International Journal of Computer Vision (2024) 132:1310–1335
https://doi.org/10.1007/s11263-023-01936-1

Learning Robust Multi-scale Representation for Neural Radiance Fields from Unposed Images

Nishant Jain[1] · Suryansh Kumar[2,3] · Luc Van Gool[2]

Received: 5 April 2023 / Accepted: 17 October 2023 / Published online: 11 November 2023

Abstract

We introduce an improved solution to the neural image-based rendering problem in computer vision. Given a set of images taken from a freely moving camera at train time, the proposed approach could synthesize a realistic image of the scene from a novel viewpoint at test time. The key ideas presented in this paper are (i) Recovering accurate camera parameters via a robust pipeline from unposed day-to-day images is equally crucial in neural novel view synthesis problem; (ii) It is rather more practical to model object's content at different resolutions since dramatic camera motion is highly likely in day-to-day unposed images. To incorporate the key ideas, we leverage the fundamentals of scene rigidity, multi-scale neural scene representation, and single-image depth prediction. Concretely, the proposed approach makes the camera parameters as learnable in a neural fields-based modeling framework. By assuming per view depth prediction is given up to scale, we constrain the relative pose between successive frames. From the relative poses, absolute camera pose estimation is modeled via a graph-neural network-based multiple motion averaging within the multi-scale neural-fields network, leading to a single loss function. Optimizing the introduced loss function provides camera intrinsic, extrinsic, and image rendering from unposed images. We demonstrate, with examples, that for a unified framework to accurately model multiscale neural scene representation from day-to-day acquired unposed multi-view images, it is equally essential to have precise camera-pose estimates within the scene representation framework. Without considering robustness measures in the camera pose estimation pipeline, modeling for multi-scale aliasing artifacts can be counterproductive. We present extensive experiments on several benchmark datasets to demonstrate the suitability of our approach.

Keywords Neural radiance fields · Motion averaging · Multiscale representation · Single image depth prediction

1 Introduction

Using neural fields to represent a 3D scene from its multi-view (MV) images has recently become popular for solving novel view synthesis problems. This is primarily due to the Mildenhall et al. (2021) work on neural radiance fields popularly known as NeRF. NeRF's idea to scene representation has shown promising results on several computer vision, graphics, and robotics problems (Yu et al., 2021; Sucar et al., 2021; Zhang et al., 2021; Liu et al., 2020; Martel et al., 2021; Kaya et al., 2022; Lee et al., 2022). Yet, its original design choice has inherent challenges in handling day-to-day MV images captured from a freely moving camera. For instance, NeRF shows visual artifacts on multiple scale images (Barron et al., 2021), and its performance degrades even with subtle inaccuracies in camera pose estimates (Lin et al., 2021). Therefore, to make NeRF and similar approaches more usable for arbitrarily captured MV images, the approach must generalize to more realistic indoor and outdoor scenes with dramatic camera motion.

While recently proposed Mip-NeRF (Barron et al., 2021) solves the multiscale issues with NeRF, it assumes known camera parameters, i.e., ground-truth camera poses as well

Communicated by Guang Yang.

✉ Suryansh Kumar
k.sur46@gmail.com

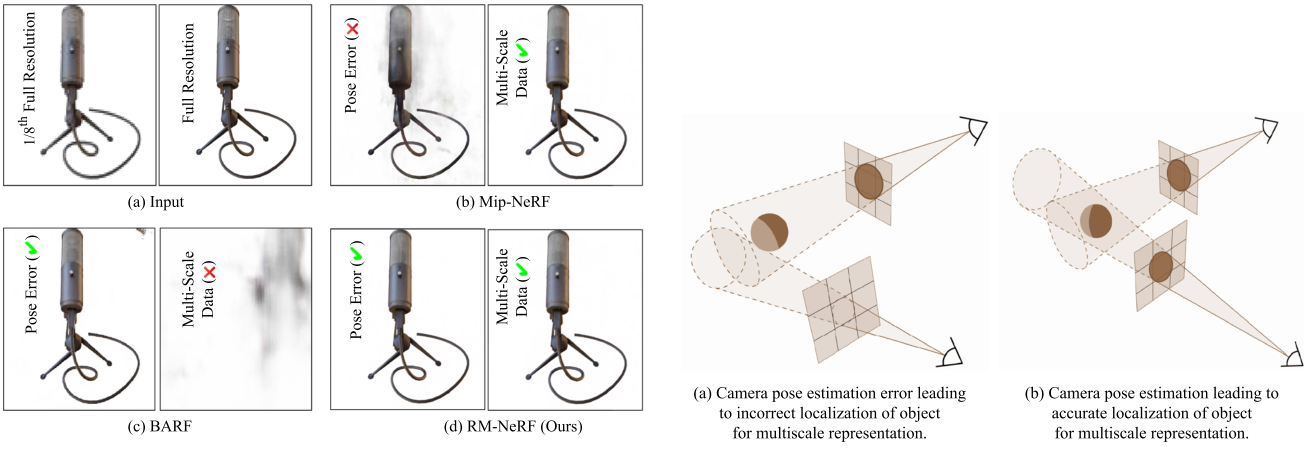
Nishant Jain
njain@cs.iitr.ac.in

Luc Van Gool
vangool@vision.ee.ethz.ch

[1] IIT Roorkee, Roorkee, India

[2] CVL Lab at ETH, Zurich, Switzerland

[3] School of PVFA, Texas A & M University, College Station Texas, USA



(a) Input

(b) Mip-NeRF

(c) BARF

(d) RM-NeRF (Ours)

(a) Camera pose estimation error leading to incorrect localization of object for multiscale representation.

(b) Camera pose estimation leading to accurate localization of object for multiscale representation.

(a) **Left**: Qualitative Comparison on Mic dataset

(b) **Right**: Intuition on importance of camera pose accuracy.

Fig. 1 Left: **a** Multi-scaled, multi-view images with camera pose error is fed to related NeRF-based methods (Mildenhall et al., 2021; Barron et al., 2021; Lin et al., 2021). **b** Barron et al. (2021) Mip-NeRF can handle multi-scale imaging effects but fails if the camera poses error persists. **c** Lin et al. (2021) BARF can handle the camera pose error for same-scale images but fails for multi-scale images. **d** Our approach works well for both cases. **Right**: **a** Error in the camera pose estimation can lead to incorrect cone casting in the volume space leading to misguided localization of the object for proper modeling. **b** Correct camera poses certify the proper modeling of the object volume for each sampled canonical frustum

as camera intrinsics are given, or estimated via off-the-shelf COLMAP software (Schonberger & Frahm, 2016). On the other hand, recent works such as BARF (Lin et al., 2021), NeRF–(Wang et al., 2021b), SC-NeRF (Jeong et al., 2021) introduced formulations to simultaneously estimate camera pose yet unsuitable for multiscale unposed images (Jain et al., 2022). Furthermore, available methods in this same vein often ignore the relative camera motion between images, which is a critical prior to absolute camera pose estimation. In both of these independent research directions, a gap exists, i.e., BARF (Lin et al., 2021) and similar methods can jointly solve the camera pose with neural fields representation but cannot address multiscale image issues. On the contrary, Mip-NeRF (Barron et al., 2021) can handle multi-scale images but assumes the correct camera pose. Hence, in this work, we introduce a simple and effective approach to fill this gap. By utilizing the fundamentals of scene rigidity, relative camera motion, and scene depth prior, we jointly address the multiscale issues and the challenges in camera parameter estimation within neural fields approaches (see Fig. 1a). Consequently, our self-contained approach performs well for handheld captured MV images.

To put the notions intuitively, we show in Fig. 1b that the correct intersection of the conical frustum for object localization—as proposed in Mip-NeRF (Barron et al., 2021), is possible if both the camera poses are correctly known. One trivial way to solve this is to jointly optimize for object representation and camera pose as done in BARF (Lin et al., 2021) and NeRF–(Wang et al., 2021b) with Mip-NeRF representation idea. As is known that the bundle-adjustment (BA) based joint optimization is complex, sub-optimal, requires good initialization, and can handle only certain types of noise and outlier distribution (Chatterjee & Govindu, 2017). So, conditioning the multi-scale rendering representation based on BA-type optimization could complicate the approach, hence not an encouraging take on the problem.

To solve the above mentioned challenges, we propose an approach that leverages the fundamentals of scene rigidity and other scene priors that could be estimated from images. Firstly, we estimate the camera motion robustly without having explicit information about the object's 3D position assuming a rigid scene (Govindu, 2001). Secondly, we estimate the geometric prior per frame without using any camera information by relying on single image depth prior (Ranftl et al., 2021). This helps in overcoming the object's geometry and radiance ambiguity in multi-scale neural radiance fields representation (Barron et al., 2021). Thirdly, we use relative camera motion prior between the frames with predicted depth to further improve the absolute camera pose solution between frames.

At the heart of the proposed approach lies the idea of disentangling geometry, radiance, and camera parameters in multi-scale neural radiance fields representation for making novel view synthesis more usable and practical. Our approach introduces graph-neural network-based multiple motion averaging with multi-scale feature modeling and per-frame depth prior to solving the problem. For single image depth prior, we rely on Ranftl et al. (2021) work. In this article, we claim the following contributions.

Contributions

- We propose a novel view synthesis approach to jointly estimate camera parameters and multi-scale scene representation from daily captured multi-view images.
- The introduced approach exploits rigid scene assumption to disentangle the camera motion estimation variables from explicit 3D geometry variables. Furthermore, radiance and shape ambiguity is resolved by utilizing the per-image scene depth prior.
- The proposed loss function utilizes multi-scale scene representation and per-view scene depth with graph neural network-based multiple motion averaging for robust camera pose parameters estimation leading to improved scene representation.

This article extends our published paper at the British Machine Vision Conference (BMVC), 2022 (Jain et al., 2022). Firstly, the proposed approach extends it to recover accurate scene representation and camera poses starting from entirely random poses. It is referred to as RM-NeRF (w/o pose). Another extension presented is that we further relax the requirement of camera intrinsics as input and recover correct scene representation from randomly initialized camera extrinsic and intrinsic parameters. We refer it as RM-NeRF (E2E). Thus, the proposed approach serves as a unified framework, eliminating the dependency on other third-party modules.

Experimental results show that our proposed extensions initialized with random intrinsic and extrinsic camera parameters, are quite effective. To test this, we presented the RM-NeRF (Jain et al., 2022) with very noisy pose initialization in one of the experiment and with COLMAP poses in the next experiment. We observed that the introduced extensions are quite effective and provides commendable results compared to the baseline experiments on day-to-day captured images, which we collected using our phone by randomly walking around an object. Our approach achieves better camera pose estimates and novel view synthesis results than the existing NeRF-based baseline methods when tested on the standard benchmark dataset (Mildenhall et al., 2021; Knapitsch et al., 2017). Additionally, our approach outperforms RM-NeRF (Jain et al., 2022) on tanks and temples dataset (Knapitsch et al., 2017) as well on recently proposed NAVI dataset (Jampani et al., 2023) under similar experimental settings. Refer to Sect. 4 for more details.

2 Related Works

Recently, neural radiance fields (NeRF) based implicit scene representation has gained significant attention in the computer vision and graphics community with several extensions. As a result, discussing all the NeRF-related methods is beyond the scope of the article, and interested readers may refer to Tewari et al. (2022) paper for reference. Here, we keep the related work discussion concise and concern ourselves with methods relevant to our proposed approach.

2.1 Neural Fields for Scene Representation

NeRF (Mildenhall et al., 2021) represents a rigid scene as a continuous volumetric field parametrized by a multi-layer perceptron (MLP). It assumes a fully calibrated setting with well-posed input images, i.e., correct camera pose and internal camera calibration matrix is known, and images are captured in a dome setting. Once the experimental setup is prepared, NeRF for each pixel sample points along rays that are traced from the camera's center of projection. Later, these sampled points are transformed using positional encoding to represent each point in a high-dimensional feature vector before being fed to an MLP for density and color estimation for novel view synthesis at test time.

(i) Multiscale NeRF. Barron et al. (2021) introduced Mip-NeRF to overcome the limitations with NeRF in rendering multi-resolution images, i.e., MV images are captured at a varying distance from the object. Instead of sampling points along the rays traced from the camera center of projection, Mip-NeRF queries samples along a conical frustum interval region approximated using 3D Gaussian to render the corresponding pixel. Since the image acquisition setup used in NeRF is unrealistic for many practical day-to-day captured videos, Mip-NeRF broadens the scope of NeRF formulation to commonly acquired multi-view and multi-scale image acquisition setups. Yet, the Mip-NeRF assumption on the availability of ground-truth camera pose parameters is uncommon and could substantially restrict its application.

(ii) Uncalibrated NeRF. Recently, a few methods have appeared to jointly solve camera pose and object's neural representation via NeRF formulation. For example, BARF (Lin et al., 2021) leverages photometric bundle adjustment to estimate the camera poses and recover scene representation jointly. Recently, NeRF−−(Wang et al., 2021b) introduced an approach for estimating intrinsic and extrinsic camera calibration while training the NeRF model. Nonetheless, these extensions of NeRF work well for the same scale images; accordingly, its usage is limited to a synthetic multi-view dome or hemispherical setup. Not long ago, NoPe-NeRF (Bian et al., 2022) utilized depth maps to estimate camera poses via point cloud alignment and a surface-based photometric loss. As a result, it can reconstruct the scene from randomly initialized poses. Other related work includes iNeRF (Yen-Chen et al., 2021) that solves camera poses given a well-trained NeRF model, and SC-NeRF (Jeong et al., 2021). The method jointly learns the camera parameters and scene representation using a loss function that enforces geometric consistency for a given camera model.

(iii) NeRF extension with other scene priors. There have been several attempts to make the NeRF approach either faster or more generalizable by extracting valuable features from the input images. One line of works (Yu et al., 2021; Wang et al., 2021a) involves estimating a feature volume from an image via a generalizable CNN and then feeding the feature vector into the MLP to generalize NeRF idea. Another line of works (Chen et al., 2021; Xu et al., 2022) estimates scene 3D structure prior via MVS-based methods (Yao et al., 2018). Combining input image features with recovered 3D structure, it learns better scene representation, and such approaches are shown to converge faster.

2.2 Camera Pose Estimation

Widely used approaches to camera pose estimation from multi-view images are based on image key-points matching and incrementally solve camera pose (Agarwal et al., 2011) or use global BA (Triggs et al., 2000) with five-point (Nistér, 2004) or eight-point algorithm (Hartley, 1997) running at the back-end. Yet, such methods can provide sub-optimal solutions and may not robustly handle outliers inherent to the unstructured images. To address such an intrinsic challenge with pose estimation, Govindu (Govindu, 2001) initiated and later authored a series of robust multiple rotation averaging (MRA) approaches (Govindu, 2016; Chatterjee & Govindu, 2017). The benefit of using MRA is that it uses multiple estimates of noisy relative motion to recover absolute camera pose based on view-graph representation and rotation group structure (Govindu, 2006) i.e., SO(3). Contrary to the robust conventional rotation averaging approaches (Chatterjee & Govindu, 2017; Aftab et al., 2014; Hartley et al., 2011), in this work, we adhere to graph neural network-based approaches for robust camera pose estimation via a learned view-graph module, which helps in better removal of the erroneous poses nodes in the graph (Yang et al., 2021; Gilmer et al., 2017; Purkait et al., 2020; Li & Ling, 2021).

Note that part of our work was published as a conference proceeding at the British Machine Vision Conference (BMVC), 2022 (Jain et al., 2022). Nevertheless, this journal version is a substantial extension of the conference paper both in terms of formulation, experimentation and ablation.

3 Problem Statement and Our Approach

Given a set of multi-view images captured from a freely moving handheld camera, the goal is to recover accurate camera pose and learn a better neural scene representation for novel view synthesis. In our problem setting, we predict single image depth prediction (SIDP) prior per frame using off-the-shelf pre-trained model (Ranftl et al., 2021).

As discussed, a freely moving camera could lead to scene observation at different pixel resolutions, and therefore, we propose to utilize the mipmapping approach to model the scene representation (Barron et al., 2021). For joint optimization of camera pose with the scene representation parameters, we compose our proposed pipeline with graph-neural network-based robust motion averaging, where the initial pose could be initialized *randomly or via off-the-shelf algorithms*. Still, we are plagued by radiance-geometry ambiguity, so we introduce SIDP per frame to resolve it. Another advantage SIDP brings is that we can use relative camera pose prior per frame to further improve the camera motion estimates and respective scene 3D parameters.

Based on the above discussion, we propose three algorithmic variations of our proposed idea, which is based on the following variations in the experimental initial setup (i) The basic version, dubbed as **RM-NeRF** takes a noisy set of poses estimated using COLMAP (Schönberger et al., 2016) with multi-scale MV images as input. It assumes no depth prior per frame while the intrinsic camera matrix is known. (ii) Similar to the first version, we assume the intrinsic camera matrix as well as per view depth prior is known; however, the camera pose is randomly initialized. We call this version of our algorithm as **RM-NeRF (w/o pose)**. (iii) Assuming SIDP per view and randomly initialized camera pose, the third variation **RM-NeRF (E2E)** estimates camera intrinsic, camera extrinsic, and scene representation from multi-view images. Fig. 2 provides the complete pipeline of our algorithm. Depending on our assumptions about the known priors, we utilize the different modules shown in the diagram to optimize the proposed overall loss function.

Next, we discuss the technical details pertaining to our approach pipeline. We begin with a discussion on multiscale representation for NeRF followed by multiple motion averaging. These two concepts form the basis of our methodology i.e., **RM-NeRF**.

(a) Multiscale Representation for NeRF. By leveraging pre-filtering techniques in rendering (Amanatides, 1984) i.e., tracing a cone instead of ray, Mip-NeRF (Barron et al., 2021) learns the scene representation by training a single neural network, which can be queried at arbitrary scales. Furthermore, contrary to NeRF, which uses point-based sampling along each pixel ray to form their positional encoding (PE) feature vector, Mip-NeRF uses the volume of each conical frustum along the cone to model the integrated positional encoding (IPE) features. The positional encoding $\gamma(\mathbf{x})$ (as defined in NeRF (Mildenhall et al., 2021)) of all the point within the conical frustum, having center at $\mathbf{o}$ and axis in the direction

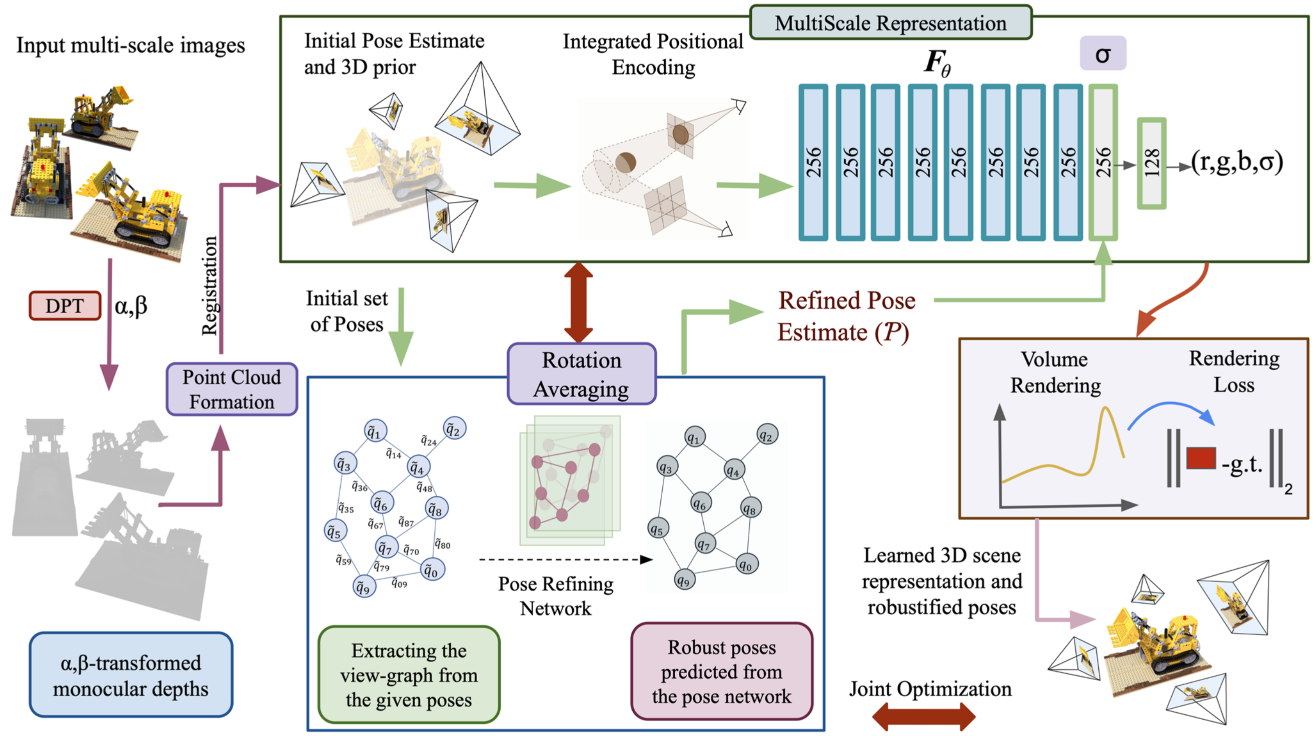

Input multi-scale images
Initial Pose Estimate and 3D prior
Integrated Positional Encoding
MultiScale Representation
F_θ
σ
256
256
256
256
256
256
256
256
256
128
(r,g,b,σ)
DPT
α,β
Registration
Point Cloud Formation
Initial set of Poses
Rotation Averaging
Refined Pose Estimate ($\mathcal{P}$)
Volume Rendering
Rendering Loss
-g.t.
2
$\tilde{q}_1$ $\tilde{q}_2$ $\tilde{q}_3$ $\tilde{q}_4$ $\tilde{q}_5$ $\tilde{q}_6$ $\tilde{q}_7$ $\tilde{q}_8$ $\tilde{q}_9$ $\tilde{q}_0$
$\tilde{q}_{14}$ $\tilde{q}_{24}$ $\tilde{q}_{36}$ $\tilde{q}_{48}$ $\tilde{q}_{35}$ $\tilde{q}_{67}$ $\tilde{q}_{87}$ $\tilde{q}_{80}$ $\tilde{q}_{59}$ $\tilde{q}_{70}$ $\tilde{q}_{79}$ $\tilde{q}_{09}$
Pose Refining Network
q_1 q_2 q_3 q_4 q_5 q_6 q_7 q_8 q_9 q_0
Extracting the view-graph from the given poses
Robust poses predicted from the pose network
α,β-transformed monocular depths
Learned 3D scene representation and robustified poses
Joint Optimization

Fig. 2 We propose to jointly solve camera poses and learn the multi-scale scene representation. The pipeline consists of a camera pose refining network, single image depth prior to estimate camera pose and the IPE (Integrated Positional Encoding) by casting well-posed conical frustums through the pixels. Later, those are fed to the MLP network for learning the scene representation for novel view synthesis. $\mathcal{P}$ denotes set of camera pose. Here, DPT symbolizes single image depth prediction model from Ranftl et al. (2021) work

d, is formulated as

$$\gamma^*(\mathbf{o},\mathbf{d},\dot{r},t_0,t_1)=\frac{\int\gamma(\mathbf{x})\mathbf{F}(\mathbf{x},\mathbf{o},\mathbf{d},\dot{r},t_0,t_1)d\mathbf{x}}{\int\mathbf{F}(\mathbf{x},\mathbf{o},\mathbf{d},\dot{r},t_0,t_1)d\mathbf{x}}, \qquad (1)$$

where **F** is an indicator function regarding whether a point lies inside the frustum in the given range $[t_0, t_1]$ and $\dot{r}$ is the ray corresponding to the axis. Since Eq. (1) is computationally intractable with no closed form solution, it is approximated using multivariate Gaussian which provides “integrated positional encoding” (IPE) feature, proposed in Barron et al. (2021).[1]

(b) Scene Rigidity and Multiple Motion Averaging. Assume a pin-hole camera model with intrinsic calibration matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ and extrinsic calibration $\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^{3\times1}$ as the rotation matrix and translation vector, respectively w.r.t assumed reference. We can relate i^{th} image pixel $x = [u_i, v_i, 1]^T$ to its corresponding 3D point $\mathbf{x} = [x_i, y_i, z_i]^T$ using the following popular projective geometry relation, i.e.,

$$s[u_i, v_i, 1]^T = \mathbf{K}\big[\mathbf{R} \mid \mathbf{t}\big]\big[x_i, y_i, z_i, 1\big]^T \qquad (2)$$

where s is the constant scale factor. Equation (2) indicate a non-linear interaction between 3D scene point and camera motion. Yet, the classical epipolar geometry model suggests that if the scene is rigid $x'^T\mathbf{E}\,x = 0$ must hold (Hartley & Zisserman, 2003), where x' is the image correspondence of x in the next image frame. It is well-studied that **E** can be decomposed into **R** and **t** such that $\mathbf{E} = [\mathbf{t}]_\times\mathbf{R}$, where $\mathbf{E} \in \mathbb{R}^{3\times3}$ is the essential matrix and $[\mathbf{t}]_\times \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix representation of the translation vector. Using this epipolar relation, we can estimate rigid camera motion without making use of any actual 3D observation. Nonetheless, rigid motion solution based on epipolar algebraic relation is not robust to outliers and may provide unreliable results with more multi-view images (Chatterjee & Govindu, 2017). So to estimate robust camera motion independent of 3D scene point in a computationally efficient way led to the success of robust motion averaging approaches in geometric computer vision (Govindu, 2006; Aftab et al., 2014; Chatterjee & Govindu, 2017). Moreover, given rotations, solving translations generally becomes a linear problem (Chatterjee & Govindu, 2017). Consequently, solution to motion averaging reduces to rotation averaging problem.

[1] For more details and derivations, refer Barron et al. (2021) work.

3.1 RM-NeRF: Formulation and Optimization

Let $\mathcal{I}$ be the set of multi-view images taken at different distances from the object (see top left: Fig. 2). RM-NeRF aims at simultaneously updating the MLP parameterized multi-scale representation network (θ) and set of camera poses $\mathcal{P}$, given estimated noisy poses $\tilde{\mathcal{P}}$ and camera intrinsics $\mathbf{K}$. Assuming the favorable distribution model $\Phi()$, we can write the overall goal of RM-NeRF as

$$\theta, \mathcal{P} \sim \Phi(\theta, \mathcal{P}|\mathcal{I}, \tilde{\mathcal{P}}, \mathbf{K}). \tag{3}$$

The above formulation can further be simplified based on rigid scene assumption. As a result, we can optimize for the camera pose without explicit knowledge of 3d points in the scene space. Accordingly, we simplify the Eq. (3) as

$$\Phi(\theta, \mathcal{P}|\mathcal{I}, \tilde{\mathcal{P}}, \mathbf{K}) = \overbrace{\Phi(\theta|\mathcal{I}, \mathcal{P}, \mathbf{K})}^{\text{Multiscale MLP}} \cdot \overbrace{\Phi(\mathcal{P}|\mathcal{I}, \tilde{\mathcal{P}}, \mathbf{K})}^{\text{Motion averaging}}. \tag{4}$$

Equation (4) allows a separate modeling scheme for camera pose recovery and the 3D scene representation. Next, we describe our motion averaging approach for camera motion estimation, followed by its modification to recover robust camera pose estimates leading to RM-NeRF joint optimization.

3.1.1 Graph Neural Networks for MRA

Assume a directed view-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (see Fig. 2 center bottom). A vertex $\mathcal{V}_j \in \mathcal{V}$ in this view graph corresponds to j^{th} camera absolute rotation R_j and $\mathcal{E}_{ij} \in \mathcal{E}$ corresponds to the relative orientation $\tilde{R}_{ij}$ between view i and j (in Fig. 2 represented as quaternions). Here, we assume noisy relative camera motion for view graph initialization. We aim to recover accurate absolute pose R_j and jointly model the object representation. Conventionally, in the presence of noise, the camera motion is obtained by solving the following optimization problem to satisfy well-known compatibility criteria for rotation group (Hartley et al., 2013), i.e.,

$$\underset{\{R_j\}}{\operatorname{argmin}} \sum_{\mathcal{E}_{ij} \in \mathcal{E}} \rho\Big(d(\tilde{R}_{ij}, R_j R_i^{-1})\Big), \tag{5}$$

where, $d(.)$ denotes a suitable metric on $SO(3)$ and $\rho(.)$ is the robust loss function defined over that metric. Minimizing this cost function $\rho(.)$ in Eq. (5) using conventional method may not be apt for several types of noise distribution observed in the real-world multi-view images. Therefore, we adhere to learn the noise distribution from the input data at train time and infer the noisy pattern to robustly predict absolute rotation. We pre-train graph neural network in a supervised setting to learn the mapping f that takes noisy relative rotation $\tilde{R}_{ij}$ and predict absolute rotations i.e., $\{R_j^f\} := f(\tilde{R}_{ij}; \Theta)$, where Θ is the network parameters.

We now discuss working of our camera pose network performing multiple rotation averaging (MRA) based on message passing graph neural networks (GNNs). We first discuss the working of Message Passing Networks(MPNN) involving a graph node and its neighbours.

(i) Message Passing Scheme. Given a directed view-graph $\mathcal{G}$ (see Fig. 3) with N cameras and M pairwise relative orientation, we use the message passing neural network approach to operate on it. Let $m_j^{(t)}$ be the message functions that correspond to the message from the neighboring nodes $u \in \mathcal{N}_j$. Denoting $\psi^{(t)}$ as the update functions (T layers), and $h_j^{(t-1)}$ the state of node j at time step $(t-1)$, the feature node state $h_j^{(t)}$ at time t in the graph is updated as:

$$h_j^{(t)} = \psi^{(t)}\big(h_j^{(t-1)}, m_j^{(t)}\big) \tag{6}$$

$\psi^{(t)}$ corresponds to concatenation operation followed by a 1D convolution and ReLUs. Intuitively, the node j state at time t is updated via update function $\psi^{(t)}$ based on the current message function value $m_j^{(t)}$ and state of the node j at $t-1$ (concatenation). Yet, we want to have a smooth update of the graph node value hence 1D convolution. The message function $m_j^{(t)}$ at node j due to all neighbor $\mathcal{N}_j$ is expressed as

$$m_j^{(t)} = \Omega_{\mathcal{V}_i \in \mathcal{N}_j} h_{i \to j}^{(t)} \tag{7}$$

Here, $\Omega(.)$ denotes a differentiable function like the softmax activation function, $h_{i \to j}^{(t)} := \Psi^{(t)}(h_j^{(t-1)}, h_i^{(t-1)}, e_{ij})$ is the accumulated message for the edge $\mathcal{E}_{ij}$ at t. $\Psi^{(t)}$ is concatenation operations followed by 1D convolution and ReLU activation. In our setup, $\mathcal{N}_j$ is the set of all neighboring cameras connected to $\mathcal{V}_j$ and e_{ij} is the edge feature of the edge $\mathcal{E}_{ij}$. For more details on the messaging passing algorithmic details refer Gilmer et al. (2017); Purkait et al. (2020)

(ii) Robustifying Poses using GNN. The GNNs pipeline for estimating robust pose consists of three major steps:

(1) Cleaning the view-graph. We first estimate the relative rotations $\tilde{R}_{ij}$ from the noisy rotations R_i due to input data. Next, we apply cycle consistency check to remove the outliers (Aftab et al., 2014; Hartley et al., 2013). Local cyclic graph structure of the view-graph must provide orientation close to identity. Violation of such a local constraint helps in removal of bad camera pose estimates. In this work, we have proposed three approaches,.i.e, RM-NeRF, RM-NeRF (w/o) pose and RM-NeRF (E2E). For RM-NeRF, we initialized the camera poses in the view-graph using COLMAP (Schönberger et al., 2016). For the rest of our approaches, we initialized the view-graph with random camera poses.

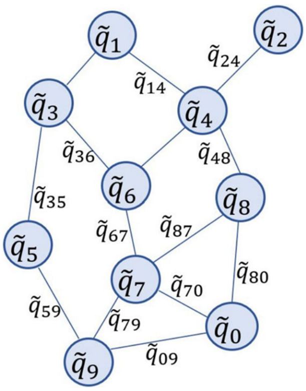


Fig. 3 An example view graph extracted form the input poses. The vertex set $\mathcal{V}_j$ of this viewgraph correspond to initial absolute orientation of each of the images and the edge set $\mathcal{E}_{ij}$ correspond to the relative orientations between the image pairs

(2) Computing noisy initial solution using the extracted relative rotations. For this, we build a minimum spanning tree (MST) using all the nodes in the view-graph by fixing the root node to be the node with maximum neighbours (greatest fan-out). Then, we generate an initial absolute rotation $\hat{R}_i$ for each node by solving for the motion variables from the root pose value to other nodes along the tree structure.

(3) Refining the initial solution using Graph Neural Networks. For applying GNNs, we require features corresponding to each node in a view graph. We use the rotation matrix in the initial solution corresponding to every node in the graph as its input feature. Furthermore, we also pass the observed relative rotations $\tilde{R}_{ij}$ as edge features to the GNN following the formulation described in the previous paragraph. Furthermore, instead of directly passing these relative rotations as edge features, we instead pass the discrepancy between these observed relative rotations and the initial solution resulting in the the edge feature $e_{uv} = \hat{R}_v^{-1}\tilde{R}_{uv}\hat{R}_u$, to the GNN.

The resultant input view graph then becomes $\mathcal{G} = \{\hat{R}_i, e_{ij}\}$ leading to a supervised learning problem $R_j^f := f(\{\hat{R}_i, e_{ij}\}; \Theta)$, which is trained using the rotation averaging loss function. Moreover, we know that relative rotation between any 2 nodes in the viewgraph is invariant to any constant angular deviation in form of rotation matrix R to both the nodes and thus, both the solution sets $\{R_i R, R_j R\}$ and $\{R_i, R_j\}$ result in the same discrepancy when using the rotation averaging loss function. To handle this issue involving an unknown global ambiguity in the rotations, we add a regularizer in our objective function to handle such a discrepancy between the absolute rotations. This results in the following objective function

$$\mathcal{L} = \sum_{\mathcal{G}\in\mathcal{D}} \sum_{\mathcal{E}_{ij}\in\mathcal{E}} d_Q(q_{ij}^f, q_{ij}) + \beta \sum_{\mathcal{V}_j\in\mathcal{V}} d_Q(q_j^f, q_j), \tag{8}$$

where $d_Q = \min\{\|p-q\|_2, \|p+q\|_2\}$ measures distance between two quaternion p, q. Thus, our overall **RM-NeRF** loss solves for accurate camera poses and scene representation jointly. Concretely, we combine Eq. (8) ($\mathcal{L}_{mra}$) with the squared error between the true $C(\mathbf{r})$ and predicted $\hat{C}(\mathbf{r})$ pixel colors ($\mathcal{L}_{rgb}$) to define the overall **RM-NeRF** loss $\mathcal{L}$ as

$$\overbrace{\sum_{\mathcal{E}_{ij}\in\mathcal{E}} d_Q(q_{ij}^f, q_{ij}) + \beta \sum_{\mathcal{V}_j\in\mathcal{V}} d_Q(q_j^f, q_j)}^{\mathcal{L}_{mra}} + \overbrace{\sum_{\mathbf{r}\in\mathcal{R}} \|C(\mathbf{r}) - \hat{C}(\mathbf{r})\|_2^2}^{\mathcal{L}_{rgb}}, \tag{9}$$

Here, β is a scalar constant. q_{ij}'s symbolizes corresponding quaternion representation of the rotation matrix defined in Eq. (8). $\mathcal{V}$ denotes the vertex set of the view graph corresponding to the scene being optimized and $\mathcal{E}$ denotes the corresponding edge set.

3.1.2 RM-NeRF Joint Optimization

Let's denote MLP parameters in rendering network as θ and camera pose network parameters as Θ. Our objective is to optimize for the parameters θ and Θ jointly such that Eq. (9) loss is as minimum as possible. Using gradient based optimization for this search process requires calculating $\nabla_\theta\mathcal{L}$ and $\nabla_\Theta\mathcal{L}$. As $\mathcal{L}_{mra}$ is independent of rendering network, we have $\nabla_\theta\mathcal{L} = \nabla_\theta\mathcal{L}_{rgb}$. This appears to be similar as previous optimization landscape for the rendering network, but here the poses would be changing continuously resulting in different numeric value of the gradient, making the optimization difficult to converge. Now, for the pose network $\nabla_\Theta\mathcal{L}$ will have 2 terms: $\nabla_\Theta\mathcal{L}_{rgb}$ and $\nabla_\Theta\mathcal{L}_{mra}$. The second term is easy to handle given the pose network is able to solve the rotations as shown in Purkait et al. (2020). The first term is something that would entangle the search process for θ and Θ. For clarity, let's assume the loss due to predicted color as $\Phi(\theta, \gamma(\mathcal{P}))$, where $\mathcal{P}$ (with slight abuse of notation) denotes the poses having rotations predicted by the pose network, γ denotes the positional encoding (Rahaman et al., 2019), then the gradient of $\Phi(\theta, \gamma(\mathcal{P}))$ w.r.t the pose network parameters Θ can be computed using backpropagation as:

$$\nabla_\Theta\mathcal{L}_{rgb} = \frac{\partial\Phi(\theta, \gamma(\mathcal{P}))}{\partial\Theta} = \frac{\partial\Phi(\theta, \gamma(\mathcal{P}))}{\partial\gamma(\mathcal{P})}\frac{\partial\gamma(\mathcal{P})}{\partial\mathcal{P}}\frac{\partial\mathcal{P}}{\partial\Theta} \tag{10}$$

Differentiating this γ function might result in updates being favourable to higher frequencies (k) as pointed out previously in Lin et al. (2021). Accordingly, we modify γ function further to

$$\gamma^*(x, k) = e^{g(k)}\gamma(x), \tag{11}$$

where $g(k) = \min(\frac{t-k}{b}, 0)$, t is annealed from 0 to maximum number of modes and b is a scalar constant. The term $\nabla_{\Theta}\mathcal{L}_{rgb}$ shown in Eq. (10) results in correlated updates on MLP network and pose network parameters and can result in a highly non-convex optimization. To make optimization stable, we use the following weighted loss function:

$$\mathcal{L} = \lambda\mathcal{L}_{mra} + (1-\lambda)\mathcal{L}_{rgb} \tag{12}$$

where λ is a scalar constant.

3.2 RM-NeRF (w/o Pose): Random Initial Camera Poses

Our approach RM-NeRF as discussed in the previous subsection works well in practice, given that the initial camera pose estimates are not random. The intuition behind this is that pose-refining based on MRA is used in scenarios where the amount of noise in the initial estimated poses is distributed across the global pose graph, hence we can recover a good overall solution. Yet, RM-NeRF formulation could fail if the pose graph initialization is reasonably erroneous. This puts a hard constraint of providing reasonable camera pose initialization to perform MRA.

To overcome such a practical limitation, we propose an extension to the introduced **RM-NeRF** formulation. The idea is if we have prior knowledge about the scene's geometry, we could constrain the camera motion per view at the same time and resolve radiance-geometry ambiguity in neural image rendering. To this end, we assume depth per frame is known and predicted using Ranftl et al. (2021) network.[2] Given the depth prediction per view, we constrain scene point alignment via relative camera pose. Yet, solving for camera pose using such a constraint could lead to sub-optimal solution and requires further pose refinement. Nevertheless, it fits our purpose of using the MRA. Accordingly, we use our pose-refining GNN (same as RM-NeRF) in an iterative manner.

RM-NeRF (w/o pose) extension is based on the intuition that refining the camera poses estimated using 3d scene point alignment loss per view at each step via GNN can help avoid sub-optimal solution in the optimization landscape of the overall objective. Thus, we propose a loss function that allows camera poses to be initialized randomly and still be able to recover good camera pose estimates. For each step during optimization, the current poses are updated using the point cloud alignment and then refined using our pose-refining GNN.

The goal is to model the network parameters (θ) and the correct camera pose set ($\mathcal{P}$) using the input image set ($\mathcal{I}$) and known camera intrinsics ($\mathbf{K}$) involving an intermediate monocular depth prediction per view and reasonable camera pose set $\hat{\mathcal{P}}$ estimation using monocular depth, followed by refinement. For better abstraction, we can define the proposed intuition in terms of following equation.

$$\begin{aligned}\theta, \mathcal{P} &\sim \Phi(\theta, \mathcal{P}|\mathcal{I}, \mathbf{K})\\ &= \Phi(\theta, \mathcal{P}|\mathcal{I}, \mathbf{K}, \mathcal{D})\Phi(\mathcal{D}|\mathcal{I})\\ &= \Phi(\theta, \mathcal{P}|\mathcal{I}, \mathbf{K}, \hat{\mathcal{P}})\Phi(\hat{\mathcal{P}}|\mathcal{D}, \mathbf{K})\Phi(\mathcal{D}|\mathcal{I}).\end{aligned} \tag{13}$$

Here, $\mathcal{D}$ denotes the set of predicted depth map per frame. Given that we are feeding the initial estimate $\hat{\mathcal{P}}$ into motion averaging network with parameters Θ to predict the refined pose $\mathcal{P}$, this leads to the following relation: $\mathcal{P} = f_{\Theta}(\hat{\mathcal{P}})$. This allows us to randomly initialize the camera pose set $\hat{\mathcal{P}}$. Thus, given $\mathcal{I}$, $\mathcal{D}$, and $\hat{\mathcal{P}}$, we perform an iterative optimization by minimizing the chamfer distance $\mathcal{L}_{cd}$ of the scene points between views leading to depth and camera pose refinement.

Similar to the concurrent work Nope-NeRF (Bian et al., 2022), we define two learnable parameters α_i, β_i to transform each monocular depth D_i to a global frame for multi-view consistency. Denoting transformed depth as D_i^*, we write

$$D_i^* = \alpha_i D_i + \beta_i. \tag{14}$$

Such transformation parameters is learnt by aligning the transformed and rendered depth ($\hat{D}$) via an MLP loss

$$\mathcal{L}_d = ||D^* - \hat{D}||_2. \tag{15}$$

Here, α_i, β_i are scalar parameters. The role of α is to fix the scale in the monocular depth map. Whereas β is responsible for handling the additive bias. This is because relative depth should be consistent over different camera viewpoints. A constant scale and bias factor seem sufficient to fix it for each view. Similar to Bian et al. (2022), assuming the known transformed depths for i^{th} and j^{th} image along with their relative pose $T_{ij} = T_j T_i^{-1}$, we unprojected the depth maps to scene point clouds P_i and P_j respectively. Here, $T_i, T_j \in \mathcal{P}$. The camera pose corresponding to each of these images should be such that relative pose T_{ji} aligns P_i to P_j. Thus, the Chamfer Distance ($\mathcal{L}_{cd}$) between the P_j and the transformed point cloud $T_{ji}P_i$ becomes a suitable objective function constraint for the camera poses. Using this objective as an additional loss function, we arrive at the following overall loss function $\mathcal{L}_{agg}$—across all the training images—to optimize the MLP parameters θ, transformation parameters α, β and the randomly initialized set of poses $\mathcal{P}$:

$$\mathcal{L}_{agg} = \left(\mathcal{L}_{mra} + \mathcal{L}_{rgb} + \mathcal{L}_d\right) + \sum_{i,j} \mathcal{L}_{cd}\left(T_{ji}P_i, P_j\right) \tag{16}$$

The loss proposed in Eq. (16) captures our overall notions. Yet, it is complex and challenging to optimize efficiently

[2] With the recent progress in single image depth prediction (SIDP) network, it is quite a reasonable assumption.

compared to RM-NeRF, accounting for the fact that we want to allow for random camera pose initialization. To understand this better, let us look at the gradient descent-based update term of optimization variables in Eq. (16). At any optimization step t, the rendering MLP network parameters (θ^t) are updated just w.r.t. the rendering loss $\mathcal{L}_{rgb}$ with the gradient term being $\frac{\partial \mathcal{L}_{rgb}}{\partial \theta}$ (same as RM-NeRF). On the other hand, the overall updates involved in camera pose estimation is intricate. For each step t, we first update the initial estimate of each pose $\hat{\mathcal{P}}_i^t$ using chamfer distance:

$$\hat{\mathcal{P}}_i^t = \hat{\mathcal{P}}_i^{t-1} - \alpha \sum_j \frac{\partial}{\partial \hat{\mathcal{P}}} \mathcal{L}_{cd}(T_j T_i^{-1} P_i, P_j) \tag{17}$$

Given the updated poses $\hat{\mathcal{P}}^t$ and the MLP parameters θ^t, we now update the camera pose network parameters Θ^t using $\mathcal{L}_{mra}$ and $\mathcal{L}_{rgb}$ similar to **RM-NeRF** i.e.,

$$\Theta^t = \Theta^{t-1} - \beta \frac{\partial}{\partial \Theta} \left(\mathcal{L}_{mra}(f_\Theta(\hat{\mathcal{P}}^t)) + \Phi(\theta^t, \gamma(f_\Theta(\hat{\mathcal{P}}^t))) \right) \tag{18}$$

Note, however we use disjoint set of loss functions for updating $\hat{\mathcal{P}}_i^t, \Theta^t$ and therefore, do not update $\hat{\mathcal{P}}_i^t$ using $\mathcal{L}_{rgb}$ or $\mathcal{L}_{mra}$ due to the term $\frac{\partial \Theta^t}{\hat{\mathcal{P}}_i^t}$, and deal with them only using Θ^t. Such updates in Eqs. (17) and (18) can be either simultaneously for each step or applied alternatively for some fixed number of steps. We follow the later strategy and update $\hat{\mathcal{P}}_i^t$, Θ^t alternatively, for a fixed number of steps (K).

3.3 RM-NeRF (E2E): Unknown Intrinsic Camera Matrix

Even though our **RM-NeRF (w/o pose)** method overcomes the requirement of good initialization of camera pose-graph variables, it still requires the intrinsic matrix for a given image set, which may not be available for real-world multi-view data. To handle this, we introduce the third extension of our algorithm referred as **RM-NeRF (E2E)**.

RM-NeRF (E2E) estimates both camera poses and intrinsic camera parameters from the multi-view image set. Additionally, it is able to work well with randomly initialized intrinsic matrix and camera poses, given the $\mathcal{D}$ is provided or predicted via a trained model. The overall loss is similar to **RM-NeRF (w/o poses)** except now the intrinsic matrices $\mathbf{K}$ is estimated leveraging the following relation among $\theta, \mathcal{P}, \mathbf{K}$.

$$\theta, \mathcal{P} \sim \Phi(\theta, \mathcal{P}, \mathbf{K} | \mathcal{I}, \hat{\mathcal{P}}) \Phi(\hat{\mathcal{P}} | \mathcal{D}) \Phi(\mathcal{D} | \mathcal{I}) \tag{19}$$

Note, α and β are updated only using the term $\mathcal{L}_d$ and the intrinsic ($\mathbf{K}$), extrinsic ($\mathcal{P}$) and MLP network parameters (θ) are updated using the three terms except $\mathcal{L}_d$ in Eq. (16)

$$\mathbf{K}, \mathcal{P}, \theta = \arg \min_{\mathbf{K}, \mathcal{P}, \theta} \left(\mathcal{L}_{mra} + \mathcal{L}_{rgb} + \mathcal{L}_{cd} \right) \tag{20}$$

where $\mathcal{L}_{cd}$ is shorthand for $\sum_{i,j} \mathcal{L}_{cd}(T_{ji}, P_i, P_j)$.
Updating K. The overall objective for the updating intrinsics at step t ($\mathbf{K}^t$) involves minimizing losses $\mathcal{L}_{rgb}$ and $\mathcal{L}_{cd}$:

$$\mathbf{K}^t = \mathbf{K}^{t-1} - \frac{\partial}{\partial \mathbf{K}} (\mathcal{L}_{rgb} + \mathcal{L}_{cd}) \tag{21}$$

The intrinsics are being updated alongside camera poses $\hat{\mathcal{P}}$ w.r.t. loss $\mathcal{L}_{cd}$ and pose-refining GNN parameters Θ w.r.t. loss $\mathcal{L}_{rgb}$. Such a strategy may lead to suboptimal solution due to the complex nature of optimization. Thus, we only update intrinsics $\mathbf{K}$ using the rendering loss $\mathcal{L}_{rgb}$ alongside the GNN parameters Θ.

3.4 Optimization Implementation Details

RM-NeRF. We begin with a disjoint optimization scheme for camera poses and scene representation by fixing $\lambda = 1$ for some initial number of epochs. For this case, Eq. (4) depicts the modified formulation of the problem statement. After the initial optimization of both the networks via biased weighting strategy, λ is annealed by using an exponential decay, i.e., $\lambda = \lambda_0 e^{-kt}$ where $\lambda_0 = 1$. This annealing goes till $\lambda = 0.5$ and then we fix it at 0.5 for the remaining optimization process.
RM-NeRF (w/o pose). Here, we perform initial updates using only the $\mathcal{L}_{cd}$ and $\mathcal{L}_d$ loss function for some number of epochs and then use all the loss function terms (equi-weighted) to update the variables and parameters. Our idea to train the overall model this way is a warm-up step, given that the optimization landscape can be pretty complex.
RM-NeRF (E2E). For this case, we first perform updates only using the $\mathcal{L}_{cd}$, $\mathcal{L}_d$ and $\mathcal{L}_{rgb}$ losses (equi-weighted). This leads to an initial estimate of the camera intrinsics and poses. Furthermore, we include the $\mathcal{L}_{mra}$ while performing updates in an equi-weighted fashion.

4 Experimental Setup, Results and Ablations

Our overall pipeline involves optimizing parameters for two neural networks: (a) Graph Neural Network (GNN) for robust refinement of camera poses and (b) MLP network to learn the multi-scale NeRF representation for the scene. For the GNN, we follow the architecture of FineNet, proposed by Purkait et al. (2020). For the MLP, we use the same architecture and sampling scheme as the Mip-NeRF paper (Barron et al., 2021) ("coarse" and "fine" sampling involving 128 samples each to render the color for a given pixel).

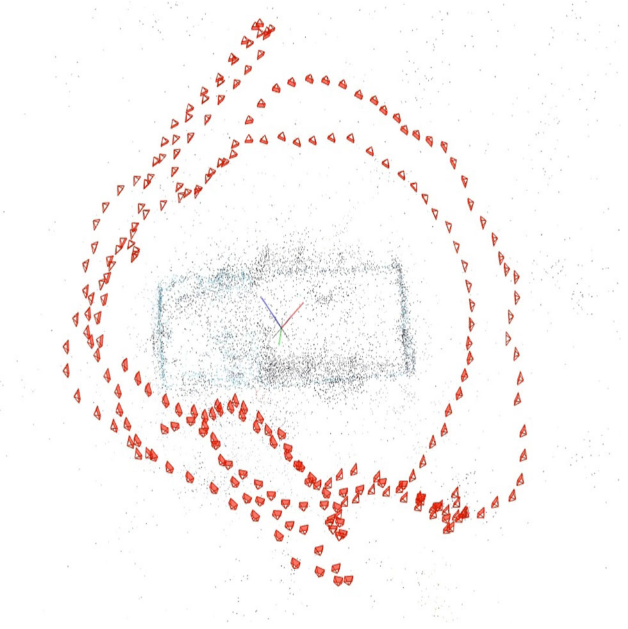


Fig. 4 Camera poses recovered by COLMAP (Schonberger & Frahm, 2016) corresponding to the Truck scene in Tanks and Temples dataset (Knapitsch et al., 2017), comprising a total of 251 images captured by gradually moving around the truck object

Furthermore, we pre-train our pose-refining GNN in a supervised setting using a dataset comprising synthetically generated view graphs, proposed by Purkait et al. (2020). This dataset contains 1200 view graphs with up to 30% outliers. Additionally, the dataset comprises noisy rotations where the noise is sampled from a Gaussian distribution with a standard deviation between 5°and 30°. We used 20% of the dataset to test and train the network on the remaining examples. Once trained, our pose refinement network achieves a mean angular error (MAE) of 2.09° and a median angular error of 1.1° on the test set. We do this pre-training for 250 epochs using a learning rate of 5×10^{-5} and a weight decay of 10^{-4}. Also, we drop one-fourth of the edges in the view graph to minimize overfitting at train time.

The trained pose-refinement network is now updated alongside the scene representation MLP network for each scene. Given refined rotations, a linear optimizer is employed to solve translation (Chatterjee & Govindu, 2017). We set the value of hyperparameter β (used in Eq. 8) to 0.1 during both the pretraining and pose refinement stages. Similarly, we fix the value of hyperparameter b from Eq. (11) to be 10 for all the experiments. The training of our proposed involves optimizing the MLP network for 100k iterations per scene using the Adam Optimizer (Kingma & Ba, 2014). We use a batch size of 4096 and a logarithmically varying learning rate (between 5×10^{-4} and 5×10^{-5}).

All of our experiments have been carried out on a 32 GB Nvidia V100 GPU computing machine.

4.1 Test Sets and Results

We evaluate our proposed method under two settings:
(i) This setting involves analysis of the introduced approach on a synthetic dataset. This dataset is generated by rendering images of 3D object using a pre-defined and well structured camera poses covering 360° view of the object. Thus, ground-truth poses are well defined. For this experiments, we use the Blender dataset provided by Mildenhall et al. (2021)[3] and its multi-scaled version Barron et al. (2021), consisting of single object scene centered around a single object without any background. Each scene in this dataset comprises of 100 images with 800 × 800 resolution with is captured by moving the camera along a fixed hemisphere surrounding this object. We simulate realistic scenario for this dataset by (a) using the multi-scaled version representing varying distance of camera from the object and (b) adding noise to the ground truth poses.
(ii) This setting tests our method on the real-world images which are acquired by a freely moving camera with no access to ground-truth camera parameters. The camera parameters have to be estimated. For this, we use the Tanks and Temples dataset (Knapitsch et al., 2017), comprising various scenes from indoor and outdoor real-world settings. Other than this, we test our approach on the popular ScanNet dataset (Dai et al., 2017) comprising a diverse set of indoor scenes. Additionally, we show results on two other datasets. This first one captures a box using a regular phone with arbitrary camera trajectory, and the second is a scene taken from a recent work by Yen-Chen et al. (2022a).

More details regarding each dataset are provided in the following subsections.

4.1.1 Multi-scaled Images of Object

For evaluating our method in an object-centric synthetic setting comprising a complete 360° view, we study the multi-scaled version of the NeRF Blender dataset, proposed by Barron et al. (2021). It is generated by resizing each image in the NeRF Blender dataset to three different,resolutions and concatenating them with the original dataset. Resizing these images does not change the ground-truth poses, however the intrinsics for images at each resolution are updated accordingly. This resizing can be interpreted as changing the camera distance from the center of the object and thus, this dataset is more closely aligned with real-world setting as compared to Blender dataset (Mildenhall et al., 2021).

We further perturb the ground-truth camera poses to make this dataset close to the real-world setting. Given all the cameras in ground-truth poses lie on a hemisphere, we limit ourselves to only perturbing the rotational parameters. For this, we first sample Gaussian noise $\delta \mathbf{p} \sim \mathcal{N}(\mathbf{0}, 1e^{-1}\mathbf{I})$ in

[3] CC-BY−3.0 license.

Table 1 Comparison of our proposed methods (RM-NeRF and RM-NeRF (w/o pose)) with the baselines, on the Multi-scale Blender dataset using synthetically perturbed poses

	Lego		Ship		Drums		Mic		Chair		Ficus		Materials		Hotdog	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
Mip-NeRF	21.52	0.06	24.54	0.07	13.34	0.075	24.71	0.05	29.1	0.049	22.47	0.055	19.7	0.089	27.09	0.053
BARF	10.88	0.55	8.81	0.74	11.56	0.76	12.35	0.57	14.35	0.47	11.88	0.65	12.28	0.61	14.28	0.46
Base A	11.67	0.49	14.28	0.28	13.25	0.67	12.28	0.41	15.12	0.20	12.31	0.25	13.31	0.42	16.17	0.39
Base B	12.46	0.37	13.43	0.31	11.32	0.58	14.26	0.29	13.71	0.42	11.56	0.52	12.22	0.47	15.87	0.42
NeRF–	16.89	0.094	19.89	0.12	15.67	0.074	18.35	0.08	20.22	0.098	14.44	0.13	15.77	0.22	18.69	0.20
Base C	18.28	0.089	16.32	0.22	17.25	0.070	19.42	0.073	18.67	0.114	16.32	0.12	16.58	0.207	17.55	0.223
Point-NeRF	20.12	0.079	22.47	0.10	18.32	0.066	22.37	0.061	26.67	0.077	20.23	0.107	17.23	0.112	24.45	0.092
Ours																
RM-NeRF	27.01	0.044	26.59	0.067	26.07	0.043	32.8	0.008	35.23	0.031	29.28	0.032	24.8	0.061	32.5	0.028
RM-NeRF (w/o pose)	26.34	0.048	26.02	0.081	25.23	0.051	32.1	0.009	34.67	0.042	29.04	0.044	23.2	0.072	31.7	0.030
With same initialization as RM-NeRF																
RM-NeRF (w/o pose)	**27.91**	**0.033**	**27.23**	**0.056**	**26.37**	**0.046**	**32.9**	**0.008**	**35.98**	**0.029**	**30.12**	**0.031**	**25.8**	**0.052**	**32.8**	**0.021**

Our method shows significantly better performance as compared to the existing BARF (Lin et al., 2021), Mip-NeRF (Barron et al., 2021) NeRF–(Wang et al., 2021b) baselines and the newly defined Base A, Base B, Base C baselines. Furthermore, these results reflect that our method can efficiently handle multi-scale issues and pose errors simultaneously

The bold values indicate that the best results in the corresponding category

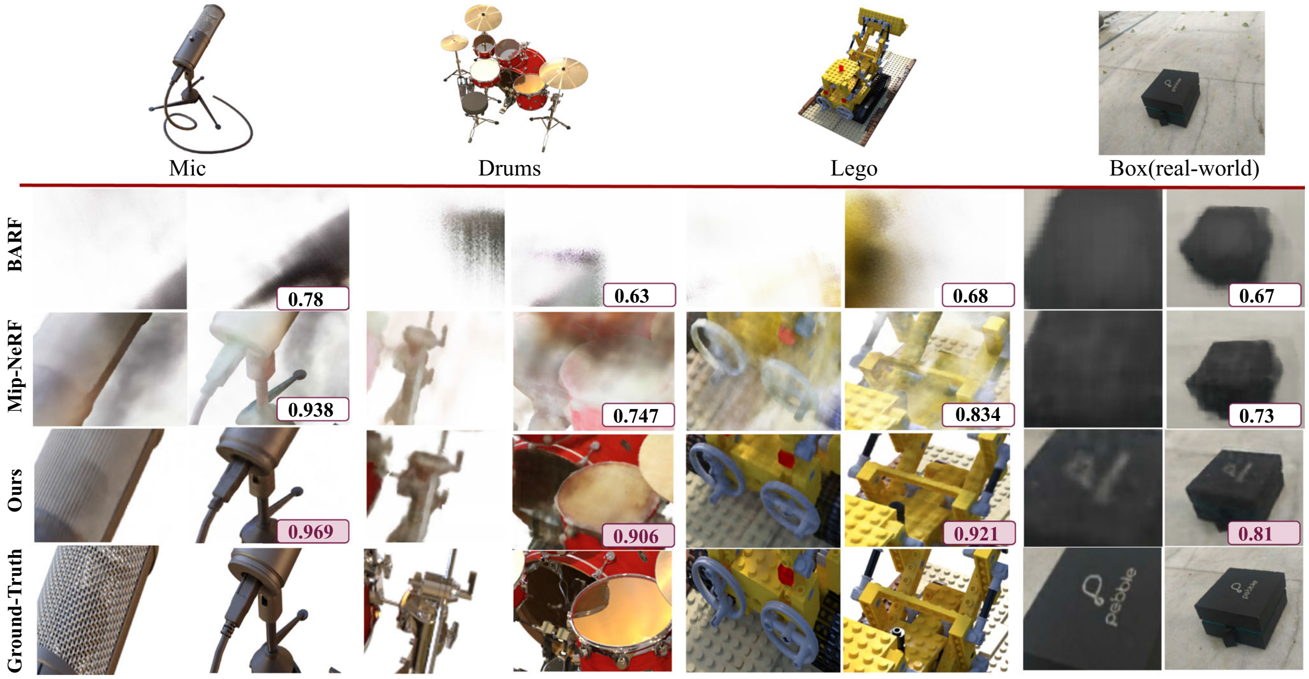


Fig. 5 Qualitative Comparison of our method, and the baselines Mip-NeRF (Barron et al., 2021), BARF (Lin et al., 2021) on the multi-scaled version of the NeRF Blender dataset (Mildenhall et al., 2021), with noise added to camera poses, and our black-box sequence. We have used three scenes (Lego, Drums and Mic) from the Multi-Scale Blender dataset. Our method is able to localize and reconstruct the object much accurately as compared to the baselines for all the scenes including the black-box sequence

the axis angle space, convert it into rotation matrix representation and multiply it with the rotations of the ground truth poses, hence perturbing ground-truth camera orientation.

Baselines. We compare the proposed RM-NeRF, RM-NeRF (w/o pose) methods with the Mip-NeRF, NeRF–(Wang et al., 2021b), BARF (Lin et al., 2021) baselines to highlight their effectiveness in dealing with camera pose errors and multi-scale images simultaneously. To further demonstrate the challenges with this dataset setting, we define three new baselines and evaluate them against our methods. These baselines are designed by combining existing baselines that can deal with pose errors and multi-scale issues separately. This leads to the first baseline (**Base A**) being a result of directly combining Mip-NeRF and BARF by replacing the positional encoding function to create the Mip-NeRF Integrated Positional Encoding with the pose encoding function used by BARF. Both its components, Mip-NeRF and BARF, when used separately, can either solve multi-scale issues (Mip-NeRF) or pose errors (BARF). On similar lines, we define a second baseline (**Base B**) which involves feeding BARF estimated poses to the Mip-NeRF multi-scale modeling scheme. Finally, we define the third baseline (**Base C**), which combines Mip-NeRF with NeRF– by replacing the positional encoding scheme in NeRF– with the Integrated Positional encoding that uses frustum-based volumetric modeling for each pixel, instead of rays. Other than these baselines, we compare our proposed methods against the recent Point-NeRF (Xu et al., 2022) method, which has shown to be quite efficient in convergence leading to high-quality renderings on the original Blender dataset (Mildenhall et al., 2021).

Results. Table 1 provides the quantitative comparison results on this dataset for RM-NeRF, RM-NeRF (w/o pose), and the baseline methods. We reported the results using the popular PSNR and LPIPS metrics averaged across all resolutions. For RM-NeRF (w/o pose), we initialized the orientation angles randomly and then converted them to rotation matrices. It can be observed that the baselines provides inferior view synthesis results for this setup. The inferior results of baselines Base A, Base B, Base C as well Mip-NeRF, BARF and NeRF– shows that naively combining the multi-scale representation with existing pose refining methods is not an apt solution. Hence, showing the importance of our proposed RM-NeRF, RM-NeRF (w/o) appraoches. Note RM-NeRF (w/o pose) shows commendable results from randomly initialized rotations.

Figure 5 provide the qualitative result comparison for the same. Both our approaches are able to achieve good quality renderings in this setup compared to the baselines.

Inducing both rotational and translation errors. Here, we analyze a more challenging setting where we introduce translation and rotation errors to the camera poses on the multi-scale blender dataset. A normal distribution with a

Table 2 Comparison of our proposed methods (RM-NeRF and RM-NeRF (w/o pose)) with the baselines, on the Multi-scale Blender dataset using synthetically perturbed poses, this time adding noise to both rotational and translational parameters

	Lego		Ship		Drums		Mic		Chair		Ficus		Materials		Hotdog	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
Mip-NeRF	19.37	0.09	22.13	0.10	12.28	0.089	22.99	0.09	28.2	0.066	21.36	0.065	17.9	0.095	26.23	0.073
BARF	9.98	0.65	8.831	0.84	10.34	0.91	11.23	0.77	13.12	0.71	10.67	0.76	11.88	0.77	12.98	0.56
NeRF–	16.89	0.094	19.89	0.12	15.67	0.074	18.35	0.08	20.22	0.098	14.44	0.13	15.77	0.22	18.69	0.20
Ours																
RM-NeRF	26.10	0.071	26.12	0.081	25.24	0.083	31.6	0.012	34.11	0.053	28.43	0.042	24.1	0.073	31.3	0.052
RM-NeRF (w/o pose)	26.34	0.048	26.02	0.081	25.23	0.051	32.1	0.009	34.67	0.042	29.04	0.044	23.2	0.072	31.7	0.030
With Same Initialization of RM-NeRF (w/o pose) as RM-NeRF																
RM-NeRF (w/o pose)	**27.23**	**0.039**	**26.95**	**0.068**	**26.06**	**0.051**	**32.3**	**0.010**	**34.88**	**0.034**	**29.34**	**0.043**	**25.1**	**0.061**	**32.1**	**0.033**

Again, our methods shows significantly better performance as compared the other methods thereby verifying their effectiveness in simultaneous presence of rotational, translational errors and multi-scale issues

The bold values indicate that the best results in the corresponding category

Table 3 Increased noise

	Lego		Ship		Drums		Mic		Chair		Ficus		Materials		Hotdog	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
RM-NeRF	23.21	0.093	22.24	0.12	23.22	0.091	29.26	0.041	30.34	0.065	25.21	0.088	20.2	0.14	27.65	0.11
RM-NeRF (w/o pose)	**26.34**	**0.048**	**26.02**	**0.081**	**25.23**	**0.051**	**32.1**	**0.009**	**34.67**	**0.042**	**29.04**	**0.044**	**23.2**	**0.072**	31.7	**0.030**

Comparison of our proposed method RM-NeRF (w/o pose) with the previously existing RM-NeRF on the Mulit-scale Blender dataset, but with increased noise as compared to Table 1. Here, the basic RM-NeRF algorithm struggles to perform whereas the RM-NeRF has quite decent performance

The bold values indicate that the best results in the corresponding category

standard deviation of 0.34 is used to sample the noisy translation, keeping the strategy for introducing rotational error the same as before. We keep our approach the same, i.e., use rotation averaging to refine the camera orientation solution and then use these refined solutions to estimate the translations. The results for this setup in Table 2 show that our method supersedes the other methods showing its effectiveness in a more realistic scenario.

The need for RM-NeRF (w/o pose). We now analyze the effect of further increasing the perturbations to the Multi-Scale Blender dataset G.T. poses. We again sample the noise from a gaussian but this time the std is doubled. The results for RM-NeRF and RM-NeRF (w/o pose) are provided in table X for this setup. It can be observed that upon increasing noise, RM-NeRF might results in poor rendering and thus, for a very random scene trajectory, RM-NeRF (w/o pose) should be the chosen option. Also, we have added another row in Tables 1 and 2 where we use the initialization for RM-NeRF (w/o pose) to be same as RM-NeRF (noise added to G.T. poses). In this case, RM-NeRF (w/o pose) is able to outperform the RM-NeRF algorithm further proving its usefulness.

4.1.2 Tank and Temples

This dataset, proposed by Knapitsch et al. (2017), comprises of challenging large scale real-world scenes. This dataset is widely used for bench-marking the 3D reconstruction algorithm. However, lately, this dataset has been popularly used in evaluating Novel View Synthesis methods designed to learn large scale scene representations with unconstrained camera pose trajectory. Accordingly, we used some sequences of this dataset to evaluate our proposed methods against the popular baselines. Specifically, we have used 8 sequences from this dataset namely ‘Truck’, ‘M60’, ‘Train’, ‘Family’, ‘Ignatius’, ‘Horse’, ‘Museum’ and ‘Francis’, comprising both indoor and outdoor sequences with significant camera motion. For example, the Truck sequence consists of image set containing a 360° view of the subject captured freely at a varying distance from the object. Since there are no ground-truth poses available, COLMAP is used to estimate the initial poses and intrinsic camera matrix (see Fig. 4 for the COLMAP result on Truck sequence). *Note that our methods RM-NeRF (w/o pose) and RM-NeRF (E2E) are initialized with completely random camera poses for all the experiments performed on this dataset*.

Baselines. We compared view synthesis results of our approaches with the following baseline methods on this dataset: *(a)* Mip-NeRF Barron et al. (2021), Point-NeRF (Xu et al., 2022) that uses COLMAP poses. *(b)* NoPe-NeRF (Bian et al., 2022) that uses randomly initialized camera poses. *(c)* NeRF–(Wang et al., 2021b) that do not use either intrinsic or extrinsic camera parameters.

Results. Table 4 shows the quantitative comparison results using the PSNR and LPIPS metrics. It shows results under three different scenarios, the top set of results corresponds to the default scenario where COLMAP poses are used as input, the middle set of results corresponds to the scenario where poses are initialized randomly, and the bottom set of results corresponds to the scenario where both camera intrinsics and extrinsics are unknown. For each setup, we use a different set of methods (baselines and our proposed method) for evaluation based on the input requirements. From the top setup, it can be observed that our RM-NeRF improves the camera pose accuracy once initialized using COLMAP camera pose results and provide improved image renderings when compared with the baselines. For the middle setup, it can be inferred that our approach RM-NeRF (w/o pose) surpasses the baselines, starting from random poses, thereby proving to be effective and robust in camera pose estimation than COLMAP and the Nope-NeRF (Bian et al., 2022) on this dataset. Finally, in the bottom setup, our method RM-NeRF (E2E) outperforms NeRF– and provides results comparable to our RM-NeRF (w/o pose) approach and baselines with random camera intrinsic parameters. We further show qualitative results for our RM-NeRF approach along with the Mip-NeRF (using COLMAP poses) and NoPe-NeRF (two-stage training (Bian et al., 2022)) in Fig. 6 for better insights. Clearly, our approach shows better image rendering results compared to the baselines.

The results obtained on this dataset demonstrate the potential of our approach in enhancing the view-synthesis framework to real-world scenes. Our joint optimization for modeling scene representation and camera pose estimation could provide favorable results for real-world scenes, including scenarios where no extra information other than the image set is given. Whereas relying on only COLMAP poses (Schonberger & Frahm, 2016) with existing neural rendering approach may provide good results on a synthetic scene or a well-controlled setup. Yet, for a general real-world video sequence, such methods could lead to erroneous camera pose estimates leading to inferior view-synthesis results. Hence, our joint formulation provides robustness to the camera pose while giving better multi-scale image rendering.

RM-NeRF (w/o pose, E2E) with COLMAP poses. We use the Tanks and Temples dataset to conduct this study. We use COLMAP camera poses to initialize the RM-NeRF (w/o poses) and RM-NeRF (E2E) methods and compare them against the RM-NeRF result. The results are provided in Table 4. It is easy to infer from the table that both RM-NeRF (E2E) and RM-NeRF (w/o pose) are able to outperform the base RM-NeRF method. This shows that the extension of RM-NeRF proposed in the article is better in a direct comparison setup, hence an encouraging take on the problem. The proposed extension could also work for a more realistic setting where the trajectory is sparse, and COLMAP might not be a very reliable pipeline, as we will also show in the later subsection with more realistic datasets.

4.1.3 ScanNet

ScanNet is a widely used dataset to benchmark 3D reconstruction and semantic segmentation algorithm results for indoor scenes. Its train and validation sets contain 2.5M RGB-D images for 1512 scans acquired in 707 spaces. This dataset is collected using hardware-synchronized RGB and depth cameras of an iPad Air 2 at 30Hz exhibiting a realistic hardware setup.

We studied the performance of our approaches on this dataset to observe view-synthesis result on room-scale indoor scene. Recently NoPe-NeRF (Bian et al., 2022) also performed such a study; therefore, we used their experimental setup for this study. This experiment involves subsampling the image sequences leading to 80–100 images per sequence. It is evaluated under two settings: **(I)** with intrinsics: where the intrinsic matrix is known, and we optimize for camera poses and **(II)** w/o intrinsics: where we optimize for both the camera intrinsic as well as extrinsic parameters. In setting **(I)**, we compare our RM-NeRF (w/o pose) method with the NoPe-NeRF baseline, which assumes given intrinsic parameters. For setting **(II)**, we compare our RM-NeRF (E2E) method with the NeRF–baseline, which optimizes both camera extrinsic and intrinsic parameters.

Table 5 shows both settings' experimental results using PSNR and LPIPS metrics. For each scenario, our proposed method outperforms the relevant baseline. This shows the effectiveness of our proposed joint optimization approaches in view-synthesis for indoor scene, thereby showcasing the benefit of motion averaging and multi-scale modeling.

4.2 NAVI Dataset

NAVI is a recently proposed image collection dataset comprising scenes captured in the wild. It contains images of an object with various backgrounds and illumination conditions, which is an apt setting to test our proposed approaches. We evaluate the three proposed approaches on six complex scenes. Meanwhile, COLMAP performs poorly on 19 out of the 36 scenes from this dataset. For RM-NeRF, we use the COLMAP pose initialization whereas, for RM-NeRF (w/o pose) and RM-NeRF (E2E), camera parameters are initialized randomly. Table 6 provides the results of our approaches on this dataset. It is easy to infer that RM-NeRF (w/o pose) can surpass RM-NeRF in this setup, demonstrating the advantage of our introduced extension in this article. Furthermore, the RM-NeRF (E2E) is better as compared to RM-NeRF in this case and is marginally inferior to the RM-NeRF (w/o pose), showing the possibility of a fully uncalibrated framework without sacrificing much on the rendering quality. Thus, RM-NeRF (E2E) method might be an excellent **self-contained framework** for view-synthesis in-the-wild or unstructured scenes instead of relying on third-party software such as COLMAP.

4.3 Black-Box Dataset

4.3.1 Images Taken from a Freely Moving Camera

To simulate a general real-world multi-view image capture setup for view-synthesis, we collect a dataset using a freely moving mobile phone. We captured 22 images of a simple black-box object (refer Fig. 7) using 16 out of them for training the model. The camera poses are shown in Fig. 8 demonstrating that the images are captured at varying distance from the object. The aim of this experiment is to show that for such real-world scene capture using COLMAP camera pose is not encouraging take for modeling view-synthesis problem. We must further refine the camera pose via mindful optimization. Accordingly, we compare our RM-NeRF approach result with Mip-NeRF by using COLMAP camera poses as initialization.

Figure 5 shows the qualitative results for this scene in the last two columns alongside the Multi-Scale Blender dataset. For completeness, we included the results of the BARF method (Lin et al., 2021). It is quite intuitive to assume that Mip-NeRF could have localized the object incorrectly, whereas our method can cast the apt cone in the scene space for object localization. And therefore, our method provides much better image rendering results. Additionally, it helps us deduce that assuming COLMAP poses as ground-truth poses can be misleading for real-world scenarios. This shows how a robust estimation approach on top of COLMAP camera poses initialization might effectively generalize the method for day-to-day captured multiview images.

We further study the RM-NeRF (w/o pose) and RM-NeRF (E2E) performance on the black box dataset. The results for both these sequences are shown in Fig. 9. Clearly, the result shows the suitability of our introduced extension. The result demonstrates that it is quite possible to model the scene representation without access to COLMAP camera poses or camera intrinsic parameters without sacrificing much on the view-synthesis rendering quality.

4.3.2 Specular Objects

We studied the behavior of RM-NeRF (w/o pose) on objects with specular surfaces. As it is well-known that specular objects are often challenging to model for view synthesis, we test the limits of RM-NeRF (w/o pose) by conducting this study. Even recent works such as Yen-Chen et al. (2022b) highlight this issue on their dataset comprising objects such as 'spoon', 'fork', etc. This dataset comprises eight object-

Table 4 Comparison of our proposed methods with the baselines, on the Tanks and Temples dataset, under three different setups based on input required by the methods being evaluated

	Truck		M60		Train		Family		Ignatius		Horse		Museum		Francis	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
COLMAP poses and intrinsics																
Mip-NeRF	25.37	0.34	26.42	0.34	24.97	0.48	25.98	0.42	24.22	0.46	27.45	0.26	26.71	0.43	29.57	0.40
Point-NeRF	25.29	0.34	26.54	0.33	24.84	0.49	25.76	0.44	24.13	0.47	26.88	0.29	26.77	0.43	28.88	0.43
RM-NeRF	**26.12**	**0.32**	**27.60**	**0.32**	**25.78**	**0.42**	**27.12**	**0.29**	**25.29**	**0.37**	**27.89**	**0.25**	**27.32**	**0.41**	**29.89**	**0.38**
COLMAP intrinsics, random poses																
BARF	24.72	0.53	24.89	0.54	21.99	0.67	22.87	0.61	21.22	0.64	23.88	0.48	22.89	0.59	25.12	0.65
NoPe-NeRF	25.28	0.35	26.31	0.36	24.88	0.49	25.93	0.44	23.88	0.49	27.28	0.28	26.68	0.43	29.28	0.42
RM-NeRF (w/o pose)	**25.81**	**0.33**	**26.71**	**0.32**	**24.88**	**0.49**	**26.07**	**0.41**	**24.47**	**0.44**	**27.64**	**0.25**	**26.89**	0.42	**29.48**	**0.40**
w/o COLMAP poses and intrinsics																
NeRF–	24.98	0.51	24.67	0.55	21.87	0.69	22.98	0.55	21.03	0.67	23.17	0.45	22.17	0.51	25.62	0.50
RM-NeRF (E2E)	**25.12**	**0.36**	**25.88**	**0.39**	**24.07**	**0.52**	**24.89**	**0.47**	**23.28**	**0.57**	**26.02**	**0.34**	**24.97**	**0.48**	**27.97**	**0.48**
with same initialization as RM-NeRF																
RM-NeRF(w/o pose)	**26.93**	**0.28**	**28.12**	**0.30**	**26.46**	**0.34**	**27.95**	**0.24**	**26.34**	**0.28**	**28.34**	**0.20**	**28.09**	**0.37**	**30.43**	**0.31**
RM-NeRF (E2E)	26.57	0.31	27.96	0.31	25.99	0.42	27.89	0.28	25.78	0.37	28.09	0.24	27.78	0.39	29.97	0.37

The table shows PSNR and LPIPS metric values for each of the methods on various scenes of the dataset. Clearly, in all the three scenarios, our methods are able to surpass the baselines corresponding to that setup, thereby proving to be effective for real-world cases

The bold values indicate that the best results in the corresponding category

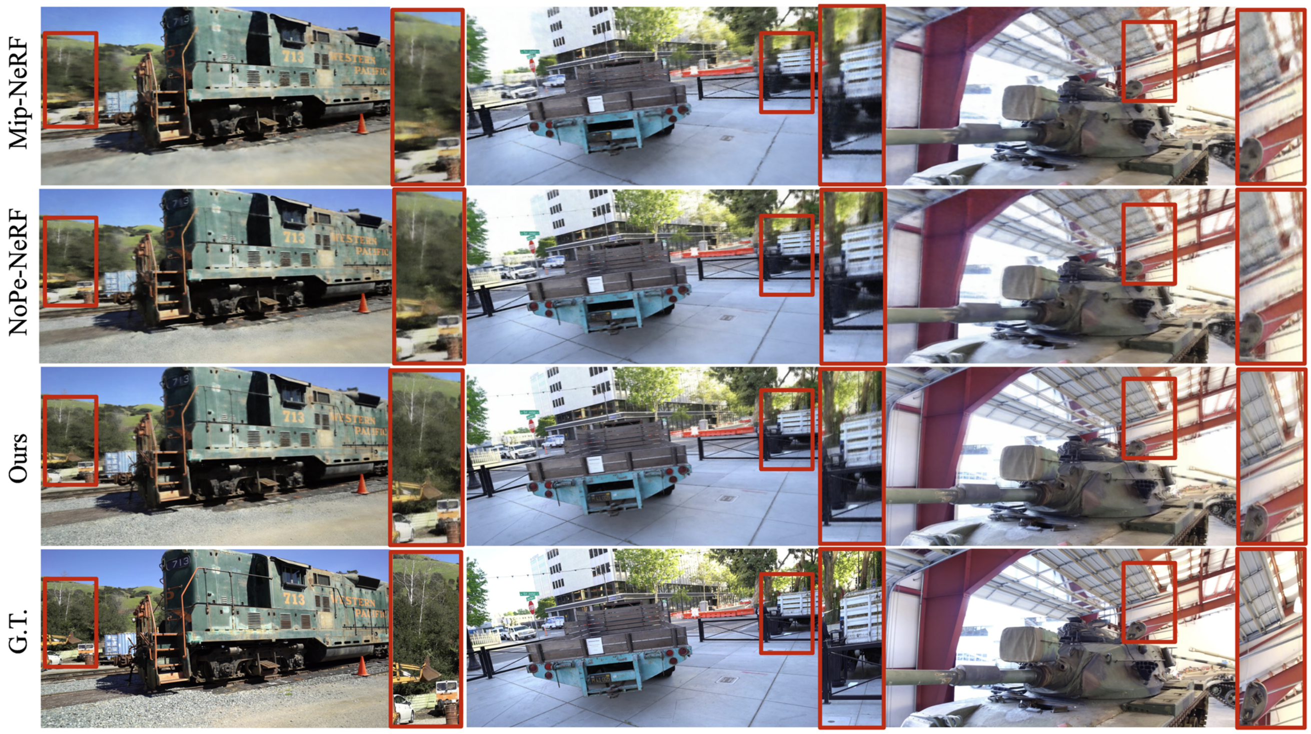


Fig. 6 Qualitative Comparison of our RM-NeRF method, Mip-NeRF (Barron et al., 2021) and NoPe-NeRF (Bian et al., 2022) on the Tanks and Temples dataset comprising real world scenes. We have visualized the synthesized images from all the approaches on 3 scenes namely Truck, M60 and Train. Our method clearly provide better synthesized images as emphasized by the red boxes (Color figure online)

Table 5 ScanNet

Scene	With intrinsics		w/o intrinsics	
	Nope-NeRF	Ours (w/o pose)	NeRF–	Ours (E2E)
0079_00	32.47/0.41	**33.12/0.39**	30.59/0.49	**31.88/0.47**
0418_00	31.33/0.34	**32.07/0.32**	30.00/0.40	**31.23/0.46**
0301_00	30.83/0.36	**30.83/0.35**	27.84/0.45	**29.14/0.42**
0431_00	33.83/0.39	**34.09/0.38**	31.44/0.45	**32.23/0.44**

Performance comparison of our methods with NoPe-NeRF (Bian et al., 2022) and NeRF–Wang et al. (2021b) methods on 4 scenes of ScanNet dataset. The values in the table are of the format PNSR/LPIPS, respectively
The bold values indicate that the best results in the corresponding category

Table 6 Results on Navi dataset

Method	PSNR↑	LPIPS↓	SSIM↓	Rotation° ↓	Translation↓
RM-NeRF	22.41	0.34	0.73	26.54	24.91
RM-NeRF (w/o pose)	**23.12**	**0.29**	**0.79**	**22.76**	**21.23**
RM-NeRF (E2E)	22.97	0.33	0.78	24.13	22.23

Comparison of our proposed methods on navi dataset. For this, we have used 6 scenes from its in-the-wild image collection where COLMAP struggles to reconstruct accurate poses
The bold values indicate that the best results in the corresponding category

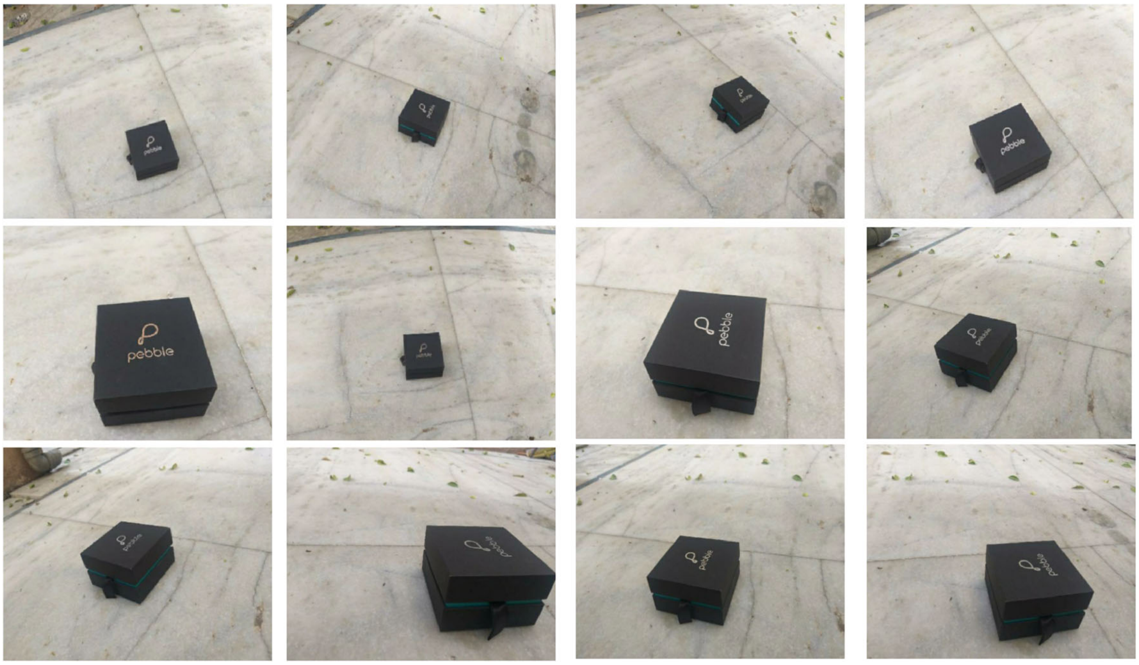


Fig. 7 A subset of images corresponding to our black box dataset, captured using phone by randomly moving w.r.t. the box (Sect. 4.3.1). The aim behind this using scene is to mimic a very generic scenario representing day-to-day captured multi-view image sets

centric scenes with 30–50 images of a reflective material object per scene. The scenes are captured using a slow-moving mobile phone around an object. Yen-Chen et al. (2022b) estimates the camera poses and object's 3D using COLMAP. For testing, 8–10 images per scene are used.

For this experiment, we used the shiny fork object sequence comprising 38 training images. We followed the same setup as Yen-Chen et al. (2022b) for evaluation. Figure 10 shows the camera poses and sparse 3D points recovered for this scene. Figure 11 shows our method's view synthesis result on this sequence.

Following the setup, we separate eight images for testing. Figure 10 shows the camera poses and 3D points recovered via COLMAP on this scene, demonstrating the challenges in dealing with specular surfaces. Figure 11 provides our method's view synthesis result on this sequence. Figure 11 result shows 4 rendered images obtained using our method. The model hasn't seen the object from this viewpoint at train time. Despite favorable results in modeling view-synthesis for such an object, it is observed that our method has clear limitations in modeling it.

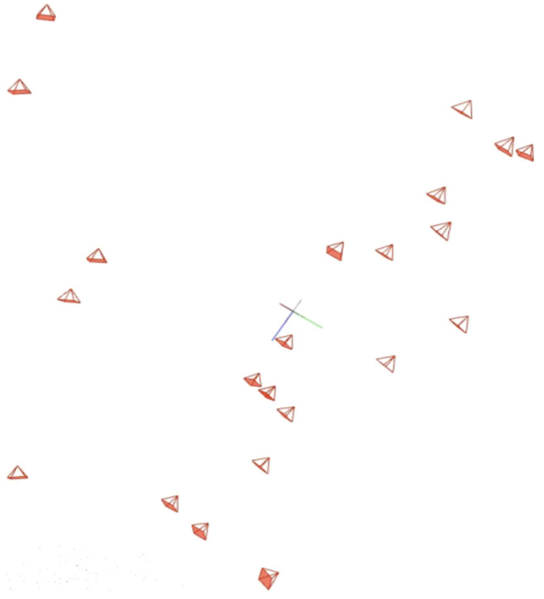


Fig. 8 Camera poses and a sparse collection of 3D points corresponding to our black-box sequence (described in Sect. 4.3.1) recovered using COLMAP (Schonberger & Frahm, 2016). These estimated camera poses point verify the randomness in motion while capturing the back-box scene. Also, the recovered set of 3D points (bottom-left) appear to be quite sparse further pointing to the failure of COLMAP in recovering the correct 3D and poses for this case

4.4 Ablations

4.4.1 Synthetic Datasets

Here, we analyse our method's performance on the original Blender data (Mildenhall et al., 2021) with camera pose error, Multi-Scaled Blender data (Barron et al., 2021) with ground truth poses and a variation of our RM-NeRF method with unbiased weighting of the rotation averaging and rendering losses for updating camera poses.

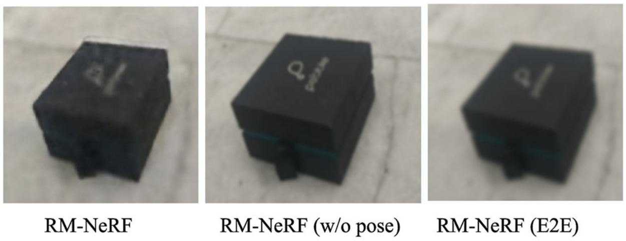


Fig. 9 Results on the black box dataset for the proposed version. Here, RM-NeRF (w/o pose) provides more accurately rendering of the object, while RM-NeRF and RM-NeRF(E2E) approaches results are quite similar. Note that RM-NeRF (E2E) start from a randomly initialized pose without the knowledge of camera intrinsic parameters, hence demonstrating its suitability

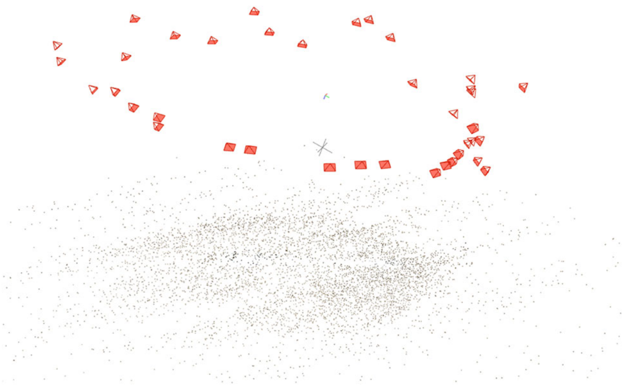


Fig. 10 Approximate poses recovered by using COLMAP on the scene centered around a reflective fork object (Sect. 4.3.2), proposed in Yen-Chen et al. (2022b)

(a) Same Scale Images with Camera Pose Error. We analyze our method's image-rendering results with the Mip-NeRF and BARF on the original NeRF Blender dataset in the presence of noisy rotations. Here, all images have the same resolution and are captured at a fixed distance from the object. The results for this setup are shown in Table 7, using the PSNR, LPIPS, and SSIM metrics for the four scenes. The statistical comparison show that our method supersedes the Mip-NeRF baseline. Also, it is comparable to the BARF method performance, which was designed to handle pose errors for single-scale datasets where all images are taken at approximately the same distance from the object.

(b) Multi Scale Images with Ground-Truth Pose. We further study the multi-scale case on the multi-scale Blender dataset, but this time without perturbing the ground truth poses. Table 8 compares the PSNR, LPIPS and SSIM values for this scenario. Once again, our method performance is similar to the current baselines, and the difference to the best method (Barron et al., 2021) is minor, thereby showing the effectiveness of our approach. The point to note is, using our method, we don't have to rely on a separate module for estimating accurate pose and it is recovered jointly with the object's neural representation.

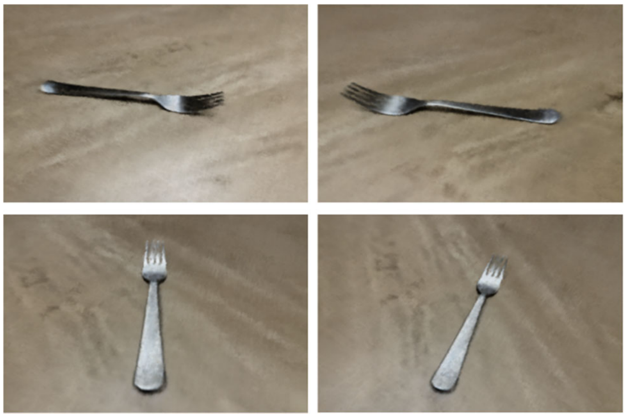


Fig. 11 Qualitative evaluation of our RM-NeRF (w/o pose) method on a real-world scene (captured using phone) centered around a shiny fork object (Sect. 4.3.2), proposed in Yen-Chen et al. (2022b)

Looking at the rendered image quality results of both the tables, i.e., Tables 7 and 8, our method performs well on both settings showing a clear advantage.

(c) Unbiased Optimization of Eq. (12). We performed this study to provide a better insight into our weighted loss optimization strategy for the RM-NeRF method. For this, we initialized $\lambda = 0.5$ in Eq. (12) in the overall optimization. Table 9 provides the results for this study using the PNSR, LIPIPS metric on the Multi-Scale blender dataset with noisy camera poses. The unbiased optimization variant result is denoted as Ours† in Table 9. The results clearly show the benefit of biased optimization in Sect. 3.1.2. It can be observed that this unbiased optimization results in inferior performance due to the complex optimizing landscape. Hence, the proposed biased optimization is suited for such loss function optimization. Initially, the bias is built toward estimating correct poses, which is gradually decreased with time. This leads to adequate minima after convergence.

4.4.2 Real-World Datasets

Here, we study our approach under two circumstances often observed in any imaging data used for practical purposes. The first study corresponds to noisy monocular depth maps, and the second corresponds to the noise in input images. To study both cases, we use the Tanks and Temples dataset (Knapitsch et al., 2017) comprising real-world images. We add some noise to the predicted monocular depth maps to simulate a general scene, which is usually valid for complex and cluttered scenes. Similarly, we also analyze the performance of our proposed methods in the presence of noise in the input images, accounting for the case with unclear or occluded regions in the day-to-day collected images.

(a) Noisy 3D prior. Noise in the estimated monocular depths ultimately leads to noisy 3D prior for: (a) updating initial

Table 7 Performance comparison of the proposed RM-NeRF method and Mip-NeRF, BARF baselines using 4 scenes from the NeRF Blender dataset (Mildenhall et al., 2021), with noise added to the poses

	Lego			Ship			Drums			Mic		
	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
Mip-NeRF	17.90	0.089	0.82	22.90	0.107	0.71	14.07	0.110	0.799	21.90	0.064	0.930
BARF	**27.61**	0.05	**0.92**	**26.18**	0.121	**0.74**	23.68	0.095	0.880	27.03	0.060	0.960
RM-NeRF(ours)	27.10	**0.048**	**0.92**	25.45	**0.069**	0.73	**24.98**	**0.072**	**0.907**	**30.03**	**0.027**	**0.963**

The evaluation is carried out using PSNR, LPIPS and SSIM metrics

The bold values indicate that the best results in the corresponding category

Table 8 Performance comparison of the proposed RM-NeRF method and Mip-NeRF, BARF baselines using 4 scenes from the multi-scale Blender dataset (Barron et al., 2021), utilizing ground truth poses

	Lego			Ship			Drums			Mic		
	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
Mip-NeRF	**29.34**	**0.045**	**0.938**	**28.64**	**0.0651**	**0.778**	**26.9**	**0.0452**	0.92	**34.70**	0.0088	**0.978**
BARF	11.18	0.520	0.700	9.230	0.760	0.480	11.2	0.680	0.66	12.18	0.530	0.740
RM-NeRF (ours)	29.30	**0.045**	0.929	28.57	0.0653	**0.778**	26.7	0.0455	**0.93**	34.30	**0.0082**	0.969

The evaluation is carried out using PSNR, LPIPS and SSIM metrics
The bold values indicate that the best results in the corresponding category

Table 9 Performance analysis of our method with unbiased joint optimization ($\lambda = 0.5$ in Eq. (12))

	Mean error (°)		Single-scale dataset		Multi-scale dataset	
	Ours†	Ours	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
Lego	0.86	**0.03**	24.0(**27.1**)	0.8(**0.5**)	22.2(**27.0**)	0.6(**0.4**)
Ship	0.74	**0.05**	23.7(**25.4**)	1.0(**0.7**)	23.3(**26.5**)	0.7(**0.7**)
Drum	0.83	**0.03**	20.2(**24.9**)	0.8(**0.7**)	15.6(**26.0**)	0.7(**0.4**)
Mic	0.42	**0.07**	24.2(**30.0**)	0.6(**0.3**)	22.6(**30.0**)	0.4(**0.1**)
Chair	0.47	**0.06**	27.4(**33.7**)	0.3(**0.5**)	25.7(**35.2**)	0.7(**0.3**)
Ficus	0.78	**0.07**	24.2(**28.6**)	0.7(**0.4**)	22.7(**29.2**)	0.7(**0.3**)
Mat.	1.12	**0.05**	20.2(**25.1**)	0.7(**0.7**)	18.8(**24.8**)	0.9(**0.6**)
H.D.	0.62	**0.05**	26.5(**33.1**)	0.6(**0.3**)	23.3(**32.5**)	0.8(**0.3**)

This ablation is denoted as Ours†. We analyze both pose estimation and learned scene representations of this ablation. The first two columns compare the mean error in estimating poses using this ablation and our method which uses a biased optimization strategy in a single scale scale case. The remaining columns analyze the PSNR and LPIPS metric values for this ablation on the original and multi-scale versions of Blender dataset. For reference, we have provided the PSNR and LPIPS values for our RM-NeRF method, from Table 1, in the brackets

The bold values inside the bracket shows the performance that is achieved using the proposed biased joint optimization as compared to the unbiased one

Table 10 Performance analysis of our RM-NeRF (w/o pose) method and the NoPe-NeRF (Bian et al., 2022), PointNeRF (Xu et al., 2022) baselines on scenes from Tanks and Temples dataset, when noise is added to the 3D prior (monocular depth maps in case of RM-NeRF (w/o pose), NoPe-NeRF and MVS depths in case of PointNeRF)

	NoPe-NeRF		Point-NeRF		Ours	
	Noise	Clean	Noise	Clean	Noise	Clean
M60	24.47	26.31	24.23	26.54	**24.91**	**26.71**
Family	23.12	25.93	22.97	25.76	**23.78**	**26.07**
Ignatius	21.67	23.88	21.79	24.13	**22.23**	**24.47**

The bold values indicate the best results in the corresponding category

poses $\hat{\mathcal{P}}$ for our RM-NeRF (w/o pose), RM-NeRF (E2E) methods via point cloud alignment and (b) aligning depths to a global frame using rendered depth and parameters α, β. To simulate this, we add noise $\delta \sim \mathcal{N}(\mathbf{0}, 1e^{-1}\mathbf{I})$ to a fraction of pixels in the normalized estimated depth maps, sampled uniformly throughout the image.

Table 10 shows our method's results with noisy poses and the other relevant baselines in this setup on three sequences from the Tanks and Temples dataset, namely 'M60', 'Family', and 'Ignatius'. For comparison, it shows results with and without (*w/o*) this noise. We compared the results with the recent NoPe-NeRF (Bian et al., 2022) and Point-NeRF (Xu et al., 2022) method. For Point-NeRF, we add the noise δ to the MVS estimated depth maps. All the methods' performance degrades with noise, yet our method performs better than others.

(b) Image quality and noise. Day-to-day collected images can have noise due to low-quality imaging sensors or bad physical condition of the scene, which could effect the scene representation using images. We performed this study to observe the sensitivity of the proposed approaches to such noisy images. For this, we add holes to the given scene images, at randomly selected locations. Alongside this, we also add a small gaussian blur (std=0.1) to certain randomly sampled locations in the image. This is done to simulate the occlusion effect.

Table 11 shows the results for this experiment on three Tanks and Temples dataset namely 'M60', 'Family', and 'Ignatius'. All the three proposed approaches have observable drop in their PSNR values, with E2E version having the maximum, when using these noisy images. It shows the vulnerability of our approach to noisy images.

4.5 Analyzing Estimated Camera Intrinsic Parameters

Here, we provide the statistical results obtained using our RM-NeRF (E2E) approach in estimating intrinsic camera

Table 11 Performance analysis of our proposed methods on three scenes from the Tanks and Temples dataset (Knapitsch et al., 2017), with noise added to the input images

	RM-NeRF		RM-NeRF (w/o pose)		RM-NeRF (E2E)	
	Noise	Clean	Noise	Clean	Noise	Clean
M60	26.91	27.60	26.12	26.71	24.79	25.88
Family	26.41	27.12	25.43	26.07	23.93	24.89
Ignatius	24.71	25.29	23.71	24.47	22.07	23.28

Table 12 Performance comparison (in percentage error) of our RM-NeRF (E2E) method and the baselines for focal length estimation on the Tanks and Temples dataset (Knapitsch et al., 2017)

	Tanks and temples			ScanNet		
	M60	Family	Ignatius	0079	0418	0301
SC-NeRF	2.8	26.9	2.3	6.3	4.9	8.7
NeRF−−	3.8	3.7	2.2	6.1	4.7	8.9
Ours	**1.6**	**1.3**	**1.1**	**2.5**	**3.3**	**3.1**

The table shows focal error Δf results of our method compared to NeRF– (Wang et al., 2021b), and SC-NeRF (Jeong et al., 2021). The results are compiled w.r.t. assumed ground-truths for these datasets, i.e., COLMAP results for Tanks and Temples and BundleFusion for ScanNet

The bold values indicate that the lowest focal length error

parameters. We compared our approach's result with popular NeRF–(Wang et al., 2021b) and recently proposed SC-NeRF (Jeong et al., 2021), which estimates both camera intrinsic and extrinsic parameters. For the experimental evaluation, we use three scenes from Tanks and Temples and ScanNet datasets. For Tanks and Temples, the authors have provided COLMAP parameters as the pseudo ground truth, which we used as it is for evaluation. Likewise, ScanNet's camera poses recovered using Bundle Fusion are treated as the pseudo ground truth.

Table 12 shows the difference in focal lengths, Δf (in pixels) estimation by our approach compared to relevant baselines. Our approach performs better than the baselines, hence can be a good step towards differentiable intrinsic estimation leading to a complete end-to-end pipeline for joint estimation camera parameters—intrinsic, extrinsic, and scene representation, using images.

4.6 Motion Averaging Analysis

A critical idea from multi-view geometry used in this article is motion averaging. In this section, we provide insights into the usefulness of motion averaging in providing robustness to the camera pose estimation pipeline.

4.6.1 View Graphs Analysis

We conducted a simple test using the Blender dataset to show the benefits of view-graph modeling in motion averaging for robust camera pose estimation. For this, we perturbed one-fifth of the camera poses corresponding to each scene and constructed a view graph. We analyze camera pose errors before and after applying our pose-refining network to the view graph constructed using perturbed poses. Table 13 result shows the robustness of our method in camera pose estimation. It can be observed that our pose-refining network handles noise efficiently and significantly reduces the overall camera pose error.

Table 13 Our pose-refinement GNN results on noisy camera poses synthesized from Blender dataset (Mildenhall et al., 2021)

	Noisy (°)		Improved (°)	
	Mean	rms	Mean	rms
Lego	1.78	4.24	**0.031**	**0.041**
Ship	2.12	4.02	**0.052**	**0.063**
Drums	1.98	3.65	**0.038**	**0.052**
Mic	2.36	5.12	**0.073**	**0.091**
Chair	1.76	3.45	**0.056**	**0.071**
Ficus	2.46	5.34	**0.065**	**0.096**
Materials	1.58	3.55	**0.046**	**0.074**
Hotdog	2.23	4.78	**0.052**	**0.071**

The column corresponding to the Noisy (°) and Improved (°) denote the camera pose estimation error before and after applying our method, respectively

Bold values indicate that the lower error in camera pose recovery

4.6.2 Camera Pose Error Analysis in Presence of Noisy Feature Key-Point Correspondence

Images captured in a real-world setting often contain noises leading to misleading keypoint matching between frames. This can affect the camera pose estimation results using the popular COLMAP framework. And therefore, to analyze this effect of noisy correspondences on our pose-refining method and COLMAP, we perform this simple experiment on the Lego scene from the Blender dataset. We first generate pair-wise keypoint correspondences for images. Then, we add noise to these correspondences and estimate the relative pose between every image pair using these noisy correspondences. We then estimate absolute rotations from these noisy relative pose estimates using both approaches, i.e., COLMAP and our pose-refining GNN. Using these absolute camera rotation estimates, we compute corresponding camera translations. Figure 12 shows the difference in predicted poses by each of these methods w.r.t. the ground truth poses provided by the dataset. Our approach offers robustness to such noise and attains significantly lesser error (consistent for all the images) compared to COLMAP in this scenario. The difference in the recovered camera pose is shown using lines. The results clearly show the effectiveness of our approach. Our approach gives better results than COLMAP, and the recovered camera pose is consistently better across images.

The quantitative results for these experiments are provided in Table 14. For clarity, the rotation and translation error are provided separately.

Limitations. Our proposed approaches could perform poorly on scenes containing specular and highly reflective surfaces. Moreover, further improvements could be made to our RM-NeRF (E2E) approach to apply to images collected from the internet, where each image of the same object is captured from a different camera.

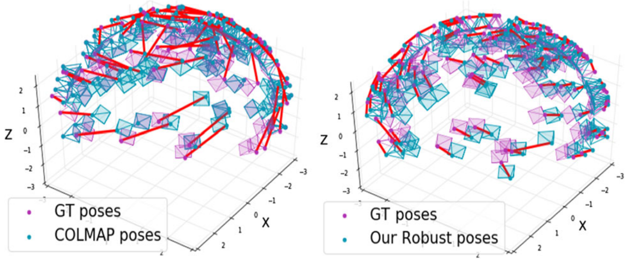


Fig. 12 Performance comparison of COLMAP and our pose-refining GNN w.r.t. ground truth poses on the Lego scene from the Blender dataset (Mildenhall et al., 2021), in presence of noise to the matched keypoint correspondences. The read line corresponds to the translation error in the estimated poses. Left: Camera pose output from COLMAP (Schonberger & Frahm, 2016) along with the ground truth poses provided by the dataset. Right: Camera pose output from our poses refining GNN alongside the ground truth poses. It can be observed that our estimated poses are much closer to the ground truth as compared to COLMAP

Table 14 Quantitative results for the noisy image keypoint correspondence case

	Rot. error (°)		Trans. error (cm)	
	COLMAP	Ours	COLMAP	Ours
Lego	18.3°	**12.4°**	10.2	**7.1**

Bold values indicate that the lower error in camera rotation and camera translation recovery, respectively

5 Conclusion and Future Directions

In this paper, we introduce two extensions of our published work that allows the NeRF based scene representation for continuous view synthesis to work well for daily captured multi-view images. Specifically, the proposed approaches addresses the practical view synthesis issues around multi-scale images and the unavailability of camera parameters at train time. These issues are addressed using concepts from multi-view geometry, NeRF representation, and existing robust camera pose estimation literature. Although the proposed approaches may not be perfect, they open up the scope for modeling a randomly captured image set using continuous neural volumetric rendering without relying on third-party software such as COLMAP, hence self-contained framework. One interesting future direction is to extend our RM-NeRF (E2E) method to a scenario where different cameras are used to capture the multi-view images, leading to continuously varying intrinsic parameters. This can further broaden the scope of these NeRF-based methods to randomly collect images of a scene from the internet uploaded by different users. Another interesting direction is to extend our RM-NeRF (w/o pose) and RM-NeRF (E2E) methods to scenes containing specular objects exhibiting interreflection.

Dataset Availability The datasets used in this paper are publicly available. Their names and links are as follows:

1. NeRF Blender Dataset
2. Multi-Scale Blender Dataset
3. Tanks and Temples Dataset
4. ScanNet Dataset
5. Fork Scene
6. **Black Box example**: This is not public yet, but we will put online after the reviews.

Other underlying data related to paper such as authors Orcid-ID and institution affiliation are publicly available online.

References

Aftab, K., Hartley, R., & Trumpf, J. (2014). Generalized weiszfeld algorithms for lq optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*(4), 728–745.

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., & Szeliski, R. (2011). Building Rome in a day. *Communications of the ACM, 54*(10), 105–112.

Amanatides, J. (1984). Ray tracing with cones. *ACM SIGGRAPH Computer Graphics, 18*(3), 129–135.

Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., & Srinivasan, P. P. (2021). Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5855–5864.

Bian, W., Wang, Z., Li, K., Bian, J. W., & Prisacariu, V. A. (2022). Nope-nerf: Optimising neural radiance field with no pose prior. arXiv preprint arXiv:2212.07388.

Chatterjee, A., & Govindu, V. M. (2017). Robust relative rotation averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(4), 958–972.

Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., & Su, H. (2021). Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14124–14133.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the computer vision and pattern recognition (CVPR)*. IEEE.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR.

Govindu, V. M. (2001). Combining two-view constraints for motion estimation. In *CVPR* (Vol. 2). IEEE.

Govindu, V. M. (2006). Robustness in motion averaging. In *Asian conference on computer vision*, pp. 457–466. Springer.

Govindu, V. M. (2016). Motion averaging in 3d reconstruction problems. In *Riemannian computing in computer vision*, pp. 145–164. Springer.

Hartley, R., Aftab, K., & Trumpf, J. (2011). L1 rotation averaging using the Weiszfeld algorithm. In *CVPR 2011*, pp. 3041–3048. IEEE.

Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(6), 580–593.

Hartley, R., Trumpf, J., Dai, Y., & Li, H. (2013). Rotation averaging. *International Journal of Computer Vision, 103*(3), 267–305.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. London: Cambridge University Press.

Jain, N., Kumar, S., & Gool, L.V. (2022). Robustifying the multi-scale representation of neural radiance fields. In *33rd British machine* *vision conference 2022*, BMVC 2022, London, UK, November 21–24, 2022. BMVA Press.

Jampani, V., Maninis, K.K., Engelhardt, A., Truong, K., Karpur, A., Sargent, K., Popov, S., Araujo, A., Martin-Brualla, R., Patel, K., Vlasic, D., Ferrari, V., Makadia, A., Liu, C., Li, Y., & Zhou, H. (2023). Navi: Category-agnostic image collections with high-quality 3d shape and pose annotations. In: arXiv preprint.

Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., & Park, J. (2021). Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5846–5854.

Kaya, B., Kumar, S., Sarno, F., Ferrari, V., & Van Gool, L. (2022). Neural radiance fields approach to deep multi-view photometric stereo. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1965–1977.

Kingma, D. P., & Ba, J. (2014). Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Knapitsch, A., Park, J., Zhou, Q. Y., & Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, *36*(4).

Lee, S., Chen, L., Wang, J., Liniger, A., Kumar, S., & Yu, F. (2022). Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robotics and Automation Letters*, *7*(4), 12070–12077.

Li, X., & Ling, H. (2021). Pogo-net: Pose graph optimization with graph neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5895–5905.

Lin, C. H., Ma, W. C., Torralba, A., & Lucey, S. (2021). Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5741–5751.

Liu, L., Gu, J., Zaw Lin, K., Chua, T. S., & Theobalt, C. (2020). Neural sparse voxel fields. *Advances in Neural Information Processing Systems, 33*, 15651–15663.

Martel, J. N., Lindell, D. B., Lin, C. Z., Chan, E. R., Monteiro, M., & Wetzstein, G. (2021). Acorn: adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics (TOG), 40*(4), 1–13.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM, 65*(1), 99–106.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(6), 756–770.

Purkait, P., Chin, T. J., & Reid, I. (2020). Neurora: Neural robust rotation averaging. In *European conference on computer vision*, pp. 137–154. Springer.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., & Courville, A. (2019). On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR.

Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12179–12188.

Schonberger, J. L., & Frahm, J. M. (2016). Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113.

Schönberger, J. L., Zheng, E., Pollefeys, M., & Frahm, J. M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *European conference on computer vision (ECCV)*.

Sucar, E., Liu, S., Ortiz, J., & Davison, A. J. (2021). imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6229–6238.

Tewari, A., et al. (2022). Advances in neural rendering. In *Computer graphics forum, Wiley Online Library*, Vol. 41, pp. 703–735.

Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). Bundle adjustment—a modern synthesis. In *Proceedings of the international workshop on vision algorithms: theory and practice*, ICCV'99, pp. 298–372. Springer, London, UK.

Wang, Q., Wang, Z., Genova, K., Srinivasan, P. P., Zhou, H., Barron, J. T., Martin-Brualla, R., Snavely, N., & Funkhouser, T. (2021a). Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699.

Wang, Z., Wu, S., Xie, W., Chen, M., & Prisacariu, V. A. (2021b). Nerf–: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064.

Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., & Neumann, U. (2022). Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5438–5448.

Yang, L., Li, H., Rahim, J. A., Cui, Z., & Tan, P. (2021). End-to-end rotation averaging with multi-source propagation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11774–11783.

Yao, Y., Luo, Z., Li, S., Fang, T., & Quan, L. (2018). Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 767–783.

Yen-Chen, L., Florence, P., Barron, J.T., Lin, T. Y., Rodriguez, A., & Isola, P. (2022a). Nerf-supervision: Learning dense object descriptors from neural radiance fields. In *2022 international conference on robotics and automation (ICRA)*, pp. 6496–6503. IEEE.

Yen-Chen, L., Florence, P., Barron, J. T., Lin, T. Y., Rodriguez, A., & Isola, P. (2022b). NeRF-Supervision: Learning dense object descriptors from neural radiance fields. In *IEEE conference on robotics and automation (ICRA)*.

Yen-Chen, L., Florence, P., Barron, J. T., Rodriguez, A., Isola, P., & Lin, T. Y. (2021). inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 1323–1330. IEEE.

Yu, A., Ye, V., Tancik, M., & Kanazawa, A. (2021). 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.

Zhang, X., Srinivasan, P. P., Deng, B., Debevec, P., Freeman, W. T., & Barron, J. T. (2021). Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG), 40*(6), 1–18.