SPARF: Neural Radiance Fields from Sparse and Noisy Poses

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

Website: prunetruong.com/sparf.github.io/

Abstract

Neural Radiance Field (NeRF) has recently emerged as a powerful representation to synthesize photorealistic novel views. While showing impressive performance, it relies on the availability of dense input views with highly accurate camera poses, thus limiting its application in real-world scenarios. In this work, we introduce Sparse Pose Adjusting Radiance Field (SPARF), to address the challenge of novel-view synthesis given only few wide-baseline input images (as low as 3) with noisy camera poses. Our approach exploits multi-view geometry constraints in order to jointly learn the NeRF and refine the camera poses. By relying on pixel matches extracted between the input views, our multi-view correspondence objective enforces the optimized scene and camera poses to converge to a global and geometrically accurate solution. Our depth consistency loss further encourages the reconstructed scene to be consistent from any viewpoint. Our approach sets a new state of the art in the sparse-view regime on multiple challenging datasets.

1. Introduction

Novel-view synthesis (NVS) has long been one of the most essential goals in computer vision. It refers to the task of rendering unseen viewpoints of a scene given a particular set of input images. NVS has recently gained tremendous popularity, in part due to the success of Neural Radiance Fields (NeRFs) [30]. NeRF encodes 3D scenes with a multi-layer perceptron (MLP) mapping 3D point locations to color and volume density and uses volume rendering to synthesize images. It has demonstrated remarkable abilities for high-fidelity view synthesis under two conditions: dense input views and highly accurate camera poses.

Both these requirements however severely impede the usability of NeRFs in real-world applications. For instance, in AR/VR or autonomous driving, the input is inevitably much sparser, with only few images of any particular object or region available per scene. In such sparse-view scenario, NeRF rapidly overfits to the input views [11, 22, 32], leading to inconsistent reconstructions at best, and degenerate solutions at worst (Fig. 1 left). Moreover, the de-facto standard to estimate per-scene poses is to use an off-the-shelf Structure-from-Motion approach, such as COLMAP [38]. When provided with many input views, COLMAP can generally estimate accurate camera poses. Its performance nevertheless rapidly degrades when reducing the number of views, or increasing the baseline between the images [58].

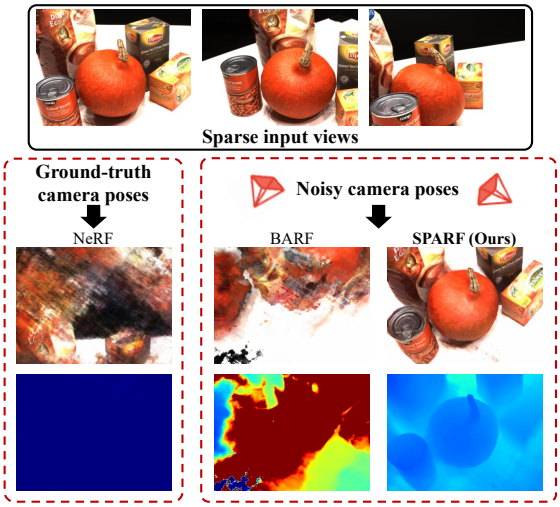

Figure 1. **Novel-view rendering from sparse images**. We show the RGB (second row) and depth (last row) renderings from an unseen viewpoint under sparse settings (3 input views only). Even with ground-truth camera poses, NeRF [30] overfits to the training images, leading to degenerate geometry (almost constant depth). BARF [24], which can successfully handle noisy poses when dense views are available, struggles in the sparse regime. Our approach SPARF instead produces realistic novel-view renderings with accurate geometry, given only 3 input views with noisy poses.

Multiple works focus on improving NeRF's performance in the sparse-view setting. One line of research [6, 56] trains conditional neural field models on large-scale datasets. Alternative approaches instead propose various regularization on color and geometry for per-scene training [11, 19, 22, 32, 34]. Despite showing impressive results in the sparse scenario, all these approaches assume *perfect camera poses* as a pre-requisite. Unfortunately, estimating accurate camera poses for few wide-baseline images is challenging [58] and has spawned its own research direction [1, 7, 14–16, 28, 64],

*This work was conducted during an internship at Google.

hence making this assumption unrealistic.

Recently, multiple approaches attempt to reduce the dependency of NeRFs on highly accurate input camera poses. They rely on per-image training signals, such as a photometric [9, 24, 29, 50, 52] or silhouette loss [5, 23, 59], to jointly optimize the NeRF and the poses. However, in the sparse-view scenario where the 3D space is under-constrained, we observe that it is crucial to explicitly *exploit the relation between* the different training images and their underlying scene geometry, to enforce learning a *global and geometrically accurate solution*. This is not the case of previous works [5, 23, 24, 50, 52, 59], which hence fail to register the poses in the sparse regime. As shown in Fig. 1, middle for BARF [24], it leads to poor novel-view synthesis quality.

We propose Sparse Pose Adjusting Radiance Field (SPARF), a joint pose-NeRF training strategy. Our approach produces realistic novel-view renderings given only *few wide-baseline input images* (as low as 3) with *noisy camera poses* (see Fig. 1 right). Crucially, it does not assume any prior on the scene or object shape. We introduce novel constraints derived from multi-view geometry [17] to drive and bound the NeRF-pose optimization. We first infer pixel correspondences relating the input views with a pre-trained matching model [44]. These pixel matches are utilized in our multi-view correspondence objective, which minimizes the re-projection error using the depth rendered by the NeRF and the current pose estimates. Through the explicit connection between the training views, the loss enforces convergence to a global and geometrically accurate pose/scene solution, consistent across all training views. We also propose the depth consistency loss to boost the rendering quality from novel viewpoints. By using the depth rendered from the training views to create pseudo-ground-truth depth for unseen viewing directions, it encourages the reconstructed scene to be consistent *from any viewpoint*. We extensively evaluate and compare our approach on the challenging DTU [20], LLFF [39], and Replica [40] datasets, setting a new state of the art on all three benchmarks.

2. Related Work

We review approaches focusing on few-shot novel view rendering as well as joint pose-NeRF refinement.

Sparse input novel-view rendering: To circumvent the requirement of dense input views, a line of works [6, 8, 25, 42, 48, 56] incorporates prior knowledge by pre-training conditional models of radiance fields on large posed multi-view datasets. Despite showing promising results on sparse input images, their generalization to out-of-distribution novel views remains a challenge. Multiple works [11, 19, 22, 32, 34] follow a different direction, focusing on per-scene training for few-shot novel view rendering. DietNeRF [19] compares CLIP [33] embeddings of rendered and training views. InfoNeRF [22] penalizes the NeRF overfitting to limited input views with a ray entropy regularization. Similarly, Barron *et al*. [4] introduce a distortion loss, which encourages sparsity of the density in each ray. In RegNeRF, Niemeyer *et al*. [32] propose to regularize the geometry and appearance of rendered patches with a depth smoothness and normalizing flow objectives. Recently, a number of works [11, 34, 51, 57] incorporate depth priors to constraint the NeRF optimization. Notably, DS-NeRF [11] improves reconstruction accuracy by including additional sparse depth supervision. Related are also approaches that learn a signed distance function (SDF), aiming for accurate 3D reconstruction in the sparse-view scenario [26, 53]. However, all these works assume perfect poses as a prerequisite. We instead propose a novel training strategy leading to accurate geometry and novel-view renderings in the sparse regime, *even when facing imperfect input poses*.

Joint NeRF and pose refinement: Several approaches attempt to reduce NeRF's reliance on highly accurate input camera poses [9, 24, 29, 50, 52]. BARF [24] and NeRF-- [50] jointly optimize the radiance field and camera parameters of initial noisy poses, relying on the photometric loss as the only training signal. SiNeRF [52] and GARF [9] propose different activation functions, easing the pose optimization. GNeRF [29] introduces a sequential training approach including a rough initial pose network that uses GAN-style training, thereby circumventing the need for initial pose estimates. SCNeRF [21] proposes a geometric loss minimizing the ray intersection re-projection error at previously extracted sparse correspondences to optimize over camera extrinsics and intrinsics. A number of works [5, 23, 54] also combine the photometric objective with a silhouette or mask loss, requiring accurate foreground segmentation, and limiting their applicability to objects. Related are also implicit SLAM systems [2, 41, 63], which progressively optimize over the geometry and camera estimates of an input RGB-D sequence. While previous works assume a dense coverage of the 3D space, Zhang *et al*. propose NeRS [59], which tackles the task of single object reconstruction by deforming a unit sphere over time while refining poses of few input views. However, NeRS is restricted to simple objects with a known shape prior. We instead assume access to only few wide-baseline RGB images with noisy pose estimates, without any prior on the scene or object shape.

3. Preliminaries

We first briefly introduce notation, the basics of NeRF representation, and camera operations.

Camera pose: Let $P_i^{c2w} = \left[R_i^{c2w}|\mathbf{t}_i^{c2w}\right] \in SE(3)$ be the camera-to-world transform of camera i, where $R_i^{c2w} \in SO(3)$ and $\mathbf{t}_i^{c2w} \in \mathbb{R}^3$ are the rotation and translation, respectively. We denote as $K \in \mathbb{R}^{3\times3}$ the intrinsic matrix.

For the rest of the manuscript, we drop the superscript ^{c2w}. As a result, unless otherwise stated, $P = P^{c2w}$ and all 3D quantities are defined in the world coordinate system.

Camera projection: For any vector $\mathbf{x} \in \mathbb{R}^l$ of dimension l, $\bar{\mathbf{x}} \in \mathbb{R}^{l+1}$ corresponds to its homogeneous representation, *i.e.* $\bar{\mathbf{x}} = \left[\mathbf{x}^T, 1\right]$. We additionally define π to be the camera projection operator, which maps a 3D point in the camera coordinate frame $\mathbf{x}^c \in \mathbb{R}^3$ to a pixel coordinate $\mathbf{p} \in \mathbb{R}^2$. Likewise, π^{-1} is defined to be the backprojection operator, which maps a pixel $\mathbf{p}$ and depth z to a 3D point $\mathbf{x}^c$.

$$\pi(\mathbf{x}^c) \cong K\mathbf{x}^c, \qquad \pi^{-1}(\mathbf{p}, z) = zK^{-1}\bar{\mathbf{p}}\,. \tag{1}$$

Scene representation: We adopt the NeRF [30] framework to represent the underlying 3D scene and image formation. A neural radiance field is a continuous function that maps a 3D location $\mathbf{x} \in \mathbb{R}^3$ and a unit-norm ray viewing direction $\mathbf{d} \in \mathbb{S}^2$ to an RGB color $\mathbf{c} \in [0,1]^3$ and volume density $\sigma \in \mathbb{R}^+$. It can be formulated as

$$[\mathbf{c}, \sigma] = F_\theta\left(\gamma_x(\mathbf{x}), \gamma_d(\mathbf{d})\right)\,. \tag{2}$$

Here, F is an MLP with parameters θ, and $\gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^{3+6L}$ is a positional encoding function with L frequency bases.

Volume rendering: Given a camera pose P_i, each pixel coordinate $\mathbf{p} \in \mathbb{R}^2$ determines a ray in the world coordinate system, whose origin is the camera center of projection $\mathbf{o}_i = \mathbf{t}_i$ and whose direction is defined as $\mathbf{d}_{i,\mathrm{p}} = R_i K_i^{-1}\bar{\mathbf{p}}$. We can express a 3D point along the viewing ray associated with $\mathbf{p}$ at depth t as $\mathbf{r}_{i,\mathrm{p}}(t) = \mathbf{o}_i + t\mathbf{d}_{i,\mathrm{p}}$. To render the color $\hat{\mathbf{I}}_{i,\mathrm{p}} \in [0,1]^3$ at pixel $\mathbf{p}$, we sample M discrete depth values t_m along the ray within the near and far plane $[t_n, t_f]$, and query the radiance field F_θ (2) at the underlying 3D points. The corresponding predicted color and volume density values $\{(\mathbf{c}_m, \sigma_m)\}_{m=1}^M$ are then composited as,

$$\hat{\mathbf{I}}_{i,\mathrm{p}} = \hat{I}(\mathbf{p}; \theta, P_i) = \sum_{m=1}^{M} \alpha_m \mathbf{c}_m\,, \tag{3}$$

$$\text{where} \quad \alpha_m = T_m\left(1 - \exp(-\sigma_m \delta_m)\right)\,, \tag{4}$$

$$T_m = \exp\left(-\sum_{m'=1}^{m} \sigma_{m'}\delta_{m'}\right)\,. \tag{5}$$

T_m denotes the accumulated transmittance along the ray from t_n to t_m, and $\delta_m = t_{m+1} - t_m$ is the distance between adjacent samples. Similarly, the approximate depth of the scene viewed from pixel $\mathbf{p}$ is obtained as,

$$\hat{z}_{i,\mathrm{p}} = \hat{z}(\mathbf{p}; \theta, P_i) = \sum_{m=1}^{M} \alpha_m t_m\,. \tag{6}$$

Here, $\hat{I}$ and $\hat{z}$ denote the RGB and depth rendering functions. In practice, NeRF [30] trains two MLPs, a coarse network F_θ^c and a fine network F_θ^f, where the former is used to guide sampling along the ray for the latter, thereby enabling more accurate estimation of (3)-(6).

Photometric loss: Given a dataset of n RGB images $\mathcal{I} = \{I_1, I_2, ..., I_n\}$ of a scene associated with initial noisy poses $\hat{\mathcal{P}} = \left\{\hat{P}_1, \hat{P}_2, ..., \hat{P}_n\right\}$, previous approaches [9, 24, 50, 52] optimize the radiance field function F_θ along with the camera pose estimates $\hat{\mathcal{P}}$ using a photometric loss as follows,

$$\mathcal{L}_{\text{photo}}(\theta, \hat{\mathcal{P}}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{\mathrm{p}} \left\| I_i(\mathbf{p}) - \hat{I}(\mathbf{p}; \theta, \hat{P}_i)\right\|_2^2\,. \tag{7}$$

While this works well with dense views, it fails in the sparse regime. We propose an approach to effectively refine the poses and train the neural field for this challenging scenario.

4. Method

This work addresses the challenge of novel view synthesis based on neural implicit representations, in the sparse-view regime. In particular, we assume access to only *sparse input views* with *noisy camera pose estimates*. The training image collection contains few images (as low as 3) and they present large viewpoint variations.

This leads to two major challenges: (i) given only few input images, the NeRF model [30] instantly overfits to the training views without learning a meaningful 3D geometry, even with perfect input camera poses [19, 22, 32]. As shown in Fig. 1, this leads to degenerate novel view renderings, including for similar train/test viewing directions. The problem becomes amplified when considering noisy input camera poses. (ii) Previous pose-NeRF refinement approaches [5,9,24,50,52] were designed considering a dense coverage of the 3D space, *i.e.* many input views. They apply their training objectives, *e.g.* the photometric loss (7), on each training image *independently*. However, in the sparse-view regime, *i.e.* where the 3D space is under-constrained, such supervision is often too weak for the pose/NeRF system to converge to a *globally consistent* geometric solution. Failure to correctly register the training poses also leads to poor novel view rendering quality (see Fig. 1, 4).

We propose SPARF, a simple, yet effective training strategy to jointly learn the scene representation and refine the initial training poses, tailored for the sparse-view scenario. As the prominent source of inspiration, we draw from well-established principles of multi-view geometry [17], which we adapt to the NeRF framework. In Sec. 4.1, we introduce our *multi-view correspondence objective* as the main driving signal for the joint pose-NeRF training. By relying on pixel correspondences between the training views, the loss enforces convergence to a global and accurate geometric solution consistent across all training views, thereby solving both challenges (i) and (ii). Moreover, in Sec. 4.2 we propose an additional term, *i.e.* the *depth consistency loss*,

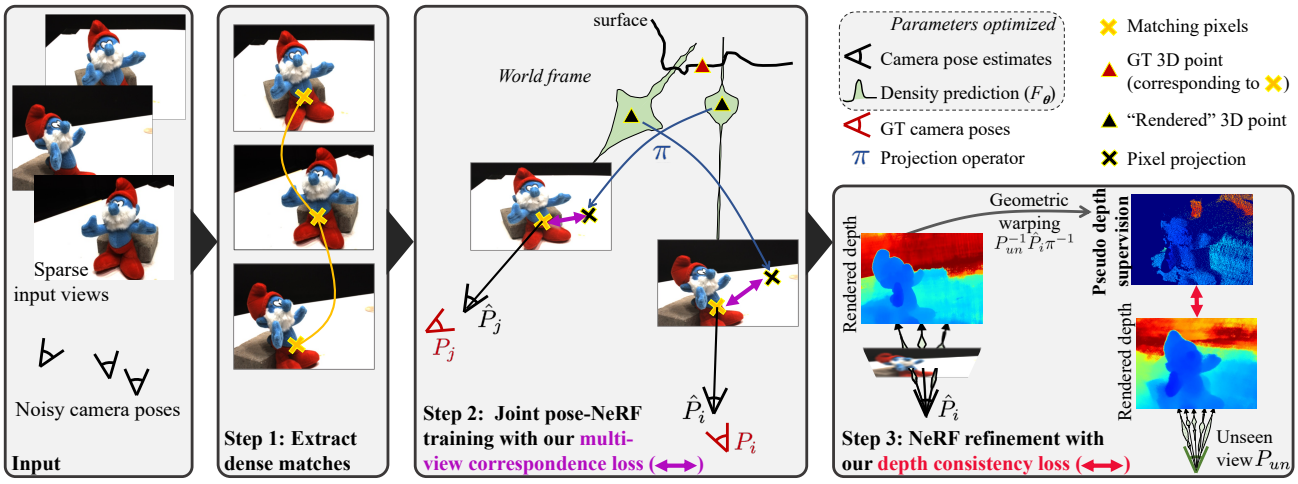

Figure 2. Our approach **SPARF** for joint pose-NeRF training given only *few input images* with *noisy camera pose* estimates. We first rely on a pre-trained dense correspondence network [44] to extract matches between the training views. Our multi-view correspondence loss (Sec. 4.1) minimizes the re-projection error between matches, *i.e.* it enforces each pixel of a particular training view to project to its matching pixel in another training view. We use the rendered NeRF depth (6) and the current pose estimates $\hat{\mathcal{P}}$ to backproject each pixel in 3D space. This constraint hence encourages the learned scene and pose estimates to converge to a global and accurate geometric solution, consistent across all training views. Our depth consistency loss (Sec. 4.2) further uses the rendered depths from the training viewpoints to create pseudo-depth supervision for unseen viewpoints, thereby encouraging the reconstructed scene to be consistent from any direction.

which encourages the learned scene geometry to be consistent across *all viewpoints*, including those for which no RGB supervision is available. In doing so, it boosts novel-view rendering quality, further tackling the overfitting problem (i). We present our final training strategy in Sec. 4.3 and visualize our approach in Fig. 2.

4.1. Multi-View Correspondence Loss

Directly overfitting on the training images leads to a corrupted neural radiance field collapsing towards the provided views, even when assuming perfect camera poses [11, 19, 32]. With noisy input poses, the problem becomes amplified, making it impossible to use the photometric loss (7) as the main signal for the joint pose-NeRF training. We propose a training objective, the multi-view correspondence loss, to enforce learning a *globally consistent 3D solution* over the optimized scene geometry and camera poses.

Multi-view geometry constraint: We draw inspiration from principles of multi-view geometry [17]. We assume that given an image pair (I_i, I_j), we can obtain pairs of matching pixels $\mathbf{p} \in I_i$ and $\mathbf{q} \in I_j$. We then compute estimates of the depth at both pixels $\hat{z}_{i,\text{p}} = \hat{z}(\mathbf{p}; \theta, \hat{P}_i)$ and $\hat{z}_{j,\text{q}} = \hat{z}(\mathbf{q}; \theta, \hat{P}_j)$ according to eq. (6). Principles of multi-view geometry dictate that both pixels must backproject to the same 3D point in the world coordinate system. This is formulated as $\hat{P}_j \pi^{-1}(\mathbf{q}, \hat{z}_{j,\text{q}}) = \hat{P}_i \pi^{-1}(\mathbf{p}, \hat{z}_{i,\text{p}})$. However, when translating this constraint into a training objective, the magnitude of the loss is subject to large variations depending on the scene scale and the initial camera poses, requiring a tedious tuning of the loss weighting.

Training objective: We instead project the 3D points back to image space, therefore minimizing the distance between pixels rather than directly in 3D space. We illustrate this objective in Fig. 2 (steps 1-2). For a randomly sampled training image pair (I_i, I_j), our multi-view correspondence objective is formulated as $\mathcal{L}_{\text{MVCorr}}(\theta, \hat{\mathcal{P}}) = \sum_{\text{p} \in \mathcal{V}} \mathcal{L}_\text{p}$, where

$$\mathcal{L}_\text{p} = w_\text{p} \rho \left(\mathbf{q} - \pi \left(\hat{P}_j^{-1} \hat{P}_i \ \pi^{-1}(\mathbf{p}, \hat{z}(\mathbf{p}; \theta, \hat{P}_i)) \right) \right) . \quad (8)$$

Here ρ denotes the Huber loss function [18] and $w_\text{p} \in [0, 1]$ is the confidence associated with the correspondence $(\mathbf{p}, \mathbf{q})$, which we obtain as detailed below. We additionally define the set $\mathcal{V} = \{\mathbf{p} : w_\text{p} \geq \kappa\}$, where $\kappa = 0.95$. The homogenization operations were omitted for clarity.

Our loss serves two purposes. By connecting the training images through correspondences, our multi-view correspondence objective enforces the learned geometry and camera poses to converge to a solution geometrically consistent across all training images. This is unlike the photometric loss (7) which applies supervision on each training image independently. Moreover, the underlying constraint is only satisfied if the learned 3D points converge to the true reconstructed scene (up to a similarity). As such, the objective (8) provides direct supervision on the rendered depth (6), implicitly enforcing it to be close to the surface.

Correspondence prediction: Any classical [27, 35] or learned [13, 36, 43, 45, 46] matching approach could be used to obtain the matches relating pairs of training views. We rely on a pre-trained dense correspondence regression network, in particular PDC-Net [44]. It predicts a match $\mathbf{q}$ for each pixel $\mathbf{p}$, along with a confidence w_p. We found

the high number of accurate matches to be beneficial for our joint pose-NeRF refinement. Similar conclusions were derived in the context of dense versus sparse depth supervision [11,34]. The dense correspondence map also implicitly imposes a smoothness prior to the rendered depth. In suppl., we present results using a sparse matcher [12, 36] instead.

4.2. Improving Geometry at Unobserved Views

The multi-view correspondence loss favors a global and geometrically accurate solution, consistent across all training images. Nevertheless, the reconstructed scene often still suffers from inconsistencies when seen from novel viewpoints. For those, no RGB supervision is available during training. We propose an additional training objective, the depth consistency loss, which encourages the learned geometry to be consistent from any viewing direction.

Depth consistency loss: The main idea is to use the depth maps rendered from the training viewpoints to create pseudo-depth supervision for novel, unseen, viewpoints (Fig. 2, step 3). We sample a virtual pose P_{un}, in practice obtained as an interpolation between the poses of two close-by training views. For a pixel $\mathbf{p}$ in a sampled training image I_i, $\mathbf{r}_{\text{p}}^{un} = P_{un}^{-1}\hat{P}_i\ \pi^{-1}(\mathbf{p}, \hat{z}(\mathbf{p};\theta,\hat{P}_i))$ is the corresponding 3D point in the coordinate system of the unseen view P_{un}. $\mathbf{y} \in \mathbb{R}^2$ denotes its pixel projection in view P_{un} as $\mathbf{y} = \pi\left(\mathbf{r}_{\text{p}}^{un}\right)$, and z_{y} is its projected depth in P_{un}, *i.e.* $z_{\text{y}} = \left[\mathbf{r}_{\text{p}}^{un}\right]_3$, where $[\cdot]_3$ refers to taking the third coordinate of the vector. We formulate our depth consistency loss as,

$$\mathcal{L}_{\text{DCons}}(\theta) = \sum_{\text{p}} \gamma_{\text{y}}\, \rho\left(z_{\text{y}} - \hat{z}(\mathbf{y};\theta,P_{un})\right)\,. \tag{9}$$

To account for occlusion and out-of-view projections in which (9) is invalid, we have included a visibility mask $\gamma_{\text{y}} \in [0,1]$. We explain its definition in the section below.

Since the pseudo-depth supervision z_{y} is created from renderings, it is subject to errors. For this reason, we find it important to backpropagate through the pseudo-supervision $\mathbf{y}$ and z_{y}. Note that we however do not backpropagate through the pose estimate $\hat{P}_i$. Moreover, as verified experimentally in Tab. 2, our depth consistency objective (9) is complementary to our multi-view correspondence loss (8), the latter enforcing an *accurate* reconstructed geometry while the former ensures it is *consistent* from any viewpoint.

Visibility mask γ_{y}: We first exclude points if their pixel projections $\mathbf{y}$ is outside of the virtual view, by setting the mask as $\gamma_{\text{y}} = 0$. The depth consistency loss is also invalid for pixels that are occluded by the reconstructed scene in the virtual view. To identify these occluded pixels, we follow the strategy of [10]. In particular, we check whether there are occupied regions on the ray between the camera center $\mathbf{o}_{un}$ of P_{un} and the 3D point $\mathbf{r}_{un,\text{y}}(z_{\text{y}})$ at depth z_{y}. We compute how occluded a 3D point is with its transmittance (5) in the unseen view, as $\gamma_{\text{y}} = T_{un,z_{\text{y}}}$. Intuitively, γ_{y} is close to 1 if there is no point with a large density between the camera center $\mathbf{o}_{un}$ and $\mathbf{r}_{un,\text{y}}(z_{\text{y}})$, otherwise it is close to 0. Next, we present our overall training framework.

4.3. Training Framework

Staged training: Our final training objective is formulated as $\mathcal{L}(\theta,\hat{\mathcal{P}}) = \mathcal{L}_{\text{photo}}(\theta,\hat{\mathcal{P}}) + \lambda_{\text{c}}\mathcal{L}_{\text{MVCorr}}(\theta,\hat{\mathcal{P}}) + \lambda_{\text{d}}\mathcal{L}_{\text{DCons}}(\theta)$, where λ_{c} and λ_{d} are predefined weighting factors. The training is split into two stages. In the first part, the pose estimates are trained jointly with the coarse MLP F_θ^c. However, due to the exploration of the pose space at the early stages of training, the learned scene tends to showcase blurry surfaces. As a result, in the second training stage, we freeze the pose estimates and train both the coarse and fine networks F_θ^c and F_θ^f. This ensures that the fine network learns a sharp geometry, benefiting from the pre-trained coarse network. From a practical perspective, our training objectives can be integrated at a low computational cost, since the RGB or depth pixel renderings (3)-(6) can be shared between the three loss terms.

Coarse-to-fine positional encoding: In BARF [24], Lin *et al*. propose to gradually activate the high-frequency components of the positional encodings (2) over the course of the optimization. We refer the reader to [24] for the exact formulation. While originally proposed in the context of pose refinement, we found that this strategy is also extremely beneficial in the sparse-view setting, even when the poses are fixed. It prevents the network from immediately overfitting to the training images, thereby avoiding the worst degenerate geometries. We therefore adopt this coarse-to-fine positional encoding approach as default.

5. Experimental Results

We evaluate the proposed SPARF for novel-view rendering in the few-view setting, in particular when only three input views are available. Results with different numbers of views are provided in suppl. We extensively analyze our method and compare it to earlier approaches, setting a new state of the art on multiple datasets. Further results, visualizations, and implementation details are provided in suppl.

5.1. Experimental Settings

Datasets and metrics: We report results on the **DTU** [20], **LLFF** [39] and **Replica** [40] datasets, for the challenging scenario of 3 input views. DTU is composed of complex object-level scenes with wide-baseline views spanning a half hemisphere. We adhere to the protocol of [56] and evaluate on their reported test split of 15 scenes. Following [32], we additionally evaluate all methods with the object masks applied to the rendered images, to avoid penalizing methods for incorrect background predictions. On LLFF, we follow community standards [30] and use every 8$^{\text{th}}$ image as

	Method	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
I	Full PE [30]	8.41 (9.34)	0.31 (0.63)	0.71 (0.36)	0.87
II	Smaller MLP model	9.03 (10.06)	0.34 (0.65)	0.68 (0.34)	0.79
III	No PE	16.11 (18.40)	0.68 (0.80)	0.37 (0.24)	**0.30**
IV	CF PE [24] (Sec. 4.3)	**16.27** (**18.41**)	**0.69** (**0.81**)	**0.29** (**0.14**)	0.39

Table 1. Comparison of different positional encoding strategies applied to NeRF [30] on DTU (3 views), using ground-truth poses. Results in (·) are computed by masking the background.

the test set. We sample the training views evenly from the remaining images. For the Replica dataset, which depicts videos of room-scale indoor scenes, we subsample every k^{th} frame, from which we randomly select a triplet of consecutive training images. As metrics, we report the average rotation and translation errors for pose registration, and PSNR, SSIM [49] and LPIPS [60] for view synthesis. On the DTU and Replica datasets, we additionally compare the rendered depth with the available ground-truth depth and compute the mean depth absolute error (DE).

Implementation details: We train our approach for 100K iterations, which takes about 10 hours on a single A100 GPU. As pose parametrization, we adopt the continuous 6-vector representation [62] for the rotation and directly optimize the translation vector. We provide all training hyperparameters in the supplementary.

5.2. Method Analysis

We first perform a comprehensive analysis of our approach, on DTU [20], considering only 3 input views.

Impact of positional encoding: Training on sparse input views using the standard NeRF [30] immediately overfits to the provided images, even with perfect poses. We noticed that the overfitting is largely due to the high-frequency positional encodings (PE), and thus experimented with different PE strategies. We present the results in Tab. 1. The standard NeRF (I) with high-frequency PE [30] leads to degenerate geometry and novel view renderings. In (II), using a simplified MLP makes little difference. While training without PE (III) largely prevents overfitting, the coarse-to-fine PE strategy [24] leads to the best result, as shown in (IV).

Ablation study: In Tab. 2, we ablate the key components of our approach, here assuming fixed ground-truth poses and starting from NeRF with coarse-to-fine PE. Adding our multi-view correspondence loss (8) results in drastically better performance on all metrics. Including our depth-consistency module (9) further leads to a small improvement, achieving the best performance overall. Also note that our depth-consistency module (9) works best in collaboration with our multi-view correspondences loss (8) since the latter is needed to learn an accurate geometry.

MV-Corr (8)	DCons (9)	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
✗	✗	16.27 (18.41)	0.69 (0.81)	0.29 (0.14)	0.39
✗	✓	15.86 (18.91)	0.71 (0.82)	0.28 (0.14)	0.20
✓	✗	18.13 (20.81)	0.77 (**0.87**)	0.22 (**0.10**)	0.10
✓	✓	**18.30** (**21.01**)	**0.78** (**0.87**)	**0.21** (**0.10**)	**0.08**

Table 2. Ablation study 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 [24].

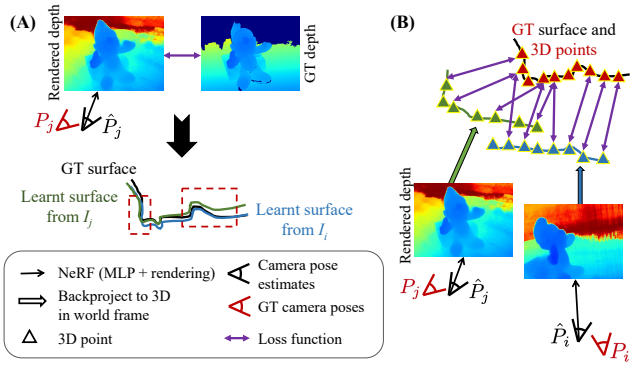

Figure 3. We compare two training objectives using *ground-truth depth* for pose-NeRF training in the sparse-view regime. In (A), a loss comparing for each training image the rendered depth (6) with the ground truth one, can learn locally perfect geometry (as highlighted by the dashed red rectangles). However, the NeRF/poses do not converge to a global solution, because the optimized poses and geometry of the different images are disjoint. Instead, supervising the learned 3D points of each training image to be equal to the ground-truth 3D points in (B) solves this issue, by enforcing the system to converge to a global (unique) geometric solution.

Intuition on pose-NeRF training losses: We first want to build an intuition on what loss might be suitable for joint pose-NeRF training in the *sparse regime*. To do so, we use ground-truth depth or 3D data in two alternative training losses, which we compare here. We illustrate this experiment in Fig. 3 and present results in Tab. 3, top part. As in previous work [24], for each scene of DTU [20], we synthetically perturb the ground-truth camera poses with 15% of additive gaussian noise. In (I), we train with an L1 loss comparing the rendered depth (6) with the ground-truth depth (Fig. 3A). Surprisingly, this loss struggles to refine the poses. Instead, in (II) we minimize the distance between the *learned 3D points* (rendered depth (6) backprojected to world frame) and the ground-truth 3D points, as illustrated in Fig. 3B. This training loss successfully registers the poses, resulting in drastically better novel-view rendering quality. As the main insight from this experiment, we hypothesize that, in the sparse-view regime, it is crucial to enforce an explicit geometric connection between the different training images and their underlying scene geometry. This is not the case in (I), where the depth loss favors per-image locally accurate geometry, but the NeRF/poses can converge to disconnected solutions for each training image.

Comparison of losses for pose-NeRF training: In Tab. 3 bottom part, we then compare our loss (8) to objectives commonly used for joint pose-NeRF training. The photometric loss (7) (III), even associated with a mask/silhouette

	Losses	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE↓
I	Photo. + L1 **GT** depth	7.3	28.9	13.8 (14.0)	0.54 (0.70)	0.46 (0.27)	0.17
II	Photo. + L1 **GT** 3D points	**0.4**	**1.5**	**18.8 (20.3)**	**0.75 (0.84)**	**0.21 (0.11)**	**0.07**
III	Photo. (7)	10.3	51.5	10.7 (9.8)	0.43 (0.62)	0.59 (0.36)	1.9
IV	Photo. + mask loss [23, 59]	13.2	57.7	-	-	-	-
V	MVCorr (8)	1.98	6.6	-	-	-	0.19
VI	**Photo. + MVCorr** ((7)-(8))	**1.85**	**5.5**	**16.0 (17.8)**	**0.68 (0.81)**	**0.28 (0.14)**	**0.13**

Table 3. Comparison of training objectives for joint pose-NeRF refinement on DTU [20] with initial noisy poses (3 views). Rotation errors are in degree and translation errors are multiplied by 100. Results in (·) are computed by masking the background.

loss [5, 23, 59] in (IV), completely fails to register the poses, thus leading to poor novel-view synthesis performance. This is in line with our hypothesis that it is important to explicitly exploit the *geometric relation* between the training views for successful registration. Moreover, because the 3D space is under-constrained in the sparse-view regime, multiple neighboring poses can lead to similar mask losses. While our multi-view correspondence loss (8) alone (V) already drastically outperforms the photometric loss (III) in terms of pose and learned geometry (depth error), combining the two in (VI) leads to the best performance. This is because, through the correspondences, our approach favors a NeRF/pose solution consistent across all training images. Note that this version neither includes our depth consistency loss (9) (Sec. 4.2) nor our staged training (Sec. 4.3).

5.3. Comparison to SOTA with Noisy Poses

Here, we evaluate SPARF, our joint pose and NeRF training approach. Results with different pose initialization schemes are presented in the supplementary.

Baselines: We compare to BARF [24], the state-of-the-art in pose-NeRF refinement when assuming dense input views. It is representative of a line of approaches [9, 24, 29, 50, 52] using the photometric loss (7) as the main signal. We also experiment with adding the depth regularization loss of [32] or the ray sparsity loss of [4] to BARF, which we denote as RegBARF and DistBARF respectively. We additionally compare to SCNeRF [21], which uses a geometric loss based on correspondences, minimizing the rays' intersection re-projection error. For a fair comparison, we integrate coarse-to-fine PE [24] (Sec. 4.3) in all methods.

Results on DTU: Following [9, 24, 52], for each scene, we synthetically perturb the ground-truth camera poses with 15% of additive gaussian noise. The initial poses thus have

Method	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
BARF [24]	10.33	51.5	10.71 (9.76)	0.43 (0.62)	0.59 (0.36)	1.90
RegBARF [24, 32]	11.20	52.8	10.38 (9.20)	0.45 (0.62)	0.61 (0.38)	2.33
DistBARF [4, 24]	11.69	55.7	9.50 (9.15)	0.34 (0.76)	0.67 (0.36)	1.90
SCNeRF [21]	3.44	16.4	12.04 (11.71)	0.45 (0.66)	0.52 (0.30)	0.85
SPARF (Ours)	**1.81**	**5.0**	**17.74 (18.92)**	**0.71 (0.83)**	**0.26 (0.13)**	**0.12**

Table 4. Evaluation on DTU [20] (3 views) with noisy initial poses. Rotation errors are in ° and translation errors are multiplied by 100. Results in (·) are computed by masking the background.

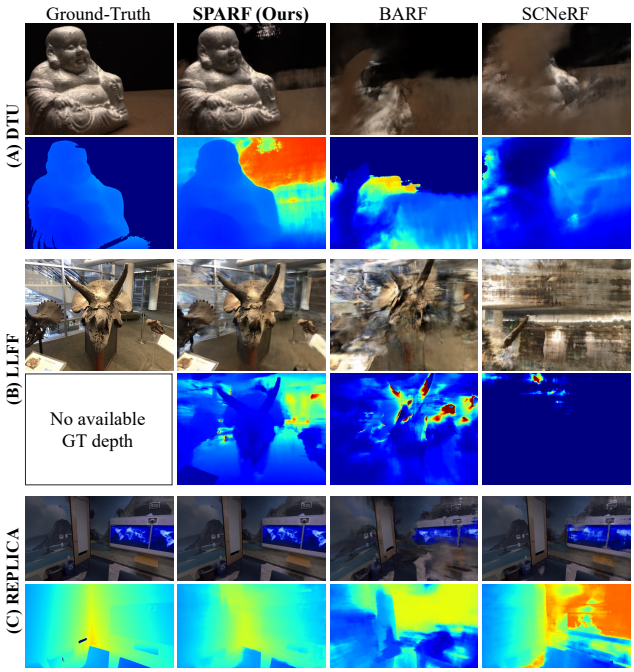

Figure 4. Novel-view rendering (RGB and depth). The input (not shown here) contains 3 images with initial noisy camera poses.

an average rotation and translation error of 15° and 70 respectively. We show initial and optimized poses in Fig. 5. From the results in Tab. 4 and Fig. 4A, we observe that BARF, RegBARF, and DistBARF completely fail to register the poses, leading to poor view-synthesis quality. SCNeRF's geometric loss performs better at registering the poses but the learned scene still suffers from many inconsistencies. This is because SCNeRF's loss [21] does not influence the learned radiance field function, and thus, cannot prevent the NeRF model from overfitting to the sparse input views. Since our multi-view correspondence loss (8) acts on *both* the camera pose estimates and the learned neural field by enforcing them to fit the correspondence constraint, it leads to an accurate reconstructed scene. Our approach SPARF hence significantly outperforms all others both in novel-view rendering quality and pose registration.

Results on LLFF: The LLFF dataset consists of 8 complex forward-facing scenes. Following [24], we initialize all camera poses with the *identity* transformation and present results in Tab. 5. In [24], Lin *et al.* show that BARF almost perfectly registers the camera poses given *dense input views*. However, we show here that it strug-

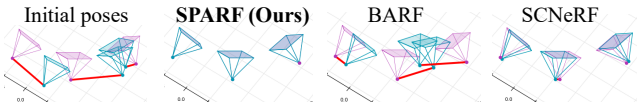

Figure 5. Optimized poses on DTU with 3 input views. We compare the ground-truth poses (in pink) with the optimized ones (in blue). In the first column, the initial noisy poses are in blue.

	Rot. (°) ↓	Trans. (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BARF [24]	2.04	11.6	17.47	0.48	0.37
RegBARF [24, 32]	1.52	5.0	18.57	0.52	0.36
DistBARF [4, 24]	5.59	26.5	14.69	0.34	0.49
SCNeRF [21]	1.93	11.4	17.10	0.45	0.40
SPARF (Ours)	**0.53**	**2.8**	**19.58**	**0.61**	**0.31**

Table 5. Evaluation on the forward-facing dataset LLFF [39] (3 views) starting from initial identity poses.

	Method	Rot (°) ↓	Trans (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
G	**SPARF (Ours)**	Fixed GT poses		*26.43*	**0.88**	**0.13**	*0.39*
F	NeRF [30]	Fixed poses obtained		20.99	0.73	0.32	1.33
	DS-NeRF [11]	from COLMAP (run w.		23.52	0.81	0.20	0.99
	SPARF (Ours)	PDC-Net [44] matches)		25.03	*0.84*	*0.15*	0.66
R	BARF [24]	3.35	16.96	20.73	0.72	0.30	0.84
	RegBARF [24, 32]	3.66	20.87	20.00	0.70	0.32	1.00
	DistBARF [4, 24]	2.36	7.73	22.46	0.77	0.23	0.47
	SCNeRF [21]	*0.65*	*4.12*	22.54	0.79	0.24	0.73
	DS-NeRF [11]	1.30	5.04	24.75	0.83	0.20	0.69
	SPARF (Ours)	**0.15**	**0.76**	**26.98**	**0.88**	**0.13**	**0.36**

Table 6. Evaluation on Replica [40] (3 views) with initial poses obtained by COLMAP [38, 44]. The initial rotation and translation errors are 0.39° and 3.01 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. The best and second-best results are in red and blue respectively.

gles in the sparse-view setting, thereby severely impacting the accuracy of novel view synthesis. While adding the depth smoothness loss (RegBARF) improves results, our approach SPARF outperforms all previous works. A qualitative comparison is shown in Fig. 4B.

Results on Replica: To demonstrate that our approach is also applicable to non-forward-facing indoor scenes, we evaluate on the Replica dataset in Tab. 6 and Fig. 4C. As pose initialization, we use COLMAP [38] with improved matches, *i.e.* using PDC-Net [44]. The initial pose estimates thus have an average rotation and translation error of respectively 0.39° and 3.01. Comparing the top (G) and middle part (F) of Tab. 6, we show that even such a low initial error impacts the novel-view rendering quality when using fixed poses. In the bottom part (R), our pose-NeRF training strategy leads to the best results, matching the accuracy obtained by our approach with perfect poses (top row, G).

5.4. Comparison to SOTA with Ground-Truth Poses

Finally, we show that our approach brings significant improvement in novel view rendering quality even when considering *fixed ground-truth* poses.

Baselines: We compare to works specifically designed to tackle per-scene few-shot novel view rendering, namely DietNeRF [19], DS-NeRF [11], InfoNeRF [22] and RegNeRF [32], along with the standard NeRF [30] and MipNeRF [3]. For completeness, we also compare against a state-of-the-art conditional model, PixelNeRF [56], trained on DTU [20] and further finetuned per-scene on LLFF [39].

Method	DTU PSNR ↑	DTU SSIM ↑	DTU LPIPS ↓	LLFF PSNR ↑	LLFF SSIM ↑	LLFF LPIPS ↓
PixelNeRF [56]	**19.36** (18.00)	*0.70* (*0.77*)	*0.32* (0.23)	7.93	0.27	0.68
PixelNeRF-ft [56]	-	-	-	16.17	0.44	0.51
MipNeRF [3]	7.64 (8.68)	0.23 (0.57)	0.66 (0.35)	14.62	0.35	0.50
NeRF [30]	8.41 (9.34)	0.31 (0.63)	0.71 (0.36)	13.61	0.28	0.56
DietNeRF [19]	10.01 (11.85)	0.35 (0.63)	0.57 (0.31)	14.94	0.37	0.5
InfoNeRF [22]	11.23 (-)	0.44 (-)	0.54 (-)	-	-	-
RegNeRF [32]	15.33 (*18.89*)	0.62 (0.75)	0.34 (*0.19*)	*19.08*	*0.59*	0.34
DS-NeRF [11]	16.52 (-)	0.54 (-)	0.48 (-)	18.00	0.55	*0.27*
SPARF (Ours)	*18.30* (**21.01**)	**0.78** (**0.87**)	**0.21** (**0.10**)	**20.20**	**0.63**	**0.24**

Table 7. Evaluation on DTU [20] and LLFF [39] (3 views), with fixed ground-truth poses. Results in (·) are computed by masking the background. Results of [3, 19, 32] are taken from [32]. The best and second-best results are in red and blue respectively.

Results: We present results on DTU and LLFF in Tab. 7. Compared to previous per-scene approaches [19, 22, 32] that only apply different regularization to the learned scene, our multi-view correspondence loss (8) provides a strong supervision on the rendered depth, implicitly encouraging it to be close to the true surface. Our depth consistency objective (9) further boosts the performance, by directly enforcing the learned scene to be consistent from any viewpoint. As a result, our approach SPARF performs best compared to all baselines on both datasets and for all metrics. The only exception is PSNR on the whole image compared to conditional model PixelNeRF [56]. This is because DTU has black backgrounds, where a wrong color prediction (like in Fig. 4A for SPARF) has a large impact on the PSNR value. For conditional models which rely on feature projections, it is easier to predict a correct background color. However, most real-world applications are more interested in accurately reconstructing the object of interest than the background. When evaluated only in the object region, our SPARF obtains 3.24dB higher PSNR than PixelNeRF.

6. Conclusion

We propose SPARF, a joint pose-NeRF training strategy capable of producing realistic novel-view renderings given few wide-baseline input images with noisy camera pose estimates. By integrating two novel objectives inspired by multi-view geometry principles, we set a new state of the art on three challenging datasets.

Limitations and future work: Our approach is only applicable to input image collections where each image has covisible regions with at least one other. Moreover, the performance of our method depends on the quality of the matching network. Filtering strategies or per-scene online refinement of the correspondence network thus appear as promising future directions. An interesting direction is also to refine the camera intrinsics and distortion parameters along with the extrinsics. Finally, using voxel grids to encode the radiance field [31] instead of an MLP could lead to faster convergence, and potentially even better results.

Appendix

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 [30] 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 [24] 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.

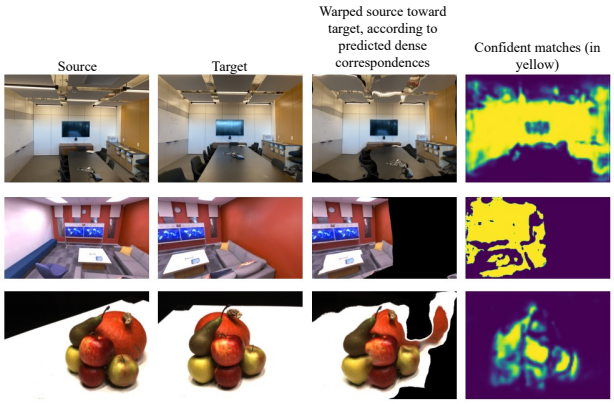

Figure 6. Dense matches and associated confidence predicted by PDC-Net [44] on pair examples of the LLFF, DTU, and Replica datasets. PDC-Net predicts the dense correspondences relating the target to the source. In the 3rd column, we show the source (1^{st} column) warped towards the target (2^{nd} column), 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.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 [44]. 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 [44], 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 [44]. We show examples of dense matches estimated by PDC-Net in Fig. 6.

Matches selection: We only apply the multi-view correspondence loss (Sec. 4.1) 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.

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 pa-

per, 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 (7). We also sample an image pair and apply the multi-view correspondence loss (Sec. 4.1) on a random subset of 1024 matches. For the depth consistency loss (Sec. 4.1), 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 [24] (Sec. 4.3). 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. We set the weights λ_c and λ_d associated with respectively the multi-view correspondence loss (Sec. 4.1) and the depth consistency loss (Sec. 4.2) to $\lambda_\text{c} = 10^{-3}$ and $\lambda_\text{d} = 10^{-3}$.

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 (7) 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 $\lambda_\text{c} = 10^{-4}$ and $\lambda_\text{d} = 10^{-3}$ 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 [24], 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 [21], we adopt the 6-vector representation [62]. 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 col-

umn 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 [62]. 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,

$$R = f\left(\begin{bmatrix} | \\ \mathbf{r} \\ | \end{bmatrix}\right) = \begin{bmatrix} | & | & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \\ | & | & | \end{bmatrix}, \tag{10}$$

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,

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

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 (7).

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 (7).

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 (7) 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 (7).

COLMAP: We run COLMAP [38] 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 [37], which we modify to fit our needs. We try to use SuperPoint [12] and SuperGlue [36], which has become the defacto state-of-the-art sparse matching approach. As an alternative, we also use PDC-Net matches [44], 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 [44] 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 [21]), 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 [44] correspondences, *i.e*. the same matches that we rely on in our multi-view correspondence loss (Sec. 4.1).

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), we follow community standards [30] and use every 8^{th} image as the test set. We sample the training views evenly from the remaining images.

DTU: Following previous works [11, 32], 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 [32], we additionally evaluate all methods with the object masks applied to the rendered images. The object masks are obtained from [32, 54]. 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 [24, 50] is to align the two sets of pose trajectories (optimized and ground-truth) globally with a Sim(3) transformation using Umeyama algorithm [47] in an ATE toolbox [61]. 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 [47].

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

$$R_{err} = cos^{-1} \frac{Tr(R^{-1}\hat{R}) - 1}{2} , \tag{11}$$

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 [24, 50, 55] 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 [38], 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) [49], and the Learned Perceptual Image Patch similarity (LPIPs) metric [60], which estimates the distance between an image pair in a learned feature space.

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.

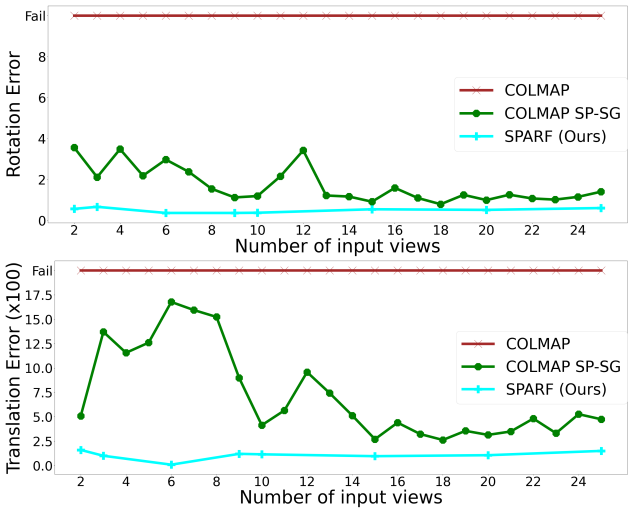

Figure 7. Rotation and translation errors versus the number of input views on a scene of the DTU dataset [20]. The standard COLMAP [38] fails to estimate initial poses for each number of input views, including relatively high numbers (> 20). Failing in that case means that COLMAP does not find a pose for at least one image of the set. COLMAP with better sparse matches (SuperPoint and SuperGlue [12, 36]) 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.

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 [38] 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.

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

Here, we analyze the performance of COLMAP [38] for different numbers of input views. In Fig. 7, 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 [12, 36] (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.

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), without our depth consis-

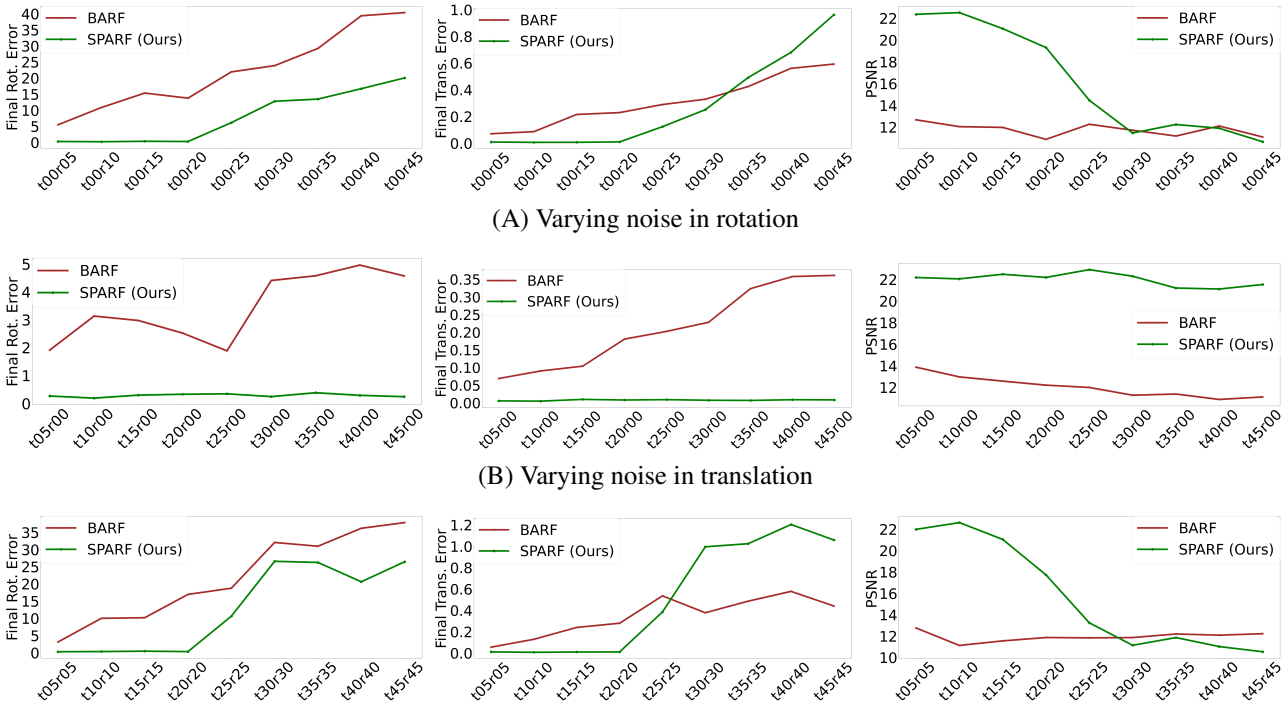

Figure 8. Pose registration error and PSNR obtained by BARF and our SPARF for different levels of initial noise. This experiment is performed on one scene of the DTU dataset, considering 3 input views. Rotation errors are in degree and translation errors are multiplied by 100. Results of PSNR (↑) are computed by masking the background.

tency loss (Sec. 4.2) nor our staged training (Sec. 4.3). 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. 8. We investigate 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 [24]. 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.

Figure 9. 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.

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. 9, 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 [44] does not predict any reliable matches. Note that the alternative matching approach SuperPoint-SuperGlue [12, 36] is also unable to extract correspondences in that case.

C.3. Impact of different correspondences

Our multi-view correspondence loss (8) (Sec. 4.1) 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 [44], any hand-crafted or learned matching network could be used. We here compare using the dense correspondence regression network PDC-Net [44] with the state-of-the-art sparse matcher SuperGlue [36]. In

	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 [44]	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 8. Performance of our joint pose-NeRF training, when using different pre-trained correspondence networks. The results are computed on DTU [20] 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), without including our depth consistency objective (Sec. 4.2) nor our staged training (Sec. 4.3). 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 [21], we use PDC-Net [44] correspondences.

	Rot. (°) ↓	Trans. (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BARF [24]	2.04	11.6	17.47	0.48	0.37
SCNeRF [21]	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 9. Performance of our joint pose-NeRF training, when using different pre-trained correspondence networks. As in Tab. 8 for DTU, the evaluation is here performed on the forward-facing dataset LLFF [39] (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), without including our depth consistency objective (Sec. 4.2) nor our staged training (Sec. 4.3).

combination with the SuperGlue matcher, we use the SuperPoint [12] detector and descriptor.

In Tab. 8, we present results on DTU, of our joint pose-NeRF refinement approach, trained using the multi-view correspondence objective (Sec. 4.1) with these two alternative matching methods. As a reference, we also include results of BARF [24] and SCNeRF [21]. 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 (8) 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. 9, we present the same comparison, on the LLFF dataset. Using PDC-Net or SP-SG matches results in a similar performance.

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. 10. 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

		Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
I	Photo. (7)	10.3	51.5	10.7 (9.8)	0.43 (0.62)	0.59 (0.36)	1.9
II	+ MVCorr (8)	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 (9)	**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)	**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 10. Ablation study on DTU [20] (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) is then not included. Rotation errors are in degree and translation errors are multiplied by 100. Results in $(\cdot)$ are computed by masking the background.

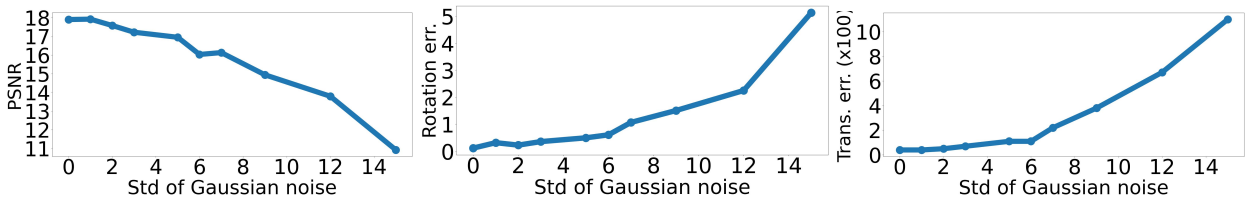

Figure 10. Evaluation of SPARF* on one scene of DTU, with different levels of Gaussian noise added to ground-truth image matches. Here, SPARF* indicates that we only use the combination of the photometric loss with our multi-view correspondence objective (Sec. 4.1), without including our depth consistency objective (Sec. 4.2) nor our staged training (Sec. 4.3).

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. 10. From (I) to (II), adding our multi-view correspondence loss (8) leads to drastically better pose registration than training with only the photometric loss (7) (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), 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) slightly improves the rendering performance, leading to the best results overall.

Comparison of different training schedules: In the bottom part of Tab. 10, 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) 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 re-initialized. 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 [50].

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 [10]. We ablate the impact of this visibility mask in the depth consistency loss formulation in Tab. 11. 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 [20]. In the main paper, we showed results when considering three input views and starting from initial

	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
MVCorr (8) of m.p.	18.13 (20.81)	0.77 (**0.87**)	0.22 (**0.10**)	0.10
+ DCons (9) of m.p.	**18.30** (**21.01**)	**0.78** (**0.87**)	**0.21** (**0.10**)	**0.08**
No Vis mask (Sec. 4.2)	17.89 (21.05)	0.77 (0.86)	0.22 (0.11)	0.09

Table 11. 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 [24].

		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 [30]	Fixed poses obtained		8.95 (9.77)	0.30 (0.60)	0.72 (0.38)	1.25	Fixed poses obtained		8.88 (9.66)	0.31 (0.62)	0.73 (0.37)	1.28
	DS-NeRF [11]	from COLMAP (run w.		11.89 (13.28)	0.46 (0.69)	0.49 (0.25)	0.38	from COLMAP (run w.		11.61 (12.81)	0.46 (0.70)	0.51 (0.25)	0.60
	DS-NeRF w. CF PE [11, 24]	SP-SG [36] matches)		16.58 (17.58)	0.66 (0.77)	0.29 (0.17)	0.21	PDC-Net [44] matches)		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 [24]	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 [24, 32]	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 [21]	*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 [11]	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 12. Evaluation on 14 scenes of the DTU dataset (3 views) with initial poses obtained by COLMAP using SP-SG [36] (left) or PDCNet [44] (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 [24] (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 [24]	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 [24, 32]	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 [4, 24]	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 [21]	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 13. Evaluation on DTU [20] 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.

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 [38]. 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 [38] mostly fails when reducing the number of input views to 3 (see Fig. 7). As a result, to obtain the initial camera pose estimates, we experiment with COLMAP run with matches predicted by SuperPoint and SuperGlue [36] (SP-SG) or PDC-Net [44]. 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. 9). We thus present results on the remaining 14 scenes in Tab. 12. 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.

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 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. 10).

Results with 6 and 9 views: In Tab. 4 of the main paper, we evaluate our proposed approach SPARF for joint pose-

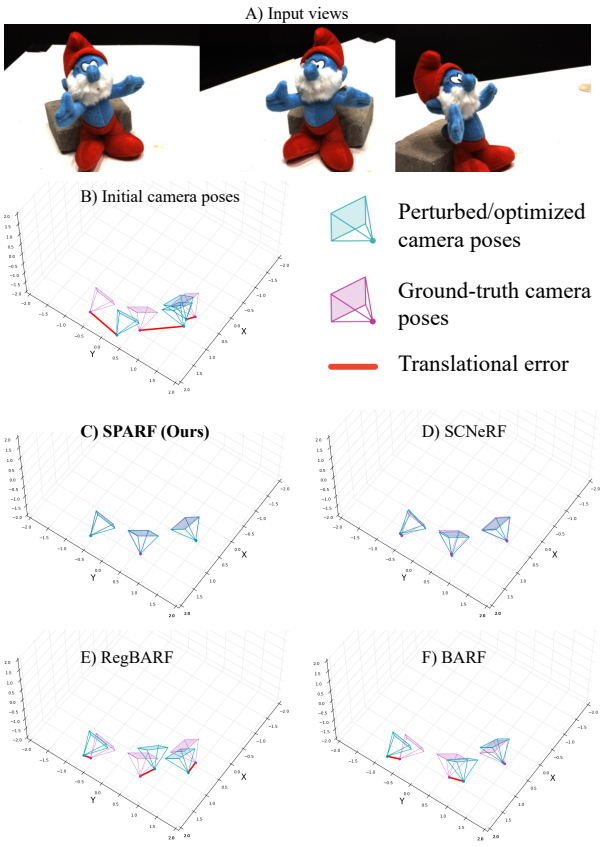

Figure 11. Initial and optimized poses on one scene of the DTU dataset, given 3 input views.

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. 13. 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.

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.

	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
BARF [24]	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 14. 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.

Qualitative comparisons: We provide qualitative comparisons for the 3-view regime. In Fig. 11, 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. 13.

Finally, we provide extensive examples of the novel-view synthesis capabilities of our approach SPARF in Fig. 14. 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. 14. Even in this setting, our SPARF significantly outperforms baseline BARF [24] 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. 14) are slightly worse than when trained on only a subset of 9 views (Tab. 13).

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 [20].

Results with 2, 6 and 9 views: As for DTU [20], 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

	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 [24]	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 [24, 32]	*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 [4, 24]	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 [21]	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 15. Evaluation on LLFF [39] 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.

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.

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. 12. 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. 15, 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. 16. 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.

	Rot. (°) ↓	Trans. (x 100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BARF [24]	0.85	0.26	25.09	0.77	**0.20**
SPARF*	**0.77**	**0.23**	**25.18**	**0.78**	**0.20**

Table 16. Evaluation on LLFF [39], 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), without including our depth consistency objective (Sec. 4.2) nor our staged training (Sec. 4.3).

Results with all views: For completeness, in Tab. 16 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, we evaluated multiple approaches on Replica, with 3 input views and initial poses obtained by COLMAP [38] with PDC-Net [44] 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 [38] run with PDC-Net [44] matches. For completeness, we here present the same comparison, when the initial poses are obtained with COLMAP with SuperPoint [12] and SuperGlue [36] matches instead. It corre-

		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		19.50	0.66	0.41	1.63
	DS-NeRF [11]	from COLMAP (run w.		21.55	0.74	0.26	0.91
	SPARF (Ours)	SP-SG [36] matches)		22.18	0.74	0.25	0.93
R	BARF [24]	3.23	18.05	19.41	0.68	0.34	0.95
	SCNeRF [21]	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 17. Evaluation on Replica [40] (3 views) with initial poses obtained by COLMAP [38, 44] with SP-SG [36] 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.

sponds to an initial rotation and translation errors of 2.61° and 15.31 respectively. The results are presented in Tab. 17.

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. 17, 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).

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. 18. 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.

	Pose Registration		**Novel View Synthetis**			
	Rot (°) ↓	Trans (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
BARF [24]	12.81	39.96	16.39	0.60	0.52	2.3
RegBARF [24, 32]	9.0	29.34	17.05	0.62	0.48	1.11
DistBARF [4, 32]	5.28	20.45	19.82	0.69	0.36	0.68
SCNeRF [21]	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 18. 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.

Qualitative comparisons: In Fig. 17, 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. 18, 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, 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 [32].

Results on DTU: We present results on DTU in Tab. 19. Our approach SPARF sets a new state of the art on all metrics 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. 20. 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.

	PSNR ↑			SSIM ↑			LPIPS ↓			DE ↓		
	3	6	9	3	6	9	3	6	9	3	6	9
PixelNeRF [56]	**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 [30]	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 [19]	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 [32]	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 [11]	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 19. Evaluation on the DTU dataset [20], 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 [3, 6, 19, 32, 56] are from [32]. 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 [56]	7.93	8.74	8.61	0.27	0.28	0.27	0.68	0.68	0.67
SRF [8]	12.3	13.1	13.0	0.25	0.29	0.30	0.59	0.59	0.61
MVSNeRF [6]	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 [30]	13.61	16.70	18.45	0.28	0.43	0.51	0.56	0.40	0.31
MipNeRF [3]	14.62	20.87	24.26	0.35	0.69	*0.81*	0.50	0.26	*0.17*
DietNeRF* [19]	14.94	21.75	24.3	0.37	0.72	0.80	0.5	0.25	0.18
RegNeRF* [32]	*19.08*	*23.10*	**24.86**	*0.59*	**0.76**	**0.82**	0.34	*0.21*	**0.16**
DS-NeRF [11]	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 20. Evaluation on the LLFF dataset [39], 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 [3] as their base architecture, while the others use NeRF [30]. Results of [3, 6, 19, 32, 56] are from [32]. The best and second-best results are in red and blue respectively.

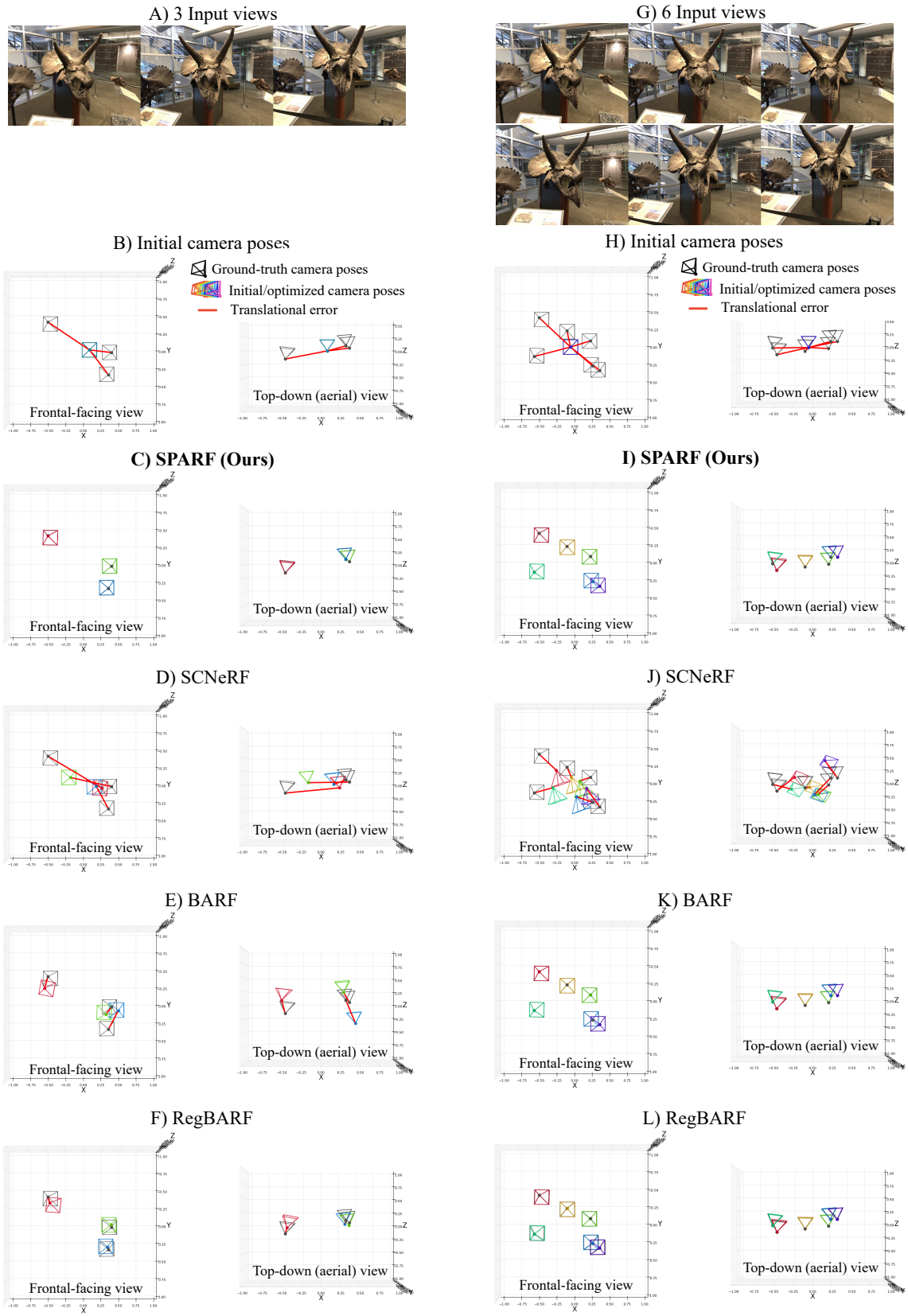

Figure 12. Initial and optimized camera poses on the scene 'horns' of the LLFF dataset. We consider 3 or 6 input views with initial identity poses.

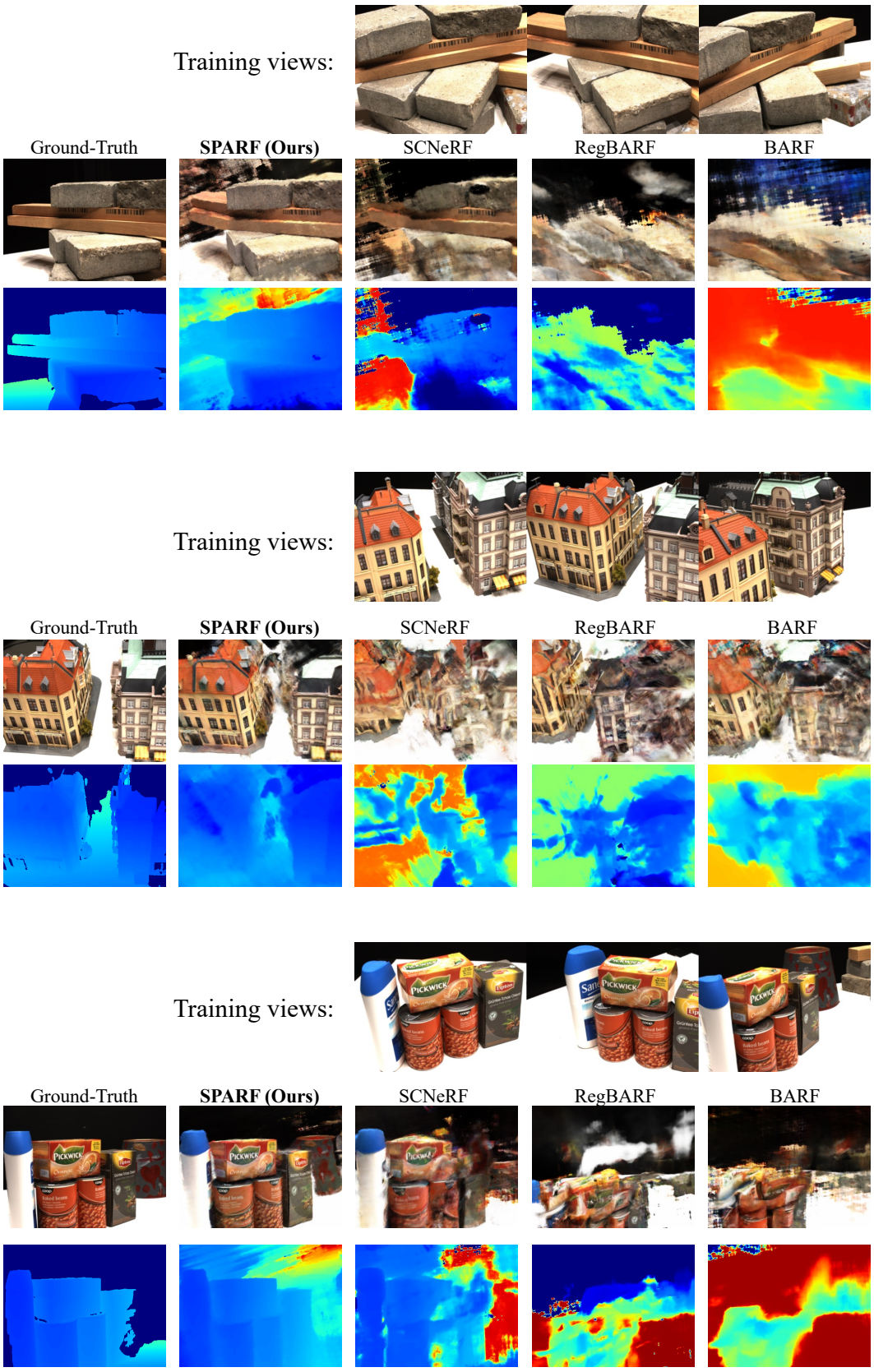

Figure 13. Novel-view renderings of alternative joint pose-NeRF training approaches on the DTU dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from an unseen viewpoint. We consider 3 input views with initial noisy poses. The initial camera poses are created by perturbing the ground-truth poses with 15% of additive Gaussian noise.

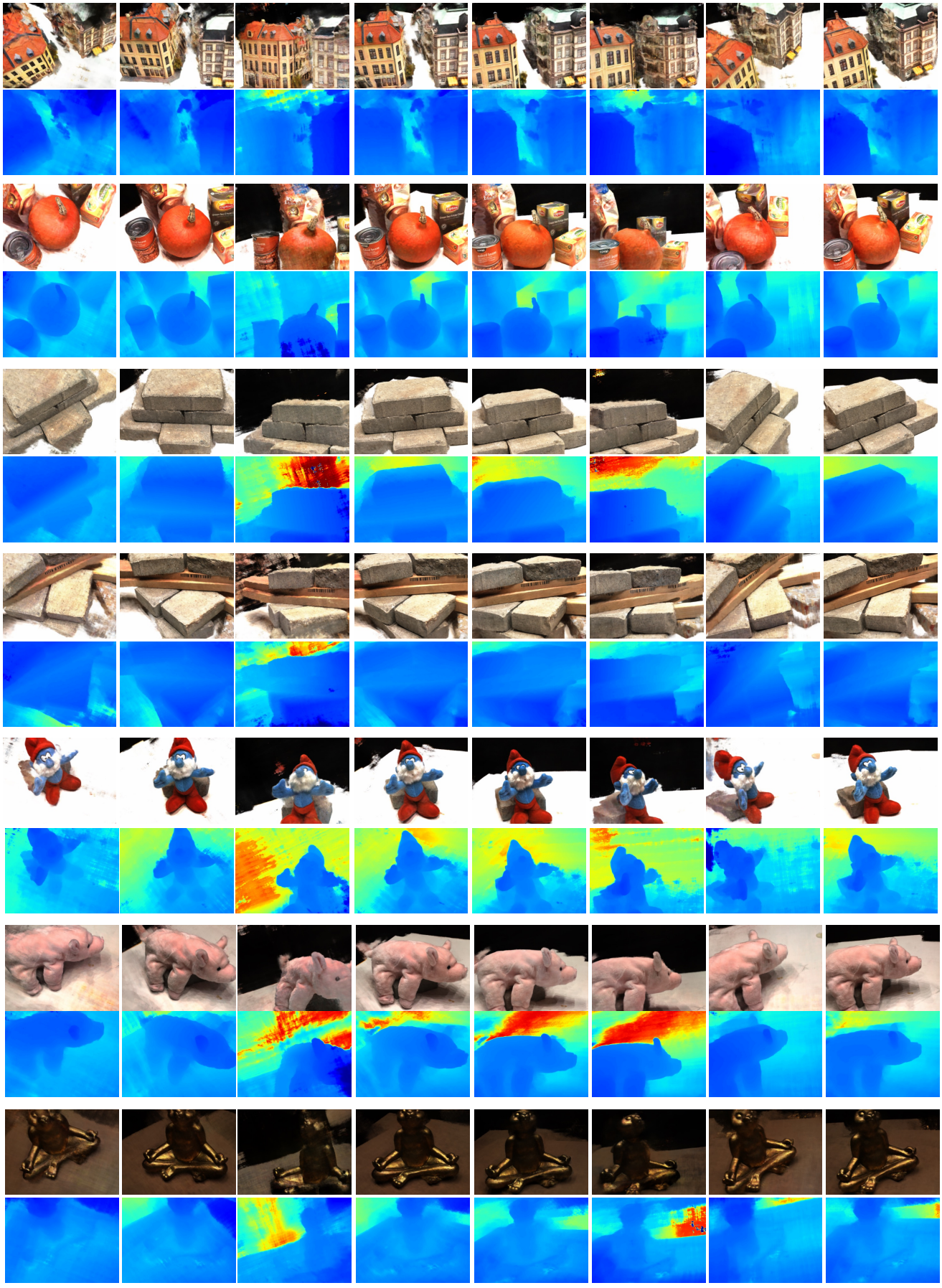

Figure 14. Novel-view renderings of our SPARF on the DTU dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from multiple unseen viewpoints. In each scene, we consider 3 input views (not shown here) with initial noisy poses, created by perturbing the ground-truth poses with 15% of additive Gaussian noise.

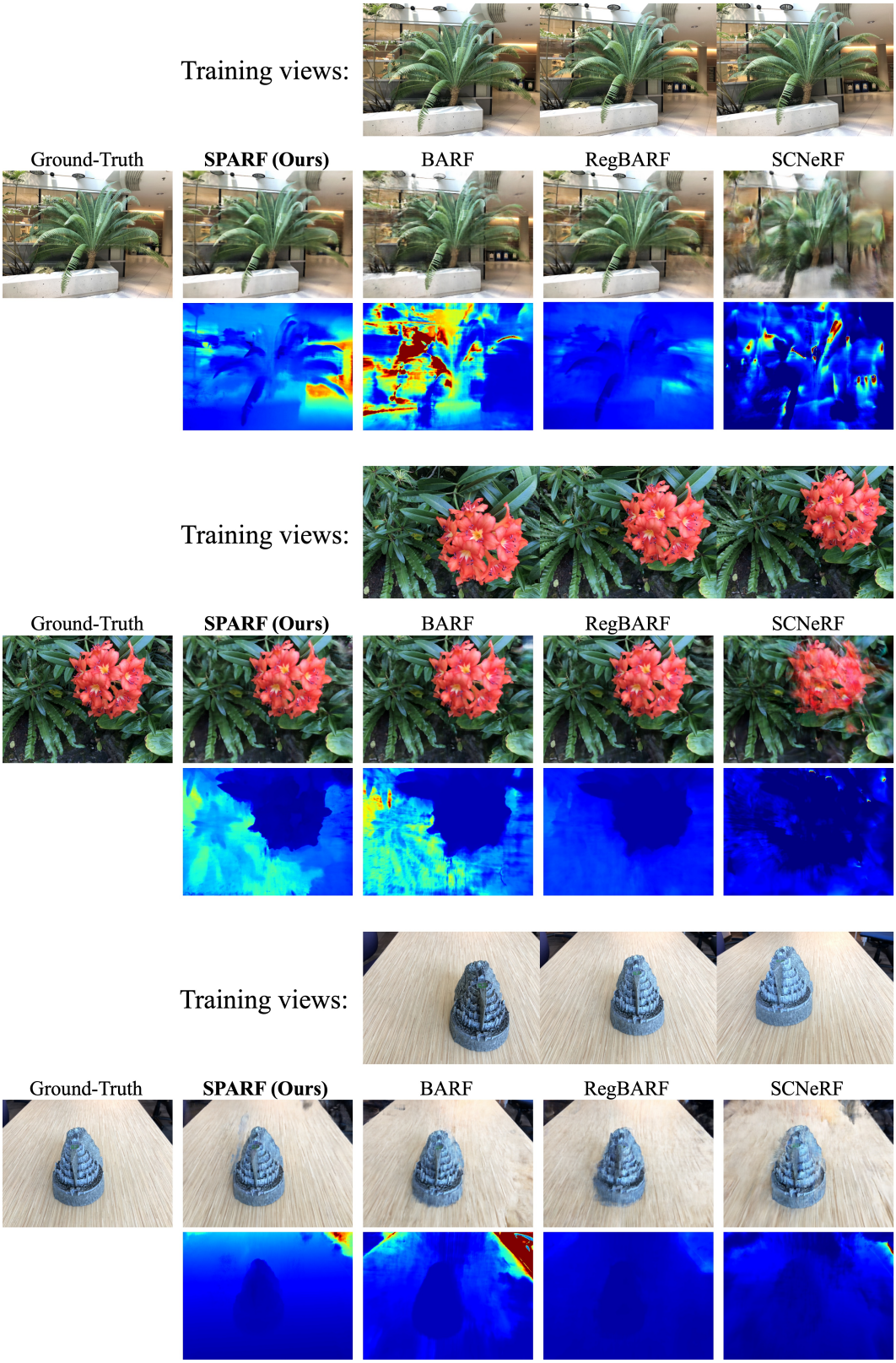

Figure 15. Novel-view renderings of alternative joint pose-NeRF training approaches on the LLFF dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from an unseen viewpoint. We consider 3 input views with initial identity poses.

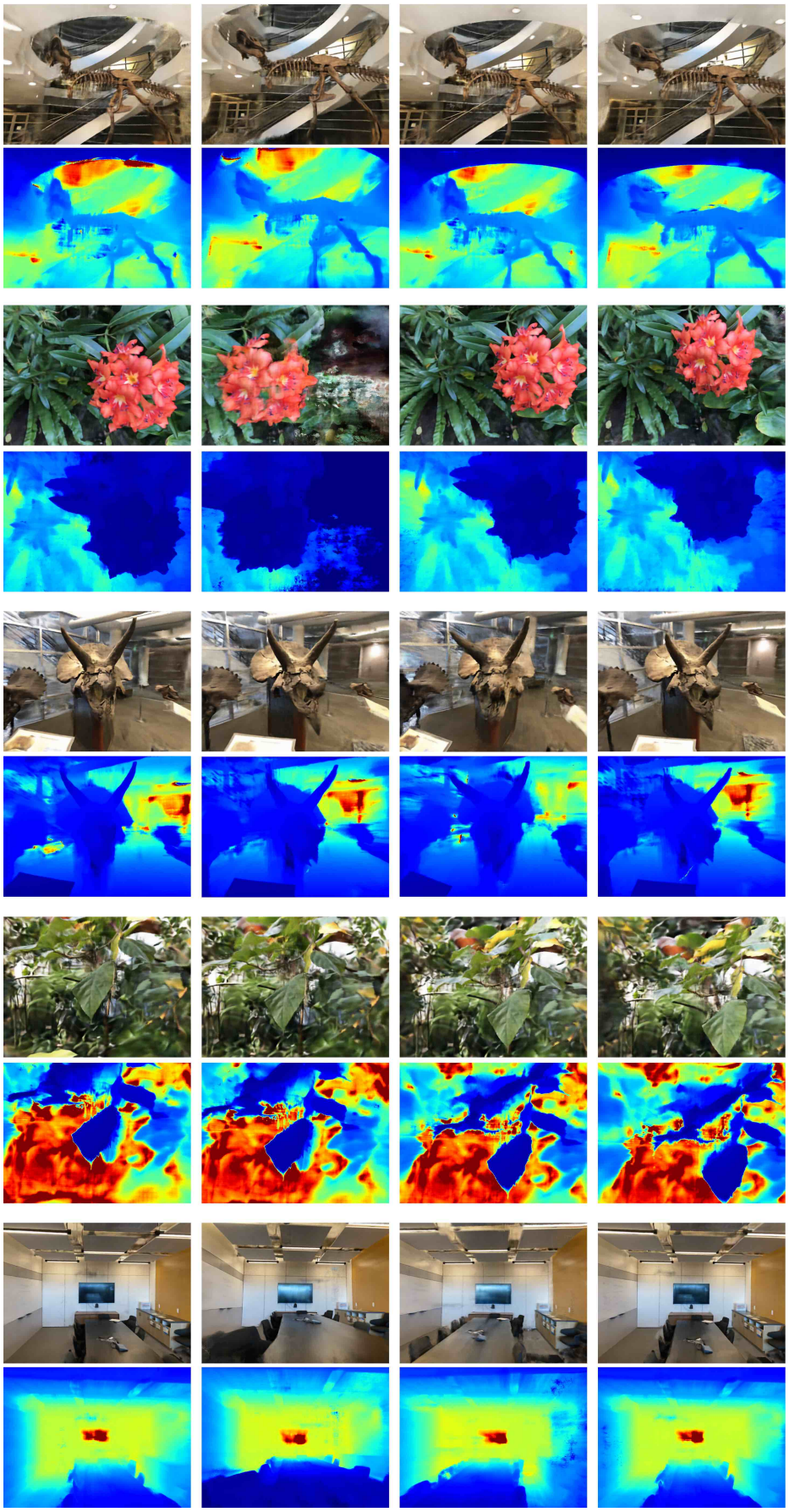

Figure 16. Novel-view renderings of our SPARF on the LLFF dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from multiple unseen viewpoints. In each scene, we consider 3 input views (not shown here) with initial identity poses.

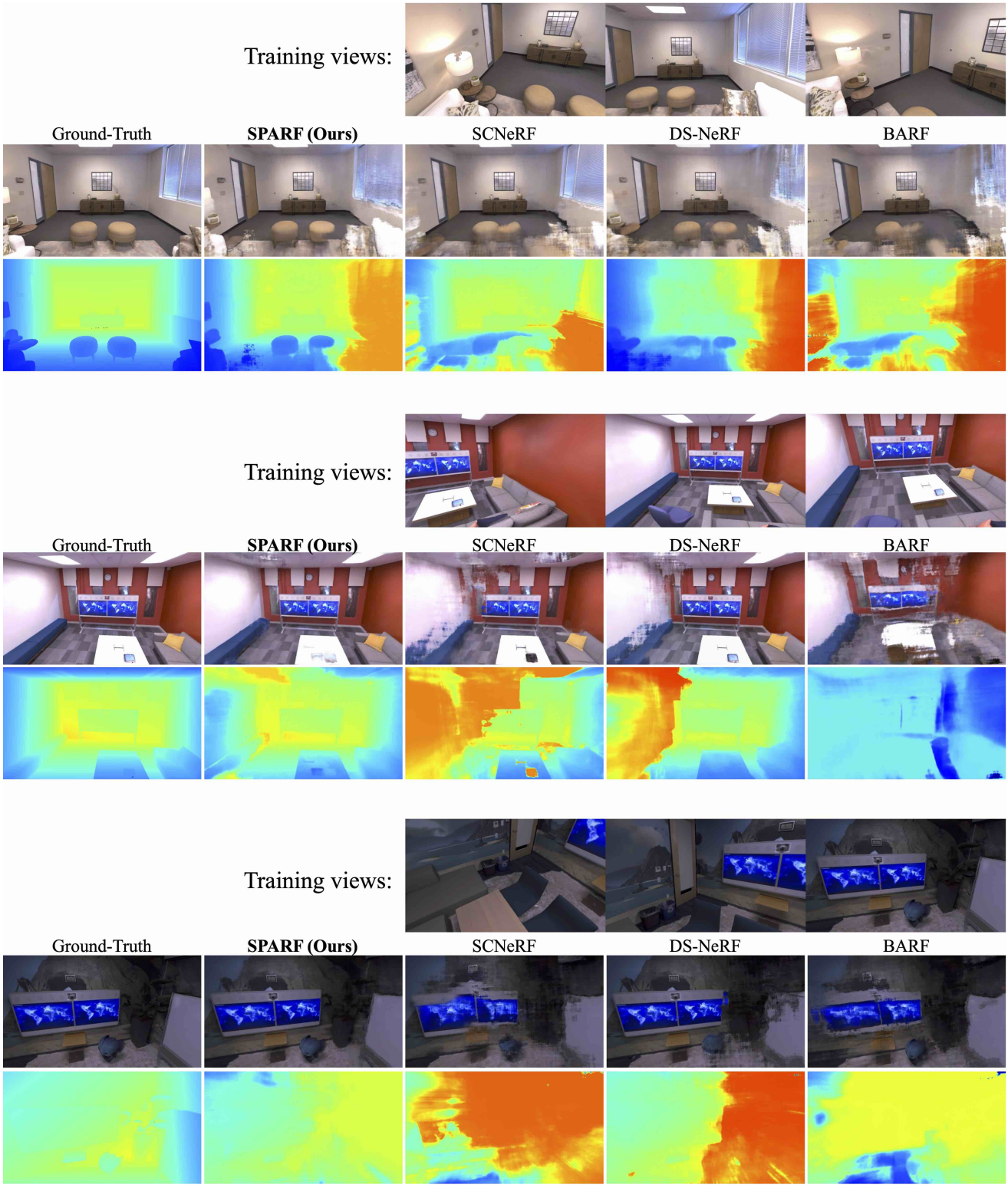

Figure 17. Novel-view renderings of alternative joint pose-NeRF training approaches on the Replica dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from an unseen viewpoint. On each scene, we consider 3 input views (not shown here) with initial poses obtained by COLMAP [38] with PDC-Net matches [44].

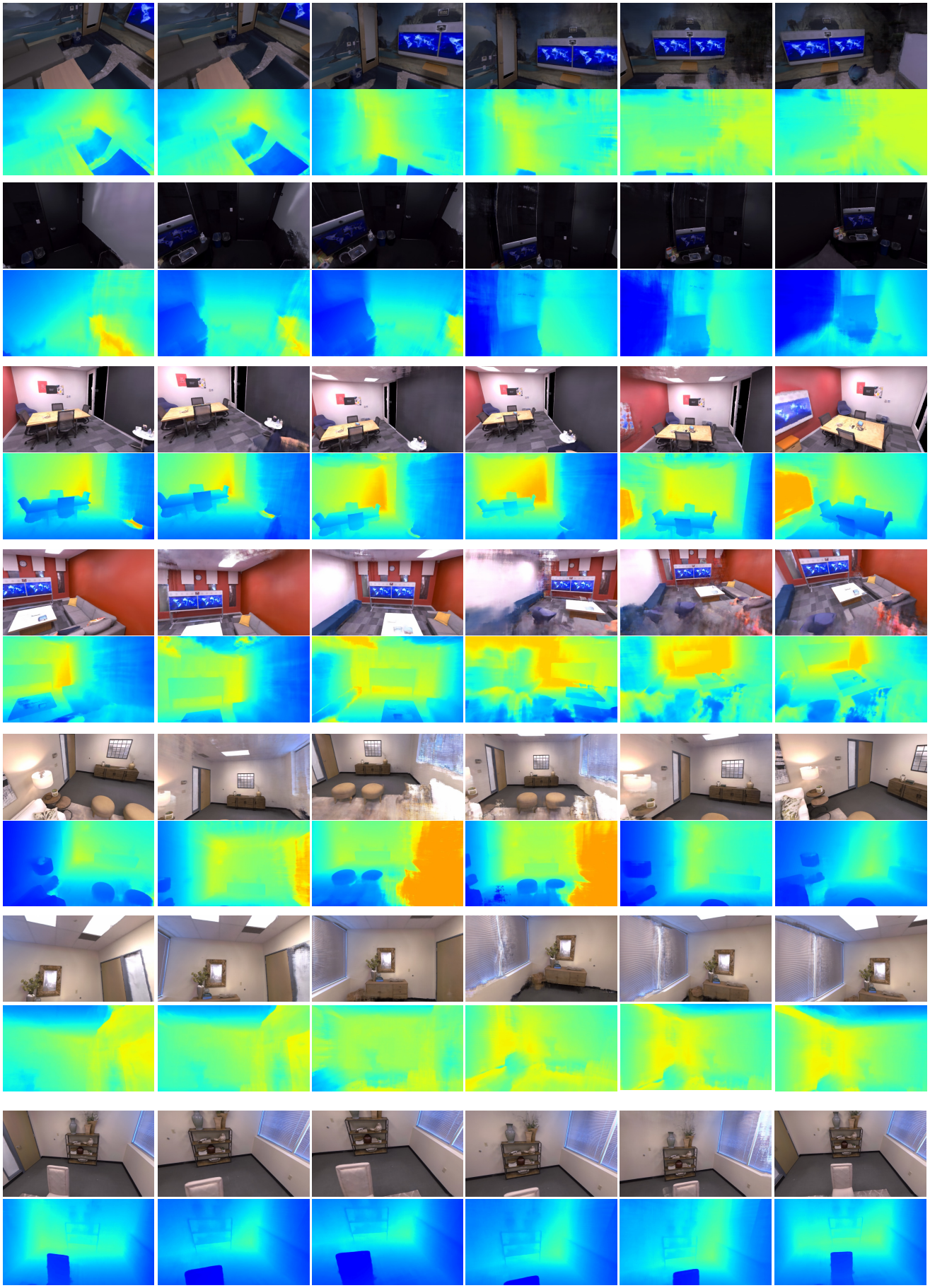

Figure 18. Novel-view renderings of our SPARF on the Replica dataset. For each scene, we show the RGB (first row) and depth (second row) renderings from multiple unseen viewpoints. On each scene, we consider 3 input views (not shown here) with initial poses obtained by COLMAP [38] with PDC-Net matches [44].

References

[1] Wei Wang 0108, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Backpropagation-friendly eigendecomposition. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Edward A. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3156–3164, 2019. 1

[2] Dejan Azinovic, Ricardo Martin-Brualla, Dan B. Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 6280–6291. IEEE, 2022. 2

[3] 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. 8, 21

[4] 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. 2, 7, 8, 17, 19, 20

[5] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan T. Barron, Hendrik P.A. Lensch, and Varun Jampani. SAMURAI: Shape And Material from Unconstrained Real-world Arbitrary Image collections. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 7

[6] 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. 1, 2, 21

[7] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3257–3267, 2021. 1

[8] 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. 2, 21

[9] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. GARF: gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *CoRR*, abs/2204.05735, 2022. 2, 3, 7

[10] 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. 5, 16

[11] 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. 1, 2, 4, 5, 8, 12, 17, 20, 21

[12] 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. 5, 11, 13, 14, 15, 19

[13] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 4

[14] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In *Computer Vision - ECCV 2018 Workshops - Munich, Ger-*

many, September 8-14, 2018, Proceedings, Part I, pages 738–745, 2018. 1

[15] Antoine Fond, Luca Del Pero, Nikola Sivacki, and Marco Paladini. End-to-end learning of keypoint detection and matching for relative pose estimation. *CoRR*, abs/2104.01085, 2021. 1

[16] Johan Fredriksson, Viktor Larsson, Carl Olsson, and Fredrik Kahl. Optimal relative pose with unknown correspondences. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1728–1736. IEEE Computer Society, 2016. 1

[17] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 2, 3, 4

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001. 4

[19] 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. 1, 2, 3, 4, 8, 21

[20] 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. 2, 5, 6, 7, 8, 13, 15, 16, 17, 18, 21

[21] 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. 2, 7, 8, 10, 11, 15, 17, 19, 20

[22] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 12902–12911. IEEE, 2022. 1, 2, 3, 8

[23] Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. Neroic: neural rendering of objects from online image collections. *ACM Trans. Graph.*, 41(4):56:1–56:12, 2022. 2, 7

[24] 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, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20

[25] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 7814–7823, 2022. 2

[26] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. *ECCV*, 2022. 2

[27] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004. 4

[28] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *Advanced Concepts for Intelligent Vision Systems - 18th International Conference, ACIVS 2017, Antwerp, Belgium, September 18-21, 2017, Proceedings*, pages 675–687, 2017. 1

[29] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6331–6341. IEEE, 2021. 2, 7

[30] 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, 3, 5, 6, 8, 9, 12, 17, 21

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 8

[32] 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. 1, 2, 3, 4, 5, 7, 8, 12, 17, 19, 20, 21

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 2

[34] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 1, 2, 5

[35] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. ORB: an efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2564–2571, 2011. 4

[36] 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, 11, 13, 14, 17, 19, 20

[37] Paul-Edouard Sarling. HLOC: Github project page. https://github.com/cvg/Hierarchical-Localization, 2021. 11

[38] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR 2016, Las Vegas, NV, USA*, pages 4104–4113, 2016. 1, 8, 11, 13, 17, 19, 20, 27, 28

[39] 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. 2, 5, 8, 15, 19, 21

[40] 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. 2, 5, 8, 20

[41] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6209–6218. IEEE, 2021. 2

[42] Alex Trevithick and Bo Yang. GRF: learning a general radiance field for 3d representation and rendering. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15162–15172, 2021. 2

[43] Prune Truong, Stefanos Apostolopoulos, Agata Mosinska, Samuel Stucky, Carlos Ciller, and Sandro De Zanet. Glampoints: Greedily learned accurate match points. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10731–10740, 2019. 4

[44] 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. 2, 4, 8, 9, 11, 12, 14, 15, 17, 19, 20, 27, 28

[45] Prune Truong, Martin Danelljan, and Radu Timofte. GLU-Net: Global-local universal network for dense flow and correspondences. In *2020 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2020*, 2020. 4

[46] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. In *Preprint*, 2021. 4

[47] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380, 1991. 12

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

[49] 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. 6, 13

[50] 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. 2, 3, 7, 12, 13, 16

[51] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5590–5599. IEEE, 2021. 2

[52] Yitong Xia, Hao Tang, Radu Timofte, and Luc Van Gool. Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. *CoRR*, abs/2210.04553, 2022. 2, 3, 7

[53] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. Ps-nerf: Neural inverse rendering for multi-view photometric stereo. *CoRR*, abs/2207.11406, 2022. 2

[54] 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. 2, 12

[55] 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. 13

[56] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 1, 2, 5, 8, 21

[57] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[58] Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXI*, volume 13691 of *Lecture Notes in Computer Science*, pages 592–611. Springer, 2022. 1

[59] Jason Y. Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. In *Conference on Neural Information Processing Systems*, 2021. 2, 7

[60] 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. 6, 13

[61] 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. 12

[62] 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. 6, 10, 11

[63] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[64] Bingbing Zhuang and Manmohan Chandraker. Fusing the old with the new: Learning relative camera pose with geometry-guided uncertainty. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 32–42, 2021. 1