

Invertible Neural Warp for NeRF

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

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

Abstract. This paper tackles the simultaneous optimization of pose and Neural Radiance Fields (NeRF). Departing from the conventional practice of using explicit global representations for camera pose, we propose a novel overparameterized representation that models camera poses as learnable rigid warp functions. We establish that modeling the rigid warps must be tightly coupled with constraints and regularization imposed. Specifically, we highlight the critical importance of enforcing invertibility when learning rigid warp functions via neural network and propose the use of an Invertible Neural Network (INN) coupled with a geometry-informed constraint for this purpose. We present results on synthetic and real-world datasets, and demonstrate that our approach outperforms existing baselines in terms of pose estimation and high-fidelity reconstruction due to enhanced optimization convergence.

Keywords: Neural Radiance Fields · Joint scene reconstruction and pose estimation · Implicit Neural Representations

1 Introduction

NeRF [25] has recently emerged as a compelling approach for synthesizing photo-realistic images from novel views. NeRF employs a multi-layer perceptron (MLP) to model a volumetric representation of a 3D scene. It operates by minimizing the photometric loss, which is the discrepancy between rendered images and actual images. NeRF’s ability to reconstruct high-fidelity signals, coupled with its memory efficiency, has propelled its adoption across a wide array of applications [13, 18, 31, 43, 44, 46–48], demonstrating its significant impact and versatility.

One of the primary challenges with NeRF is the requirement for precisely known camera poses for each captured image. To address this challenge, several approaches such as BARF [23], NeRFmm [40], and GARF [10] have been developed. These methods facilitate the simultaneous optimization of the NeRF and the camera poses, using a compact, six-dimensional vector to represent the camera poses efficiently. However, this compact parameterization, while prevalent in contemporary structure from motion (SfM) literature [32, 34], has been shown to struggle with poor basin of convergence when solved simultaneously with a NeRF [9]. Drawing wisdom from machine learning, where overparameterization has been recognized as a catalyst for enhanced optimization convergence

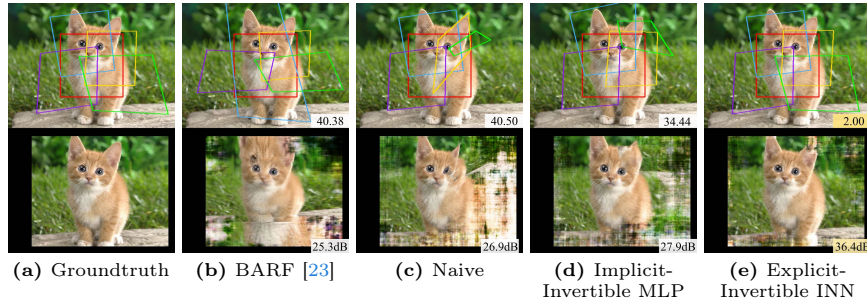


Fig. 1: We investigate how overparameterizing rigid warps of rays with an MLP benefits the joint optimization task of camera pose and NeRF. This example estimates the warps that align the color-coded patches in Fig. 1a while solving for the neural field. Unlike “BARF” and “naive” MLP pose overparameterization methods fail catastrophically, enforcing invertibility, either *implicitly* (Fig. 1d) or *explicitly* (Fig. 1e) significantly improves warp estimation, see Sec. 4.1 for details of each method. We establish that *invertibility* is a crucial for MLP-based rigid warp representation.

in modern deep neural networks [22, 26, 35], this paper explores the potential of pose overparameterization for simultaneous pose and neural field estimation.

Our approach: In traditional NeRF setups, accurately known extrinsic camera pose, comprising of a global rotation and translation for each image, are used to explicitly map pixel coordinates and the camera center to determine the viewing rays in a global world coordinate system [25]. Following the warping operation, the colors and volume densities along each ray in the world coordinate space is manipulated individually through a photometric loss function. In this paper, we explore scenarios where camera poses are not known. Specifically, we propose using a neural network to model the *rigid warp function* of ray. While it may seem counterintuitive to replace a succinct pose function with a more complex MLP, we argue that the enhanced convergence properties of such overparameterization [1, 26, 27] – in conjunction with the right constraint and prior – outweigh the increased functional flexibility.

Additionally, we highlight the critical role of enforcing *invertibility* when learning rigid warps using an MLP.¹ To achieve an approximate bijective solution, one remedy is to use an auxillary network to represent the backward warp; however this will introduce computational overhead. To this end, we propose explicitly modeling inversions in the neural network architectures, formally learning an Invertible Neural Network (INN). Our results demonstrate that opting for an architecture that is explicitly invertible is more effective for jointly optimizing both the pose and radiance field, outperforming existing strong baselines [9, 23]. Notably, our INN-based approach achieves an improvement of over 50% in pose accuracy when compared to the standard SE3 parameterization [23].

¹ Our use of invertibility strictly adheres to the well-established mathematical definition. Let f be a function whose domain is \mathcal{X} and codomain is \mathcal{Y} . f is invertible iff there exists a function g from \mathcal{Y} to \mathcal{X} such that $g(f(x)) = x \ \forall x \in \mathcal{X}$ and $f(g(y)) = y \ \forall y \in \mathcal{Y}$ [14]. We use bijective and invertible interchangeably throughout our paper.

2 Related Works

2.1 Joint NeRF and pose estimation

Despite NeRF demonstrating compelling results in novel view synthesis, NeRF requires accurate camera poses. The differentiable nature of volume rendering used in NeRF facilitates the backpropagation through the scene representation to update the camera poses. NeRFmm [40] demonstrates the possibility of optimizing camera poses within the NeRF framework. BARF [23] introduces a coarse-to-fine positional encoding scheduling to improve the joint optimization of NeRF and camera poses, and remains a widely adopted method. GARF [10] and SiNeRF [42] advocate for leveraging the smoothness inherent in non-traditional activations to mitigate the noisy gradients due to high frequencies in positional embeddings. In contrast, NoPe-NeRF [5] uses monocular depth prior as a geometry prior to constrain the relative poses. SPARF [37] and SCNeRF [16] demonstrate that using keypoint matches or dense correspondence can constrain the relative pose estimates with ray-to-ray correspondence losses. Park *et al.* [30] proposes a preconditioning strategy to enhance camera pose optimization. DBARF [8] proposes using low-frequency feature maps to address the joint optimization problem for generalizable NeRF. Bian *et al.* [4] proposes a pose residual field which learn the pose corrections to refine the initial camera pose for neural surface reconstruction. Closest to our approach is a very recent work L2G by Chen *et al.* [9], which tackles the camera pose representation using an overparameterization strategy. However, we achieve overparameterization in different manner. Unlike L2G which learn an MLP to predict rigid $SE(3)$ transformations, we propose using an MLP to model the rigid warp function between the pixel and the ray space. Our work argues that while overparameterization can be achieved in different manner, it is tightly coupled with the regularization and constraints imposed. In our case, invertibility of warps becomes an essential constraint.

2.2 Overparameterization

Overparameterization in deep learning involves employing models with a substantially greater number of parameters than the quantity of training data. Recent research [22, 26, 35] demonstrates that neural networks, when overparameterized, can effectively generalize to new, unseen data, noting that an increase in parameters often correlates with a decrease in test error. Moreover, [26] revealed that this ability to generalize does not necessarily require explicit regularization, suggesting that the optimization process of overparameterized neural networks inherently prefers solutions that are more likely to generalize well. Additionally, [1, 2, 27, 29] illustrates that overparameterized networks are capable of consistently finding a global minimum through gradient-based optimization methods. These insights highlight the remarkable ability of overparameterized neural networks to deliver accurate predictions on unseen data, emphasizing their robustness and efficacy in generalizing beyond the training dataset.

2.3 Invertible Neural Networks for Deformation Fields

Recent advancements in Invertible Neural Networks (INNs) [3, 7, 12] have broadened their use in the field of 3D deformation. These networks are particularly useful in modeling homomorphisms, where the mapping between any frames are bijective and continuous. Because of this capability, they are now being used in various areas, particularly modeling deformation in spatial [17, 33, 45] and temporal [6, 20, 28, 39].

3 Methodology

We define the mathematical notations for the camera operations and the joint camera pose estimation in Sec. 3.1. Further, we outline our approach in Sec. 3.2.

3.1 Bundle-Adjust NeRF Preliminaries

Camera pose We consider a set of T input images as $\{\mathcal{I}_t\}_{t=1}^T$ taken by camera with intrinsic matrix $K \in \mathbb{R}^{3 \times 3}$. For each camera t corresponding to these images, BARF-style approaches [23] define its *camera-to-world* (C2W) as $P = (\mathbf{R}_t, \mathbf{t}_t) \in SE(3)$, where $\mathbf{R}_t \in SO(3)$ and $\mathbf{t}_t \in \mathbb{R}^3$, respectively.

Camera projection For any vector $\mathbf{x} \in \mathbb{R}^l$ of dimension l , we define its homogeneous representation $\bar{\mathbf{x}} \in \mathbb{R}^{l+1}$ as $\bar{\mathbf{x}} = [\mathbf{x}^T, 1]$. We define π as the camera projection operator, which maps a 3D point in the *camera coordinate frame* denoted as $\mathbf{x}^{(C)} \in \mathbb{R}^3$ to a corresponding 2D pixel coordinate $\mathbf{u} \in \mathbb{R}^2$. π^{-1} denotes the camera backprojection that maps a pixel \mathbf{u} coordinate and depth z to a 3D point in the camera coordinate as $\mathbf{x}^{(C)}$, for e.g., $\pi(\mathbf{x}^{(C)}) \cong K\mathbf{x}^{(C)}$ and $\pi^{-1}(\mathbf{u}, z) = zK^{-1}\bar{\mathbf{u}}$. We use $^{(C)}$ and $^{(W)}$ to denote that it is defined within the camera and world coordinate system respectively.

NeRF NeRF represents the volumetric field of a 3D scene as $f(\gamma(\mathbf{x}), \gamma(\mathbf{d})) \rightarrow (\mathbf{c}, \sigma)$, which maps a 3D location $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction \mathbf{d} to a RGB color $\mathbf{c} \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}$. $\gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^{3+6L}$ is the positional embedding function with L frequency bases [25]. This function is parameterized using an MLP as $f_{\Theta_{rgb}}$. Given T input images $\{\mathcal{I}_t\}_{t=1}^T$ with corresponding camera poses $\{P_t\}_{t=1}^T$, NeRF is optimized by minimizing photometric loss \mathcal{L}_{rgb} between synthesized images $\hat{\mathcal{I}}$ and original image \mathcal{I} as

$$\min_{\Theta_{rgb}} \sum_{t=1}^T \sum_{\mathbf{u} \in \mathbb{R}^2} \|\hat{\mathcal{I}}(\mathbf{u}, P_t; \Theta_{rgb}) - \mathcal{I}_t(\mathbf{u})\|_2^2, \quad (1)$$

Volume rendering For simplicity, let's start by assuming the rendering operation of NeRF operates in the *camera coordinate system*. We will generalize this later. Each pixel coordinate determines a viewing direction \mathbf{d} in the camera coordinate system, whose origin is the camera center of projection $\mathbf{o}^{(C)}$. We can define a 3D point along the camera ray associated with \mathbf{u} sampled at depth z_i

as $\mathbf{r}^{(C)}(z) = \mathbf{o}^{(C)} + z_{i,u}K^{-1}\bar{\mathbf{u}}$.² To render the colour of $\hat{\mathcal{I}}_{i,u}$ at pixel coordinate \mathbf{u} , we sample M discrete depth values along the ray between a near bound z_n and a far bound z_f . For each sampled value, we query the NeRF $f_{\Theta_{rgb}}$ to obtain the corresponding radiance fields. The output from NeRF is then aggregated to render the RGB colour as

$$\hat{\mathcal{I}}(\mathbf{u}) = \int_{z_n}^{z_f} T(\mathbf{u}, z) \sigma(\mathbf{r}(z)) \mathbf{c}(\mathbf{r}(z)) \delta z, \quad (2)$$

where $T(\mathbf{u}, z) = \exp(-\int_{z_n}^z \sigma(\mathbf{r}(z')) \delta z')$ denotes the accumulated transmittance value along the ray. We refer readers to [21, 25] for more details of volume rendering operation. In practice, Eq. (2) is approximated using M points sampled along the ray at depth $\{z_i\}_{i=1}^M$, which produces radiance field outputs as $\{\mathbf{y}_i\}_{i=1}^M$. Denoting the ray compositing function in Eq. (2) as $g(\cdot) \in \mathbb{R}^{4M} \rightarrow \mathbb{R}^3$, we can rewrite $\hat{\mathcal{I}}(\mathbf{u}) = g(\{\mathbf{y}_i\}_{i=1}^M)$. Finally, given a camera pose P , we can then transform the ray $\mathbf{r}^{(C)}$ to the *world coordinate* to obtain $\mathbf{r}^{(W)}$ through a 3D rigid transformation \mathcal{T} . The rendered image is then obtained as

$$\hat{\mathcal{I}}(\mathbf{u}, \mathbf{p}) = g\left(\left\{f(\mathcal{T}(\mathbf{r}^{(C)}(z), P); \Theta_{rgb})\right\}_{i=1}^M\right). \quad (3)$$

Joint optimization of pose and NeRF Prior works [10, 23, 40, 42] demonstrate that it is feasible to optimize both camera pose and NeRF by minimizing \mathcal{L}_{rgb} in Eq. (1). This is achieved by considering P as optimizable parameters, see Sec. 3.1 for its parameterization. Consequently, ray \mathbf{r} is now dependent on the camera pose. Mathematically, this joint optimization can be rewritten as

$$\min_{P, \Theta_{rgb}} \sum_{t=1}^T \sum_{\mathbf{u} \in \mathbb{R}^2} \underbrace{\|\hat{\mathcal{I}}(\mathcal{T}(\mathbf{r}^{(C)}(z, \bar{\mathbf{u}}); P); \Theta_{rgb}) - \mathcal{I}_i(\mathbf{u})\|_2^2}_{G(\cdot)}. \quad (4)$$

3.2 Invertible Neural Warp for Ray Transform

We propose to overparameterize P using an Invertible Neural Network (INN). There exist two options for parameterization: (i) use a separate INN for each camera P_t ; (ii) use a single INN that is shared across all frames, coupled with a learnable code that is unique to frame t . Drawing inspiration from the dynamic NeRF method used for representing deformation fields [6, 31, 38, 39] and also considering parameter efficiency, we have chosen to pursue the latter strategy for our proposed pose overparameterization – a single, globally shared neural $\Theta_{\mathcal{W}}$ network across all frames, coupled with an optimizable per-frame latent code $\Phi_t \in \mathbb{R}^D$, see supp. (Sec. C) for the comparison of using multiple-INNs versus single-INN. Consequently, we can rewrite $G(\cdot)$ in Eq. (6) as $h(\mathbf{r}^{(C)}; \Theta_{\mathcal{W}}, \Phi_t)$, where $h(\cdot) : \mathbb{R}^{3+D} \rightarrow \mathbb{R}^3$. Fig. 2 presents our approach. In our approach, we

² This can be succinctly written as $\mathbf{r}^{(C)}(z) = z_{i,u}\mathbf{d}$ as $\mathbf{o}^{(C)}$ is $[0, 0, 0]^T$ in camera coordinate space.

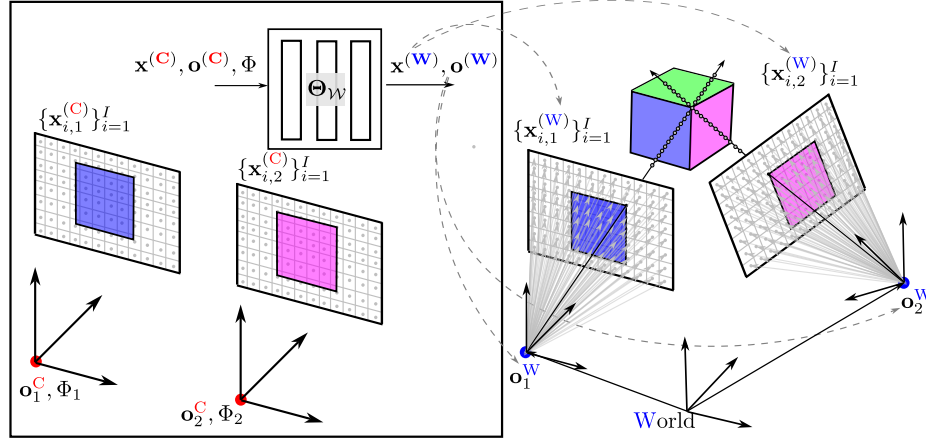


Fig. 2: An overview of our INN-based approach, illustrated using two views \mathcal{I}_1 and \mathcal{I}_2 . INN which is denoted as h_{Θ_W} takes the pixel locations in the camera coordinate system $\mathbf{x}_{i,t}^{(C)}$, along with the frame-dependent latent code Φ_t , and output its corresponding location in the world coordinate system as $\mathbf{x}_{i,t}^{(W)}$, see Sec. 3.2 for full details.

model each pixel in the camera coordinate system $\mathbf{x}_{i,t}^{(C)}$ as an individual ray. Our proposed INN is designed for transforming these rays from camera coordinate to the world coordinate. Specifically, our proposed INN takes in the pixel coordinates $\mathbf{x}_{i,t}^{(C)}$ and camera center $\mathbf{o}_t^{(C)}$, both defined in the camera coordinates t , coupled with the frame-dependent latent code and outputs their corresponding equivalent $\mathbf{x}_{i,t}^{(W)}$ and camera center $\mathbf{o}_t^{(W)}$ in the world coordinates.

Rigidity prior In our formulation, each pixel is represented as an individual ray within the camera coordinate system. This inherently relaxes the rigidity constraint. As a result, the output from the INN does not necessarily conform to a global rigid motion. Given known camera-world correspondences $(\mathbf{x}_{i,t}^{(C)}, \mathbf{x}_{i,t}^{(W)})$, we can solve a closed-form rigid registration problem to determine a global pose, which can be integrated into our optimization problem as a rigidity prior \mathcal{L}_{rigid}

$$\min_{T^*} \sum_{i=1}^L \|\mathbf{x}_{i,t}^{(C)} - T^* \circ \mathbf{x}_{i,t}^{(W)}\|_2^2. \quad (5)$$

Final optimization problem We solve our final optimization problem as

$$\min_{\Phi_t, \Theta_W, \Theta_{rgb}} \sum_{t=1}^T \sum_{\mathbf{u} \in \mathbb{R}^2} \|\hat{\mathcal{I}}(h(\mathbf{r}^{(c)}; \Theta_W, \Phi_t); \Theta_{rgb}) - \mathcal{I}_i(\mathbf{u})\|_2^2 + \lambda \mathcal{L}_{rigid}. \quad (6)$$

3.3 Advantages of INN for Overparameterizing Rigid Ray Warps

BARF-based approach parameterizes camera pose P of each frame using a $SE(3)$ (see Sec. 3.1), which guarantees that \mathcal{T} is a bijective mapping. Therefore, when

overparameterizing camera poses, it is crucial that the neural network adheres to the bijection property because this one-to-one correspondence ensures that there is a unique output in the *world* space for every point in the *camera* space. As we will demonstrate in Sec. 4.2, simply applying a rudimentary strategy (denoted as Naive) when overparameterizing the rigid warps of ray (camera-world) with an MLP often does not achieve convergence, see Fig. 1c and Tab. 1. Consequently, to attain this bijective property using an MLP, it is necessary to introduce an auxiliary network to model the backward warps (Implicit-Invertible MLP). While effective, it presents a significant drawback: it results in a twofold rise in the computational complexity due to the existence of the backward network to enforce the self-consistency. To mitigate this substantial increase in computational demands inherent in the modified MLP approach, we propose the use of INNs for parameterize these bijections. INN implements the bijective mappings by composing affine transformations into several blocks. Within each block, the input coordinates are divided into two segments; the first part remains constant and is used to parameterize the transformation that is applied to the second part [6]. Besides their inherent invertibility, INN also offers the advantage of homeomorphic property, which potentially facilitate a more flexible optimization trajectory that is less susceptible to a suboptimal minimum trajectory, see Sec. 4.5

4 Experiments

4.1 Baselines

We compare our approach with two representative methods in pose-NeRF joint optimization: the standard global *SE3*-approach BARF [23], and the overparameterized representation L2G [9]. For all experiments, we use the original implementations including their default settings for coarse-to-fine scheduling, architecture and hyperparameters, see supp. (Sec. A) for more details. Additionally, in our 2D planar experiments, we include a comparison with another two variants called the Naive (MLP) and Implicit-Invertible MLP to execute ray transform. This comparison is specifically designed to highlight the significance of invertibility when employing MLPs for executing ray transformation.

Local-to-global (L2G) [9] L2G uses an MLP to predict rigid *SE3* transformation for each ray. These predicted transformation parameters are then used to analytically estimate the transformed coordinates in the world space.

Naive This is the simplest version of our baseline that uses one primary network, denoted as $h_{fwd} : (\mathbf{x}^{(C)}, \Phi_t) \rightarrow \mathbf{x}^{(W)}$, to learn the forward mapping which takes in the coordinates from the camera space (coupled with a per-frame latent code), and outputs the corresponding coordinates in the world space.

Implicit-Invertible MLP We use two networks h_{fwd} and h_{bwd} to enforce approximate invertibility. Alongside primary network h_{fwd} , we use a secondary network $h_{bwd} : (\mathbf{x}^{(W)}, \Phi_t) \rightarrow \hat{\mathbf{x}}^{(C)}$ to invert the outputs from the primary network h_{fwd} . To minimize deviations from bijections, we introduce a regularization term $\mathcal{L}_{implicit}$ as $\|\mathbf{x}^{(C)} - \hat{\mathbf{x}}^{(C)}\|_2^2$ into the optimization problem Eq. (6).

Explicit-Invertible INN (Ours) We explicitly model inversions in the neural network architecture by formally learning an INN. We have chosen to utilize architecture proposed by NDR-INN [6], see supp. (Sec. A.1) for the architecture details that we use for all our experiments.

4.2 2D Planar Neural Image Alignment

Following BARF [9, 10, 23], we learn a 2D neural image field, for creating a homography-based panoramic image from N patches cropped from the original image, each generated with random homography perturbations. Specifically, we learn a 2D coordinate network $f(\Theta_{rgb})$ to render the stitched image. Each pixel in the N training patch is warped using the estimated homography H to create the rendered image. We choose the “cat” image from ImageNet [11]. We initialize patch warps as identity and fix the gauge freedom by anchoring the first warp to align the neural image to the original image [9, 10, 23]. We randomly generate 20 different homography instances, with scale-noise parameter 0.1 and 0.2 for homography and translation, respectively. We solve Eq. (5) for the homography using a Direct Linear Transform (DLT) solver³.

Experiment settings. We evaluate our proposed method against BARF [23], and our three overparameterized network variants: Naive, Implicit- and Explicit-Invertible MLPs, as detailed in Sec. 4.1. For both naive and Implicit-Invertible MLPs, we utilized a Leaky-ReLU MLP with five 256-dimensional hidden units, and a 16-dimensional latent code to represent the frame-dependent embeddings ϕ_t . We also follow the default coarse-to-fine scheduling established by BARF. In the case of Implicit-Invertible MLP, we use the same architecture both h_{fwd} and h_{bwd} . We use the Adam optimizer [19] to optimize for both the network weights Θ_{rgb} and $\Theta_{\mathcal{W}}$. We set the learning rate for both Θ_{rgb} and $\Theta_{\mathcal{W}}$ at 1×10^{-3} , with both decaying exponentially to 1×10^{-4} and 1×10^{-5} , respectively. We set the weighting term for \mathcal{L}_{rigid} for both overparameterized MLPs to 1×10^2 and the consistency term $\mathcal{L}_{implicit}$ for Implicit-Invertible MLP to 1×10^1 .

Robustness to noise perturbations Fig. 3 analyzes the robustness of BARF and our approach under different noise perturbations across 20 different homography instances. Each run began with the initialization of the homography using the “groundtruth”, and noise perturbations were gradually introduced, ranging from 0 to 0.3 to the translation component of the homography. Fig. 3 indicates that our representation exhibits a higher tolerance to noise compared to BARF.

Results. Tab. 1 summarizes the statistical results from 20 runs, where we report the warp error and patch reconstruction error. We quantify the warp error in terms of corner error, defined as the L2 distance between the groundtruth corner position and estimated corner position, and PSNR as the metric to assess the reconstruction quality. We used 5-pixel threshold to define the success

³ <https://github.com/kornia/kornia>

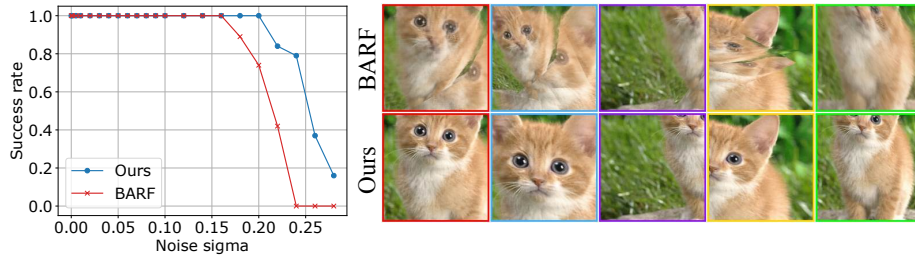


Fig. 3: Basin convergence analysis of our approach versus BARF in a 2D planar experiment. On the left, we show the success rate of 20 runs, where we initialize the homography using *groundtruth*, and gradually introduced noise perturbations to the translation component. The noise scale is varied from 0 to 0.30. We used 5-pixel threshold to determine success convergence. Notably, our approach (blue) demonstrates higher noise tolerance compared to BARF (red). On the right, we show a qualitative comparison when both methods are perturbed with the highest magnitude of noise.

Table 1: Statistical result for 20 homography runs, with scale noise of 0.1 for homography and 0.2 for translation. The warp error is quantified in terms of corner error, and the patch reconstruction error is measured in PSNR. We provide mean and std dev for the evaluation. We used 5-pixel threshold to determine success convergence.

	Corner error (px) ↓		Patch PSNR ↑		Success rate ↑ (Upper bound:1.00)
	Mean	Std. dev.	Mean	Std. dev.	
BARF [23]	29.63	28.18	28.94	4.38	0.30
Naive	85.59	30.31	25.86	2.07	0.00
Implicit-Invertible	13.92	22.93	33.70	3.93	0.65
Explicit-Invertible (INN)	4.70	6.47	34.71	2.37	0.75

convergence. Fig. 1 presents a qualitative result for a homography instance. By enforcing approximate invertibility, the Implicit-Invertible MLP demonstrates a significantly higher rate of successful convergence compared to the naive version of MLP, with success rates improved by 65%. We further show that by *explicitly* injecting invertibility into the architecture (Ours), the success rate is increased to 75%. This result reinforces that *invertibility* is crucial when learning rigid warp functions via overparameterization, and using an architecture that guarantees bijective property is effective in ensuring the pose converge to an optimal solution during joint optimization in practice. Henceforth, we will focus exclusively on using our INN-based approach for subsequent results. For completeness, we also compare with another setup, where each frame is parameterized with a neural network, see supp. (Sec. B.1). Interestingly, apart from the parameter efficiency, we find that using a single global neural network for all frames is sufficient to converge to a good pose solutions. We hypothesized that the difference may be attributed to the benefits of gradient sharing in the shared neural network setup. This result is aligned with the findings by Bian *et al.* [4]. Consequently, we adhere to the design where we use one single INN shared across all the frames, coupled with a frame-specific latent code for the rest of our experiments.

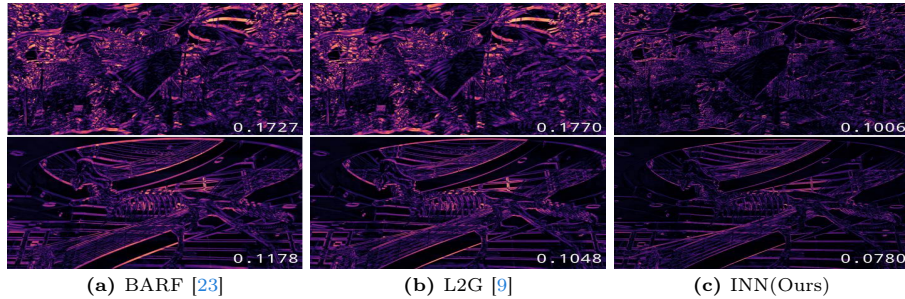


Fig. 4: Qualitative analysis of reconstruction error on leaves (**top**) and trex (**bottom**). We present the average image reconstruction error through insets. Our approach presents the lowest misalignment error, as indicated darker areas in the error map.

4.3 Neural Radiance Fields (NeRF)

In this section, we compare our proposed representation with BARF [23] and L2G [9]. We assume known intrinsics for all methods. We perform our experiments on both the LLFF [24] (Sec. 4.4), DTU [15] (Sec. 4.6) as well as Blender datasets in supp. (Sec. B2). We solve Eq. (5) for the global rigid $SE3$ transformation using Umeyama algorithm⁴.

Evaluation metrics. For pose estimation, we report the accuracy of the poses after globally aligning the optimized poses to the groundtruth [9, 10, 23, 37].⁵ We assess view synthesis using PSNR, SSIM and LPIPS. A standard procedure in view synthesis evaluation involves performing test-time photometric optimization on the trained models. This additional step is intended to *factor out the pose errors*, which may otherwise compromise the quality of the synthesized views [9, 10, 23, 36, 37]. This process is akin to a *pose refinement* method which minimises the photometric error on the synthesized image while keeping the trained NeRF model fixed. However, it is important to recognize that this pose correction may not accurately represent the initial accuracy of the methods in terms of pose estimation for view synthesis. Therefore, we opt to report the view synthesis quality both *before* and *after* the pose refinement step. On DTU, we extend our evaluation to include comparisons of rendered depth with ground-truth depth using mean depth absolute error, as well as reconstruction accuracy using Chamfer distance. For the Chamfer evaluation, we utilize the optimized poses estimated from all methods and employ a neural surface reconstruction algorithm called Voxurf [41] for geometry reconstruction. Further details on pose alignment and metrics computations in the supp. (Sec. A).

4.4 Forward Facing Scenes: LLFF

Experiment settings. The standard LLFF benchmark dataset [24] consists of eight real-world, forward-facing scenes captured using hand-held cameras.

⁴ <https://github.com/naver/roma>

⁵ For our proposed method, we evaluate the estimated global poses Eq. (5).

Table 2: Evaluation of the LLFF dataset [24] using initial *identity* poses. We compute results for BARF [23] and L2G [9] using their codebases. The results are **averaged** over all **eight** scenes, see supp. (Sec. B) for a breakdown of results per scene.

	Pose accuracy		Novel view synthesis					
	Rotation	Translation	<i>Before</i> test-time			<i>After</i> test-time		
	($^{\circ}$)	($\times 100$)	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
BARF [23]	0.90	0.40	17.00	0.41	0.32	23.82	0.72	0.24
L2G [9]	0.48	0.30	17.99	0.46	0.26	24.35	0.75	0.21
INN(Ours)	0.31	0.24	19.31	0.52	0.25	24.28	0.74	0.22

Following [10, 23, 37], we initialize all camera poses with the *identity* transformations for all the methods. We employ the same training and testing split as in BARF [23]. We use the evaluation metrics described in Sec. 4.3.

Implementation details. We train all methods for 200k iterations and randomly sample 2048 pixel rays at each optimization step [9, 10, 23]. We train without hierarchical sampling. We set the learning for Θ_{rgb} to be 1×10^{-3} decaying to 3×10^{-4} , and 5×10^{-4} for Θ_W decaying to 1×10^{-6} . We also follow the default coarse-to-fine scheduling by BARF [23], see supp. for full details.

Results. Our approach achieves a substantial reduction of rotation errors (70% vs. BARF and 35% vs. L2G) and translation errors (50% vs. BARF and 20% vs. L2G). Additionally, the superior performance of both our method and L2G over BARF further highlight the merits of overparameterization for simultaneous pose and neural field estimation. This significant improvement in camera pose accuracy directly enhances the performance of view synthesis, as evident in Tab. 2. In particular, instances such as *trex* and *leaves* observe a significant improvement. Fig. 4 illustrates the absolute error between the original image and the rendered image on these two instances. Our approach presents the lowest misalignment error, as indicated darker areas in the error map. It is important to note, however, while L2G and our approach appear comparable after test-time optimization, this refinement procedure has mitigated the pose estimation noise to improve PSNR, as discussed in Sec. 4.3. For more qualitative results and ablation studies, please refer to supp. (Sec. B).

4.5 Homeomorphism perspective: A qualitative analysis

We present a qualitative analysis on a single-view pose estimation that sheds light on the empirical effectiveness of our approach compared to the L2G method [9]. Our method leverages the concept of an INN, which predict homeomorphisms—continuous, invertible transformations not limited to the rigid motions of the $SE(3)$ group. Unlike the L2G method, which constrains pose estimation within the rigid bounds of $SE(3)$, our INN-based approach embraces a broader spectrum of transformations. This grants the optimization process a higher degree of flexibility, allowing for a diverse range of optimization paths and facilitating a smoother trajectory towards the solution. To empirically validate our

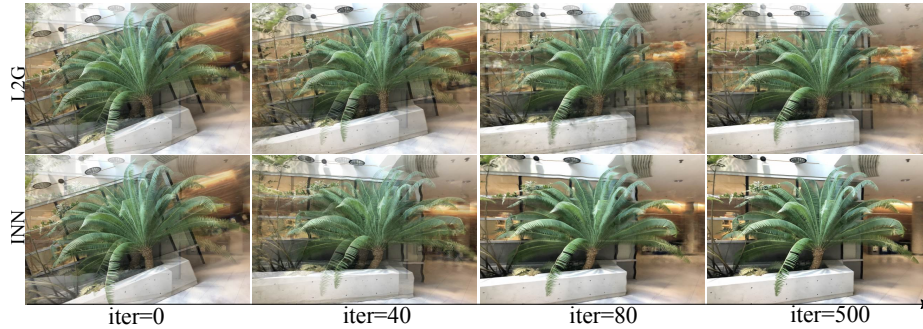


Fig. 5: Qualitative analysis of intermediate rendered image compared to superimposed Groundtruth (lighter visualization) when using L2G [9] and our approach on single-view pose estimation. Notably, when using our approach for pose estimation, we observe noticeable *deformation* in the rendered scene depicted in the bottom row (zoom in for better view). These deformations indicate that at each iteration, the INN predicts general homeomorphisms that are not rigid transformation. Thus yielding a flexible optimization trajectory that does not land in a suboptimal minimum trajectory.

Table 3: Absolute pose accuracy evaluation of the DTU dataset [15] using initial *noisy* poses that corresponds to an average rotation and translation error of 15° and 70 respectively. The *upper* section shows rotation errors in degrees, while the *lower* section displays translation errors ($\times 100$). **red box** denotes the **best** result.

Methods	Metric	Scan IDs Initial rotation err: 15° , translation err ($\times 100$) : 70														mean
		24	37	40	55	63	65	69	83	97	105	106	110	114	118	
BARF	rotation	1.04	1.88	0.55	2.85	0.43	0.78	3.88	0.54	2.20	3.56	11.81	2.01	1.21	2.60	2.52
L2G		0.90	3.07	0.57	10.81	0.74	0.94	1.55	0.59	0.64	2.85	13.15	5.92	2.90	12.47	4.08
INN(Ours)		0.21	0.48	0.73	1.64	0.40	0.62	2.75	0.55	0.54	0.30	3.89	2.91	1.06	0.25	1.17
BARF	translation	3.00	4.08	2.09	11.63	1.43	2.78	12.80	2.13	6.12	12.40	25.88	5.52	4.08	5.04	7.07
L2G		2.40	8.80	2.13	32.74	2.78	3.48	5.08	2.31	2.37	5.34	36.20	14.01	8.64	37.16	11.67
INN(Ours)		0.50	1.01	0.97	5.24	0.96	0.78	5.30	0.67	1.19	0.43	9.96	11.52	3.78	0.64	3.07

hypothesis, we conducted experiments using a trained NeRF model to estimate the camera pose relative to a 3D scene, aiming to minimize the photometric error between NeRF-rendered and actual observed images. Despite starting from the same initial pose (off by 20°), and employing a random sampling of 2048 rays per iteration for all methods, our INN approach outperformed the L2G method. The L2G method often converged to suboptimal poses, whereas the INN method achieved accurate pose estimation, as evidenced in Fig. 5 where the NeRF-rendered images is well-aligned with the groundtruth. This success can be attributed to the INN’s ability to predict a general homeomorphism, a transformation more general than a rigid transformation, which offers a significant advantage in navigating the optimization landscape more effectively and avoiding suboptimal local minima.

4.6 360° Scenes: DTU

Experimental settings. We evaluated on 14 test scenes from DTU [4]. Following [9, 23, 37], we synthetically perturb the ground-truth camera poses with

Table 4: Geometry evaluation of the DTU dataset [15], using initial *noisy* poses that corresponds to an average rotation and translation error of 15° and 70 respectively. The *upper* section presents mean absolute depth, while the *lower* section displays reconstruction error measured by Chamfer distance. **red box** denotes the **best** result.

Pose	Metric	Scan IDs Initial rotation err: 15° , translation err ($\times 100$) : 70														mean
		24	37	40	55	63	65	69	83	97	105	106	110	114	118	
BARF	depth \downarrow (abs)	0.11	0.15	0.11	0.12	0.14	0.10	0.26	0.17	0.17	0.46	0.53	0.17	0.10	0.17	0.20
L2G		0.08	0.24	0.12	0.38	0.17	0.12	0.11	0.18	0.10	0.18	0.24	0.17	0.12	0.84	0.22
INN(Ours)		0.06	0.14	0.13	0.12	0.15	0.08	0.18	0.17	0.11	0.11	0.13	0.30	0.08	0.06	0.13
BARF	Chamfer \downarrow	5.47	4.07	4.39	7.64	4.83	5.28	6.43	5.74	8.06	7.21	7.75	8.29	8.24	5.47	6.35
L2G		4.37	9.08	4.41	6.67	8.03	5.92	3.80	6.00	6.30	5.59	7.43	7.35	8.29	8.16	6.53
INN(Ours)		1.56	2.25	5.55	5.68	4.58	4.27	5.51	5.69	5.09	3.73	8.15	7.85	7.35	1.14	4.89

15% of additive Gaussian noise, which corresponds to an average rotation and translation error of 15° and 70, respectively. For a fair comparison, we used the same initialization for all methods. We refer the readers to supp. (Sec B.5) for the results with Colmap initialization.

Implementation details. As BARF [23] and L2G [9] have not tested on DTU datasets, given that the DTU dataset encompass 360° scenes which is similar to the original Blender dataset, we adopted the original hyperparameters used by the author for training BARF and L2G on the blender dataset. Following Bian *et al.* [4], we multiply the output of their local warp network by a small factor which is $\alpha = 0.01$ for L2G [9]. For our approach, we set the learning rate for Θ_{rgb} to be 1×10^{-3} decaying to 1×10^{-4} , and Θ_W to start from 5×10^{-4} and decaying to 1×10^{-8} . We use the default coarse-to-fine scheduling by BARF [23].

Results. As demonstrated in Tab. 3, we outperform all baselines by a considerable margin across majority of the sequences in pose accuracy. Overall, our approach achieves approximately a 50% improvement in rotation and a 60% improvement in translation over BARF. When compared to L2G, our method shows a 70% increase in accuracy for both rotation and translation. Additionally, our approach also consistently surpasses both baselines in geometry evaluation, as evidenced in the depth and reconstruction error in Tab. 4 and qualitative results in Fig. 6. For the quantitative results for novel view synthesis and additional qualitative results, we refer the readers to see supp. (Sec B).

5 Conclusion

In this paper, we examine the benefits of overparameterizing poses via a MLP in the joint optimization task of camera pose and NeRF. We establish that invertibility is a crucial property. We further show that using an Invertible Neural Network, inherently equipped with a guaranteed bijection property, significantly improves the convergence in pose optimization compared to existing representative methods.

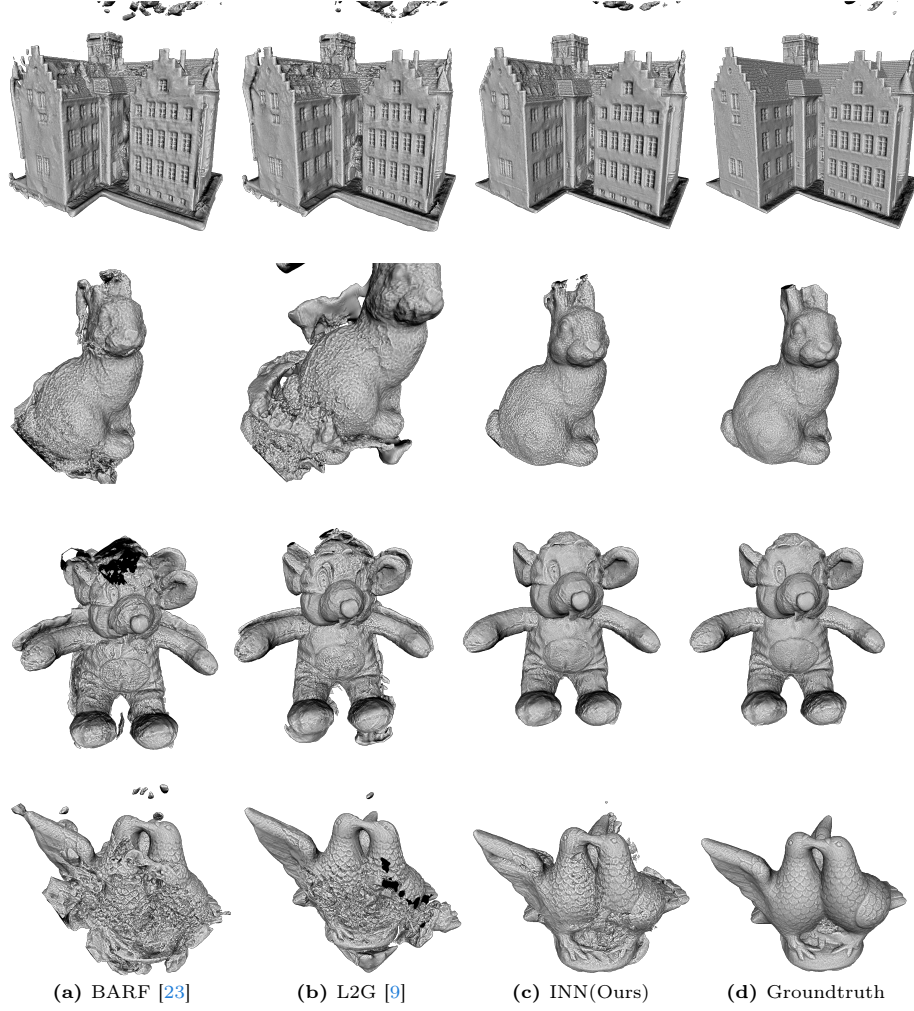


Fig. 6: We showcase the reconstruction results for *scan24*, *scan55*, *scan105* and *scan106*. All meshes are generated using a neural surface reconstruction algorithm called Voxurf [41] using the poses estimated by different approaches.

Acknowledgement We thank Chee-Kheng (CK) Chng for insightful discussions and technical feedback.

References

1. Allen-Zhu, Z., Li, Y., Liang, Y.: Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems* **32** (2019) 2, 3
2. Arora, S., Du, S., Hu, W., Li, Z., Wang, R.: Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In: *International Conference on Machine Learning*. pp. 322–332. PMLR (2019) 3
3. Behrmann, J., Grathwohl, W., Chen, R.T., Duvenaud, D., Jacobsen, J.H.: Invertible residual networks. In: *International conference on machine learning*. pp. 573–582. PMLR (2019) 4
4. Bian, J.W., Bian, W., Prisacariu, V.A., Torr, P.: Porf: Pose residual field for accurate neural surface reconstruction. *arXiv preprint arXiv:2310.07449* (2023) 3, 9, 12, 13
5. Bian, W., Wang, Z., Li, K., Bian, J.W., Prisacariu, V.A.: Nope-nerf: Optimising neural radiance field with no pose prior. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4160–4169 (2023) 3
6. Cai, H., Feng, W., Feng, X., Wang, Y., Zhang, J.: Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *Advances in Neural Information Processing Systems* **35**, 967–981 (2022) 4, 5, 7, 8
7. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. *Advances in neural information processing systems* **31** (2018) 4
8. Chen, Y., Lee, G.H.: Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 24–34 (2023) 3
9. Chen, Y., Chen, X., Wang, X., Zhang, Q., Guo, Y., Shan, Y., Wang, F.: Local-to-global registration for bundle-adjusting neural radiance fields. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8264–8273 (2023) 1, 2, 3, 7, 8, 10, 11, 12, 13, 14
10. Chng, S.F., Ramasinghe, S., Sherrah, J., Lucey, S.: Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In: *European Conference on Computer Vision*. pp. 264–280. Springer (2022) 1, 3, 5, 8, 10, 11
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. pp. 248–255. Ieee (2009) 8
12. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. *arXiv preprint arXiv:1605.08803* (2016) 4
13. Guo, M., Fathi, A., Wu, J., Funkhouser, T.: Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503* (2020) 1
14. Jeffreys, H., Jeffreys, B.: *Methods of mathematical physics*. Cambridge university press (1999) 2
15. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 406–413. IEEE (2014) 10, 12, 13

16. Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., Park, J.: Self-calibrating neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5846–5854 (2021) [3](#)
17. Jiang, C., Huang, J., Tagliasacchi, A., Guibas, L.J.: Shapeflow: Learnable deformation flows among 3d shapes. *Advances in Neural Information Processing Systems* **33**, 9745–9757 (2020) [4](#)
18. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19729–19739 (2023) [1](#)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014) [8](#)
20. Lei, J., Danilidis, K.: Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6624–6634 (2022) [4](#)
21. Levoy, M.: Efficient ray tracing of volume data. *ACM Transactions on Graphics (TOG)* **9**(3), 245–261 (1990) [5](#)
22. Li, Y., Liang, Y.: Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems* **31** (2018) [2](#), [3](#)
23. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5741–5751 (2021) [1](#), [2](#), [3](#), [4](#), [5](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#)
24. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)* **38**(4), 1–14 (2019) [10](#), [11](#)
25. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021) [1](#), [2](#), [4](#), [5](#)
26. Neyshabur, B., Tomioka, R., Srebro, N.: Norm-based capacity control in neural networks. In: Conference on learning theory. pp. 1376–1401. PMLR (2015) [2](#), [3](#)
27. Nguyen, Q.N., Mondelli, M.: Global convergence of deep networks with one wide layer followed by pyramidal topology. *Advances in Neural Information Processing Systems* **33**, 11961–11972 (2020) [2](#), [3](#)
28. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Occupancy flow: 4d reconstruction by learning particle dynamics. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5379–5389 (2019) [4](#)
29. Oymak, S., Soltanolkotabi, M.: Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory* **1**(1), 84–105 (2020) [3](#)
30. Park, K., Henzler, P., Mildenhall, B., Barron, J.T., Martin-Brualla, R.: Camp: Camera preconditioning for neural radiance fields. *ACM Transactions on Graphics (TOG)* **42**(6), 1–11 (2023) [3](#)
31. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5865–5874 (2021) [1](#), [5](#)
32. Parra, A., Chng, S.F., Chin, T.J., Eriksson, A., Reid, I.: Rotation coordinate descent for fast globally optimal rotation averaging. In: Proceedings of the

- IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4298–4307 (2021) [1](#)
33. Paschalidou, D., Katharopoulos, A., Geiger, A., Fidler, S.: Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3204–3215 (2021) [4](#)
34. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) [1](#)
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**(1), 1929–1958 (2014) [2](#), [3](#)
36. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., et al.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–12 (2023) [10](#)
37. Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4190–4200 (2023) [3](#), [10](#), [11](#), [12](#)
38. Wang, C., MacDonald, L.E., Jeni, L.A., Lucey, S.: Flow supervision for deformable nerf. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21128–21137 (2023) [5](#)
39. Wang, Q., Chang, Y.Y., Cai, R., Li, Z., Hariharan, B., Holynski, A., Snavely, N.: Tracking everything everywhere all at once. arXiv preprint arXiv:2306.05422 (2023) [4](#), [5](#)
40. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf-: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021) [1](#), [3](#), [5](#)
41. Wu, T., Wang, J., Pan, X., Xu, X., Theobalt, C., Liu, Z., Lin, D.: Voxurf: Voxel-based efficient and accurate neural surface reconstruction. arXiv preprint arXiv:2208.12697 (2022) [10](#), [14](#)
42. Xia, Y., Tang, H., Timofte, R., Van Gool, L.: Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. arXiv preprint arXiv:2210.04553 (2022) [3](#), [5](#)
43. Xu, C., Wu, B., Hou, J., Tsai, S., Li, R., Wang, J., Zhan, W., He, Z., Vajda, P., Keutzer, K., et al.: Nerf-det: Learning geometry-aware volumetric representation for multi-view 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23320–23330 (2023) [1](#)
44. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5438–5448 (2022) [1](#)
45. Yang, G., Belongie, S., Hariharan, B., Koltun, V.: Geometry processing with neural fields. *Advances in Neural Information Processing Systems* **34**, 22483–22497 (2021) [4](#)
46. Zhan, H., Zheng, J., Xu, Y., Reid, I., Rezatofghi, H.: Activermap: Radiance field for active mapping and planning. arXiv preprint arXiv:2211.12656 (2022) [1](#)
47. Zhao, F., Yang, W., Zhang, J., Lin, P., Zhang, Y., Yu, J., Xu, L.: Humannerf: Efficiently generated human radiance field from sparse inputs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7743–7753 (2022) [1](#)

48. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12786–12796 (2022) [1](#)