# GNeRF: GAN-based Neural Radiance Field without Posed Camera

Quan Meng[1] Anpei Chen[1] Haimin Luo[1] Minye Wu[1]
Hao Su[2] Lan Xu[1] Xuming He[1] Jingyi Yu[1]
[1] ShanghaiTech University [2] University of California, San Diego
{mengquan,chenap,luohm,wumy,xulan1,hexm,yujingyi}@shanghaitech.edu.cn {haosu}@eng.ucsd.edu

## Abstract

*We introduce GNeRF, a framework to marry Generative Adversarial Networks (GAN) with Neural Radiance Field reconstruction for the complex scenarios with unknown and even randomly initialized camera poses. Recent NeRF-based advances have gained popularity for remarkable realistic novel view synthesis. However, most of them heavily rely on accurate camera poses estimation, while few recent methods can only optimize the unknown camera poses in roughly forward-facing scenes with relatively short camera trajectories and require rough camera poses initialization. Differently, our GNeRF only utilizes randomly initialized poses for complex outside-in scenarios. We propose a novel two-phases end-to-end framework. The first phase takes the use of GANs into the new realm for coarse camera poses and radiance fields jointly optimization, while the second phase refines them with additional photometric loss. We overcome local minima using a hybrid and iterative optimization scheme. Extensive experiments on a variety of synthetic and natural scenes demonstrate the effectiveness of GNeRF. More impressively, our approach outperforms the baselines favorably in those scenes with repeated patterns or even low textures that are regarded as extremely challenging before.*

## 1. Introduction

Recovering 3D representations from multi-view 2D images is one of the core tasks in computer vision. Recently, significant progress has been made with the emergence of neural radiance fields methods (e.g., NeRF [29]), which represents a scene as a continuous 5D function and uses volume rendering to synthesize new views. Although NeRF and its follow-ups [23, 27, 53, 46] achieve an unprecedented level of fidelity on a range of challenging scenes, most of these methods rely heavily on knowing the accurate camera poses, which is yet a long-standing but challenging task. Conventional camera pose estimation process suffers in challenging scenes with repeated patterns, varying lighting, or few keypoints, and building on these methods adds additional uncertainty to the NeRF training process.

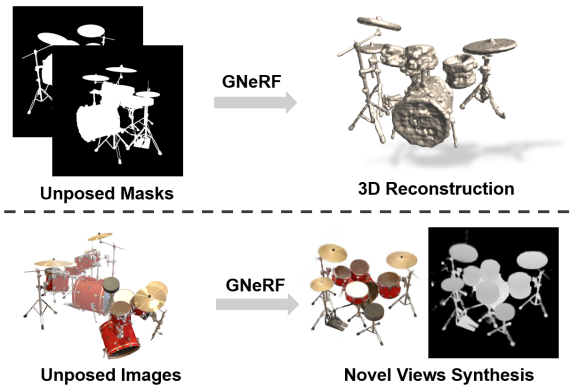


Figure 1. Our approach estimates both camera poses and neural radiance fields using only randomly initialized poses in complex scenarios, even at the extreme case when the input views are only texture-less gray masks.

To explore the possibilities of alleviating the dependence on accurate camera pose information, recently, iNeRF [52] and NeRF−− [47] attempt to optimize camera pose along with other parameters when training NeRF. While certain progress has been made, both of them can only optimize camera poses when relatively short camera trajectories with reasonable camera poses initialization are available. It is worth noting that, NeRF−− is limited to roughly forward-facing scenes, the focus of iNeRF is camera pose estimation but not radiance field estimation, and it assumes a trained NeRF which in turn requires known camera poses as supervision. When greater viewpoint uncertainty presents, camera poses estimation is extremely challenging and prone to falling into local minima.

To this end, we propose **GNeRF**, a novel algorithm that can estimate both camera poses and neural radiance fields when the cameras are initialized at random poses in complex scenarios. Our algorithm has two phases: the first phase gets coarse camera poses and radiance fields with adversarial training; the second phase refines

them jointly with a photometric loss. Taking the use of Generative Adversarial Networks (GANs) into the new realm of camera poses estimation, we extend the NeRF model to jointly optimize 3D representation and camera poses to complex scenes with large displacements. Instead of directly propagating the photometric loss back to the camera pose parameters, which is sensitive to challenging conditions (e.g., less texture and varying lighting) and apt to fall into local minima, we propose a hybrid and iterative optimization scheme. Our learning pipeline is fully differentiable and end-to-end trainable, allowing our algorithm to perform well in the challenging scenes where COLMAP-based [38] methods completely fail due to challenges such as repeated patterns, low textures, noise, even at the extreme cases when the input views are a collection of gray masks, as is shown in Fig. 1. Furthermore, our method can predict new poses of images belonging to the same scene through the trained inversion network without tedious per-scene pose estimation (e.g., COLMAP-like methods) or time-consuming gradient-based optimization (e.g., iNeRF and NeRF−−).

We experiment with our GNeRF on a variety of synthetic and natural scenes. We demonstrate results on par with COLMAP-based NeRF methods in regular scenes; more impressively, our method outperforms the baselines in general cases that are regarded as extremely challenging before.

## 2. Related Works

**Neural 3D Representations** Classic approaches largely rely on discrete representations such as meshes [12], voxel grids [6, 50, 42], point clouds [9]. Recent neural continuous implicit fields are gaining increasing popularity, due to their capability of representing high level of details. But these methods need costly 3D annotations. To bridge the gap between 2D information and 3D representations, differential rendering tackles such integration for end-to-end optimization by obtaining useful gradients of the rendering process. Liu *et al*. [24] proposes the first usage of neural implicit surface representations in differentiable rendering. Mildenhall *et al*. [29] proposes differentiable volume rendering and achieves more view-consistent reconstructions of the scene. However, these methods assume accurate camera poses as a prerequisite.

Recently, several methods attempt to lessen the recline on precomputed camera poses. Adding noise to ground-truth camera poses, IDR [51] produces accurate 3D surface reconstruction by simultaneously learning 3D representation and camera poses. Adding random offset to camera poses, iNeRF [52] performs pose estimation by inverting a trained neural radiance field. Initializing camera poses to the identity matrix, NeRF−− [47] demonstrates satisfactory novel view synthesis results in forward-facing scenes by optimizing camera parameters and radiance field jointly. In contrast to these methods, our method does not depend on camera pose initialization and is not sensitive to challenging scenes.

**Pose Estimation** Traditional techniques typically rely on Structured-from-Motion (SfM) [10, 1, 48, 38] which extracts local descriptor (e.g.SIFT [26]), performs matching to find 2D-3D correspondence, estimates candidate poses and then chooses the best pose hypothesis by RANSAC [11]. Other retrieval-based methods [14, 36, 41, 7] find images similar to the query image and establish the 2D-3D correspondence efficiently by matching the query image against the database images. Recently, deep learning-based methods attempt to regress the camera pose directly from 2D images. PoseNet [19] is the firstly end-to-end approach that adopts a modified truncated GoogleNet as pose regressor. Different architectures [45, 28, 32, 49] or pose losses [18, 3] are utilized which lead to a significant improvement. Auxiliary tasks such learning relative pose estimation [44, 37] or semantic segmentation [37] lead to a further improvement. For a better generalization of the network, hybrid pose learning methods shift the learning towards local or related problems: [22, 2] propose to regress the relative pose of a query image to the known poses based on image retrieval.

Our method belongs to deep learning-based methods, but in contrast to these approaches, our method is trained per scene in a self-supervised manner rather than pretrained on a large dataset with pose annotations.

**3D-Aware Image Synthesis** Generative adversarial nets, or more generally the paradigm of adversarial learning, have led to significant progress in various image synthesis tasks [30, 17, 40]. But these methods operate on 2D space of pixels, ignoring the 3d structure of our natural scene. 3D-aware image synthesis correlates 3D model with 2D images, enabling explicit modification of 3D model [33, 13, 13, 34, 39, 35, 5, 4]. Earlier 3D-aware image synthesis methods like RenderNet [33] introduces rendering convolutional networks with a projection unit that can render 2D images from 3D shapes. PLATONICGAN [13] use a voxel-based representation and a family of differentiable rendering layers to discover the 3D structure of an object from an unstructured collection of 2D images. HoloGAN [34] use deep voxels representation and learn it also without any 3D shapes supervision. For these methods, the combination of differentiable rendering layers and implicit 3D representation can lead to entangled latents and destroy multi-view consistency. The most recent and relevant to ours are GRAF [39], GIRFFE [35] and pi-GAN [4], with the expressiveness of NeRF, these methods allow disentangled shape, appearance modification of the generated objects. All these methods require abundant data to train the generative network. Conversely, we learn the coarse

generative network with limited data and refine it with photometric constraints.

## 3. Preliminary

We first introduce the basic camera and scene representation, as well as notations for our method in this section.

**Camera Pose** Formally, we represent the camera pose/extrinsic parameters based on its position/location in 3D space and its rotation from a canonical view. For the camera position, we simply adopt a 3D vector in Euclidean space, denoted as $\mathbf{t} \in \mathbb{R}^3$. For the camera rotation, the widely-used representations such as quaternions and Euler angles are discontinuous and difficult for neural networks to learn. Following the seminal work [55], we use a continuous 6D vector $\mathbf{r} \in \mathbb{R}^6$ to represent 3D rotations, which is more suitable for learning. Concretely, given a rotation matrix $\mathbf{R} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3\end{bmatrix} \in \mathbb{R}^{3\times3}$, we compute the rotation vector $\mathbf{r}$ by dropping the last column of the rotation matrix.

From the 6D vector, we can also recover the original rotation matrix using a Gram-Schmidt-like process, in which the last column is computed by a generalization of the cross product to three dimension [55].

**NeRF Scene Representation** We adopt the NeRF framework to represent the underlying 3D scene and image formation, which encodes a scene as continuous volumetric radiance field of color and density [29]. Specifically, given a 3D location $\mathbf{x} \in \mathbb{R}^3$ and 2D viewing direction $\mathbf{d} \in [-\pi, \pi]^2$ as inputs, the NeRF model defines a 5D vector-valued function $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ based on an MLP network, where its outputs are an emitted color $\mathbf{c} \in \mathbb{R}^3$ and volume density $\sigma$, and $\Theta$ are network parameters. To render an image from a NeRF model, the NeRF model follows the classical volume rendering principles [16].

For each scene, the NeRF framework learns a separate neural representation network with a dataset of RGB images of the scene, the corresponding camera poses and intrinsic parameters, and scene bounds. Concretely, given a dataset of calibrated RGB images $\mathcal{I} = \{I_1, I_2, \cdots, I_n\}$ of a single scene, the corresponding camera poses $\Phi = \{\phi_1, \phi_2, \cdots, \phi_n\}$ and a differentiable volume renderer $G$, the NeRF model optimizes the continuous volumetric scene function $F_\Theta$ by a photometric loss as below,

$$\mathcal{L}_N(\Theta, \Phi) = \frac{1}{n}\sum_{i=1}^{n} \|I_i - \hat{I}_i\|_2^2, \quad \hat{I}_i = G(\phi_i; F_\Theta) \tag{1}$$

## 4. Methods

Our goal is to learn a NeRF model $F_\Theta$ from $n$ uncalibrated images $\mathcal{I}$ of a single scene without knowing their camera poses. To this end, we treat the camera poses of those images $\Phi$ as values of a latent variable, and propose an iterative learning strategy that jointly estimates the camera poses and learns the NeRF model. As the overview of our approach in Fig. 2 illustrates, the key ingredient of our method is a novel NeRF estimation strategy based on an integration of an adversarial loss and an inversion network (Phase A). This enables us to generate a coarse estimate of the implicit scene representation $F_\Theta$ and the camera poses $\Phi$ from a learned camera pose encoder. Given the initial estimate, we use photometric loss to refine the NeRF scene model and those camera poses (Phase B). Additionally, we develop a regularized NeRF optimization step that refines the NeRF scene model and those camera poses. Interestingly, our pose-free NeRF estimation process can also further improve the refined scene representation and camera poses from the regularized NeRF optimization step. Consequently, our learning algorithm iterates over the afore-mentioned NeRF estimation and optimization step to further overcome local minima between the two phases (AB...AB).

In the following, we first present our pose-free NeRF estimation procedure in Sec 4.1, and then introduce the regularized NeRF optimization step in Sec 4.2. The overall iterative learning process is detailed in Sec 4.3.

### 4.1. Pose-free NeRF Estimation

As the initial stage of our method, we do not have a reasonable camera pose estimation for each image. Our goal for this stage is to predict a rough pose for each image and also learn a rough radiance field of the scene. As shown in the left of Fig. 2, we use adversarial learning to achieve the goals. Our architecture contains two parts: a generator $G$ and a discriminator $D$. Taking a camera pose $\phi$ as input, the generator $G$ will synthesize the image observed at the view by predicting the neural radiance field and performing NeRF-like volume rendering. The set of synthesized images from many sampled camera poses will be decomposed into patches and compared against the set of real image patches by the discriminator $D$. $G$ and $D$ are trained adversarially, as is done by the classical GAN work. This adversarial training allows us to roughly learn the radiance field and estimate camera poses at random initialization.

Formally, we minimize a distribution distance between the real image patches $P_d(I)$ from the training set $\mathcal{I}$ and the generated image patches $P_g(I|\Theta)$, which are defined as below:

$$\Theta^* = \arg\min_\Theta Dist\left(P_g(I|\Theta)||P_d(I)\right) \tag{2}$$

$$P_g(I|\Theta) = \int_\phi G(\phi; F_\Theta)P(\phi)d\phi \tag{3}$$

To minimize the distribution distance, we adopt the following GAN learning framework based on an adversarial

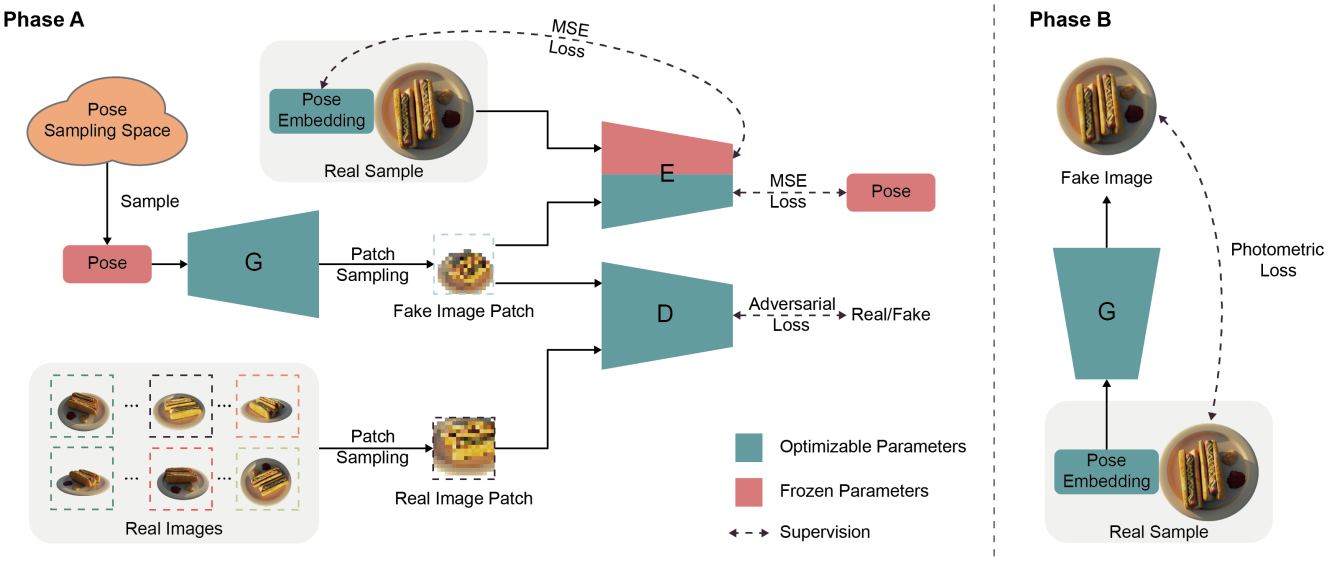


Figure 2. **The pipeline of GNeRF.** Our pipeline learns the radiance fields and camera poses jointly in two phases. In phase A, we randomly sample poses from a predefined poses sampling space and generate corresponding images with the NeRF (G) model. The discriminator (D) learns to classify real and fake image patches. The inversion network (E) takes in the fake image patches and learns to output their poses. Then, with the inversion network's parameters frozen, we optimize the pose embeddings of real images in the dataset. In phase B, we utilize the photometric loss to refine radiance fields and pose embeddings jointly. We follow a hybrid and iterative optimization strategy of the pattern 'A $\rightarrow$ AB... AB $\rightarrow$ B' in the training process.

loss $\mathcal{L}_A$ defined as follows:

$$\min_{\Theta}\max_{\eta}\mathcal{L}_A(\Theta,\eta) = \mathbb{E}_{I\sim P_d}[\log(D(I;\eta))] + \mathbb{E}_{\hat{I}\sim P_g}[\log(1-D(\hat{I};\eta))] \quad (4)$$

where $\eta$ are the network parameters of the discriminator $D$ and $\mathbb{E}$ denotes expectation.

Along with the two standard components, we train an inversion network $E$ that maps images to the corresponding camera poses. We train the inversion network with the pairs of randomly sampled camera poses and generated image patches. The inputs of the inversion network are from uniformly sampled and fixed-size image patches, and the outputs are the corresponding camera poses. Formally, denote the parameters of the inversion network $E$ as $\theta_E$, and its loss function can be written as,

$$\mathcal{L}_E(\theta_E) = \mathbb{E}_{\phi\sim P(\phi)}\left[\|E(G(\phi;F_\Theta);\theta_E)-\phi\|_2^2\right] \quad (5)$$

We note that the inversion network is trained in a self-supervised manner, which exploits the synthetic image patches and their corresponding camera poses as the training data. With the increasingly better-trained generator, the inversion network would be able to predict camera poses for real image patches. After the overall training is converged, we apply the inverse network to generate camera pose estimates $\{\phi_i^{'} = E(I_i), I_i \in \mathcal{I}\}$ for the training image set $\mathcal{I}$.

### 4.2. Regularized NeRF Optimization

After the pose-free NeRF estimation step, we obtain an initial NeRF model and camera pose estimates for the training images. Due to the sparse sampling of the input image patches and the constrained capability of the inversion network, neither the NeRF representation nor the estimated camera poses $\Phi' = \{\phi_i^{'}\}$ are accurate enough. However, they provide a good initialization for the overall training procedure, which allows us to introduce a refinement step for the NeRF model and camera poses.

We note that existing work like NeRF$--$ can search a limited scope in the pose space during NeRF optimization. However, the pose optimization problem in the standard NeRF model is highly non-convex, and hence their results strongly depend on camera pose initialization and are still insufficient for our challenging test scenarios. To mitigate this issue, we develop a regularized NeRF optimization process, which regularizes the gradient descent-based model optimization by the pose prediction from the learned inversion network. Intuitively, with the adversarial training of the NeRF model, the domain gap between synthesized fake images and true images is narrowing, so those pose predictions provide a reasonable and effective constraint for the joint radiance fields and pose optimization.

Formally, we define a hybrid loss function $\mathcal{L}_R$ that combines the photometric reconstruction errors and an L2 loss penalizing the deviation from the predictions of the

inversion network, which can be written as below,

$$\mathcal{L}_R(\Theta, \Phi) = \mathcal{L}_N(\Theta, \Phi) + \frac{\lambda}{n} \sum_{i=1}^{n} \|E(I_i; \theta_E) - \phi_i\|_2^2 \quad (6)$$

where $\lambda$ is the weighting coefficient and $\mathcal{L}_N(\Theta, \Phi)$ is the NeRF loss defined in Eqn. 1.

### 4.3. Iterative Learning Strategy

Due to the notorious non-convex property of camera poses optimization, we propose an iterative learning strategy to further improve the quality of the NeRF model and pose estimation. Such a design is based on our empirical findings that the pose-free NeRF estimation can also improve the refined NeRF and camera poses from the regularized optimization step. As a result, our overall training procedure additionally refines the radiance field and camera poses by iterating over the pose-free NeRF estimation step (Sec 4.1) and the regularized NeRF Optimization step (Sec 4.2).

### 4.4. Training

More concretely, we initially sample camera poses $\phi$ randomly from the prior pose distribution. To train the generative radiance field, we follow a similar patch sampling strategy as GRAF [39], i.e., randomly sample patch patterns with dynamic scale but fixed size from the image domain. Except for the input image patch of the inversion network, we uniformly sample a fixed size sparse patch to represent the whole image with which we estimate the corresponding camera pose. We progressively scale the camera intrinsics to maximize receptive field at the beginning to stabilize training and decrease it along the training to concentrate on fine details.

### 4.5. Implementation Details

We adopt the network architecture of the orginal NeRF [29] and its hierarchical sampling strategy to our generator. Differently, we only utilize the same MLPs in hierarchical sampling strategy to ensure the pose spaces of "coarse" and "fine" networks are aligned during camera pose optimization. The numbers of sampled points of both coarse sampling and importance sampling are set to 64. The discriminator network follows GRAF [39], in which instance normalization [43] over features and spectral normalization [31] over weights are applied. We borrow the Vision Transformer Network [8] to build our inversion network, whose last layer is modified to output a camera pose.

Before network optimization, we initialize all camera extrinsics to be an identity matrix. We use RMSprop [21] algorithm to optimize the generator and the discriminator with learning rates of 0.0005 and 0.0001, respectively. As for the inversion network and camera poses, we use Adam [20] algorithm with learning rates of 0.0001 and 0.005.

## 5. Experiments

Here we compare our method with other approaches which require camera poses or a coarse camera initialization on view synthesis task and evaluate our method in various scenarios. We run our experiments on a PC with Intel i7-8700K CPU, 32GB RAM, and a single Nvidia RTX TITAN GPU, where our approach takes 30 hours to train the network on a single scene.

### 5.1. Novel View Synthesis Comparison

We firstly compare pose estimation quality via novel view synthesis task using Synthetic-NeRF [29] and DTU [15] datasets. For the Synthetic-NeRF dataset, we use six scenes rendered from cameras arranged on an upper hemisphere, on each of which we take all 100 camera images as training data and randomly sample 8 images for testing following the same dataset splits as the original NeRF [29]. The selection of these test samples is based on maximizing their mutual angular distance between views so that test samples can cover different perspectives of the object as much as possible. We mainly conduct the comparison on the Synthetic-NeRF dataset. To further explore the boundary of our approach, we also utilize five representative scenes on the DTU dataset with 43 images for training and 6 images for testing on each scene.

Specifically, we use two other approaches in our comparison. One is original NeRF [29] with precalibrated camera poses from COLMAP [38], denoted by **C+n**. The other one **C+r** is similar to NeRF−− [47], which is initialized by precalibrated camera poses and optimizes camera poses via gradient descent approach in NeRF-based architecture. For evaluation, we need to estimate the camera poses of the test view images. Since our method does not require pose initialization, the camera poses of test view for rendering is directly estimated by our well-trained model.

Our method outperforms the **C+n** in challenging scenes while achieving similar results on regular scenes. These challenging scenes do not have enough keypoints for pose estimation, which makes Nerf which needs precise poses as input fails to synthesis good results. **C+r** have a better performance than **C+n**'s. However, limited by the poor pose initialization, **C+r** can not produce the same performance as ours in most challenging scenes.

As in Fig. 3, we show the visualization comparison with methods on the Synthetic-NeRF and DTU datasets. We see that our method, which does not require pre-computed camera poses, outperforms other COLMAP-based NeRF methods on challenging scenes that only have few reliable keypoints, e.g., Synthetic-NeRF dataset,

| Data | Scene | ↑ PSNR | | | | ↑ SSIM | | | | ↓ LPIPS | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **C+n** | **C+r** | Ours | GT | **C+n** | **C+r** | Ours | GT | **C+n** | **C+r** | Ours | GT |
| Synthetic-NeRF | Chair | 14.05 | 14.89 | **31.30** | 32.84 | 0.80 | 0.83 | **0.94** | 0.97 | 0.31 | 0.17 | **0.08** | 0.04 |
| | Drums | 12.07 | 11.96 | **24.30** | 26.71 | 0.74 | 0.71 | **0.90** | 0.93 | 0.43 | 0.34 | **0.13** | 0.07 |
| | Hotdog | 12.80 | 29.40 | **32.00** | 29.72 | 0.79 | 0.95 | **0.96** | 0.95 | 0.32 | **0.06** | 0.07 | 0.04 |
| | Lego | 9.80 | 26.65 | **28.52** | 31.06 | 0.69 | 0.86 | **0.91** | 0.95 | 0.45 | 0.11 | **0.09** | 0.04 |
| | Mic | 13.10 | 17.92 | **31.07** | 34.65 | 0.81 | 0.86 | **0.96** | 0.97 | 0.35 | 0.22 | **0.06** | 0.02 |
| | Ship | 10.55 | 26.33 | **26.51** | 28.97 | 0.64 | 0.72 | **0.85** | 0.82 | 0.45 | 0.29 | **0.21** | 0.15 |
| DTU | Scan4 | 22.05 | **24.23** | 22.88 | 25.52 | 0.69 | 0.72 | **0.82** | 0.78 | 0.32 | **0.20** | 0.37 | 0.18 |
| | Scan48 | 6.718 | 10.40 | **23.25** | 26.20 | 0.52 | 0.62 | **0.87** | 0.90 | 0.65 | 0.60 | **0.21** | 0.21 |
| | Scan63 | **27.80** | 26.61 | 25.11 | 32.19 | **0.90** | **0.90** | **0.90** | 0.93 | 0.21 | **0.19** | 0.29 | 0.24 |
| | Scan104 | 10.52 | 13.92 | **21.40** | 23.35 | 0.48 | 0.55 | **0.76** | 0.82 | 0.60 | 0.59 | **0.44** | 0.36 |

Table 1. **Quantitative comparison among COLMAP-based NeRF [29] (C+n), COLMAP-based NeRF with additional refinement (C+r) and ours on the Synthetic-NeRF [29] dataset and DTU [15] dataset.** We report PSNR, SSIM and LPIPS metrics to evaluate novel view synthesis quality. Our method without posed camera generates novel views on par with COLMAP-based NeRF and is more robust to challenging scene where COLMAP-based NeRF fails.

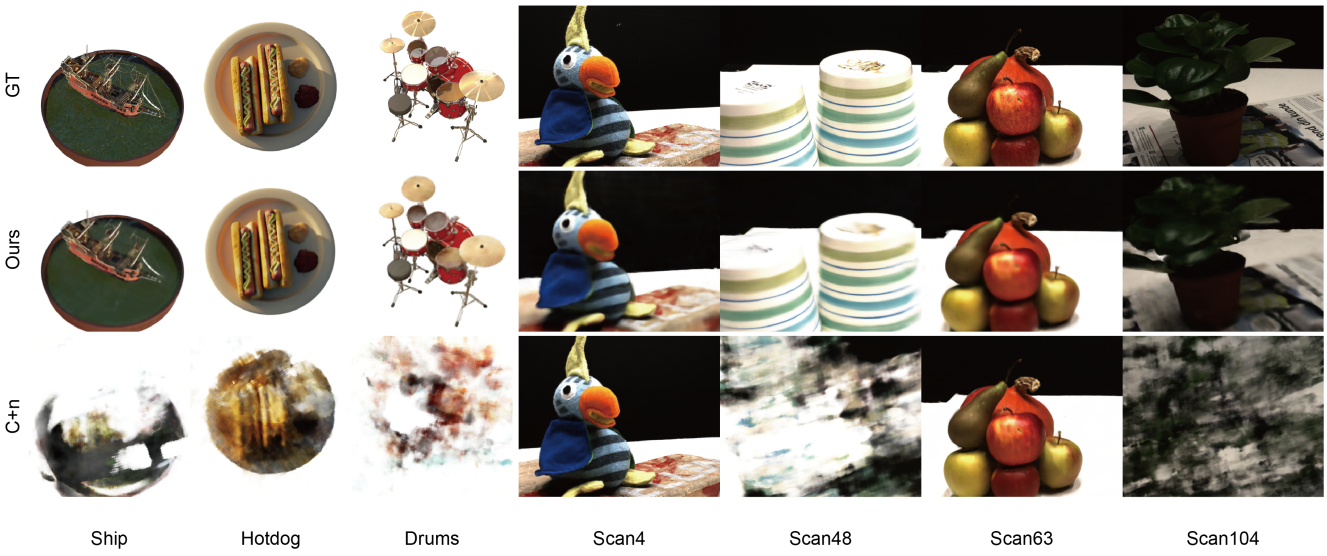


Figure 3. **Qualitative comparison between COLMAP-based NeRF (C+n) and ours on novel view synthesis quality on Synthetic-NeRF [29] dataset and DTU [15] dataset.**

scan48, and scan104 while achieving satisfactory results on par with it on the other regular scenes with enough keypoints. We show the quantitative performance of all the three methods in Tab. 1 on the Synthetic-NeRF and DTU datasets. We notice that our method achieves the highest performance in all the synthetic scenes and these challenging scenes in the realistic dataset, except under the setting with ground-truth poses. For other scenes, our method generates satisfactory results on par with the COLMAP-based NeRF methods.

Additionally, to further demonstrate our architecture's ability to learn the high-quality 3D representation without camera poses, we also compare with the state-of-the-art 3D surface reconstruction method, IDR [51], by comparing the rendering quality. Note that the IDR method requires image masks and a rough camera initialization, while our method does not need them. We follow the same setting that training the model on 49 images of each scene and report the mean PSNR as evaluation metrics. To have a relatively fair comparison, we report both the PSNR computed on the whole image and within the mask, which is the same evaluation protocol as IDR. The qualitative results are in Tab. 2 and the quantitative results are in Fig. 4. It can be seen that our volume-rendering-based method produces more natural images, while IDR produces results with more artifacts and fewer fine details.

| Methods | Scan48 | Scan97 | Scan104 |
|---|---|---|---|
| IDR [51] | 21.17 | 17.42 | 12.26 |
| Ours(masked) | 20.40 | 19.40 | 19.81 |
| Ours | **25.71** | **24.52** | **25.70** |

Table 2. **Quantitiative rendering quality comparison between IDR and ours on DTU [15] dataset.** The evaluation metric is PSNR.

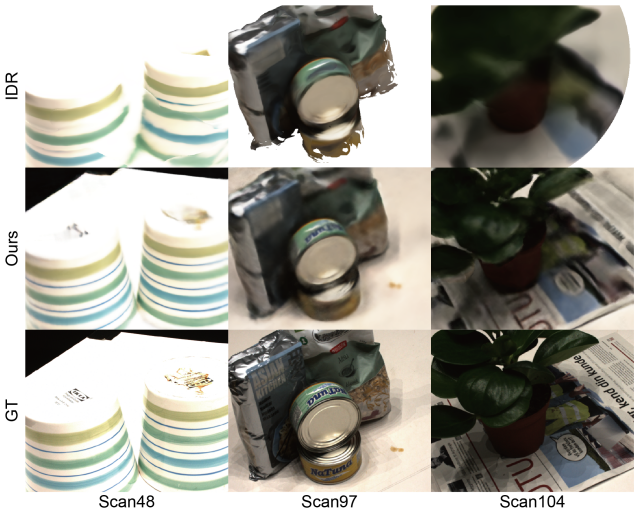


Figure 4. **Qualitative rendering quality comparison between IDR [51] and ours on DTU dataset.**

| Scene | COLMAP [38] | | Ours | |
|---|---|---|---|---|
| | ↓ Rot(deg) | ↓ Trans | ↓ Rot(deg) | ↓ Trans |
| Chair | 2.261 | 0.724 | **0.363** | **0.018** |
| Drums | 5.561 | 0.300 | **0.204** | **0.010** |
| Hotdog | 3.328 | 0.218 | **2.349** | **0.122** |
| Lego | 13.471 | 0.821 | **0.430** | **0.023** |
| Mic | 3.949 | 0.260 | **1.865** | **0.031** |
| Ship | 3.821 | 0.223 | **3.721** | **0.176** |

Table 3. **Quantitative camera poses accuracy comparison between COLMAP and ours on Synthetic-NeRF [29] dataset.** We report camera rotation difference (Rot) and translation difference (Trans).

## 5.2. Evaluation

We evaluate the accuracy of camera poses estimation on the Synthetic-NeRF dataset. In Tab. 3, we report the translation and rotation difference computed with the ATE toolbox [54]. In the Synthetic-NeRF dataset, the camera poses estimated by our method outperform the COLMAP [38] method whose performance is sensitive to scenes with repeated patterns (lego, chair) and few textures (drums).

In Tab. 4 and Fig. 5, we show an ablation study over different components of our model. Our full architecture of the combination of adversarial training, inversion network, and photometric loss achieves the best performance. Without either the adversarial loss or the inversion network, the model is incapable to learn correct geometry; without the photometric loss, the model is only capable to get coarse

| Adver | Inver | Photo | ↑ PSNR | ↓ Rot(deg) | ↓ Trans |
|---|---|---|---|---|---|
| | ✓ | ✓ | 19.31 | 108.22 | 2.53 |
| ✓ | | ✓ | 13.82 | 132.85 | 3.05 |
| ✓ | ✓ | | 20.60 | 5.91 | 0.24 |
| ✓ | ✓ | ✓ | **31.30** | **0.36** | **0.02** |

Table 4. **Ablation study.** We report PSNR, camera rotation difference (Rot), and translation difference (Trans) of the full model (the last row) and three configurations by removing the adversarial loss (Adver), the inversion network (Inver), and the photometric loss (Photo), respectively. Removing adversarial loss or inversion network prevents the model from learning reasonable camera poses. Removing photometric loss prevents the model from getting accurate camera poses.

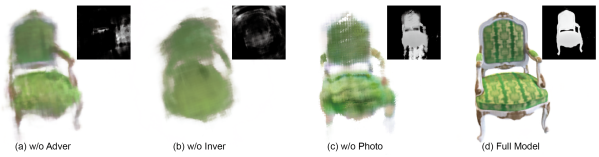


Figure 5. **Ablation study.** We visualize novel view RGB images and depth maps of the four different configurations.

| A, B | A, AB...AB, B | ↑ PSNR | ↓ Rot(deg) | ↓ Trans |
|---|---|---|---|---|
| ✓ | | 29.23 | 0.592 | 0.034 |
| | ✓ | **31.30** | **0.363** | **0.018** |

Table 5. **Optimization schemes analysis.** We compare two optimization schemes: 'A, B'; 'A, AB...AB, B'. The additional iterative optimization enables our model to achieve much better results.

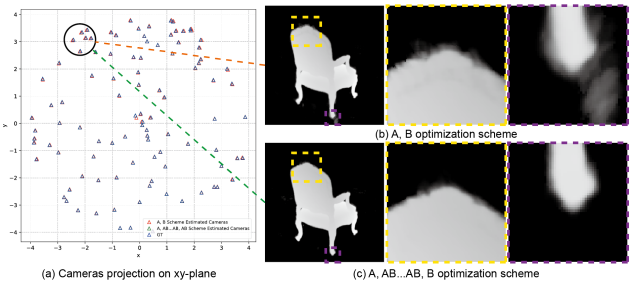


Figure 6. **Optimization schemes analysis.** On the left, we visualize the projection of camera poses on $xy$-plane of the obtained image from the two optimization schemes. On the right, we show depth maps of the view in the circled camera region and two part details (yellow and purple insets) of the two schemes.

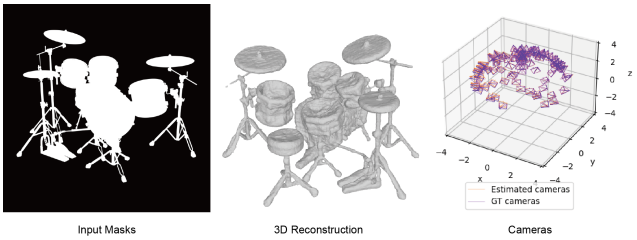


Figure 7. **3D reconstruction and camera pose estimation from a collection of masks without pose information.**


Figure 8. **Image Noise Analysis.** Despite adding intense noise on training images, our method is able to learn accurate radiance fields and camera poses of the noisy images while COLMAP-based NeRF methods completely fail.

radiance fields.

In Tab. 5 and Fig. 6, we analyze different optimization schemes. We represent Phase A and Phase B as A and B respectively. Our adopted iterative optimization scheme on the pattern 'A, AB...AB, B' achieves much higher image quality and camera pose accuracy than that of 'A, B'. In Fig. 6, the iterative optimization scheme gets much finer geometry along the edge, and the estimated camera poses align much closer to the ground-truth camera poses. These results demonstrate that the iterative learning strategy can further help overcome local minima.

### 5.3. Applications

**3D Reconstruction from Masks without Pose** In Fig. 7, we learn 3D representation and camera poses from a collection of unposed masks by optimizing the radiance fields and camera poses simultaneously. We then extract the 3D representation using the marching cubes algorithm [25]. This case further demonstrates that our architecture can estimate camera poses from high-level features of an image without reliance on keypoints or textures, which is fundamentally different from conventional pose estimation methods. This ability of our method can be applied to other applications, such as the task of reconstructing transparent objects whose visual appearance is too complex for image-based reconstruction. Since it is much easier to obtain the masks of their shapes either by semantic segmentation tools or other sensors.

**Image Noise Analysis** In Fig 8, we test our method on images by adding intense noise. The COLMAP-based NeRF methods completely fail to estimate the camera poses of images with large noise, leading to failure of learning the radiance fields. In contrast, our method is not sensitive to the noise and still able to learn radiance fields and estimate the poses of the noisy image.

## 6. Discussion and Conclusion

**Limitations.** Firstly, our method does not depend on camera pose initialization, but it does require a reasonable camera pose sampling distribution. For different datasets, we rely on a camera sampling distribution not far from the true distribution to alleviate the difficulties of learning the adversarial loss for radiance field estimation. This could potentially be mitigated by learning a pose sampling network to map a standard distribution to the underlying poses distribution automatically. Secondly, jointly optimizing camera poses and scene representation is a challenging task and opt to fall in local minima. Although in real datasets, we achieve good novel view synthesis quality on par with NeRF if the accurate camera poses present, our optimized camera poses are still not so accurate as of the COLMAP when there are sufficient amounts of reliable keypoints. This might be due to that our inversion network, which maps images to camera poses, could only take in $64 \times 64$ image patches for computation efficiency, and important information for fine camera pose estimation might have been discarded. This might be fixed by increasing the input patch size or importance sampling.

**Conclusion.** We have presented GNeRF, a GAN-based framework to reconstruct neural radiance fields and estimate camera poses when the camera poses are completely unknown and scene conditions can be complicated. Our framework is fully differentiable and end-to-end trainable. Specifically, our first phase enables GAN-based joint optimization for the 3D representation and the camera poses, and our hybrid and iterative scheme in the second phase would further refine the results robustly. Extensive experiments demonstrate the effectiveness of our approach. Impressively, our approach has demonstrated promising results on those scenes with repeated patterns or even low textures, which have been regarded as extremely challenging before. We believe our approach is a critical step towards the more general neural scene modeling goal using less human-crafted priors.

## References

[1] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001. 2

[2] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–767, 2018. 2

[3] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018. 2

[4] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. *arXiv preprint arXiv:2012.00926*, 2020. 2

[5] Anpei Chen, Ruiyang Liu, Ling Xie, and Jingyi Yu. A free viewpoint portrait generator with dynamic styling. *arXiv preprint arXiv:2007.03780*, 2020. 2

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2

[7] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

[9] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2

[10] Olivier Faugeras, Quang-Tuan Luong, and Theo Papadopoulo. *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2001. 2

[11] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[12] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019. 2

[13] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9984–9993, 2019. 2

[14] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606. IEEE, 2009. 2

[15] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 5, 6, 7

[16] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 3

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2

[18] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017. 2

[19] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 5

[22] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 929–938, 2017. 2

[23] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020. 1

[24] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. *arXiv preprint arXiv:1911.00767*, 2019. 2

[25] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 8

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[27] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv preprint arXiv:2008.02268*, 2020. 1

[28] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 879–886, 2017. 2

[29] 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. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 1, 2, 3, 5, 6, 7

[30] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[31] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5

[32] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1525–1530. IEEE, 2017. 2

[33] Thu Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yong-Liang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. *arXiv preprint arXiv:1806.06575*, 2018. 2

[34] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 2

[35] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. *arXiv preprint arXiv:2011.12100*, 2020. 2

[36] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 2

[37] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018. 2

[38] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2, 5, 7

[39] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv preprint arXiv:2007.02442*, 2020. 2, 5

[40] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4570–4580, 2019. 2

[41] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 3, pages 1470–1470. IEEE Computer Society, 2003. 2

[42] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 2

[43] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 5

[44] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6939–6946. IEEE, 2018. 2

[45] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 627–637, 2017. 2

[46] 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. *arXiv preprint arXiv:2102.13090*, 2021. 1

[47] 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. 1, 2, 5

[48] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 2

[49] Jian Wu, Liwei Ma, and Xiaolin Hu. Delving deeper into convolutional neural networks for camera relocalization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5651. IEEE, 2017. 2

[50] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2

[51] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction with implicit lighting and material. *arXiv e-prints*, pages arXiv–2003, 2020. 2, 6, 7

[52] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020. 1, 2

[53] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 1

[54] 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)*, pages 7244–7251. IEEE, 2018. 7

[55] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 3