2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Density-aware NeRF Ensembles: Quantifying Predictive Uncertainty in Neural Radiance Fields

Niko Sünderhauf, Jad Abou-Chakra, Dimity Miller

***Abstract*— We show that *ensembling* effectively quantifies model uncertainty in Neural Radiance Fields (NeRFs) if a density-aware epistemic uncertainty term is considered. The naive ensembles investigated in prior work simply average rendered RGB images to quantify the model uncertainty caused by conflicting explanations of the observed scene. In contrast, we additionally consider the termination probabilities along individual rays to identify epistemic model uncertainty due to a lack of knowledge about the parts of a scene *unobserved* during training. We achieve new state-of-the-art performance across established uncertainty quantification benchmarks for NeRFs, outperforming methods that require complex changes to the NeRF architecture and training regime. We furthermore demonstrate that NeRF uncertainty can be utilised for next-best view selection and model refinement.**

I. Introduction

Neural Radiance Fields [1] (NeRFs) implicitly represent the geometry and appearance of complex 3D scenes as a continuous function that is implemented as a relatively simple deep neural network. NeRFs and other implicit representations were met with an immense interest in the past two years, leading to an often-quoted "explosion" of work in this area [2]. While most of this work has been conducted by the computer graphics and vision communities, researchers in robotics have quickly started to explore possible use cases of NeRFs for important robotics tasks such as navigation [3], SLAM [4], [5], or manipulation [6]–[8]. Since the appearance of highly efficient NeRFs, such as Instant-NGP [9], that can be trained in seconds rather than many hours, the adoption of NeRFs for robotics has become palpable.

NeRFs provide an interesting new take on the long-standing problem of how to represent the 3D world for robotics, with all its geometric and semantic complexity [10]–[12]. However, NeRFs face the same challenges as other deep learning approaches in robotics [13] – they lack the ability to express uncertainty in their predictions.

The incorporation of uncertainty and the generally probabilistic nature of data, estimations, and predictions is well-established in large parts of the robotics literature [14], yet it is still a highly active area of research in deep learning [15]. Approaches range from principled Bayesian Deep Learning [16], [17] to simple but surprisingly effective approximate methods such as MC Dropout [18] or Deep Ensembles [19]. Despite the large body of work in NeRFs, little research has investigated adopting the above methods to quantify the predictive uncertainty of NeRFs.

The authors are with Queensland University of Technology (QUT) in Brisbane, Australia, and acknowledge the ongoing support of the QUT Centre for Robotics. Contact: niko.suenderhauf@qut.edu.au

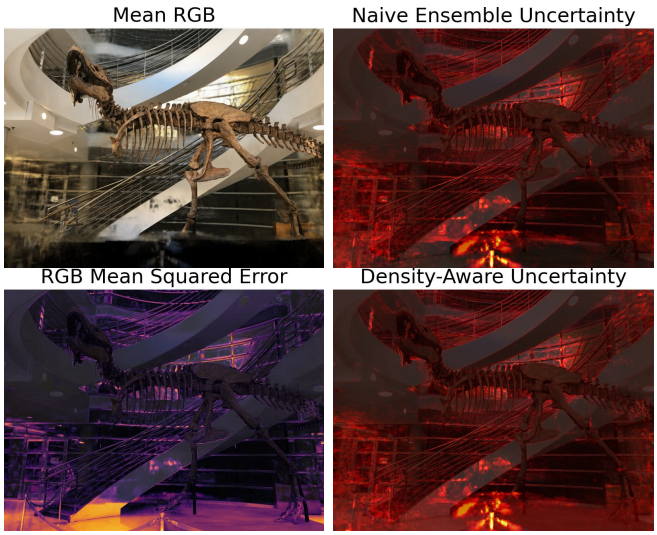

Fig. 1: An ensemble of Neural Radiance Fields can render a mean RGB image and use the colour variance in pixel space to quantify its predictive uncertainty (top). However, this naive approach to ensembling often does not capture the epistemic uncertainty in parts of a scene that were *unobserved* during training. In our example, the ensemble agrees to render the unobserved bottom portion of the images in black, resulting in negligible uncertainty, despite the high error (bottom left) when compared to the ground truth. We show that an epistemic uncertainty term that captures the termination probabilities along each ray must be considered in addition to the RGB variance to make ensembling an effective approach to quantifying uncertainty in Neural Radiance Fields (bottom right).

Our paper addresses this gap. Specifically, we show that *ensembling* [19] is an effective approach to quantify uncertainty in Neural Radiance Fields and that prior work has dismissed the ensembles approach prematurely [20], [21]. Our key insight is that instead of only averaging and calculating variance in the RGB space, a NeRF ensemble should consider an additional epistemic uncertainty [22] term that depends on the densities and termination probabilities along individual rays. We show that such a *density-aware* ensemble of Instant-NGP NeRFs [9] achieves new state-of-the-art performance in terms of uncertainty quality, and can furthermore effectively select the next-best view when iteratively building a training dataset and refining an implicit object model.

II. Related Work

A. Neural Radiance Fields

Neural Radiance Fields [1] learn a radiance and density field that – in combination with a volumetric rendering process – can explain a set of posed training images and realistically

render novel views. Many variations of the original NeRF have emerged [23], including follow-up works [9], [24]–[27] that highlight the impact of the input encoding on training and inference speed. Instant-NGP [9] is one of the fastest and most optimised formulations, using a parametric encoding alongside a smaller MLP. It trains in seconds and renders at real-time rates, making it a great candidate for adoption in robotics, and thus is the backbone chosen for our work. NeRFs provide an exciting new approach to scene representation in robotics, and a variety of use cases are currently explored. For example, NiceSLAM [5] and iMap [4] use a NeRF to represent a map within a SLAM system, while [6], [28] use it as a decoder within an autoencoder framework to learn an alternative representation for planning and reinforcement learning, and [7], [8] use NeRFs for data augmentation.

B. Uncertainty Quantification in Deep Learning

Uncertainty Quantification in Deep Learning is a rapidly growing field and we refer the reader to [15] for a recent review. Kendall et al. [22] identified two relevant types of uncertainty for computer vision – aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty refers to uncertainty in the input data, introduced by noise or random processes [22]. In contrast, epistemic uncertainty refers to uncertainty in the model's learnt parameters, representing the model's lack of knowledge due to a finite training dataset [22]. Deep Ensembles is a popular approach for uncertainty quantification, producing predictive uncertainty that captures both aleatoric and epistemic uncertainty [19].

C. Uncertainty Quantification for NeRFs

Stochastic NeRF (S-NeRF) [20] is one of the few papers investigating uncertainty quantification for NeRFs. It reformulates the NeRF optimisation as a Bayesian estimation problem, and applies a Variational Inference approach to effectively estimate the posterior distribution over the parameters of all possible radiance fields given the observed training data. Mean and variance for each rendered pixel can be calculated by sampling from the approximated posterior distribution over all radiance fields. The variance in pixel space is then used as the uncertainty measure for a particular pixel. Conditional-Flow NeRF (CF-NeRF) [21] by the same authors builds on this work and relaxes some of S-NeRF's constraints on the involved distributions, especially the independence assumption between radiance and density.

Both methods [20], [21] involve a complex reformulation of the NeRF architecture, rendering process, and its training regime, limiting the applicability of the proposed approach to other NeRF implementations such as Instant NGP [9]. In contrast, the ensembles approach we propose here is extremely simple to implement as it does not require any changes in the underlying NeRF architecture. While the research field around implicit representations continues to evolve rapidly and better or faster NeRF formulations are regularly proposed, we see the ability of easy adaptation as a main advantage of our method.

III. Density-Aware NeRF Ensembles

A. Preliminaries

A NeRF is a parametric function $f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{d}) : \mathbb{R}^3 \times \mathbb{R}^2 \rightarrow \mathbb{R}^4$, implemented as a deep neural network with parameters $\boldsymbol{\theta}$, that encodes the density ρ and colour $\mathbf{c}$ of all point-direction pairs $(\mathbf{x}, \mathbf{d})$ in a scene. These density and colour predictions can be used to render a new view of the scene represented by the NeRF through the following process. For a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with origin point $\mathbf{o}$, direction vector $\mathbf{d}$, near and far bounds t_n and t_f, and with $T(t)$ indicating the accumulated transmittance along the ray (the probability that the ray travels from t_n to t without hitting another particle), the expected colour $C(\mathbf{r})$ of camera ray $\mathbf{r}(t)$ is

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \cdot \rho(\mathbf{r}_{(t)}) \cdot \mathbf{c}(\mathbf{r}_{(t)}, \mathbf{d})\, dt \tag{1}$$

In practice this integral is approximated via the quadrature rule, using discrete sums and stratified sampling from N evenly spaced bins to make the rendering process tractable. Through a series of simplifications that are not relevant for the remainder of the paper (we refer the interested reader to [1]), this becomes the convenient expression

$$C(\mathbf{r}) = \sum_{i=1}^{N} \overbrace{\underbrace{\prod_{j=1}^{i-1}(1 - o_j)}_{\text{transmittance}} \cdot \underbrace{o_i}_{\substack{\text{occupancy} \\ \text{probability}}}}^{\text{termination probability}} \cdot \overbrace{\mathbf{c}_i}^{\text{colour}} \tag{2}$$

with o_i being the occupancy probability $1 - \exp(-\rho_i \delta_i)$. Given a NeRF $f_{\boldsymbol{\theta}}$ and equation (2), we can render an image $\mathcal{I}$ by evaluating $C(\mathbf{r})$ for all rays $\mathbf{r}$ that pass through the camera center and the image plane. In the following, we use the notation $c_{\boldsymbol{\theta}}(\mathbf{r})$ to indicate the predicted colour along ray $\mathbf{r}$ using the rendering process of (2) and the NeRF $f_{\boldsymbol{\theta}}$.

B. NeRF Ensembles for Predictive RGB Uncertainty

Following the Deep Ensembles approach [19], we propose to quantify the predictive uncertainty of a NeRF by training an *ensemble* of networks $\{f_{\boldsymbol{\theta}_k}\}_{k=1\ldots M}$. The M ensemble members are initialised with different parameters $\boldsymbol{\theta}_k^{(0)}$, but trained on the same data. By interpreting the ensemble as a uniformly-weighted mixture model, the members' predictions are combined through averaging, and the predictive uncertainty is expressed as the variance over the individual member predictions. With an ensemble of NeRFs, the expected colour of ray $\mathbf{r}$ in a scene is

$$\boldsymbol{\mu}_{\text{RGB}}(\mathbf{r}) = \frac{1}{M} \sum_{k=1}^{M} c_{\boldsymbol{\theta}_k}(\mathbf{r}). \tag{3}$$

The predictive uncertainty can be expressed as the variance over the individual member predictions:

$$\boldsymbol{\sigma}^2_{\text{RGB}}(\mathbf{r}) = \frac{1}{M} \sum_{k=1}^{M} \left(\boldsymbol{\mu}(\mathbf{r}) - c_{\boldsymbol{\theta}_k}(\mathbf{r})\right)^2. \tag{4}$$

$\boldsymbol{\mu}_{\text{RGB}}$ and $\boldsymbol{\sigma}^2_{\text{RGB}}$ can be calculated very easily by rendering the M individual RGB images $\mathcal{I}_i$ and calculating the mean

and variance directly in pixel space. Both will be 3-vectors over the RGB colour channels, i.e we do not consider the covariance *between* colour channels.

We combine the variances from the colour channels into a single variance by taking the average along the three channels:

$$\bar{\sigma}^2_{\text{RGB}}(\mathbf{r}) = \frac{1}{3} \cdot \sum_{c \in \{RGB\}} \boldsymbol{\sigma}^2_{\text{RGB},(c)}(\mathbf{r}), \tag{5}$$

where $\boldsymbol{\sigma}^2_{\text{RGB},(c)}(\mathbf{r})$ indicates the variance associated with colour channel c.

C. Limitations of Simple Ensembling

Ensembling NeRFs and using the variance in RGB space to quantify the predictive uncertainty (i.e. aleatoric and epistemic) is a simple and partially effective method. However, this approach can fail to capture the model's epistemic uncertainty arising from parts of the scene that have *not* been observed during training.

Fig. 1 illustrates an instructive example; when rendering a novel view that exposes parts of the floor, the NeRF is forced to render areas of the scene that have never been observed during training. Although one would expect the NeRF ensemble to express high uncertainty in these image regions, all ensemble members agree to render this area in black, resulting in negligible variance in colour space $\bar{\sigma}^2_{\text{RGB}}$.

D. Density-Aware Ensembles to Capture Epistemic Uncertainty in Unseen Areas

A closer inspection of the ensemble predictions along a ray $\mathbf{r}$ in previously unobserved regions reveals that the individual NeRFs assign a low termination probability along all sample points on $\mathbf{r}$. The sum of the termination probabilities along the ray is close to zero (see Fig. 4 (left)), indicating the model does not assign belief to the hypothesis that the ray intersects the scene geometry within the near and far rendering bounds t_n and t_f. In short, writing the sum of the termination probabilities along a ray $\mathbf{r}$ as $q_{\boldsymbol{\theta}_k}(\mathbf{r})$, we observe

$$q_{\boldsymbol{\theta}_k}(\mathbf{r}) = \sum_{i=1}^{N} \overbrace{\underbrace{\prod_{j=1}^{i-1}(1-o_j)}_{\text{transmittance}} \cdot \underbrace{o_i}_{\substack{\text{occupancy} \\ \text{probability}}}}^{\text{termination probability at sample } i} \approx 0. \tag{6}$$

The average summed termination probability along ray $\mathbf{r}$ across the ensemble is then given by

$$\bar{q}(\mathbf{r}) = \frac{1}{M} \sum_{k=1}^{M} q_{\boldsymbol{\theta}_k}(\mathbf{r}), \tag{7}$$

where $\bar{q}(\mathbf{r}) \approx 1$ for rays that intersect with the scene structure observed during training, and $\bar{q}(\mathbf{r}) \approx 0$ otherwise. We interpret this as an expression of some of the model's *epistemic* uncertainty, arising from a fundamental lack of knowledge about the scene geometry and appearance along $\mathbf{r}$. To capture this uncertainty, we introduce an additional density-aware epistemic term $\sigma^2_{\text{epi}}(\mathbf{r})$ that we define as:

$$\sigma^2_{\text{epi}}(\mathbf{r}) = \left(1 - \bar{q}(\mathbf{r})\right)^2 \tag{8}$$

Finally, we combine the uncertainty measures based on the RGB-variance and above epistemic measure into the overall uncertainty $\psi^2(\mathbf{r})$:

$$\psi^2(\mathbf{r}) = \bar{\sigma}^2_{\text{RGB}}(\mathbf{r}) + \sigma^2_{\text{epi}}(\mathbf{r}) \tag{9}$$

The predicted rendered colour along a ray according to the ensemble is then modelled as a Gaussian with diagonal covariance matrix:

$$\tilde{C}(\mathbf{r}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\text{RGB}}(\mathbf{r}), \mathbf{I}_{3\times 3} \cdot \psi^2(\mathbf{r})\right) \tag{10}$$

We call this method *Density-aware* Ensembling, as $\sigma^2_{\text{epi}}(\mathbf{r})$ depends on the individual density predictions along each ray. As the experiments in the next section will show, $\sigma^2_{\text{epi}}(\mathbf{r})$ and $\bar{\sigma}^2_{\text{RGB}}(\mathbf{r})$ are complementary, capturing different aspects of the model's aleatoric and epistemic uncertainty.

IV. Experiments and Results

A. Experimental Setup

Datasets: We follow the evaluation protocol for uncertainty quantification in NeRFs established by [20], evaluating our approach on the eight scenes of the LLFF dataset [1] – 3 outdoor scenes (*Flower, Leaves, Orchids*) and 5 indoor scenes (*Fortress, Horns, T-Rex, Room, Fern*). As motivated in [20], we randomly split the available views from the individual datasets into 20% for training, and keep the remaining 80% for testing. This way, we can evaluate our model in a scenario where only a few (4-12) training views of a scene are available, which is more realistic for robotics applications than the dense viewpoint coverage typically encountered in NeRF datasets.
Baselines: Using the established datasets for evaluation allows us to directly compare with S-NeRF [20] and the baseline results published in [20]. These are Monte Carlo Dropout sampling [18], a naive ensembling approach based on deep ensembles [19], and NeRF in the Wild (NeRF-W) [29].

For the Monte Carlo Dropout baseline, [20] added a Dropout layer after every odd layer in the network, sampled 5 times, and used the variance in RGB space as the uncertainty measure. The Naive Ensembling baseline also uses the RGB variance, after training an ensemble with 5 members. The NeRF-W experiment in [20] was conducted by removing the latent embedding components and keeping only the uncertainty estimation layers. We refer the reader to [29] for details on the architecture of NeRF-W.

We additionally compare against CF-NeRF [21], by the authors of [20]. Since they follow the same evaluation protocol, we can directly compare against the results reported in [21].
Evaluation Metric: We report the Negative Log-Likelihood (NLL) as a principled and established way of assessing the quality of predictive uncertainty. NLL measures the likelihood of the *true* colour at a pixel under a Gaussian model $\mathcal{N}(\boldsymbol{\mu}_{\text{RGB}}(\mathbf{r}), \psi^2(\mathbf{r}))$ formed by the *predicted* mean colour $\boldsymbol{\mu}_{\text{RGB}}$ and our squared uncertainty measure as variance.
Implementation: We implement our Density-aware Ensembles based on the publicly available Instant-NGP [9] implementation. A simple modification lets us access the

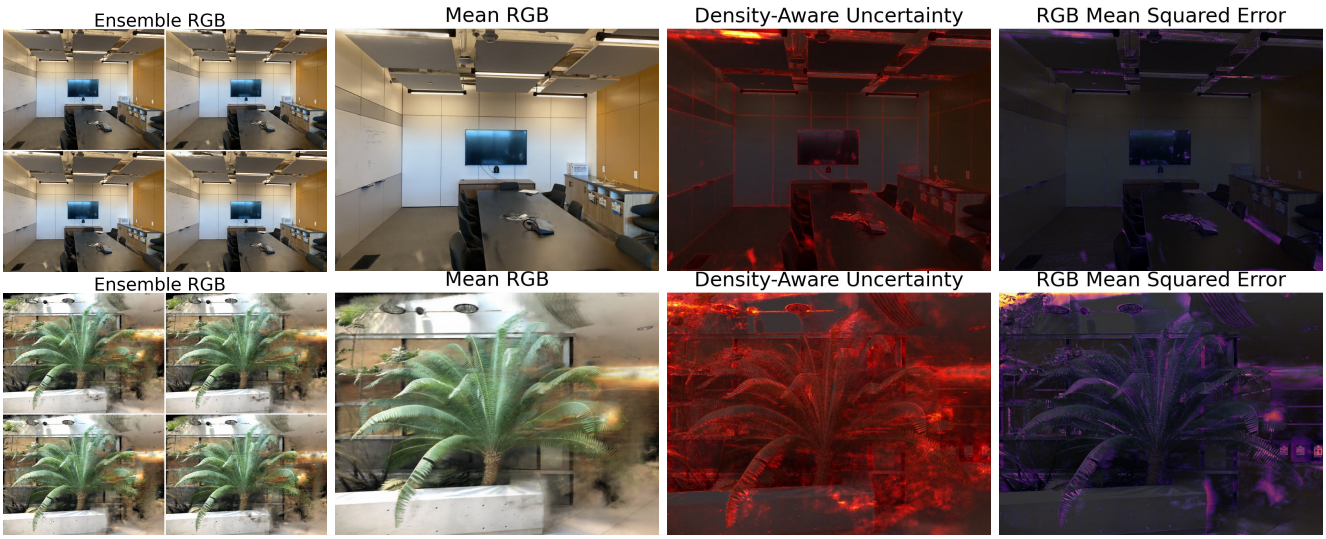

Fig. 2: Qualitative results for two views of the *Room* and *Fern* scene of the LLFF dataset.

TABLE I: Measuring uncertainty quantification with Negative Log Likelihood (NLL) for baselines and different ensemble sizes of our proposed method. Considered baselines are Monte Carlo Dropout (MC-DO), a naive ensembles approach implemented by [20], NeRF in the Wild (NeRF-W) [29], S-NeRF [20], and CF-NeRF [21]. The latter paper only published average NLL over all scenes instead of individual results. A † indicates results taken from [20], ‡ from [21].

		Negative Log-Likelihood ↓						Ablation Ensemble Size (NLL ↓)			
Dataset	Training Views (Ours)	MC-DO† $M=5$	Naive Ens† $M=5$	NeRF-W† [29]	S-NeRF† [20]	CF-NeRF‡ [21]	Ours $M=5$	$M=2$	$M=4$	$M=8$	$M=10$
Flower	7	4.63	1.63	1.71	1.27	–	**1.00**	1.88	1.13	0.90	0.85
Fortress	8	5.19	2.29	1.04	-0.03	–	**-1.30**	-1.28	-1.29	-1.30	-1.30
Leaves	5	2.72	2.66	0.79	**0.68**	–	0.97	2.32	1.10	0.80	0.73
Horns	12	4.18	2.17	0.78	0.60	–	**-0.55**	0.06	-0.50	-0.64	-0.66
T-Rex	11	4.10	2.28	1.91	1.37	–	**-0.31**	2.69	0.00	-0.65	-0.69
Fern	4	4.90	2.47	2.16	2.01	–	**-0.98**	-0.89	-0.97	-0.99	-1.00
Orchids	5	5.74	2.23	2.24	1.95	–	**-0.28**	0.06	-0.17	-0.29	-0.31
Room	8	5.06	2.13	4.93	2.35	–	**-1.35**	-1.29	-1.34	-1.35	-1.35
Average		4.57	2.23	1.95	1.27	0.57	**-0.35**	0.44	-0.26	-0.44	-0.47

TABLE II: Ablation study on the influence of the individual components of the uncertainty measure ψ^2. We report the average-mean and average-median NLL per scene along with the standard deviations. See the text for explanation.

	Negative Log-Likelihood ↓ ($M=10$)					
	$\psi^2 = \sigma^2_{\text{RGB}} + \sigma^2_{\text{epi}}$		$\psi^2 = \sigma^2_{\text{RGB}}$		$\psi^2 = \sigma^2_{\text{epi}}$	
Dataset	Mean	Median	Mean	Median	Mean	Median
Flower	0.85 ± 0.26	-0.07 ± 0.18	1997 ± 2764	0.76 ± 0.39	2.85 ± 0.95	0.46 ± 0.38
Fortress	-1.30 ± 0.07	-1.40 ± 0.01	-0.51 ± 1.17	-2.26 ± 0.35	-1.26 ± 0.10	-1.42 ± 0.01
Leaves	0.73 ± 0.174	-0.20 ± 0.10	3762 ± 3821	-0.06 ± 0.15	4.76 ± 1.72	0.72 ± 0.32
Horns	-0.66 ± 0.26	-1.35 ± 0.10	3.08 ± 2.81	-1.49 ± 0.26	0.35 ± 0.60	-1.42 ± 0.08
T-Rex	-0.69 ± 0.53	-1.50 ± 0.05	137 ± 480	-1.62 ± 0.821	7.36 ± 2.62	-1.49 ± 0.05
Fern	-1.00 ± 0.11	-1.30 ± 0.05	14.2 ± 39.7	-1.66 ± 0.23	-0.90 ± 0.14	-1.39 ± 0.02
Orchids	-0.31 ± 0.09	-0.95 ± 0.06	1.51 ± 0.55	-0.911 ± 0.11	0.51 ± 0.26	-1.09 ± 0.09
Room	-1.35 ± 0.08	-1.43 ± 0.01	-0.54 ± 1.51	-2.58 ± 0.20	-1.30 ± 0.10	-1.44 ± 0.01
Average	-0.47	-1.02	739	-1.23	1.55	-0.88

underlying densities along each ray during rendering. This enables us to calculate $q_{\boldsymbol{\theta}_k}(\mathbf{r})$ as per equation (6). We train 10 ensemble members for 5,000 training steps. Our machine with a Nvidia RTX 3090 trains for around 26 seconds per ensemble member, but we note that the process could be effectively parallelised. To calculate $\bar{\sigma}^2_{\text{RGB}}$, we render novel views at the full resolution of the ground truth image and calculate mean and variance in pixel space (see Section III-B).

B. Results and Ablation Study

Main Results: We show the main results of our experiments in Table I and qualitative results in Fig. 2. Our Density-aware Ensemble with 5 members achieves a better NLL on average across all scenes of the LLFF dataset. While S-NeRF [20] and CF-NeRF [21] report average performance of 1.27 and 0.57, our ensemble sets a new state of the art with a NLL of -0.35. We achieve a lower NLL on all individual scenes

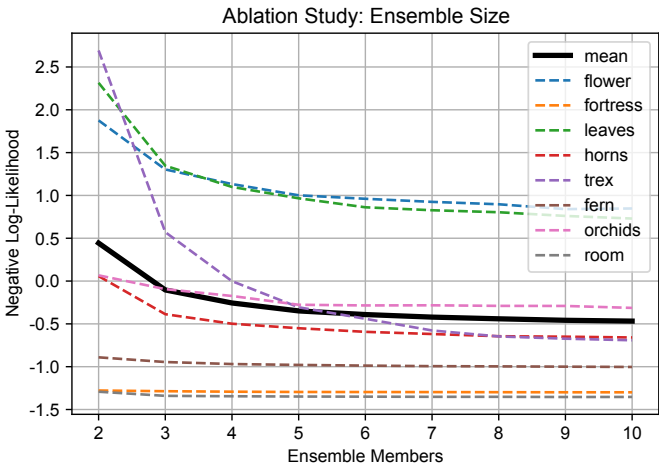

Fig. 3: Ensembling is feasible and effective for moderate ensemble sizes. The Negative Log-Likelihood converges quickly for most scenes, improving negligibly beyond sizes of 5 except for *T-Rex*.

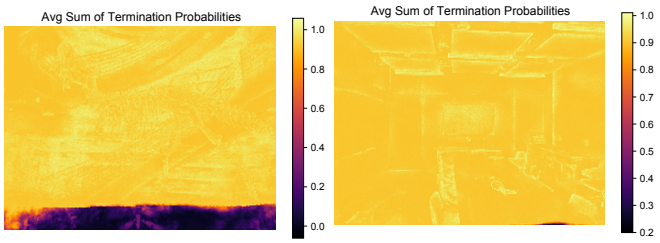

Fig. 4: The average sum of termination probabilities $\bar{q}_{\boldsymbol{\theta}_k}(\mathbf{r})$ for a view from the *T-Rex* and *Room* scenes [1] (RGB images in Fig. 1 and 2). The model assigns small termination probability to rays corresponding to parts of the view that were never observed in training (bottom part, left image). Interestingly, we also observe small fluctuations in termination probability per ray in other parts of the scene, where the model assigns a belief of slightly less than 1. Our ablation indicates this is correlated with the prediction error.

apart from *Leaves*.

Influence of Ensemble Size: We ablate the influence of M, the number of ensemble members. While we reported our main results for $M = 5$ ensemble members to directly compare to prior work, Table I also reports the results for different choices of M in the four rightmost columns. As expected, a larger ensemble results in higher quality uncertainty estimates, achieving an average Negative Log-Likelihood of -0.47 for $M = 10$ compared to -0.35 for the smaller ensemble with $M = 5$. This is consistent with previous findings on the performance of sampling-based uncertainty quantification [18], [19].

In Addition, Fig. 3 plots the achieved Negative Log-Likelihood for $M = 2 \dots 10$. From this plot, it is apparent that although the performance increases monotonically with M, the gains become more and more diminishing for larger M. This is significant for practical use cases, as it indicates that even a relatively small NeRF ensemble can express uncertainty of high quality.

Influence of Individual Uncertainty Measures: Our proposed uncertainty measure $\psi^2(\mathbf{r})$ consists of two terms, the mean RGB variance $\bar{\sigma}^2_{\text{RGB}}$ and the epistemic variance σ^2_{epi}. To

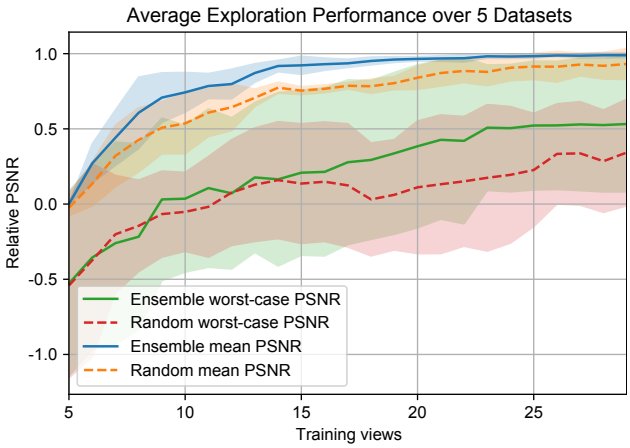

Fig. 5: A NeRF ensemble can select informative next-best training views. Starting from 5 highly similar views, we incrementally add new views, chosen at random or based on the ensemble uncertainty.

better understand their individual influence, Table II reports results when using only either term, or both terms combined.

For this ablation study, we report both the mean-average and the mean-median Negative Log-Likelihood, along with their standard deviations, for every scene in the LLFF datset. The mean-average NLL is calculated by averaging the NLL over all pixels in a rendered test image, and then calculating the mean over all test images in a scene. In contrast, the mean-median NLL calculates the *median* NLL over all pixels, before averaging over all test views. Reporting both metrics reveals that using only the RGB-based variance σ^2_{RGB} as the uncertainty measure is affected by severe outliers, i.e. pixels with very high (bad) NLL. This becomes clear when comparing the mean-average and the mean-median performances: while the former is subject to outliers, the latter is relatively robust against outliers. As explained in the motivation for our method in Section III-B, the outlier pixels with very high NLL are caused by parts of the scene that were not observed during training.

As a somewhat surprising result, we observe that the epistemic uncertainty term σ^2_{epi} by itself achieves reasonable performance on most scenes. Upon closer inspection, this is caused by small fluctuations in the individual $q_{\boldsymbol{\theta}_k}(\mathbf{r})$, the sum over the termination probabilities per ray. We observe that the model assigns slightly lower $q_{\boldsymbol{\theta}_k}(\mathbf{r})$ for rays with higher uncertainty, i.e. close to but not exactly 1. We illustrate this in Fig. 4 for views from two scenes.

However, using the sum of σ^2_{RGB} and σ^2_{epi} as the proposed uncertainty measure $\psi^2(\mathbf{r})$ consistently yields the best results, indicating the complementarity of both components.

V. Using Uncertainty for Next-Best View Selection and Model Refinement

We now describe an experiment that utilises the ensemble uncertainty for next-best view selection and model refinement.

Datasets: We use five synthetic scenes (*Lego, Hotdog, Ficus, Drums, Microphone*) from the dataset published alongside the original NeRF paper [1]. In these scenes, the cameras of

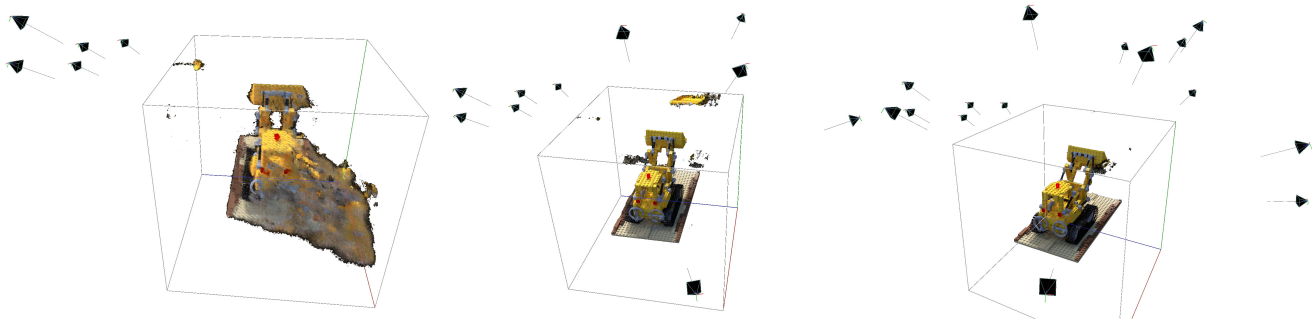

Fig. 6: Next-best view selection using ensemble uncertainty. Left: 5 initial and very similar training views lead to high model error. Centre: After selecting 5 views, the model error is significantly reduced. Right: After adding another 5 views to the training set.

the training set are positioned in a semi-sphere around the object, facing the object centre. From the training dataset of every scene, we randomly select 5 images from very similar viewpoints for the initial training, and keep the remaining images as candidate views to select the next-best (most informative) training image.

The test views are independent from the training data and used to evaluate the quality of the scene reconstruction, using the PSNR (Peak Signal-to-Noise Ratio) metric, as is standard in the NeRF literature [1], [20].

Method, Metric and Baseline: Starting with the initial training set of 5 views, we train an ensemble of $M = 5$ NeRFs for 2,000 steps each. We then calculate the predictive uncertainty for each of the *remaining* unused candidate views, and select the view with the maximum uncertainty as the next view to be added to the training data. We then retrain the NeRF and iterate this process (see Fig. 6). As a simple baseline method, we select the next-best view at random.

After every training iteration, we evaluate the model quality with the independent test dataset. Comparing the rendered test views with the ground truth images, we calculate the average PSNR over all test images. We identify the lowest PSNR to measure the worst-case performance, i.e. the test view with the highest error compared to the ground truth.

To make the PSNR comparable across all scenes, we re-scale PSNR so that the average PSNR at the beginning of our evaluation loop (i.e. only using the initial 5 views) is 0, and that the best average PSNR across all iterations (usually the PSNR of the final iteration which has access to most training images) is equal to 1. We can then plot the average and worst-case performance in one plot for both strategies of selecting the next best view (uncertainty-based selection and random selection) in Fig. 5.

Results: The uncertainty measured by our ensembling approach is an effective way of selecting the next-best view to add to the training dataset. As we can see in Fig. 5, the average-case performance is better compared to the random baseline. Although both methods eventually converge to the same performance as more images are used for training, using uncertainty for view selection achieves a higher gain in quality per training image by selecting more informative views.

Similarly, the worst-case performance is significantly improved when selecting views using our quantified uncertainty. This indicates that our density-aware ensemble uncertainty is an effective proxy for ground-truth error – by selecting the view with the highest uncertainty, we effectively choose to add the view that causes a high error to the training set for the next iteration.

These results are significant for robotics and active vision, where gathering exhaustive training data is too expensive or time consuming, and a trade-off between representation quality and the number of training views (or the time required to gather those views) has to be considered.

VI. Discussion, Conclusions, and Future Work

We have shown that the key to making ensembling an effective approach for uncertainty quantification in NeRFs is to combine the simple RGB variance with an additional epistemic uncertainty term that is informed by the predicted densities along each individual ray.

A major advantage of our method is the simplicity of adopting ensembling strategies to new emerging variants of NeRFs. This allows us to leverage progress regarding training time, required training views, representational power or rendering speed, while maintaining the ability to readily quantify predictive uncertainty in emerging NeRFs in the future. NeRF ensembling approaches are already computationally feasible after the appearance of hash-encoding NeRFs [9] that can train in mere seconds (with the option of training all ensemble members in parallel) and render in real time.

Our qualitative results (Fig. 4) suggest that there is no strict distinction between aleatoric and epistemic [22] uncertainty in our uncertainty measure. Being able to separately quantifying both is worthwhile for future work, since in some applications (e.g. next-best view selection) epistemic uncertainty is more informative than aleatoric uncertainty.

Another interesting direction for future work is to use our density-aware ensemble to guide exploration and navigation of a mobile robot through an unknown scene, while mapping it with the NeRF. While we have shown the uncertainty to be effective in selecting the next-best view from a set of candidate views, we are interested in extracting the gradient of the uncertainty measure with respect to the camera pose, and plan or control a trajectory for scene exploration based on this local gradient information.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] F. Dellaert and L. Yen-Chen, "Neural volume rendering: Nerf and beyond," *arXiv preprint arXiv:2101.05204*, 2020.

[3] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[4] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[5] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.

[6] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3d neural scene representations for visuomotor control," in *Conference on Robot Learning*. PMLR, 2022, pp. 112–123.

[7] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "Nerf-supervision: Learning dense object descriptors from neural radiance fields," in *International Conference on Robotics and Automation (ICRA)*, 2022.

[8] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," in *Conference on Robot Learning*. PMLR, 2022, pp. 526–536.

[9] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[10] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[11] D. M. Rosen, K. J. Doherty, A. Terán Espinoza, and J. J. Leonard, "Advances in inference and representation for simultaneous localization and mapping," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 215–242, 2021.

[12] S. Garg, N. Sünderhauf, F. Dayoub, D. Morrison, A. Cosgun, G. Carneiro, Q. Wu, T.-J. Chin, I. Reid, S. Gould *et al.*, "Semantics for robotic mapping, perception and interaction: A survey," *Foundations and Trends® in Robotics*, vol. 8, no. 1–2, pp. 1–224, 2020.

[13] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford *et al.*, "The limits and potentials of deep learning for robotics," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2002.

[15] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.

[16] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.

[17] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[18] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2016, pp. 1050–1059.

[19] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[20] J. Shen, A. Ruiz, A. Agudo, and F. Moreno-Noguer, "Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations," in *International Conference on 3D Vision (3DV)*, 2021.

[21] J. Shen, A. Agudo, F. Moreno-Noguer, and A. Ruiz, "Conditional-Flow NeRF: Accurate 3D Modelling with Reliable Uncertainty Quantification," in *European Conference on Computer Vision (ECCV)*, 2022.

[22] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.

[23] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 641–676.

[24] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[25] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision (ECCV)*, 2022.

[26] Sara Fridovich-Keil and Alex Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *CVPR*, 2022.

[27] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," 2021.

[28] D. Driess, I. Schubert, P. Florence, Y. Li, and M. Toussaint, "Reinforcement learning with neural radiance fields," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[29] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.