Appendix

This supplementary material presents proofs for our equations and algorithm in Sec. A. Then, we give more implementation details and results on view selection in Sec. B. Afterward, we show one ablation study on our active mapping system to show the effectiveness of each module in Sec. C. Finally, we introduce more implementation details and results on uncertainty quantifications in Sec. D and do runtime analysis in Sec. E.

A 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, 26, 29]. 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.* [28], who unified previous active learning approaches via Fisher Information.

Proof for Eq. 3 We compute the Expected Information Gain (EIG) of acquisition samples with:

$$
\begin{aligned}
&\mathcal{I}[\mathbf{w}^*; \{\mathbf{y}_i^{acq}\}|\{\mathbf{x}_i^{acq}\}, D^{train}] \\
&= H[\mathbf{w}^*|D^{train}] - H[\mathbf{w}^*|\{\mathbf{y}_i^{acq}\}, \{\mathbf{x}_i^{acq}\}, D^{train}] && (11) \\
&\approx -\frac{1}{2}\log\det \mathbf{H}''[\mathbf{w}^*|D_{train}] - \\
&\left(-\frac{1}{2}\log\det \mathbf{H}''[\mathbf{w}^*|\{\mathbf{y}_i^{acq}\}, \{\mathbf{x}_i^{acq}\}, D^{train}]\right) && (12) \\
&= -\frac{1}{2}\log\det \mathbf{H}''[\mathbf{w}^*|D_{train}] + \\
&\left(\frac{1}{2}\log\det\left(\mathbf{H}''\left[\{\mathbf{y}_i^{acq}\}|\{\mathbf{x}_i^{acq}\}, \mathbf{w}^*\right] + \mathbf{H}''[\mathbf{w}^*|D^{train}]\right)\right) && (13) \\
&= \frac{1}{2}\log\det\left(\mathbf{H}''[\{\mathbf{y}_i^{acq}\}|\{\mathbf{x}_i^{acq}\}, \mathbf{w}^*]\ \mathbf{H}''[\mathbf{w}^*|D^{train}]^{-1} + I\right) && (14) \\
&\leq \frac{1}{2}\operatorname{tr}\left(\mathbf{H}''[\{\mathbf{y}_i^{acq}\}|\{\mathbf{x}_i^{acq}\}, \mathbf{w}^*]\ \mathbf{H}''[\mathbf{w}^*|D^{train}]^{-1}\right) && (15)
\end{aligned}
$$

where we apply Bayes' theorem in Eq. 13. We can derive Eq. 15 because the Hessian matrix $\mathbf{H}''$ is symmetric, positive semidefinite. And for any symmetric, positive semidefinite matrices A with eigenvalues λ_i:

$$
\begin{aligned}
&\log\det(A+I) <= \log\prod_i(\lambda_i+1) && (16) \\
&= \sum_i \log(\lambda_i+1) \leq \sum_i \lambda_i = tr(A) && (17)
\end{aligned}
$$

the equality holds when $A = 0$.

Proof of Eq. 6 - 7 .

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

$$\mathbf{H}''[\mathbf{y}|\mathbf{x}, \mathbf{w}^*] = \mathrm{Cov}\left[\mathbf{H}'[\mathbf{y}|\mathbf{x}, \mathbf{w}^*]\right] \tag{18}$$

$$= \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)^T \mathrm{Cov}\big[\nabla_{\mathbf{z}} H[\mathbf{y}|\,\mathbf{z}]\big] \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*) \tag{19}$$

$$= \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)^T \underset{p(y|x,w^*)}{\mathbb{E}} \big[\nabla^2_{\mathbf{z}} H[\mathbf{y}|\,\mathbf{z}]\big] \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*) \tag{20}$$

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. 1, $p(\mathbf{y}|\,\mathbf{z} = f(\mathbf{x}; \mathbf{w}^*)) \sim \mathcal{N}(\mathbf{y}; \mathbf{z}, 1)$. Thus $\mathbb{E}_{p(y|x,w^*)}\big[\nabla^2_{\mathbf{z}} H[\mathbf{y}|\,\mathbf{z}]\big] = 1$ for any $\mathbf{y}$ and $\mathbf{z}$. Therefore:

$$\nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)^T \underset{p(y|x,w^*)}{\mathbb{E}} \big[\nabla^2_{\mathbf{z}} H[\mathbf{y}|\,\mathbf{z}]\big] \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)$$

$$= \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*)^T \nabla_{\mathbf{w}} f(\mathbf{x}; \mathbf{w}^*) \tag{21}$$

B More Implementation Details and Qualitative 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. We implement the computation of Fisher Information on Plenoxels [52] and 3D Gaussian Splatting [25] with customized CUDA kernels. As shown in Eq. 7, the diagonal Hessian matrixes can be implemented as efficiently as a back-propagation. Therefore, our customized CUDA kernel, which computes diagonal Hessians for 3D Gaussian Splatting, achieved more than 70 fps on an Nvidia RTX3090 GPU. 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 [25]. 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. 7 and Fig. 8. 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 [43]	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 8: Quantitative 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 [25, 38, 43, 52] 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 [43]. To validate our reimplementation, we compare our implementation with the original ActiveNeRF implementation on the MipNeRF-360 dataset in Table 8. 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 View Selection Experiments with BayesRays BayesRays [16] does not work on view selection and cannot be easily adapted to the recent 3D Gaussian Splatting model because 3D Gaussian Splatting does not take query points as input. Therefore, we conduct experiments with the source code from the authors that use NeRFactor [62] as the backbone and perform view selection based on the L2 norm of the rendered uncertainty map on each candidate's view. The initial training views and schedules for adding new training views are the same as those for other experiments.

Details about Our Implementation with Plenoxels Backend The initial views are uniformly sampled based on the translation vector of all camera poses. For the Blender 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

Table 9: Ablation Study on Gibson and MP3D We test and compare each module on the Gibson and MP3D dataset. It shows the effectiveness of our goal selection and path selection module in the mapping system.

Method	Gibson		MP3D	
	PSNR ↑	Dpeth MAE (m) ↓	PSNR ↑	Depth MAE (m) ↓
Frontier	19.35	0.1751	16.68	0.3627
Frontier + Goal Selection	22.13	0.1092	18.88	0.2178
Frontier + Goal Selection + Path Selection (**Ours**)	22.58	0.0924	19.96	0.1667

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. 9 and Fig. 10. 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. 11.

Details about Our Implementation with Active Mapping For the SLAM system, we use the SplaTAM [24] implementation. We add 1 in every 4 frames into the keyframe list and run the mapping optimization every 4 frames. We construct a 2D metric map from depth observations. Each cell in the 2D map has three states: unvisited, occupied, and free. All cells are initialized as unvisited. The cell size is 5 cm. In each observation, we unproject depth images into 3D point points and project them onto bird-eye view maps as obstacles with Orthographic Projection. We also uniformly sample 10 particles between 0 and 80 percent of valid depth value for each ray as free particles. These free particles are also projected onto the 2D map. The cells which these free particles fall into are marked as free cells. To avoid a potential collision, Gaussians within a height range between 0.12m and 1.3m are also projected onto the 2D map as obstacles. Frontiers are defined as the free cells which are adjacent to unvisited cells. For the goal selection, we randomly sample 256 view poses near the frontiers. We compute EIG for each goal and select the top 40 viewpoints. Afterward, we use A* to plan a path from the current robot position to each goal. A greedy path follower is adopted to plan actions to reach the goal. The maximum action planning horizon is set to 20. We then use Batch View Selection to calculate EIGs for each path and select the one with the highest EIG for execution.

C Ablation on Active Mapping

We conduct experiments to test the effectiveness of modules in our active mapping system. The results are summarized in Table 9. Here, **Frontier** is the naive frontier exploration algorithm. **Frontier + Goal Selection** calculates the EIG of sampled viewpoints near the frontier and chooses the largest one as the goal. **Frontier + Goal Selection + Path Selection** not only selects the goal with the highest EIG score but also selects the planned path with the highest information gain using batch view selection. From the table, we find with FisherRF goal point selection, the rendering quality increases from 19.35 to 22.13 in PSNR on the Gibson dataset and 16.68 to 18.88 on the MP3D dataset. Additionally, with path selection, the PSNR increases to 22.58 on the Gibson dataset and 19.96 on the MP3D dataset. Similarly, we also witness a decrease in the depth error on both Gibson and MP3D datasets when adding our modules. It shows the effectiveness of each module in our mapping system.

D More Details and Results on Uncertainty Estimation

In line with prior approaches in uncertainty estimation [55,56], we conducted evaluations on the Light Field (LF) Dataset [70] 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 12. 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.

E Runtime Analysis

First, we conduct an ablation study on the computation efficiency of our approximation. Since it is infeasible to compute the determinant and inverse of the entire Hessian matrix, we test the running time with different approximations on a subsampled matrix of 287962^2. Table 10 (a) summarizes the mean and standard deviation. The table shows trace computation is more than 1000

(a)	Log Det	Trace	Inverse
Time	1.2 s ± 0.1 s	0.8 ms ± 0.6 ms	6.0 s ± 0.1 s
(b)	ActiveNeRF	BayesRays	Ours
Time	6.9 s ± 6.4 ms	1.1 s ± 2.5 ms	11.3 ms ± 33.3 μs

Table 10: Ablation Study on the Computation Efficiency for View Selection and Uncertainty Quantification

times faster than determinant and inverse computation. Second, we also compare our view selection and uncertainty quantification speed with BayesRays and ActiveNeRF in Table 10 (b). Thanks to our customized CUDA implementation to calculate the trace of the Hessian matrix, our method only costs 11.3 ms for evaluating one viewpoint, which is much faster than ActiveNerF (6.9s) and BayesRays (1.1s).

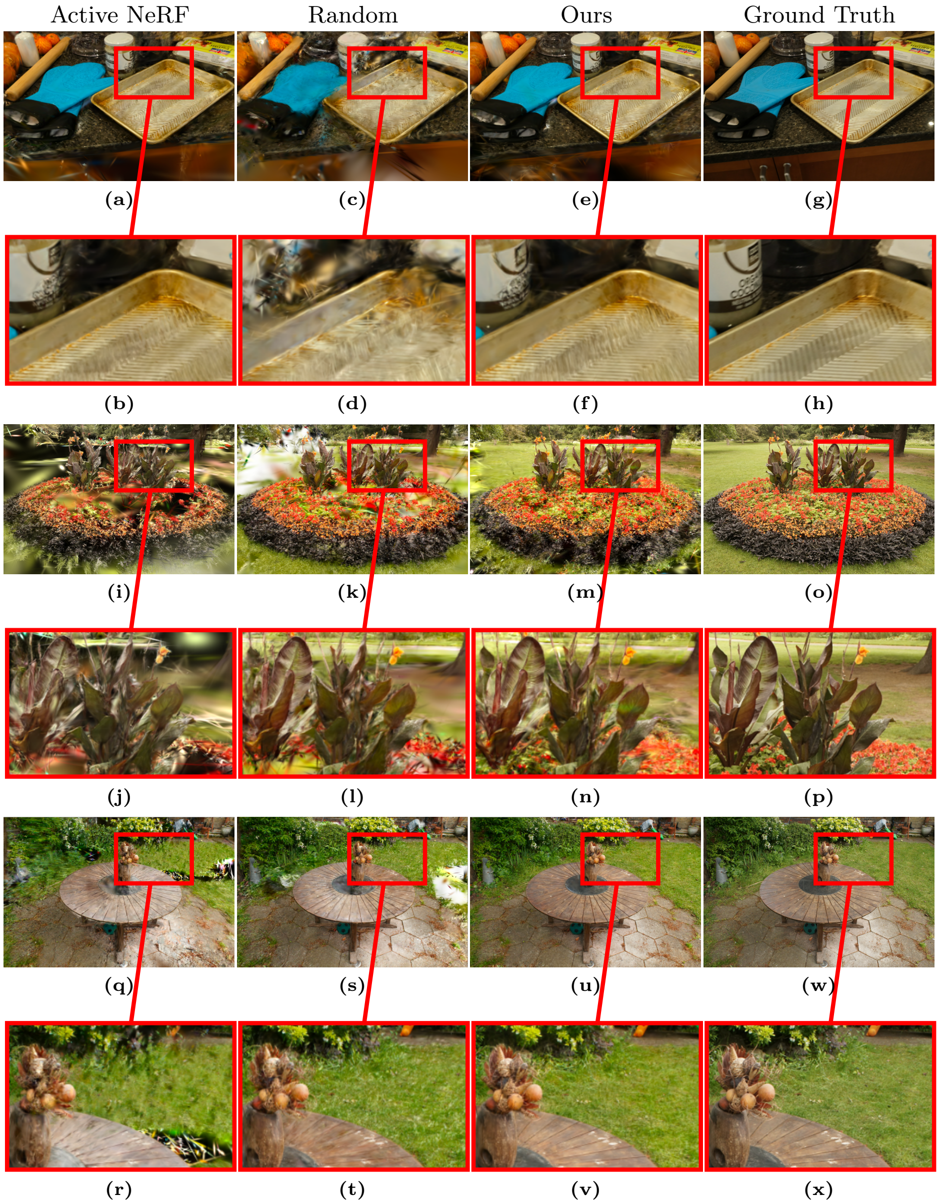

Fig. 7: Zoomed-in Qualitative Study of Our Method on MipNeRF-360 Dataset Every second rows are zoom-in figures. Visualizations are the results of the test set after being trained with 20 training views. All the methods have the same 3D Gaussian Splatting Backend except for different view selection algorithms. Our method could capture details that could only be covered by a limited number of cameras because of our active view selection algorithm.

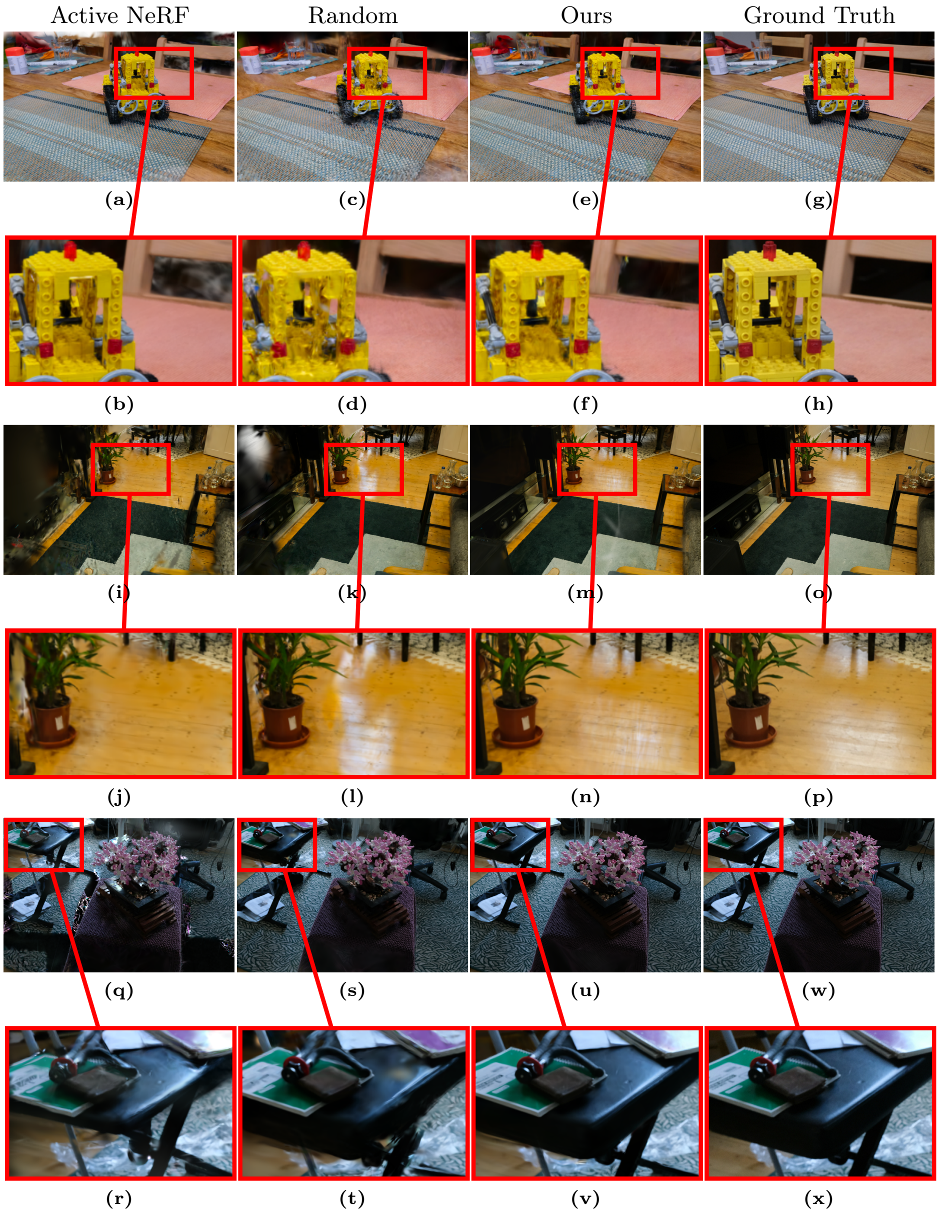

Fig. 8: Zoomed-in Qualitative Study of our method on Mip360 Dataset(cont.) Every second rows are zoom-in figures. Visualizations are the results of the test set after being trained with 20 training views. All the methods have the same 3D Gaussian Splatting Backend except for different view selection algorithms.

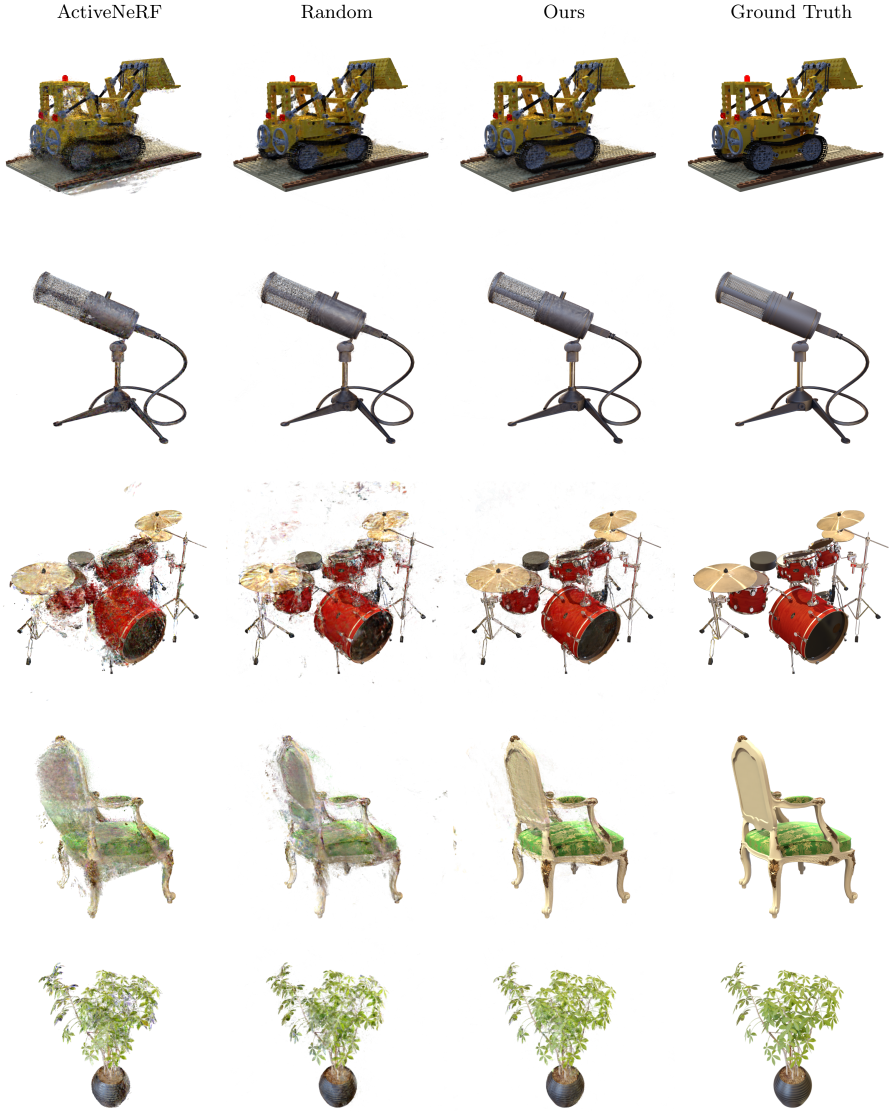

Fig. 9: Qualitative Comparisons on Blender Datasets with 20 Training Views and Plenoxels Backend. We compare our method implemented with the Plenoxels backend with other methods using the Plenoxels backend as well. All the models are trained with the same setting except for the view selection algorithms. 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 views chosen by our algorithm could better regularize the model.

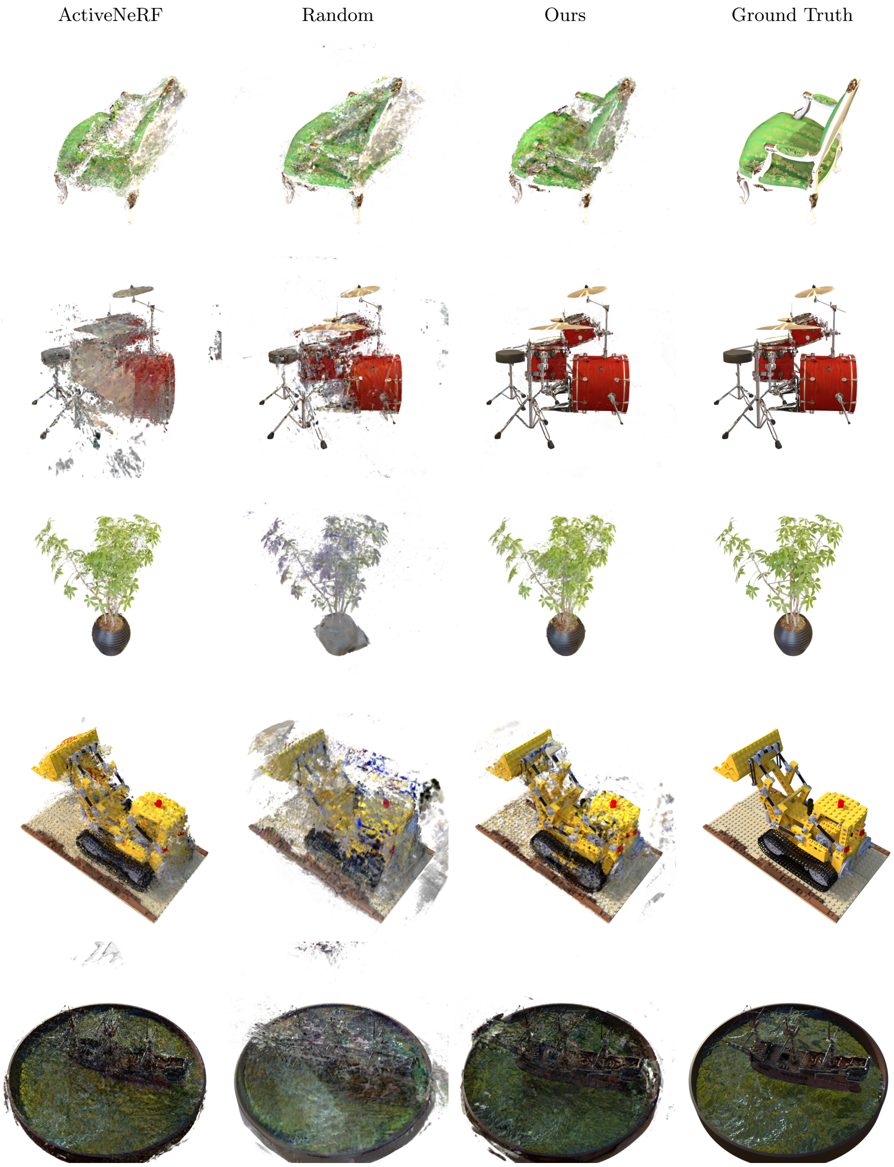

Fig. 10: Qualitative Results on Blender Datasets with 10 Training Views and Plenoxels Backend We compare our method with other methods on the Plenoxels backend. The rendering results in the figure are generated by models trained with ten views in total. Although reconstructing from extremely limited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

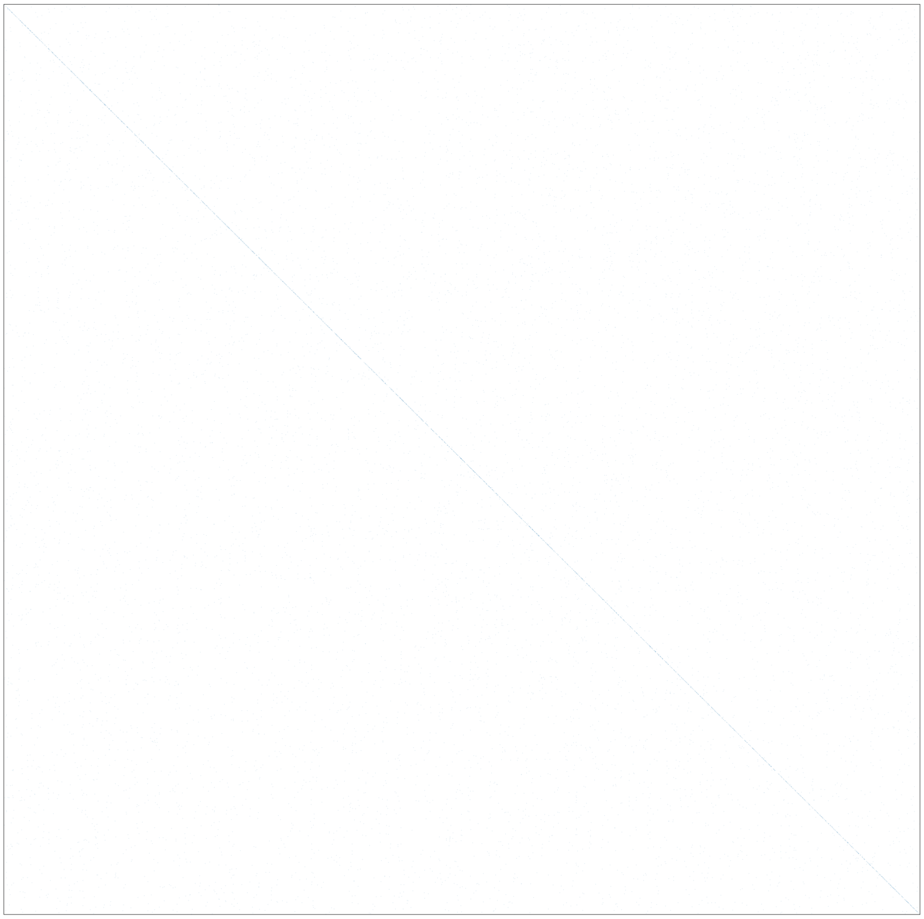

Fig. 11: An example of the Hessian matrix on the Parameters of Plenoxel. We compute the Hessian of the NLL function of volumetric rendering following the Eq. (7). As it is impractical to compute the full Hessian matrix, we randomly subsample 10,000 parameters with non-zero Jacobians to visualize the 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.

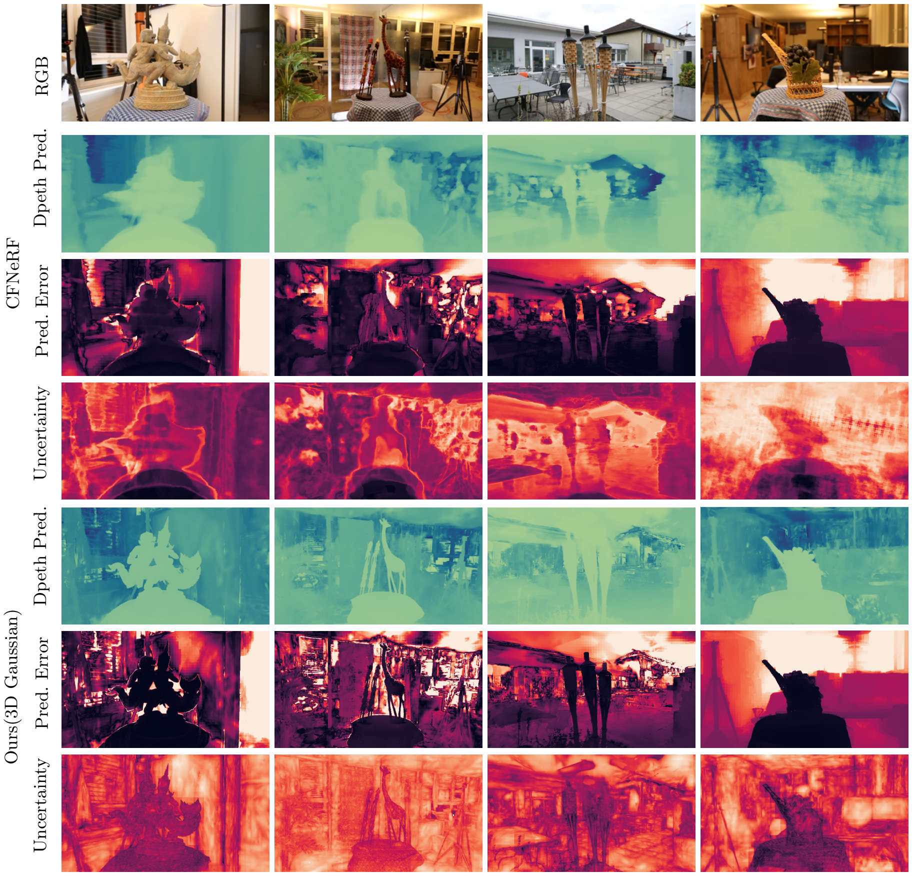

Fig. 12: Uncertainty Qualitative Visualization on LF Dataset Here, we show the qualitative comparisons between our method and CF-NeRF. Both methods are trained using four views in the LF dataset, following the configurations proposed by CF-NeRF [55]. We take the logarithm on the uncertainty map for better visualization.

References

1. Ash, J.T., Goel, S., Krishnamurthy, A., Kakade, S.M.: Gone fishing: Neural active learning with fisher embeddings. In: NeurIPS. vol. abs/2106.09675 (2021) 4, 15
2. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: ICLR (2020), https://openreview.net/forum?id=ryghZJBKPS 4, 15
3. Bajcsy, R.: Active perception. Proceedings of the IEEE **76**(8), 966–1005 (1988) 4
4. Bajcsy, R., Aloimonos, Y., Tsotsos, J.K.: Revisiting active perception. Autonomous Robots **42**, 177–196 (2018) 4
5. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022) 9
6. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: ICCV (2023) 9
7. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. International Conference on 3D Vision (2017) 12
8. Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural slam. arXiv preprint arXiv:2004.05155 (2020) 5
9. Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., Hennig, P.: Laplace redux–effortless Bayesian deep learning. In: NeurIPS (2021) 7
10. Dellaert, F., Yen-Chen, L.: Neural volume rendering: Nerf and beyond (2021) 4
11. Dhami, H., Sharma, V.D., Tokekar, P.: Pred-nbv: Prediction-guided next-best-view planning for 3d object reconstruction. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7149–7154. IEEE (2023) 5
12. Dornhege, C., Kleiner, A.: A frontier-void-based approach for autonomous exploration in 3d. Advanced Robotics **27**(6), 459–468 (2013) 5
13. Gao, K., Gao, Y., He, H., Lu, D., Xu, L., Li, J.: Nerf: Neural radiance field in 3d vision, a comprehensive review (2023) 4
14. Georgakis, G., Bucher, B., Arapin, A., Schmeckpeper, K., Matni, N., Daniilidis, K.: Uncertainty-driven planner for exploration and navigation. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 11295–11302. IEEE (2022) 5
15. Georgakis, G., Bucher, B., Arapin, A., Schmeckpeper, K., Matni, N., Daniilidis, K.: Uncertainty-driven planner for exploration and navigation. In: ICRA (2022) 12, 13
16. Goli, L., Reading, C., Sellán, S., Jacobson, A., Tagliasacchi, A.: Bayes' Rays: Uncertainty quantification in neural radiance fields. arXiv (2023) 2, 4, 10, 14, 17
17. Guédon, A., Monasse, P., Lepetit, V.: Scone: Surface coverage optimization in unknown environments by volumetric integration. Advances in Neural Information Processing Systems **35**, 20731–20743 (2022) 5
18. Guédon, A., Monnier, T., Monasse, P., Lepetit, V.: Macarons: Mapping and coverage anticipation with rgb online self-supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 940–951 (2023) 5
19. Hinton, G.E., van Camp, D.: Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the Sixth Annual Conference on Computational Learning Theory. p. 5–13. COLT '93, Association for Computing Machinery, New York, NY, USA (1993). https://doi.org/10.1145/168304.168306, https://doi.org/10.1145/168304.168306 3

20. Hochreiter, S., Schmidhuber, J.: Simplifying neural nets by discovering flat minima. In: NeurIPS (1994) 3
21. Houlsby, N., Huszar, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning. CoRR **abs/1112.5745** (2011), http://dblp.uni-trier.de/db/journals/corr/corr1112.html#abs-1112-5745 6
22. Jin, L., Chen, X., Rückin, J., Popović, M.: Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. arXiv preprint arXiv:2303.01284 (2023) 4
23. Karaman, S., Frazzoli, E.: Incremental sampling-based algorithms for optimal motion planning. Robotics Science and Systems VI **104**(2), 267–274 (2010) 5
24. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. arXiv preprint arXiv:2312.02126 (2023) 5, 8, 18
25. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/ 1, 3, 4, 8, 9, 10, 16, 17
26. Kirsch, A., Amersfoort, J.v., Gal, Y.: Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In: NeurIPS (2019) 6, 15
27. Kirsch, A., Farquhar, S., Atighehchian, P., Jesson, A., Branchaud-Charron, F., Gal, Y.: Stochastic batch acquisition for deep active learning. arXiv preprint arXiv:2106.12059 (2021) 4
28. Kirsch, A., Gal, Y.: Unifying approaches in active learning and active sampling via fisher information and information-theoretic quantities. Transactions on Machine Learning Research (2022), https://openreview.net/forum?id=UVDAKQANOW, expert Certification 3, 4, 6, 15
29. Kothawade, S.N., Beck, N.A., Killamsetty, K., Iyer, R.K.: SIMILAR: Submodular information measures based active learning in realistic scenarios. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=VGDFaLNFFk 4, 15
30. LaValle, S.: Rapidly-exploring random trees: A new tool for path planning. Research Report 9811 (1998) 5
31. Lee, S., Chen, L., Wang, J., Liniger, A., Kumar, S., Yu, F.: Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. IEEE Robotics and Automation Letters **7**(4), 12070–12077 (2022) 2, 4
32. Lindley, D.V.: On a Measure of the Information Provided by an Experiment. The Annals of Mathematical Statistics **27**(4), 986 – 1005 (1956). https://doi.org/10.1214/aoms/1177728069, https://doi.org/10.1214/aoms/1177728069 6
33. Lluvia, I., Lazkano, E., Ansuategi, A.: Active mapping and robot exploration: A survey. Sensors **21**(7), 2445 (2021) 5
34. MacDonald, L.E., Valmadre, J., Lucey, S.: On progressive sharpening, flat minima and generalisation (2023) 3
35. MacKay, D.J.C.: Bayesian Interpolation. Neural Computation **4**(3), 415–447 (05 1992). https://doi.org/10.1162/neco.1992.4.3.415, https://doi.org/10.1162/neco.1992.4.3.415 7
36. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: CVPR (2021) 2
37. Matsuki, H., Murai, R., Kelly, P.H.J., Davison, A.J.: Gaussian Splatting SLAM (2024) 5

38. 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) 9, 17
39. Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P., Martin-Brualla, R., Barron, J.T.: MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF (2022), https://github.com/google-research/multinerf 9
40. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, https://doi.org/10.1145/3528223.3530127 7
41. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. Mathematical programming **14**, 265–294 (1978) 7
42. Ortiz, J., Clegg, A., Dong, J., Sucar, E., Novotny, D., Zollhoefer, M., Mukadam, M.: isdf: Real-time neural signed distance fields for robot perception. In: Robotics: Science and Systems (2022) 5
43. Pan, X., Lai, Z., Song, S., Huang, G.: Activenerf: Learning where to see with uncertainty estimation. In: ECCV. pp. 230–246. Springer (2022) 2, 4, 5, 9, 11, 17
44. Papachristos, C., Khattak, S., Alexis, K.: Uncertainty-aware receding horizon exploration and mapping using aerial robots. In: 2017 IEEE international conference on robotics and automation (ICRA). pp. 4568–4575. IEEE (2017) 5
45. Placed, J.A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L., Castellanos, J.A.: A survey on active simultaneous localization and mapping: State of the art and new frontiers. IEEE Transactions on Robotics (2023) 5
46. Ramakrishnan, S.K., Al-Halah, Z., Grauman, K.: Occupancy anticipation for efficient exploration and navigation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. pp. 400–418. Springer (2020) 5
47. Ramakrishnan, S.K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J.M., Undersander, E., Galuba, W., Westbury, A., Chang, A.X., Savva, M., Zhao, Y., Batra, D.: Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2021) 12
48. Ran, Y., Zeng, J., He, S., Chen, J., Li, L., Chen, Y., Lee, G., Ye, Q.: Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. IEEE Robotics and Automation Letters **8**(2), 1125–1132 (Feb 2023). https://doi.org/10.1109/lra.2023.3235686, http://dx.doi.org/10.1109/LRA.2023.3235686 2, 5
49. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In: ICCV (2021) 7
50. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Gupta, B.B., Chen, X., Wang, X.: A survey of deep active learning. ACM computing surveys (CSUR) **54**(9), 1–40 (2021) 4
51. Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: Point-slam: Dense neural point cloud-based slam. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18433–18444 (2023) 5
52. Sara Fridovich-Keil and Alex Yu, Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022) 3, 7, 8, 16, 17

53. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A Platform for Embodied AI Research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 12
54. Schervish, M.: Theory of Statistics. Springer Series in Statistics, Springer New York (2012), https://books.google.com/books?id=s5LHBgAAQBAJ 5
55. Shen, J., Agudo, A., Moreno-Noguer, F., Ruiz, A.: Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In: ECCV (2022) 2, 4, 14, 19, 26
56. Shen, J., Ruiz, A., Agudo, A., Moreno-Noguer, F.: Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. CoRR **abs/2109.02123** (2021), https://arxiv.org/abs/2109.02123 4, 14, 19
57. Shen, S., Michael, N., Kumar, V.: Autonomous indoor 3d exploration with a micro-aerial vehicle. In: 2012 IEEE international conference on robotics and automation. pp. 9–15. IEEE (2012) 5
58. Smith, S., Le, Q.V.: A bayesian perspective on generalization and stochastic gradient descent. In: ICLR (2018), https://openreview.net/pdf?id=BJij4yg0Z 3
59. Sucar, E., Liu, S., Ortiz, J., Davison, A.: iMAP: Implicit mapping and positioning in real-time. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021) 5
60. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: CVPR (2022) 7
61. Sünderhauf, N., Abou-Chakra, J., Miller, D.: Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In: ICRA (2023) 4
62. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023) 17
63. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: NeurIPS. vol. 2, pp. 1398–1402. IEEE (2003) 9
64. Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S.: Gibson env: Real-world perception for embodied agents. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9068–9079 (2018) 12
65. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'. pp. 146–151. IEEE (1997) 4, 8
66. Yan, C., Qu, D., Wang, D., Xu, D., Wang, Z., Zhao, B., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. arXiv preprint arXiv:2311.11700 (2023) 5
67. Yan, D., Liu, J., Quan, F., Chen, H., Fu, M.: Active implicit object reconstruction using uncertainty-guided next-best-view optimization (2023) 4
68. Yan, Z., Yang, H., Zha, H.: Active neural mapping. In: ICCV (2023) 2, 5, 12, 13
69. Ye, K., Dong, S., Fan, Q., Wang, H., Yi, L., Xia, F., Wang, J., Chen, B.: Multi-robot active mapping via neural bipartite graph matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14839–14848 (2022) 5
70. Yücer, K., Sorkine-Hornung, A., Wang, O., Sorkine-Hornung, O.: Efficient 3d object segmentation from densely sampled light fields with applications to 3d reconstruction. ACM Trans. Graph. **35**(3) (mar 2016). https://doi.org/10.1145/2876504, https://doi.org/10.1145/2876504 14, 19

71. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting (2023) 5
72. Zhan, H., Zheng, J., Xu, Y., Reid, I., Rezatofighi, H.: Activermap: Radiance field for active mapping and planning (2022) 2, 4
73. Zhan, X., Wang, Q., hao Huang, K., Xiong, H., Dou, D., Chan, A.B.: A comparative survey of deep active learning (2022) 4
74. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018) 9
75. Zhu, C., Ding, R., Lin, M., Wu, Y.: A 3d frontier-based exploration tool for mavs. In: 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI). pp. 348–352. IEEE (2015) 5
76. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam (2021) 5