Instant Uncertainty Calibration of NeRFs Using a Meta-Calibrator

Niki Amini-Naieni[1], Tomas Jakab[1], Andrea Vedaldi[1], and Ronald Clark[1]

University of Oxford

Abstract. Neural Radiance Fields (NeRFs) have markedly improved novel view synthesis, but accurate uncertainty quantification in their image predictions remains an open problem. The prevailing methods for estimating uncertainty, including the state-of-the-art Density-aware NeRF Ensembles (DANE) [29], quantify uncertainty without calibration. This frequently leads to over- or under-confidence in image predictions, which can undermine their real-world applications. In this paper, we propose a method which, for the first time, achieves calibrated uncertainties for NeRFs. To accomplish this, we overcome a significant challenge in adapting existing calibration techniques to NeRFs: a need to hold out ground truth images from the target scene, reducing the number of images left to train the NeRF. This issue is particularly problematic in sparse-view settings, where we can operate with as few as three images. To address this, we introduce the concept of a meta-calibrator that performs uncertainty calibration for NeRFs with a single forward pass without the need for holding out any images from the target scene. Our meta-calibrator is a neural network that takes as input the NeRF images and uncalibrated uncertainty maps and outputs a scene-specific calibration curve that corrects the NeRF's uncalibrated uncertainties. We show that the meta-calibrator can generalize on unseen scenes and achieves well-calibrated and state-of-the-art uncertainty for NeRFs, significantly beating DANE and other approaches. This opens opportunities to improve applications that rely on accurate NeRF uncertainty estimates such as next-best view planning and potentially more trustworthy image reconstruction for medical diagnosis.

Keywords: NeRFs · Uncertainty calibration · Few-shot learning

1 Introduction

Recent advancements in scene representations have led to promising new approaches for novel view synthesis and scene reconstruction. Among these, Neural Radiance Fields (NeRFs) [16] have emerged as a particularly powerful tool, offering unprecedented levels of realism and detail in rendered images. The core idea behind NeRFs is to represent a scene as a continuous vector-valued function, parameterized by a neural network, that maps spatial coordinates and view directions to color and density values. This approach enables the creation

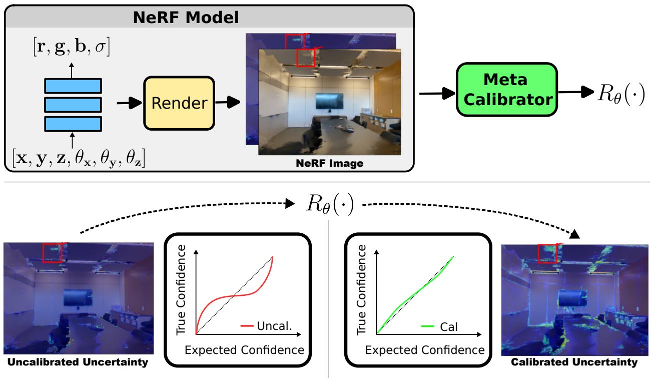


Fig. 1: We propose a method for efficiently calibrating the uncertainties from NeRF models. Our approach is based on a meta-calibrator that takes as input features from the rendered NeRF images and uncalibrated uncertainty maps and predicts the calibration function, $R_{\boldsymbol{\theta}}(\cdot)$, for the NeRF model. Our meta-calibrator generalizes across scenes, so it only needs to be trained once, and can predict the calibration function in a single forward pass without any ground truth data from the target scene.

of detailed 3D representations from sets of 2D images, revolutionizing 3D reconstruction.

However, despite their impressive capabilities, traditional NeRF models lack an essential component: an accurate measure of uncertainty in their predictions. Accurate uncertainties are crucial for applying NeRFs to safety-critical problems such as MRI image reconstruction from sparse data [7], where unreliable confidence estimates could lead to misdiagnosis. More accurate uncertainties could also enhance practical methods such as uncertainty-guided next-best view planning techniques [9]. Prior approaches have attempted to estimate NeRF uncertainties [5, 14, 26, 27, 29], but they all overlook the problem of calibration. Thus, the uncertainties they output are not as accurate as they could be.

In particular, the state-of-the-art uncertainty estimation method for NeRFs, Density-aware NeRF Ensembles (DANE) [29], produces uncalibrated uncertainties. As a result, the confidence intervals and variances do not match the true confidences, meaning it has limited applicability to real-world problems. This constraint is significant as it restricts the use of NeRFs in safety-critical and sparse-data settings, where knowing the confidence in predictions is crucial.

The best NeRF methods for the sparse-view setting overlook the problem of calibration as well. FlipNeRF [24], the state-of-the-art technique for sparse-view reconstruction, uses uncertainties from an uncalibrated mixture of Laplacians to enhance its training process. Therefore, the uncertainties it outputs at inference as an artifact of training are not accurate.

In this paper, we present a novel approach for obtaining calibrated uncertainties for NeRF models in the sparse-view setting. Our strategy integrates the Laplacian mixture from FlipNeRF [24] with the calibration techniques by Kuleshov et al [12]. However, naively applying the calibration method by Kuleshov et al to FlipNeRF does not work due to a significant challenge of the sparse-view setting: there is a lack of held-out data from the target scene for fitting the calibrator. **Specifically, holding out just one image for calibration could decrease the size of the training set by over 30 %, resulting in significant performance degradation of the NeRF.** To overcome this, we make use of a unique observation: while calibration curves exhibit significant variation across scenes, they also demonstrate a significant regularity in their structure. Utilizing this insight, we propose the concept of a meta-calibrator that learns a low-dimensional representation of the NeRF calibration curves and infers them from scene features. We motivate and show why this meta-calibrator is necessary and demonstrate that it achieves more accurate uncertainties than DANE [29] without holding out any images from the target scene.

Specifically, our contributions are: (1) the first investigation into obtaining calibrated uncertainties from NeRFs, (2) a novel meta-calibrator for fitting the calibration model without using held-out data, and (3) experiments on the real-world Local Light Field Fusion (LLFF) [15] and DTU [8] datasets showing that our meta-calibrator achieves state-of-the-art and well-calibrated uncertainties for real scenes. We also demonstrate that our uncertainties can be leveraged for effective next-best view planning.

2 Related Work

Neural Radiance Fields (NeRFs). NeRFs [16] are a popular method for novel view synthesis. From a set of 2D images, NeRFs learn a neural network representation of a single scene. A trained NeRF model outputs estimates of the volume density and emitted radiance at any 3D location and viewing direction. Novel views can be generated by applying volume rendering [10] to the density and radiance values predicted by the NeRF model for points along rays cast into the scene. Due to their simplicity and impressive performance, NeRFs have become a popular technique for solving a variety of rendering problems.

Over the last few years, several extensions of NeRFs have been explored [11, 18, 21, 28, 31]. These include speeding up training and inference [13, 23, 30], modeling dynamic scenes [3], learning from a sparse set of training views [1, 6, 19, 24, 25], and estimating the uncertainty in NeRF predictions [5, 14, 22, 26, 27, 29]. Sparse NeRF methods aim to accurately render novel views when only a few training views are available from the target scene. NeRF uncertainty estimation techniques strive to accurately predict the confidence in the views rendered. Although uncertainty estimation is particularly important in the sparse-view setting, where NeRF renderings are especially unreliable, the main aim of sparse NeRF methods is not to output accurate uncertainties.

Sparse-view NeRF Methods. Despite this, recent sparse NeRF methods do produce uncertainties as an artifact of their training process. Both MixNeRF [25] and FlipNeRF [24], the state-of-the-art approach, model the RGB color channels given a ray as independent random variables that follow a mixture model. FlipNeRF further uses the uncertainties of the pixel colors to regularize the training process, producing more accurate image reconstructions at inference. However, neither MixNeRF nor FlipNeRF outputs calibrated uncertainties.

Our method significantly extends sparse NeRF methods to obtain more accurate and well-calibrated uncertainties at inference. To benefit from the superior performance of FlipNeRF [24] at sparse novel view synthesis, we apply the proposed meta-calibrator to the learned distribution from FlipNeRF, producing more accurate uncertainties without sacrificing state-of-the-art image quality. However, our approach can be applied to any NeRF method that outputs uncertainties, so it is distinct from FlipNeRF and MixNeRF. In essence, the proposed meta-calibrator augments sparse NeRF methods to achieve state-of-the-art uncertainty, beating both FlipNeRF and techniques designed explicitly for NeRF uncertainty estimation.

Uncertainty in NeRFs. The growing line of methods specifically designed for accurately estimating NeRF uncertainties [5,14,22,26,27,29] do not address the problem of calibration. The current state-of-the-art uncertainty estimation technique for NeRFs is Density-aware NeRF Ensembles (DANE) [29]. DANE adds an epistemic uncertainty term to a naive ensembles approach with five ensemble members. Thus, DANE is very costly as it requires training five NeRFs to obtain uncertainty estimates. Another NeRF uncertainty estimation approach is Stochastic Neural Radiance Fields (S-NeRF) [27], which learns a probability distribution over all possible radiance fields by modeling the volume density and radiance as random variables that follow a joint distribution. S-NeRF employs variational inference to sample from an approximation to this distribution and uses the variances of the sampled pixel colors as the estimated uncertainties. Conditional-flow NeRF (CF-NeRF) [26] builds on S-NeRF by combining latent variable modeling and conditional normalizing flows to relax the strong constraints S-NeRF imposes over the radiance distribution. Despite the growing number of techniques in this area of study, all prior work does not consider calibration, outputting unreliable uncertainties as a result. Our work achieves more accurate uncertainties than these prior methods by filling the gap of uncertainty calibration for NeRFs and drawing on well-established techniques in calibrated regression [12].

Uncertainty Calibration in Deep Learning. While uncertainty calibration has been studied for Bayesian deep learning methods [4, 12], it has not been adapted for or applied successfully to NeRF uncertainty estimates. This may be because NeRFs introduce additional complexity in the uncertainty estimation process as the neural network model needs to be trained *per-scene*.

This makes uncertainty calibration challenging as methods for calibrated regression [12] require either using the training set or held-out data to achieve calibrated uncertainties. Using the training set to fit the calibrator results in severe overfitting (see Sec. 2 of the Supplementary). Holding out data from the target scene for calibration means there is less data to train the NeRF, making it more inaccurate at novel view synthesis (see Sec. 3 of the Supplementary). Thus, a trivial application of [12] to NeRFs would not be satisfactory. In this work, we propose the concept of a meta-calibrator that, in contrast to [12], does not require held-out data and achieves calibrated uncertainty estimates for NeRFs.

3 Method

In this paper, we present a method that calibrates NeRF uncertainties. To this end, we propose a novel meta-calibrator that accepts uncalibrated NeRF uncertainties and predicted images as inputs and outputs a scene-specific calibration curve, correcting the uncalibrated confidence levels. An overview of our method can be seen in Fig. 1. Crucially, our approach does not require holding out any images from the target scene. Thus, it can be applied to sparse-view settings, where holding out a single image could significantly harm the NeRF's performance. In Sec. 3.1, we explain the necessary background concepts, in Sec. 3.2, we describe how we obtain the corrected uncertainty values, and in Sec. 3.3 we detail our meta-calibrator.

3.1 Preliminaries

Neural Radiance Fields (NeRFs). Neural Radiance Fields (NeRFs) [16] represent a scene as a continuous vector-valued function with inputs a Cartesian point $\mathbf{x} = (x, y, z)$ and unit viewing direction vector $\mathbf{d} = (u, v, w)$ and outputs an emitted radiance $\boldsymbol{\mathcal{c}} = (\mathcal{r}, \mathcal{g}, \mathcal{b})$ and volume density σ. By optimizing the weights $\boldsymbol{\Theta}$ of a neural network approximation $\mathbf{F}_{\boldsymbol{\Theta}}$ to this representation, NeRFs can render the color of any pixel in a synthetic image of the scene. To achieve this, principles from classical volume rendering [10] are applied to the radiance and density values estimated by $\mathbf{F}_{\boldsymbol{\Theta}}$ for points along a ray cast from the origin $\mathbf{x_0}$ of the virtual camera, through the pixel, and into the scene. More specifically, the expected color $\mathbf{c}(\mathbf{r})$ of a camera ray $\mathbf{r}(t) = \mathbf{x_0} + t\mathbf{d}$ with near and far bounds t_n and t_f is:

$$\mathbf{c}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\boldsymbol{\mathcal{c}}(\mathbf{r}(t), \mathbf{d})dt, \tag{1}$$

where $T(t) = e^{-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds}$. The integral in Eq. (1) is estimated with numerical quadrature to obtain the colors of the pixels in the synthetic image from the NeRF model outputs. NeRF optimizes $\boldsymbol{\Theta}$ according to Eq. (1) with gradient descent, since the numerical quadrature is differentiable. Conventional NeRFs do not provide an uncertainty associated with their predictions, so extensions like DANE [29] have been developed.

Base NeRF Uncertainties. To obtain the initial uncertainties, we have two options for our base model: (1) DANE [29], the state-of-the-art method for NeRF uncertainty estimation or (2) FlipNeRF [24], the state-of-the-art method for sparse novel view synthesis. DANE requires training five NeRFs per scene and provides poor image quality in the sparse-view setting, so we choose FlipNeRF. We show in the experiments that applying our meta-calibrator to FlipNeRF results in more accurate and well-calibrated uncertainties than those output by DANE. Our base FlipNeRF uncertainties are inferred from a mixture of Laplacians with location and scale parameters learned during training.

FlipNeRF [24] models the joint distribution of the color $\mathbf{C} = (R, G, B)$ given a ray $\mathbf{r}$ with a mixture of Laplacians:

$$p(\mathbf{C} = \mathbf{c}|\mathbf{r}) = \sum_{j=1}^{M} \pi_j \mathcal{F}(\mathbf{C} = \mathbf{c}; \boldsymbol{\mu_j}, \boldsymbol{\beta_j}), \tag{2}$$

where M is the number of sampled points along the ray $\mathbf{r}$. $\mathcal{F}(\mathbf{C} = \mathbf{c}; \boldsymbol{\mu_j}, \boldsymbol{\beta_j})$ is the 3D Laplacian probability density with location parameter $\boldsymbol{\mu_j} = (\mu_j^R, \mu_j^G, \mu_j^B)$ and scale parameter $\boldsymbol{\beta_i} = (\beta_j^R, \beta_j^G, \beta_j^B)$ evaluated at the color $\mathbf{c}$. More specifically, the mixing coefficients $\{\pi_j\}_{j=1}^{M}$ are the normalized coefficients of the radiance values along a ray in Eq. (2), the location parameters $\{\boldsymbol{\mu_j}\}_{j=1}^{M}$ are the estimated RGB radiance values, and the scale parameters $\{\boldsymbol{\beta_j}\}_{j=1}^{M}$ are an additional output of the model. These parameters are optimized by FlipNeRF during training.

We now go into detail about how the location and scale parameters are obtained. FlipNeRF [24] learns these parameters by minimizing the negative log-likelihood of the training set $\mathcal{D} = \{(\mathbf{r_i}, \mathbf{c_i})\}_{i=1}^{N}$ containing the rays $\mathbf{r_i}$ and colors $\mathbf{c_i}$ from the pixels in the ground truth images of the scene assuming the distribution in Eq. (2). FlipNeRF additionally minimizes the average scale parameters through an auxiliary uncertainty-aware emptiness loss for reducing floating artifacts. The negative log-likelihood loss and the uncertainty-aware emptiness loss are added to FlipNeRF's total training loss, which incorporates other terms such as the mean squared error. As a result of this training process, the location and scale parameters can be inferred by FlipNeRF at new poses. We obtain the uncalibrated confidence levels of predicted ray colors from these location and scale parameters.

Using the distribution in Eq. (2), we can easily compute the confidence level for a given ray $\mathbf{r_t}$, which we denote as F_t. The CDF for the Laplacian mixture has a closed form expression parameterized by the location and scales output by the pretrained FlipNeRF [24]. Thus, we can use this CDF to predict the confidence level of the ground truth color $\mathbf{c_t}$ of any ray by evaluating it at the given color value, $p_t = F_t(\mathbf{c_t})$. However, these initial confidence levels are uncalibrated and, hence, inaccurate.

Calibrated regression. In [12], Kuleshov et al extend calibration methods for classification to regression. They define a forecaster $H : \mathcal{X} \to (\mathcal{Y} \to [0,1])$ as a function that outputs for each $x_t \in \mathcal{X}$, a CDF F_t. Given a pretrained forecaster H, they suggest training an auxiliary model $R : [0,1] \to [0,1]$ by fitting R to a recalibration dataset $D = \left\{ \left([H(x_t)](y_t), \hat{P}([H(x_t)](y_t)) \right) \right\}_{t=1}^{T}$, where

$$\hat{P}(p) = |\{y_t : [H(x_t)](y_t) \leq p \text{ for } t = 1, \ldots, T\}|/T$$

is the empirical confidence level corresponding to the predicted confidence level p. The fitted R forms the calibration curve that corrects the expected confidence levels. We can now obtain predictive posterior values that closely match the true confidences using $\hat{F}_t \equiv R \circ F_t$ for the test data.

However, as mentioned before, there are numerous challenges associated with applying this recalibration procedure. Firstly, note that it requires ground-truth values (y_t) for the predictions we are recalibrating. This means that in order to prevent overfitting we need to reserve part of the training dataset specifically for fitting the calibrator (which leaves less data for training the NeRF — a significant issue if we only have a few input views). Secondly, it does not actually prescribe how to compute a suitable uncertainty value from the predicted distribution. In Sec. 3.3, we address the former issue, and in Sec. 3.2, we address the latter.

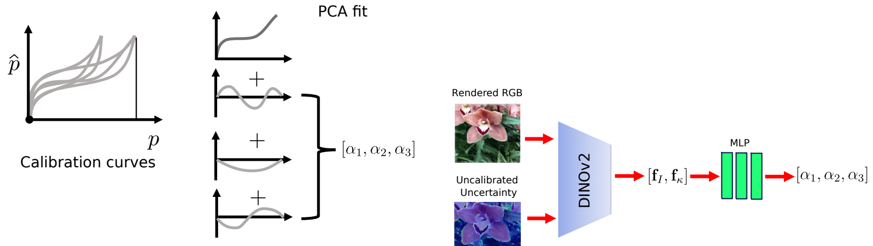
3.2 Calculating uncertainty

The predictive posterior in Eq. (2) provides a distribution over likely ray colors for a NeRF model, but it does not inherently offer a straightforward metric for quantifying uncertainty at a specific point in the reconstruction. Intuitively, as the variance of this distribution increases, so does the uncertainty of the model's output at that point. Therefore, the variance or standard deviation is a popular choice for quantifying the uncertainty [24,26,27,29]. However, in the case of the corrected mixture distribution obtained from raymarching, this can be slow to compute especially if it has to be done for each pixel. Therefore, we turn to a metric that can be calculated directly from the calibrated CDF $\hat{F}^t$.

We propose to use the interquartile range of each calibrated distribution:

$$\kappa^C(\mathbf{r_t}) = [\hat{F}_t^C]^{-1}\left(\frac{3}{4}\right) - [\hat{F}_t^C]^{-1}\left(\frac{1}{4}\right), \tag{3}$$

where $\kappa^C(\mathbf{r_t})$ represents the uncertainty at a ray $\mathbf{r_t}$ in the color channel C. This difference provides a measure of the statistical dispersion and thus serves


PCA fit
$\hat{p}$
p
Calibration curves
+
+
+
$[\alpha_1, \alpha_2, \alpha_3]$
Rendered RGB
Uncalibrated Uncertainty
DINOv2
$[\mathbf{f}_I, \mathbf{f}_\kappa]$
MLP
$[\alpha_1, \alpha_2, \alpha_3]$

(a) Creating a parametric model of the calibration curves.

(b) Predicting the curve parameters.

Fig. 2: Meta-calibrator design. In stage (a) we fit a low-dimensional parameteric model of the calibration curves. The meta-calibrator then predicts these curve parameters from rendered images of the scene and their associated uncalibrated uncertainty maps (b).

as a robust measure of the spread of the output channel. By averaging the interquartile range over the color channels, we obtain a single scalar value that effectively quantifies the uncertainty of the NeRF model for the given ray. As shown in our experiments, this method is very computational efficient, and it provides an accurate estimate of uncertainty.

3.3 Meta-calibrator

To overcome the challenge that, especially in the sparse-view setting, there is no held-out data available for fitting the calibrator, we propose a novel meta-calibrator that infers the calibration curves from uncalibrated NeRF predictions. To do this, we leverage the insight that the calibration curves demonstrate significant regularity. We posit a low-dimensional model of the calibration curves can be learned and predicted using the images and uncalibrated uncertainty maps inferred by the NeRF, enabling us to estimate the calibration function *without evaluating the empirical confidence levels using held-out data from the target scene*. We now describe this meta-calibrator (illustrated in Fig. 2) in detail.

A Parametric Model for Calibration Curves. We first fit a low-dimensional representation of the calibration curves using Principal Component Analysis (PCA). To create the training set for learning this representation, we sample held-out images from K scenes and apply the calibration procedure by Kuleshov et al. [12] to form K ground truth calibration curves. To construct the training vector $\mathbf{v_k} \in \mathbb{R}^{1\times M}$ for scene $k \leq K$, we sample M evenly spaced points along its ground truth calibration curve. We find that fitting the PCA model using only a few scenes (21 in our case) provides a good enough approximation to capture the variation in the test curves (see Sec. 4). Here, $\mathbf{V} = [\mathbf{v}_k] \in \mathbb{R}^{K\times M}$ contains the ground truth calibration curves for the training scenes, with K representing the number of curves and M the sample count along each curve. PCA is then used to determine the basis vectors $\mathbf{U} = (\mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_n})$ and coefficients $\boldsymbol{\theta} = (\alpha_1, \alpha_2, ..., \alpha_n)$, so that each calibration curve can be represented as:

$\mathbf{v}_k = \sum_{i=1}^{n} \alpha_i \mathbf{u}_i$. The parameters $\boldsymbol{\theta}$ fully describe the calibration functions.

To find the optimal number of components, we compute the explained variance, and find that in our case most of the variance is explained using only $n = 3$ components (see experiments). To ensure the calibrator is monotonically increasing, we derive the final calibration function $R_{\boldsymbol{\theta}}(\cdot)$ using isotonic regression applied to the curve approximated by $\boldsymbol{\theta} \cdot \mathbf{U}$. The idea is that the low-dimensional representation of the calibration curves encoded in $\mathbf{U}$ will generalise to new target scenes without any additional scene-specific data.

Predicting Calibration Parameters. What remains is to estimate the calibration parameters, $\boldsymbol{\theta}$, for a new scene. As we do not want to use additional held-out data from the target scene, we propose predicting these parameters using scene-specific features computed from the pretrained NeRF outputs. This approach is motivated by the human ability to visually identify inaccuracies in the renderings such as floaters and unnatural artifacts. Specifically, we use a Multi-Layer Perceptron (MLP) with three layers of output size: [128, 128, 3] and Leaky ReLU activations throughout except the last layer as the meta-calibrator to estimate $\boldsymbol{\theta}$ given features extracted by the DINOv2 model [20] from rendered images ($\mathbf{f}_I$) and uncalibrated uncertainty maps ($\mathbf{f}_\kappa$). The goal here is to have DINOv2 extract features that describe the rendering imperfections, correlating with the calibration curve. We find that training the MLP model on only a few scenes (30 in our case) allows it to generalize well to new test scenes. Once trained, the meta-calibrator can predict the calibration curve of a new target scene as: $\boldsymbol{\theta} = MLP([\mathbf{f}_I, \mathbf{f}_\kappa])$, without using any additional ground truth data.

In summary, the meta-calibrator can correct the confidence levels of the model without requiring ground truth data at any stage, suggesting potential enhancements to applications that rely on uncertainty such as next-best view selection (see Sec. 4.3).

4 Experiments

The objective of the experiments is to: 1) validate that our approach achieves more accurate uncertainties (lower negative log-likelihood and calibration error) than state-of-the-art approaches for NeRF uncertainty estimation (Sec. 4.1); 2) demonstrate the meta-calibrator improves the accuracy of the uncalibrated uncertainties (decreases both the negative log-likelihood and calibration error) without requiring any held-out data from the target scene (Sec. 4.2); 3) explain the motivation for certain meta-calibrator design decisions; and 4) show that our uncertainties can be leveraged for applications such as next-best view planning (Sec. 4.3).

For additional results showing: 1) why the PCA representation of the calibration curves is necessary; 2) that using the training set results in severe overfitting; 3) that holding out data results in poor performance at image reconstruction; 4) the influence of the number of samples along the ray on the uncertainty quality; and 5) the efficiency of our uncertainty metric (Eq. (3)) over other approaches, please refer to Sections 1, 2, 3, 4, and 6 of the Supplementary respectively.

Metrics and calibration curves. We use a variant of the calibration error from [12] to evaluate the effectiveness of the meta-calibrator. Specifically, given a test set $\mathcal{D} = \{(\mathbf{r_t}, \mathbf{c_t})\}_{t=1}^{T} = \{(\mathbf{r_t}, (r_t, g_t, b_t))\}_{t=1}^{T}$, we report:

$$ERR = \frac{1}{T}\sum_{t=1}^{T}(p_t - \hat{P}(p_t))^2, \tag{4}$$

where p_t is the expected confidence level for data point $(\mathbf{r_t}, c_t)$, and $\hat{P}(p_t)$ is the empirical frequency of data points within that confidence level. More specifically, for each $t \in \{1, \ldots, T\}$, we set $p_t = M_t^C(c_t)$ and,

$$\hat{P}(p) = |\{c_t : M_t^C(c_t) \leq p \text{ for } t = 1, \ldots, T\}|/T, \tag{5}$$

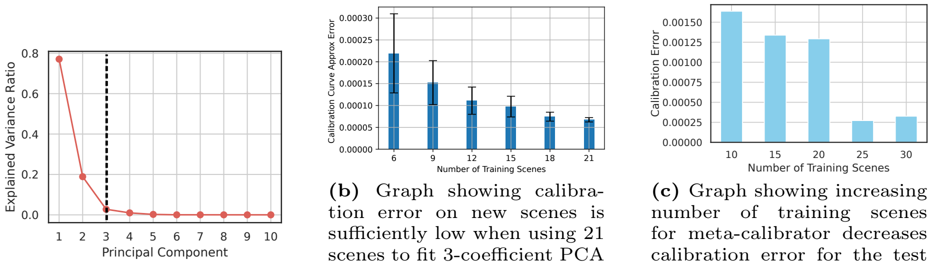
where $M \equiv F$ for uncalibrated errors and $M \equiv \hat{F}$ for calibrated errors, and $C \in \{R, G, B\}$. Note that this formulation of the calibration error is equivalent to the one in [12] with a confidence level for every unique p_t and weights that more significantly penalize errors from frequently predicted confidence levels.

Following [12], we plot $\{(p_t, \hat{P}(p_t))\}_{t=1}^{T}$ before and after calibration for each color channel to generate calibration curves. A perfectly calibrated forecaster would produce the straight line $p_t = \hat{P}(p_t)$ as each expected confidence level would equal the empirical one. Intuitively, our version of the calibration error is the mean squared vertical distance of points on the calibration curve from a perfectly straight line. If an expected confidence level occurs N times in the test data, its distance is counted N times in the mean. Following [29], we additionally report the negative log-likelihood (NLL) of the test data averaged across all scenes. Following [24], we include PSNR and LPIPS [32] to evaluate image quality.

Datasets. We use 30 scenes from 3 datasets: Realistic Synthetic 360° [16], the subset of scenes in DTU [8] used in [24], and LLFF [15] for training the meta-calibrator and test it on a hold-out scene from either LLFF or DTU to show it generalizes to new target scenes.

Baselines We compare our approach against the state-of-the-art method for NeRF uncertainty estimation DANE [29] as well as other methods in Sec. 4.1. In Tab. 1, following [29], we implement the naive ensembles approach and DANE using a public implementation of Instant-NGP [2,17] and 5 ensemble members. In Sec. 4.2, we compare the uncalibrated uncertainties to the uncertainties calibrated by our meta-calibrator.

Meta-calibrator Design The results guiding our decisions to use 3 Principal Component Analysis (PCA) components to represent the calibration curves, fit the PCA components using 21 training scenes, and train the meta-calibrator on 30 scenes to predict the PCA coefficients are shown in Fig. 3.



(a) Explained variance plot showing most of the variance of the calibration curves is explained using only 3 PCA components.

(b) Graph showing calibration error on new scenes is sufficiently low when using 21 scenes to fit 3-coefficient PCA model (i.e. it's an order of magnitude lower than we can expect from the final calibration, meaning good generalization to unseen scenes).

(c) Graph showing increasing number of training scenes for meta-calibrator decreases calibration error for the test scene *T-Rex* from LLFF [15], with 30 scenes resulting in good generalization to the held-out scene and well-calibrated uncertainties.

Fig. 3: Meta-calibrator design decisions. Results showing using 3 components for Principal Component Analysis (PCA) model of calibration curves, 21 scenes to fit PCA model, and 30 scenes to train meta-calibrator achieves good generalization to new test scenes.

Table 1: Quantitative results on standard sparse NeRF benchmark. Our proposed approach results in significantly better uncertainties and image quality than the state-of-the-art NeRF uncertainty estimation method DANE [29] does on the challenging 3-view LLFF [15] dataset. Specifically, our meta-calibrator reduces the calibration error to 6% of DANE's calibration error and the negative log-likelihood to be over 100 % lower than DANE's. Note: ***lower calibration error (Cal. Err.) and negative log-likelihood (NLL) values indicate more accurate uncertainties.*** Results are averaged over all 8 scenes in LLFF.

	Uncertainty		Image Quality	
	Cal. Err. (↓)	NLL (↓)	PSNR (↑)	LPIPS (↓)
Naïve Ens.	0.0505	4.39	15.19	0.646
DANE [29]	0.0441	3.75	15.19	0.646
Ours	**0.0026**	**-0.68**	**19.34**	**0.235**

4.1 Comparison to State-of-the-art

In this section, we compare our approach to prior methods for NeRF uncertainty estimation. We achieve more accurate uncertainties (94 % reduction in calibration error and over 100 % reduction in negative log-likelihood) than those estimated by DANE [29], the state-of-the-art method. These results are shown in Tab. 1 for the challenging 3-view LLFF [15] dataset from prior work on sparse novel view synthesis [19,24,25] and Tab. 2 for the less challenging version of LLFF from prior work on NeRF uncertainty estimation [14, 26, 27, 29]. In Fig. 5a, we compare the calibration curves from our approach to DANE's, illustrating that the meta-calibrator predicts expected confidences that match the true ones while DANE does not.

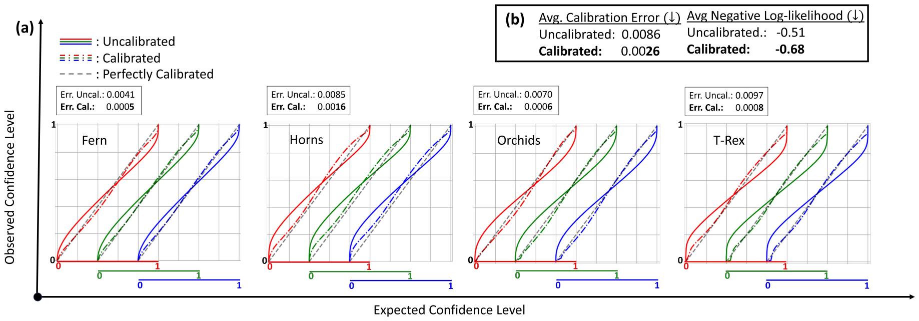


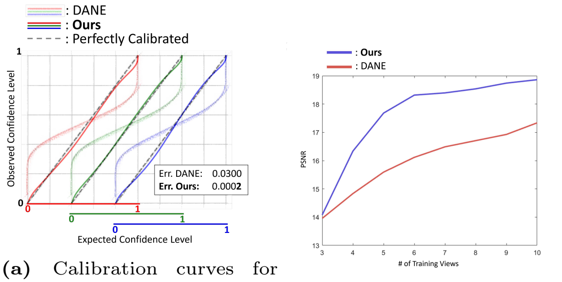
Fig. 4: Quantitative comparison of uncalibrated and calibrated uncertainties. In (a), we show calibration curves on test data from four scenes in LLFF [15]. The color of each curve indicates the color channel it corresponds to. The calibrated curves are much closer to the ideal calibration (dashed lines), demonstrating that the meta-calibrator works very well. In (b), the average calibration error and negative log-likelihood before and after calibration are reported for LLFF, clearly showing the meta-calibrator improves the accuracy of the uncertainties (lowering calibration error and negative log-likelihood). To test generalization, the meta-calibrator was also applied to held-out scenes in DTU [8], achieving a 70 % reduction in calibration error on average.

4.2 Comparison to Uncalibrated Uncertainties

In this section, we compare our uncalibrated base NeRF uncertainties to our calibrated uncertainties obtained from applying the proposed meta-calibrator. In Fig. 6 we show that the calibrated uncertainties better highlight floaters and other errors in the NeRF renderings. In Fig. 4, we show that the meta-calibrator predicts expected confidences that closely match the true ones, lowering both the calibration error and the negative log-likelihood of the uncalibrated uncertainties.

Table 2: Quantitative results on standard NeRF uncertainty estimation benchmark. Here, we present results on the LLFF [15] dataset used in prior work [14, 26, 27, 29] on uncertainty estimation for NeRFs. This dataset is less challenging than the one in Tab. 1 since 4-12 views are used for training instead of 3. Our proposed approach results in significantly better uncertainties than prior methods for NeRF uncertainty estimation on all 8 scenes in LLFF. Note: ***lower negative log-likelihood values indicate more accurate uncertainties.*** M indicates the number of ensemble members, and MC-DO refers to Monte Carlo Dropout sampling with M sample configurations. This table is Tab. 1 from [29] with our results added as an additional column. Please refer to [29] for further details.

		Negative Log-likelihood (↓)							
Scene	# of Train. Views	MC-DO $M = 5$	Naïve Ens. $M = 5$	NeRF-W [14]	S-NeRF [27]	CF-NeRF [26]	DANE [29] $M = 5$	DANE [29] $M = 10$	**Ours**
Fern	4	4.90	2.47	2.16	2.01	—	-0.98	-1.00	**-1.41**
Orchids	5	5.74	2.23	2.24	1.95	—	-0.28	-0.31	**-0.84**
Leaves	5	2.72	2.66	0.79	0.68	—	0.97	0.73	**-1.19**
Flower	7	4.63	1.63	1.71	1.27	—	1.00	0.85	**-2.05**
Fortress	8	5.19	2.29	1.04	-0.03	—	-1.30	-1.30	**-1.99**
Room	8	5.06	2.13	4.93	2.35	—	-1.35	-1.35	**-2.17**
T-Rex	11	4.10	2.28	1.91	1.37	—	-0.31	-0.69	**-1.49**
Horns	12	4.18	2.17	0.78	0.60	—	-0.55	-0.66	**-2.18**
Avg.		4.57	2.23	1.95	1.27	0.57	-0.35	-0.47	**-1.67**



(a) Calibration curves for the *Fern* scene from 3-view LLFF [15].

(b) Next-best View Planning on *Horns* from LLFF [15].

Fig. 5: Comparison to DANE [29]. Results comparing our uncertainties to those from the state-of-the-art method DANE. In (a) we show DANE's RGB calibration curves are not closely aligned with the perfectly calibrated lines, meaning it is miscalibrated. It is significantly over-confident for expected confidence levels close to 1 and under-confident for confidence levels close to 0. In comparison, the curves for our approach are extremely close to the ideal calibration (dashed lines), demonstrating that the meta-calibrator works very well, predicting expected confidences that match the true ones. This is also verified by how our calibration error is over two orders of magnitude smaller than DANE's. In (b) we show that our approach results in more efficient performance gains over DANE for next-best view planning.

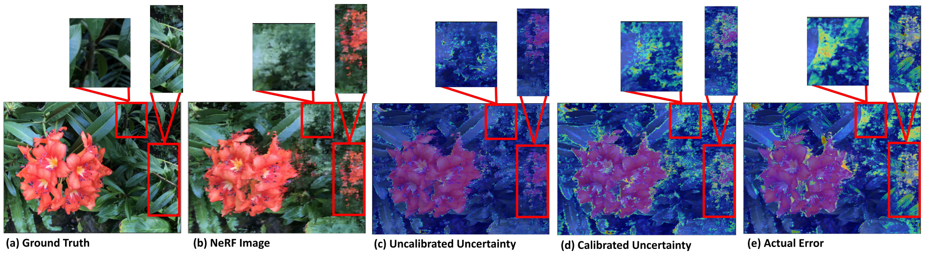


Fig. 6: Qualitative comparison of uncalibrated and calibrated uncertainties from the *Flower* scene in LLFF [15]. The calibrated uncertainties (d) clearly detect incorrect regions (indicated by the red boxes) better than the uncalibrated uncertainties (c) do. This is apparent by noting that (d) and (e) look more similar than (c) and (e).

4.3 Application: Next-best View Planning

In this section, we show that our uncertainties can be leveraged for next-best view planning. Specifically, we start by training the NeRF model for 2000 iterations on a training set of three images. The next-best view is selected by evaluating the average calibrated pixel uncertainty (obtained using the meta-calibrator) for each of the candidate views, and the view with the highest uncertainty is added to the training set. The average PSNR of the test images is reported after each training iteration. In Fig. 5b we show that using our approach results in greater performance gains (higher PSNRs) than DANE [29] does. In the Fig. 1 of the Supplementary, we also compare the information gain from rays selected according to the highest calibrated uncertainties to the information gain from rays selected according to the highest uncalibrated uncertainties and show our meta-calibrator produces higher average PSNRs on the test set for scenes in DTU [8] than the uncalibrated uncertainties do. Thus, calibration specifically re-orders the pixel uncertainties so that rays more likely to raise the PSNR are picked earlier in next-best view planning. In Sec. 5 of the Supplementary, we include a detailed theoretical example showing that such re-ordering is possible.

5 Conclusion

In this paper we addressed the open problem of obtaining calibrated uncertainties from NeRF models. We introduce the concept of a meta-calibrator that infers the calibration curves from scene features, and using this approach achieve state-of-the-art uncertainty without holding out any ground truth data from the target scene. By enabling efficient and accurate calibration of NeRF models without relying on additional data, our method represents a significant step forward in the practical application of NeRF to real-world scenarios and opens up new avenues for the use of NeRF in situations where data is limited and uncertainty is critical.

Acknowledgements. The authors would like to thank Seunghyeon Seo for his support on FlipNeRF, and Oishi Deb for her insightful feedback on uncertainty estimation. Niki Amini-Naieni is funded by an AWS Studentship, the Reuben Foundation, and the AIMS CDT program at the University of Oxford. Tomas Jakab is sponsored by EPSRC VisualAI EP/T028572/1 and Andrea Vedaldi by ERC-CoG UNION 101001212.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV. pp. 5835–5844 (2021)
2. Bhalgat, Y.: Hashnerf-pytorch. `https://github.com/yashbhalgat/HashNeRF-pytorch/` (2022)
3. Gafni, G., Thies, J., Zollhöfer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: CVPR. pp. 8649–8658 (2021)
4. Ghoshal, B., Tucker, A.: On calibrated model uncertainty in deep learning. In: ECML (2022)
5. Goli, L., Reading, C., Sellán, S., Jacobson, A., Tagliasacchi, A.: Bayes' Rays: Uncertainty quantification in neural radiance fields. ArXiv **abs/2309.03185** (2023)
6. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: ICCV. pp. 5865–5874 (2021)
7. Jang, T.J., Hyun, C.M.: Nerf solves undersampled mri reconstruction. ArXiv **abs/2402.13226** (2024)
8. Jensen, R.R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: CVPR (2014)
9. Jin, L., Chen, X., Ruckin, J., Popovi'c, M.: Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. In: IROS (2023)
10. Kajiya, J.T., Von Herzen, B.P.: Ray tracing volume densities. In: SIGGRAPH. p. 165–174 (1984)
11. Kosiorek, A.R., Strathmann, H., Zoran, D., Moreno, P., Schneider, R., Mokr'a, S., Rezende, D.J.: Nerf-vae: A geometry aware 3d scene generative model. In: ICML (2021)
12. Kuleshov, V., Fenner, N., Ermon, S.: Accurate uncertainties for deep learning using calibrated regression. In: ICML. pp. 2796–2804 (2018)
13. Liu, L., Gu, J., Lin, K.Z., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. In: NIPS (2020)
14. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: CVPR (2021)
15. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. In: TOG (2019)
16. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
17. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. In: ACM Trans. Graph. (2022)

18. Neff, T., Stadlbauer, P., Parger, M., Kurz, A., Mueller, J.H., Chaitanya, C.R.A., Kaplanyan, A., Steinberger, M.: Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In: CGF. pp. 45–59 (2021)
19. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S.M., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: CVPR (2022)
20. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023)
21. Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Q., Zhou, X., Bao, H.: Animatable neural radiance fields for modeling dynamic human bodies. In: ICCV (2021)
22. Ran, Y., Zeng, J., He, S., Li, L., Chen, Y., Lee, G.H., Chen, J., Ye, Q.: Neurar: Neural uncertainty for autonomous 3d reconstruction. In: RAL (2023)
23. Rebain, D., Jiang, W., Yazdani, S., Li, K., Yi, K.M., Tagliasacchi, A.: Derf: Decomposed radiance fields. In: CVPR. pp. 14148–14156 (2020)
24. Seo, S., Chang, Y., Kwak, N.: Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In: ICCV (2023)
25. Seo, S., Han, D., Chang, Y., Kwak, N.: Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs. In: CVPR. pp. 20659–20668 (2023)
26. Shen, J., Agudo, A., Moreno-Noguer, F., Ruiz, A.: Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In: ECCV (2022)
27. Shen, J., Ruiz, A., Agudo, A., Moreno-Noguer, F.: Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. In: 3DV. pp. 972–981 (2021)
28. Sitzmann, V., Martel, J.N., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: NIPS (2020)
29. Sünderhauf, N., Abou-Chakra, J., Miller, D.: Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In: ICRA (2023)
30. Wang, P., Liu, Y., Chen, Z., Liu, L., Liu, Z., Komura, T., Theobalt, C., Wang, W.: F2-nerf: Fast neural radiance field training with free camera trajectories. In: CVPR (2023)
31. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. ArXiv **abs/2010.07492** (2020)
32. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)