FisherRF: Active View Selection and Uncertainty Quantification for Radiance Fields using Fisher Information

Supplementary Material

This supplementary material presents proofs for our equations and algorithm in Sec. 6. Then, we give more implementation details and results on view selections in Sec. 7. Finally, we introduce more implementation details and results on uncertainty quantifications in Sec. 8.

6. Proof of Equations in the Main Paper

Active Learning with Fisher Information has been widely studied in Machine Learning and Deep Learning in previous literatures [1, 2, 15, 17]. We provide proofs for the key equations in the main paper for completeness. Most of our formulations and notations are inspired by Kirsch *et al*. [14], who unified previous active learning approaches via Fisher Information.

Proof for Eq. 7 We compute the expected information gain of acquisition samples with:

$$\mathcal{I}[\mathbf{w}^*; \{\mathbf{y}_i^{acq}\} | \{\mathbf{x}_i^{acq}\}, D^{train}]$$
$$= H[\mathbf{w}^* | D^{train}] - H[\mathbf{w}^* | \{\mathbf{y}_i^{acq}\}, \{\mathbf{x}_i^{acq}\}, D^{train}] \tag{14}$$
$$\approx -\frac{1}{2} \log \det \mathbf{H}''[\mathbf{w}^* | D_{train}] - \left(-\frac{1}{2} \log \det \mathbf{H}''[\mathbf{w}^* | \{\mathbf{y}_i^{acq}\}, \{\mathbf{x}_i^{acq}\}, D^{train}]\right) \tag{15}$$
$$= -\frac{1}{2} \log \det \mathbf{H}''[\mathbf{w}^* | D_{train}] + \left(\frac{1}{2} \log \det \left(\mathbf{H}''\left[\{\mathbf{y}_i^{acq}\} | \{\mathbf{x}_i^{acq}\}, \mathbf{w}^*\right] + \mathbf{H}''[\mathbf{w}^* | D^{train}]\right)\right) \tag{16}$$
$$= \frac{1}{2} \log \det \left(\mathbf{H}''[\{\mathbf{y}_i^{acq}\} | \{\mathbf{x}_i^{acq}\}, \mathbf{w}^*]\ \mathbf{H}''[\mathbf{w}^* | D^{train}]^{-1} + I\right) \tag{17}$$
$$\leq \frac{1}{2} \operatorname{tr} \left(\mathbf{H}''[\{\mathbf{y}_i^{acq}\} | \{\mathbf{x}_i^{acq}\}, \mathbf{w}^*]\ \mathbf{H}''[\mathbf{w}^* | D^{train}]^{-1}\right) \tag{18}$$

where we apply Bayes' theorem in Eq. 16. We can derive Eq. 18 because the Hessian matrix $\mathbf{H}''$ is symmetric, positive semidefinite. And for any symmetric, positive semidefinite matrices A with eigenvalues λ_i:

$$\log \det(A + I) <= \log \prod_i (\lambda_i + 1) \tag{19}$$
$$= \sum_i \log(\lambda_i + 1) \leq \sum_i \lambda_i = tr(A) \tag{20}$$

the equality holds when $A = 0$.

Proof of Eq. 10 - 11 .

Let $\mathbf{z} = f(\mathbf{x}; \mathbf{w}^*)$ be the rendering result of our model.

$$\mathbf{H}''[\mathbf{y} | \mathbf{x}, \mathbf{w}^*] = \operatorname{Cov}\left[\mathbf{H}'[\mathbf{y} | \mathbf{x}, \mathbf{w}^*]\right] \tag{21}$$
$$= \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)^T \operatorname{Cov}\left[\nabla_{\mathbf{z}} H[\mathbf{y} | \, \mathbf{z}]\right] \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*) \tag{22}$$
$$= \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)^T \mathop{\mathbb{E}}_{p(y|x,w^*)} \left[\nabla_{\mathbf{z}}^2 H[\mathbf{y} | \, \mathbf{z}]\right] \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*) \tag{23}$$

Please note we use H to notate entropy, $\mathbf{H}'$ for Jacobian and $\mathbf{H}''$ for the Hessians of log probability. As our log probability function is a Gaussian error function defined in Eq. 4, $p(\mathbf{y} | \, \mathbf{z} = f(\mathbf{x}; \mathbf{w}^*)) \sim \mathcal{N}(\mathbf{y}; \mathbf{z}, 1)$. Thus $\mathbb{E}_{p(y|x,w^*)} \left[\nabla_{\mathbf{z}}^2 H[\mathbf{y} | \, \mathbf{z}]\right] = 1$ for any $\mathbf{y}$ and $\mathbf{z}$. Therefore:

$$\nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)^T \mathop{\mathbb{E}}_{p(y|x,w^*)} \left[\nabla_{\mathbf{z}}^2 H[\mathbf{y} | \, \mathbf{z}]\right] \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)$$
$$= \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)^T \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*) \tag{24}$$

7. More Implementation Details and Qualatative Results on Active View Selection

Implementation Details of Active View Selections with the 3D Gaussian Splatting Backend We use random seed 0 for all experiments. The initial views are uniformly sampled based on the translation vector of all camera poses. The code will be made public soon. To prevent overfitting in the initial stages, the training process for parameters of spherical harmonics in the original 3D Gaussian Splatting begins by optimizing only the zero-order component. Subsequently, one band of spherical harmonics is introduced after every 1,000 iterations until all four bands of spherical harmonics are activated [13]. 3D Gaussian Splatting is more prone to overfitting in our case, especially in the background of real-world datasets, because we have much fewer views (20 views vs. around 150 views). Therefore, we introduce one band of spherical harmonics every 5,000 iterations. This change is applied to all the models, so the baseline models are also benefited. Following the original training procedure of 3D Gaussian Splatting, all the models are trained for 30,000 iterations.

We provide more visualizations of our method with the 3D Gaussian Splatting backend in Fig. 8 and Fig. 9. Our model could select the most informative views to avoid the degeneration of 3D Gaussian Splatting models when the number of viewpoints is highly limited.

Method	Bicycle	Counter	garden	kitchen	room	Stump	TreeHill	Bonsai	Flowers
ActiveNeRF [25]	12.63	11.69	13.69	12.15	NaN	15.49	NaN	12.77	11.65
3D Gaussian + ActiveNeRF	18.08	17.76	19.91	20.15	20.32	18.14	15.71	19.32	12.90

Table 5. **Quatitative Comparison between the Original Implementation of ActiveNeRF and Our Reimplementation on Mip-360 Dataset** Here we compare the PSNR of our implementation of ActiveNeRF on 3D Gaussian Splatting and the original ActiveNeRF model on MipNeRF-360 dataset. Our implementation performs better than the original implementation on all the scenes. We found the original implementation of ActiveNeRF provided by the authors is prone to collapse on the MipNeRF-360 dataset even after multiple reruns, producing NaN (Not a Number) results in this table for respective splits.

Details about our reimplementation of ActiveNeRF with 3D Gaussian Splatting and Plenoxels For the ActiveNeRF implementation on 3D Gaussian splatting, we assign each 3D Gaussian with an additional variance parameter σ^2 and follow the original rendering equation [13, 22, 25, 29] to compute the variance of each pixel. Similarly, each grid vertex is assigned a variance parameter in our reimplementation of ActiveNeRF with Plenoxels. The variance parameters are updated along with other model parameters during training. When performing active view selection, we select views with the greatest variance reduction following the original paper [25]. To validate our reimplementation, we compare our implementation with the original ActiveNeRF implementation on the MipNeRF-360 dataset in Table 5. Our reimplementation of ActiveNeRF is much better than the original implementation of ActiveNeRF on MipNeRF-360 datasets and Blender Dataset. The comparative study on the Blender Dataset has been provided in Table. 1 from our main paper.

Details about our implementation with Plenoxels backend The initial views are uniformly sampled based on the translation vector of all camera poses. For the Blendar dataset, we initialize the grid resolution to 256 and upsample the grid to 512 in the middle of the training progress. For the 20-view case, we train the model from 4 initial views and sample 4 views every 4 epochs with a total of 20 epochs. For the 10-view case, we train the model from 2 initial views and sample one more training view every four epochs with a total of 36 epochs. The learning rate for density is initially set to 30 and linearly decreases to 0.05. The learning rate for spherical harmonics is initially set to 1e-2 and then linearly decreases to 5e-6. During grid upsampling, in order to prevent out-of-memory, we filtered voxels with opacity smaller than 5e-3 and kept the number of voxels less than 22 million. All the other training settings remain the same as the original implementation. We only conducted comparative studies on the Blender Dataset for our Plenoxel backend as we found Plenoxels cannot produce valid results when trained with limited viewpoints on real-world datasets like MipNeRF360. We provide more qualitative comparisons in Fig. 10 and Fig. 11. We also provided an enlarged figure of the Hessian matrix to help readers better understand the distributions and sparsity of the Hessian matrix in Radiance Field models in Fig. 12.

	Statue ↓	Africa ↓	Torch ↓	Basket ↓	Average ↓
CF-NeRF	0.43	0.29	0.41	0.10	0.31
Ours	0.21	0.26	0.24	0.18	0.22

Table 6. **Quantitative Results on Uncertainty Estimation on LF Dataset with the Same Setting as CF-NeRF [32]** Numbers are AUSE; We use the same training and test view as the CF-NeRF. Although 3D Gaussian Splatting is not designed for forward-facing scenes, our uncertainty estimation algorithm outperformed previous state-of-the-art.

8. More Details and Results on Uncertainty estimation

We compare our method with CF-NeRF on the LF dataset under two settings. One is to select the first view in every ten views as the training set (360°). The other is to use the view indices in CF-NeRF paper (Original) [32]. Due to limited space, we only presented the results of the 360° setting in the main paper. Here we present the results of the original settings of CF-NeRF in Table 6.

In line with prior approaches in uncertainty estimation [31, 32], we conducted evaluations on the Light Field (LF) Dataset [39] using the Area Under Sparsification Error (AUSE) metric. This metric involves a two-step pixel filtering process: first, pixels are filtered based on their absolute error with respect to the ground truth depth, and then they are filtered based on their uncertainty values. The disparity in the mean absolute error among the remaining pixels resulting from these two sparsification steps yields two distinct error curves. The AUSE is subsequently computed as the area between these two curves, providing an assessment of the correlation between uncertainties and the predicted errors. As we do not have a view selection process in uncertainty quantification benchmark, we train the 3D Gaussian Splatting models for 3,000 iterations and the maximum degree of spherical harmonics is set to 2 to prevent overfitting. For CF-NeRF, we use the official implementation to train models from scratch in the LF dataset, as the author did not provide checkpoints for every scene. The error during sparsification is normalized before area calculation. To calculate the area under curves in the AUSE metric, we sampled 100 points and used the trapezoid method to calculate the area under the curve. The qualitative comparisons are in Figure 13. The uncertainty visualization shows that our method can produce a more reasonable estimation of uncertainty, especially for background. For example, in the *statue* scene, our method gives high uncertainty to the closet in the background, which also has a high depth error, while CF-NeRF gives low uncertainty.

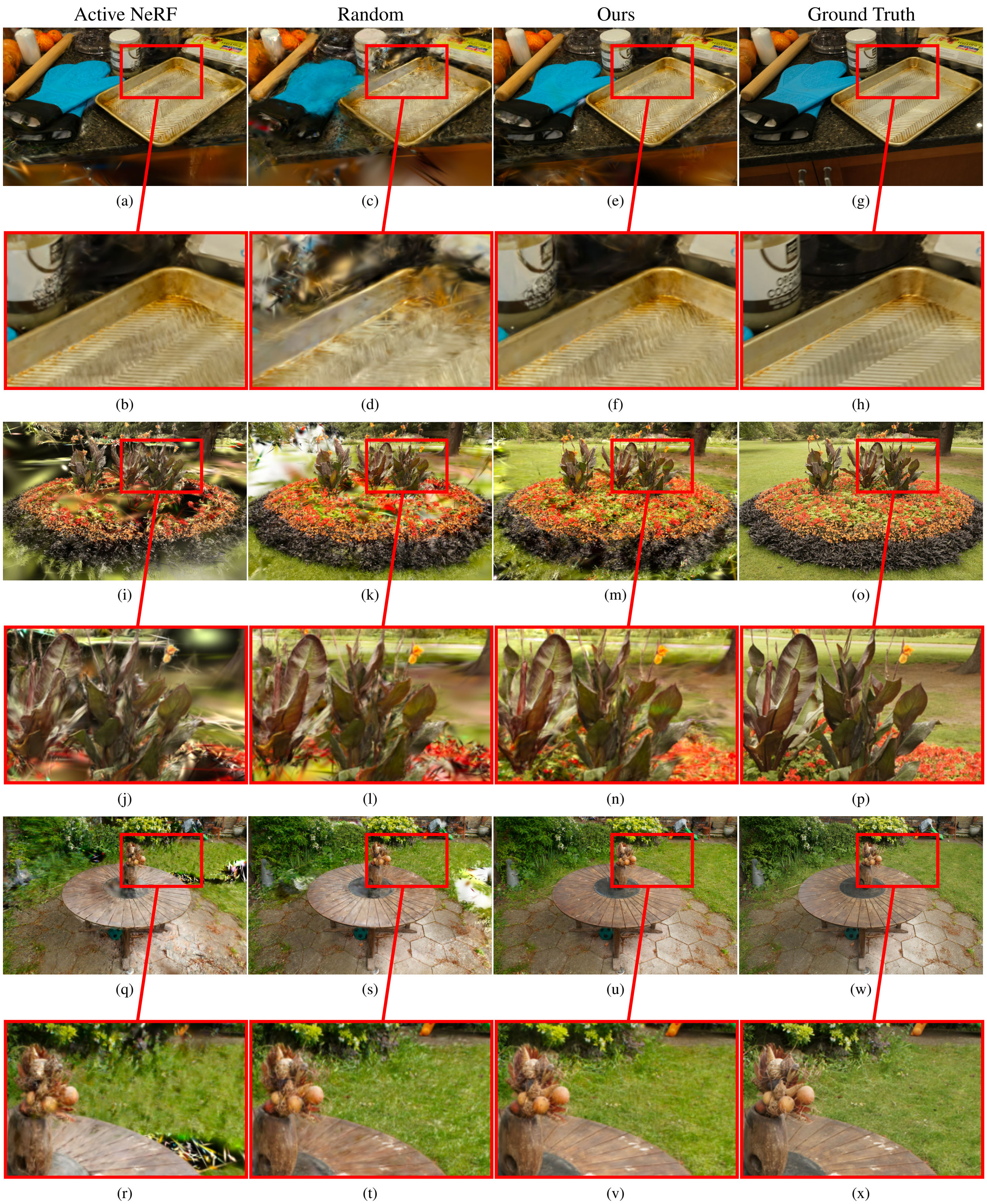

Figure 8. **Zoomed-in Qualitative Study of Our Method on MipNeRF-360 Dataset** Every second rows are zoom-in figures. Visualizations are the results of the test set after being trained with 20 training views. All the methods have the same 3D Gaussian Splatting Backend except for different view selection algorithms.

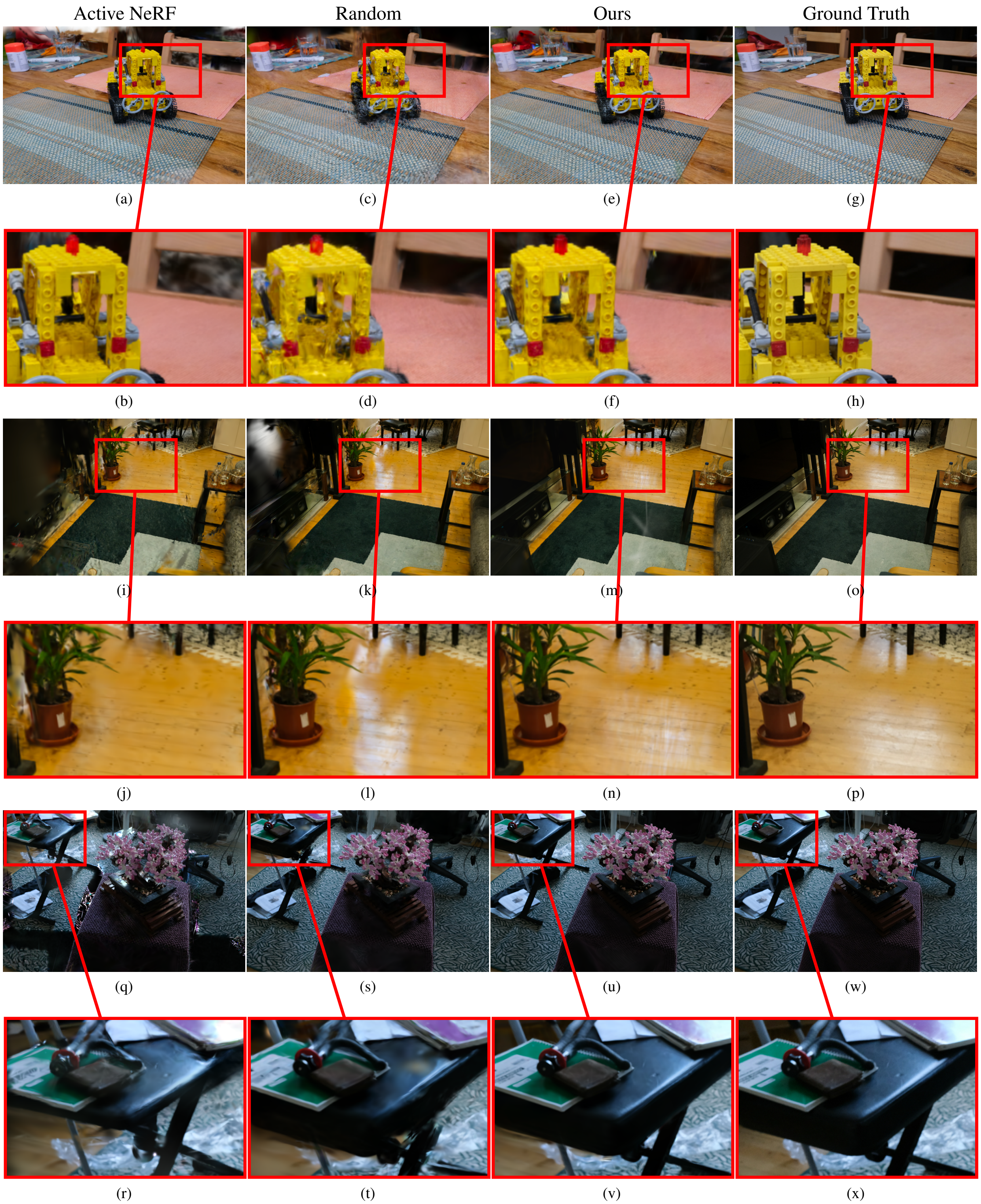

Figure 9. **Zoomed-in Qualitative Study of our method on Mip360 Dataset(cont.)** Every second rows are zoom-in figures. Visualizations are the results of the test set after being trained with 20 training views. All the methods have the same 3D Gaussian Splatting Backend except for different view selection algorithms.

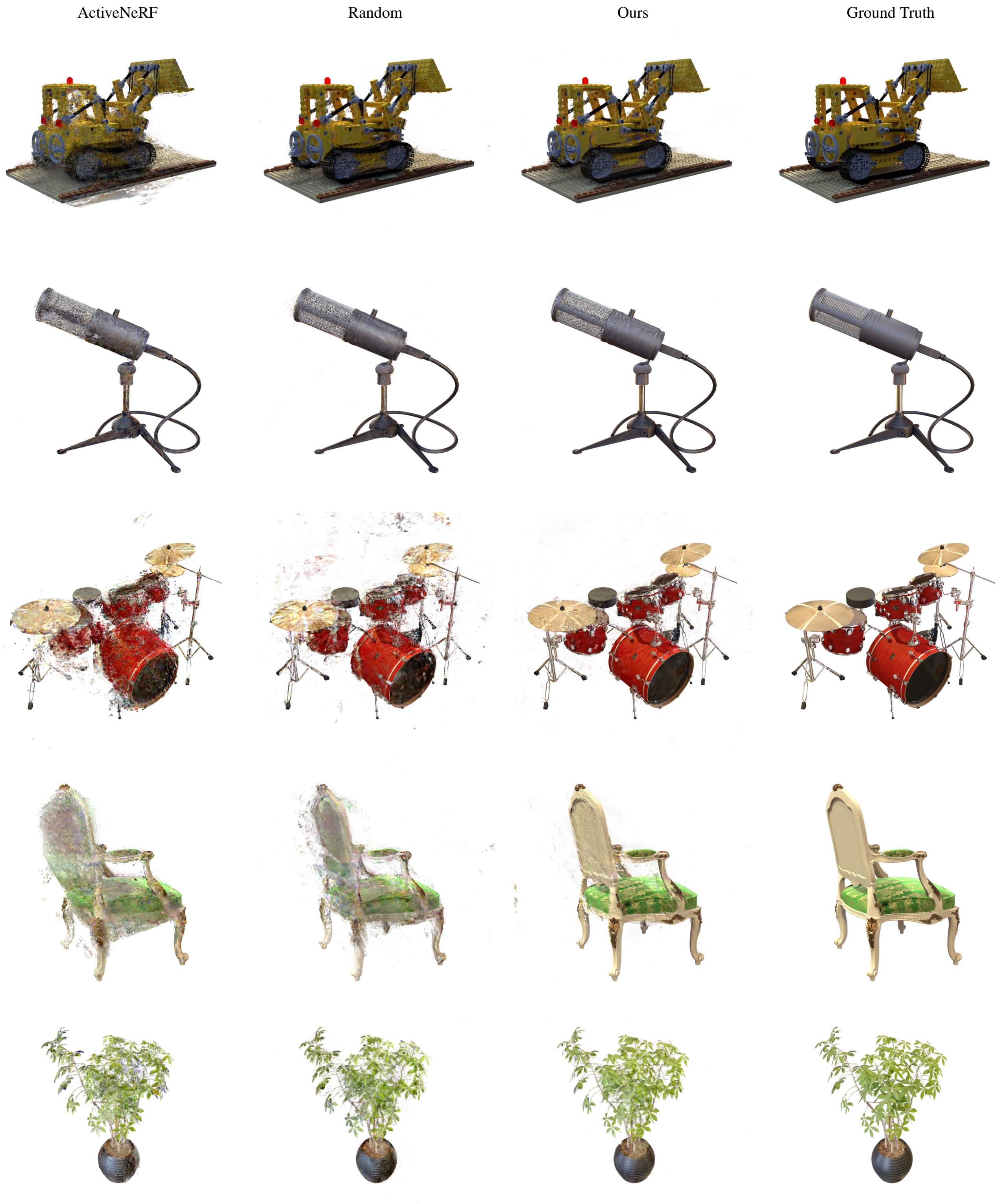

Figure 10. **Qualitative Comparisons on Blender Datasets with 20 Training Views and Plenoxels Backend.** We compare our method implemented with the Plenoxels backend with other methods using the Plenoxels backend as well. All the models are trained with the same setting except for the view selection algorithms. The models visualized in the figure are trained with 20 views in total, and four views are selected each time. Although methods with the Plenoxels backend generally have more artifacts and imperfections, our model still exhibits fewer artifacts compared to baseline models because the selected views by our algorithm could better regularize the model.

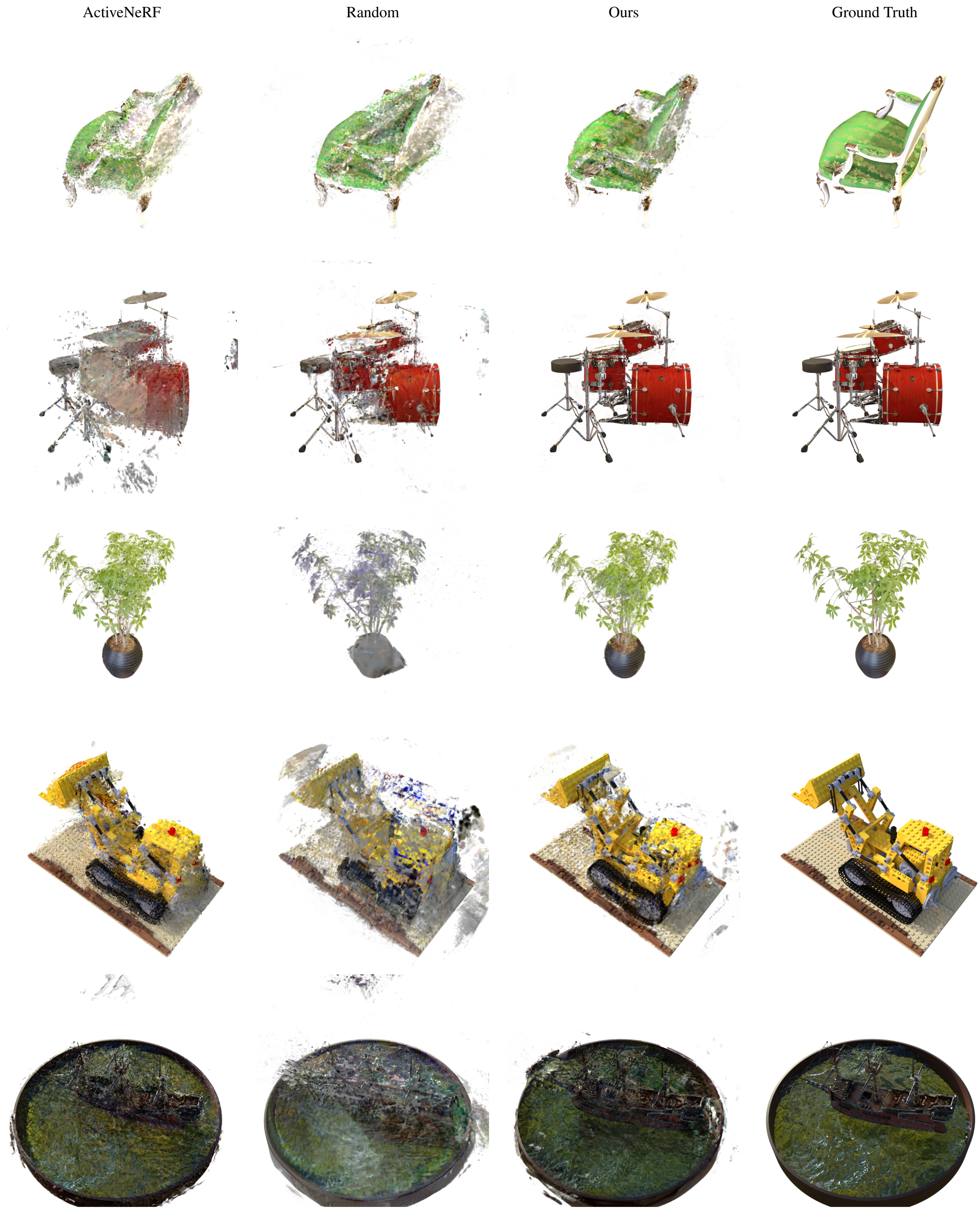

Figure 11. **Qualitative Results on Blender Datasets with 10 Training Views and Plenoxels Backend** We compare our method with other methods on the Plenoxels backend. The rendering results in the figure are generated by models trained with ten views in total. Although reconstructing from extremely limited viewpoints is much more challenging, our model still exhibits better qualitative rendering results compared to baseline models.

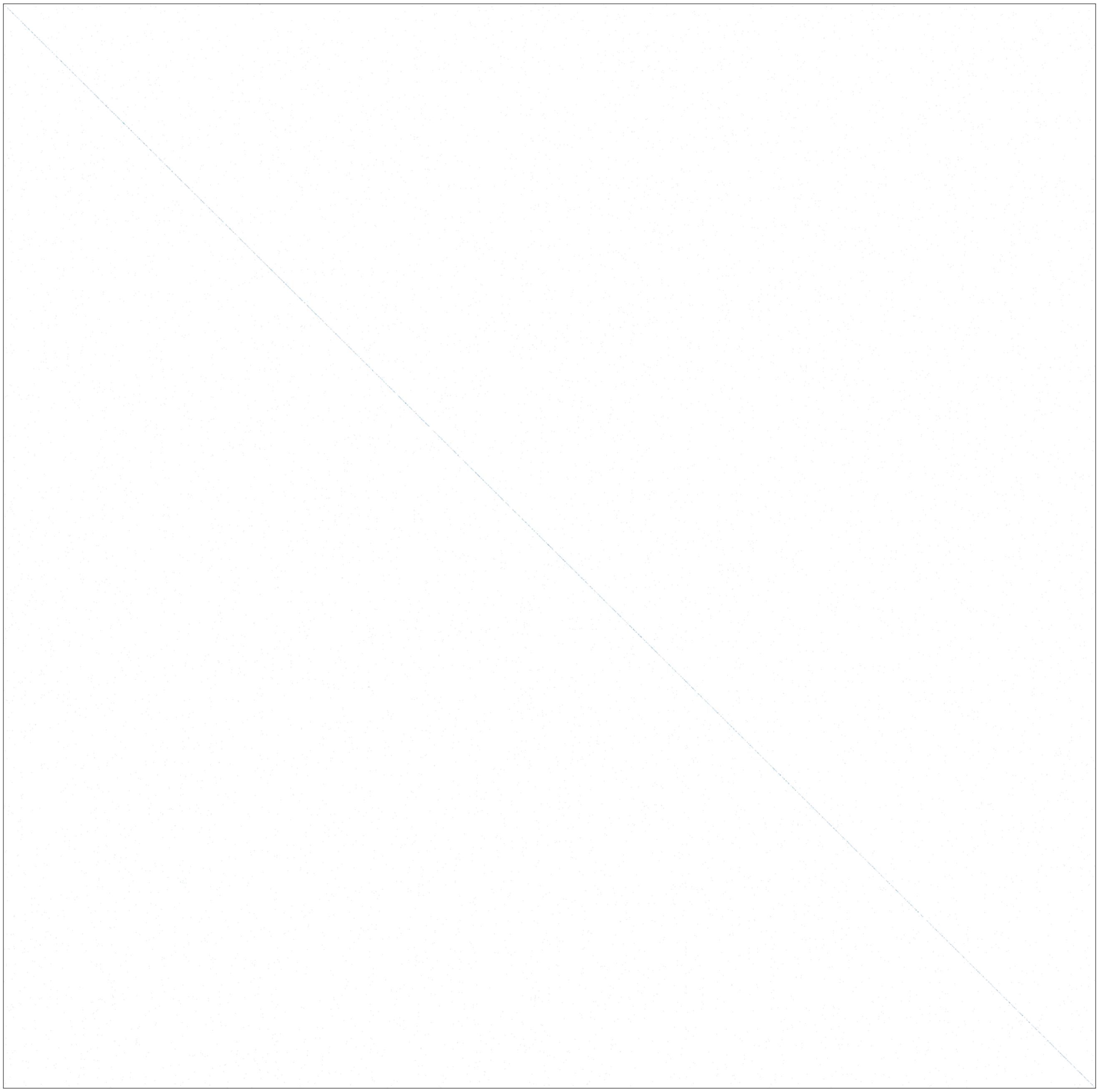

Figure 12. **An example of the Hessian matrix on the Parameters of Plenoxel.** We compute the Hessian of the NLL function of volumetric rendering following the Eq. (11). As it is impractical to compute the full Hessian matrix, we randomly subsample 10,000 parameters with non-zero Jacobians to visualize the Hessian matrix. We could observe the strong diagonal pattern of the Hessian matrix because, unlike densely connected neural networks, each parameter in Plenoxel is associated with a fixed grid vertex.

Figure 13. **Uncertainty qualitative visualization on LF dataset** Here, we show the qualitative comparisons between our method and CF-NeRF. Both methods are trained using four views in the LF dataset, following the configurations proposed by CF-NeRF [32]. We take the logarithm on the uncertainty map for better visualization.

References

[1] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*, 2020.

[2] Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham M. Kakade. Gone fishing: Neural active learning with fisher embeddings. In *NeurIPS*, 2021.

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023.

[5] Weirong Chen, Suryansh Kumar, and Fisher Yu. Uncertainty-driven dense two-view structure from motion. *IEEE Robotics and Automation Letters*, 8(3):1763–1770, 2023.

[6] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux–effortless Bayesian deep learning. In *NeurIPS*, 2021.

[7] Frank Dellaert and Lin Yen-Chen. Neural volume rendering: Nerf and beyond, 2021.

[8] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review, 2023.

[9] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' Rays: Uncertainty quantification in neural radiance fields. *arXiv*, 2023.

[10] Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, page 5–13, New York, NY, USA, 1993. Association for Computing Machinery.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. In *NeurIPS*, 1994.

[12] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745, 2011.

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023.

[14] Andreas Kirsch and Yarin Gal. Unifying approaches in active learning and active sampling via fisher information and information-theoretic quantities. *Transactions on Machine Learning Research*, 2022. Expert Certification.

[15] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *NeurIPS*, 2019.

[16] Andreas Kirsch, Sebastian Farquhar, Parmida Atighehchian, Andrew Jesson, Frederic Branchaud-Charron, and Yarin Gal. Stochastic batch acquisition for deep active learning. *arXiv preprint arXiv:2106.12059*, 2021.

[17] Suraj Nandkishor Kothawade, Nathan Alexander Beck, Krishnateja Killamsetty, and Rishabh K Iyer. SIMILAR: Submodular information measures based active learning in realistic scenarios. In *Advances in Neural Information Processing Systems*, 2021.

[18] D. V. Lindley. On a Measure of the Information Provided by an Experiment. *The Annals of Mathematical Statistics*, 27 (4):986 – 1005, 1956.

[19] Lachlan Ewen MacDonald, Jack Valmadre, and Simon Lucey. On progressive sharpening, flat minima and generalisation, 2023.

[20] David J. C. MacKay. Bayesian Interpolation. *Neural Computation*, 4(3):415–447, 1992.

[21] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[22] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[23] Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman, Ricardo Martin-Brualla, and Jonathan T. Barron. MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF, 2022.

[24] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022.

[25] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *ECCV*, pages 230–246. Springer, 2022.

[26] Yunlong Ran, Jing Zeng, Shibo He, Jiming Chen, Lincheng Li, Yingfeng Chen, Gimhee Lee, and Qi Ye. Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters*, 8(2):1125–1132, 2023.

[27] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021.

[28] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

[29] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

[30] M.J. Schervish. *Theory of Statistics*. Springer New York, 2012.

[31] Jianxiong Shen, Adria Ruiz, Antonio Agudo, and Francesc Moreno-Noguer. Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. *CoRR*, abs/2109.02123, 2021.

[32] Jianxiong Shen, Antonio Agudo, Francesc Moreno-Noguer, and Adria Ruiz. Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In *ECCV*, 2022.

[33] Sam Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. 2018.

[34] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022.

[35] Niko Sünderhauf, Jad Abou-Chakra, and Dimity Miller. Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In *ICRA*, 2023.

[36] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *NeurIPS*, pages 1398–1402. IEEE, 2003.

[37] Dongyu Yan, Jianheng Liu, Fengyu Quan, Haoyao Chen, and Mengmeng Fu. Active implicit object reconstruction using uncertainty-guided next-best-view optimization, 2023.

[38] Zike Yan, Haoxiang Yang, and Hongbin Zha. Active neural mapping. In *ICCV*, 2023.

[39] Kaan Yücer, Alexander Sorkine-Hornung, Oliver Wang, and Olga Sorkine-Hornung. Efficient 3d object segmentation from densely sampled light fields with applications to 3d reconstruction. *ACM Trans. Graph.*, 35(3), 2016.

[40] Huangying Zhan, Jiyang Zheng, Yi Xu, Ian Reid, and Hamid Rezatofighi. Activermap: Radiance field for active mapping and planning, 2022.

[41] Xueying Zhan, Qingzhong Wang, Kuan hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B. Chan. A comparative survey of deep active learning, 2022.

[42] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.