ActiveNeRF: Learning where to See with Uncertainty Estimation - Supplementary Material

Xuran Pan[1], Zihang Lai[2†], Shiji Song[1], and Gao Huang[1✉]

[1] Tsinghua University, Beijing 100084, China
pxr18@mails.tsinghua.edu.cn {shijis, gaohuang}@tsinghua.edu.cn
[2] Carnegie Mellon University, Pennsylvania 15213, United States
zihangl@andrew.cmu.edu

A. Derivation of Posterior Distribution (Eq.(17))

Given prior distribution as:

$$P^{(\text{pri})} = P(c(\mathrm{r}_2(t_k))|D_1) \sim \mathcal{N}(\bar{c}(\mathrm{r}_2(t_k)), \bar{\beta}^2(\mathrm{r}_2(t_k))). \tag{1}$$

Given new data distribution as:

$$p(C(\mathrm{r}_2)|c(\mathrm{r}_2(t_k))) \sim \mathcal{N}(\alpha_k \bar{c}(\mathrm{r}_2(t_k)) + b(t_k), \bar{\beta}^2(\mathrm{r}_2)). \tag{2}$$

The corresponding posterior distribution can be formulated as:

$$P^{(\text{post})} \sim \mathcal{N}\left(\gamma \frac{C(\mathrm{r}_2) - b(t_k)}{\alpha_k} + (1-\gamma)\bar{c}(\mathrm{r}_2(t_k)), \frac{\bar{\beta}^2(\mathrm{r}_2(t_k))\bar{\beta}^2(\mathrm{r}_2)}{\alpha_k^2 \bar{\beta}^2(\mathrm{r}_2(t_k)) + \bar{\beta}^2(\mathrm{r}_2)}\right), \tag{3}$$

$$\text{with} \quad \gamma = \frac{\alpha_k^2 \bar{\beta}^2(\mathrm{r}_2(t_k))}{\alpha_k^2 \bar{\beta}(\mathrm{r}_2(t_k)) + \bar{\beta}^2(\mathrm{r}_2)}. \tag{4}$$

Proof.

$$P^{(\text{post})} = P(c(\mathrm{r}_2(t_k))|D_1, \mathrm{r}_2) = \frac{p(C(\mathrm{r}_2)|c(\mathrm{r}_2(t_k)))p(c(\mathrm{r}_2(t_k))|D_1)}{\int p(C(\mathrm{r}_2)|c(\mathrm{r}_2(t_k)))p(c(\mathrm{r}_2(t_k))|D_1) dc(\mathrm{r}_2(t_k))} \tag{5}$$

$$\propto \exp\left(-\frac{(C(\mathrm{r}_2) - \alpha_k c(\mathrm{r}_2(t_k)) - b(t_k))^2}{2\bar{\beta}^2(\mathrm{r}_2)}\right) \exp\left(-\frac{(c(\mathrm{r}_2(t_k) - \bar{c}(\mathrm{r}_2(t_k)))^2}{2\bar{\beta}^2(\mathrm{r}_2(t_k))}\right) \tag{6}$$

$$\propto \exp\left(-\frac{1}{2}\frac{\alpha_k^2 \bar{\beta}^2(\mathrm{r}_2(t_k)) + \bar{\beta}^2(\mathrm{r}_2)}{\bar{\beta}^2(\mathrm{r}_2(t_k))\bar{\beta}^2(\mathrm{r}_2)}(c(\mathrm{r}_2(t_k)) - \gamma \frac{C(\mathrm{r}_2) - b(t_k)}{\alpha_k} - (1-\gamma)\bar{c}(\mathrm{r}_2(t_k)))^2\right). \tag{7}$$

† Work done during an internship at Tsinghua University.
✉ Corresponding author.

B. Derivation of Posterior Distribution with Multiple New Data (Eq.(22))

For simplicity, we consider the situation with two new inputs, and the derivations for more inputs are similar. We change some symbols in the main paper with short notations. The derivation in the main paper can be applied readily with simple substitutions. For $\mathrm{x} = \mathrm{r}_1(t_1) = \mathrm{r}_2(t_2)$, given prior distribution as:

$$P^{(\mathrm{pri})} = P(c(\mathrm{x})|D_1) \sim \mathcal{N}(\bar{c}(\mathrm{x}), \beta_0^2). \tag{8}$$

Given new data distribution as:

$$p(\mathrm{r}_1|\mathrm{x}) \sim \mathcal{N}(\alpha_1 \mathrm{x} + b_1, \beta_1^2), \tag{9}$$
$$p(\mathrm{r}_2|\mathrm{x}) \sim \mathcal{N}(\alpha_2 \mathrm{x} + b_2, \beta_2^2). \tag{10}$$

The corresponding posterior distribution can be formulated as:

$$P^{(\mathrm{post})} \sim \mathcal{N}\left(\gamma_1 \frac{C(\mathrm{r}_1) - b_1}{\alpha_1} + \gamma_2 \frac{C(\mathrm{r}_2) - b_2}{\alpha_2} + \gamma_3 \bar{c}(\mathrm{x}), \gamma_3 \beta_0^2\right), \tag{11}$$

where

$$\gamma_1 = \frac{\alpha_1^2 \beta_0^2 \beta_2^2}{\alpha_1^2 \beta_0^2 \beta_2^2 + \alpha_2^2 \beta_0^2 \beta_1^2 + \beta_1^2 \beta_2^2}, \tag{12}$$
$$\gamma_2 = \frac{\alpha_2^2 \beta_0^2 \beta_1^2}{\alpha_1^2 \beta_0^2 \beta_2^2 + \alpha_2^2 \beta_0^2 \beta_1^2 + \beta_1^2 \beta_2^2}, \tag{13}$$
$$\gamma_3 = \frac{\beta_1^2 \beta_2^2}{\alpha_1^2 \beta_0^2 \beta_2^2 + \alpha_2^2 \beta_0^2 \beta_1^2 + \beta_1^2 \beta_2^2}. \tag{14}$$

Proof.

$$P^{(\mathrm{post})} = P(c(\mathrm{x})|D_1, \mathrm{r}_1, \mathrm{r}_2) = \frac{p(C(\mathrm{r}_1)|c(\mathrm{x}))p(C(\mathrm{r}_2)|c(\mathrm{x}))p(c(\mathrm{x})|D_1)}{\int p(C(\mathrm{r}_1)|c(\mathrm{x}))p(C(\mathrm{r}_2)|c(\mathrm{x}))p(c(\mathrm{x})|D_1) dc(\mathrm{x})} \tag{15}$$

$$\propto \exp\left(-\frac{(C(\mathrm{r}_1) - \alpha_1 c(\mathrm{x}) - b_1)^2}{2\beta_1^2} - \frac{(C(\mathrm{r}_2) - \alpha_2 c(\mathrm{x}) - b_2)^2}{2\beta_2^2} - \frac{(c(\mathrm{x}) - \bar{c}(\mathrm{x}))^2}{2\beta_0^2}\right) \tag{16}$$

$$\propto \exp\left(-\frac{1}{2}\frac{1}{\gamma_3 \beta_0^2}(c(\mathrm{x}) - \gamma_1 \frac{C(\mathrm{r}_1) - b_1}{\alpha_1} - \gamma_2 \frac{C(\mathrm{r}_2) - b_2}{\alpha_2} - \gamma_3 \bar{c}(\mathrm{x}))^2\right). \tag{17}$$

C. Model Architectures

The detailed model architecture for ActiveNeRF is similar to the original NeRF [1]. We first use the positional encoding function to generate high frequency details. Specifically, the encoding function is formulated as:

$$\gamma(p) = (\sin(2^0 \pi p), \cos(2^0 \pi p), \cdots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p)), \tag{18}$$

which separately applied to 3D coordinates x, y, z and Cartesian viewing direction d_x, d_y, d_z. We follow the configurations in NeRF and set $L = 10$ for coordinates, and $L = 4$ for directions.

A MLP with 8 fully-connected layers is first adopted to process the encoded 3D coordinates, with residual connection on the 4_{th} layer. Then, a single fully connected layer is adopted to predict the volume density σ. Different from original NeRF, we add an additional fully connected layer with softplus activation function to predict the corresponding variance. The latent feature is then concatenated with encoded viewing directions to produce RGB color.

D. Training Configurations.

In our experiments, we follow the settings in NeRF, and sample 64, 128 points for coarse and fine models respectively. We use the Adam optimizer with an initial learning rate at $5e^{-4}$ which decays exponentially to $5e^{-5}$ during optimization. We use a batch size of 1024 rays and train our model on a single RTX2080Ti GPU.

E. Additional Qualitative Results

We provide additional visualization results for static and active scenarioes, as shown in Figure 1$\sim$3.

F. Ablation Study

We further evaluate the effectiveness of the candidate evaluation module alone. We substitute the acquisition function with heuristic approaches while keeps the uncertainty module. We denote the baseline approaches as *ActiveNeRF + Random* and *ActiveNeRF + FVS* as introduced in the main paper. The results are shown in Table 1 and ActiveNeRF outperforms other approaches consistently.

Table 1. Ablation on acquisition function : *Setting I* is same as the setting I in the Table 2 of main paper, which includes 4 initial observations and 4 extra observations obtained at 40K, 80K,120K and 160K iterations.

Sampling Rate (r)	PSNR↑	SSIM↑	LPIPS↓
Setting I, 20 total observations:			
ActiveNeRF (Full)	**26.24**	**0.856**	**0.124**
ActiveNeRF+FVS	26.05	0.852	0.146
ActiveNeRF+Random	24.77	0.801	0.188

We also investigate the performance of ActiveNeRF when the new perceptions are evaluated in a lower resolution. It can be seen in Table 2 that our model can still achieve competitive performances with sampling rate as 5, and reduces the time consumption. When the sampling rate becomes larger than 10, the quality of novel view synthesis is gradually affected.

Table 2. Ablation on sampling rate

Sampling Rate (r)	Time	PSNR↑	SSIM↑	LPIPS↓
Setting I, 20 total observations:				
1 (Full)	2.20h	**26.24**	**0.856**	**0.124**
5	2.12h	26.12	0.855	0.124
10	2.10h	25.15	0.812	0.135
20	2.09h	24.67	0.799	0.167

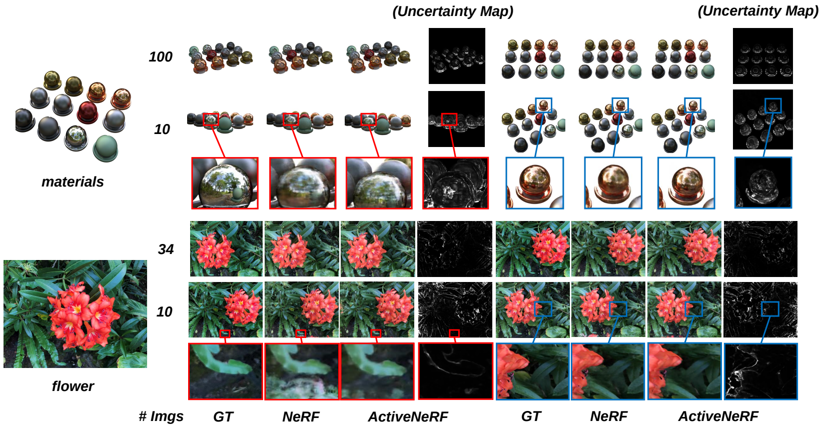

Fig. 1. Qualitative results on synthetic and realistic scenes with different fractions of training samples. Several observations can be made: First, ActiveNeRF performs significantly better than NeRF in the low-shot setting (*e.g.,* See Ln. 2 and 3). Also, ActiveNeRF and NeRF obtain similar qualitative performance when all images are used (See Ln. 1 and 4), suggesting modeling uncertainty has no negative impact on the quality of view synthesis.

References

1. 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: European conference on computer vision. pp. 405–421. Springer (2020)

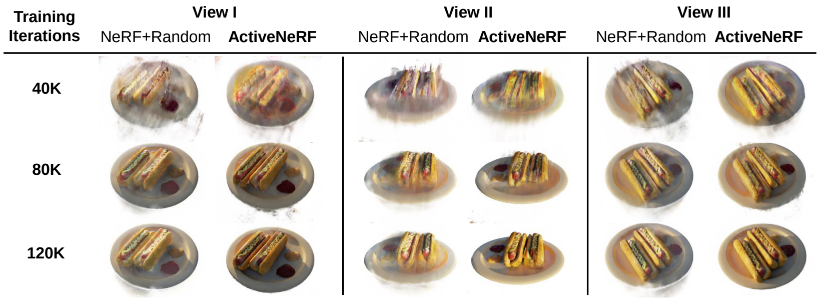

Fig. 2. Comparison of ActiveNeRF with *NeRF+Random*. The initial training set has 4 images, and we capture 4 new perceptions every 40K iterations. Our candidate evaluation function is proved to outperform heuristic approaches on the quality of view synthesis.

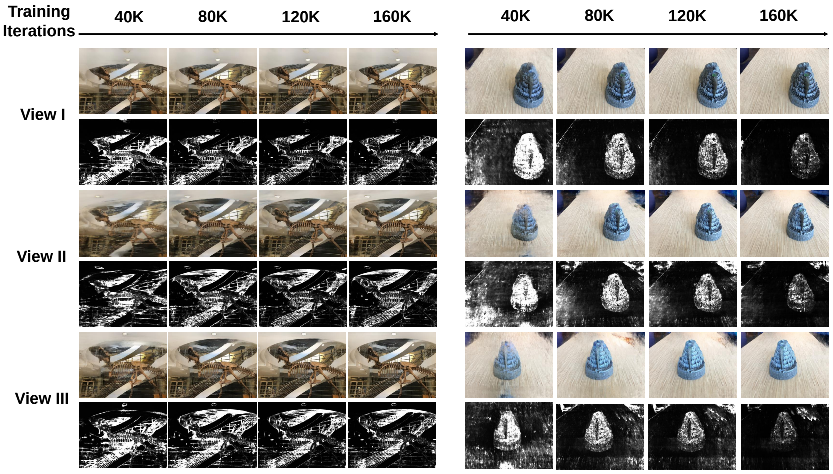

Fig. 3. Qualitative results of ActiveNeRF with active iterations on realistic scenes. We capture new perceptions every 40K iterations. Improved synthesis quality can be observed at unobserved regions.