Conditional-Flow NeRF: Accurate 3D Modelling with Reliable Uncertainty Quantification Supplementary Materials

In this Supplementary Materials, we firstly give more additional details about our CF-NeRF implementation (Sec. 1). Secondly we analyze the model size as well as time performance for all methods (Sec. 2). Then we describe a set of ablation studies to give more insights about the performance of our model (Sec. 3). Finally, we provide a set of additional qualitative results (Sec. 4).

1 Additional Implementation Details

Training details We use the same MLP-based architecture used in original NeRF [6] as a backbone network for our CF-NeRF and the rest baselines. In particular, we use 512 hidden units for all layers. During training and inference in CF-NeRF, we sample 32 radiance-density pairs for mean and variance estimation for each ray. We optimize all the models for 100,000-200,000 steps with a batch size of 512 and uniformly sampled 128 points across each ray using Adam optimizer with default hyper-parameters. For CF-NeRF, each sample from the latent prior distribution is shared for different spatial-location and viewing-direction inputs in each batch during training. To avoid overfitting with the sparse number of training views used in our experiments, we employ an additional depth loss based on [2] during optimization. This loss is weighted with a value of $1e-2$ for our method and the rest baselines. Additionally, we set a value of 0.01 as the weight for the Entropy term.

Conditional Normalizing Flows As for invertible transformation functions in our Conditional Normalizing Flow(CNF), we use the Sylvester Flows [1] defined as:

$$\mathbf{z}_k = \mathbf{z}_{k-1} + \mathbf{A}h(\mathbf{B}\mathbf{z}_{k-1} + b)\,, \tag{1}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{b}$ are flow parameters of each transformation function. Additionally, h is an hyperbolic tangent activate function. These flow parameters are conditional functions of the 5D location-direction pairs, while the samples from the latent distributions are transformed to radiance and density by sequentially using these transformation functions $f_{1:K}$. In our CF-NeRF, we use four flows for the radiance and density CNFs with the dimensions of the conditional feature into each flow set to 64.

Metrics As a metric used to assess the quality of the depth prediction, we use the δ-threshold [3]. This metric is defined as follows:

$$\%\text{ of } y_i \text{ s.t. } max(\frac{y_i^*}{y_i}, \frac{y_i}{y_i^*}) = \delta_{k=1,2,3} < \tau^{k=1,2,3}\,, \tag{2}$$

Table 1. Model size.

Add layers?	D.E. (N=5)	Drop. (N=5)	NeRF-W	S-NeRF	CF-NeRF
No	11.5M	2.31M	2.31M	2.31M	2.38M
Yes	14.1M	2.85M	2.85M	2.83M	-

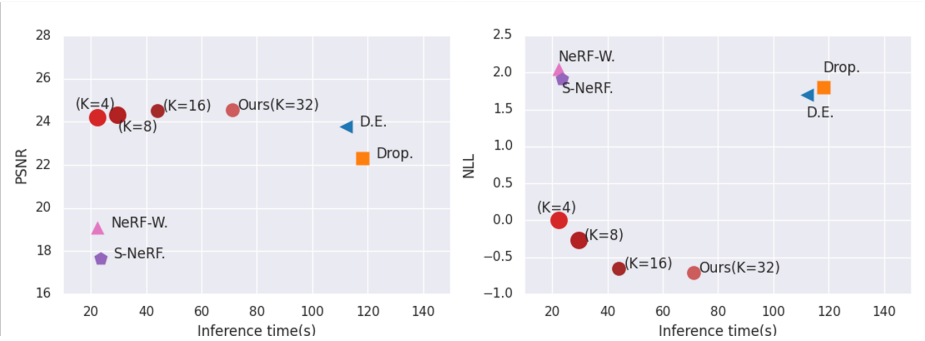

Fig. 1. Time performance vs. Accuracy.

where we set the threshold $\tau = 1.25$ as done in previous works [3]. Note that we only report δ_3 due to space limitations in the main paper.

2 Model size & time performance.

The table below shows the model size of our CF-NeRF and the rest baselines. As stated in L462, for the sake of fairness, we add additional layers for the latter so that they have a similar computational complexity than our CF-NeRF. Concretely, all the results in the paper are obtained based on the version with additional layers. From the table we can observe that adding the CNF only increases model parameters by a negligible number (∼70K) compared to the baselines without extra layers. In contrast, adding additional layers dramatically increases the model size (∼540K each).

Regarding inference time, the figure below shows the latency and performance (PSNR and NLL) of our CF-NeRF and all the compared baselines. The reported results are obtained for a 640×360 image on a 2080Ti GPU. It can be observed that our CF-NeRF performs the best both on image quality (PSNR) and uncertainty estimation (NLL) with a reasonable inference time compared to all the baselines. Moreover, properly reducing the number of samples during inference (K=8∼16) in CF-NeRF dramatically reduces the inference time with a negligible impact in performance. Last but not least, our CF-NeRF framework is enough general to be readily integrated in future work (L74) with other efficient NeRF variants like Voxel-based [7] .

Table 2. Results of our ablation studies: Quality and uncertainty quantification metrics on rendered images and depth-maps over LF dataset. Best results are shown in bold. See text for more details.

	Methods	Quality Metrics			Uncertainty Metrics		
		PSNR↑	SSIM↑	LPIPS↓	AUSE RMSE↓	AUSE MAE↓	NLL↓
RGB images	CF-NeRF w/o Entropy	23.40	0.81	0.258	0.068	0.048	-0.448
	CF-NeRF w/ Single Flow	23.82	0.83	0.228	0.081	0.039	-0.578
	CF-NeRF	**24.78**	**0.86**	**0.168**	**0.051**	**0.026**	**-0.710**
		RMSE↓	MAE↓	δ_3 ↑	AUSE RMSE↓	AUSE MAE↓	NLL↓
Depth	CF-NeRF w/o Entropy	0.121	0.078	0.76	0.224	0.143	7.88
	CF-NeRF w/ Single Flow	0.170	0.111	0.64	0.229	0.138	8.16
	CF-NeRF	**0.118**	**0.074**	**0.81**	**0.110**	**0.071**	**5.09**

3 Ablations

In order to give insights into some design decisions of our proposed CF-NeRF, we provide results for two ablation experiments. Concretely, we conduct experiments over the LF dataset. Results are shown in Table 2. In the following, we describe each of the experiments in more detail.

Entropy term. We remove the entropy term and train our CF-NeRF only using the NLL as the training loss. On both generated RGB images and depth-maps, we achieve better performance by using the Entropy term as well across all metrics, including the prediction error and its associated uncertainty. This is consistent with what we have discussed in the main paper that, maximizing the Entropy term intuitively prevents the optimized distribution to degenerate into a deterministic function where all the probability is assigned into a single radiance field $\mathcal{F}$, thus losing the ability to quantify correct uncertainty.

Single Flow. Our CF-NeRF uses two conditional normalizing flows(CNF) for modelling the distribution of radiance and density. However, a more efficient strategy could be to jointly model their distributions using a single flow in order to take into account the possible dependence between them. As we can see in Table 2, this variant obtains worse performance compared to our CF-NeRF with two CNFs in terms of prediction quality and uncertainty estimation. This drop in performance is especially high in the case of depth-map estimation. This can be explained because using a single CNF for radiance and density distribution contradicts the fact that the volume density must be independent of the emitted radiance to obtain optimal results, as was previously discussed in [6,8].

4 More Results

Interpolation videos An intuitive advantage of the explicit distribution modelling over the radiance fields in our CF-NeRF is that, we can conveniently analyze the learned radiance fields by interpolating in the latent space [5,4]. The shared latent variable allows to model the joint distribution of all the radiance-density pairs in the scene in contrast to S-NeRF and hence could avoid the noisy

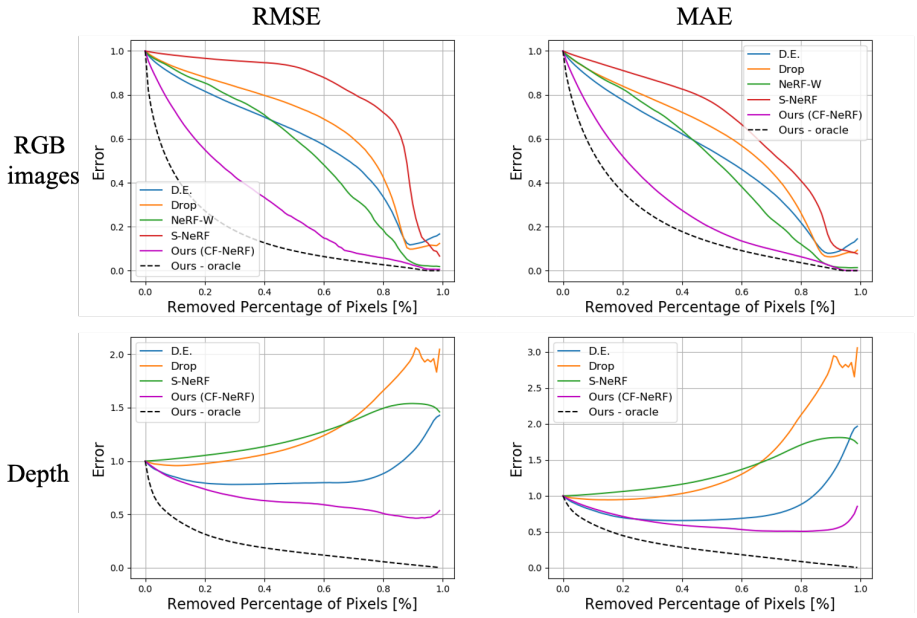

Fig. 2. Sparsification curves obtained by different methods of estimating uncertainty associated with rendered RGB images and estimated depth.

results. More formally, we define the interpolation value as,

$$f_L(\mathbf{z}_1, \mathbf{z}_2, \lambda) = \lambda \mathbf{z}_1 + (1 - \lambda)\mathbf{z}_2, \tag{3}$$

where $\mathbf{z}_1$ and $\mathbf{z}_2$ are two random samples from the latent distribution with $\lambda \in [0, 1]$. Then the density and radiance can be obtained through our proposed CNF, following our inference process to render novel views and depth. To see the dynamic interpolation results we provide a video attached to this supplementary material. By looking at different frames in the dynamic interpolated results, S-NeRF tends to generate noisy image and depth predictions with random and incoherent changes between adjacent frames obtained using two adjacent interpolation values. In contrast, our CF-NeRF can generate more coherent and smoothly changing frames, both on rendered RGB images and estimated depth-maps. This clearly demonstrates the advantages of our proposed Latent Variable Modelling for CF-NeRF in order to efficiently model the joint distribution over all the possible radiance and density pairs in the scene.

Sparsification plots Fig. 2 shows the additional related sparsification curves on the synthetic novel views and estimated depth averagely over the LF dataset. Note that NeRF-W is not able to estimate uncertainty on depth and hence cannot generate the sparsification curve on depth. When evaluated over all pixels, all methods perform similarly. As we remove the pixels with high uncertainty from

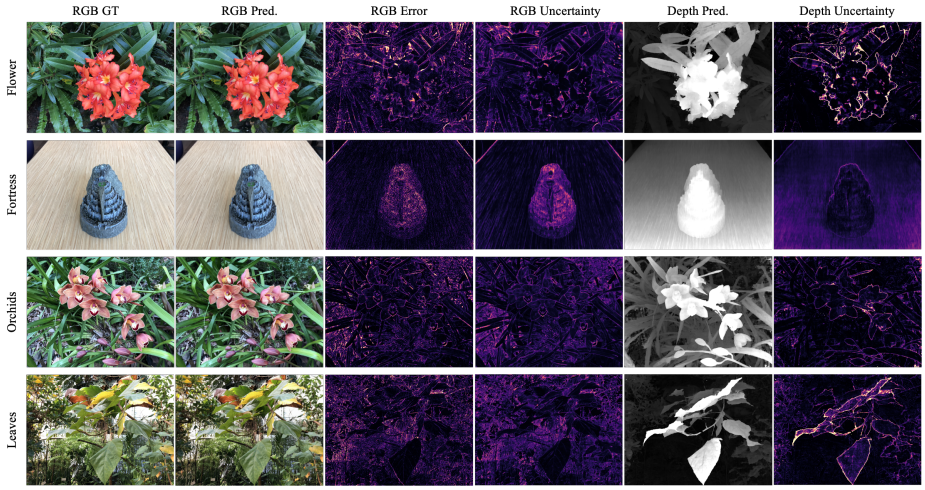

Fig. 3. More qualitative results obtained by our CF-NeRF over LLFF dataset.

1% to 100%, our method always obtains the lowest value and fits closest with the oracle curve. This demonstrates that our estimated uncertainty correlates significantly better with the prediction error than the others.

More qualitative results Fig. 3 shows more qualitative results obtained by our CF-NeRF for the scenes in the simple LLFF dataset. Moreover, Fig. 4 shows additional qualitative results obtained by our CF-NeRF across other scenes in the LF dataset: *Africa, Statue, Torch*. For each scene, we show not only the predicted RGB views and the estimated depth-maps, but also their associated uncertainty estimations.

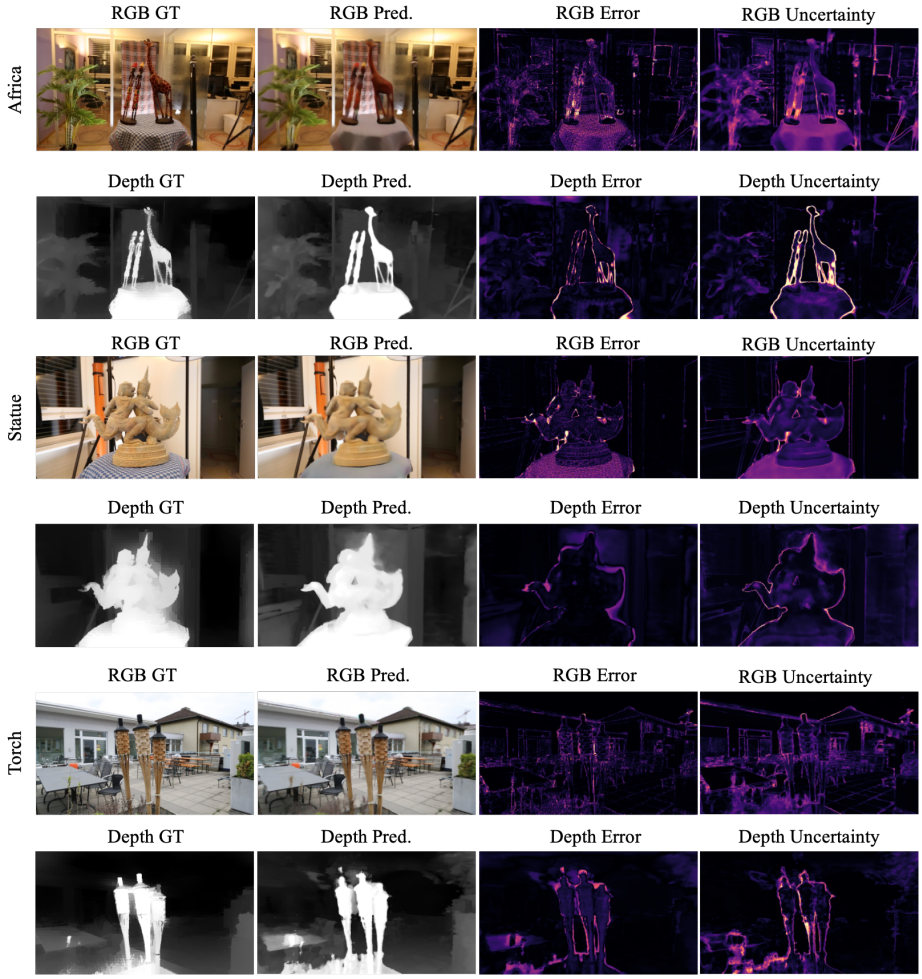

Fig. 4. More results obtained by our CF-NeRF over LF dataset.

References

1. van den Berg, R., Hasenclever, L., Tomczak, J., Welling, M.: Sylvester normalizing flows for variational inference. In: proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI) (2018)
2. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free (2021)
3. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NIPS (2014)
4. Lesniak, D., Sieradzki, I., Podolak, I.T.: Distribution-interpolation trade off in generative models. In: ICLR (2019)
5. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2013)
6. 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)
7. Yu, A., Fridovich-Keil, S., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks (2021)
8. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields (2020)