Supplementary Material for: Invertible Neural Warp for NeRF

Shin-Fang Chng, Ravi Garg, Hemanth Saratchandran, and Simon Lucey

Adelaide University
Australian Institute for Machine Learning
shinfang.chng@adelaide.edu.au
https://sfchng.github.io/ineurowarping-github.io/

In this supplementary material, we provide additional details about our approach, which include the network architecture, experiment settings and results. Sec. A details the training and implementation details for our approach as well as baselines. Additional results are presented in Sec. B.

A Training and Implementation Details

A.1 Our approach

Architecture detail of Invertible Neural Network (INN) We illustrate the design of our invertible network $h_{\Theta_\mathcal{W}}$ in Fig. 1, see [2] for an in-depth explanation about the INN. The network is composed of three main blocks, each containing a multilayer perceptron (MLP). These blocks are different in their pattern of partitioning input coordinates along different axes. Within each block, a randomly chosen axis is used to split the input coordinates (for e.g., $[x, y, z]^T$). The first subset of these coordinates (for e.g., z), along with a frame-specific latent code Φ_t, is fed into the MLP. The MLP computes a 2D rotation and translation. This transformation is then applied to the second subset of the input coordinates (for e.g., $[x, y]^T$). This procedure is repeated for each of the other coordinates in the subsequent blocks. For all our experiments, we use the same network architecture. We use three blocks, with each block consisting of two layers of 128-dim hidden units. Following [2,11], we also apply positional encoding [8] to the MLP's input coordinates, and we set the number of frequencies used in this positioning encoding as 6. For the frame-specific latent code, we use a 16-dimensional code for 2D experiments in Sec. 4.2 of the main paper, and a 128-dimensional code for 3D experiments covered in Secs. 4.3 to 4.6 of the main paper.

INN (Ours) on LLFF Following BARF [6], we resize the images to 480×640. At each optimization steps, we randomly sample 2048 pixel rays. We use the same NeRF architecture as in BARF and L2G, see Sec. A.1 for details of the INN architecture. We used the learning rate for $\mathbf{\Theta}_{rgb}$ at 1×10^{-3}, which decays to 1×10^{-4}. For the INN field $\mathbf{\Theta}_\mathcal{W}$, the learning rate starts at 5×10^{-4} and decays to 1×10^{-8}. For the frequencies used for positional encoding $\gamma(.)$ defined

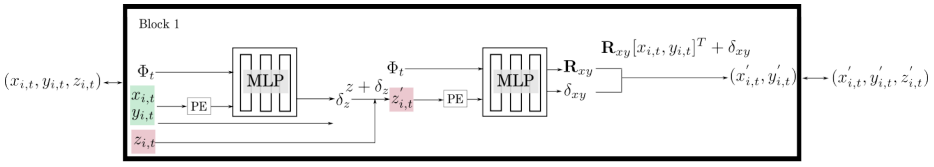


Fig. 1: Network architecture for the warp field $h_{\boldsymbol{\Theta}_\mathcal{W}}$. This diagram displays only the first block of the architecture, noting that subsequent blocks follow the same structure but differ in their splitting patterns. We denote PE as positional encoding. As mentioned in the paper, this architecture is fully invertible by design.

Table 1: Weighting term λ for the rigidity prior $\mathcal{L}_{rigid}$ in Eq. (8) of the main paper used to obtain the results on the LLFF dataset (Sec. 4.4 of the main paper).

Scenes	Weighting term for $\mathcal{L}_{rigid}$
fern	10^4
flower	10^4
fortress	10^5
horns	10^4
leaves	10^3
orchids	10^3
trex	10^4
room	10^3

in Sec. 3.1 of the main paper, we use $L = 10$ for 3D points and $L = 4$ for the viewing direction. The coarse-to-fine scheduler for BARF is linearly adjusted from iteration $20K$ to $100K$. Additionally, we provide the weighting term λ for $\mathcal{L}_{rigid}$ in Eq. (8) of the main paper used for each scene in Tab. 1.

Table 2: Weighting term λ for the rigidity prior $\mathcal{L}_{rigid}$ in Eq. (8) of the main paper used to obtain the results on the DTU dataset (Sec. 4.5 of the main paper).

Scans	24	37	40	55	63	65	69	83	97	105	106	110	114	118
Weighting term for $\mathcal{L}_{rigid}$	10^4	10^4	10^2	10^2	10^3	10^3	10^2	10^3	10^3	10^2	10^3	10^3	10^2	10^3

INN (Ours) on DTU Following [9], we resize the images to 300×400. At each optimization steps, we randomly sample 1024 pixel rays. We use the same NeRF architecture as in BARF and L2G, see Sec. A.1 for details of the INN architecture. We used the learning rate for $\boldsymbol{\Theta}_{rgb}$ at 1×10^{-3}, which decays to 3×10^{-4}. For the INN field $\boldsymbol{\Theta}_\mathcal{W}$, the learning rate starts at 5×10^{-4} and decays to 1×10^{-6}. For the frequencies used for positional encoding $\gamma(.)$ defined in Sec. 3.1 of the main paper, we use $L = 10$ for 3D points and $L = 4$ for the

viewing direction. The coarse-to-fine scheduler for BARF is linearly adjusted from iteration $20K$ to $100K$. Additionally, we provide the weighting term λ for $\mathcal{L}_{rigid}$ in Eq. (8) of the main paper used for each scene in Tab. 2.

A.2 Baselines

BARF [6] on LLFF We use the original code implementation provided by the authors [6]. For completeness, we include some details about the hyperparameters here, see [6] for full details. We used the learning rate for $\mathbf{\Theta}_{rgb}$ at 1×10^{-3}, which decays to 1×10^{-4}. For the pose P, the learning rate starts at 3×10^{-3} and decays to 1×10^{-5}. For the frequencies used for positional encoding $\gamma(.)$ defined in Sec. 3.1 of the main paper, we use $L = 10$ for 3D points and $L = 4$ for the viewing direction. The coarse-to-fine scheduler for BARF is linearly adjusted from iteration $20K$ to $100K$.

L2G [3] on LLFF We use the original code implementation including their proposed settings for architecture, coarse-to-fine scheduling, weighting term λ and hyperparameters [3]. For completeness, we include some details about the settings here, see [3] for full details. The learning rate for $\mathbf{\Theta}_{rgb}$ is set at 1×10^{-3}, which decays to 1×10^{-4}. For the warp field $\mathbf{\Theta}_{\mathcal{W}}$, the learning rate starts at 3×10^{-3} and decays to 1×10^{-8}. For the weighting term λ, we use their default settings, as outlined in their supplementary materials. For the frequencies used for positional encoding $\gamma(.)$ defined in Sec. 3.1 of the main paper, we use $L = 10$ for 3D points and $L = 4$ for the viewing direction. The coarse-to-fine scheduler for BARF is linearly adjusted from iteration $20K$ to $100K$.

BARF [6] on DTU As BARF [6] has not been tested on DTU datasets, we used the settings proposed by the author for training BARF on the Blender dataset as both Blender and DTU are comprised of 360° scenes. We used the learning rate for $\mathbf{\Theta}_{rgb}$ at 5×10^{-4}, which decays to 1×10^{-4}. For the pose P, the learning rate starts at 1×10^{-3} and decays to 1×10^{-5}. For the frequencies used for positional encoding $\gamma(.)$ defined in Sec. 3.1 of the main paper, we use $L = 10$ for 3D points and $L = 4$ for the viewing direction. The coarse-to-fine scheduler for BARF is linearly adjusted from iteration $20K$ to $100K$.

L2G [3] on DTU As L2G [3] has not been tested on DTU datasets, we used the settings proposed by the author for training L2G on the Blender dataset as both Blender and DTU are comprised of 360° scenes. We used the learning rate for $\mathbf{\Theta}_{rgb}$ at 5×10^{-4}, which decays to 1×10^{-4}. For the warp field $\mathbf{\Theta}_{\mathcal{W}}$, the learning rate starts at 1×10^{-3} and decays to 1×10^{-8}. We set the weighting term λ to 1×10^{2} for all the scans. Following Bian *et al.* [1], we multiply the output of the local warp field $\mathbf{\Theta}_{\mathcal{W}}$ by a constant small factor of 0.01. For the frequencies used for positional encoding $\gamma(.)$ defined in Sec. 3.1 of the main paper, we use $L = 10$ for 3D points and $L = 4$ for the viewing direction. The coarse-to-fine scheduler for BARF is linearly adjusted from iteration $20K$ to $100K$.

A.3 Additional details on Metrics

Pose alignment As mentioned in Sec. 4.3 of the main paper, we evaluate the pose accuracy by globally aligned the optimized poses to the groundtruth. This is because when optimizing camera poses and a neural radiance field, the optimized solutions of the scene geometry and the camera poses are up to a 3D similarity transformation. Following [9], we align both optimized and groundtruth poses globally using Umeyama algorithm [10, 14] [1]. Using the mathematical notations established in Sec. 3.1 of the main paper, we define the pose accuracy evaluation as below. The rotation error θ_t between the groundtruth poses of camera $\mathbf{R}^*$ and the aligned poses $\tilde{\mathbf{R}}$ for each camera t is computed as

$$\theta_t = \cos^{-1} \frac{\text{trace}(\mathbf{R}_t^* \tilde{\mathbf{R}}_t^T) - 1}{2}. \tag{1}$$

The translation error δ_t is computed as the Euclidean distance between the estimate $\tilde{\mathbf{t}}_t$ and the groundtruth position $\mathbf{t}_t^*$.

Novel view synthesis To assess the performance of view-synthesis, we report the mean Peak Signal-to-Noise Ratio (PSNR) [8], the Structural Similarity Index (SSIM) [12] and the Learned Perceptual Image Patch similarity (LPIPS) metric [13]. Following BARF [6], we use the AlexNet network version for calculating the LPIPS metric.

For the evaluation of depth, given that the optimized scene is subject to a 3D similarity, we align the scale of the predicted depth with the scale determined from the alignment procedure. We compute the mean absolute difference between the scaled predicted depth and ground-truth depth. We consider only those areas where valid ground-truth depth data is available in our evaluation.

B Additional Results

B.1 Single-INN versus Multi-INNs

In this section, we compare two different setups on the 2D planar neural image alignment problem, as detailed in Sec. 4.2 in the main paper. The first method, which we call "single-INN" involves using a global network coupled with a frame-specific code. Specifically, this global network is an INN consisting of three blocks, where each block contains two layers of 128-dim hidden units. We use a 16-dimensional code in this setup. The second method, termed "multi-INNs" employs a separate network for each individual patch. Each network is also an INN consisting of 3 blocks, with each block containing two layers of 32-dim hidden units. Tab. 3 indicates that a single global neural network (single-INN +

[1] We use Umeyama implementation from https://github.com/uzh-rpg/rpg_trajectory_evaluation/blob/master/src/rpg_trajectory_evaluation/align_trajectory.py

per-patch latent code) is adequate for converging to a good pose solutions. While the multi-INNs approach outperforms BARF by 25%, its overall accuracy is not as high as that of the single-INN approach. We postulate that the difference may be attributed to the benefits of gradient sharing in the shared neural network setup.

Table 3: Comparison of Single-INN versus Multi-INNs across 20 homography runs, with scale noise of 0.1 for homography and 0.2 for translation. The warp error is quantified in terms of corner error, and the patch reconstruction error in measured in PSNR. We provide mean and standard deviation (std. dev.) for the evaluation. We used 5-pixel threshold to determine success convergence.

	Corner error (px) ↓		Patch PSNR ↑		Success rate ↑
	Mean	Std. dev.	Mean	Std. dev.	(Upper bound:1.00)
BARF [6]	29.63	28.18	28.94	4.38	0.30
Multi-INNs	8.81	11.41	33.08	3.83	0.55
Single-INN + per-patch latent code	**4.70**	**6.47**	**34.71**	**2.37**	**0.75**

B.2 Ablations on INN (Sec. 4.4 of the main paper)

Tab. 4 ablates the key components of INN on the LLFF dataset using poses initialized at identity. When the weights of the INN are kept constant and made non-optimizable, the training fails completely. This outcome indicates that the INN weights play important role in accurately representing the transformation between the camera coordinate space and the world coordinate space. On the other hand, when the frame-specific code is fixed (with optimizable INN weights), there is a significant improvement in performance. This result emphasized the role of the frame-specific code in aiding the INN to distinguish different frames, rather than in representing the geometric transformation. As we observe that allowing the frame-specific code to be optimizable generally leads to better performance in general, we adopt this approach throughout our experiments.

Table 4: Ablations on the LLFF dataset (leaves), using poses initialized at *identity*. We use ✗ and ✓ symbols to represent the optimization status of weights during training: ✗ indicates that the parameters were non-optimizable, while ✓ indicates they were optimizable throughout training. The "-" notation is used to indicate unsuccessful training. The training process fails when the weights of the INN are not optimized.

		Pose accuracy		Novel view synthesis					
		Rotation	Translation	*Before* test-time			*After* test-time		
		(°)	(×100)	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
BARF	[6]	1.03	0.23	12.22	0.15	0.43	18.78	0.54	0.35
INN weights $\Theta_{\mathcal{W}}$	frame-specific code Φ_t								
✗	✗	-	-	-	-	-	-	-	-
✗	✓	-	-	-	-	-	-	-	-
✓	✗	0.35	0.20	16.30	0.39	0.35	18.76	0.54	0.34
✓	✓	0.29	0.17	16.00	0.36	0.35	19.01	0.56	0.33

B.3 Additional results on Blender

Table 5: Absolute pose accuracy evaluation for each scene of the Blender dataset [8], using initial noisy poses that corresponds to an average rotation and translation error of 15° and 26 respectively. The *upper* section presents rotation errors in degrees, while the *lower* section displays translation errors which are multiplied by 100. red box denotes the **best** result.

		Scenes								
Methods	Metric	chair	drums	ficus	hotdog	materials	mic	ship	lego	**mean**
L2G	rotation	0.11	0.07	0.21	0.27	2.41	2.40	0.20	0.08	0.72
INN(Ours)		0.12	0.06	0.15	0.24	0.81	0.08	1.26	0.09	0.35
L2G	translation	0.60	0.31	1.33	1.40	6.40	5.01	0.58	0.37	2.00
INN(Ours)		0.50	0.28	0.88	1.36	2.62	0.30	0.99	0.36	0.91

Table 6: Evaluation of novel view synthesis of the Blender dataset [7] **before** test-time optimization, using initial *identity* poses. red box denotes the **best** result.

		Scenes								
Methods	Metric	chair	drums	ficus	hotdog	materials	mic	ship	lego	**mean**
L2G	PSNR ↑	31.34	25.60	26.12	24.54	14.24	21.00	26.77	27.64	24.66
INN(Ours)		31.69	24.89	26.23	25.19	15.98	30.03	25.11	26.95	25.76
L2G	SSIM ↑	0.97	0.90	0.93	0.89	0.67	0.90	0.77	0.92	0.87
INN(Ours)		0.97	0.89	0.93	0.90	0.72	0.96	0.74	0.92	0.88
L2G	LPIPS ↓	0.03	0.09	0.05	0.05	0.15	0.06	0.17	0.05	0.08
INN(Ours)		0.03	0.11	0.06	0.06	0.13	0.05	0.21	0.05	0.09

B.4 Additional results on LLFF (Sec. 4.4 of the main paper)

We present a detailed per-scene comparison of pose accuracy in Tab. 7. We also include novel view synthesis results, both before and after test-time optimization in Tab. 8 and Tab. 9, respectively. We present the qualitative results in Fig. 2. Overall, the significant improvement in camera pose accuracy directly enhances the performance of novel view rendering across all evaluated metrics (PSNR/SSIM/LPIPS), as can be seen in Tab. 8. As discussed in Sec. 4.3 of the main paper, test-time optimization [3,4,6,9] is a pose refinement step. This step factors out the pose inaccuracies to minimize their impact on the quality of novel view synthesis. As a result of this optimization, the substantial improvement in camera pose accuracy become less pronounced in the novel view synthesis evaluation. The performance gap between BARF and our method, initially around 20% is significantly narrowed to approximately 2% after test-time optimization, as can be seen in Tab. 9.

Table 7: Absolute pose accuracy evaluation for each scene of the LLFF dataset [7], using initial *identity* poses. The *upper* section presents rotation errors in degrees, while the *lower* section displays translation errors which are multiplied by 100. red box denotes the **best** result.

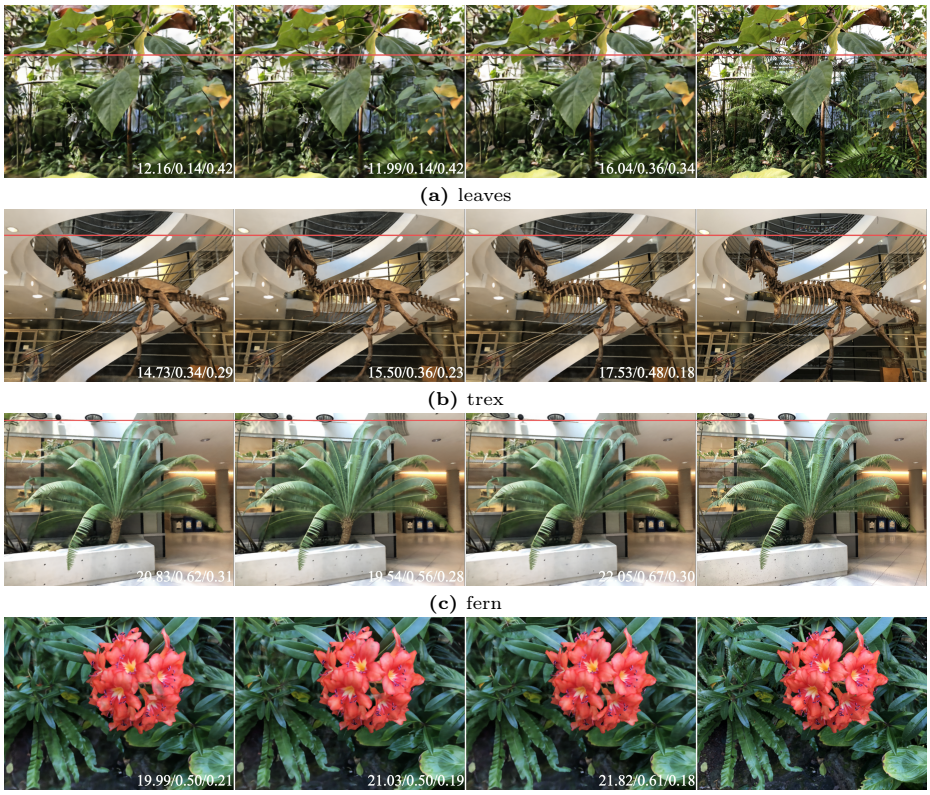
		Scenes								
Methods	Metric	fern	flower	fortress	horns	leaves	orchids	trex	room	**mean**
BARF	rotation	0.21	0.27	0.44	3.26	1.03	0.63	1.07	0.27	0.90
L2G		0.23	0.27	0.21	0.27	1.11	0.63	0.85	0.30	0.48
INN(Ours)		0.19	0.18	0.26	0.31	0.29	0.61	0.43	0.23	0.31
BARF	translation	0.20	0.23	0.36	1.42	0.23	0.40	0.17	0.22	0.40
L2G		0.19	0.24	0.23	0.22	0.37	0.40	0.58	0.22	0.30
INN(Ours)		0.19	0.22	0.26	0.18	0.17	0.35	0.36	0.18	0.24

Table 8: Evaluation of novel view synthesis of the LLFF dataset [7] **before** test-time optimization, using initial *identity* poses. red box denotes the **best** result.

		Scenes								
Methods	Metric	fern	flower	fortress	horns	leaves	orchids	trex	room	**mean**
BARF	PSNR ↑	20.80	19.60	22.15	11.16	12.22	12.92	14.75	22.41	17.00
L2G		19.54	19.35	25.20	18.13	12.05	13.05	15.30	21.27	17.99
INN(Ours)		22.04	21.24	25.11	17.19	16.00	12.89	17.19	22.84	19.31
BARF	SSIM ↑	0.61	0.49	0.42	0.30	0.15	0.16	0.33	0.80	0.41
L2G		0.56	0.47	0.70	0.50	0.14	0.17	0.36	0.78	0.46
INN(Ours)		0.66	0.58	0.63	0.46	0.36	0.16	0.46	0.81	0.52
BARF	LPIPS ↓	0.32	0.24	0.16	0.60	0.43	0.36	0.32	0.14	0.32
L2G		0.28	0.22	0.11	0.30	0.43	0.31	0.27	0.13	0.26
INN(Ours)		0.30	0.20	0.11	0.31	0.35	0.34	0.21	0.17	0.25

Table 9: Evaluation of novel view synthesis of the LLFF dataset [7] **after** test-time optimization, using initial *identity* poses. red box denotes the **best** result.

		Scenes								
Methods	Metric	fern	flower	fortress	horns	leaves	orchids	trex	room	**mean**
BARF	PSNR ↑	23.79	23.58	29.19	21.06	18.78	19.35	23.15	31.66	23.82
L2G		24.32	24.25	29.56	22.77	18.98	19.45	23.24	32.22	24.35
INN(Ours)		24.22	24.72	29.67	23.35	19.01	19.62	23.63	30.01	24.28
BARF	SSIM ↑	0.71	0.70	0.82	0.70	0.54	0.57	0.78	0.94	0.72
L2G		0.74	0.73	0.85	0.73	0.56	0.60	0.80	0.95	0.75
INN(Ours)		0.73	0.73	0.85	0.74	0.56	0.59	0.81	0.92	0.74
BARF	LPIPS ↓	0.31	0.21	0.13	0.30	0.35	0.29	0.20	0.10	0.24
L2G		0.26	0.18	0.10	0.28	0.33	0.24	0.17	0.08	0.21
INN(Ours)		0.29	0.18	0.10	0.27	0.33	0.26	0.16	0.14	0.22



(a) leaves

(b) trex

(c) fern

(d) flower

Fig. 2: Qualitative comparison of the results of various methods on a test image **before** pose-refinement. The comparison is arranged in columns: BARF (1st column), L2G (2nd column), our method (3nd column) and groundtruth (4th column). Each image includes an inset displaying its respective PSNR/SSIM/LPIPS values (zoom in for better view). We also add a solid red line to highlight any misalignment of each image compared to the groundtruth.

B.5 Additional Results on DTU with Colmap Initialization (Sec. 4.6 of the main paper)

We report our quantitative results using poses obtained from Colmap: the accuracy of pose estimation in Tab. 10, the performance of view synthesis in Tab. 11 and the depth evaluation in Tab. 12 as well as reconstruction accuracy in Tab. 13. Overall, our approach effectively refines these initial poses during the joint optimization, leading to a reduction in pose errors by approximately 30% for both rotation and translation. Additionally, attaining more accurate poses also contributes to improvement in novel view rendering quality (Tab. 11) and depth evaluation (Tab. 12) as well as reconstruction accuracy (Tab. 13).

Table 10: Absolute pose accuracy evaluation of the DTU dataset [7] **before** test-time optimization, using **Colmap** initialization that corresponds to an average rotation and translation error of 0.5° and 0.9 respectively. orange row denotes the result with Colmap initialization.

		Scan IDs														
Methods	Metric	24	37	40	55	63	65	69	83	97	105	106	110	114	118	**mean**
COLMAP (Initial)		0.31	0.54	0.26	0.27	0.96	0.32	0.34	0.46	0.60	0.36	0.28	0.45	0.31	0.24	0.41
BARF		**0.16**	0.75	0.51	0.99	**0.30**	0.78	**0.17**	0.45	0.53	0.28	0.38	**0.20**	0.26	0.28	0.43
L2G	rotation	0.20	0.48	**0.19**	0.43	0.45	0.61	0.27	0.29	0.49	0.23	1.66	0.25	0.33	0.35	0.44
INN(Ours)		0.20	**0.43**	**0.19**	**0.31**	0.45	**0.30**	0.27	**0.19**	**0.37**	**0.15**	**0.18**	0.25	**0.24**	**0.19**	**0.27**
COLMAP (Initial)		0.80	1.20	0.70	0.40	1.70	0.50	0.60	1.00	2.00	0.80	0.70	1.20	0.50	0.60	0.91
BARF	translation	0.50	2.37	1.88	3.70	**0.69**	2.82	**0.49**	1.80	1.81	1.07	1.37	0.42	0.94	0.87	1.48
L2G	(×100)	0.60	1.70	**0.62**	1.17	1.10	1.93	0.85	0.90	1.69	0.77	7.92	**0.31**	1.04	1.00	1.54
INN(Ours)		**0.42**	**0.93**	**0.62**	**0.70**	0.87	**0.84**	0.63	**0.57**	**1.15**	**0.52**	**0.42**	0.56	**0.62**	**0.59**	**0.67**

Table 11: Evaluation of novel view synthesis of the DTU dataset [7] **before** test-time optimization, using **Colmap** initialization.

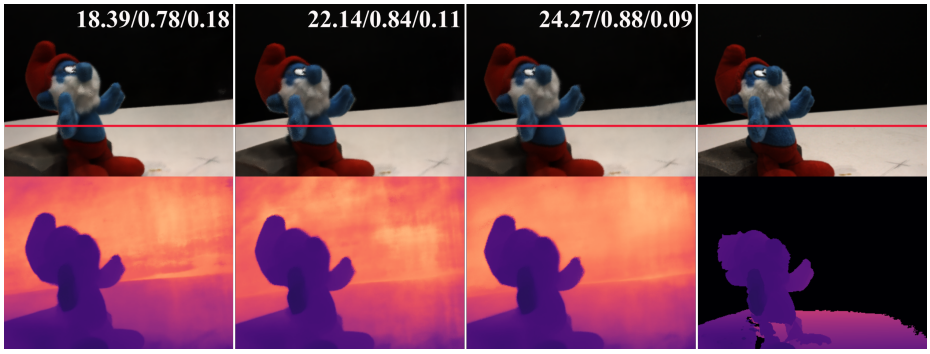
		Scan IDs														
Methods	Metric	24	37	40	55	63	65	69	83	97	105	106	110	114	118	**mean**
BARF		**23.37**	16.07	23.60	19.43	**24.54**	17.97	**26.29**	20.95	19.65	23.28	23.44	28.27	24.14	26.60	22.69
L2G	PSNR ↑	23.29	**18.99**	**24.58**	23.19	22.43	19.47	23.58	24.14	19.22	25.04	18.92	29.46	24.52	26.13	23.07
INN(Ours)		21.63	18.90	24.37	**25.15**	22.23	**22.52**	23.46	**25.00**	**20.19**	**27.87**	**26.50**	**29.81**	**25.87**	**31.57**	**24.65**
BARF		**0.79**	0.66	0.66	0.60	**0.91**	0.72	**0.82**	0.79	**0.67**	0.80	0.76	0.91	0.79	0.82	0.76
L2G	SSIM ↑	0.78	**0.71**	**0.77**	0.70	0.87	0.75	0.74	0.85	0.64	0.83	0.65	**0.92**	0.78	0.81	0.77
INN(Ours)		0.73	0.70	**0.77**	**0.75**	0.86	**0.81**	0.72	**0.87**	**0.67**	**0.89**	**0.83**	**0.92**	**0.82**	**0.90**	**0.80**
BARF		**0.14**	0.18	0.25	0.31	**0.08**	0.26	**0.19**	0.16	0.24	0.13	**0.19**	**0.12**	**0.19**	**0.17**	**0.19**
L2G	LPIPS ↓	0.15	**0.16**	**0.21**	**0.26**	0.09	0.25	0.22	**0.14**	**0.23**	**0.12**	0.33	**0.12**	0.20	0.18	**0.19**
INN(Ours)		0.17	0.18	0.24	0.28	0.10	**0.23**	0.25	**0.14**	0.24	**0.12**	**0.19**	0.13	**0.19**	0.18	**0.19**

Table 12: Depth evaluation (DE) in absolute error of the DTU dataset [5] **before** test-time optimization, using **Colmap** initialization.

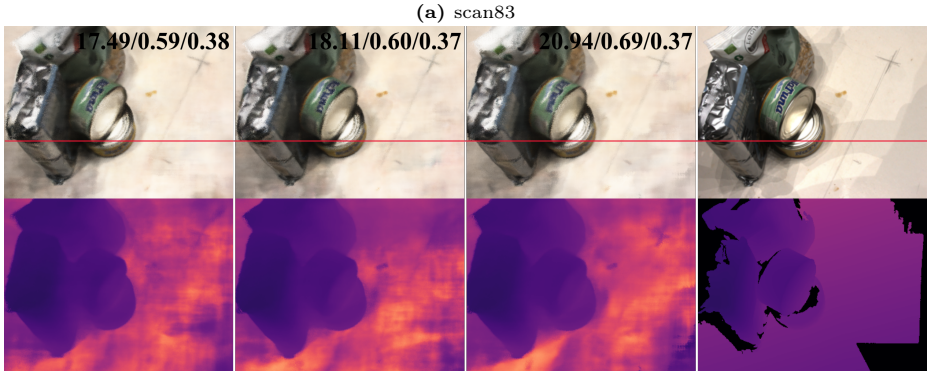
		Scan IDs														
Methods	Metric	24	37	40	55	63	65	69	83	97	105	106	110	114	118	**mean**
BARF	DE ↓	**0.15**	0.33	0.30	0.33	**0.24**	0.23	**0.12**	0.31	**0.15**	0.23	0.11	0.12	0.06	0.12	0.20
L2G		0.17	**0.26**	0.16	0.18	0.28	0.21	0.17	0.25	0.16	0.22	0.37	0.10	0.07	0.15	0.20
INN(Ours)		0.16	0.28	**0.15**	**0.13**	0.28	**0.10**	0.17	**0.20**	0.16	**0.13**	**0.07**	**0.07**	**0.04**	**0.11**	**0.15**

Table 13: Reconstruction accuracy evaluation of the DTU dataset [5] measured in Chamfer distance **before** test-time optimization, using **Colmap** initialization.

		Scan IDs														
Methods	Metric	24	37	40	55	63	65	69	83	97	105	106	110	114	118	**mean**
BARF	↓	**1.86**	7.14	4.09	7.06	**1.82**	5.34	**1.12**	5.15	5.48	3.65	2.32	1.90	2.54	1.08	3.61
L2G		2.29	4.21	1.08	2.10	2.66	4.17	1.32	3.24	4.00	2.59	8.03	1.48	2.93	1.28	2.96
INN(Ours)		1.97	**3.21**	**0.73**	**0.72**	2.27	**1.76**	1.64	**1.94**	**2.13**	**0.94**	**1.04**	**1.04**	**1.37**	**0.83**	**1.54**

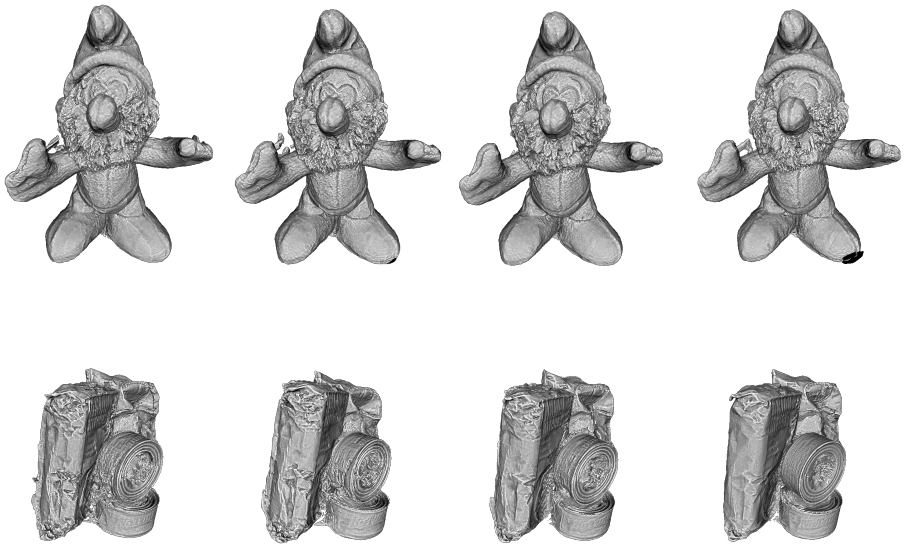


(a) scan83



(b) scan97

Fig. 3: Qualitative comparison of the rendered rgb along with its depth of various methods on test images using Colmap initialization. The comparison is arranged in columns: BARF (1st column), L2G (2nd column), our method (3nd column) and groundtruth (4th column). Each image includes an inset displaying its respective PSNR/SSIM/LPIPS values. We also add a solid red line to highlight any misalignment of each image compared to the groundtruth.



(a) BARF [6] **(b)** L2G [3] **(c)** Ours **(d)** Groundtruth

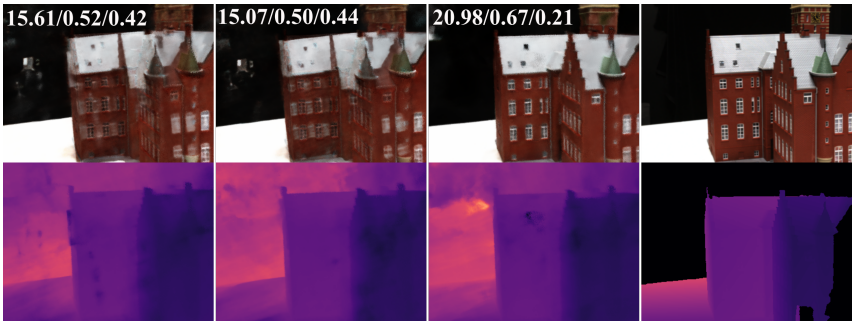
Fig. 4: We showcase the qualitative results for *scan83* and *scan97* in Tab. 13. All meshes are generated using a neural surface reconstruction algorithm called Voxurf using the poses estimated by different approaches.

B.6 Additional Results on DTU with Noisy Initialization (Sec. 4.6 of the main paper)

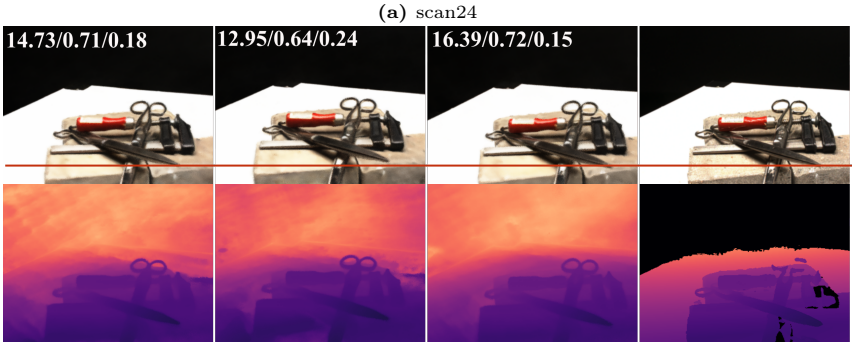
In the main paper (Sec. 4.6), we present the evaluation covering the pose accuracy, depth as well as reconstruction accuracy using initial noisy camera poses. For completeness, we additionally include here an evaluation of novel view synthesis (Tab. 14) as well as qualitative comparisons (Fig. 5 and Fig. 6).

Table 14: Evaluation of novel view synthesis of the DTU dataset [7] **before** test-time optimization, using initial *noisy* poses that corresponds to an average rotation and translation error of 15° and 70 respectively.

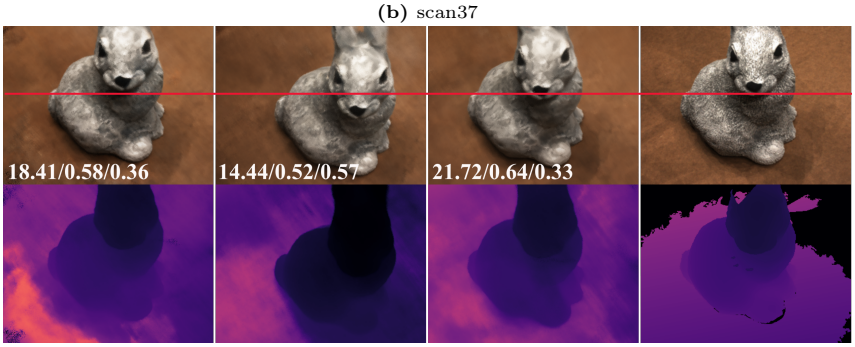
		Scan IDs | **Initial rotation err: 15°, translation err (×100) : 70**														
Methods	Metric	24	37	40	55	63	65	69	83	97	105	106	110	114	118	**mean**
BARF		15.73	14.50	19.63	16.10	21.18	17.94	16.52	**20.93**	14.93	12.24	16.40	**18.95**	**18.59**	22.56	17.59
L2G	PSNR ↑	16.71	12.41	**19.80**	13.81	17.57	16.94	**17.73**	20.23	18.36	15.64	16.11	17.13	14.00	17.52	16.71
INN(Ours)		**21.54**	**16.95**	18.65	**18.59**	**22.35**	**19.25**	16.87	20.11	**19.85**	**22.36**	**17.22**	17.58	17.87	**29.90**	**19.94**
BARF		0.57	0.62	**0.65**	0.54	0.86	0.72	0.47	**0.78**	0.55	0.51	0.54	**0.73**	**0.63**	0.70	0.63
L2G	SSIM ↑	0.58	0.56	**0.65**	0.44	0.80	0.70	**0.51**	0.77	0.64	0.60	0.55	0.65	0.51	0.49	0.60
INN(Ours)		**0.72**	**0.66**	0.63	**0.58**	**0.87**	**0.73**	0.48	0.77	**0.67**	**0.78**	**0.62**	0.68	0.60	**0.88**	**0.69**
BARF		0.27	0.26	**0.25**	0.43	**0.10**	**0.26**	0.50	**0.18**	0.36	0.50	0.56	**0.24**	**0.25**	0.27	0.32
L2G	LPIPS ↓	0.24	0.34	**0.25**	0.66	0.15	0.28	**0.35**	**0.18**	**0.25**	0.29	0.59	0.38	0.40	0.62	0.36
INN(Ours)		**0.17**	**0.20**	0.30	**0.38**	**0.10**	0.27	0.47	0.19	**0.25**	**0.14**	**0.42**	0.33	0.28	**0.17**	**0.26**



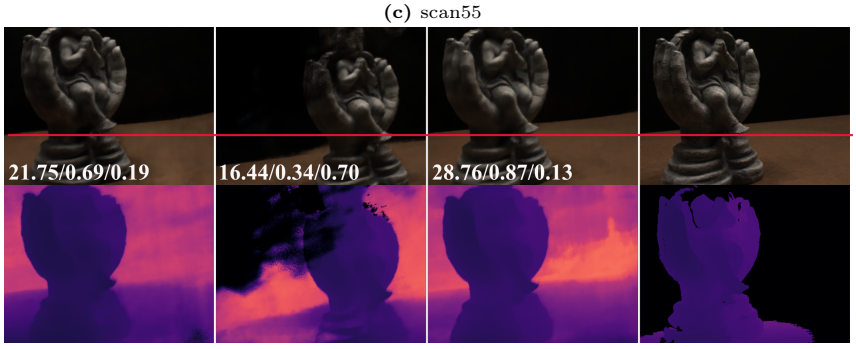
(a) scan24



(b) scan37

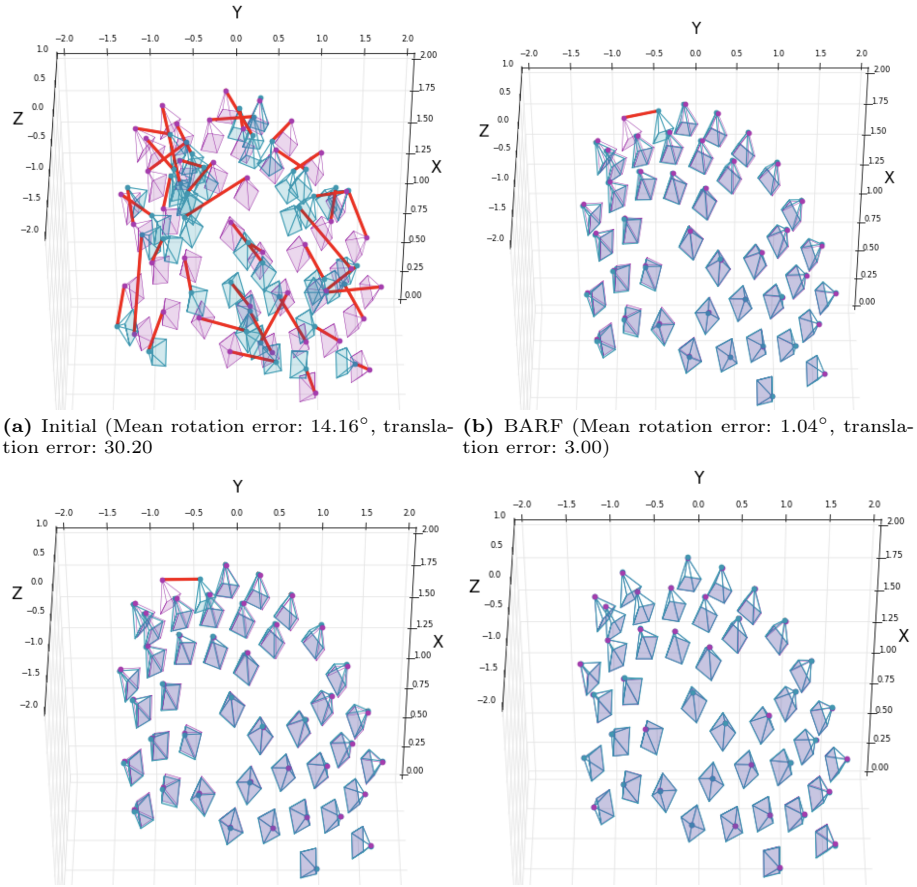


(c) scan55



(d) scan118

Fig. 5: Qualitative comparison of the rendered rgb along with its depth using noisy pose initialization. The comparison is arranged in columns: BARF (1st column), L2G (2nd column), our method (3nd column) and groundtruth (4th column). Each image includes an inset displaying its respective PSNR/SSIM/LPIPS values. We also add a solid red line to highlight any misalignments when compared to the groundtruth.



(a) Initial (Mean rotation error: 14.16°, translation error: 30.20

(b) BARF (Mean rotation error: 1.04°, translation error: 3.00)

(c) L2G (Mean rotation error: 0.90°, translation error: 2.40)

(d) Our method (Mean rotation error: 0.21°, translation error: 0.50)

Fig. 6: Visual comparison of the initial noisy poses (Fig. 6a) and estimated post-aligned camera poses using different approaches (Fig. 6b, Fig. 6c, Fig. 6d), for the DTU scene 'scan24'. In this comparison, the color blue denotes the perturbed/optimized camera pose, magenta represents ground truth camera poses.

References

1. Bian, J.W., Bian, W., Prisacariu, V.A., Torr, P.: Porf: Pose residual field for accurate neural surface reconstruction. arXiv preprint arXiv:2310.07449 (2023) 3
2. Cai, H., Feng, W., Feng, X., Wang, Y., Zhang, J.: Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. Advances in Neural Information Processing Systems **35**, 967–981 (2022) 1
3. Chen, Y., Chen, X., Wang, X., Zhang, Q., Guo, Y., Shan, Y., Wang, F.: Local-to-global registration for bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8264–8273 (2023) 3, 8, 13
4. Chng, S.F., Ramasinghe, S., Sherrah, J., Lucey, S.: Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In: European Conference on Computer Vision. pp. 264–280. Springer (2022) 8
5. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 406–413. IEEE (2014) 12
6. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5741–5751 (2021) 1, 3, 4, 5, 6, 8, 13
7. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) **38**(4), 1–14 (2019) 7, 8, 9, 11, 14
8. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 1, 4, 7
9. Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4190–4200 (2023) 2, 4, 8
10. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis & Machine Intelligence **13**(04), 376–380 (1991) 4
11. Wang, Q., Chang, Y.Y., Cai, R., Li, Z., Hariharan, B., Holynski, A., Snavely, N.: Tracking everything everywhere all at once. arXiv preprint arXiv:2306.05422 (2023) 1
12. WangZhou, B., Sheikh, H., et al.: Image qualityassessment: From errorvisibility-tostructural similarity. IEEE Transon ImageProcessing **13**(4), 600 (2004) 4
13. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) 4
14. Zhang, Z., Scaramuzza, D.: A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7244–7251. IEEE (2018) 4