SPARF: Neural Radiance Fields from Sparse and Noisy Poses
-
Supplementary Material

Prune Truong[1,2*] Marie-Julie Rakotosaona[2] Fabian Manhardt[2] Federico Tombari[2,3]
[1]ETH Zurich [2]Google [3]Technical University of Munich
prune.truong@vision.ee.ethz.ch {mrakotosaona, fabianmanhardt, tombari}@google.com

In this supplementary material, we provide additional details about our approach, experiment settings, and results. In Sec. A, we first give implementation details, both in terms of network architecture and training hyper-parameters. We then follow by extensively detailing the evaluation datasets and setup in Sec. B.

In Sec. C, we provide additional analysis on the proposed SPARF. In particular, we analyze the robustness of our joint pose-NeRF training approach to the camera pose initialization. We also present additional ablative experiments and give insights into failure cases. Importantly, we also look at the impact of using different correspondence predictors and the influence of the quality of the predicted matches.

In Sec. D, we present more detailed quantitative and qualitative results for our joint pose-NeRF refinement approach SPARF. Notably, we start from different camera pose initialization schemes than in the main paper and train with different numbers of input views. For completeness, we also provide comparisons of our approach SPARF to BARF with noisy input poses, but when *all* training views are available, *i.e.* in the many-view regime.

Finally, we provide additional quantitative results when considering fixed ground-truth poses in Sec. E. In particular, we experiment with more input views, *i.e.* 6 and 9 images instead of 3.

A. Training and Implementation Details

In this section, we first describe the architecture of the proposed SPARF. We additionally share all training details and hyper-parameters. For completeness, we also give details about the architectures and/or experimental setups used when training or evaluating baseline works.

A.1. NeRF architecture

We adopt the network architecture of the original NeRF [12] and its hierarchical sampling strategy with some modifications. The coarse and fine MLPs both have 128 hidden units in each layer. The numbers of sampled points of both coarse sampling and importance sampling are set to 128, and we use the softplus activation on the volume density output σ for improved stability.

Moreover, we found that for joint pose-NeRF optimization on LLFF, the same results are achieved with or without hierarchical sampling, *i.e.* with a single coarse or a coarse and fine MLPs. For these experiments, we therefore only use a single MLP, since it decreases the training time.

Depth parametrization: On the DTU and Replica datasets, we sample the 3D points along the ray linearly in metric space, between the pre-defined near and far plane $[t_n, t_f]$. On LLFF however, we follow [11] and sample points along each ray linearly in the inverse depth (disparity) space, where the lower and upper bounds are $1/t_n$ = 1 and $1/t_f$ = 0.05 respectively.

A.2. Correspondence prediction

To predict the matches relating the input image pairs, we use a recent state-of-the-art dense correspondence regression network, in particular PDC-Net [19]. It predicts for each pixel the conditional probability density of the flow vector given the input image pair. In practice, this translates to predicting the mean flow vector for each pixel, which corresponds to the match, and a confidence value. As the confidence value, we use the P_R operator [19], which represents the probability that the predicted flow vector is within a certain radius of the true match. For more details, we refer the reader to the PDC-Net publication [19]. We show examples of dense matches estimated by PDC-Net in Fig. 1.

Matches selection: We only apply the multi-view correspondence loss (Sec. 4.1 of m.p.) on correspondences which are predicted confidently, *i.e.* for which P_R is above a certain threshold $P_R > \gamma$. In practice, we choose $\gamma = 0.95$. We optionally also further filter the correspondences by keeping only the ones that are mutually consistent, *i.e.* for which the cyclic consistency is below 1.5 pixels.

[*] This work was conducted during an internship at Google.

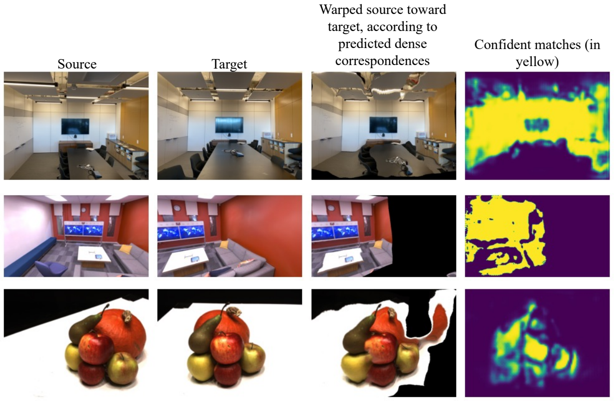

Source
Target
Warped source toward target, according to predicted dense correspondences
Confident matches (in yellow)

Figure 1. Dense matches and associated confidence predicted by PDC-Net [19] on pair examples of the LLFF, DTU, and Replica datasets. PDC-Net predicts the dense correspondences relating the target to the source. In the 3rd column, we show the source (1^{st} column) warped towards the target (2^{nd} column), according to those predicted correspondences. The warped source (3^{rd} column) should resemble the target (2^{nd} column). The correspondences deemed reliable by PDC-Net are highlighted in yellow in the last column.

A.3. Training details

Here, we describe the training hyper-parameters used in our experiments.

Staged training: As explained in Sec. 4.3 of the main paper, our joint pose-NeRF training is split into two stages. In the first one, the pose estimates are jointly trained with the coarse MLP, while in the second one, the pose estimates are frozen and both coarse and fine MLPs are trained. The first training stage accounts for 30% of the total training iterations.

We compute the matches between all-to-all views at the beginning of the training. At each iteration, the following procedure takes place. We sample x random pixels from all the training images, on which we apply the photometric loss (eq. 7 of m.p.). We also sample an image pair and apply the multi-view correspondence loss (Sec. 4.1 of m.p.) on a random subset of 1024 matches. For the depth consistency loss (Sec. 4.1 of m.p.), we sample a training view I_i associated with camera $\hat{P}_i$, find the closest other training view (according to current pose estimates), and compute an ”unknown” camera pose P_{un} as an interpolation of the two. We then randomly sample 1024 pixels in the training view I_i, for which we compute the depth consistency loss.

Coarse-to-fine positional encoding: For all datasets, we use the following scheme for the coarse-to-fine positional encoding of [11] (Sec. 4.3 of m.p.). When jointly refining the poses and training the NeRF, we linearly adjust the frequency width of the positional encoding from 40% to 70% of the training iterations. This means that for 40% of the training, there are no positional encodings applied to the 3D points and the ray directions. This mostly corresponds to when the camera poses are optimized.

When the input poses are fixed, we instead adjust the positional encoding from 10% to 50% of the training iterations. This is because the goal of the coarse-to-fine positional encoding is in that case to prevent overfitting at the early stages of training.

Training schedule with 3 input views: When the poses are fixed, we train for 50K iterations on DTU and Replica, and for 70K iterations on LLFF. For the joint pose-NeRF refinement, we instead train for 100K iterations on all datasets.

Training for longer (*i.e.* 100K iterations) with fixed ground-truth poses leads to similar or worse results than 50 or 70K iterations since the network starts to heavily overfit to the provided few (3) training images.

Training schedule with 6 input views: When the poses are fixed, we train for 100K iterations on DTU and Replica, and for 140K iterations on LLFF. For the joint pose-NeRF refinement, we instead train for 150K iterations on DTU and Replica, and 170K on LLFF.

Training schedule with 9 input views: When the poses are fixed, we train for 150K iterations on DTU and Replica, and for 200K on LLFF. For the joint pose-NeRF refinement, we instead train for 200K iterations on DTU and Replica, and 220K on LLFF.

Depth range: Each dataset provides a depth range $[t_n, t_f]$ within which the discrete depth values are sampled. When the initial poses are noisy, however, the provided range might not be sufficient to cover the scene. This is for example the case when we add 15% of noise to the ground-truth poses. For the joint pose-NeRF training, we therefore use a modified depth range $[(1-\epsilon)t_n, (1+\epsilon)t_f$, where $\epsilon = 0.3$.

Loss weighting: Our final loss formulation is provided in Sec. 4.3 of m.p.. We set the weights λ_{c} and λ_{d} associated with respectively the multi-view correspondence loss (Sec. 4.1 of m.p.) and the depth consistency loss (Sec. 4.2 of m.p.) to $\lambda_{\mathrm{c}} = 10^{-3}$ and $\lambda_{\mathrm{d}} = 10^{-3}$.

The intuition behind the weight λ_{c} is that the multi-view correspondence loss should have a magnitude in the same range as the photometric loss (eq. 7 of m.p.) since it is the main driving force of the pose optimization at the early stages of training. The correspondence prediction is nevertheless prone to errors. After the poses have converged, it can lead to errors in the learned geometry. In particular, if the weight of the multi-view correspondence loss is too high, the NeRF model can learn a wrong geometry, which is consistent with the wrong correspondences, even when it violates the photometric loss. To account for that, we gradually halve the weights λ_{c} every 10K iterations, after the poses are frozen. This enables the photometric signal to gradually gain more and more importance, thus correcting possible errors in the learned geometry. When the poses are fixed to ground truth, we also halve the weights λ_c every 10K iterations, starting from the beginning of the training.

As for the weight λ_d, the idea is that the depth consistency loss should account for less than the multi-view correspondence loss. The reason is that the latter ensures the model learns an *accurate* geometry while the former makes sure it is *consistent* from any viewing directions.

Only on DTU with fixed ground-truth poses, we find it beneficial to set $\lambda_c = 10^{-4}$ and $\lambda_d = 10^{-3}$ instead. We believe that a larger weight can have a negative impact as it amplifies possible errors in the correspondence predictions, which are reverberated on the learned scene geometry.

Pose parametrization: As in BARF [11], we optimize the world-to-camera transformation matrices. For the camera position, we simply adopt a 3D embedding vector in Euclidean space, denoted as $\mathbf{t} \in \mathbb{R}^3$, which we can directly update throughout the optimization. However, directly learning the rotation offset for each element of a rotation matrix would break the orthogonality of the rotation matrix.

The widely-used representations such as quaternions and Euler angles are discontinuous. Following [10], we adopt the 6-vector representation [28]. In particular, we use and optimize a continuous embedding vector $\mathbf{r} \in \mathbb{R}^6$ to represent 3D rotations, which is more suitable for learning. Concretely, given a rotation matrix $R = [\mathbf{a_1}\ \mathbf{a_2}\ \mathbf{a_3}] \in \mathbb{R}^{3\times3}$, we compute the rotation vector $\mathbf{r}$ by dropping the last column of the rotation matrix.

From the 6D pose embedding vector $\mathbf{r}$, we can then recover the original rotation matrix R using a Gram-Schmidt-like process, in which the last column is computed by a generalization of the cross product to three dimension [28]. It is formulated as a function f, which takes as input $\mathbf{r} = [\mathbf{a}_1^T, \mathbf{a}_2^T]$ and enables to recover the full rotation matrix, as follows,

$$R = f\left(\begin{bmatrix} | \\ \mathbf{r} \\ | \end{bmatrix}\right) = \begin{bmatrix} | & | & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \\ | & | & | \end{bmatrix}, \qquad (1)$$

where $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3 \in \mathbb{R}^3$ are $\mathbf{b}_1 = N(\mathbf{a}_1)$, $\mathbf{b}_2 = N(\mathbf{a}_2 - (\mathbf{b}_1 \cdot \mathbf{a}_2)\mathbf{b}_1)$, and $\mathbf{b}_3 = \mathbf{b}_1 \times \mathbf{b}_2$, and $N(\cdot)$ denotes L2 norm. At every iteration, the estimates of the rotation and translation parameters $\hat{R}^{w2c}$ and $\hat{\mathbf{t}}^{w2c}$ are updated as,

$$\hat{R}^{w2c} = f(\hat{\mathbf{r}}_0^{w2c} + \Delta\mathbf{r}), \quad \hat{\mathbf{t}}^{w2c} = \mathbf{t}_0^{w2c} + \Delta\mathbf{t}\,.$$

Here, $\hat{\mathbf{r}}_0^{w2c}$ and $\mathbf{t}_0^{w2c}$ denote the initial (noisy) camera rotation and translation parameters.

Hyper-parameters used for DTU: We use the Adam optimizer to optimize the network weights and the camera poses. For the network, we use an initial learning rate of 5×10^{-4}, which is exponentially decreased to 1×10^{-4} throughout the training. For the camera poses, we instead use an initial learning rate of 1×10^{-3} decaying to 1×10^{-4}. We resize the images to 300 × 400 and randomly sample 1024 pixel rays at each optimization step for the photometric loss (eq. 7 of m.p.).

Hyper-parameters used for LLFF: We use the Adam optimizer with an initial learning rate of 1×10^{-3} exponentially decreased to 1×10^{-4} throughout the training, for the network. For the camera poses, we instead use an initial learning rate of 3×10^{-3} decaying to 1×10^{-5}. We resize the images to 378 × 504 and randomly sample 2048 pixel rays at each optimization step for the photometric loss (eq. 7 of m.p.).

For joint pose-NeRF optimization on LLFF, we found in beneficial to only add the multi-view correspondence loss and the depth consistency loss after 1K iterations of training. This means that for the first 1K iterations, only the photometric signal (eq. 7 of m.p.) is used. This is because for some scenes, applying the multi-view correspondence loss from the beginning can lead to the background being in front of the foreground. Applying only the photometric loss at the very beginning of the training avoids this artifact. Our additional losses can then drive the poses and the geometry correctly. Moreover, we found that for joint pose-NeRF optimization on LLFF, the same results are achieved with or without hierarchical sampling. For these experiments, we therefore only use a single MLP, since it decreases the training time.

Hyper-parameters used for Replica: We use the same training hyper-parameters as for DTU. That is, for the network we use the Adam optimizer with an initial learning rate of 5×10^{-4} which is exponentially decreased to 1×10^{-4} throughout the training. For the camera poses, we instead use an initial learning rate of 1×10^{-3} decaying to 1×10^{-4}. We resize the images to 360 × 600 and randomly sample 1024 pixel rays at each optimization step for the photometric loss (eq. 7 of m.p.).

COLMAP: We run COLMAP [16] using the default parameters, with some exceptions. As recommended in the official documentation to better handle few images with a wide baseline, we reduce the minimum triangulation angle. We also enable the triangulation of two-view tracks, which can in rare cases improve the stability of sparse image collections by providing additional constraints in the bundle adjustment. To increase the number of matches, we use the more discriminative DSP-SIFT features instead of plain SIFT and also estimate the affine feature shape. Finally, we enable guided feature matching. We experiment with different pixel projection thresholds for the PnP pose estimation (default is 12) but see little impact on the initial pose registration results.

Since COLMAP often fails in the sparse-view scenario, we replace the feature matching of the standard COLMAP with better and more recent matching approaches. As reference implementation, we use the HLOC toolbox [15], which we modify to fit our needs. We try to use SuperPoint [7] and SuperGlue [14], which has become the defacto state-of-the-art sparse matching approach. As an alternative, we also use PDC-Net matches [19], a state-of-the-art dense matching approach. Note that we also use the latter to establish the correspondences between the training images in our approach SPARF. When using SuperPoint and SuperGlue, we set all default parameters. For the SuperGlue model, we use the indoor weights since both Replica and DTU scenes are taken indoors. We also set the default settings for PDC-Net.

Additional runtime: The multi-view correspondence loss and depth consistency objective add a factor of around 1.5 to the optimization time, regardless of the number of views. To predict the matches, PDC-Net [19] runs at 10fps on 300 × 400 images.

A.4. Baselines

BARF: We use the published code base for experiments using BARF.

SCNeRF: We use the official code base to obtain the implementation of the ray distance re-projection loss (named projected ray distance in [10]), which we integrate into our code. In the original paper, the projected ray distance is scaled with a weight $\lambda = 10^{-4}$. We kept this weighting for the LLFF experiments. However, we found that increasing this weight to $\lambda = 10^{-1}$ leads to much improved results on the DTU and Replica datasets. The projected ray distance loss relies on extracted correspondences between the views. For fairness, we use PDC-Net [19] correspondences, *i.e.* the same matches that we rely on in our multi-view correspondence loss (Sec. 4.1 of m.p.).

PixelNeRF: For evaluation results, we run the provided pre-trained model on the official code base.

DS-NeRF: We use the official code base to obtain the implementation of the depth loss, which we integrate into our code base. For the results on DTU with fixed ground-truth poses, we report the results from the publication. Nevertheless, we were unable to reproduce them using the official code base, where the configuration files for DTU are not released. We suspect that the authors used a 'trick' in the NeRF architecture to prevent heavy overfitting, *e.g.* for example reducing the positional encoding frequency. The results provided in the original publication for LLFF are computed using a different train/test split. We therefore re-train on our train/test splits using the released configuration files.

B. More Details on Datasets and Metrics

In this section, we provide details about the evaluation datasets and metrics.

B.1. Datasets

LLFF: As image resolution, we resize the images to $1/8^{th}$ of their original size, resulting in images of size 378×504. As stated in the main paper (Sec. 5.1 of m.p.), we follow community standards [12] and use every 8th image as the test set. We sample the training views evenly from the remaining images.

DTU: Following previous works [6, 13], we adhere to the evaluation protocol from PixelNerf and use the following 15 scan IDs as the test set: 8, 21, 30, 31, 34, 38, 40, 41, 45, 55, 63, 82, 103, 110, 114. The following image IDs (starting with “0”): 25, 22, 28, 40, 44, 48, 0, 8, and 13 are used as input. For the 3 and 6 input scenarios, we use the first 3/6 image IDs, respectively. For evaluation, the remaining images are used with the exception of the following image IDs due to wrong exposure: 3, 4, 5, 6, 7, 16, 17, 18, 19, 20, 21, 36, 37, 38, 39. We use an image resolution of 300×400. Following [13], we additionally evaluate all methods with the object masks applied to the rendered images. The object masks are obtained from [13, 23]. This is because, in most applications, it is more important to render the object of interest with high quality, rather than the background. Applying the foreground mask to the rendered images thus avoids penalizing methods for incorrect background predictions, regardless of the quality of the rendered object of interest.

Replica: We use the following 7 scenes as the test set: room0, room1, room2, office0, office1, office2, and office3. Each scene features a video of an indoor room, with between 1500 to 3000 frames. To create a realistic sparse-view scenario, where only few wide-baseline images per scene are available, we sub-sample every k^{th} frame, from which we randomly select a triplet of consecutive training images. Because each scene has a different frame rate, we adapt the sampling rate k to each scene individually. It is chosen such as each sampled image has a minimum of 20% covisible regions with another selected view. The exact sampling parameters will be included in the released code. We use an image resolution of 340×600.

B.2. Metrics

Alignment: When refining the camera poses, we evaluate the quality of registration by globally pre-aligning the optimized poses to the ground truth ones. This is necessary because both the scene geometry and camera poses are variable up to a 3D similarity transformation. The standard procedure [11, 22] is to align the two sets of pose trajectories (optimized and ground-truth) globally with a Sim(3) transformation using Umeyama algorithm [20] in an ATE toolbox [27]. Nevertheless, we found this strategy to give very unstable and unreliable results when the trajectory contains very few views (*i.e.* less than 9), which is the scenario we are focusing in this paper.

As a result, we perform the alignment in a RANSAC-inspired process. We sample every possible pair of cameras in one set, and compute the Sim(3) transformation (scale/rotation/translation) relating it to the same camera pair in the other trajectory. This gives us a set of possible Sim(3) transformations relating the optimized to the ground-truth trajectories. We then keep as global Sim(3) transformation the one leading to the lowest average camera alignment error. This process is done for the alignment when less than nine input views are available. Otherwise, we use the standard Umeyama algorithm [20].

Pose registration: After the optimized poses are aligned with the ground-truth ones, we can compute pose registration metrics. In particular, we report the average rotation and translation errors. The rotation error $|R_{err}|$ is computed as the absolute value of the rotation angle needed to align ground-truth rotation matrix R with estimated rotation matrix $\hat{R}$, such as

$$R_{err} = cos^{-1}\frac{Tr(R^{-1}\hat{R}) - 1}{2} \ , \tag{2}$$

where operator Tr denotes the trace of a matrix. The translation error T_{err} is measured as the Euclidean distance $\left\|\hat{T} - T\right\|$ between the estimated $\hat{T}$ and the ground-truth position T. Note that on all datasets, the positions of the poses are not in metric space, such that the translation error has no units.

Novel-view rendering: To evaluate the quality of novel view synthesis while being minimally affected by camera misalignment, we transform the test views to the coordinate system of the optimized poses by applying the scale/rotation/translation from the alignment analysis. To evaluate view synthesis in that case, we follow previous works [11, 22, 24] and run an additional step of test-time photometric optimization on the trained models to factor out the pose error from the view synthesis quality. In essence, it is a more fine-grained gradient-driven camera pose alignment which minimises the photometric error on the synthesised image, while keeping the NeRF model fixed. This test-time photometric optimization is run in experiments where the poses are refined. For fairness, we also use it in experiments where we fix the initial noisy poses, *e.g.* obtained by COLMAP [16], to differentiate the novel-view rendering quality from the initial pose error.

To evaluate the view-synthesis performance, we report the mean Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [21], and the Learned Perceptual Image Patch similarity (LPIPs) metric [26], which estimates the distance between an image pair in a learned feature space.

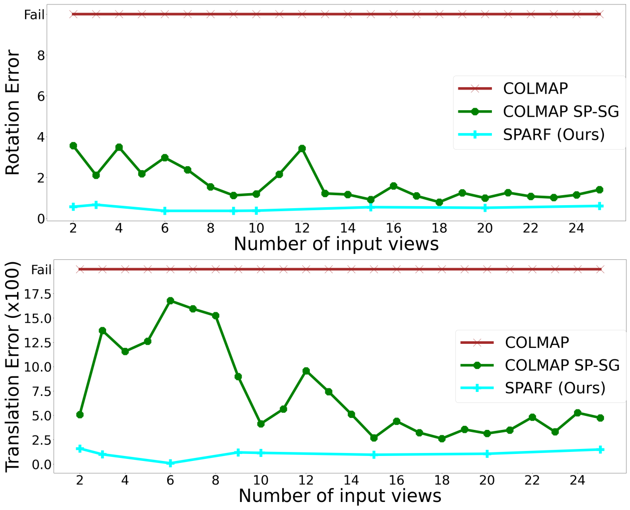
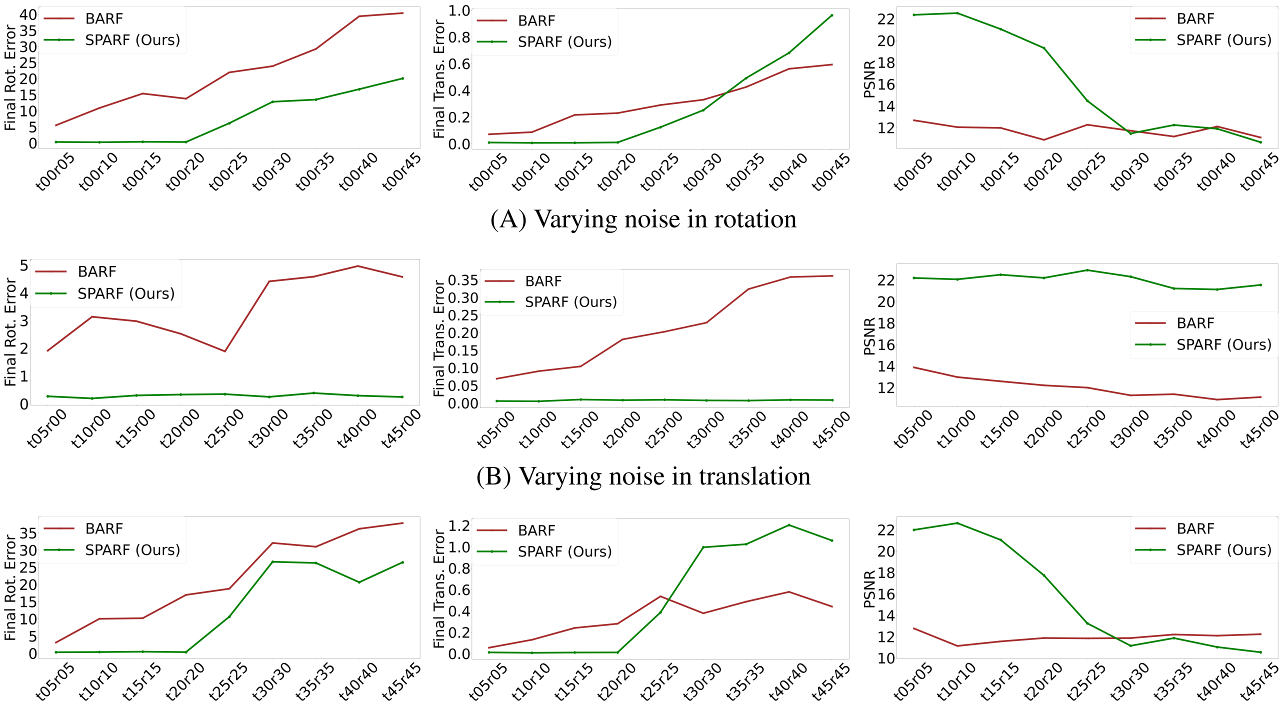


Figure 2. Rotation and translation errors versus the number of input views on a scene of the DTU dataset [9]. The standard COLMAP [16] fails to estimate initial poses for each number of input views, including relatively high numbers (> 20). Failing in that case means that COLMAP does not find a pose for at least one image of the set. COLMAP with better sparse matches (SuperPoint and SuperGlue [7, 14]) performs a lot better. Nevertheless, for very few images (< 9), the estimated poses are noisy, which can drastically impact the quality of the trained NeRF. Our approach SPARF can successfully refine those poses in the sparse-view regime, and consequently, train a better-performing NeRF. Also, note that the quality of our pose refinement approach SPARF stays constant when increasing the number of input images (> 9). It consistently outperforms COLMAP-SP-SG in that regime as well.

For the depth evaluation, we first multiply the predicted depth with the scale from the alignment (since the optimized scene is variable up to a 3D similarity), such that it is in the same range than the ground-truth depth. We then compute the absolute difference between the predicted and ground-truth depths, averaged over the valid ground-truth depth areas.

C. Additional Method Analysis

In this section, we present additional analyses of the proposed approach SPARF. We first look at the degradation faced by COLMAP [16] when reducing the number of input views. We also analyze the robustness of our approach SPARF to different pose initialization, and provide insights into failure cases. Additionally, we look at the impact of using different correspondence predictors and the influence of the quality of the predicted matches. Finally, we present additional ablation studies.



(A) Varying noise in rotation

(B) Varying noise in translation

(C) Varying noise in rotation and translation

Figure 3. Pose registration error and PSNR obtained by BARF and our SPARF for different levels of initial noise. This experiment is performed on one scene of the DTU dataset, considering 3 input views. Rotation errors are in degree and translation errors are multiplied by 100. Results of PSNR (↑) are computed by masking the background.

C.1. Performance of COLMAP when reducing the number of views

Here, we analyze the performance of COLMAP [16] for different numbers of input views. In Fig. 2, we plot the rotation and translation errors obtained by the standard COLMAP, COLMAP with SuperPoint-SuperGlue matches and our joint pose-NeRF refinement approach SPARF, versus the number of input views. Even for a relatively high number of input views (> 20), the standard COLMAP fails to estimate initial poses. This is because the images show significant viewpoint variations. Replacing the matches with those predicted by SuperPoint and SuperGlue [7, 14] (COLMAP SP-SG) leads to much better results. Nevertheless, for very few images (< 9), it is very challenging to estimate high-accuracy poses. COLMAP SP-SG predicts initial poses with a rotation error between 2 and 4°, and a translation error comprised between 5.0 and 17.5. Training a NeRF with such noisy initial poses results in a drastic drop in performance compared to training with perfect input poses. Our approach SPARF can successfully refine those initial poses while training the NeRF. As a result, the final optimized poses have much lower rotation and translation errors. It consequently leads to a better-performing NeRF model.

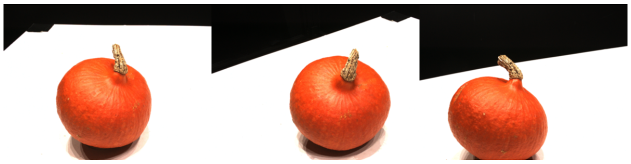


Figure 4. Failure case example of our approach SPARF. The object, *i.e.* the pumpkin, is almost fully symmetric with many homogeneous surfaces. The correspondence network fails to extract reliable correspondences relating the input views. As a result, our approach is unable to refine the noisy initial poses.

C.2. Robustness to pose initialization and failure cases

Robustness to pose initialization: We next investigate the robustness of our joint pose-NeRF refinement approach to different levels of initial noisy poses. For this experiment, our approach SPARF only uses our multi-view correspondence loss objective (Sec. 4.1 of m.p.), without our depth consistency loss (Sec. 4.2 of m.p.) nor our staged training (Sec. 4.3 of m.p.). We create the noisy initial poses by synthetically perturbing the ground-truth poses with different levels of additive Gaussian noise. We present results on a randomly sampled scene of DTU in Fig. 3. We investigate

	Over all scenes						Over only correctly registered scenes						
	Pose registration		Novel-view synthesis				Pose registration			Novel-view synthesis			
	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE↓	Nbr. corr. sc. (/15)	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE↓
BARF	10.3	51.5	10.7 (9.8)	0.43 (0.62)	0.59 (0.36)	1.9	2	2.56	9.23	16.6 (17.4)	0.66 (0.76)	0.28 (0.18)	0.29
SCNeRF [19]	3.44	16.4	12.0 (11.7)	0.45 (0.66)	0.52 (0.30)	0.85	10	1.06	4.42	12.1 (12.6)	0.51 (0.68)	0.47 (0.28)	0.80
SPARF* (PDC-Net)	**1.85**	**5.5**	**16.0 (17.8)**	**0.68 (0.81)**	**0.28 (0.14)**	**0.13**	**14**	**0.26**	**0.6**	16.8 (**19.1**)	0.69 (**0.81**)	0.25 (**0.12**)	**0.08**
SPARF* (SP-SG)	5.95	19.24	14.8 (16.1)	0.64 (0.79)	0.36 (0.18)	0.19	11	0.55	2.05	**17.0 (19.1)**	**0.70** (0.80)	**0.24** (0.13)	0.09

Table 1. Performance of our joint pose-NeRF training, when using different pre-trained correspondence networks. The results are computed on DTU [9] with initial noisy poses (3 views). We simulate noisy poses by adding 15% of random noise to the ground-truth poses. Here, SPARF* indicates that we only use the combination of the photometric loss with our multi-view correspondence objective (Sec. 4.1 of m.p.), without including our depth consistency objective (Sec. 4.2 of m.p.) nor our staged training (Sec. 4.3 of m.p.). Rotation errors are in degree and translation errors are multiplied by 100. Results in $(\cdot)$ are computed by masking the background. Nbr. corr. sc. designates the number of correctly registered scenes. We consider a scene to be correctly registered when the average rotation is below 10° and the average translation is below 10. Note that for SCNeRF [10], we use PDC-Net [19] correspondences.

	Rot. (°) ↓	Trans. (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BARF [11]	2.04	11.6	17.47	0.48	0.37
SCNeRF [10]	1.93	11.4	17.10	0.45	0.40
SPARF* (PDC-Net)	**0.53**	2.8	**19.50**	**0.61**	0.32
SPARF* (SP-SG)	**0.53**	**3.0**	19.48	0.60	**0.32**

Table 2. Performance of our joint pose-NeRF training, when using different pre-trained correspondence networks. As in Tab. 1 for DTU, the evaluation is here performed on the forward-facing dataset LLFF [17] (3 views) starting from initial identity poses. Here, SPARF* indicates that we only use the combination of the photometric loss with our multi-view correspondence objective (Sec. 4.1 of m.p.), without including our depth consistency objective (Sec. 4.2 of m.p.) nor our staged training (Sec. 4.3 of m.p.).

perturbing only the rotation matrix, only the translation vector, or both in respectively (A), (B), and (C). As a reference, we also include results of BARF [11]. Our approach SPARF can handle up to 20% of noisy rotations, which corresponds to about 20°. Interestingly, our SPARF is extremely robust to translation noise, successfully registering poses with up to 45% translation noise. When both rotation and translation noises are included, our method is robust to 20% of noise, the rotation being the limiting factor.

Failure cases: Our approach SPARF depends on the quality of the predicted correspondences. If only too few or inaccurate matches can be extracted between the input views, the joint pose-NeRF training will likely fail.

It is particularly difficult to predict reliable correspondences for (almost) symmetric objects or for scenes containing many homogeneous surfaces. Such a challenging example is presented in Fig. 4, which corresponds to 'scan30' of the DTU dataset. The depicted pumpkin is almost symmetric and has mostly uniform surfaces. On these images, the pre-trained correspondence network PDC-Net [19] does not predict any reliable matches. Note that the alternative matching approach SuperPoint-SuperGlue [7, 14] is also unable to extract correspondences in that case.

C.3. Impact of different correspondences

Our multi-view correspondence loss (eq. 8 of m.p.) (Sec. 4.1 of m.p.) relies on a pre-trained correspondence network to predict matches between the training views. As stated in the main paper, while we use PDC-Net [19], any hand-crafted or learned matching network could be used. We here compare using the dense correspondence regression network PDC-Net [19] with the state-of-the-art sparse matcher SuperGlue [14]. In combination with the SuperGlue matcher, we use the SuperPoint [7] detector and descriptor.

In Tab. 1, we present results on DTU, of our joint pose-NeRF refinement approach, trained using the multi-view correspondence objective (Sec. 4.1 of m.p.) with these two alternative matching methods. As a reference, we also include results of BARF [11] and SCNeRF [10]. Sparse matchers particularly struggle to detect repeatable keypoints and predict reliable matches on images with repetitive structures and homogeneous surfaces. Dense matching approaches are more robust to these conditions. As a result, SP-SG finds an insufficient number of matches on 4 scenes out of 15, compared to 1 scene out of 15 for dense correspondence network PDC-Net. When matches are unreliable or in insufficient number, our joint pose-NeRF training is likely to fail, since our multi-view correspondence loss (eq. 8 of m.p.) relies on the predicted correspondences. As a result, when considering all scenes, SPARF* with SP-SG obtains a worse pose registration and novel-view synthesis performance than SPARF* with PDC-Net. Note nevertheless that the novel-view synthesis results are still significantly better than that of BARF and SCNeRF. When taking the average only over the "correctly registered scenes" instead, SPARF* with PDC-Net or SP-SG matches leads to similar pose registration and novel-view synthesis quality.

In Tab. 2, we present the same comparison, on the LLFF dataset. Using PDC-Net or SP-SG matches results in a similar performance.

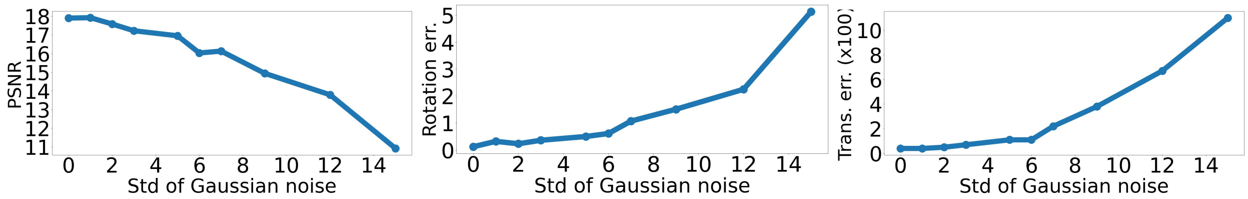


Figure 5. Evaluation of SPARF* on one scene of DTU, with different levels of Gaussian noise added to ground-truth image matches. Here, SPARF* indicates that we only use the combination of the photometric loss with our multi-view correspondence objective (Sec. 4.1 of m.p.), without including our depth consistency objective (Sec. 4.2 of m.p.) nor our staged training (Sec. 4.3 of m.p.).

Impact of noisy matches: As an additional experiment, we added different levels of Gaussian noise to ground-truth matches on DTU and trained our joint pose-NeRF refinement approach SPARF * using those matches. The goal is to analyze the robustness of SPARF * to noisy correspondences. We conducted this experiment on one scene of DTU, with 3 input views associated with noisy poses, and present the results in Fig. 5. As previously, we consider 15% of initial additive Gaussian noise. SPARF is robust to quite noisy matches (standard-deviation up to 6 pixels) but sees its performance drop with highly erroneous correspondences.

C.4. Additional ablation study

Ablation study for joint pose-NeRF refinement: In Tab. 2 of the main paper, we ablated key components of our approach, considering fixed ground-truth poses on the DTU dataset. Here, we ablate our approach when refining initial noisy poses along with training the NeRF model. As previously, we consider 15% of initial additive Gaussian noise. We present results in the top part of Tab. 3. From (I) to (II), adding our multi-view correspondence loss (eq. 8 of m.p.) leads to drastically better pose registration than training with only the photometric loss (eq. 7 of m.p.) (I). The rendering quality also radically improves. This is in part due to the better pose registration, which is necessary to obtain a decent rendering quality. It is also enabled by the fact that our multi-view correspondence loss not only drives the camera poses but also applies direct supervision on the rendered depth, enforcing it to be close to the surface. As such, it enables learning an accurate scene geometry. In (III), we introduce our staged training (Sec. 4.3 of m.p.), which is composed of two parts. In the first stage, we refine the poses while training the coarse network F_θ^c. In the second part, we freeze the pose estimates and train both the coarse and fine networks F_θ^c and F_θ^f. Comparing (II) to (III), we observe that introducing this second stage leads to better PSNR and SSIM metrics. This is because the fine network can learn a sharp geometry benefiting from the frozen, registered camera poses and the pre-trained coarse network. On the other hand, when jointly training the camera poses and both coarse and fine MLP (II), the learned scene often has a slightly blurry surface due to the exploration of the pose space. Finally, further including our depth consistency objective (Sec. 4.2 of m.p.) slightly improves the rendering performance, leading to the best results overall.

		Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
I	Photo. (eq. 7 of m.p.)	10.3	51.5	10.7 (9.8)	0.43 (0.62)	0.59 (0.36)	1.9
II	+ MVCorr (eq. 8 of m.p.)	1.85	5.5	16.0 (17.8)	0.68 (0.81)	0.28 (0.14)	0.13
III	+ Staged training	**1.81**	**5.0**	17.58 (18.62)	**0.71** (0.82)	**0.26** (**0.13**)	0.13
IV	+ DCons (eq. 9 of m.p)	**1.81**	**5.0**	**17.74** (**18.92**)	**0.71** (**0.83**)	**0.26** (**0.13**)	**0.12**
II	Fully joint pose-NeRF	1.85	5.5	16.0 (17.8)	0.68 (0.81)	0.28 (0.14)	0.13
III	Staged training (Sec. 4.3 of m.p.)	**1.81**	**5.0**	17.58 (18.62)	0.71 (0.82)	0.26 (0.13)	0.13
V	Restart NeRF	1.84	5.3	**17.80** (**19.07**)	**0.72** (**0.83**)	**0.25** (**0.12**)	**0.12**

Table 3. Ablation study on DTU [9] (3 views) with noisy initial poses. In the top part, from (I) to (IV), we progressively add (+) each component. In the bottom part, we compare multiple training schedules for the joint pose-NeRF training. The depth consistency loss (Sec. 4.2 of m.p.) is then not included. Rotation errors are in degree and translation errors are multiplied by 100. Results in (·) are computed by masking the background.

Comparison of different training schedules: In the bottom part of Tab. 3, we further compare different training schedules for joint pose-NeRF training. As previously explained, jointly training the poses with both the coarse and fine MLPs in (II) can lead to blurry surfaces. As demonstrated in (III), our staged training (Sec. 4.3 of m.p.) largely solves this problem, leading to better rendering quality. Nevertheless, it is worth noting that the best results are obtained with the NeRF restarting approach corresponding to (V). In (V), the NeRF is first jointly trained with the poses. Once the poses have converged, the optimized pose estimates are frozen and both coarse and fine MLPs are reinitialized. Both MLPs are then trained from scratch, considering fixed optimized poses. This approach can remove some of the artifacts learned during the pose optimization, that might still be present in our staged training (III). This restarting approach was also found to be the best alternative in [22].

Impact of visibility mask in depth consistency loss: In Sec. 4.2 of the main paper, we introduce our depth consistency loss. However, the proposed loss is only valid in pixels of the training views for which the projections in the virtual view are not occluded by the reconstructed scene, seen from the virtual view. We therefore use a visibility mask, following the same formulation as [5]. We ablate the impact of this visibility mask in the depth consistency loss formulation in Tab. 4. We observe that removing the visibility mask leads to a notable drop in performance in PSNR, probably because the NeRF model learns surfaces that are actually occluded, leading to artifacts in the geometry and therefore the renderings.

D. Additional Results with Initial Noisy Poses

In this section, we provide additional results considering initial noisy poses. In particular, we experiment with different initialization schemes. We also use different numbers of input views and present extensive qualitative results. Finally, we experiment with training considering all available training views (*i.e*. 25), instead of a subset.

D.1. Results on the DTU dataset

Here, we present additional results for our joint pose-NeRF refinement approach SPARF, evaluated on the DTU dataset [9]. In the main paper, we showed results when considering three input views and starting from initial noisy poses, created by synthetically perturbing ground-truth poses. Here, we first evaluate starting from an alternative initialization scheme, in particular initial poses obtained by COLMAP [16]. Moreover, we also evaluate for different numbers of input views, in particular 6 or 9. We also show multiple qualitative comparisons for the 3-view setting.

Initialization with COLMAP: On the DTU input images, COLMAP [16] mostly fails when reducing the number of input views to 3 (see Fig. 2). As a result, to obtain the initial camera pose estimates, we experiment with COLMAP run with matches predicted by SuperPoint and SuperGlue [14] (SP-SG) or PDC-Net [19]. Both COLMAP-SP-SG and COLMAP-PDCNet fail to obtain initial pose estimates on one out of the 15 scenes composing the test set ('scan30', see Fig. 4). We thus present results on the remaining 14 scenes in Tab. 5. In the middle part of the table (F), we fix the initial poses, which we consider as "pseudo-ground-truth", and train the NeRF model. In the bottom part (R), we instead compare multiple joint pose-NeRF refinement approaches. Finally, in the top part (G), we present the results of SPARF, trained considering fixed ground-truth poses for reference.

	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
MVCorr (eq. 8 of m.p.)	18.13 (20.81)	0.77 (**0.87**)	0.22 (**0.10**)	0.10
+ DCons (eq. 9 of m.p.)	**18.30** (**21.01**)	**0.78** (**0.87**)	**0.21** (**0.10**)	**0.08**

Table 4. Impact of the visibility mask for our depth consistency loss (Sec. 4.2 of the main paper). Results are computed on the DTU dataset (3 views), with fixed ground-truth poses. Results in (·) are computed by masking the background. All networks use the coarse-to-fine PE [11].

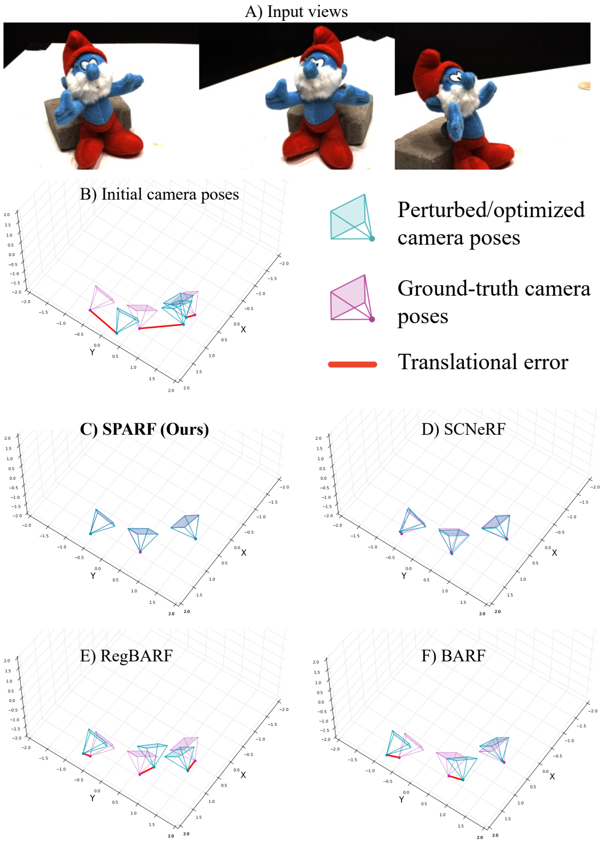


Figure 6. Initial and optimized poses on one scene of the DTU dataset, given 3 input views.

SP-SG sometimes struggles with homogeneous surfaces, where it is difficult to extract repeatable keypoints. It leads to an initial rotation and translation error of respectively 1.34° and 6.84. PDC-Net, which can heavily rely on smoothness properties when predicting dense matches, performs better on homogeneous regions. It results in slightly better initial poses, *i.e*. with an initial rotation and translation error of 0.75° and 3.87 respectively.

For both initialization schemes, the trend is the same. Considering the COLMAP poses as "pseudo-ground-truth" and training the NeRF with fixed poses (part F) leads to significantly worse results than when using ground-truth poses (top part, G), particularly in PSNR and SSIM. This is because the NeRF learns artifacts caused by the wrong positioning of the poses. Instead, using our approach to jointly refine the poses and train the NeRF (R) narrows the gap between fixed COLMAP poses (F) and the ideal case of fixed ground-truth poses (G). Note that the latter case of fixed ground-truth poses is unrealistic in practice. Notably, when refining the poses, SPARF obtains similar performance in

		Initial COLMAP SP-SG Rot: 1.34°, Trans (×100): 6.84						Initial COLMAP PDCNet Rot: 0.75°, Trans (×100): 3.87					
		Rot. (°) ↓	Trans (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓	Rot. (°) ↓	Trans (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
G	**SPARF** (Ours)	Fixed GT poses		18.56 (20.84)	0.77 (0.86)	0.22 (0.11)	0.08	Fixed GT poses		18.56 (20.84)	0.77 (0.86)	0.22 (0.11)	0.08
F	NeRF [12]	Fixed poses obtained from COLMAP (run w. SP-SG [14] matches)		8.95 (9.77)	0.30 (0.60)	0.72 (0.38)	1.25	Fixed poses obtained from COLMAP (run w. PDC-Net [19] matches)		8.88 (9.66)	0.31 (0.62)	0.73 (0.37)	1.28
	DS-NeRF [6]			11.89 (13.28)	0.46 (0.69)	0.49 (0.25)	0.38			11.61 (12.81)	0.46 (0.70)	0.51 (0.25)	0.60
	DS-NeRF w. CF PE [6, 11]			16.58 (17.58)	0.66 (0.77)	0.29 (0.17)	0.21			18.10 (19.30)	0.71 (0.80)	0.24 (0.13)	0.12
	SPARF (Ours)			17.34 (17.92)	0.68 (0.78)	0.26 (0.14)	0.15			18.42 (19.61)	0.72 (0.82)	0.22 (0.12)	0.12
R	BARF [11]	4.90	12.74	13.14 (13.01)	0.52 (0.69)	0.45 (0.25)	0.55	3.5	11.94	14.27 (14.59)	0.56 (0.70)	0.39 (0.23)	0.54
	RegBARF [11, 13]	4.3	11.0	14.65 (15.30)	0.6 (0.73)	0.38 (0.22)	0.25	3.71	9.81	15.22 (15.98)	0.60 (0.73)	0.36 (0.22)	0.25
	SCNeRF [10]	0.97	3.08	15.94 (16.73)	0.63 (0.75)	0.32 (0.19)	0.43	1.08	3.3	15.94 (16.42)	0.63 (0.75)	0.32 (0.18)	0.43
	DS-NeRF [6]	3.7	10.0	13.67 (14.30)	0.54 (0.72)	0.40 (0.22)	0.21	2.66	7.58	16.00 (16.87)	0.63 (0.77)	0.31 (0.17)	0.24
	SPARF (Ours)	0.35	0.9	18.39 (19.67)	0.73 (0.82)	0.22 (0.12)	0.09	0.3	0.7	18.52 (20.00)	0.73 (0.83)	0.21 (0.11)	0.08

Table 5. Evaluation on 14 scenes of the DTU dataset (3 views) with initial poses obtained by COLMAP using SP-SG [14] (left) or PDCNet [19] (right) matches. Note that both approaches fail to obtain the initial poses on one of the pre-defined 15 test scenes ('scan30'), which we therefore excluded from this evaluation. In the middle part (F), the initial poses are fixed and used as "pseudo-ground-truth". In the bottom part (R), the poses are refined along with training the NeRF. For comparison, in the top part (G), we use fixed ground-truth poses. All methods in the bottom part (R), which perform joint pose-NeRF training, use the coarse-to-fine PE approach [11] (Sec. 4.3 of m.p.). Results in (·) are computed by masking the background. The best and second-best results are in red and blue respectively.

	3 input views						**6 input views**						**9 input views**					
	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
BARF [11]	10.33	51.5	10.71 (9.76)	0.43 (0.62)	0.59 (0.36)	1.9	9.20	31.1	14.02 (14.22)	0.54 (0.69)	0.46 (0.27)	0.49	8.34	26.72	16.20 (16.38)	0.60 (0.73)	0.38 (0.22)	0.35
RegBARF [11, 13]	11.2	52.8	10.38 (9.20)	0.45 (0.62)	0.61 (0.38)	2.33	9.19	26.63	14.59 (14.58)	0.57 (0.70)	0.44 (0.27)	0.32	5.28	18.51	18.98 (19.08)	0.67 (0.77)	0.29 (0.18)	0.23
DistBARF [2, 11]	11.69	55.7	9.50 (9.15)	0.34 (0.76)	0.67 (0.36)	1.90	8.96	28.85	14.31 (14.60)	0.55 (0.70)	0.43 (0.26)	0.53	7.00	26.42	16.18 (16.27)	0.58 (0.71)	0.37 (0.22)	0.29
SCNeRF [10]	3.44	16.4	12.04 (11.71)	0.45 (0.66)	0.52 (0.30)	0.85	4.10	12.80	17.76 (18.16)	0.70 (0.80)	0.31 (0.18)	0.28	4.76	16.25	18.19 (18.01)	0.69 (0.81)	0.31 (0.17)	0.31
SPARF (Ours)	**1.81**	**5.0**	**17.74 (18.92)**	**0.71 (0.83)**	**0.26 (0.13)**	**0.12**	**1.31**	**2.7**	**21.39 (22.01)**	**0.81 (0.88)**	**0.18 (0.10)**	**0.09**	**1.15**	**2.55**	**24.69 (25.05)**	**0.88 (0.92)**	**0.12(0.06)**	**0.06**
SPARF - No 'scan30'	0.36	0.8	18.13 (19.53)	0.72 (0.82)	0.22 (0.11)	0.09	0.39	1.05	22.34 (23.16)	0.83 (0.88)	0.14 (0.08)	0.05	0.25	0.8	25.35 (25.86)	0.88 (0.92)	0.10 (0.06)	0.04

Table 6. Evaluation on DTU [9] with different numbers of input views (3, 6, or 9) and considering noisy initial poses. We simulate noisy poses by adding 15% of Gaussian noise to the ground-truth poses. The results for 3 input views correspond to Tab. 4 of the main paper and are repeated here for ease of comparison. Rotation errors are in ° and translation errors are multiplied by 100. Results in (·) are computed by masking the background.

LPIPS and depth error compared to the fixed ground-truth pose version. The lower PSNR and SSIM values indicate that the NeRF model still learns artifacts during the joint refinement. Note that this issue can be partially circumvented by re-initializing the NeRF model and training from scratch with fixed poses, once the poses have converged (see Tab. 3).

Results with 6 and 9 views: In Tab. 4 of the main paper, we evaluate our proposed approach SPARF for joint pose-NeRF training, when considering only *3 input views*. For completeness, we here provide results when 6 or 9 input views are available. As in the 3-view setting, we synthetically perturb the ground-truth poses by adding 15% of additive Gaussian noise. The results are presented in Tab. 6. We included the results with 3 input views for ease of comparison. The trend is similar for 3, 6, or 9 input views. BARF, RegBARF, and DistBARF struggle to refine the initial noisy poses, leading to poor novel-view rendering performance. While increasing the number of views leads to better synthesis quality, it remains drastically lower than the performance obtained by our SPARF. SCNeRF performs better at registering the poses. The rendering quality and learned geometry are nevertheless much worse than the proposed SPARF.

	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
BARF [11]	2.46	6.72	21.67 (21.71)	0.77 (0.84)	0.21 (0.13)	0.14
SPARF (Ours)	**1.0**	**1.23**	**24.77 (24.41)**	**0.85 (0.89)**	**0.15 (0.10)**	**0.05**

Table 7. Evaluation on DTU, considering all available training views (25) and initial noisy poses. We simulate noisy poses by adding 15% of Gaussian noise to the ground-truth poses. It leads to an initial rotation and translation error of 13.36° and 47.87 respectively. Rotation errors are in degree and translation errors are multiplied by 100. Results in (·) are computed by masking the background. Also note that some of the training images have inconsistent illumination, making them unsuitable for the NeRF training.

With 3, 6, or 9 input views, our SPARF outperforms all previous works. For completeness, we also provide results of our approach when excluding one of the scenes, *i.e.*'scan30', on which no correspondences are found. When excluding this scene, the rotation and translation errors of the optimized scenes are below 1° and 1 (multiplied by 100) respectively. The average novel-view rendering performance is also significantly increased.

Qualitative comparisons: We provide qualitative comparisons for the 3-view regime. In Fig. 6, we show the initial and optimized poses on one scene of DTU. We visually compare the novel-view renderings (RGB and depth) of our SPARF, SCNeRF, BARF, and RegBARF in Fig. 8.

Finally, we provide extensive examples of the novel-view synthesis capabilities of our approach SPARF in

	2 input views					6 input views					9 input views				
	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BARF [11]	5.16	37.87	15.06	0.35	0.50	**0.25**	*0.37*	*23.09*	**0.72**	**0.22**	*0.20*	*0.34*	*24.10*	**0.76**	**0.20**
RegBARF [11, 13]	*3.77*	*28.59*	*15.94*	*0.40*	*0.47*	*0.48*	0.55	22.21	*0.68*	0.26	0.93	4.2	22.68	*0.70*	*0.26*
DistBARF [2, 11]	7.32	110.0	14.06	0.30	0.55	2.38	11.23	18.31	0.52	0.37	2.81	13.41	20.36	0.59	0.34
SCNeRF [10]	4.88	44.27	14.43	0.32	0.51	2.07	8.11	21.82	0.66	0.26	0.47	3.87	22.72	0.70	0.24
SPARF (Ours)	**1.54**	**8.38**	**17.32**	**0.47**	**0.40**	**0.25**	**0.32**	**23.30**	**0.72**	*0.23*	**0.18**	**0.30**	**24.12**	**0.76**	**0.20**

Table 8. Evaluation on LLFF [17] with different numbers of input views (2, 6, or 9) and starting from initial identity poses. The results for 3 input views can be found in Tab. 5 of the main paper. Rotation errors are in ° and translation errors are multiplied by 100. The best and second-best results are in red and blue respectively.

Fig. 9. It produces realistic novel views with accurate geometry on a large variety of scenes and from many different viewing directions, given only 3 input views with noisy initial poses.

Results with all views: For completeness, we evaluate our joint pose and NeRF training approach SPARF, when many input views are available. While this is not the goal of this work, which was specifically designed for the sparse-view regime, we show here that it can generalize to the many-view setting. We present results on DTU in Tab. 7. Even in this setting, our SPARF significantly outperforms baseline BARF [11] in pose registration and novel-view synthesis performance. We note that some of the training images have inconsistent illumination, which were excluded when considering subsets. Inconsistent illumination can cause problems when training a NeRF since it relies on the photometric loss as the primary training signal. This explains why the PSNR and SSIM values obtained by SPARF with all 25 input views (Tab. 7) are slightly worse than when trained on only a subset of 9 views (Tab. 6).

D.2. Results on the LLFF dataset

Here, we present additional results for our joint pose-NeRF refinement approach SPARF, evaluated on the LLFF dataset [9].

Results with 2, 6 and 9 views: As for DTU [9], we here evaluate our pose-NeRF refinement approach when *6 or 9* input views are available instead of only 3. For completeness, we also include results when only 2 views are available.

When considering 2 or 3 input views, BARF struggles to refine the poses, which impacts its novel-view synthesis performance. Nevertheless, LLFF represents forward-facing scenes, for which a limited number of homogeneously spread views can cover the majority of the scene. As a result, for 6 input views and more, the 3D space is sufficiently constrained for BARF to successfully register the initial identity poses. In the 6 and 9 view cases, our approach SPARF and BARF obtain similar performance in pose registration and novel-view rendering quality.

	Rot. (°) ↓	Trans. (x 100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BARF [11]	0.85	0.26	25.09	0.77	**0.20**
SPARF*	**0.77**	**0.23**	**25.18**	**0.78**	**0.20**

Table 9. Evaluation on LLFF [17], considering all available training views and initial identity poses. Here, SPARF* indicates that we only use the combination of the photometric loss with our multi-view correspondence objective (Sec. 4.1 of m.p.), without including our depth consistency objective (Sec. 4.2 of m.p) nor our staged training (Sec. 4.3 of m.p.).

Interestingly, while adding the depth regularization loss (RegBARF) to the photometric loss (BARF) helps the pose registration and novel-view rendering performance in the 2 and 3-view regimes, it is harmful with denser views (6 and 9). Our approach SPARF instead does not negatively impact the performance of BARF in the 6 and 9-view scenarios. Surprisingly, SCNeRF obtains worse registration and novel-view rendering results than BARF, and consequently our approach SPARF.

We visualize the initial and optimized poses for one scene of LLFF in the 3 and 6 views scenario in Fig. 7. Here, it is visible that even 6 views can cover most of the scene, which is why BARF performs well even in this sparse-view regime. In Fig. 10, we visually compare novel-view renderings of SPARF, BARF, RegBARF, and SCNeRF in the 3-view setting. Our approach encodes the scene geometry more accurately. The RGB renderings also contain fewer artifacts and blurriness. Finally, we provide examples of the renderings produced by our approach SPARF on multiple scenes of LLFF and from different viewpoints in Fig. 11. Given as few as 3 input views with initial identity poses, SPARF produces realistic novel-view renderings from many different viewing directions. It also leads to a geometrically accurate scene.

Results with all views: For completeness, in Tab. 9 we compare joint pose-NeRF training approaches BARF and SPARF, considering all available training views of LLFF, and starting from identity poses. On this forward-facing dataset, BARF and SPARF reach a similar performance in the many-view regime.

D.3. Results on the Replica dataset

We here provide additional evaluation results on the Replica dataset, with different pose initialization schemes. We also include more qualitative examples.

Further analysis on Tab. 6 of the main paper: In Tab. 6 of the main paper, we evaluated multiple approaches on Replica, with 3 input views and initial poses obtained by COLMAP [16] with PDC-Net [19] matches. Those initial poses have an error of 0.39° and 3.01 in rotation and translation respectively. In the bottom part of the table, we show that SPARF can refine the initial poses to a final rotation and translation error of 0.15 and 0.76 respectively. While this might seem like a small improvement in terms of pose registration, the rendering quality improves a lot between SPARF with fixed COLMAP poses (F) and SPARF with pose refinement (R). This is because the provided initial rotation and translation errors are an *average* over all the scenes. Some scenes actually have an initial translation error of up to 8, which can cause a notable drop in rendering quality. Refining the poses for those scenes is then particularly beneficial in terms of rendering quality. This explains the PSNR difference between SPARF in (F) or in (R).

Moreover, some of the baselines show similar rendering quality despite larger pose differences because they struggle to learn a meaningful geometry, *i.e.* they cannot go beyond a certain PSNR. Finally, rendering scores are overall higher on Replica compared to other datasets (even for poor pose registration), because the dataset contains many homogeneous surfaces (*e.g.* wall).

COLMAP initialization w. SP-SG matches: In the main paper, we compared joint pose-NeRF refinement approaches considering initial poses obtained by COLMAP [16] run with PDC-Net [19] matches. For completeness, we here present the same comparison, when the initial poses are obtained with COLMAP with SuperPoint [7] and SuperGlue [14] matches instead. It corresponds to an initial rotation and translation errors of 2.61° and 15.31 respectively. The results are presented in Tab. 10.

Compared to initialization with COLMAP-PDCNet (Tab 6 of main paper), the same conclusions apply. Comparing the top (G) and middle part (F) of Tab. 10, we show that even a relatively low initial error impacts the novel-view rendering quality when using fixed poses. In the bottom part (R), our pose-NeRF training strategy SPARF leads to the best results, matching the accuracy obtained by our approach with perfect poses (top row, G).

		Rot (°) ↓	Trans (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
G	**SPARF**	Fixed GT poses		26.43	**0.88**	**0.13**	0.39
F	NeRF	Fixed poses obtained from COLMAP (run w. SP-SG [14] matches)		19.50	0.66	0.41	1.63
	DS-NeRF [6]			21.55	0.74	0.26	0.91
	SPARF (Ours)			22.18	0.74	0.25	0.93
R	BARF [11]	3.23	18.05	19.41	0.68	0.34	0.95
	SCNeRF [10]	0.21	1.17	23.67	0.82	0.22	0.83
	DS-NeRF	1.01	3.85	24.68	0.83	0.18	0.70
	SPARF (Ours)	**0.16**	**0.8**	**26.80**	**0.88**	0.14	**0.36**

Table 10. Evaluation on Replica [18] (3 views) with initial poses obtained by COLMAP [16, 19] with SP-SG [14] matches. The initial rotation and translation errors are 2.61° and 15.31 respectively. In the middle part (F), these initial poses are fixed and used as "pseudo-gt". In the bottom part (R), the poses are refined along with training the NeRF. For comparison, in the top part (G), we use fixed ground-truth poses.

Initial noisy poses: For completeness, we also start from synthetically perturbed ground-truth poses. In particular, as previously for DTU, we synthetically perturb the ground-truth poses with 15% of additive Gaussian noise. It leads to an initial rotation and translation errors of 15.62° and 112 (scaled by 100) respectively. This corresponds to a significantly noisier setting than starting from COLMAP poses. Results are presented in Tab. 11. BARF struggles to refine the poses. RegBARF and DistBARF lead to better pose registration and novel-view synthesis. Here, it is interesting to note that both regularizations seem to help in learning a more accurate geometry (lower depth error). Indeed, SCNeRF, which better registers the poses, still obtains a higher depth error. Our approach SPARF, which acts on *both* the learned scene geometry and the camera poses, significantly outperforms all others.

Qualitative comparisons: In Fig. 12, we qualitatively compare SPARF with BARF, DS-NeRF and SCNeRF. Our approach SPARF produces the best renderings, with significantly fewer floaters and blurry surfaces. The learned scene geometry is also significantly sharper and more accurate, as shown by the depth renderings. This is confirmed in Fig. 13, where we present additional renderings produced by SPARF on all scenes of the Replica dataset. Note that in all those cases, our approach is only trained with 3 input views, and noisy input camera poses (obtained by COLMAP-PDCNet).

E. Additional Results with Fixed GT Poses

In Sec. 5.4 of m.p., we evaluated our approach when considering fixed ground-truth poses, in the three-input-views setting. For completeness, we extend this evaluation for the cases of 6 and 9 input views. This is the same setup as in [13].

Results on DTU: We present results on DTU in Tab. 12. Our approach SPARF sets a new state of the art on all met-

	Pose Registration		Novel View Synthetis			
	Rot (°) ↓	Trans (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
BARF [11]	12.81	39.96	16.39	0.60	0.52	2.3
RegBARF [11, 13]	9.0	29.34	17.05	0.62	0.48	1.11
DistBARF [2, 13]	5.28	20.45	19.82	0.69	0.36	0.68
SCNeRF [10]	2.26	10.37	22.50	0.76	0.27	1.57
SPARF	**1.06**	**6.63**	**25.57**	**0.85**	**0.16**	**0.45**

Table 11. Evaluation on the Replica dataset (3 views) starting from noisy poses. In particular, the ground-truth poses are synthetically perturbed with 15% of additive Gaussian noise. This initialization leads to an initial rotation and translation errors of 15.62° and 112 (multiplied by 100) respectively.

	PSNR ↑			**SSIM** ↑			**LPIPS** ↓			**DE** ↓		
	3	6	9	3	6	9	3	6	9	3	6	9
PixelNeRF [25]	**19.36** (18.00)	20.46 (19.12)	20.91 (19.56)	*0.70* (*0.77*)	0.75 (0.80)	0.76 (0.81)	*0.32* (0.23)	0.30 (0.22)	0.29 (0.21)	*0.12*	*0.12*	0.13
NeRF [12]	8.41 (9.34)	17.51 (18.52)	21.45 (23.25)	0.31 (0.63)	0.73 (0.83)	*0.85* (*0.91*)	0.71 (0.36)	0.25 (0.13)	*0.14* (*0.06*)	0.87	0.21	*0.08*
DietNeRF [8]	10.01 (11.85)	18.70 (20.63)	22.16 (23.83)	0.35 (0.63)	0.67 (0.78)	0.68 (0.82)	0.57 (0.31)	0.35 (0.20)	0.34 (0.17)	-	-	-
RegNeRF [13]	15.33 (*18.89*)	19.10 (*22.20*)	*22.30* (*24.93*)	0.62 (0.75)	*0.76* (*0.84*)	0.82 (0.88)	0.34 (*0.19*)	*0.23* (*0.12*)	0.18 (0.09)	-	-	-
DS-NeRF [6]	16.52 (-)	*20.54* (-)	22.23 (-)	0.54 (-)	0.73 (-)	0.77 (-)	0.48 (-)	0.31 (-)	0.26 (-)	-	-	-
SPARF (Ours)	*18.30* (**21.01**)	**23.24** (**25.76**)	**25.75** (**27.30**)	**0.78** (**0.87**)	**0.87** (**0.92**)	**0.91** (**0.94**)	**0.21** (**0.10**)	**0.12** (**0.06**)	**0.08** (**0.04**)	**0.083**	**0.049**	**0.043**

Table 12. Evaluation on the DTU dataset [9], considering fixed ground-truth poses. We present novel-view synthesis results for different numbers of input views. Results in (·) are computed by masking the background. Results of [1, 3, 8, 13, 25] are from [13]. The best and second-best results are in red and blue respectively.

	PSNR ↑			**SSIM** ↑			**LPIPS** ↓		
	3	6	9	3	6	9	3	6	9
PixelNeRF [25]	7.93	8.74	8.61	0.27	0.28	0.27	0.68	0.68	0.67
SRF [4]	12.3	13.1	13.0	0.25	0.29	0.30	0.59	0.59	0.61
MVSNeRF [3]	17.25	19.79	20.47	0.56	0.66	0.69	0.36	0.27	0.24
PixelNeRF-ft	16.17	17.03	18.92	0.44	0.47	0.54	0.51	0.48	0.43
SRF-ft	17.07	16.75	17.39	0.44	0.44	0.47	0.53	0.52	0.50
MVSNeRF-ft	17.88	19.99	20.47	0.58	0.66	0.70	0.33	0.26	0.24
NeRF [12]	13.61	16.70	18.45	0.28	0.43	0.51	0.56	0.40	0.31
MipNeRF [1]	14.62	20.87	24.26	0.35	0.69	*0.81*	0.50	0.26	*0.17*
DietNeRF* [8]	14.94	21.75	24.3	0.37	0.72	0.80	0.5	0.25	0.18
RegNeRF* [13]	*19.08*	*23.10*	**24.86**	*0.59*	**0.76**	**0.82**	0.34	*0.21*	**0.16**
DS-NeRF [6]	18.00	21.60	22.84	0.55	0.67	0.71	*0.27*	*0.21*	0.19
SPARF (Ours)	**20.20**	**23.35**	*24.40*	**0.63**	*0.74*	0.77	**0.24**	**0.20**	0.18

Table 13. Evaluation on the LLFF dataset [17], considering fixed ground-truth poses. We present novel-view synthesis results for different numbers of input views. The top part contains conditional models trained on DTU. In the middle part, we present the same conditional models, further finetuned per scene on LLFF. Finally, in the last part, we compare per-scene NeRF-based approaches. Approaches with $*$ use the MipNeRF [1] as their base architecture, while the others use NeRF [12]. Results of [1, 3, 8, 13, 25] are from [13]. The best and second-best results are in red and blue respectively.

rics for 3, 6, or 9 input views. The only exception is PSNR on the whole image when only 3 input views are available, which we already mentioned in the main paper.

Results on LLFF: We present results on LLFF in Tab. 13. The conditional models PixelNeRF, SRF, and MVSNeRF are trained on the DTU dataset. LLFF thus serves as an out-of-distribution scenario. It appears that SRF and PixelNeRF tend to overfit to the training data, leading to poor quantitative results. MVSNeRF generalizes better to novel data. All three conditional models seem to benefit from additional fine-tuning. For 3 input views, NeRF, MipNeRF, and DietNeRF perform worse than conditional models. DS-NeRF, RegNeRF, and our approach SPARF nevertheless outperform the best conditional model, *i.e*. MVSNeRF. In the 6 and 9 view settings, all per-scene approaches except for the standard NeRF outperform MVSNeRF.

Our approach SPARF outperforms all others on all metrics in the sparsest scenario, *i.e*. when considering 3 input views. For 6 and 9 views, it obtains a slightly lower performance than MipNeRF and RegNeRF, the latter using MipNeRF as the base architecture. Nevertheless, our SPARF, which is based on the NeRF architecture, obtains drastically better results than the standard NeRF or DS-NeRF. Our approach could in theory be applied to any base network, for example, MipNeRF. As a result, we believe combining our approach with the MipNeRF base architecture could lead to even better rendering quality.

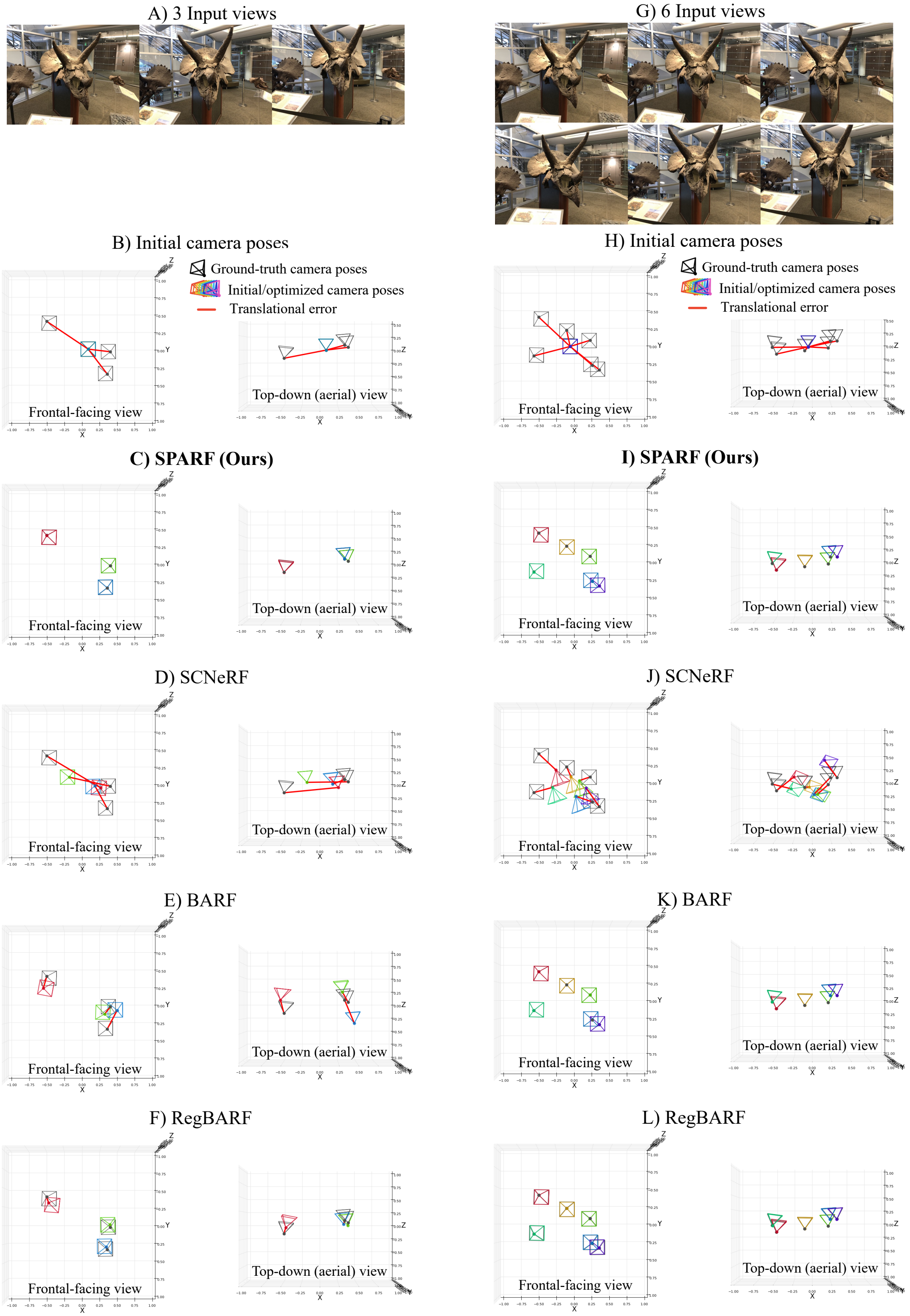


Figure 7. Initial and optimized camera poses on the scene ’horns’ of the LLFF dataset. We consider 3 or 6 input views with initial identity poses.

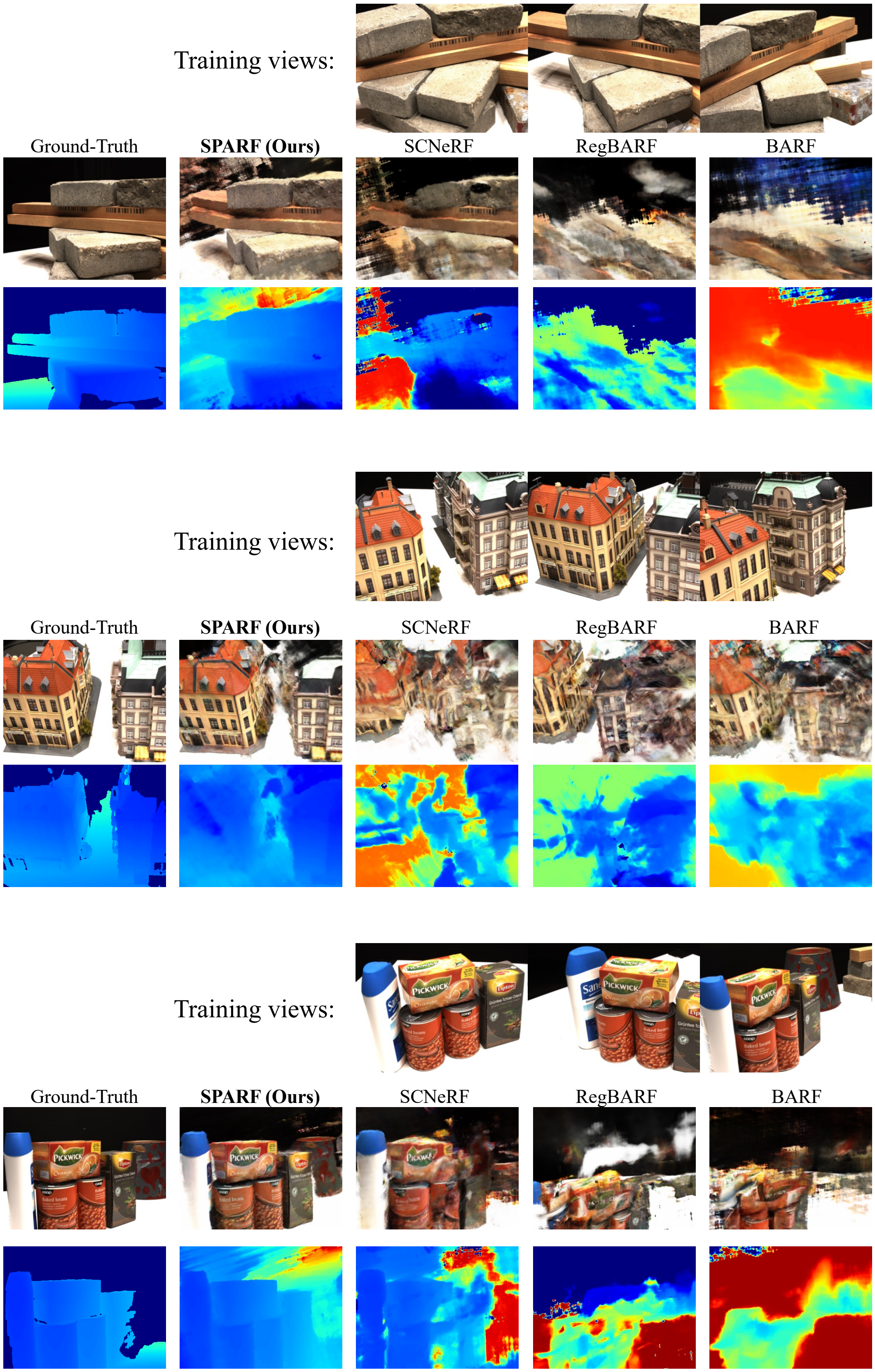


Figure 8. Novel-view renderings of alternative joint pose-NeRF training approaches on the DTU dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from an unseen viewpoint. We consider 3 input views with initial noisy poses. The initial camera poses are created by perturbing the ground-truth poses with 15% of additive Gaussian noise.

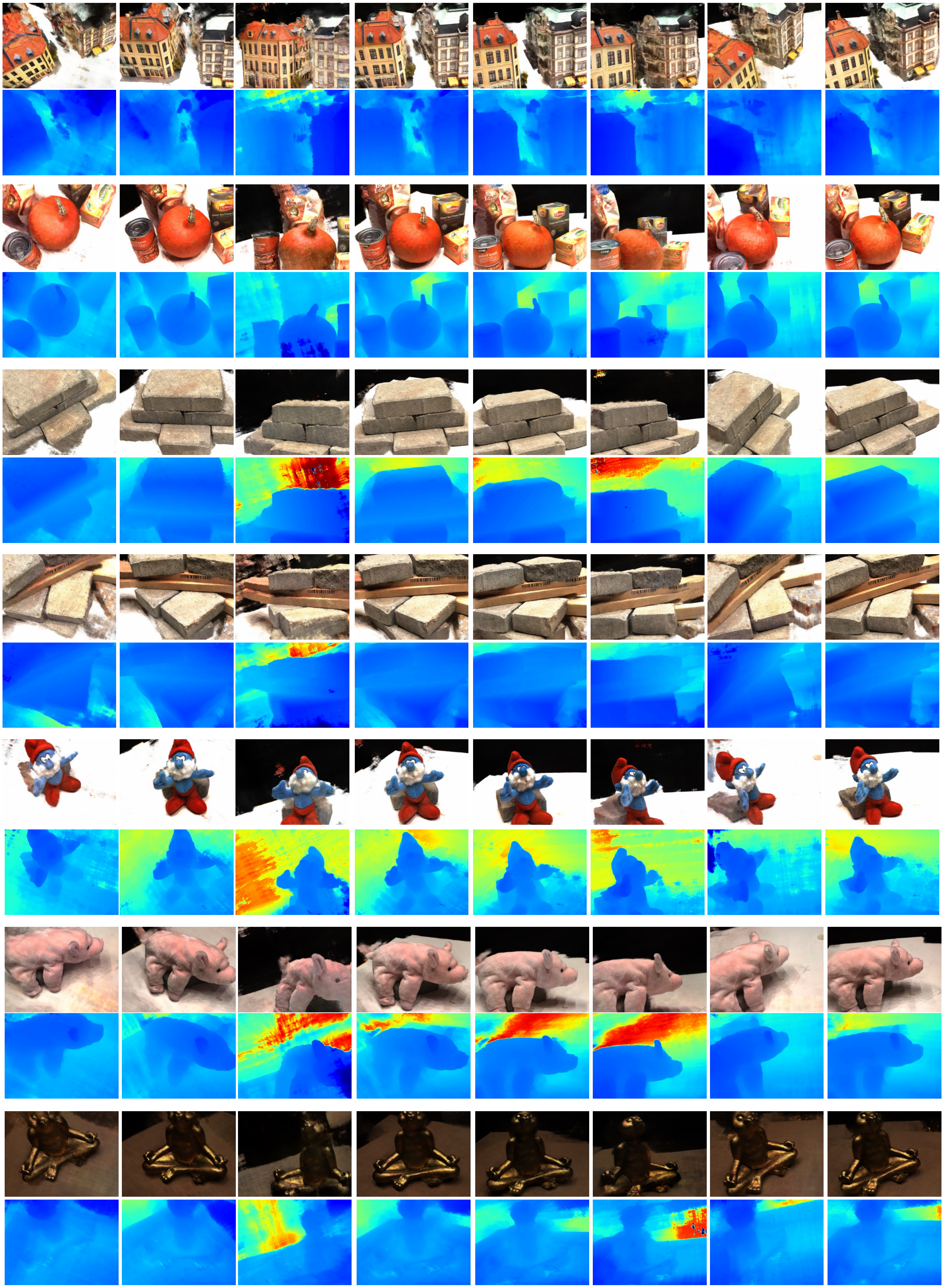


Figure 9. Novel-view renderings of our SPARF on the DTU dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from multiple unseen viewpoints. In each scene, we consider 3 input views (not shown here) with initial noisy poses, created by perturbing the ground-truth poses with 15% of additive Gaussian noise.

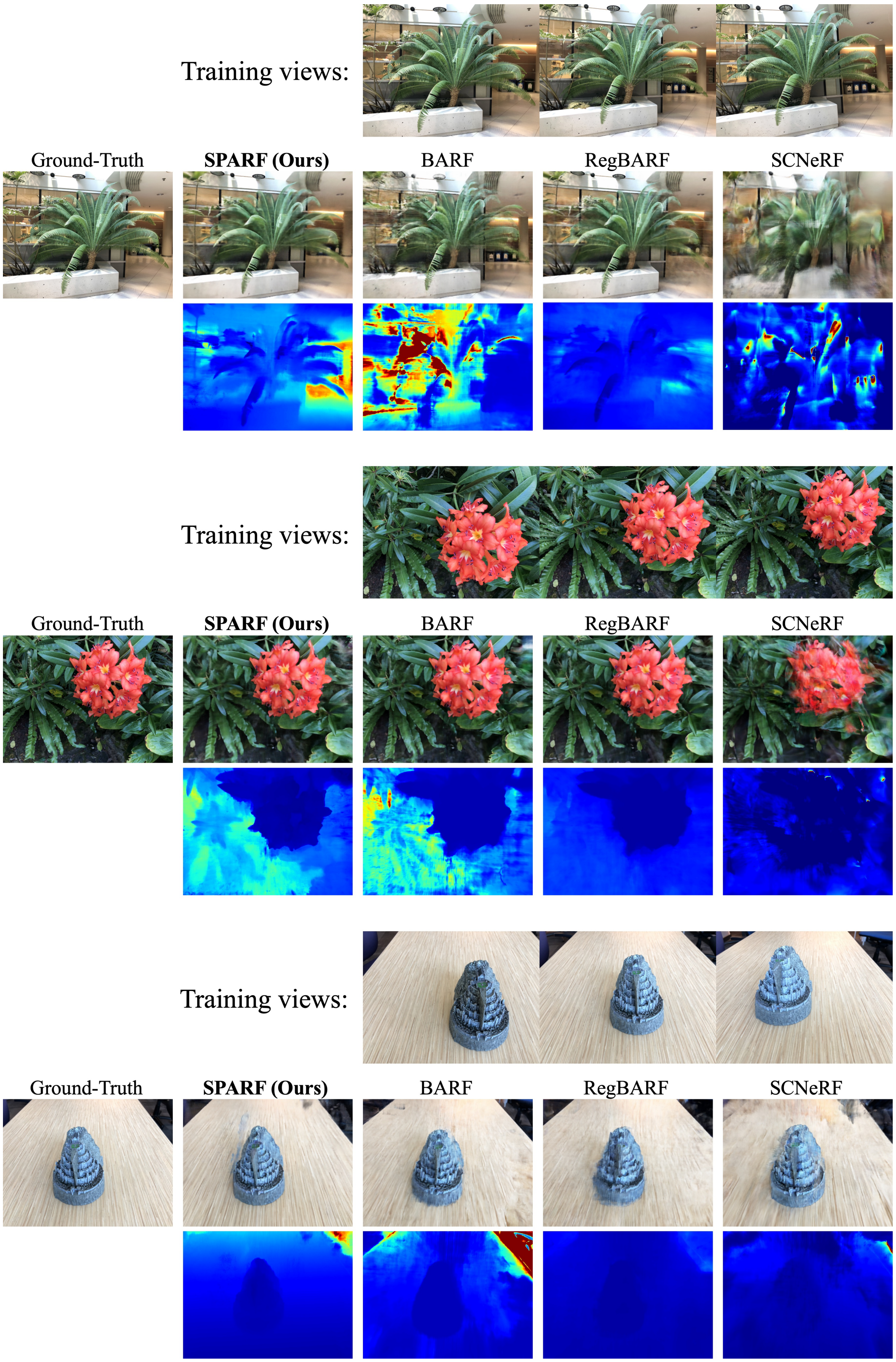


Figure 10. Novel-view renderings of alternative joint pose-NeRF training approaches on the LLFF dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from an unseen viewpoint. We consider 3 input views with initial identity poses.

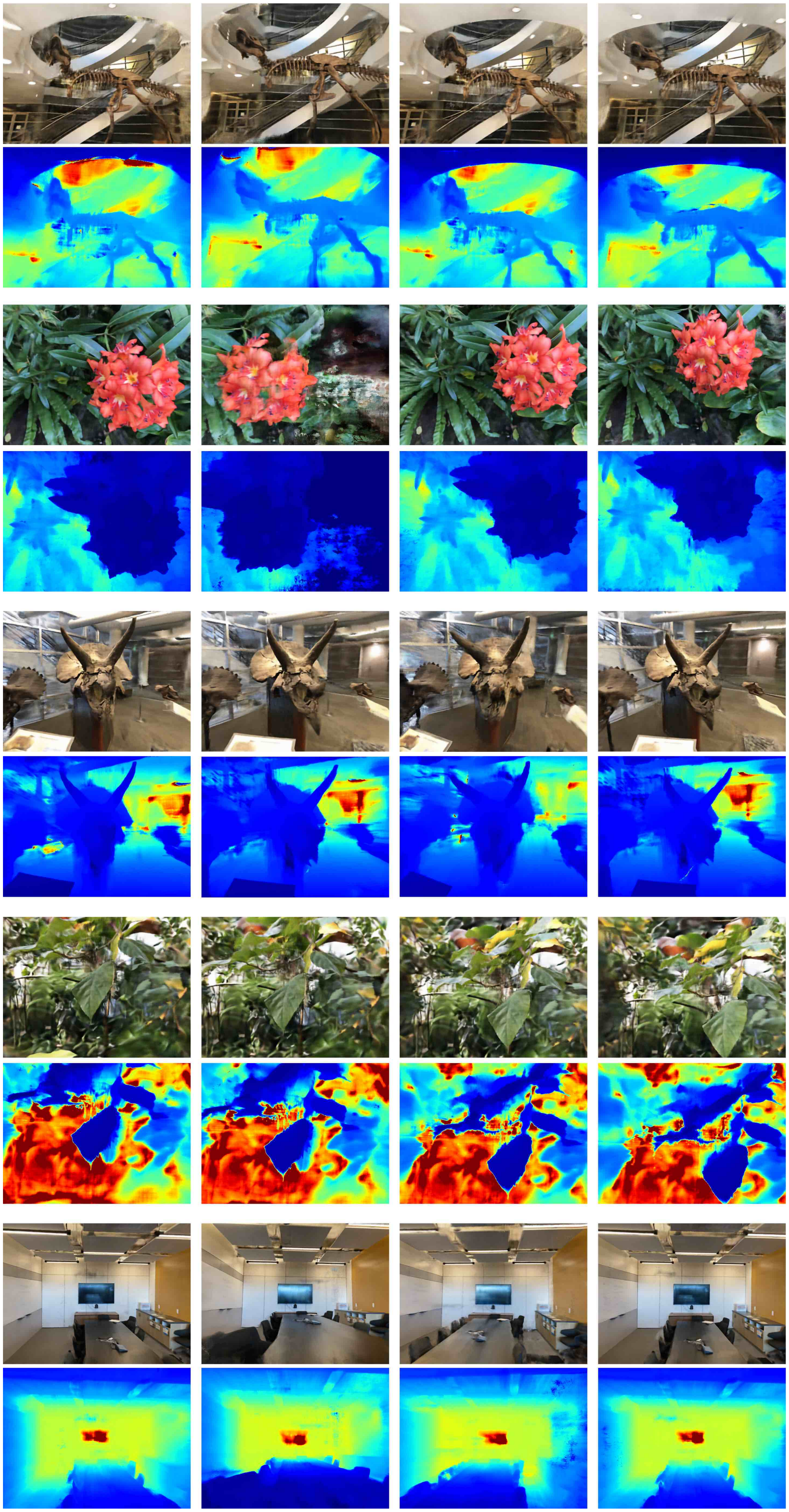


Figure 11. Novel-view renderings of our SPARF on the LLFF dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from multiple unseen viewpoints. In each scene, we consider 3 input views (not shown here) with initial identity poses.

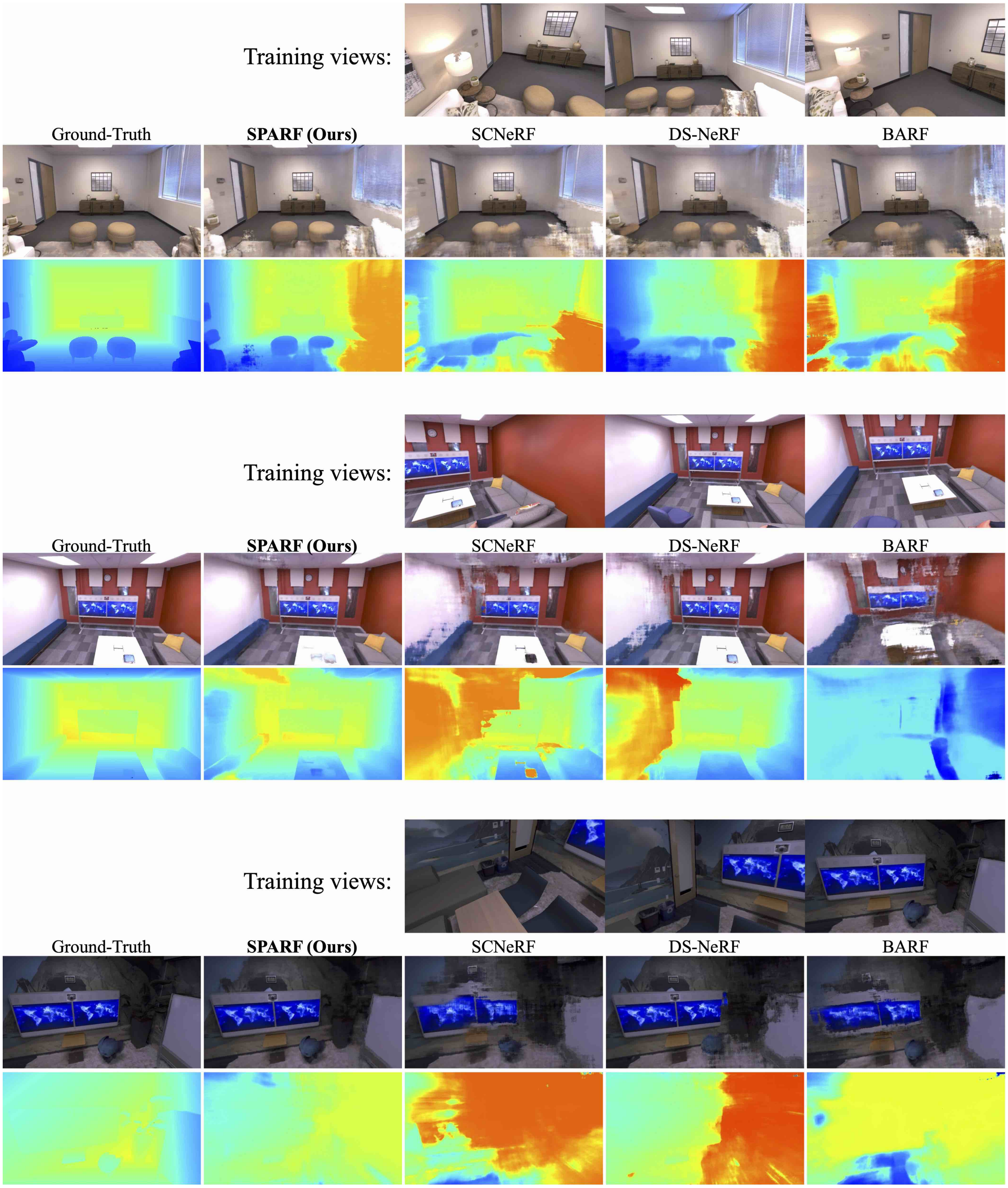


Figure 12. Novel-view renderings of alternative joint pose-NeRF training approaches on the Replica dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from an unseen viewpoint. On each scene, we consider 3 input views (not shown here) with initial poses obtained by COLMAP [16] with PDC-Net matches [19].

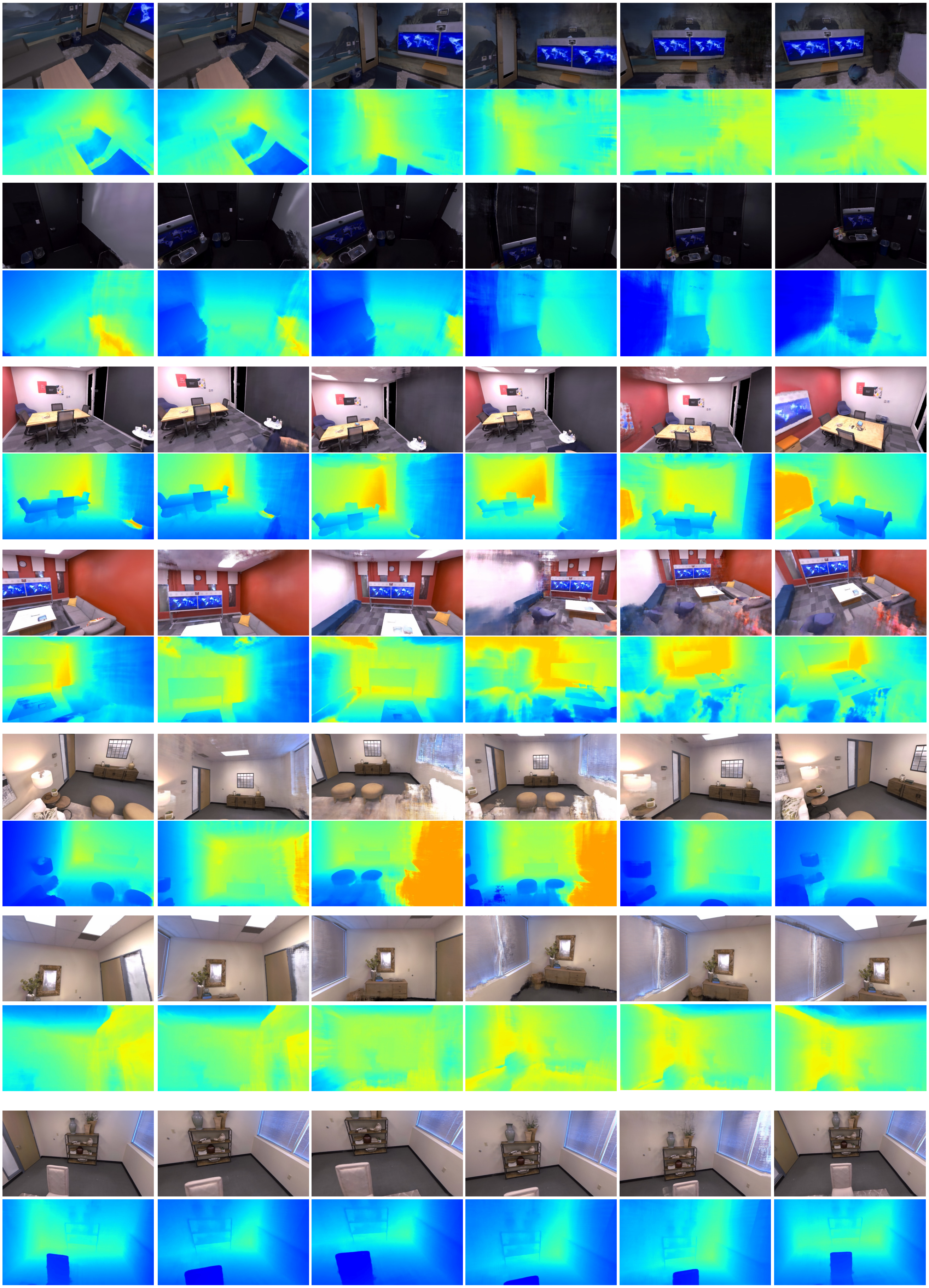


Figure 13. Novel-view renderings of our SPARF on the Replica dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from multiple unseen viewpoints. On each scene, we consider 3 input views (not shown here) with initial poses obtained by COLMAP [16] with PDC-Net matches [19].

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 *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5835–5844. IEEE, 2021. 13

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5460–5469. IEEE, 2022. 10, 11, 12

[3] 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 *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 14104–14113. IEEE, 2021. 13

[4] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7907–7916, 2021. 13

[5] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 6250–6259. IEEE, 2022. 9

[6] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 4, 10, 12, 13

[7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 224–236, 2018. 4, 5, 6, 7, 12

[8] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, October 2021. 13

[9] Rasmus Ramsbøl Jensen, Anders Lindbjerg Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 406–413. IEEE Computer Society, 2014. 5, 7, 8, 9, 10, 11, 13

[10] Yoonwoo Jeong, Seokjun Ahn, Christopher B. Choy, Animashree Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5826–5834, 2021. 3, 4, 7, 10, 11, 12

[11] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 3, 4, 5, 7, 9, 10, 11, 12

[12] 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. *Commun. ACM*, 65(1):99–106, 2022. 1, 4, 10, 13

[13] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4, 10, 11, 12, 13

[14] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4937–4946, 2020. 4, 5, 6, 7, 9, 10, 12

[15] Paul-Edouard Sarling. HLOC: Github project page. https://github.com/cvg/Hierarchical-Localization, 2021. 4

[16] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR 2016, Las Vegas, NV, USA*, pages 4104–4113, 2016. 3, 5, 6, 9, 12, 19, 20

[17] Mohammad Shafiei, Sai Bi, Zhengqin Li, Aidas Liaudanskas, Rodrigo Ortiz Cayon, and Ravi Ramamoorthi. Learning neural transmittance for efficient rendering of reflectance fields. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, page 45. BMVA Press, 2021. 7, 11, 13

[18] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard A. Newcombe. The replica dataset: A digital replica of indoor spaces. *CoRR*, abs/1906.05797, 2019. 12

[19] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5714–5724. Computer Vision Foundation / IEEE, 2021. 1, 2, 4, 7, 9, 10, 12, 19, 20

[20] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380, 1991. 5

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

[22] 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, 8

[23] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 4

[24] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. 5

[25] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 13

[26] 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 (CVPR)*, June 2018. 5

[27] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*, pages 7244–7251. IEEE, 2018. 5

[28] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5745–5753. Computer Vision Foundation / IEEE, 2019. 3