Gaussian Activated Neural Radiance Fields for High Fidelity Reconstruction & Pose Estimation

Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey

Australian Institute for Machine Learning
University of Adelaide
{shinfang.chng,sameera.ramasinghe,jamie.sherrah,simon.lucey}@adelaide.edu.au

Abstract. Despite Neural Radiance Fields (NeRF) showing compelling results in photorealistic novel views synthesis of real-world scenes, most existing approaches require accurate prior camera poses. Although approaches for jointly recovering the radiance field and camera pose exist, they rely on a cumbersome coarse-to-fine auxiliary positional embedding to ensure good performance. We present Gaussian Activated Neural Radiance Fields (GARF), a new positional embedding-free neural radiance field architecture – employing Gaussian activations – that is competitive with the current state-of-the-art in terms of high fidelity reconstruction and pose estimation.

Keywords: neural scene representation, joint scene reconstruction and pose estimation, coordinate network, view synthesis, 3D deep learning

1 Introduction

Recent work by Lin *et al.* [12] – Bundle Adjusted Neural Radiance Fields (BARF) – revealed that an architecturally-modified Neural Radiance Field (NeRF) [18] could effectively solve the joint task of scene reconstruction and pose optimization. One crucial insight from this work is that the error backpropagation to the pose parameters in traditional NeRF is hampered by large gradients due to the high-frequency components in the positional embedding. To ameliorate this effect, the authors proposed a coarse-to-fine scheduler to gradually enable the frequency support of the positional embedding layer throughout the joint optimisation. Although achieving impressive results, this workaround requires careful tuning of the frequency scheduling process through a cumbersome *multi-dimensional* parameter sweep. In this paper we investigate if this coarse-to-fine strategy can be bypassed through other means; simplifying the approach and potentially opening up new avenues for improvement.

NeRF is probably the most popular application of coordinate multi-layer perceptrons (MLPs). NeRF maps an input 5D coordinate (3D position and 2D viewing direction) to the scene properties (view-dependent emitted radiance and volume density) of the corresponding location. A crucial ingredient of most coordinate MLPs is positional encoding. Traditional MLPs suffer from *spectral-bias* – *i.e.,* they are biased towards learning low-frequency functions – when used for

signal reconstruction [22]. Thus, MLPs, in their rudimentary form, are not ideal for encoding natural signals with fine detail, which entails modeling large fluctuations [23]. To circumvent this issue, NeRF architecturally modifies the MLPs by projecting the low-dimensional coordinate inputs to a higher dimensional space using a positional embedding layer, which allows NeRF to learn high-frequency components of the target function rapidly [35, 18].

Recently, there has been an increasing advocacy towards self-contained coordinate networks. By replacing conventional activations (ReLU) with periodic activations, Sitzmann *et al.* [30] demonstrated that sine enables a MLP to learn high frequency functions without any type of positional embedding. However, sine-MLPs have been found experimentally to be sensitive to weight initialization [30, 24]. While Sitzmann *et al.* [30] proposed an initialization scheme that aids sine-MLPs to achieve faster convergence when solving for signal reconstruction, their deployment within NeRF has been limited, with most of the community still opting for positional embedding with conventional activations.
Contributions: In this paper, we draw inspiration from a recent work [24] which has advocated for a broader class of effective activation functions – beyond sine – that can also circumvent the need for positional encoding. Of particular note in this regard are Gaussian activations. To our knowledge, their use in simultaneous neural reconstruction and pose estimation has not been previously explored. We show that Gaussian activations can preserve the first-order gradients of the target function better than conventional activations enhanced with positional embedding layers. When applied to BARF – that is simultaneously solving for pose and radiance field reconstruction – sine-MLPs are quite susceptible to local minima (even with good initialization), but our proposed Gaussian Activated Neural Radiance Fields (GARF) exhibit robust state-of-the-art performance.

In summary, we present the following contributions:

- We present GARF, a self-contained approach for reconstructing neural radiance field from imperfect camera poses without cumbersome hyper-parameter tuning and model initialisation.
- We establish theoretical insights of the effect of Gaussian activation in the joint optimisation problem of neural radiance field and camera poses, supported by an extensive empirical results.

We demonstrate that our proposed GARF can successfully recover scene representations from unknown camera poses, even on challenging scenes with low-textured regions, paving the way for unlocking NeRF for real-world applications.

2 Related Work

2.1 Neural Scene Representations.

Recent works have demonstrated the potential of multi-layer perceptrons or also known as MLPs as *continuous* and *memory efficient* representation for 3D geometry, including shapes [5, 4], objects [16, 1, 20] or scenes [31, 8, 30]. Using 3D

data such as point clouds as supervision, these approaches typically optimise signed distance functions [20, 8] or binary occupancy fields [16, 4]. To alleviate the dependency of 3D training data, several methods formulate differentiable rendering functions which enables the networks to be optimised using multiview 2D images [31, 19, 18, 36]. Of particular interest is NeRF [18], which models the continuous radiance field of a scene using a coordinate-MLP in a volume rendering framework by minimising the photometric errors. Due to its simplicity and unprecedented high fidelity novel view synthesis, NeRF has attracted wide attention across the vision community [21, 2, 14, 37, 34, 44]. Numerous extensions have been made on many fronts, e.g., faster training and inference [2, 43, 27, 13], deformable fields [21], dynamic scene modeling [11, 40, 3], generalisation [38, 29] and pose estimation [12, 39, 42, 15, 7, 33, 32].

2.2 Positional Embedding for Pose Estimation.

Positional embedding is an integral component of MLPs [35, 25, 46] which enable them to learn high frequency functions in a low dimensional domain. One of the earliest roots of this approach can be traced to the work by Rahimi *et al.* [23], who discovered that random Fourier Features can be used to approximate an arbitrary stationary kernel function. Leveraging such an insight, Mildenhall et al [18, 35] recently demonstrated that encoding input coordinates with sinusoids allows MLPs to represent higher frequency content, which enables a high-fidelity neural scene reconstruction in novel view synthesis.

Despite the ability of positional embedding in enabling MLPs to represent high frequency components, it is critical to choose the right frequency scale which often involves a cumbersome parameter tuning. If the bandwidth of the signal is increased excessively, a coordinate-MLP tends to produce noisy signal interpolations [35, 26, 6].

More recently, there has been an increasing interest in using coordinate-MLPs to tackle the joint problem of neural scene reconstruction and pose optimization [12, 39, 42, 15, 7, 33, 32, 47]. Remarkably, Lin *et al.* [12] demonstrated that coordinate-MLPs entail an unanticipated drawback in camera registration – *i.e.*, large gradients due to the high frequency components in the positional encoding function could hamper the error backpropagation to the pose parameters. Based on this observation, they proposed a work-around to anneal each component of the frequency function in a coarse-to-fine manner. By enabling a smoother trajectory for the optimisation problem, they show that such a strategy can lead to better pose estimation, compared to *full* positional encoding. Unlike BARF, we take a different stance – is there a *self-contained* architecture which can tackle the pose estimation problem optimally and simultaneously attain a high fidelity neural scene reconstruction without a positional embedding?

2.3 Embedding-free Coordinate-networks.

Sitzmann *et al.* [30] alternatively proposed sinusoidal activation functions which enable coordinate MLPs to encode high frequency functions without a positional

embedding layer. Despite its potential, networks that employ sinusoidal activations are hyper-sensitive to the initialisation scheme [30, 24, 26]. Taking a step further, Ramasinghe *et al.* [24], recently broadened the understanding of the effect of different activations in MLPs. They proposed a class of novel *non-periodic* activations that can enjoy more robust performance against random initialisation than sinuosoids. Our work significantly differs from the above-mentioned works. While we also advocate for a simple and robust embedding-free coordinate network, our work focuses on the joint problem of high fidelity neural scene reconstruction and pose estimation.

3 Method

In this section, we will provide an exposition of our problem formulation and different classes of coordinate networks, characterising the relative merits of each class for joint optimisation of neural scene reconstruction and pose estimation.

3.1 Formulation

We first present the formulation of recovering the 3D neural radiance field from NeRF [18] jointly with camera poses. We denote $\mathcal{T}$ as the camera pose transformations, and F as the network in NeRF, respectively. NeRF encodes the volumetric field of a 3D scene using a coordinate-network as $F : \mathbb{R}^3 \rightarrow \mathbb{R}^4$, which maps each input 3D coordinate $\mathbf{x} \in \mathbb{R}^3$ to its corresponding volume density $\sigma \in \mathbb{R}$ and directional emitted colour $\mathbf{c} \in \mathbb{R}^3$, i.e., $F(\mathbf{x}; \boldsymbol{\Theta}) = [\mathbf{c}, \sigma]$, where $\boldsymbol{\Theta}$ is the network weights [1].

Let $\mathbf{u} \in \mathbb{R}^2$ be the pixel coordinates, $\mathcal{I} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ be the imaging function. Given a set of images $\{\mathcal{I}_i\}_{i=1}^M$, we aim to solve for a volumetric radiance field $\boldsymbol{\Theta}$ of a 3D scene and the camera poses $\{\mathbf{p}_i\}_{i=1}^M$ by minimizing the photometric loss

$$\min_{\{\mathbf{p}_i\}_{i=1}^M \in \mathfrak{se}(3), \boldsymbol{\Theta}} \sum_{i=1}^{M} \sum_{\mathbf{u} \in \mathbb{R}^2} \|\hat{\mathcal{I}}(\mathbf{u}; \mathbf{p}_i, \boldsymbol{\Theta}) - \mathcal{I}_i(\mathbf{u})\|_2^2. \tag{1}$$

First, we assume the rendering operation of NeRF in the camera coordinate system. Expressing the pixel coordinate in its homogeneous coordinate as $\tilde{\mathbf{u}}$, we can define a 3D point $\mathbf{x}_i$ along a camera ray sampled at depth t_i as $\mathbf{x}_i = t_i \tilde{\mathbf{u}}$. The estimated RGB colour of $\hat{\mathcal{I}}$ at pixel coordinate $\mathbf{u}$ is then computed by aggregating the predicted $\mathbf{c}$ and σ as

$$\hat{\mathcal{I}}(\mathbf{u}) = \int_{t_n}^{t_f} T(\mathbf{u}, t) \sigma(t\tilde{\mathbf{u}}) \mathbf{c}(t\tilde{\mathbf{u}}) dt \tag{2}$$

where $T(\mathbf{u}, t) = \exp\Big(-\int_{t_n}^{t} \sigma(t'\tilde{\mathbf{u}})\Big) dt'$, and t_n and t_f are the bounds of the depth range of interest; see [10] for more details of volume rendering operation. In practice, the integral is commonly approximated using quadrature [18]

[1] f is also conditioned on viewing direction for modeling view-dependent effect, for which we omit here in the derivation for simplicity.

which evaluates the network F at a discrete set of N points through stratified sampling [18] at depth $\{t_1, ..., t_N\}$. Therefore, this entails N querying of the network F, whose output $\{\mathbf{y}_i\}_{i=1}^N$ are composited through volume rendering. Denoting the ray compositing function as $G : \mathbb{R}^{4N} \rightarrow \mathbb{R}^3$, we can rewrite $\tilde{\mathcal{I}}(\mathbf{u})$ as $\tilde{\mathcal{I}}(\mathbf{u}) = G(\mathbf{y}_1, ..., \mathbf{y}_N)$. Given a camera pose $\mathbf{p}$, we can transform a 3D point $\mathbf{x}$ in the camera coordinate system to the world coordinate system through a 3D rigid transformation $\mathcal{T}$ to obtain the synthesized image as

$$\hat{\mathcal{I}}(\mathbf{u}; \mathbf{p}) = G\Big(\{F\big(\mathcal{T}(t_i\tilde{\mathbf{u}}; \mathbf{p}); \mathbf{\Theta}\big)\}_{i=1}^N\Big). \tag{3}$$

We solve the optimization problem (1) using gradient descent. Next, we will give a brief exposition of coordinate-networks and compare them.

3.2 Coordinate-networks

Coordinate-networks are a special class of MLPs that are used to encode signals as trainable weights. An MLP with L layers can be formulated as

$$F(\mathbf{x}) = (g^{[L]} \circ \Phi^{[L-1]} \circ g^{[L-1]} \circ \dots \Phi^{[1]} \circ g^1)(\mathbf{x}^1) + \mathbf{b}^{[L]}, \tag{4}$$

where $g^{[l]} = \mathbf{W}^{[l]}\mathbf{x}^{[l]} + \mathbf{b}^{[l]}$, $\mathbf{W}^{[l]}$ are trainable weights at the l^{th} layer, $\mathbf{b}^{[l]}$ is the bias, and $\Phi^{[l]}(\cdot)$ is a non linear function. With this definition in hand, we briefly discuss several types of coordinate-networks below.

ReLU-MLPs: employ the ReLU activation function $\Phi(x) = max(0, x)$. Despite being a universal approximator in theory, ReLU-MLPs are biased towards learning low-frequency functions [41, 22], making them sub-optimal candidates for encoding natural signals with high fidelity. To circumvent this issue, various methods have been proposed in the literature, which we shall discuss next.

PE-MLPs: are the most widely adapted class of coordinate-networks and were popularized by the seminal work of [18] through the use of positional embedding (PE). In PE-MLPs, the low-dimensional input coordinates are projected to a higher-dimensional hypersphere via a positional embedding layer $\gamma(\mathbf{x}) \in \mathbb{R}^3 \rightarrow \mathbb{R}^{3+6D}$, which takes the form

$$\gamma(\mathbf{x}) = \big[\mathbf{x}, [\sin(2\pi\mathbf{x}), \cos(2\pi\mathbf{x})], ..., [\sin\big(2^{D-1}\pi\mathbf{x}\big), \cos\big(2^{D-1}\pi\mathbf{x}\big)]\big], \tag{5}$$

where D is a hyper-parameter that controls the total number of frequency bands. After computing (5), the embedded 3D input points are then passed through a conventional ReLU-MLPs to obtain $F(\gamma(\mathbf{x}); \mathbf{\Theta})$.

Sine-MLPs: are a coordinate-network without a positional embedding proposed by [30]. In sine-MLPs, the activation function is a sinusoid of the form

$$\mathbf{x}^{[l]} \mapsto \Phi^{[l]}(\mathbf{x}^{[l]}) = \sin\Big(2\pi\omega_o \mathbf{x}^{[l]}\Big), \tag{6}$$

where w_0 is a hyperparameter. A larger w_0 increases the bandwidth of the network, allowing it to encode increasingly higher frequency functions.

Gaussian-MLPs: are a recent class of positional-embedding less coordinate-networks [24], where the activation function is defined as

$$\mathbf{x}^{[l]} \mapsto \Phi^{[l]}(\mathbf{x}^{[l]}) = \exp\left(\frac{-{\mathbf{x}^{[l]}}^2}{2\sigma^2}\right), \tag{7}$$

where σ is a hyperparameter that can be used to tune the bandwidth of the network: a larger σ corresponds to a lower bandwidth, and vise-versa.

3.3 GARF for Reconstruction and Pose Estimation

In this paper, we advocate the use of Gaussian-MLPs for jointly solving pose estimation and scene reconstruction, and show substantial empirical evidences that they yield better accuracy and easier optimization over the other choices. We speculate the reason for this superior performance as follows. The pose parameters are optimized using the gradients flow through the network. Hence, the ability to accurately represent the first-order derivatives of the encoded signal plays a key role in optimizing pose parameters. However, Sitzmann *et al.* [30] showed that PE-MLPs are incapable of accurately model first-order derivatives of the target signal, resulting in noisy artifacts. This impacts the Fourier spectrum of the network function, which is implicitly related to the derivatives. As shown in [26], the Fourier transform $f(\mathbf{k})$ of a shallow Gaussian-MLP is

$$f(\mathbf{k}) = \sum_{i=1}^{m} w_i^{(2)} \frac{(2\pi)^{\frac{n+1}{2}}\sigma}{|\mathbf{w}_i^{(1)}|} e^{-\left(\sqrt{2}\pi \frac{\mathbf{w}_i^{(1)}}{|\mathbf{w}_i^{(1)}|^2}\cdot \mathbf{k}\sigma\right)^2} \delta_{\mathbf{w}_i^{(1)}}(\mathbf{k}), \tag{8}$$

where $\mathbf{k}$ is the frequency index, $\delta_{\mathbf{w}}(\mathbf{k})$ is the Dirac delta distribution which concentrates along the line spanned by $\mathbf{w}$, and $\mathbf{w}^{(i)}$ are the weight vectors corresponding to the i^{th} layer. Note that (8) is a smooth distribution, which is parameterized by σ and $\mathbf{w}^{(i)}$'s. In other words, for a suitably chosen σ, the bandwidth of the network can be increased in a continuous manner by appropriately learning the weights. Moreover, as σ is a continuous parameter, it provides MLPs with the ability to smoothly manipulate the spectrum of the network.

In contrast, [45] demonstrated that spectrum of a PE-MLP tends to consist of discrete spikes, where the spikes are placed on the integer harmonics of the

Fig. 1. A 2D planar image alignment instance for (10). *Left*: Input image patches with $N = 6$. *Right*: The initial poses (represented as black box) are initialised as identity.

positional embedding frequencies. Approximating the ReLU function via a polynomial in the form $\rho(x) = \sum_{i=1}^{K} \alpha_i x^i$, where α_i are constants, they showed that the spectrum is concentrated on the frequency set

$$\Big\{ \sum_{d=1}^{D} s_d 2^d \pi | s_d \in \mathbb{Z} \wedge \sum_{d=1}^{D} |s_d| < K \Big\}. \tag{9}$$

Recall that in order to increase the frequency support of the positional embedding layer, one needs to increase D. It is evident that increasing D even by one adds many harmonic spikes on the spectrum at the high-frequency end, irrespective of the network weights. Therefore, it is not possible to manipulate the spectrum of the PE-MLP continuously under a controlled setting. This can result in unnecessary high-frequency components that lead to unwanted artifacts.

On the other hand, sine-MLPs are able to construct rich spectra and represent first-order derivatives accurately [30]. However, sine-MLPs are extremely sensitive to initialization. Sitzmann *et al.* [30] proposed an initialization scheme for sine-MLPs in signal reconstruction, under which they show strong convergence properties. However, we empirically demonstrate that when jointly optimizing for the pose parameters and scene reconstruction, the above initialization yields sub-par performance, making sine-MLPs highly likely to get trapped in local minima. We also show that, in comparison, Gaussian-MLPs exhibit far superior convergence properties, indicating that they entail a simpler loss landscape.

4 Experiments

This section validates and analyses the effectiveness of our proposed GARF with other coordinate networks. We first unfold the analysis on a 2D planar image alignment problem, and demonstrate extensive results on learning NeRF from unknown camera poses.

4.1 2D Planar Image Alignment

To develop intuition, we first consider the case of 2D planar image alignment problem. More specifically, let $\mathbf{u} \in \mathbb{R}^2$ be the 2D pixel coordinates and $\mathcal{I}$:

Fig. 2. Qualitative and quantitative results of the 2D planar image alignment problem. *Left*: Visualisation of the estimated poses with the $\mathfrak{sl}(3)$ error. *Center*: Reconstruction of each warped patches. *Right*: Final image reconstruction with the patch PSNR.

$\mathbb{R}^2 \to \mathbb{R}^3$, we aim to optimize a neural image representation parameterised as the weights of coordinate network F while also solving for warp parameters as

$$\min_{\{\mathbf{p}_i\}_{i=1}^N \in \mathfrak{sl}(3), \boldsymbol{\Theta}} \sum_{i=1}^{N} \sum_{\mathbf{u} \in \mathbb{R}^2} \| F(\mathcal{W}(\mathbf{u}; \mathbf{p}_i); \boldsymbol{\Theta}) - \mathcal{I}_i(\mathbf{u}) \|_2^2, \tag{10}$$

where $\mathcal{W} : \mathbb{R}^2 \to \mathbb{R}^2$ denotes the warp function parameterised as $\mathbf{p} \in \mathfrak{sl}(3)$. Given $N = 6$ patches from the image $\mathcal{I}$ generated with random homography perturbations, we aim to jointly estimate the *unknown* homography warp parameters $\mathbf{p}_i$ and network weights $\boldsymbol{\Theta}$. We fix the gauge freedom by anchoring the first patch as identity; see Fig. 1 for an example.

Experimental settings. We compare our proposed GARF with the following networks: PE-MLP with a coarse-to-fine embedding annealer (BARF) [12] and sine-MLP (SIREN) [30]. We use a 5-layer MLP with 256 hidden units for all

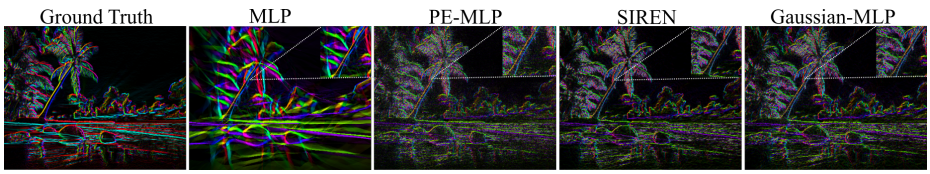

Fig. 3. Comparison of the first-order derivatives of encoded signal ∇F on solving an image reconstruction problem (Best viewed in electronic version). The first-order derivative of each function is computed using network's output with respect to the coordinates. Note that *only* the groundtruth derivative is computed using Sobel Filter.

networks. We use the Adam optimizer to optimize both the network weights $\mathbf{\Theta}$ and the warp parameters $\mathbf{p}$. We use a learning rate that begins at 1×10^{-3} for $\mathbf{\Theta}$, and 3×10^{-3} for $\mathbf{p}$, with both decaying exponentially to 1×10^{-4} for GARF and BARF. For SIREN, we use a learning rate of 1×10^{-4} for both $\mathbf{\Theta}$ and $\mathbf{p}$ decaying exponentially to 1×10^{-5}. For BARF, we use $D = 8$ frequency bands Eq. (5), which is linearly annealed over 12K iterations. At each optimization step, we randomly sample 15% of the pixel coordinates for each patch.

Initialisation. For BARF and SIREN, we use the initialisation scheme proposed in the original paper [18, 12, 30], whereas for our proposed GARF we simply use randomly initialised weights. We initialise the warp parameters $\{\mathbf{p}_i\}_{i=1}^{N}$ as identity for all models; see Fig. 1.

Results. We demonstrate the quantitative and qualitative registration results in Fig. 2. As GARF is able to correctly estimate the warp parameters of all patches, GARF can reconstruct the image with high fidelity. On the other hand, BARF and SIREN struggle with the image reconstruction due to misalignment. It is important to note that the Gaussian-MLP initialisation protocol put the proposed method at a disadvantage. This further demonstrates the robustness of Gaussian-MLP towards initialisation.

First-order derivatives analysis. For completeness, we first inspect the first-order derivations of each coordinate network when solving for an image reconstruction task as $\min_{\mathbf{\Theta}} \sum_{\mathbf{u} \in \mathbb{R}^2} \|F(\mathbf{u}; \mathbf{\Theta}) - \mathcal{I}(\mathbf{u})\|_2^2$; note that we use the same notations as in (10). As discussed in Sec. 3.3, the ability to accurately represent the first-order derivatives of the encoded signal plays a crucial role in optimizing pose parameters. Fig. 3 reinforces that the first-order derivative of the encoded signal of PE-MLP has a lot of noise artifacts – results in poor error backpropagation to pose parameters. Although a properly-initialised SIREN is capable of representing the derivatives of the signal when solving for signal reconstruction, the initialisation strategy of sine-activation is sub-optimal when jointly optimizing for neural image reconstruction and warp. As a result, the resulting function

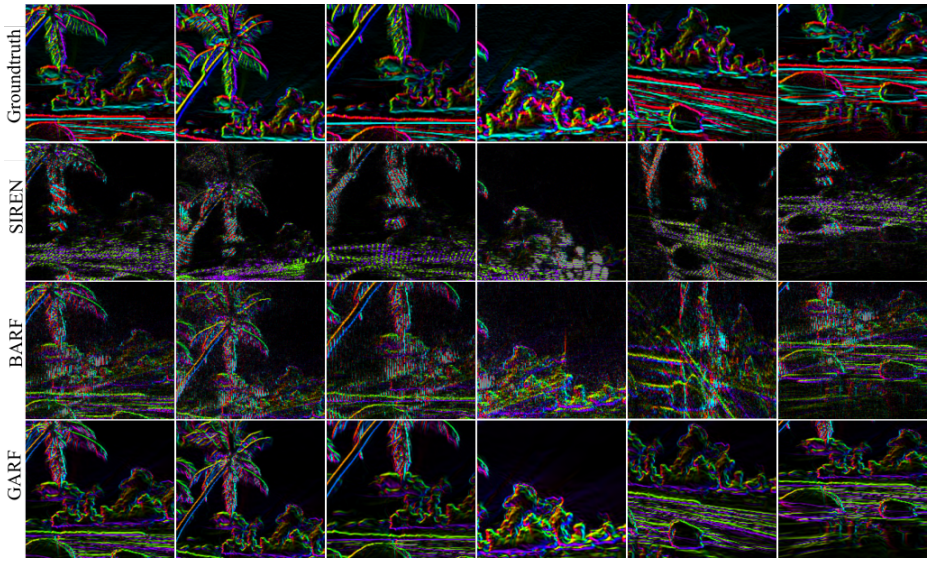

Fig. 4. Comparison of the first-order derivatives of encoded signal ∇F. The first-order derivative of each function is computed using network's output with respect to the coordinates. Note that *only* the groundtruth derivative is computed using Sobel Filter.

derivative is no longer well-defined; see Fig. 4. In contrast, GARF exhibit far superior convergence properties, albeit the model weights are initialised randomly.

Robustness of initialisation scheme. Additionally, we run an experiment to investigate the sensitivity of SIREN and GARF to initialisation. We denote $\boldsymbol{\Theta}^*$ as the optimal model weights, which is obtained by solving (10) for a neural image representation by fixing the warp parameters, and $\bar{\boldsymbol{\Theta}}$ as the randomly initialised model weights, *i.e.*, weights are initialised using PyTorch default initialisation. Our goal is to solve the joint optimisation problem (10) by initialising $\boldsymbol{\Theta}$ with different scaled model weights, *i.e.,* $\alpha\bar{\boldsymbol{\Theta}} + (1-\alpha)\boldsymbol{\Theta}^*$ by linearly adjusting α. As shown in Fig 5, GARF (*green curve*) is marginally affected by the initialisation, while SIREN (*blue curve*) fails drastically (starting from α=0.3). When SIREN is initialised carefully using the initialisation scheme proposed by Sitzmann *et al.* [30] (*red curve*), its performance decreases as α gradually increases, *i.e.*, as the perturbation to the optimal model weights increases. Note that the variance of performance in the GARF is much smaller compared to SIREN.

Generalisation of coarse-to-fine scheduling We exhaustively search through the log-space for the optimal coarse-to-fine schedulers for BARF; see supp. material for more details. The optimal coarse-to-fine hyper-parameters for each image are data-dependent, *i.e.*, the hyperparameters tuned for one image may

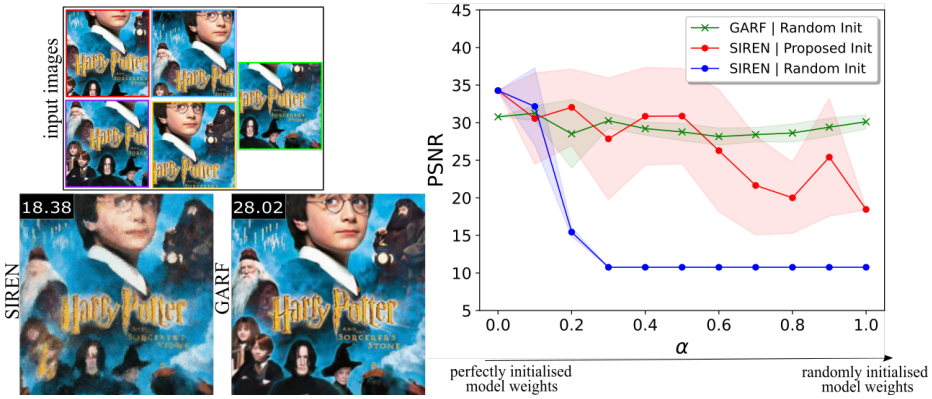

Fig. 5. *Left*: Input images and the image reconstruction for SIREN and GARF, which correspond to the *red* and *green* curve, respectively. *Right*: Robustness of the initialisation at different α. When $\alpha = 0$, all the networks are initialised with *optimal* weights; When $\alpha = 1$, all the networks are initialised with *random* weights. Note that for SIREN, we also investigate the case when SIREN strictly adheres to the initialisation scheme proposed by Sitzmann *et al.* [30] (*red*). The shaded regions correspond to the two standard deviations over 10 runs.

not be optimal for another image. In contrast to the multi-dimensional scheduler, Gaussian activation involves one-dimensional search space (7).

4.2 3D NeRF: Real World Scenes

This section investigates the task of jointly learning neural 3D representations with NeRF [18] on real world scenes where the camera poses are *unknown*. We evaluate all the methods on the standard benchmark LLFF dataset [17], which consists of 8 real world forward-facing scenes captured by hand-held cameras.

Experimental Settings. We compare our proposed GARF with BARF and reference NeRF (ref-NeRF). As we empirically observe that PE-MLP with scheduler (BARF) achieves better performance compared to PE-MLP [39] in the joint optimisation of neural radiance field and camera poses, we opted not to include the comparisons with PE-MLP here; see [12] or supp. for comparisons with PE-MLP. We parameterise the camera poses with the $\mathfrak{se}(3)$ Lie algebra and initialise them as *identity* for GARF and BARF. We assume known intrinsics.

4.3 Implementation Details.

We implement our framework following the settings from [18, 12] with some modifications. For simplicity, we train a 8-layer MLP with 256 hidden units in each layer and *without hierarchical sampling*. We resize the images to 480×640 pixels and randomly sample 2048 pixel rays every iteration, each sampled at

Table 1. Quantitative comparison of GARF (Ours), BARF [12] and ref-NeRF on real-world scenes [17] given *unknown* camera poses.

Scene	Pose accuracy				View synthesis								
	Rotation (°)		Translation (10^{-2})		PSNR ↑ (dB)			SSIM ↑			LPIPS ↓		
	[12]	Ours	[12]	Ours	[12]	Ours	ref-NeRF	[12]	Ours	ref-NeRF	[12]	Ours	ref-NeRF
flower	0.47	**0.46**	0.25	**0.22**	23.58	**26.40**	23.20	0.67	**0.79**	0.66	0.27	**0.11**	0.27
fern	**0.16**	0.47	**0.20**	0.25	23.53	**24.51**	23.10	0.69	**0.74**	0.71	0.34	**0.29**	**0.29**
leaves	1.00	**0.13**	0.30	**0.23**	18.15	**19.72**	14.42	0.48	**0.61**	0.24	0.40	**0.27**	0.58
horns	0.80	**0.03**	**0.17**	0.21	**23.03**	22.54	19.93	**0.73**	0.69	0.59	**0.29**	0.33	0.45
trex	**0.42**	0.66	**0.36**	0.48	22.63	**22.86**	21.42	0.75	**0.80**	0.69	0.24	**0.19**	0.32
orchids	0.71	**0.43**	0.42	**0.41**	19.14	**19.37**	16.54	0.55	**0.57**	0.46	0.33	**0.26**	0.37
fortress	0.17	**0.03**	0.32	**0.27**	28.48	**29.09**	25.62	0.80	**0.82**	0.78	0.16	**0.15**	0.19
room	**0.27**	0.42	**0.20**	0.32	31.43	**31.90**	31.65	0.93	**0.94**	0.94	0.11	0.13	**0.09**

$N = 128$ coordinates. We use the Adam optimizer [9] and train all models for 200K iterations, with a learning rate that begins at 1×10^{-4} decaying exponentially to 5×10^{-5}, and 3×10^{-3} for the poses $\mathbf{p}$ decaying to 1×10^{-5}. We use the default coarse-to-fine scheduling for BARF [12]. We use the same network size and sampling strategy for all the methods throughout our evaluation. Note that for BARF and ref-NeRF, we use the implementation from BARF; all the hyperparameters are configured as per proposed in the paper.

Evaluation Details. We evaluate the performance of each method in terms of pose accuracy for registration and view synthesis quality for the scene reconstruction. Following [12, 39], we evaluate the pose error by aligning optimized poses to groundtruth via Proscustes analysis which computes the similarity transformation Sim(3) between them. Note that as the "groundtruth" camera poses provided in LLFF real-world scenes are the estimations from Colmap [28], the pose accuracy is only an indicator how well the estimations agree with the classical method. We report the mean rotation and translation errors for pose, as well as PSNR, SSIM and LPIPS [18] for view synthesis in Table 1.

Results. Table 1 quantitatively contrasts the performance of GARF, BARF and ref-NeRF. As evident, Gaussian activations enable GARF to recover camera poses which matches the camera poses from off-the-shelf SfM methods. Moreover, Gaussian activations can successfully recover the 3D scene representation with higher fidelity in the absence of positional embedding, compared to BARF and ref-NeRF; see the qualitative results in Fig. 6.

4.4 Real-World Demo

To showcase the practicability of GARF, we take one step further to test it on images of low-textured scene captured using an iPhone. Fig. 7 remarkably

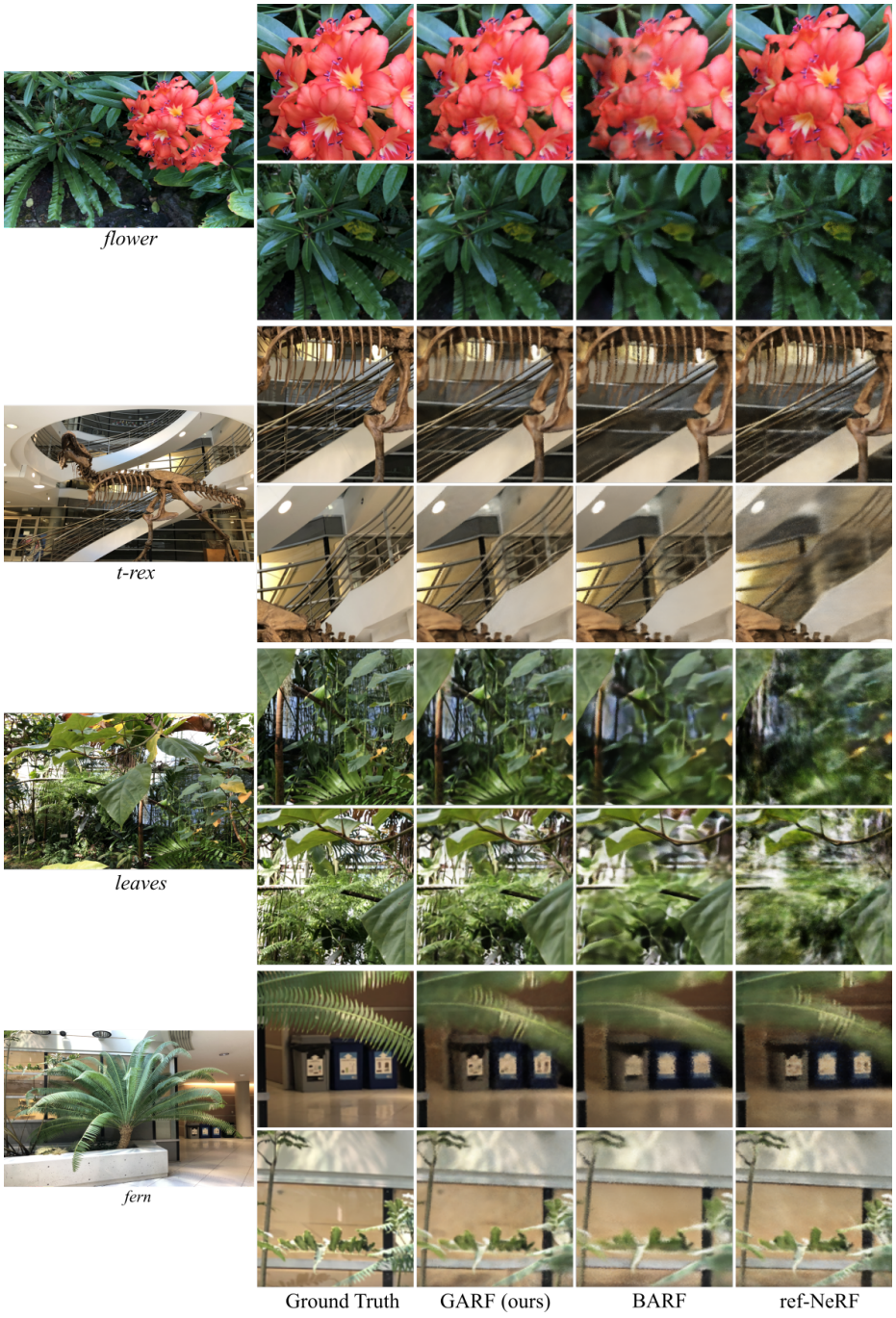

Fig. 6. Qualitative results on test-views of real world scenes [17]. While BARF and GARF can jointly optimize pose and the scene representation, GARF produces results with higher fidelity.

demonstrate the potential of GARF on a scene with a lot of low-textured region while ref-NeRF exhibits artifacts on the novel view due to existence of outliers in front-end of SfM pipeline, which results in unreliable camera pose estimations; see supp. for more results.

Fig. 7. Novel view synthesis result on a low-textured scene captured using iPhone. *Left banner*: Training images. *Top row*: Rendered image and depth using ref-NeRF. *Bottom row*: Rendered image and depth using GARF.

5 Conclusions

We present GARF, a new positional embedding-free architecture for the simultaneous neural radiance fields reconstruction and pose estimation problem without cumbersome hyperparameter and model initialisation. By establishing theoretical intuition, we demonstrate that the ability of the model to preserve the first-order gradients of the target function plays an imperative role in the joint optimization problem. Experimental results reinforced our theoretical intuition and demonstrated the superiority of GARF, even on challenging scenes with low textured region.

Despite the encouraging results, as with NeRF and its variants, GARF requires lengthy training time. Nevertheless, many of the current advances in NeRF could potentially be applied to speed up the training of GARF. We believe that there is still much more progress to be made in enabling GARF for real-time SLAM applications.

Acknowledgements

We thank Chen-Hsuan Lin, Huangying Zhan, and Tong He for fruitful discussions.

References

1. Chabra, R., Lenssen, J.E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., Newcombe, R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In: European Conference on Computer Vision. pp. 608–625. Springer (2020)
2. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. arXiv preprint arXiv:2107.02791 (2021)
3. Gao, C., Saraf, A., Kopf, J., Huang, J.B.: Dynamic view synthesis from dynamic monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5712–5721 (2021)
4. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3d shape. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4857–4866 (2020)
5. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7154–7164 (2019)
6. Hertz, A., Perel, O., Giryes, R., Sorkine-Hornung, O., Cohen-Or, D.: Sape: Spatially-adaptive progressive encoding for neural optimization. Advances in Neural Information Processing Systems **34** (2021)
7. 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)
8. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al.: Local implicit grid representations for 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6001–6010 (2020)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Levoy, M.: Efficient ray tracing of volume data. ACM Transactions on Graphics (TOG) **9**(3), 245–261 (1990)
11. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6498–6508 (2021)
12. 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)
13. Lindell, D.B., Martel, J.N., Wetzstein, G.: Autoint: Automatic integration for fast neural volume rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14556–14565 (2021)
14. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7210–7219 (2021)
15. Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., Yu, J.: Gnerf: Gan-based neural radiance field without posed camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6351–6361 (2021)
16. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)

17. 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)
18. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)
19. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020)
20. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)
21. 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)
22. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning. pp. 5301–5310. PMLR (2019)
23. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. Advances in neural information processing systems **20** (2007)
24. Ramasinghe, S., Lucey, S.: Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. arXiv preprint arXiv:2111.15135 (2021)
25. Ramasinghe, S., Lucey, S.: Learning positional embeddings for coordinate-mlps. arXiv preprint arXiv:2112.11577 (2021)
26. Ramasinghe, S., MacDonald, L., Lucey, S.: On regularizing coordinate-mlps. arXiv preprint arXiv:2202.00790 (2022)
27. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14335–14345 (2021)
28. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
29. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. Advances in Neural Information Processing Systems **33**, 20154–20166 (2020)
30. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems **33**, 7462–7473 (2020)
31. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. Advances in Neural Information Processing Systems **32** (2019)
32. Su, S.Y., Yu, F., Zollhoefer, M., Rhodin, H.: A-nerf: Surface-free human 3d pose refinement via neural rendering. arXiv preprint arXiv:2102.06199 (2021)
33. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6229–6238 (2021)

34. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. arXiv preprint arXiv:2202.05263 (2022)
35. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems **33**, 7537–7547 (2020)
36. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Wang, Y., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., et al.: Advances in neural rendering. arXiv preprint arXiv:2111.05849 (2021)
37. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. arXiv preprint arXiv:2112.10703 (2021)
38. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2021)
39. 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)
40. Xian, W., Huang, J.B., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9421–9431 (2021)
41. Xu, Z.Q.J., Zhang, Y., Xiao, Y.: Training behavior of deep neural network in frequency domain. In: International Conference on Neural Information Processing. pp. 264–274. Springer (2019)
42. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: inerf: Inverting neural radiance fields for pose estimation. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1323–1330. IEEE (2021)
43. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)
44. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4578–4587 (2021)
45. Yüce, G., Ortiz-Jiménez, G., Besbinar, B., Frossard, P.: A structured dictionary perspective on implicit neural representations. arXiv preprint arXiv:2112.01917 (2021)
46. Zheng, J., Ramasinghe, S., Lucey, S.: Rethinking positional encoding. arXiv preprint arXiv:2107.02561 (2021)
47. 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. arXiv preprint arXiv:2112.12130 (2021)