



**Instant Uncertainty Calibration of NeRFs Using a
Meta-Calibrator**
Supplementary Material

Niki Amini-Naieni¹, Tomas Jakab¹, Andrea Vedaldi¹, and Ronald Clark¹

University of Oxford

Introduction

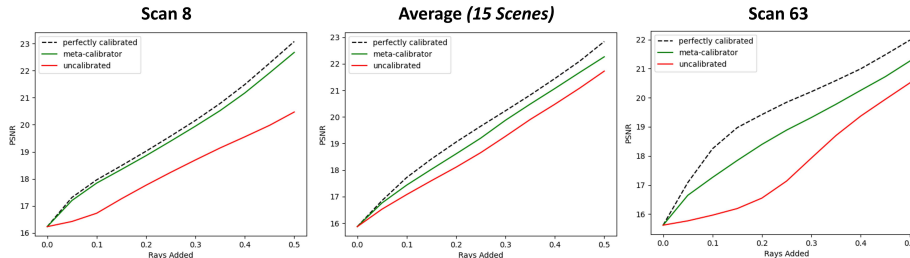


Fig. 1: Advantage of the meta-calibrator for next-best view planning. This figure shows the information gain in DTU [1] from uncalibrated and calibrated uncertainty-guided ray selection for next-best view planning. Specifically, for evenly spaced fractions $\gamma_i \in [0, 0.5]$, we plot the average PSNR over the test set assuming the top $100\% \times \gamma_i$ most uncertain pixel colors are predicted perfectly by the NeRF model. We use the updated PSNR to quantify information gain. Picking rays according to the calibrated uncertainties (green) consistently results in higher PSNRs than picking rays according to the uncalibrated uncertainties (red). Individual results for *Scan 8* (leftmost plot) and *Scan 63* (rightmost plot) and average results over all fifteen scenes (middle plot) in DTU are shown. The dashed black lines show results for theoretically perfect calibration, where the ground truth calibration curves for the test set are used to construct the calibration curves instead of the meta-calibrator.

In the supplementary material for *Instant Uncertainty Calibration of NeRFs Using a Meta-Calibrator*, we include additional details on the motivation for the meta-calibrator, applications of our approach, and experiments and code to support our design. In Sec. 1 we explain why the Principal Component Analysis (PCA) representation of the calibration curves is necessary; in Sec. 2 we show that using the training set as the calibration set results in severe overfitting; in Sec. 3, we show that holding out data results in poor performance at image reconstruction; in Sec. 4, we investigate the influence of the number of samples along the ray on the quality of the base uncertainties; in Sec. 5 we include a detailed example showing that calibration can re-order the pixel uncertainties, improving applications such as next-best view planning (see Fig. 1); in Sec. 6 we demonstrate the efficiency of our uncertainty metric over other approaches; in Sec. 7 we include more qualitative examples illustrating the calibrated uncertainty; in Sec. 8 we discuss the limitations of our approach; and in Sec. 9 we include an ethics statement. The additional details, explanations, experiments, and code provided here are intended to enhance the reader’s understanding of our approach and to further motivate, support, and explain the statements in the main paper. In summary, the supplementary material complements the content in the main paper and answers potential lingering questions.

Table of Contents

1	Why is the PCA Representation Necessary?	3
2	Using the Training Set Leads to Severe Overfitting	3
3	Holding Out Data Results in Poor Image Quality	4
4	Number of Ray Samples' Influence on Uncertainty Quality	5
5	Calibration Can Correct the Order of Pixel Uncertainties	6
6	Efficiency of Uncertainty Metric	7
7	Further Qualitative Examples	8
8	Limitations	9
9	Ethics	9
10	The Code	9

1 Why is the PCA Representation Necessary?

One might wonder why the PCA parameterization of the curves is necessary - why not simply predict a discretized representation of the curve directly? In essence, the PCA parameterization of the calibration curves allows us to simplify the complex, high-dimensional data into a low-dimensional, manageable form. This approach is favored over direct prediction of the calibration curve primarily because it is difficult to predict a high-dimensional output without a large amount of training data. The low-dimensional representation therefore improves the model's generalization capabilities for new scenes. Even in cases where a large amount of training data might be available, it is unnecessary to learn this from data because, as we show in Figure 3a in the main paper and Figure 2 in the appendix, the calibration curves themselves lie on a low-dimensional subspace. To further motivate the use of the PCA, we show an example where we compare predicting the PCA coefficients to directly predicting a discretized 384-dim representation of the curve (with an MLP of size [128,128,384]). From Fig. 3 we see that the direct prediction (red "Discretized" curve) leads to a noisier curve with higher error.

2 Using the Training Set Leads to Severe Overfitting

One might consider applying the calibration technique for regression in [2] directly to NeRFs by fitting a new calibrator on the training set for each new scene instead of using the meta-calibrator introduced in our work. To show why this will not work, in this section, we present the calibration curves of the training rays for four scenes in LLFF [3] as the solid RGB curves in Fig. 4. These curves reveal that not only are the confidence levels of the pretrained NeRF model miscalibrated for the training set, but the pattern they follow is different from the

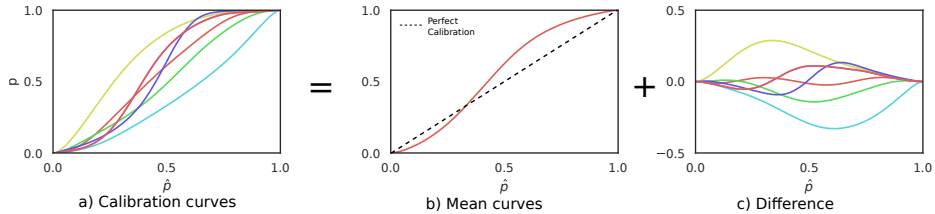


Fig. 2: Regularity in the calibration curves. This figure shows the calibration curves obtained for seven of the real-world *DTU dataset* scenes [1]. While the calibration curves vary significantly across scenes there is a high degree of regularity in this variation. We use this insight to construct a low-dimensional parameterization of the curves that our meta-calibrator can predict from scene features.

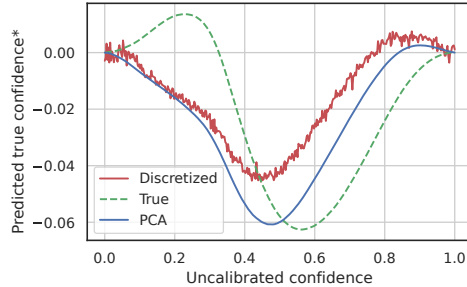


Fig. 3: PCA representation vs direct prediction of the calibration curve. Here we show an example *mean-normalized* (*) calibration prediction for the *Fern* scene from the LLFF [3] dataset. The low-dimensional PCA parameterization (blue "PCA" curve) allows the model to generalize better and better preserves the characteristics of the true calibration curves than the high-dimensional representation (red, noisy "Discretized" curve) does.

one observed in the test set, also shown in Fig. 4. The NeRF model is consistently overconfident for confidence levels closer to zero and underconfident for confidence levels closer to one for the training set but consistently underconfident for confidence levels closer to zero and overconfident for confidence levels closer to one for the test set. Thus, calibration using the training set would result in very poor generalization to the test set. Specifically, using the training set for calibration results in worse test calibration errors than leaving the NeRF model uncalibrated for all scenes in LLFF.

3 Holding Out Data Results in Poor Image Quality

While, as shown in Sec. 2, using the training set for calibration results in severe overfitting, we could also consider holding out images from the training set and using them to fit the calibrator. However, as shown in Tab. 1, this method significantly reduces the performance of the NeRF at novel view synthesis. For

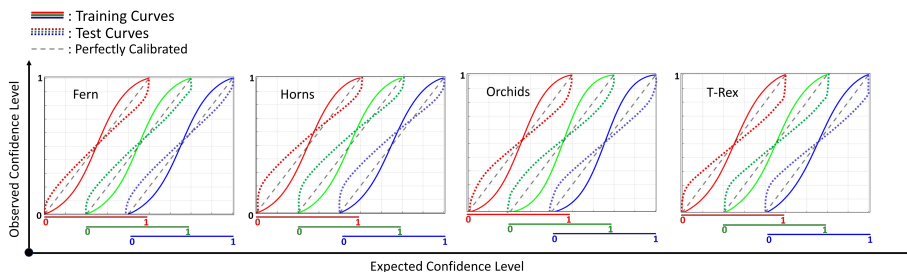


Fig. 4: Calibration curves for training and test data from four scenes in LLFF [3]. The color of each curve indicates the color channel it corresponds to. The solid red, green, and blue curves are not closely aligned with the grey dashed lines in general, showing that the pretrained NeRF model is miscalibrated, even for the training set. The expected confidence levels for the training set (solid RGB lines) follow a different pattern from the one followed by the test set (dotted RGB lines). This is apparent by observing that the dotted RGB curves are not aligned with the solid RGB curves close to zero and one. As a result, calibration with the training set (solid curves) would not generalize to the test set (dotted curves).

example, holding out just one image from the *Horns* scene in LLFF [3] reduces the PSNR by 17%. Therefore, holding out images is not an ideal technique for fitting the calibrator. Unlike holding out images, our meta-calibrator allows the NeRF model to use all available data from the target scene for training, resulting in better image quality.

Table 1: PSNR on 3-View LLFF [3] using 1, 2, and 3 views for training. Holding out views from the training set significantly reduces quality of images inferred by NeRF. This is clear by observing how small PSNRs are in row 1 (training on 1 view) vs PSNRs in row 3 (training on 3 views). Higher PSNRs indicate better image quality. Note: NeRF model was trained for 2k iterations.

Num. of Views	Room	Fern	Flower	Fortress	Horns	Leaves	Orchids	T-Rex
1	15.94	16.15	12.93	15.19	12.58	11.87	10.99	11.24
2	18.57	19.07	17.38	17.45	13.51	14.12	14.37	17.36
3	19.25	19.76	18.06	21.12	16.28	15.44	15.72	18.33

4 Number of Ray Samples’ Influence on Uncertainty Quality

The number of samples along the ray for FlipNeRF [6] determines the number of components in the Laplacian mixture model used to represent the uncertainty in

the predicted images. Intuitively, increasing the number of mixture components, and, hence, the number of ray samples, increases the precision of this representation. Supporting this concept, we show in Fig. 5 that as the number of ray samples increases, the calibration error of the base uncertainties decreases, with diminished returns after 128 samples. We use 128 ray samples for our pretrained FlipNeRF model as this produces the lowest calibration error for the base uncertainties without being as costly to train as NeRFs with higher sample counts.

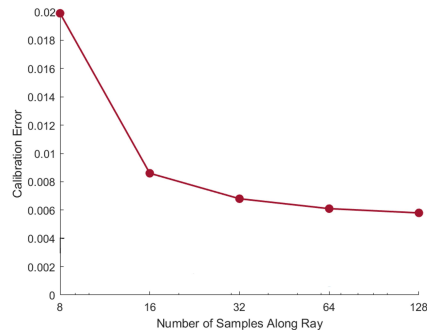


Fig. 5: Uncalibrated uncertainty quality vs number of samples along a ray for *Room* scene from LLFF [3]. The number of samples along the ray for FlipNeRF [6] determines the number of components in the Laplacian mixture model used to obtain the uncertainty in the predicted images. Higher number of samples increases the precision of the mixture model, reducing the calibration error of the base uncertainties.

5 Calibration Can Correct the Order of Pixel Uncertainties

Our meta-calibrator can re-order the pixel uncertainties even though it predicts a monotonic regression model that maps the NeRF’s expected confidences to the true ones. Here, we include a detailed theoretical example showing that such re-ordering is possible.

One might think that the order of the uncertainties is preserved by calibration as we’re fitting a monotonic curve to the expected confidences. However, this is not the case. *Rather than preserving the order of the uncertainties with respect to the pixels, the calibration preserves the monotonicity of the individual CDFs at each pixel.* To elucidate this concept, consider an example where calibration can reverse the order of uncertainties for two pixel CDFs as shown in Figure 6.

Initially, the CDF for ray 1, corresponding to pixel 1, might indicate higher uncertainty compared to ray 2 (pixel 2). However, after applying the calibration

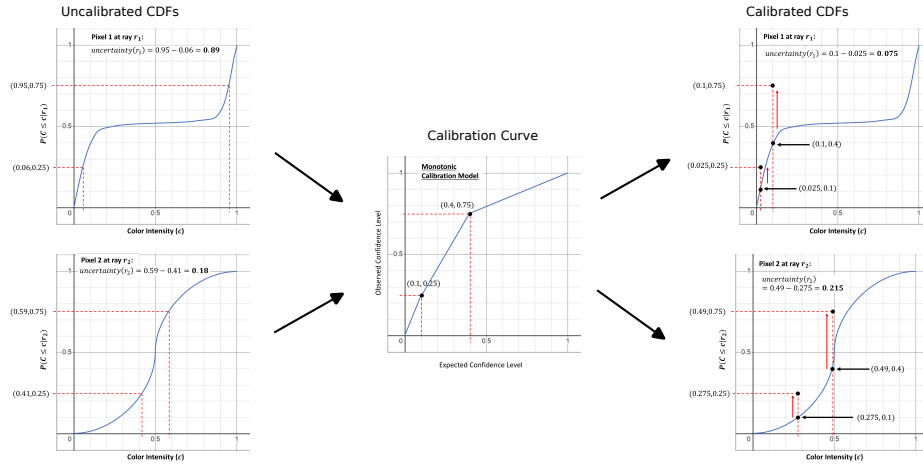


Fig. 6: Is the order of uncertainties necessarily preserved during calibration? This illustration shows that the order of uncertainties for two pixels (corresponding to rays 1 and 2) is not necessarily preserved during calibration. We start with $uncertainty(r_1) > uncertainty(r_2)$ for the left two uncalibrated CDFs and end up with $uncertainty(r_1) < uncertainty(r_2)$ after calibration on the right.

process, the order of uncertainties can be reversed. This reversal is attributed to the differing shapes and slopes of the CDFs, which are altered non-linearly during calibration. The implications of this observation are significant. It underscores the non-trivial nature of the calibration process in uncertainty modeling and suggests that calibration does not merely scale or shift uncertainties but can fundamentally alter the relation between the uncertainty values. In summary, this highlights the complexity and nuanced impact of calibration on the predicted uncertainties.

6 Efficiency of Uncertainty Metric

In this section, we provide further details on why we use the interquartile range, rather than the variance or standard deviation, to quantify the uncertainty at each pixel. While the variance of a Laplacian mixture model can be obtained in closed form from the parameters of the component distributions, in our approach, the parameters of the *calibrated* CDF (e.g., the location and scale parameters of the component CDFs) are not known. Hence, to obtain the variance of the distribution for each pixel, we would either need to sample from it or differentiate the predicted CDF to obtain the corresponding PDF and then integrate to estimate the variance. As shown in Table 2, both of the aforementioned methods are much slower than estimating the interquartile range. This is because the interquartile range can be calculated from the calibrated CDF directly.

Table 2: Timing of obtaining different uncertainty measures for the distribution of 1 pixel. Calculating the interquartile range is much faster than calculating the variance.

Uncertainty Metric	Method	Time (s)
Variance	Integration	9.807
Variance	Sampling	1.759
Interquartile Range (Ours)	Interpolation	0.008

7 Further Qualitative Examples

In this section, we provide further qualitative examples visualizing the calibrated and uncalibrated uncertainty.

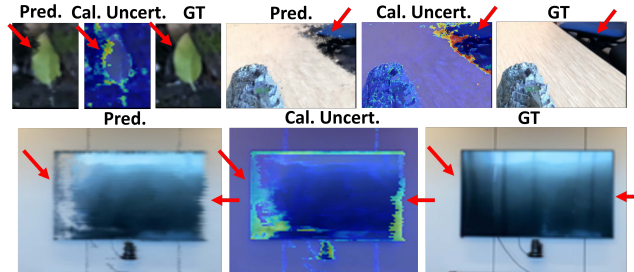


Fig. 7: Additional qualitative examples from LLFF [3]. Here, we show the NeRF render (“Pred”), calibrated uncertainties (“Cal. Uncert.”) and true image (“GT”).

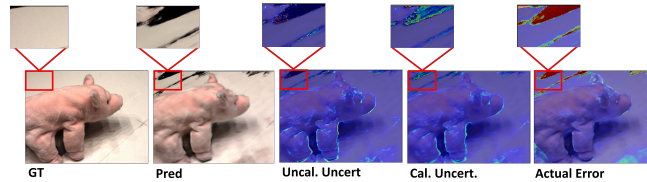


Fig. 8: Additional qualitative example from DTU [1]. Here, we show the true image (“GT”), NeRF render (“Pred”), uncalibrated uncertainties (“Uncal. Uncert.”), calibrated uncertainties (“Cal. Uncert.”), and the absolute errors (“Actual Error”). The calibrated uncertainties better correlate with the actual errors than the uncalibrated uncertainties do.

8 Limitations

In this section, we describe two limitations of our method. (1) The meta-calibrator needs to be trained once, but once trained, the frozen meta-calibrator can be applied to new scenes. (2) The quality of our calibration depends on the quality of the used image features. DINOv2 [5] works very well, but there may be other feature extractors that are more effective.

9 Ethics

We used the Realistic Synthetic 360° dataset [4], the subset of scenes in DTU [1] used in [6], and all the scenes in LLFF [3] following their terms and conditions. There is no personal data. For further details on ethics, data protection, and copyright please see <https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html>.

10 The Code

We provide documented code to train the meta-calibrator to predict the PCA coefficients for the calibration curves in the file `train_calibrator.py` and documented code to extract the DINOv2 [5] features from the uncalibrated uncertainty maps and the inferred images in the file `extract_features.py`. We plan to release the full code for our method once the paper has been accepted.

References

1. Jensen, R.R., Dahl, A., Vogiatzis, G., Tola, E., Aanaes, H.: Large scale multi-view stereopsis evaluation. In: CVPR (2014)
2. Kuleshov, V., Fenner, N., Ermon, S.: Accurate uncertainties for deep learning using calibrated regression. In: ICML. pp. 2796–2804 (2018)
3. 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)
4. 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)
5. 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)
6. Seo, S., Chang, Y., Kwak, N.: Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In: ICCV (2023)