university of groningen / faculty of science and engineering / artificial intelligence

Quantifying the Uncertainty for Neural Radiance Fields in Few-Shot Scenarios

Bachelor's Project Thesis

Micky Labrèche, s4021290, m.t.labreche@student.rug.nl,
Supervisor: Dr M.A. Valdenegro Toro

Abstract: Neural Radiance Fields (NeRFs) can accurately capture 3D volumes in neural networks but are often unreliable in real-world settings where information is limited. This is problematic for settings that require decision-making like autonomous driving. This research uses uncertainty quantification to gain insight into the reliability of predictions. This is done by applying a method called Flipout to NeRF. It uses Variational Inference to increase uncertainty quality and is found to produce better predictions and associated uncertainty than previous methods.

1 Introduction

Neural Radiance Fields (NeRFs) have seen a lot of attention since their introduction by Mildenhall et al. (2021). Their appeal lies in the ability to encode 3D volumes implicitly within a relatively uncomplicated deep neural network. This interest has led to considerable improvements in fidelity and computational efficiency.

Usually, a set of numerous training images from diverse viewpoints is required for optimal results. However, in real-world scenarios, this availability of images is rare, and even when available, it adds computational costs making it harder to create real-time applications. There has been promising research by for instance Verbin et al. (2022) looking to increase the accuracy of NeRF models with only a few training images in so-called "few-shot settings". However, they are unable to show the associated uncertainty of their predictions.

Knowledge about the uncertainty of a NeRF model is essential for real-world settings like autonomous driving or medical imaging. It can indicate areas of an object that are likely to be incorrect. This can directly influence decision-making.

Estimating the uncertainty in neural networks is achieved with Bayesian deep learning. Several techniques exist that approximate the uncertainty. Researchers have been investigating integrating these techniques with NeRFs. While these methods achieve good qualitative results, a complete analysis of the predicted uncertainty is still an area to improve upon.

This research aims to quantify the uncertainty using a general method for stochastic neural networks called "Flipout". This approach produces a lower variance than Bayes by Backprop by Blundell et al. (2015) and produces better uncertainty characteristics.

To simulate a real-world setting, this method is performed in a few-shot scenario with three training images. This makes accurate reconstructions more difficult but increases the opportunity to analyze the uncertainty in inaccurate areas.

Ultimately, this research addresses the question: "How effective are the uncertainty quantification properties of Flipout compared to other methods?"

It is found that the novel Flipout approach presents more effective way to interpret uncertainty in the context of NeRFs compared to previous methods.

2 Related work

2.1 NeRF

A Neural Radiance Field (NeRF) can capture a 3D environment in a neural network. This is done by encoding a radiance value, essentially the intensity of light, for every combination of a position in 3D space $\boldsymbol{x}$ and a viewing direction $\boldsymbol{d}$. The radiance $\boldsymbol{c}$

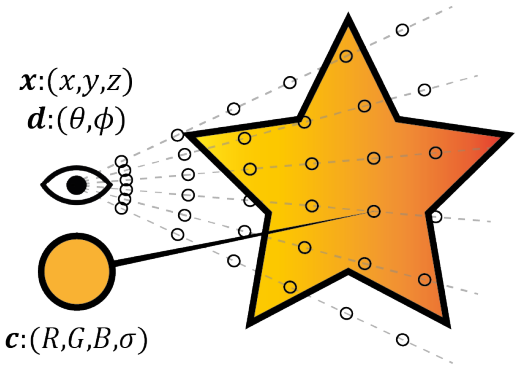

Figure 2.1: Rays are projected from a novel viewpoint. Radiance samples are taken along the rays.

consists of four values. One for each color channel in RGB, and a density σ, which can be interpreted as the ray termination probability.

To extract a 2D image from the NeRF, rays $\boldsymbol{r}(t)$ are projected into the scene from a novel viewpoint, as is visible in Figure 2.1. Samples t are taken along the ray from t_n to t_f. These samples are accumulated to calculate the final pixel color. This process is described in the volumetric rendering function (2.1), where $T(t)$ is the transmittance. See Mildenhall et al. (2021) for a more detailed explanation.

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\, \sigma(\boldsymbol{r}(t))\, \boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})\, dt \qquad (2.1)$$

To train a NeRF, usually, a large set of images from diverse viewpoints is used. Issues will emerge when it is trained on very few images. However, in real-world scenarios, a large set of images is almost never available. This makes it an interesting area for research. This research intentionally constrains the number of images available to NeRF.

2.2 NeRF in Few-Shot Scenarios

In Few-Shot scenarios, NeRF is trained on a limited set images, usually $1 \sim 4$. Figure 2.2 shows such a training set. To achieve good results with this limitation, several regularization techniques have been proposed. This section outlines two popular techniques and examines their applicability in real-world settings.

Figure 2.2: Training images used in a Few-Shot scenario. Only one side of the Lego excavator is observed.

Verbin et al. (2022) leverages a depth baseline to improve the quality. They generate a point cloud using a Structure from Motion (SfM) algorithm applied to the training data, which is then used to supervise the optimization process.

Although this method produces more accurate results, it often makes significant prediction errors related to object structure. Additionally, prediction errors from the point cloud can propagate to the NeRF representation, thereby increasing inaccuracies.

Accurate scene structure is essential for autonomous navigation. However, it's insufficient to rely solely on improved accuracy. Current predictions tend to be overconfident, providing no insight into whether the model is extrapolating.

Alternatively, Deng et al. (2022) regularize based on patches in unseen views. Their proposed loss functions encourage geometry smoothness and color consistency, removing the need for an additional depth baseline.

Still, as acknowledged by the authors, the effectiveness of this approach lies in the assumption that the true structure is mostly smooth and the color remains consistent. These conditions are not applicable in all cases and can result in blurred reconstructions when the scene includes high-frequency details.

This can be problematic in autonomous navigation, where missing detail on a sign, for example, can lead to misclassification. Similar to the previous method, there's no way to detect such misclassifications.

To meet the demands of real-world settings, it's clear that simply improving accuracy through regularization is not enough. Uncertainty quantification is a critical component that needs to be incorporated into the process.

2.3 Uncertainty Quantification with Bayesian Neural Networks

With Bayesian statistics, when estimating the probability of an observation, we not only take into account the frequency of that observation but also include prior knowledge about how likely it is to occur.

$$\mathbb{P}(\boldsymbol{w}|\boldsymbol{D}) = \frac{\mathbb{P}(\boldsymbol{D}|\boldsymbol{w})\,\mathbb{P}(\boldsymbol{w})}{\mathbb{P}(\boldsymbol{D})} \tag{2.2}$$

This concept is described in equation (2.2) and is applied in training Bayesian neural networks (BNNs). By learning a BNN, the goal is to learn the posterior probability $\mathbb{P}(\boldsymbol{w}|\boldsymbol{D})$ of the weight configurations ($\boldsymbol{w}$) given some data ($\boldsymbol{D}$). This is called the Bayesian weight posterior distribution (BWPD).

$$\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}) = \int_w \mathbb{P}(\boldsymbol{y}|\boldsymbol{w},\boldsymbol{x})\,\mathbb{P}(\boldsymbol{w}|\boldsymbol{x})\,d\boldsymbol{w} \tag{2.3}$$

After training, suppose a prediction is to be made with a given input $\boldsymbol{x}$. According to equation (2.3), the learned BWPD can be used to calculate $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$. This is called the Bayesian predictive posterior distribution (BPPD).

As can be seen, this prediction is not a single value as in regular neural networks. In fact, all weights and outputs in BNNs are modeled as probability distributions rather than single-point values. This approach is more informative because a probability distribution provides an indication of a value's uncertainty, which can be interpreted as a measure of trustworthiness.

Unfortunately, it is not possible to describe equations (2.2) and (2.3) as closed-form expressions. Additionally, an exact approximation of these expressions is computationally infeasible. This makes BNNs typically intractable.

However, it is possible to generate individual samples from the theoretical BPPD by making relatively simple adjustments to a regular neural network. The collected samples form a histogram that roughly approximates the BPPD, shown in figure 2.3. Several adjustments have been proposed.

Such as Monte Carlo Dropout by Gal & Ghahramani (2016) (where random activations are dropped during inference), Monte Carlo DropConnect by Mobiny et al. (2021) (where random weights are dropped during inference), and estimation using ensemble methods like Lakshminarayanan et al. (2017).

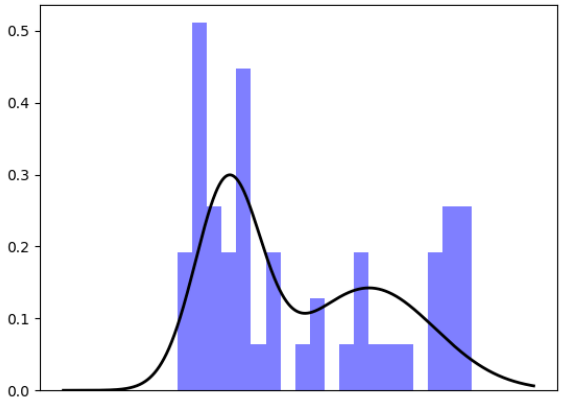

Figure 2.3: Approximation of the Bayesian Predictive Probability Distribution (BPPD) with 50 samples forming a histogram

These naive sampling methods lead to a mediocre fit for the BPPD. As seen in figure 2.3, some values are sampled too many times and others too little.

A more accurate approximation is achievable by having an explicit training objective that learns a probability density function $q_\theta(\boldsymbol{w}|\boldsymbol{D})$ that fits the $\mathbb{P}(\boldsymbol{w}|\boldsymbol{D})$ (BWPD) as close as possible.

This approach is known as variational inference, which is the main method employed in this research and is further explained in the following sections.

2.4 Variational Inference

Variational inference in BNNs has been extensively researched as a method to quantify uncertainty.

As mentioned before, the goal is to fit $q_\theta(\boldsymbol{w}|\boldsymbol{D})$ to the BWPD during training. This is done by adding a distance metric between the two distributions to the loss so that this is minimized.

$$KL(q,p) = \int_x q(x)\log\frac{q(x)}{p(x)} \tag{2.4}$$

This is done with the Kullback-Leibler divergence described in equation (2.4). This is a general metric that measures the separation between two probability distributions.

The evidence term $\mathbb{P}(\boldsymbol{D})$ in equation (2.2) cannot be assumed or computed. Therefore, The KL diver-

Paper	RGB	σ	$\mathbb{P}(RGB\|x)$	$\mathbb{P}(\sigma\|x)$	UQ Method
NeRF Mildenhall et al. (2021)	✓	✓	×	×	none
S-NeRF Shen et al. (2021)	✓	✓	✓	✓	Variational Inference
CF-NeRF Shen et al. (2022)	✓	✓	✓	✓	Variational Inference
Density Aware NeRF Sünderhauf et al. (2023)	✓	✓	✓	✓	Deep Ensembles
Flipout NeRF	✓	✓	✓	✓	Variational Inference

Table 2.1: NeRF papers with corresponding uncertainty quantification methods and which values they estimate

gence needs to be reformulated into the Evidence Lower BOund (ELBO).

Minimizing the ELBO, means that it is only possible to approximate the lower bound of the evidence during training. The evidence term could be arbitrarily large, so it is unknown what the accuracy of the resulting approximation is. However, for Bayesian deep learning, it is more effective than the naive sampling approach.

$$\mathcal{L} = KL(q_\theta(\boldsymbol{w}|\boldsymbol{D}), \mathbb{P}(\boldsymbol{w})) + \mathcal{L}_{NLL} \tag{2.5}$$

The final loss is described in equation (2.5). Where the KL divergence between the approximate BWPD $q_\theta(\boldsymbol{w}|\boldsymbol{D})$ and the prior $\mathbb{P}(\boldsymbol{w})$ is calculated and added to the $\mathcal{L}_{NLL}$, which is the negative log-likelihood as expectation over the approximate BWPD.

Some drawbacks of using variational inference are the detrimental effects on training speed. The loss can become unstable because of the added stochasticity and it may take much longer to converge.

2.5 NeRFs and Uncertainty Quantification

The concept of Bayesian Neural Networks can be used to quantify the uncertainty in NeRFs. In regular NeRF, it is possible to predict the RGB value and the density given an input. For a NeRF that adopts a BNN, it is also possible to predict the standard deviation of the RGB and of the density.

This section introduces and evaluates previous research that aims to quantify the epistemic uncertainty in NeRFs. Their goal is to find areas where the model is more uncertain about the reconstruction of a scene. Some papers mentioned in this section are found in Table 2.1.

Shen et al. (2021) use a variational inference approach which they call "Stochastic NeRF" (S-NeRF). They learn the probability distributions of the weights and the outputs. Their approach incorporates an algorithm similar to Bayes by Backprop by Blundell et al. (2015).

As simplifications, they assume that weights are normal distributions and that view-radiance pairs are independent. However, they find that this leads to a lack of consistency in the output image at adjacent pixels.

They address the independence assumption in a follow-up paper Shen et al. (2022), by modeling a complex joint distribution between each prediction. They achieve smoother outcomes by using conditional normalizing flows.

Sünderhauf et al. (2023) try an alternative method of quantifying uncertainty with Deep Ensembles, introduced by Lakshminarayanan et al. (2017). They also devise a new uncertainty term that is more informative for areas with low density.

Generally, many assumptions and simplifications are made to make these methods work. This is necessary to make the problem tractable but compromises the produced uncertainty quality.

For Deep Ensembles, they simplify the problem by not explicitly training the model to approximate

the BWPD. They treat the weight configuration of every ensemble member as a sample from the BWPD. This means many ensemble members are necessary to obtain a good approximation of the uncertainty.

The aforementioned variational inference methods do approximate the BWPD during training. However, they are unstable and produce a large variance which can compromise uncertainty quality. This is caused by the simplification that only a single weight sample from the BWPD is taken while training a batch of data. Ideally, multiple samples are taken to increase diversity. This problem is addressed in the following section.

2.6 Flipout

A BNN randomly changes the weights of a model to learn which values are more likely. To change the weights, a sample from the BWPD is taken. Only the weight matrix of the stochastic layers is adjusted with this sample.

As mentioned before, this only happens once during each training batch. This means that the same weight matrix is used for over 12 million forward passes when a batch contains 64 coarse samples and 128 fine samples over an image of 256x256 pixels. To obtain better results, a separate sample needs to be taken for each forward pass. Unfortunately, this is computationally unfeasible.

Wen et al. (2018) found that they could transform the existing weight matrix within a batch instead of taking a new sample each time. They do this in a way that the new weights are sufficiently uncorrelated while still representing a sample from the BWPD. This adds minimal computational overhead.

This allows for more variation during training, which leads to a reduced variance of the predictions and faster convergence. The technique is called Flipout and is applied to regular NeRF. It is analyzed in the following sections.

3 Experimental Setup

In this research we apply Flipout to NeRF. The base NeRF implementation uses a hierarchical sampling approach as presented in Mildenhall et al. (2021), using 64 coarse samples and 128 fine samples. The NeRF consists of 8 fully connected layers, each containing 256 dense units. This NeRF implementation doesn't use any additional optimizations. This base model is extended with various uncertainty quantification methods.

All methods, with the exception of ensembles, only use stochastic weights in the output layers of NeRF. These are in total 4 layers, consisting of the 2 output layers for RGB and Depth and the 2 layers before.

By increasing the number of stochastic layers, it is possible to produce better uncertainty. However, the produced image quality and speed of convergence can suffer. By taking more samples, the quality of the uncertainty can also be increased. Every method except ensembles uses 10 samples to make a prediction.

3.1 Uncertainty Methods

3.1.1 MC-DropConnect

Stochastic results are produced during inference by randomly setting weights to zero. The results are samples from the BPPD. Weights are dropped for every forward pass with a probability of 5%.

3.1.2 Ensembles

An ensemble consists of 5 NeRF models, all with identical architecture but different weight initialization. The results that are produced are samples from the BPPD. A standard deviation and mean can be calculated over all outputs for each forward pass.

3.1.3 Bayes By Backprop

Bayes by backprop is a variational inference method. It does not take variation in training batches into account. Given the batch size of 1, the kl-weight is set to 1. The prior parameters are sigma 1: 5.0, sigma 2: 2.0, and prior pi: 0.5.

3.1.4 Flipout

Flipout, is another variational inference method that is the main focus of this paper. This does take variation within training batches into account. Given the batch size of 1, the kl-weight is set to 1.

The prior parameters are sigma 1: 5.0, sigma 2: 2.0, and prior pi: 0.5.

3.2 Data

The data that is used for training, testing, and evaluation consists of rendered images from a 3D model using Blender. This synthetic dataset was created by Mildenhall et al. (2021). The 3D Lego excavator from this dataset is used. This 3D model is especially suitable because it contains plenty of high-frequency details.

The novel viewpoints generated in this research are not covered by images in the dataset. For this reason, a pseudo-ground truth is computed by training a regular NeRF model for 120 epochs on 100 images from the training dataset.

For the Few-Shot scenario that is used in the experiments, only 3 images are used in training instead of 100. These images share similar viewpoints. This ensures that only a specific part of the object is observed, creating an ideal scenario for evaluating the uncertainty in unseen views. The exact views used in training are shown in figure 2.2.

3.3 Analysis

3.3.1 Novel views

To analyze the performance of the models, novel views are produced that are not seen in the training data. These novel views are gathered in two settings. The first setting is called the "360-views" setting and circles around the object, varying the azimuthal angle ϕ with intervals of 60, while keeping the polar angle θ the same at a value of -30. The second is called the "unseen-views" setting and only observes the unseen half of the scene from ϕ:[30,150] with an interval of 30. It also varies the polar angle θ:[-45,45] in three intervals, to observe the topside and the underside of the object.

3.3.2 Qualitative Analysis

To analyze the quality of the predicted RGB image, the difference between this image and the RGB ground truth is calculated. The absolute difference between the color channels of each pixel is taken. All color channels are then summed forming an error map. This is done the same way for the depth maps.

To create the RGB uncertainty maps, the predicted standard deviations of the rgb are accumulated along the rays. In the volumetric rendering function, see equation (2.1), the predicted density is used to weigh the RGB uncertainty. Note that this is different from the density uncertainty. The mean of all color channels is taken to produce an uncertainty map.

To create the depth uncertainty maps, the predicted standard deviations of the density are accumulated along the rays and rendered according to the volumetric rendering function.

The color scale of each map is displayed to the right. Because these maps often contain outliers, extreme values are clipped. The 99.5 percentile of the predicted maps is calculated and used as the upper bound for scaling.

3.3.3 Quantitative Analysis

A quantitative analysis of the produced images is also performed. For this, we use the standard metrics PSNR and SSIM that analyze the similarity between the prediction and the ground truth. Additionally, the Mean Absolute Error (MAE) is taken. This is the mean over the absolute differences for each pixel as described in equation (3.1).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.1}$$

To analyze the quality of the uncertainty quantification we take the Gaussian Negative Log-Likelihood for each pixel. This estimates the probability of the ground truth given a Gaussian probability distribution. The Gaussian is defined by the predicted mean $\hat{y}$ and predicted standard deviation $\hat{\sigma}$. This is described in equation (3.2).

$$NLL = \frac{1}{n}\sum_{i=1}^{n}\log\hat{\sigma}_i^2 + \frac{(y_i - \hat{y}_i)^2}{\hat{\sigma}_i^2} \tag{3.2}$$

The minimum value in the predicted uncertainty is changed to $1e-2$. This minimizes the effect of black pixels in the background of the image.

3.3.4 Plots

Plots are used to compare the outputs of different methods. For each of the 4 output channels R,

G, B and σ, the predicted standard deviation and the error are graphed in scatter plots to show their correspondence. This is done for the "360-views" setting, meaning that the seen and the unseen side of the excavator is analyzed.

The maximum limit of the y-axis for all color channels and methods is 0.2. The maximum limit of the y-axis for the depth is 87. This is different for ensembles as they produce larger uncertainty.

In the calibration plot, the standard deviation and the accuracy are related. The accuracy is divided into bins of 20. For each bin, the mean standard deviation is calculated. Also in this case

The ROC curves indicate the separation between seen and unseen views based on the predicted uncertainty. This is done by taking two views that are close to the training data. The angles (ϕ, θ) of these views are correspondingly (-60, -30) and (-120,-30). Additionally, two opposing views, that are far from the training data, are taken with the angles (-240, -30) and (-300, -30). The False Positive Rate and True Positive Rate are calculated and plotted.

4 Results

A Flipout NeRF model is trained in a few-shot scenario and is compared with the baseline model as described in section 3.2. Additionally, it is compared to other uncertainty quantification methods as described in section 3.1. The following sections present and analyzes the results.

4.1 View Synthesis in a Few-Shot Scenario

The qualitative results of Flipout NeRF are visualized in Figure 4.2 and in Figure 4.3. Novel views are generated according to the "360-views" setting. From top to bottom, it displays the predicted value, the ground truth, the absolute error, and the uncertainty.

4.1.1 RGB View Synthesis

As is visible in the predicted RGB images in Figure 4.2, Flipout NeRF learns structures that do not correspond to the ground truth. Mainly areas that are unseen during training are being extrapolated. It can be seen in Figure 4.1 that during training, the validation loss decreases at first, but very quickly flattens out and stops improving. As expected, it shows that Flipout NeRF isn't generalizing well for all 100 views.

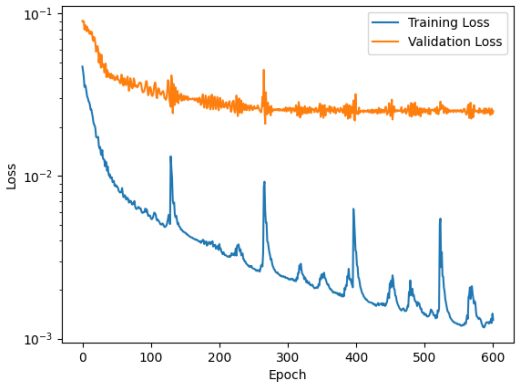

Figure 4.1: The training loss over the three training images decreases, while the validation loss over all 100 images remains the same.

Viewpoints that are in the range $\theta = [-180°, 0°]$ are further from the training viewpoints and show more extrapolation and an increased absolute error. This is supported by Table 4.1, where it can be seen that the Mean Average Error increases correspondingly with the viewing angles of Figure 4.2.

Additionally, there is a black hole in the middle of the object for unseen views. This means that these areas have a predicted density of 0. It must be noted that these areas are also highlighted by the error despite the fact that they do not represent extrapolated structures.

The low-density areas of the excavator show a large error but no uncertainty. It is expected that the uncertainty is similar to the error, but as a result of the previous observation, the uncertainty does not necessarily correspond to the error.

However, it can be seen in Table 4.1 that the Negative Log-Likelihood of the uncertainty shows the same behavior as the MAE. It increases for unseen views.

The top of the excavator shows significantly more uncertainty than the bottom. Here the structures are thinner and comparatively move a lot in screen space between different views. This could be a reason why there is more disagreement between samples.

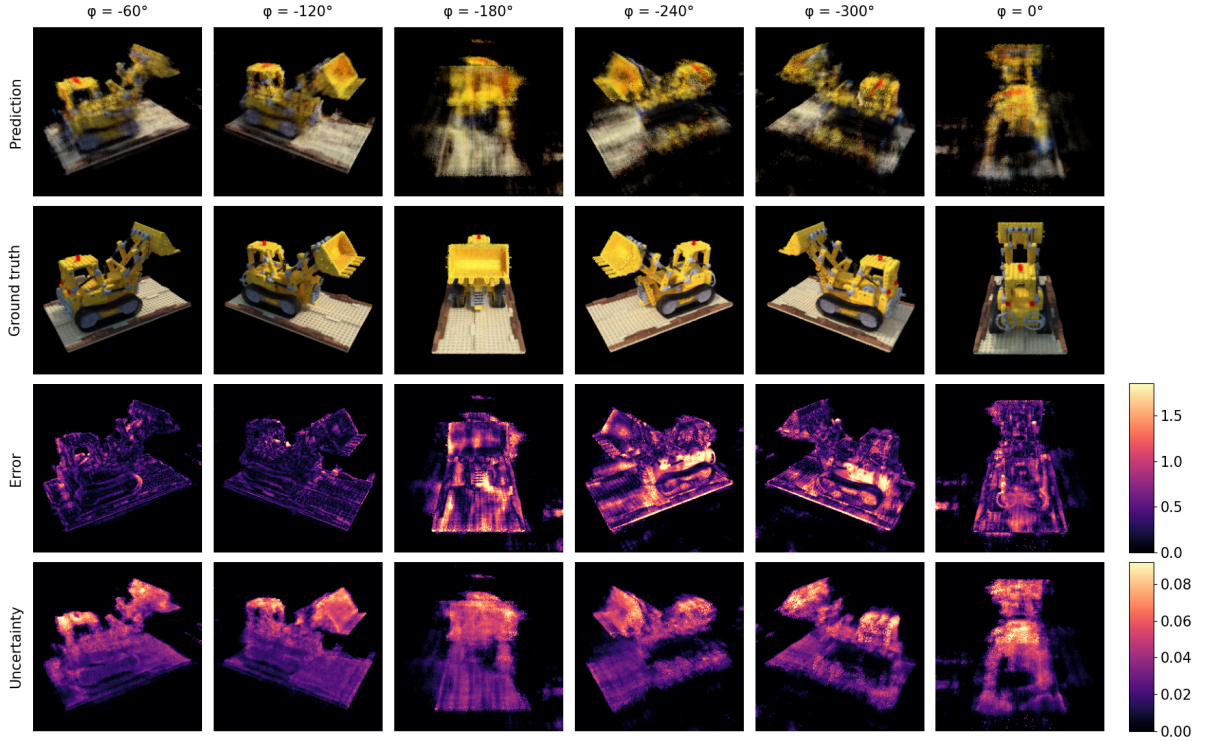

Figure 4.2: The performance of the predicted RGB is analyzed by observing views in the "360-views" setting. Novel views are generated by varying the azimuthal angle ϕ. The two most left columns are closest to the training views and show a low error. When observing the columns to the right, the error increases. It can be seen that the observed side of the excavator shows lower uncertainty than the unobserved side.

ϕ angle	-60°	-120°	-180°	-240°	-300°	0°
RGB MAE ↓	0.114	0.108	0.249	0.286	0.245	0.199
$\mathbb{P}(RGB)$ NLL ↓	17.71	17.26	112.12	195.61	159.82	77.95

Table 4.1: Performance metrics are calculated seperatly for each viewing angle in the "360-views" setting. The MEA indicates the quality of the prediction and the NLL indicates the quality of the uncertainty. As can be seen, both metrics indicate worse performance when moving away from the training views.

	RGB PSNR ↑	RGB SSIM ↑	RGB MAE ↓	$\mathbb{P}(RGB)$ NLL ↓
MC-DropConnect	14.25	0.537	0.094	336.17
Ensembles	13.92	0.473	0.109	170.99
Bayes By Backprop	**17.08**	**0.645**	**0.064**	144.50
Flipout	16.69	0.634	0.067	**96.74**

Table 4.2: The performance of different uncertainty quantification methods in the "360-views" setting is compared. Bayes By Backprop achieves a slightly better RGB prediction than Flipout. However, Flipout seems to outperform all other methods when estimating the uncertainty.

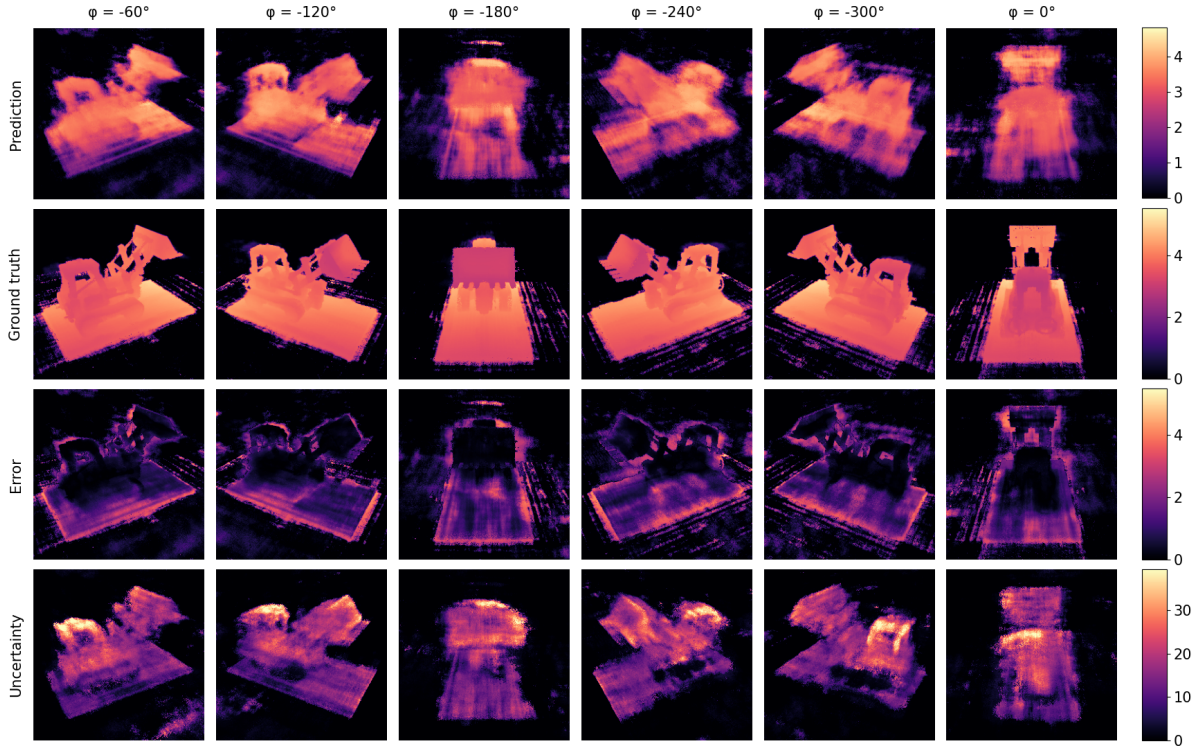

Figure 4.3: The performance of the predicted depth is analyzed in the "360-views" setting. Novel views are generated by varying the azimuthal angle ϕ. The top row shows mediocre results compared to the ground truth. This difference is not entirely captured in the error. It can be seen that the uncertainty is localized on the unseen side.

ϕ angle	-60°	-120°	-180°	-240°	-300°	0°
Depth MAE ↓	0.579	0.584	0.611	0.722	0.673	0.618
$\mathbb{P}(\sigma)$ NLL ↓	631.34	1513.42	1504.32	2525.20	1212.69	515.45

Table 4.3: Performance metrics are calculated seperatly for each viewing angle θ in the "360-views" setting. The MEA indicates the quality of the prediction and the NLL indicates the quality of the uncertainty. As can be seen, both metrics indicate worse performance when moving away from the training views.

	Depth RMSE ↓	Depth MAE ↓	$\mathbb{P}(\sigma)$ NLL ↓
MC-DropConnect	1.565	0.968	5246.01
Ensembles	1.543	0.945	1190.76
Bayes By Backprop	1.168	0.678	**969.26**
Flipout	**1.064**	**0.631**	1317.07

Table 4.4: The performance of different uncertainty quantification methods in the "360-views" setting is compared. Flipout achieves a slightly better depth prediction than Flipout. However, Bayes By Backprop seems to outperform all other methods when estimating the uncertainty.

In Table 4.2 it can be seen that Bayes By Backprop and Flipout achieve very similar quantitative results. However, Flipout performs better when it comes to uncertainty quality.

4.1.2 Depth View Synthesis

The predicted depth maps by Flipout are visible in Figure 4.3. It can be seen that structures are stretched horizontally and there are more floating artifacts. There is more extrapolation when comparing the prediction to the ground truth than in Figure 4.2.

The edges of the object are not very sharp and do not correspond to the ground truth edges. As a result, mainly the edges are highlighted by the error. This makes it harder to find smaller inaccuracies in the body of the excavator. In this case, the error is an even worse indicator of extrapolated data.

However, when analyzing Table 4.3, it seems that there is a similar pattern as is visible in the RGB results. The error and uncertainty quality seem to get worse when moving further away from the seen views.

The produced uncertainty is comparable to the uncertainty in RGB space. Yet, it shows even more localized uncertainty to the unseen side of the excavator.

Different methods are compared in Table 4.4. As can be seen, Bayes By Backprop slightly outperforms Flipout when it comes to uncertainty quality. However, Flipout achieves better overall depth quality in this case.

4.2 Uncertainty

In this section, the predicted uncertainty of Flipout is analyzed and compared to other methods. Only "unseen-views" are analyzed as described in section 3.3.1. These views are expected to have increased uncertainty within these areas.

4.2.1 Flipout

Figure 4.4 (RGB) and Figure 4.5 (depth) display the uncertainty produced by Flipout.

Both the RGB and depth uncertainty maps show uncertainty that seems to be in roughly the same areas. Specifically, around the hood and the bucket of the excavator. These areas seem to be localized at the unseen side of the excavator.

When observing the top and middle rows of the RGB uncertainty, it can be seen that there is a lack of uncertainty at the center. This is caused by the lack of predicted density as was mentioned before.

A similar argumentation can be made for the underside of the LEGO base, where there is a black spot visible underneath the tracks of the excavator.

However, other parts from the underside view show visible uncertainty despite the fact that these parts were never observed. This might be because the LEGO base is represented as a single layer, rather than a LEGO brick with a distinct top and bottom. This means that the top can be seen from the underside.

It is essential to acknowledge that the standard deviation of the RGB uncertainty is considerably small, with a maximum of 0.08 in a value range of 0 to 255. This means the predicted RGB is likely accurate.

When analyzing the depth map in Figure 4.5, it can be seen that in this case, there is high uncertainty in the middle part of the excavator as opposed to the RGB case. This means there is a predicted density for this part.

When looking at Figure 4.5, it can be seen that the predicted density of this area is low, while the uncertainty for this part is comparatively high.

The bottom row of the figure displays extrapolated floating structures, which are marked as highly uncertain. This is correct as they are not part of the scene.

For the depth maps, the standard deviation is larger than the RGB uncertainty maps, reaching a maximum value of 35. This indicates high general variability in the density.

The uncertainty seems to be robust and remains consistent across different unobserved viewpoints.

4.2.2 MC-DropConnect

As can be seen in the results generated by MC-DropConnect in Figure 4.6, large parts of the scene are not modeled or extrapolated.

The uncertainty is not localized at a particular side of the excavator. However, similarly to Flipout, it marks the hood and bucket of the excavator more uncertain.

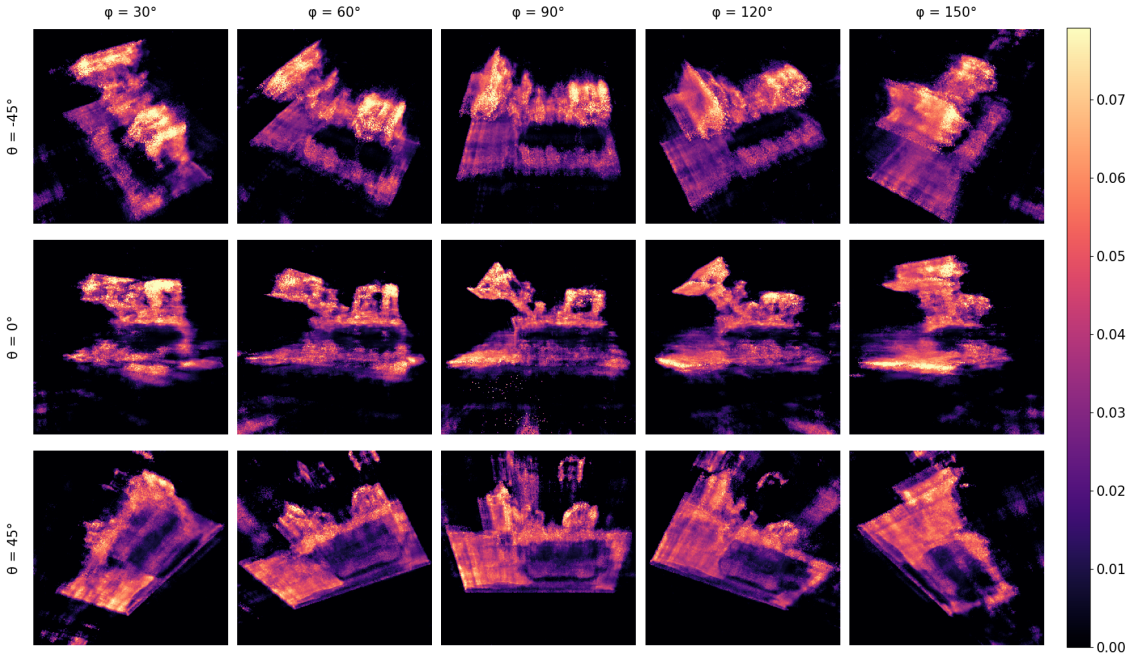

Figure 4.4: RGB uncertainty maps generated in the "unseen-views" setting by Flipout. The maps display the mean standard deviation of all color channels.

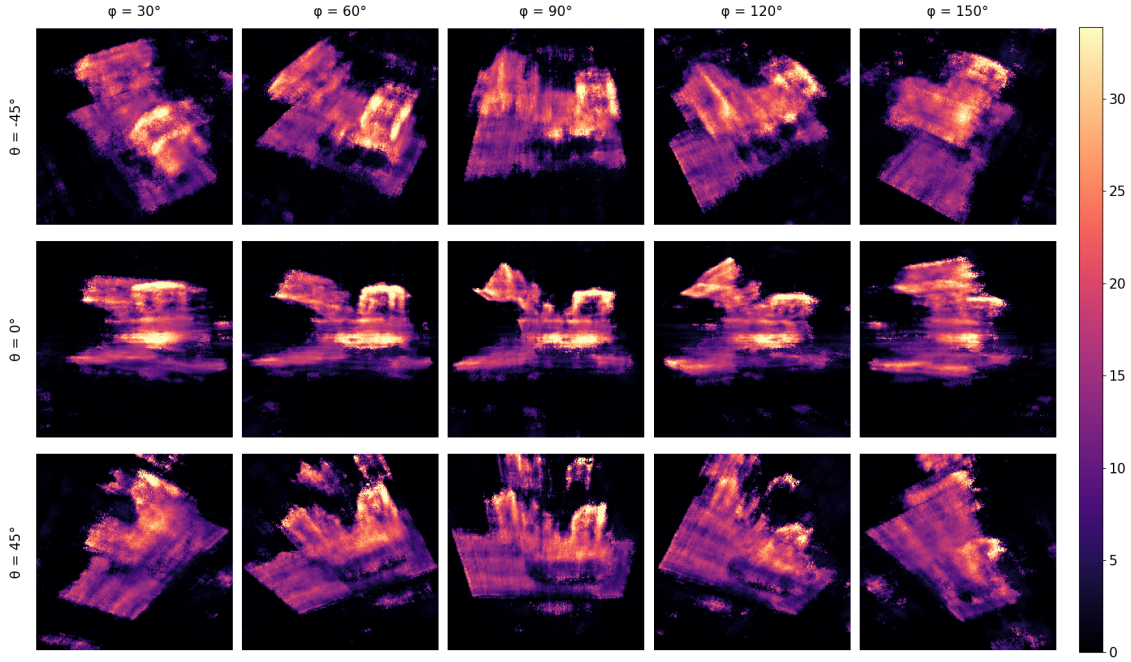

Figure 4.5: Depth uncertainty maps generated in the "unseen-views" setting by Flipout. The maps display the standard deviation of the depth.

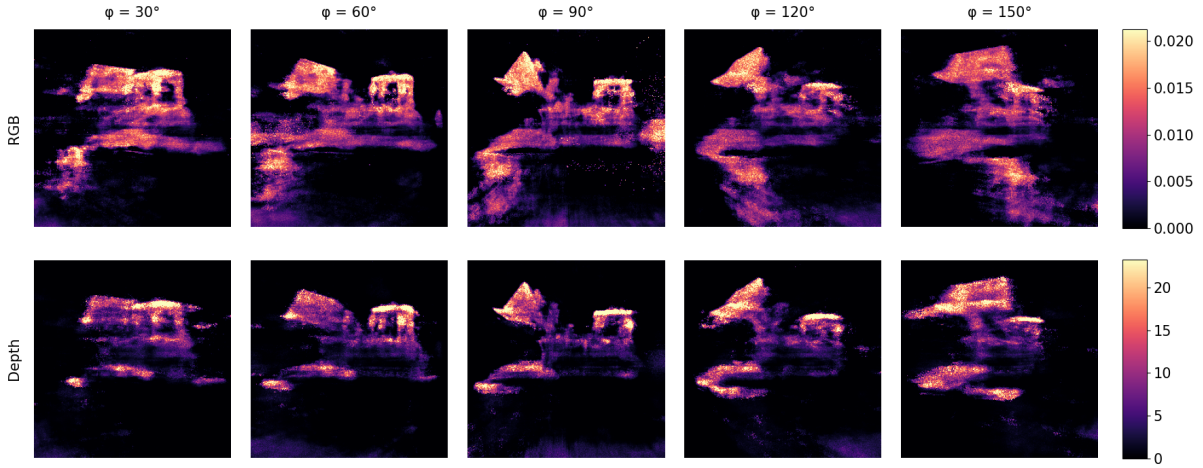

Figure 4.6: Uncertainty maps from the middle row of the "unseen-views" setting generated by MC-DropConnect. The top row is the mean RGB standard deviation and the bottom row is the depth standard deviation.

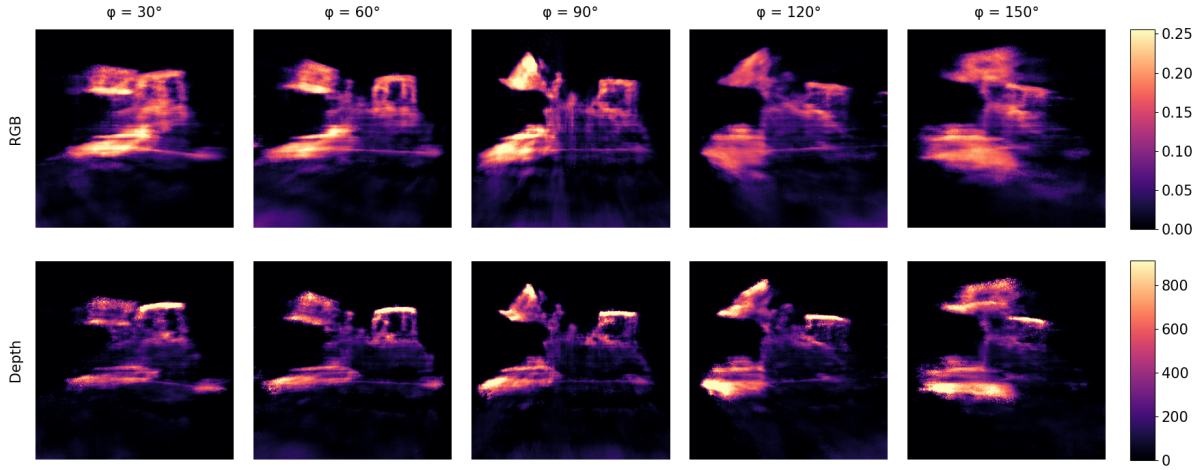

Figure 4.7: Uncertainty maps from the middle row of the "unseen-views" setting generated by Ensembles. The top row is the mean RGB standard deviation and the bottom row is the depth standard deviation.

The RGB uncertainty does not remain consistent when changing from views. The uncertainty of the bucket is view-dependent.

For these reasons, DropConnect does not seem to have ideal uncertainty quantification properties.

4.2.3 Ensembles

Figure 4.7 displays the results generated by Ensembles. It can be seen that the reconstruction of the object is way more blurry than other methods.

This could be because of the amount of disagreement between each model. Ensembles converge slower and end up with blurry results.

It seems that the RGB uncertainty does not remain consistent between views. The blurriness makes it impossible to see if the uncertainty is localized.

The depth uncertainty clearly shows uncertainty in specific areas. These areas contain thin struc-

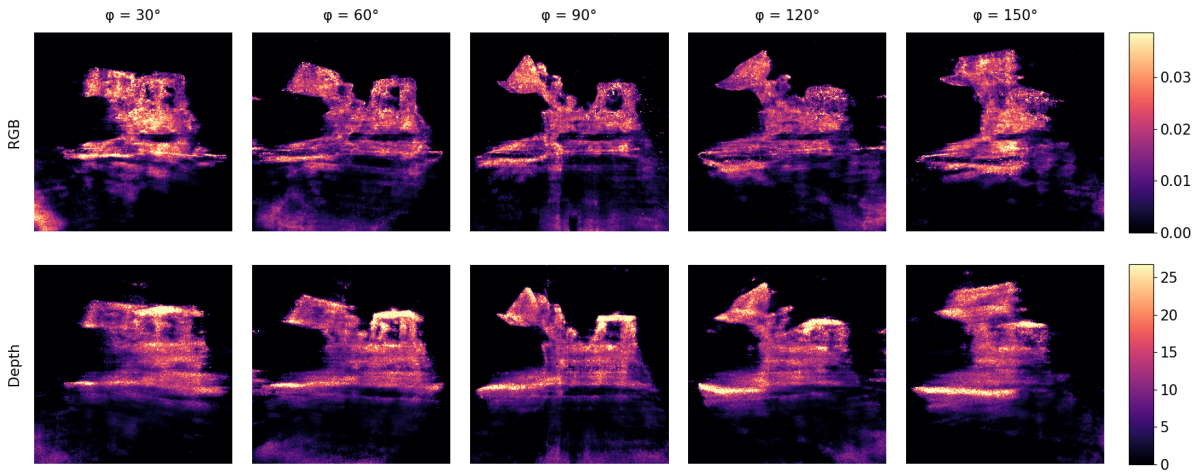

Figure 4.8: Uncertainty maps from the middle row of the "unseen-views" setting generated by Bayes By Backprop. The top row is the mean RGB standard deviation and the bottom row is the depth standard deviation.

tures. It is not clear if the uncertainty is located on the unseen side.

It must be noted that ensembles produce higher uncertainty than other methods, as can be seen from the color bar.

Ensembles are not ideal for uncertainty quantification.

4.2.4 Bayes By Backprop

The uncertainty results by Bayes By Backprop can be found in Figure 4.8.

The RGB uncertainty seems to be equally distributed over the whole excavator. Additionally, it can be seen that results are generally noisy, leading to small spots with high uncertainty. It cannot be seen which side of the excavator was observed.

The depth maps have better uncertainty characteristics, mainly highlighting thin structures and edges. However, even in these views it is not possible to discern the seen from unseen the unseen side.

This makes Bayes By Backprop not ideal for uncertainty quantification.

4.3 Error and Uncertainty Correspondence

As the error is an indication of extrapolation, it is expected that areas with a large error show more uncertainty. This is visualized in Figure 4.9 by relating the error to the uncertainty in scatter plots.

Dark areas in these plots mean that many uncertainty predictions are made for that error value. Lighter areas indicate that fewer uncertainty predictions or none at all are made for that uncertainty value.

It must be noted that as was stated before, the error does not necessarily correspond to the uncertainty.

When observing the RGB outputs it can be seen that all distribution shapes show a descending slope. All methods have a tendency to predict too high uncertainty for low error values and too little uncertainty for high error values. This does not correspond to the expectation that a higher error should cause higher uncertainty as this would result in a ascending slope.

A possible reason for this is the fact that there are more low error values in a scene than high error values, making it harder to spot a trend. Additionally, it can be seen that some slopes start from higher than others. This could be because the colors are scene dependent and some scenes have a larger tendency to create uncertainty for a specific color.

It can be seen that the distribution shape that Flipout produces, shows more variability in the standard deviation. This means there is more diver-

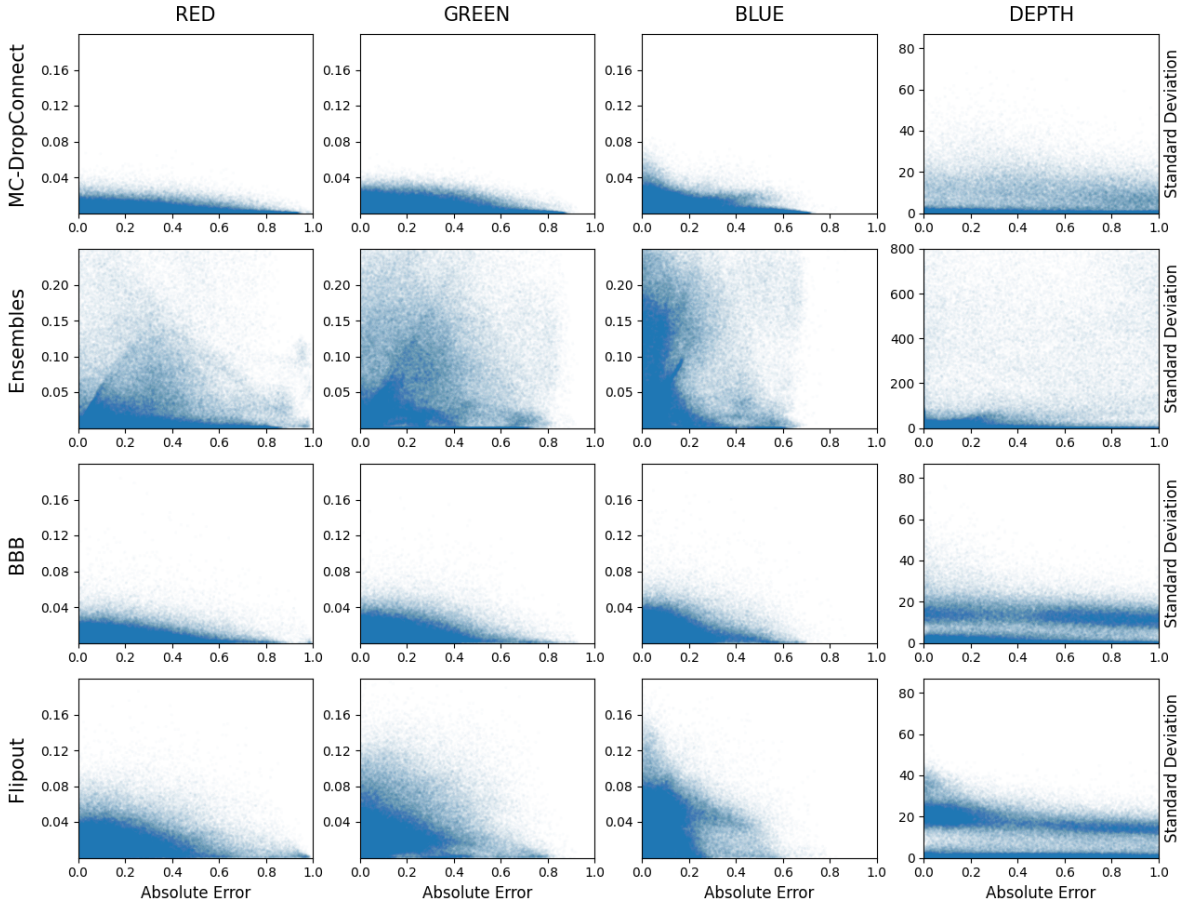

Figure 4.9: Matrix of error-uncertainty plots using novel views from the "360-views" setting. Columns correspond to model outputs and rows correspond to methods. The x-axis is the absolute error and the y-axis is the standard deviation.

sity in uncertainty predictions which leads to more useful information.

In the depth plots, it can be seen that the shape looks very different. All methods produce a flat line and the variability of the standard deviation is constant over all error values. This means that many different error values share an equal amount of uncertainty. This makes it very hard to use uncertainty as a proxy for the error.

This is probably caused by the edges that caused a large variety of error values. The uncertainty in this area however remained the same.

Also in the case of the depth plots, Flipout shows more variability making it a better candidate than previous methods.

4.4 Calibration

Similar to the previous section, it is expected that the uncertainty matches the performance of the model. In this case, Figure 4.10 shows the confidence related to the accuracy of the model.

The data is binned and an average confidence is calculated for each bin. This means that it is no longer dependent on the amount of high error areas in the image.

When observing the RGB outputs for all models, it can be seen that the lines show a linear correlation between confidence and accuracy. This is as expected and shows that the models are relatively well-calibrated.

It can be seen that there is a sudden increase in

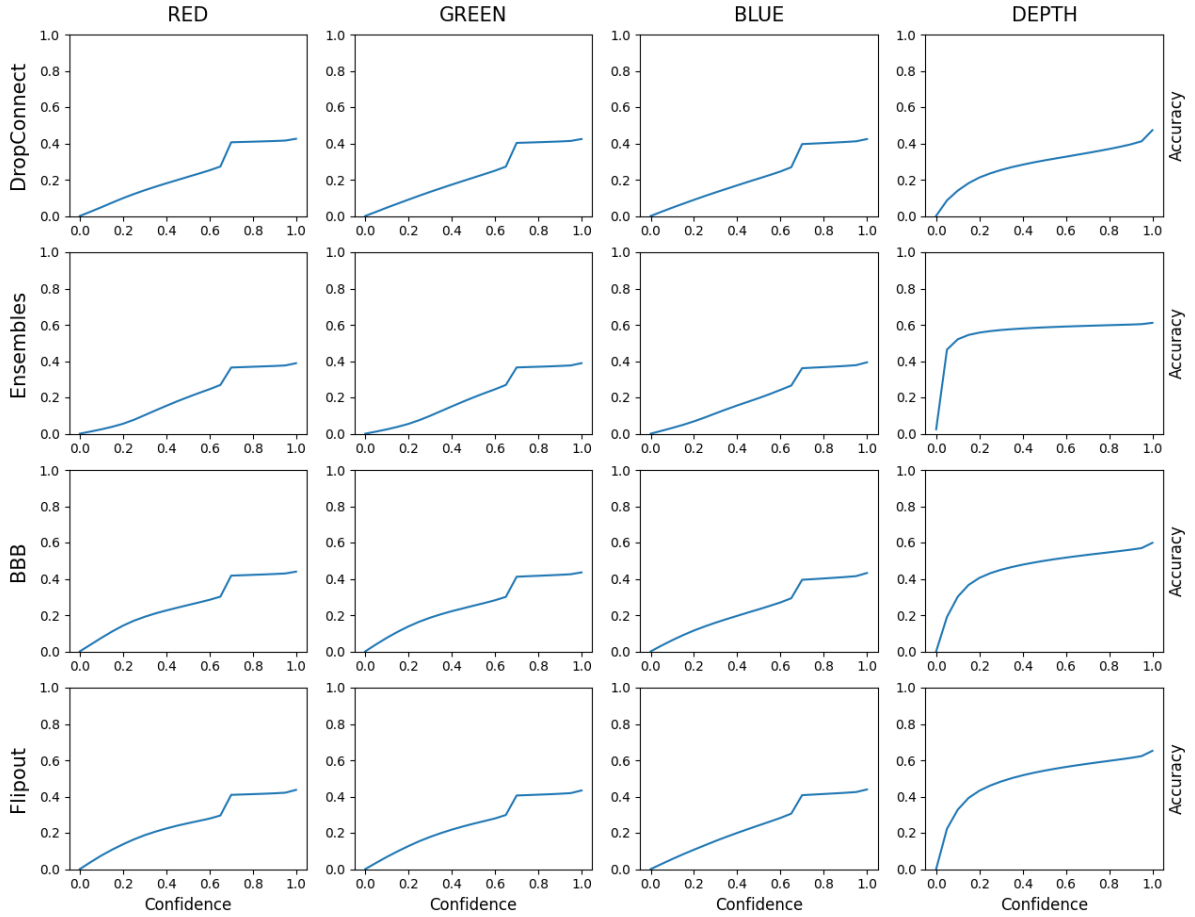

Figure 4.10: Matrix of calibration plots using novel views from the "360-views" setting. Columns correspond to model outputs and rows correspond to methods. The x-axis is the confidence and the y-axis is the accuracy.

confidence for accuracy values around 0.7. There is no apparent reason for this.

When observing the depth lines it can be seen that produce higher confidence values than expected. It can be seen that MC-DropConnect actually shows the best calibration properties, having a line that is closest to a linear correlation.

Lines that are above a straight diagonal line indicate overconfident predictions. Lines that are under a straight diagonal indicate underconfident predictions.

Flipout does not seem to differentiate from other methods in this regard.

Several methods could be implemented to correct the calibration of uncertainty quantification models.

4.5 Out of Distribution Detection

Previous sections related the uncertainty to the performance of the model. In this section, only uncertainty produced by Flipout is analyzed because it is impossible to access the ground truth in the real world. This means that it is not possible to calculate the performance of a model. However, it is possible to use the predicted uncertainty as a proxy for the performance.

Seen views are considered as the training distribution. Unseen views are out of the training distribution. The uncertainty of the two settings is com-

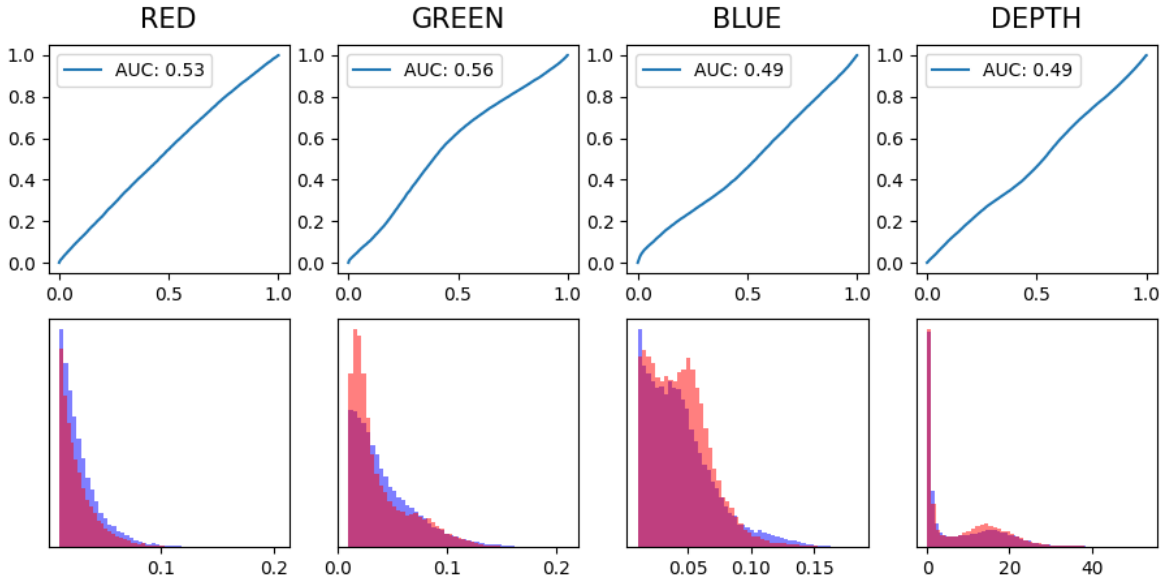

Figure 4.11: Top row: ROC plots that relate the False Positive Rate and the True Positive Rate. Bottom row: Histograms that show the distributions of seen and unseen views. Blue is seen views, red is unseen views.

pared to analyze if there is a difference and how large the difference is. This difference can be used to check if a new view is inside or outside the training distribution.

The distributions of the two settings are compared in Figure 4.11. From the histograms, it can be seen that the distributions overlap. This means that predictions from unseen views are very similar to predictions from seen views.

The ROC curve shows how separable the distributions are. When there is a clear separation, the line should be higher than a straight diagonal and have an Area Under the Curve (AUC) approaching 1.

It can be seen in Figure 4.11 that the produced ROC line is linear. This means it is not possible to differentiate the distributions from each other. The AUC is correspondingly around 0.5.

This ambiguity of uncertainty for different views can also be seen in Figure 4.4 and Figure 4.5. Here, the predicted uncertainty seems to have the same distribution for every image. The uncertainty seems to be localized but is still visible from seen views.

5 Conclusion

In the previous section, a NeRF model was trained with Flipout and evaluated against other models. This was done to quantify the uncertainty in the predictions of the model.

Both variational inference approaches, Bayes By Backprop and Flipout, produced the least extrapolation under the limitation of a few-shot scenario. However, Flipout outperformed all previous methods in uncertainty quality. This is the case for both the RGB uncertainty and the depth uncertainty.

From the synthesized novel viewpoints, it can be seen that the produced RGB and depth uncertainty is localized at the unseen side of the object. This is not the case for previous methods that produce view-dependent uncertainty or show uncertainty distributed over the whole object.

To indicate the difference between the ground truth and the prediction the absolute error is calculated over different novel views around the object. It can be seen that unseen views have an increased error. This is supported by the results of the Mean Absolute Error and the Negative Log-Likelihood for corresponding views.

When relating the error to the produced uncer-

tainty, it can be seen that Flipout produces the most variation in uncertainty values for different error values compared to other methods. This means that Flipout is a better proxy for the error than other methods.

However, the error does not capture all relevant inaccuracies. For instance, it calls attention to the unmodeled sections of the object. This is not relevant in comparison to the uncertainty that can only capture modeled section and aims to display the extrapolation.

Additionally, other error metrics than the absolute error have not been considered. Using the squared error instead could lead to different results.

For accurate calibration, it is expected that confidence and accuracy are linearly correlated. All models seem to perform similarly in this regard. The color channels for all methods show a sudden increase in confidence when the accuracy is around 0.7. Flipout seems to make more overconfident predictions for the depth than other methods.

The distribution of the training set is compared to the distribution of opposing views to detect if there is a difference. It is found that views close to the training distribution can not be differentiated from views that are far from the training distribution. This has to do with the fact that the same amount of uncertainty is visible through the object from the seen side.

In conclusion, Flipout seems to have a theoretical benefit over previous methods, this is reflected in some of the empirical results. Overall Flipout has more effective uncertainty quantification properties. However, there are still problems with detecting views that are out of distribution.

Future work could use real-world datasets to have an even more realistic setting for robotics. Additionally, it could focus on increasing the uncertainty quality by adding methods that produce calibrated uncertainty.

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural network. In *International conference on machine learning* (pp. 1613–1622).

Deng, K., Liu, A., Zhu, J.-Y., & Ramanan, D. (2022). Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 12882–12891).

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050–1059).

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, *30*.

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.

Mobiny, A., Yuan, P., Moulik, S. K., Garg, N., Wu, C. C., & Van Nguyen, H. (2021). Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific reports*, *11*(1), 5458.

Shen, J., Agudo, A., Moreno-Noguer, F., & Ruiz, A. (2022). Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In *European conference on computer vision* (pp. 540–557).

Shen, J., Ruiz, A., Agudo, A., & Moreno-Noguer, F. (2021). Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. In *2021 international conference on 3d vision (3dv)* (pp. 972–981).

Sünderhauf, N., Abou-Chakra, J., & Miller, D. (2023). Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In *2023 ieee international conference on robotics and automation (icra)* (pp. 9370–9376).

Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J. T., & Srinivasan, P. P. (2022). Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (pp. 5481–5490).

Wen, Y., Vicol, P., Ba, J., Tran, D., & Grosse, R. (2018). Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*.