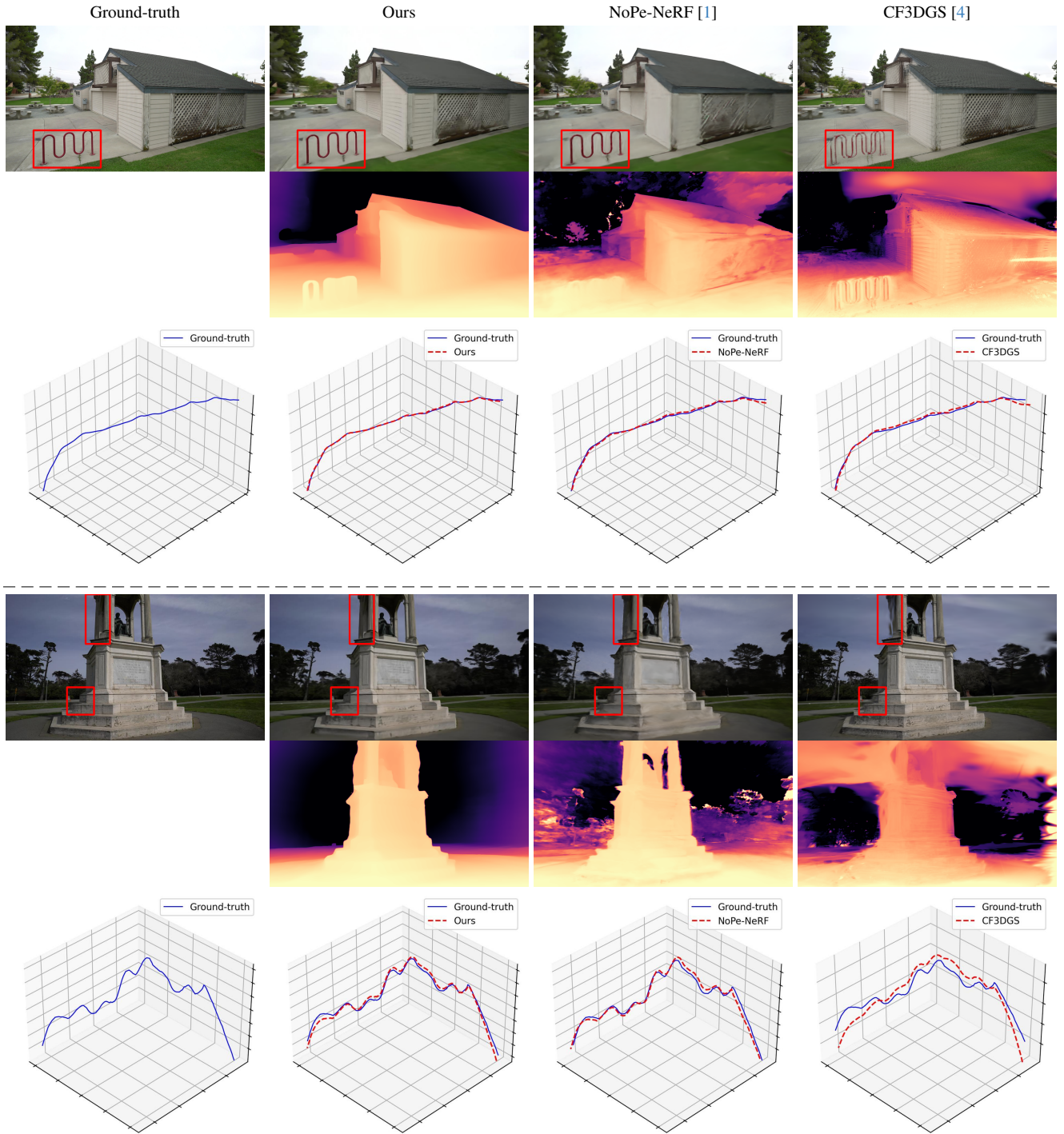


Figure 1. Qualitative results on the Tanks and Temples dataset (1). For each sample, we show images (top), depth maps (middle), camera trajectory (bottom).

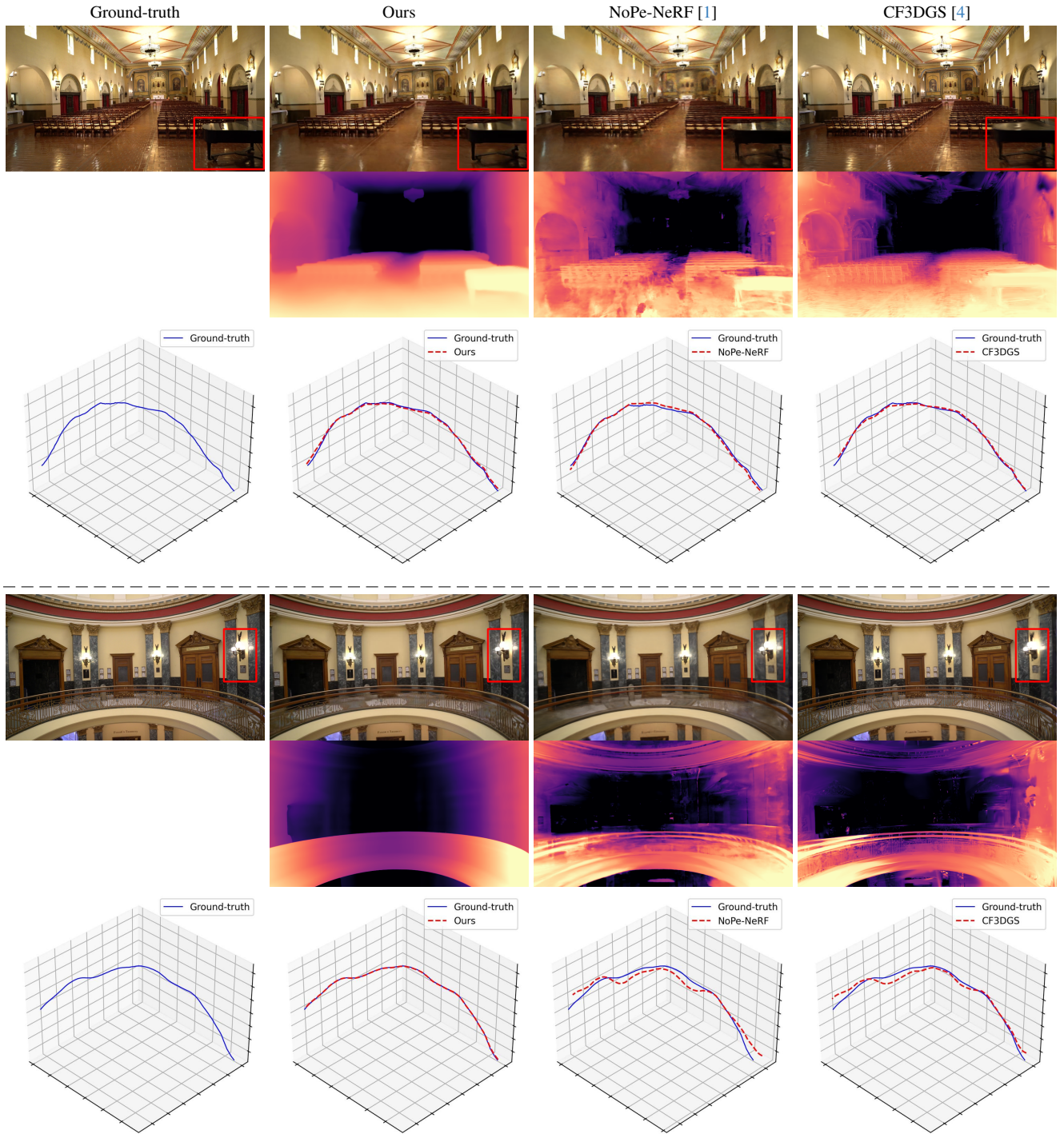


Figure 2. Qualitative results on the Tanks and Temples dataset (2). For each sample, we show images (top), depth maps (middle), camera trajectory (bottom).

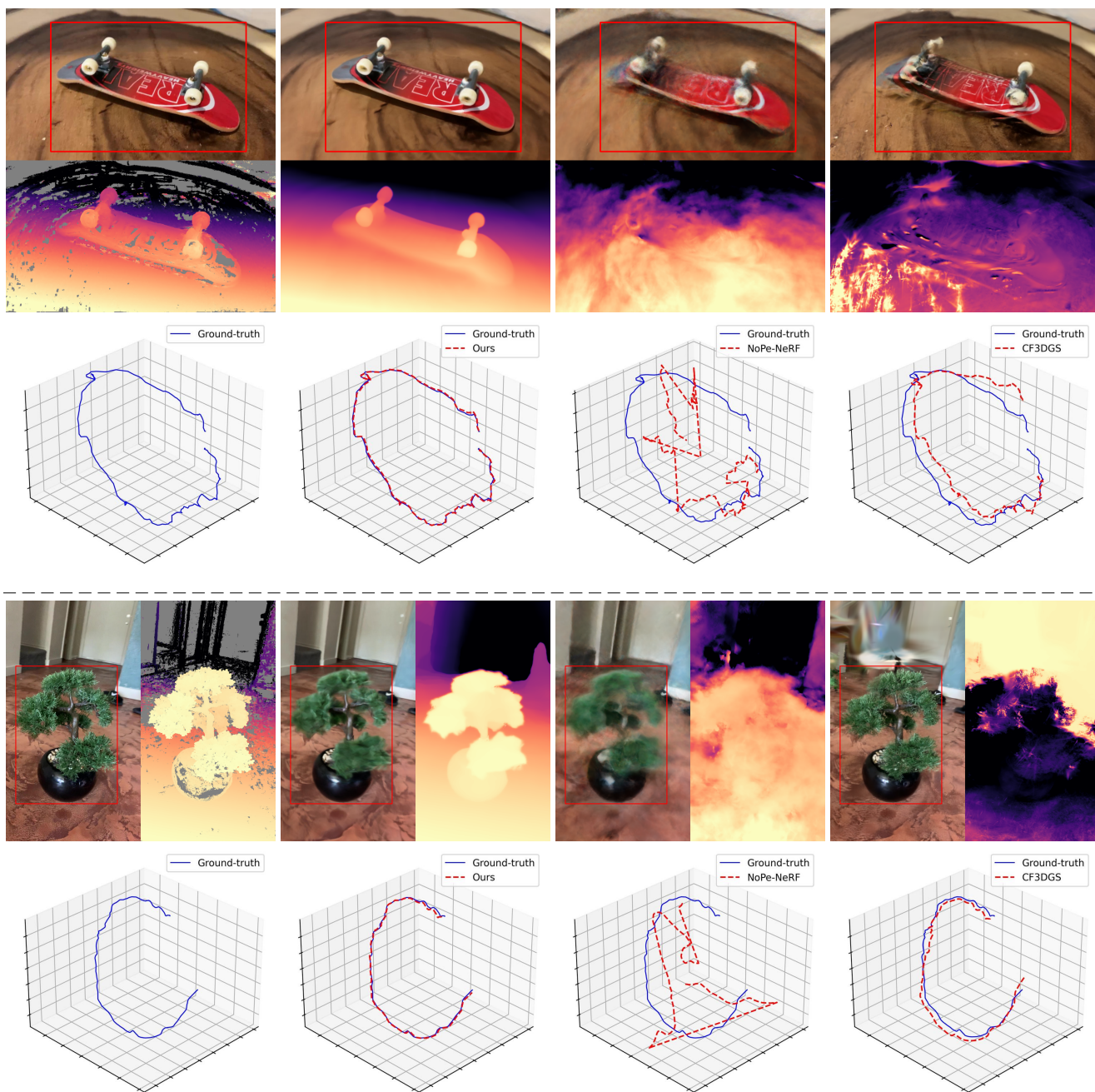


Figure 3. Qualitative results on the Co3D dataset. For each sample, we show images and depth maps (top), camera trajectory (bottom).

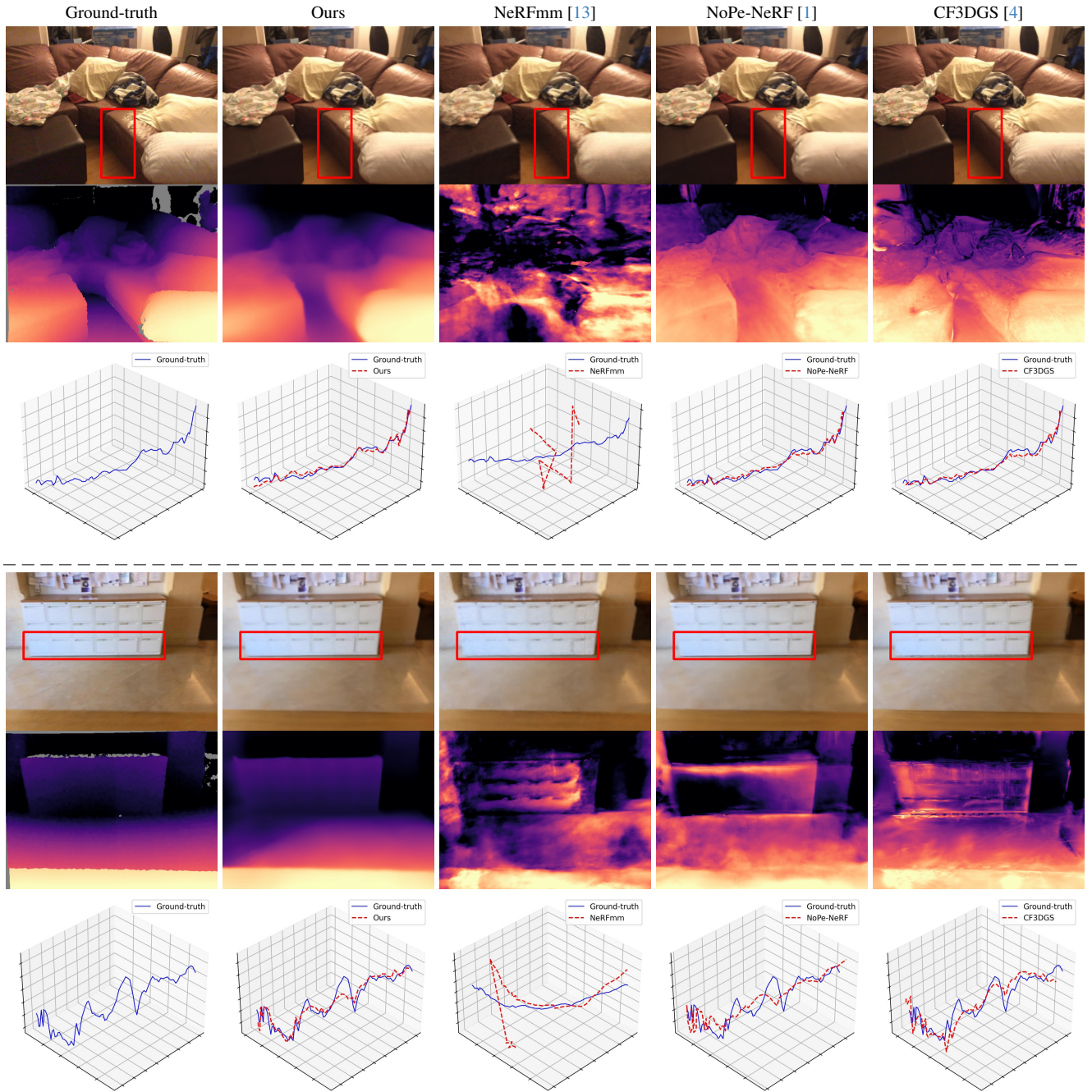


Figure 4. Qualitative results on the Scannet dataset. For each sample, we show images (top), depth maps (middle), camera trajectory (bottom).