DBARF: Deep Bundle-Adjusting Generalizable Neural Radiance Fields

Yu Chen Gim Hee Lee
Department of Computer Science, National University of Singapore
{chenyu, gimhee.lee}@nus.edu.sg

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.00011

Abstract

Recent works such as BARF and GARF can bundle adjust camera poses with neural radiance fields (NeRF) which is based on coordinate-MLPs. Despite the impressive results, these methods cannot be applied to Generalizable NeRFs (GeNeRFs) which require image feature extractions that are often based on more complicated 3D CNN or transformer architectures. In this work, we first analyze the difficulties of jointly optimizing camera poses with GeNeRFs, and then further propose our DBARF to tackle these issues. Our DBARF which bundle adjusts camera poses by taking a cost feature map as an implicit cost function can be jointly trained with GeNeRFs in a self-supervised manner. Unlike BARF and its follow-up works, which can only be applied to per-scene optimized NeRFs and need accurate initial camera poses with the exception of forward-facing scenes, our method can generalize across scenes and does not require any good initialization. Experiments show the effectiveness and generalization ability of our DBARF when evaluated on real-world datasets. Our code is available at https://aibluefisher.github.io/dbarf.

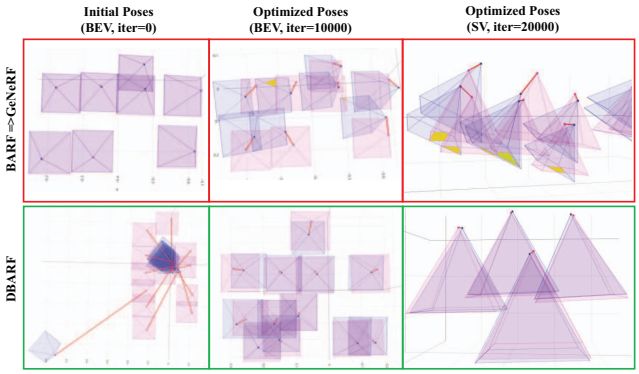

Figure 1. **Results of optimizing camera poses with BARF and DBARF**. From left to right are the initial camera poses, bird's eye view (BEV) of optimized camera poses after $1e4$ iterations, and side view (SV) of optimized camera pose after $2e4$ iterations. Red and blue denote ground truths and estimated camera poses (The inconsistent ground truth poses in different iterations are due to the randomness of selecting the training batches). **Top**: The camera poses diverge quickly when BARF [20] is applied to GeNeRF, even with the camera poses initialized by perturbing the ground truth with very small noise. **Bottom**: Results obtained by our DBARF, the camera poses are randomly initialized.

1. Introduction

The recent introduction of NeRF (Neural Radiance Fields) [28] bridges the gap between computer vision and computer graphics with the focus on the Novel view synthesis (NVS) task. NeRF demonstrates impressive capability of encoding the implicit scene representation and rendering high-quality images at novel views with only a small set of coordinate-based MLPs. Although NeRF and its variants simplify the dense 3D reconstruction part of the traditional photogrammetry pipeline that includes: the reconstruction of dense point clouds from posed images followed by the recovery and texture mapping of the surfaces into just a simple neural network inference, they still require known accurate camera poses as inputs.

Nonetheless, the acquisition of camera poses is expensive in the real world. Most NeRF-related methods obtain the camera poses by Structure-from-Motion (SfM) [4, 23, 34]. In SfM, camera poses are optimized under the keypoint-metric reprojection error in a process referred to as bundle adjustment [43]. A notorious problem of SfM is that it sometimes fails, *e.g.* in textureless or self-similar scenes, and can also take days or even weeks to complete for large-scale scenes. Consequently, one main forthcoming issue with NeRF is that its rendering quality highly relies on accurate camera poses. Recently, several works try to solve the pose inaccuracy jointly with NeRF. One of the representative works is BARF [20]. NeRF maps the pixel coordinates into high-dimensional space as Fourier features [39] before inputting into the MLPs to enable networks to learn the high-frequency part of images. However, Fourier features can be a double-edged sword when the camera poses are jointly optimized with NeRF, where gradients from high-frequency components dominate the low-frequency parts during training. To mitigate this problem, BARF draws inspiration from the non-smoothness optimization in high-dimensional functions: optimizer can get stuck at a local optimum, but the training can be easier when the objective

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.00011

function is made smoother. Consequently, BARF adopts a coarse-to-fine strategy which first masks out the high-frequency components, and then gradually reactivates them after the low-frequency components become stable. The camera poses are adjusted by the photometric loss during training instead of the keypoint-metric cost in SfM. Despite its promising results, BARF and its follow-up works [5, 26] still require the pre-computed camera poses from SfM.

One other issue with vanilla NeRF is that it needs time-consuming per-scene training. Making NeRF generalizable across scenes [3, 18, 47, 53] has recently gained increasing attention. However, similar to vanilla NeRF, GeNeRFs (generalizable NeRFs) also depend on accurate camera poses. There is no existing work that tried to optimize the camera poses jointly with GeNeRFs. **This intrigues us to investigate the replacement of NeRF with GeNeRFs in BARF.** We find that the joint optimization is non-trivial in our task settings, and the camera poses can diverge quickly even when initialized with the ground truths (*cf*. top row of Fig. 1).

In this paper, we identified two potential reasons which cause the failure of bundle adjusting GeNeRFs. The first reason is the aggregated feature outliers, which are caused by occlusions. The other reason is due to the high non-convexity of the cost function produced by ResNet features [40], which produces incoherent displacements like the issue caused by positional encodings [39] in BARF. We further proposed our method DBARF, which jointly optimizes GeNeRF and relative camera poses by a deep neural network. Our implicit training objective can be equivalently deemed as a smooth function of the coarse-to-fine training objective in BARF. Specifically, we construct a residual feature map by warping 3D points onto the feature maps of the nearby views. We then take the residual feature map as an implicit cost function, which we refer to as *cost map* in the following sections. By taking the cost map as input, we utilize a deep pose optimizer to learn to correct the relative camera poses from the target view to nearby views. We further jointly train the pose optimizer and a GeNeRF with images as supervision, which does not rely on ground truth camera poses. In contrast to previous methods which only focus on per-scene camera pose optimization, our network is generalizable across scenes.

In summary, the contributions of this work are:

- We conduct an experiment on bundle adjusting GeNeRFs by gradient descent and analyze the difficulty of jointly optimizing camera poses with GeNeRFs.
- We present DBARF to deep bundle adjusting camera poses with GeNeRFs. The approach is trained end-to-end without requiring known absolute camera poses.
- We conduct experiments to show the generalization ability of our DBARF, which can outperform BARF and GARF even without per-scene fine-tuning.

2. Related Work

Novel View Synthesis. Given posed images, vanilla NeRF [28] used an MLP to predict the volume density and pixel color for a point sampled at 3D space. The low-dimensional inputs (point coordinates and ray directions) are encoded by the positional encodings [39] to high-dimensional representations, such that the network can learn high-frequency components of images. While NeRF [28] and later follow-up works achieved great progress in improving the rendering quality, such as the anti-aliasing effects [2, 28, 55] and reflectance [46], reducing training time [9, 29, 32] and rendering time [24, 38, 52], they still require time-consuming per-scene training.

Pixel-NeRF [53] is the first that generalizes NeRF to unseen scenes. It extracts image features from a feature volume by projection and interpolation, and then the image features are fed into a NeRF-like MLP network to obtain RGB color and density values. IBRNet [47] aggregates per-point image feature from nearby views, the image features are weighted by a PointNet-like [31] architecture. Taking the weighted features as input, a ray transformer [45] is further introduced to predict density, and another MLP is used to predict the pixel color. MVSNeRF [3] constructs 3D feature cost volume from N depth hypothesis, then a neural voxel volume is reconstructed by a 3D CNN, pixel color and volume density are predicted by a MLP.

GeoNeRF [18] extends MVSNeRF by using CasMVSNet [13] to let the network be aware of scene geometry. It adopts a similar approach as IBRNet [47] to regress image color and volume density. NeuRay [25] further predicts the visibility of 3D points to tackle the occlusion issue in previous GeNeRFs, and a consistency loss is also proposed to refine the visibility in per-scene fine-tuning. Instead of composing colors by volume rendering, LFNR [37] and GPNR [36] adopts a 4D light field representation and a transformer-based architecture to predict the occlusions and colors for features aggregated from epipolar lines [15].

Novel View Synthesis with Pose Refinement. I-NeRF [22] regressed single camera pose while requiring a pretrained NeRF model and matched keypoints as constraints. NeRF−− [49] jointly optimizing the network of NeRF and camera pose embeddings, which achieved comparable accuracy with NeRF methods that require posed images. SiNeRF [50] adopts a SIREN-MLP [35] and a mixed region sampling strategy to circumvent the sub-optimality issue in NeRF−−. BARF [20] proposed to jointly train NeRF with imperfect camera poses with a coarse-to-fine strategy. During training, the low-frequency components are learned at first and the high-frequency parts

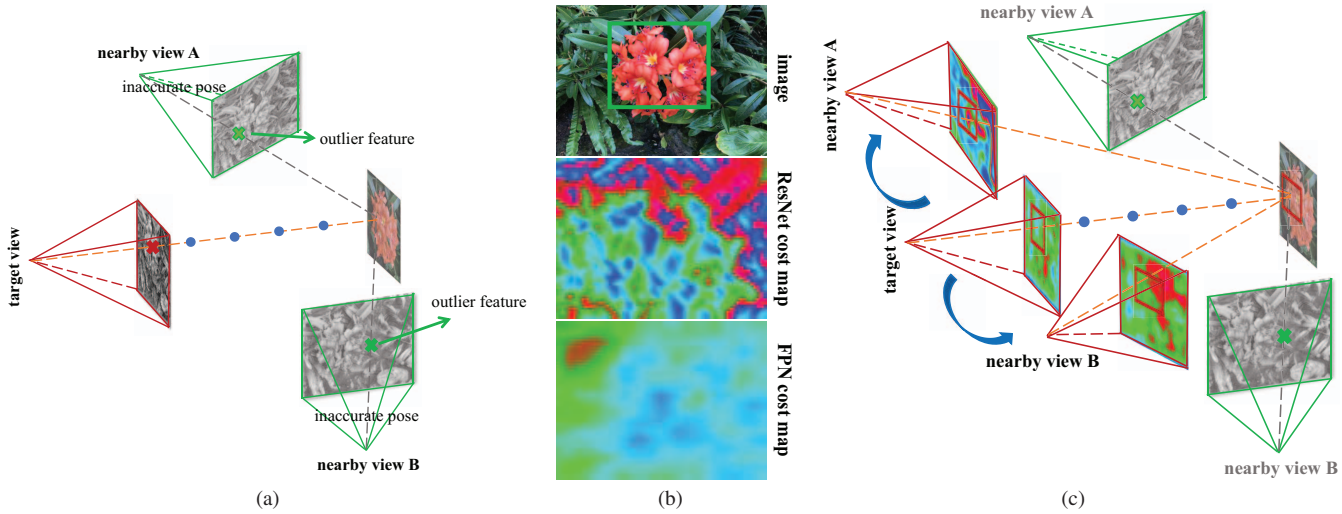

Figure 2. **The difficulties when optimizing camera poses with GeNeRFs**: a) Sampled features tend to be outliers when they are occluded. b) ResNet gives a non-smooth cost feature map (middle) while feature patches with FPN lead to a smoother cost map (bottom) c) Our method sample image patches to predict relative camera poses.

are gradually activated to alleviate gradient inconsistency issue. GARF [5] extends BARF with a positional-embedding less coordinate network. RM-NeRF [17] jointly trains a GNN-based motion averaging network [12, 30] and Mip-NeRF [1] to solve the camera pose refinement issue in multi-scale scenes. GNeRF [26] utilized an adversarial learning method to estimate camera poses. Camera poses in GNeRF are randomly sampled from prior-known camera distribution, and then a generator generates the corresponding fake images by volume rendering, together with a discriminator that classifies the real and fake images. An inversion network finally learns to predict the camera poses by taking the fake images as input. VMRF [54] can learn NeRF without known camera poses. The unbalanced optimal transport is introduced to learn the relative transformation between the real image and the rendered image, then camera poses are updated by the predicted relative poses to enable a finer training of NeRF.

None of the mentioned works can be applied to generalizable NeRFs and thus require time-consuming per-scene optimization. We also notice that there is a concurrent work [10] trying to make NeRF and pose regression generalizable. However, it only focuses on single-view rendering tasks. In contrast, we focus on the multiple views settings, which are more challenging than the single view.

3. Notations and Preliminaries

We follow the notations in BARF [20]. The image synthesis process is depicted by the equation below:

$$\hat{\mathbf{I}} = h\big(g(\omega(\mathbf{X}^1, \mathbf{P}); \mathbf{\Theta}),\ \cdots,\ g(\omega(\mathbf{X}^K, \mathbf{P}); \mathbf{\Theta})\big), \quad (1)$$

where $\mathbf{X}^k = Z_k\mathbf{u}$ is a 3D point in the camera frame, $\{Z_1, Z_2, \cdots, Z_K\}$ are the sampled depths and $\mathbf{u}$ is the camera normalized pixel coordinates in the image. $h(\cdot)$ is the ray composition function, $g(\cdot)$ is the NeRF network, $\omega(\cdot)$ denotes the rigid transformation which projects the point $Z_k\mathbf{u}$ from the camera frame to the world frame by the camera pose $\mathbf{P}$, $\mathbf{\Theta}$ denotes the network parameters.

Once we obtained the point color c_k and volume density σ_k of all the K points, the per-pixel RGB $C(\mathbf{r})$ and depth value $D(\mathbf{r})$ can be approximated with the quadrature rule:

$$C(\mathbf{r}) = \textstyle\sum_{k=1}^{K} T_k(1 - \exp(-\sigma_k\delta_k))c_k, \quad (2a)$$

$$D(\mathbf{r}) = \textstyle\sum_{k=1}^{K} T_k(1 - \exp(-\sigma_k\delta_k))Z_k, \quad (2b)$$

where $T_k = \exp(-\sum_{l=1}^{k-1}\sigma_l\delta_l)$, $\delta_k = Z_{k+1} - Z_k$ is the accumulated transmittance, and δ_k is the distance between adjacent samples. Please refer to [28] for more details on the volume rendering technique.

4. Our Method

4.1. Generalizable Neural Radiance Field

We adopt the term **GeNeRFs** to denote a bundle of Generalizable Neural Radiance Field methods [3, 18, 47, 53]. Since these methods share a common philosophy in their network architectures, we can abstract GeNeRFs into a series of high-dimensional functions.

GeNeRFs first extract 2D image features by projecting a point onto the feature map $\mathbf{F}_j$:

$$\mathbf{f} = \chi\big(\Pi(\mathbf{P}_j, \omega(\mathbf{X}_i^k, \mathbf{P}_i)), \mathbf{F}_j\big), \quad (3)$$

where $\chi(\cdot)$ is the differentiable bilinear interpolation function, $\Pi(\cdot)$ is the reprojection function which maps points from world frame to image plane, $\mathbf{P}_i$ is the camera pose of image i, and $\mathbf{P}_j$ is the camera pose of image j in the nearby

view of image i. $\mathbf{X}_i^k = Z_k \mathbf{u}_i$ is the k^{th} 3D point in image i, where $\mathbf{u}_i$ is the camera normalized pixel coordinates and Z_k is depth of the k^{th} 3D point in image i.

To render a novel view i, GeNeRFs either sample K points and aggregate pixel-level features for each emitted ray, or construct a 3D cost volume by plane sweep [13, 51], from M selected nearby views. Subsequently, per-depth volume density and pixel color are predicted by a neural network. For clarity and without losing generality, we abstract the feature aggregation function $f_a(\cdot)$ as:

$$g_k = f_a(\mathbf{f}_1^k, \mathbf{f}_2^k, \cdots, \mathbf{f}_M^k), \tag{4}$$

where $\mathbf{f}_m^k$ denotes the feature vector of image point $\mathbf{u}$ sampled at depth Z_k in image m at the nearby view of image i. The rendered target image is then given by:

$$\hat{\mathbf{I}}_{\text{target}} := \hat{\mathbf{I}}_i = h(g_1, \cdots, g_K; \mathbf{\Phi}), \tag{5}$$

where $h(\cdot)$ is the GeNeRF network, and $\mathbf{\Phi}$ is the network parameters.

Similar to vanilla NeRF, the training loss for GeNeRFs is the photometric error between the rendered target image and the ground truth target image:

$$\mathcal{L}_{\text{rgb}} = \sum_i^N \sum_{\mathbf{u}} \|\hat{\mathbf{I}}_i - \mathbf{I}_i(\mathbf{u})\|. \tag{6}$$

N is the total number of images in the training dataset.

4.2. Difficulties of Bundle Adjusting GeNeRFs

BARF [20] can jointly optimize NeRF with imperfect camera poses. The success of BARF can be largely attributed to the coarse-to-fine training strategy, which can deal with the gradient inconsistency between low-frequency components and high-frequency components. Specifically, the low-frequency components are first learned with the high-frequency part being masked out; then the high-frequency components are learned when the low-frequency components become stable. Otherwise, gradients from the high-frequency components, *i.e.* high k's tend to dominate the training process due to the positional encodings [39]:

$$\frac{\partial \gamma_k(\mathbf{P})}{\partial \mathbf{P}} = 2^k \pi \cdot [-\sin(2^k \pi \mathbf{P}), \cos(2^k \pi \mathbf{P})], \tag{7}$$

where $\gamma_k(\mathbf{P}) = [\cos(2^k \pi \mathbf{P}), \sin(2^k \pi \mathbf{P})]$.

The fact that BARF and its variants [17, 49, 50] can optimize the camera poses by gradient descent jointly with NeRF intrigues us to ask the question: **Can we also directly optimize the camera poses jointly with GeNeRFs by gradient descent just like BARF?** To answer the question, we adopt a pretrained GeNeRF model and construct a $N \times 6$ learnable pose embedding like BARF. The pose embedding is jointly trained with the GeNeRF model and optimized by Adam with a learning rate of $1e-5$. Unfortunately, we found the camera poses drifted significantly even when initialized from the ground truths. The result is illustrated in Fig. 1. Our question now becomes: **What is the reason that prevents the joint optimization of the camera poses with GeNeRFs?** Although a thorough theoretical analysis of the question is difficult due to the high complexity of GeNeRFs, we postulate the potential reasons by observing the gradient flow during back-propagation. Particularly, the gradient of $\mathcal{L}_{\text{rgb}}$ with respect to the camera poses can be written as:

$$\frac{\partial \mathcal{L}_{\text{rgb}}}{\partial \mathbf{P}_j} = \underbrace{\sum_{i \neq j}^{N} \sum_{\mathbf{u}} \sum_{k}^{K} \frac{\partial h}{\partial g_k} \cdot \frac{\partial g_k}{\partial \mathbf{f}_i^k} \cdot \frac{\partial \mathbf{f}_i^k}{\partial \mathbf{P}_j}}_{\text{image } j \text{ is one of the nearby views of image } i} + \underbrace{\sum_{m}^{M} \sum_{\mathbf{u}} \sum_{k}^{K} \frac{\partial h}{\partial g_k} \cdot \frac{\partial g_k}{\partial \mathbf{f}_m^k} \cdot \frac{\partial \mathbf{f}_m^k}{\partial \mathbf{P}_j}}_{\text{image } j \text{ is the target image}}. \tag{8}$$

Two problems can arise in the computation of the gradients of $\mathcal{L}_{\text{rgb}}$ given in Eq. 8. 1) **An image feature can be an outlier**. For example, the sampled pixel of the target view is far away from or missing its correspondences in the nearby views due to occlusion, as illustrated in Fig. 2(a). Without a special design of the network architecture, the aggregation function $f_a(\cdot)$ is not aware of occlusions. Consequently, this causes the two terms $\frac{\partial \mathbf{f}_i^k}{\partial \mathbf{P}_j}$ and $\frac{\partial \mathbf{f}_m^k}{\partial \mathbf{P}_j}$ to be erroneous, and thus causing the final gradient $\frac{\partial \mathcal{L}_{\text{rgb}}}{\partial \mathbf{P}_j}$ to be wrong. 2) **Non-smooth cost map caused by ResNet-like features.** Fig. 2b (middle) shows an example of the non-smooth cost map from ResNet. Unfortunately, the coarse-to-fine training strategy in BARF to first suppress the high-frequency components and then add them back when the low-frequency components become stabilized is not helpful since most GeNeRFs work directly on the features and do not use positional encodings.

4.3. DBARF

Based on the analysis in Sec. 4.2, we propose DBARF to jointly optimize the camera poses with GeNeRFs in the following sections. Fig. 3 shows our network architecture. To demonstrate our method in detail, we take IBRNet as the GeNeRF method, and we note that it generally does not affect the applicability of our method to other GeNeRFs.

4.3.1 Camera Poses Optimization

Given a point $\mathbf{X}_i^k$ in the camera frame of target view i, IBRNet aggregates features by projecting the point into nearby views:

$$\Pi(\mathbf{P}_j, \omega(\mathbf{X}_i^k, \mathbf{P}_i)) = \mathbf{K}_j \mathbf{P}_j \mathbf{P}_i^{-1} \mathbf{X}_i = \mathbf{K}_j \mathbf{P}_{ij} \mathbf{X}_i^k, \tag{9}$$

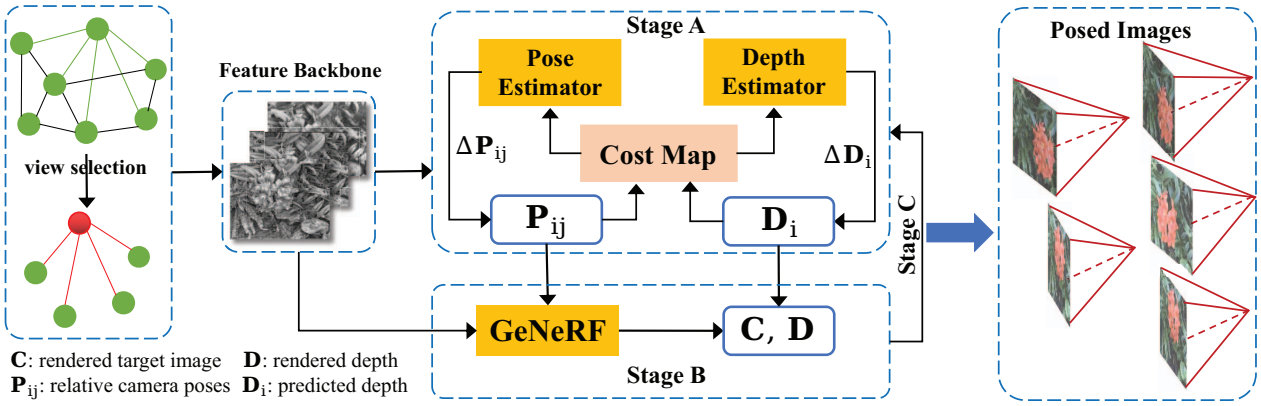

Figure 3. **Network architecture of our proposed DBARF**. The input is images and a scene graph. 1) Nearby views are selected from a scene graph since the camera poses are unknown. 2) Image features are extracted by ResNet-like [16] backbone. 3) In stage A, the image feature of the target view is warped to each nearby view by the corresponding current camera poses and depth, a cost map is constructed by the image feature difference. Camera poses and depth are recurrently optimized by taking the cost map as an implicit loss. 4) In stage B, we utilize a generalizable NeRF to predict image color and density value, and the final image is rendered by volume rendering. 5) In stage C, the pose optimizer and the generalizable NeRF are jointly learned. 6) Finally, our network outputs the posed images.

where $\mathbf{K}_j$ is the intrinsics matrix of image j, $\mathbf{P}_{ij} = \mathbf{P}_j \mathbf{P}_i^{-1}$ is the relative camera pose from image i to image j.

Suppose we have initial camera poses $\mathbf{P}^{\text{init}}$, we need to first correct the camera poses before aggregating useful image features. Since the appearances of extracted image features are inconsistent due to inaccurate initial camera poses, an intuitive solution is to construct a cost function that enforces the feature-metric consistency across the target view and all nearby views, *i.e.*:

$$\mathcal{C} = \sum_{\mathbf{u}_i} \sum_{j \in \mathcal{N}(i)} \rho\big(\|\chi\big(\mathbf{K}_j \mathbf{P}_{ij} \mathbf{X}_i^k, \mathbf{F}_j\big) - \chi(\mathbf{u}_i, \mathbf{F}_i)\big)\|, \quad (10)$$

which has been shown to be more robust than the photometric cost in Eq. (6) and the keypoint-based bundle adjustment [23]. $\rho(\cdot)$ can be any robust loss function.

However, simply adopting Eq. (10) to optimize the camera poses without knowing the outlier distribution to apply a suitable robust loss $\rho(\cdot)$ can give bad results. Furthermore, first-order optimizers can also easily get stuck at bad local minima in our task. Therefore, we seek an approach that can minimize Eq. (10) while bypassing direct gradient descent. Instead of explicitly taking Eq. (10) as an objective and optimizing camera poses by gradient descent, we implicitly minimize it by taking the feature error as an input to another neural network. Since NeRF randomly samples points in the target view during training, we lose the spatial information of the features when the neural network directly takes Eq. (10) as input. To alleviate the problem, we sample a patch $\mathcal{S}(\mathbf{u}_i)$ centered on $\mathbf{u}_i$ from the target view for the cost map generation and take the average of the aggregated feature cost map (See Fig. 2c), *i.e.*:

$$\mathcal{C} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \|\chi\big(\mathbf{K}_j \mathbf{P}_{ij} \mathbf{X}_{\mathcal{S}(\mathbf{u}_i)}, \mathbf{F}_j\big) - \chi(\mathcal{S}(\mathbf{u}_i), \mathbf{F}_i)\|, \quad (11)$$

where $\mathbf{X}_{\mathcal{S}(\mathbf{u}_i)}$ denotes the patch of 3D points which is computed from a predicted depth map $\mathbf{D}_i$ for the target image i instead of the sampled depth value $Z_{k,i}$ because it is inaccurate. We also do not compute the depth value using Eq. (2b) since NeRF does not learn the scene geometry well.

To make the implicit objective smoother to ease the joint training, inspired by BANet [40], we adopt the FPN (Feature Pyramid Network) [21] as our feature backbone. Given a cost feature map in Eq. (11), we aim at updating the relative camera poses $\mathbf{P}_{ij}$ and the depth map $\mathbf{D}_i$.

Following the RAFT-like [14, 41, 42] architecture, we adopt a recurrent GRU to predict the camera poses and depth map. Given initial camera poses $\mathbf{P}_{ij}^0$ and depth $\mathbf{D}_i^0$, we compute an initial cost map $\mathcal{C}^0$ using Eq. (11). We then use a GRU to predict the relative camera pose correction $\Delta\mathbf{P}_{ij}$ and depth correction $\Delta\mathbf{D}_k$ at the current iteration t, and update the camera poses and depth, respectively, as:

$$\mathbf{P}_{ij}^{t+1} = \mathbf{P}_{ij}^t + \Delta\mathbf{P}_{ij}, \quad \mathbf{D}_i^{t+1} = \mathbf{D}_i^t + \Delta\mathbf{D}_i. \quad (12)$$

During training, $\mathbf{P}_{ij}^0$ and $\mathbf{D}_i^0$ are randomly initialized and Eq. (12) is executed for a fixed t iteration. Note that after each iteration, the cost map $\mathcal{C}^{t+1}$ is updated by taking the current relative poses and depth map as input. Stage A in Fig. 3 illustrates the recurrent updating step.

4.3.2 Scene Graph: Nearby Views Selection

Existing GeNeRFs aggregate features from nearby views by selecting the nearest top-k nearby views with the known absolute camera poses. Since the absolute camera poses are not given in our setting, we select the nearby views using a scene graph. A scene graph records neighbors of a target view $\mathbf{I}_i$. To construct the scene graph, we extract

keypoints for each image using SuperPoint [8] and obtain feature matches for each candidate image pair using SuperGlue [33]. Wrong feature matches are filtered by checking the epipolar constraints [15]. Two images become neighbors when they share enough image keypoint matches. We simply select nearby views by sorting their neighbors according to the number of inlier matches in descending order. The scene graph construction only needs to be executed once for each scene and thus is a preprocessing step.

4.4. Training Objectives

For depth optimization, we adopt the edge-aware depth map smoothness loss in [11] for self-supervised depth prediction, which can penalize changes where the original image is smooth:

$$\mathcal{L}_{\text{depth}} = |\partial_x \mathbf{D}| \exp^{-|\partial_x \mathbf{I}|} + |\partial_y \mathbf{D}| \exp^{-|\partial_y \mathbf{I}|}, \tag{13}$$

where ∂_x and ∂_y are the image gradients.

For camera poses optimization, we adopt the warped photometric loss [14] for self-supervised pose optimization:

$$\mathcal{L}_{\text{photo}} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \left(\alpha \frac{1 - \text{ssim}(\mathbf{I}_i^{'} - \mathbf{I}_i)}{2} + (1-\alpha)\|\mathbf{I}_i^{'} - \mathbf{I}_i\|\right), \tag{14}$$

where $\mathbf{I}_i^{'}$ is warped from nearby image j to the target image i, ssim is the structural similarity loss [48].

For GeNeRF, we use the same loss of Eq. (6). Finally, our final loss function is defined as:

$$\mathcal{L}_{\text{final}} = 2^{\beta \cdot t}(\mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{photo}}) + (1 - 2^{\beta \cdot t})\mathcal{L}_{\text{rgb}}, \tag{15}$$

where $\beta = -1e5$, t is the current training iteration number.

5. Experiments

Training Datasets. We pretrain IBRNet and our method on the 63 scenes of the self-collected datasets from IBRNet [47], the 33 real scenes captured by a handheld headphone from LLFF [27] with the ground truth camera poses obtained from COLMAP, and 20 indoor scenes from the ScanNet dataset [6] with ground truth camera poses provided by BundleFusion [7]. The ground truth camera poses are provided by IBRNet, but not used in our method.

Evaluation Datasets. We evaluate BARF, IBRNet, and our method on the LLFF dataset [27] and the ScanNet dataset [6]. For IBRNet and our method, the evaluated scenes are not used during pre-training. For BARF and GARF, we train and evaluate them on the same scene in $200,000$ iterations. 1/8th and 1/20th of the images are respectively held out for testing on LLFF and ScanNet while others are reserved for finetuning. More scenes are evaluated in our supplementary materials.

Implementation Details. We adopt IBRNet [47] as the GeNeRF implementation and DRO [14] as the pose optimizer. Our method and IBRNet are both trained on a single 24G NVIDIA RTX A5000 GPU. We train our method end-to-end using Adam [19] with a learning rate of $1e-3$ for the feature extractor, $5e-4$ for GeNeRF, and $2e-4$ for pose optimizer during pretraining. For fine-tuning, the learning rate is $5e-4$ for the feature extractor, $2e-4$ for GeNeRF, and $1e-5$ for the pose optimizer. We pretrain IBRNet in $250,000$ iterations and our method in $200,000$ iterations and finetune both IBRNet and our method in $60,000$ iterations. During pretraining, for our method, we only select 5 nearby views for pose correction and novel view rendering for efficiency. During fine-tuning and evaluation, we select 10 nearby views for both our method and IBRNet. The camera poses are updated by 4 iterations in a batch. Note that vanilla NeRF [28] and IBRNet [47] use a coarse network and a fine network to predict density value and color. However, BARF [20] and GARF [5] use a single coarse network. To make a fair comparison to them, we only train a coarse network for IBRNet and our method.

5.1. Experimental Results

We evaluated both the rendering quality for novel view synthesis and pose accuracy of our method. The code of GARF [5] is not publicly available during this work, and thus we cite the quantitative results from the original paper.

Novel View Synthesis. We use PSNR, SSIM [48] and LPIPS [56] as the metrics for novel view synthesis. The quantitative results are shown in Table 1. As we can see, the rendering quality of our method surpasses both BARF and GARF, and we even outperform IBRNet on the fern, flower, and fortress scenes with the unfair advantage that IBRNet has known camera poses (ours does not). The qualitative results on the LLFF dataset are given in Fig. 4. For IBRNet and our method, we show the per-scene finetuned visual results. For the scenes of *horns* and *orchids*, our method even renders images with higher quality than IBRNet. For the *room* scene, we can observe an obvious artifact for IBRNet (floor in the green zoomed-in area). This validated the effectiveness of our method. We also present the rendering results of IBRNet and our method on the ScanNet dataset in Fig. 5. Our method renders much better results than IBRNet. Furthermore, the differences in the camera poses visualized in Fig. 5 indicate ground truth camera poses are not accurate. Refer to our supplementary for more results.

Pose Accuracy. Since our DBARF does not recover absolute camera poses, we measure the accuracy of the predicted relative camera poses. Specifically, for each test scene, we select one batch of nearby views for all images and then recover the relative poses from the target view to each nearby view. The target view's camera pose is set to identity, then

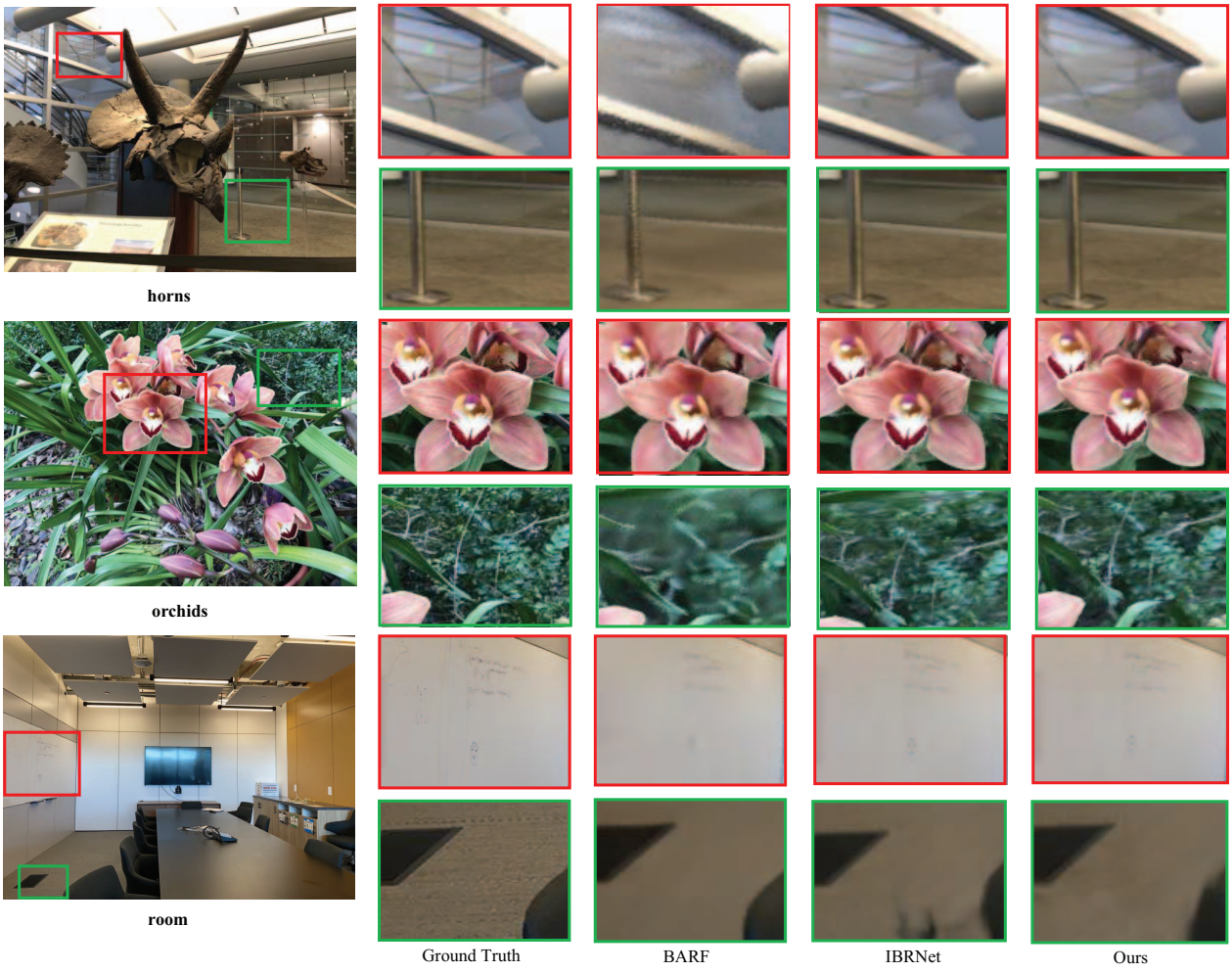

Figure 4. **The qualitative results on LLFF forward-facing dataset [27]**. We show the finetuned results for IBRNet and Ours.

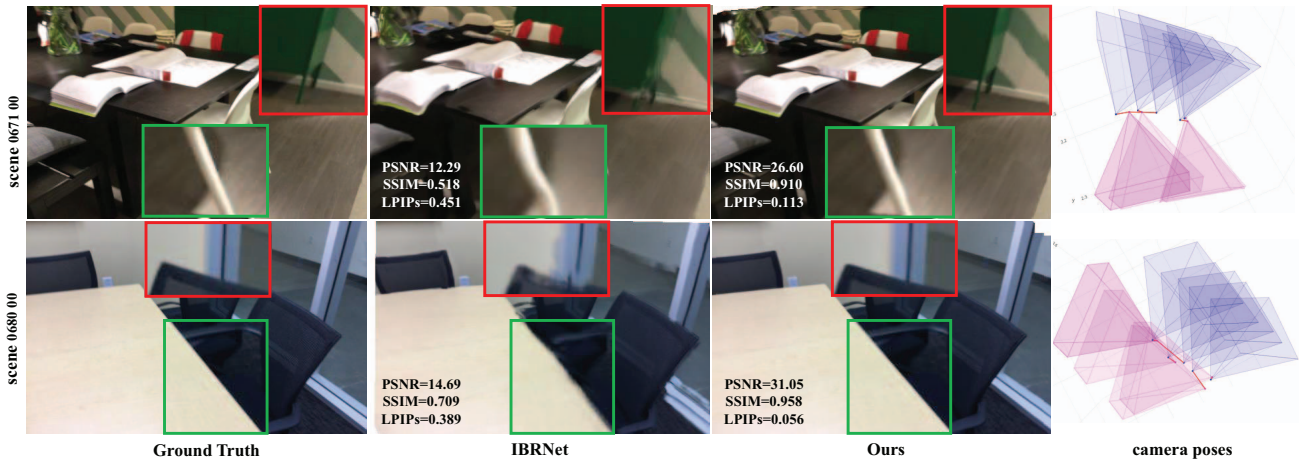

Figure 5. **The qualitative results on ScanNet dataset [6]**. We show the finetuned results for IBRNet and Ours. Red and blue are the pseudo ground truth (used by IBRNet) and the predicted camera poses of our method, respectively.

we estimate a similarity transformation to align all camera poses in that batch to ground truth by Umeyama [44]. The pose accuracy is measured by taking the average of all pose errors between the predicted relative poses and ground truth camera poses. The quantitative results are given in Table. 2.

Discussion of the Generalization of DBARF. To ablate the generalization ability of our method, we show the results of our method with and without fine-tuning in Tables. 1

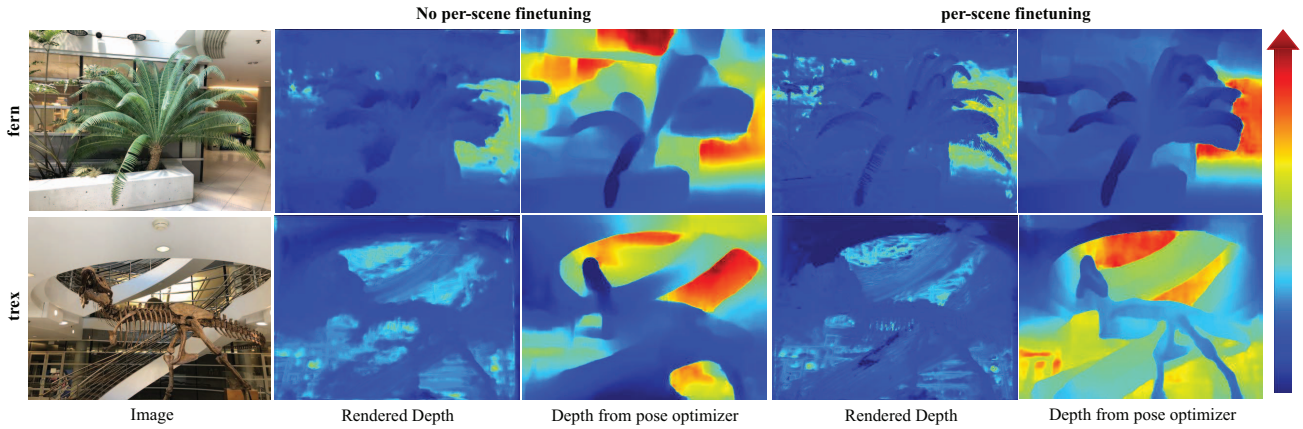

Figure 6. **Depth maps on LLFF forward-facing dataset [27]**. The rendered depth is computed from our GeNeRF after fine-tuning.

Scenes	PSNR ↑						SSIM ↑						LPIPS ↓					
	BARF [20]	GARF [5]	IBRNet [47]		Ours		BARF [20]	GARF [5]	IBRNet [47]		Ours		BARF [20]	GARF [5]	IBRNet [47]		Ours	
			✗	✓	✗	✓			✗	✓	✗	✓			✗	✓	✗	✓
fern	23.79	24.51	23.61	25.56	23.12	**25.97**	0.710	0.740	0.743	0.825	0.724	**0.840**	0.311	0.290	0.240	0.139	0.277	**0.120**
flower	23.37	**26.40**	22.92	23.94	21.89	23.95	0.698	0.790	0.849	0.895	0.793	**0.895**	0.211	0.110	0.123	0.074	0.176	**0.074**
fortress	29.08	29.09	29.05	31.18	28.13	**31.43**	0.823	0.820	0.850	0.918	0.820	**0.918**	0.132	0.150	0.087	0.046	0.126	**0.046**
horns	22.78	23.03	24.96	**28.46**	24.17	27.51	0.727	0.730	0.831	**0.913**	0.799	0.903	0.298	0.290	0.144	**0.070**	0.194	0.076
leaves	18.78	19.72	19.03	**21.28**	18.85	20.32	0.537	0.610	0.737	**0.807**	0.649	0.758	0.353	0.270	0.289	**0.137**	0.313	0.156
orchids	19.45	19.37	18.52	**20.83**	17.78	20.26	0.574	0.570	0.573	**0.722**	0.506	0.693	0.291	0.260	0.259	**0.142**	0.352	0.151
room	**31.95**	31.90	28.81	31.05	27.50	31.09	0.940	0.940	0.926	**0.950**	0.901	0.947	0.099	0.130	0.099	**0.060**	0.142	0.063
trex	22.55	22.86	23.51	**26.52**	22.70	22.82	0.767	0.800	0.818	**0.905**	0.783	0.848	0.206	0.190	0.160	**0.074**	0.207	0.120

Table 1. Quantitative results of novel view synthesis on LLFF [27] forward-facing dataset. For IBRNet [47] and our method, the results with (✔) and without (✗) per-scene fine-tuning are given.

Scenes	fern	flower	fortress	horns	leaves	orchids	room	trex
Rotation (✗)	9.96	16.74	2.18	6.08	12.98	5.90	8.76	10.09
Rotation (✔)	0.89	1.39	0.59	0.82	4.63	1.164	0.53	1.06
translation (✗)	2.00	1.56	1.06	2.45	2.56	5.13	5.48	8.05
translation (✔)	0.34	0.32	0.23	0.29	0.85	0.57	0.36	0.46

Table 2. Quantitative results of camera pose accuracy on LLFF [27] forward-facing dataset. Rotation (degree) and translation (scaled by 10^2, without known absolute scale) errors with (✔) and without (✗) per-scene fine-tuning are given.

and 2. We can observe that our method surpasses BARF and GARF on novel view synthesis even without per-scene fine-tuning. For pose accuracy, the rotation error of our method is less than 13° for most of the scenes in the LLFF dataset without fine-tuning, which is much cheaper than per-scene training from scratch. Our rotation error is less than 1.5° (except for the leaves scene) with fine-tuning. This proves that our method is generalizable across scenes. We also argue that the camera poses computed by COLMAP are only pseudo ground truth. Our camera poses are better than COLMAP since the rendering quality of our DBARF is better than IBRNet on the *fern*, *flower*, and *fortress* scenes.

Qualitative Analysis of Depth Maps. In Fig. 6, we present the depth maps computed from NeRF in Eq. (2b) (*i.e.* rendered depth maps), and those predicted by our pose optimizer. It can be observed that the depth maps from our pose optimizer are better than those from NeRF, which validates the rationalization of our analysis in Sec. 4.3.1, *i.e.* utilizing the rendered depth map from NeRF to compute the cost map may cause our DBARF to diverge. However, we can observe that while the pose optimizer generates a smoother depth map, NeRF can recover more accurate depth at scene details, especially the thin structures. We believe both depth maps can be improved under self-supervision: NeRF can learn better scene geometry, and the pose optimizer can predict more accurate camera poses with better-quality depth maps.

6. Conclusion

We analyzed the difficulties of bundle adjusting GeNeRFs, where existing methods such as BARF and its variants cannot work. Based on the analysis, we proposed DBARF that can bundle adjust camera poses with GeNeRFs, and can also be jointly trained with GeNeRFs end-to-end without ground truth camera poses. In contrast to BARF and GARF, which require expensive per-scene optimization and good initial camera poses, our proposed DBARF is generalizable across scenes and does require any initialization of the camera poses.

Acknowledgement. This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (Award Number: AISG2-RP-2020-016), and the Tier 2 grant MOE-T2EP20120-0011 from the Singapore Ministry of Education.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 5835–5844. IEEE, 2021. 3

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

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 14104–14113. IEEE, 2021. 2, 3

[4] Yu Chen, Shuhan Shen, Yisong Chen, and Guoping Wang. Graph-based parallel large scale structure from motion. *Pattern Recognit.*, 107:107537, 2020. 1

[5] 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, 6, 8

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2432–2443. IEEE Computer Society, 2017. 6, 7

[7] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(3):24:1–24:18, 2017. 6

[8] 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*, pages 224–236, 2018. 6

[9] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5491–5500. IEEE, 2022. 2

[10] Yang Fu, Ishan Misra, and Xiaolong Wang. Multiplane nerf-supervised disentanglement of depth and camera pose from videos. *CoRR*, abs/2210.07181, 2022. 3

[11] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 3827–3837. IEEE, 2019. 6

[12] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 684–691. IEEE Computer Society, 2004. 3

[13] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2492–2501. Computer Vision Foundation / IEEE, 2020. 2, 4

[14] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, Chengzhou Tang, and Ping Tan. DRO: deep recurrent optimizer for structure-from-motion. *CoRR*, abs/2103.13201, 2021. 5, 6

[15] Andrew Harltey and Andrew Zisserman. *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press, 2006. 2, 6

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE Computer Society, 2016. 5

[17] Nishant Jain, Suryansh Kumar, and Luc Van Gool. Robustifying the multi-scale representation of neural radiance fields. *CoRR*, abs/2210.04233, 2022. 3, 4

[18] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18344–18347. IEEE, 2022. 2, 3

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations*, 2015. 6

[20] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: bundle-adjusting neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 5721–5731. IEEE, 2021. 1, 2, 3, 4, 6, 8

[21] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 936–944. IEEE Computer Society, 2017. 5

[22] Yen-Chen Lin, 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*, pages 1323–1330. IEEE, 2021. 2

[23] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 5967–5977. IEEE, 2021. 1, 5

[24] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems 33*, 2020. 2

[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*, pages 7814–7823. IEEE, 2022. 2

[26] 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, pages 6331–6341. IEEE, 2021. 2, 3

[27] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4):29:1–29:14, 2019. 6, 7, 8

[28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. In *Computer Vision - ECCV 2020 - 16th European Conference*, volume 12346, pages 405–421. Springer, 2020. 1, 2, 3, 6

[29] 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, 2022. 2

[30] Pulak Purkait, Tat-Jun Chin, and Ian Reid. Neurora: Neural robust rotation averaging. In *Computer Vision - ECCV 2020 - 16th European Conference*, volume 12369, pages 137–154. Springer, 2020. 3

[31] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 77–85. IEEE Computer Society, 2017. 2

[32] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 14315–14325. IEEE, 2021. 2

[33] 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*, pages 4937–4946. Computer Vision Foundation / IEEE, 2020. 6

[34] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113. IEEE Computer Society, 2016. 1

[35] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, 2020. 2

[36] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. *CoRR*, abs/2207.10662, 2022. 2

[37] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8259–8269. IEEE, 2022. 2

[38] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5449–5459. IEEE, 2022. 2

[39] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 4

[40] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment networks. In *7th International Conference on Learning Representations*. OpenReview.net, 2019. 2, 5

[41] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *Computer Vision - ECCV 2020 - 16th European Conference*, volume 12347 of *Lecture Notes in Computer Science*, pages 402–419. Springer, 2020. 5

[42] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow (extended abstract). In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 4839–4843. ijcai.org, 2021. 5

[43] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - A modern synthesis. In *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms*, volume 1883, pages 298–372. Springer, 1999. 1

[44] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380, 1991. 7

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2

[46] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd E. Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5481–5490. IEEE, 2022. 2

[47] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699. Computer Vision Foundation / IEEE, 2021. 2, 3, 6, 8

[48] 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

[49] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *CoRR*, abs/2102.07064, 2021. 2, 4

[50] 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, 4

[51] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Computer Vision - ECCV 2018 - 15th European Conference*, volume 11212, pages 785–801. Springer, 2018. 4

[52] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 5732–5741. IEEE, 2021. 2

[53] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587. Computer Vision Foundation / IEEE, 2021. 2, 3

[54] Jiahui Zhang, Fangneng Zhan, Rongliang Wu, Yingchen Yu, Wenqing Zhang, Bai Song, Xiaoqin Zhang, and Shijian Lu. VMRF: view matching neural radiance fields. In *The 30th ACM International Conference on Multimedia*, pages 6579–6587. ACM, 2022. 3

[55] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *CoRR*, abs/2010.07492, 2020. 2

[56] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. 6