PRIMU: Uncertainty Estimation for Novel Views in Gaussian Splatting from Primitive-Based Representations of Error and Coverage

Supplementary Material

This appendix provides additional analyses, implementation details, and qualitative results that complement the main paper. Sec. A supplements the experimental setup with implementation details for the uncertainty feature maps and regression models, and also describes the regression-based FisherRF method and the AVS protocol used. Sec. B presents results of our scene-separation study, focusing specifically on uncertainty estimation for background regions. Sec. C evaluates the out-of-scene generalization capabilities of the meta-regression models used in PRIMU. Sec. D provides qualitative examples and visual comparisons of predicted uncertainty maps. In Sec. E, we explore the benefits of incorporating neighboring-pixel information to improve UE. Finally, Sec. F presents an experiment investigating depth UE for different camera baselines and object textures.

A. Supplement to Experimental Setup

This section provides additional implementation details on constructing the uncertainty feature maps and training the meta-regression models. Additionally, we provide an overview of the regression-based variant of FisherRF and describe the protocol used in our AVS experiments.

Implementation Details To compute the training-view coverage and rendering-error representations, we implement the different aggregations for individual training views from Eqs. (4) and (5) based on the standard GS rasterizer [14], both with and without the $\alpha_k(\bar{x})$ term. This yields, for each of the six coverage and six rendering-error representations, a scalar value per Gaussian primitive and training view. To obtain direction-independent Gaussian primitive representations, we aggregate these scalars over all training views according to Eqs. (4) and (5).

For the direction-dependent representations, we construct a von Mises-Fisher distribution for each Gaussian primitive training view pair, rescaled by the corresponding scalar value. Rather than storing all such distributions, we directly aggregate them according to Eqs. (7) and (8). The aggregation is performed on discrete sample points on the unit sphere based on McEwen & Wiaux symmetric sampling [29], enabling direct conversion to spherical harmonics via the forward transform [32]. We use a spherical harmonics degree of four for the direction-dependent representations.

For the coverage and rendering-error representations, we define the set $v(k)$ as all pixels in a view where Gaussian primitive k is visible (used in Eqs. (4), (5), (7) and (8)). In line with the standard GS implementation, a Gaussian primitive k is considered visible in a pixel $\bar{x}$ if $\alpha_k(\bar{x}) \geq 1/255$ and $T_k(\bar{x}) \geq 10^{-3}$.

For the lightweight meta-regression models, we use the scikit-learn [31] implementations of gradient boosting (`HistGradientBoostingRegressor`) and linear regression (`LinearRegression`). The parameters used for gradient boosting are:

- `max_iter=150`
- `learning_rate=0.3`,
- `validation_fraction=0.2`
- `random_state=42`
- `categorical_features=None`
- `scoring="neg_root_mean_squared_error`
- `early_stopping=False`.

Regression-Based FisherRF This paragraph outlines the regression-based FisherRF method used in the UE scene-separation study. Original FisherRF [11] is based on the Fisher information matrix of a GS model with respect to the gradient of the rendering function. The diagonal entries correspond to variances of Gaussian primitive parameters and are used to estimate their uncertainty. FisherRF aggregates these uncertainties using the sum of color parameter variances (including all spherical harmonics coefficients used for direction-dependent color). Pixel-wise uncertainties are then obtained via GS rendering (Eq. (2)).

We modify FisherRF to make it more comparable to our method by introducing a meta-regression step. Specifically, we compute six uncertainty feature maps based on Gaussian parameter variances for different parameter groups: (1) mean, (2) scale, (3) rotation, (4) opacity factor, (5) spherical harmonics coefficients of degree zero, and (6) higher-degree spherical harmonics coefficients. This grouping follows the original GS implementation [14], where scale and rotation together define the covariance matrix. In this way, all Gaussian primitive parameters are represented in one of the six feature maps. These feature maps are then used as input to regression models in the same manner as in our method, enabling a direct comparison of feature quality for UE.

AVS Protocol As described in the main manuscript, we largely follow the AVS protocol proposed by Jiang et al. [11]. The only deviation is that we do not reduce the image resolution in our experiments. At the start, every eighth view is removed from the candidate set and reserved

views	predicted error type		rendering error								depth error							
	evaluation on		entire views				background				entire views				background			
	dataset		LF		TUM		LF		TUM		LF		TUM		LF		TUM	
	metric		AUSE	Pears.	AUSE	Pears.	AUSE	Pears.	AUSE	Pears.	AUSE	Pears.	AUSE	Pears.	AUSE	Pears.	AUSE	Pears.
	method	model	(↓)	(↑)	(↓)	(↑)	(↓)	(↑)	(↓)	(↑)	(↓)	(↑)	(↓)	(↑)	(↓)	(↑)	(↓)	(↑)
	FisherRF		0.878	-0.224	0.957	-0.241	0.962	-0.265	0.976	-0.255	0.322	0.31	0.742	-0.08	0.361	0.231	0.758	-0.169
1 view	reg. FisherRF	grad.	0.282	0.318	0.435	0.181	0.277	0.33	0.44	0.183	0.133	0.683	0.328	0.375	0.163	0.627	0.328	0.357
		lin.	0.332	0.182	0.404	0.143	0.311	0.198	0.41	0.143	0.212	0.452	0.363	0.305	0.265	0.363	0.357	0.283
	PRIMU (ours)	grad.	0.244	0.412	0.416	0.217	0.251	0.41	0.435	0.199	0.112	0.734	0.213	0.536	0.136	0.697	0.235	0.497
		lin.	0.286	0.309	0.353	0.29	0.287	0.335	0.369	0.279	0.202	0.545	0.188	0.571	0.254	0.468	0.213	0.529
	PRIMU*(ours)	grad.	0.238	0.419	0.415	0.23	0.245	0.42	0.426	0.219	0.108	0.739	0.239	0.501	0.134	0.699	0.272	0.443
		lin.	0.276	0.35	0.362	0.296	0.273	0.377	0.373	0.282	0.202	0.557	0.226	0.521	0.253	0.477	0.26	0.456
m-view	reg. FisherRF	grad.	0.26	0.352	0.36	0.267	0.255	0.365	0.369	0.275	0.127	0.715	0.226	0.515	0.155	0.667	0.226	0.501
	PRIMU (ours)	grad.	0.222	0.446	0.335	0.302	0.226	0.45	0.353	0.293	0.099	0.777	**0.132**	**0.653**	0.119	0.745	**0.155**	**0.611**
	PRIMU*(ours)	grad.	**0.218**	**0.458**	**0.329**	**0.321**	**0.223**	**0.461**	**0.338**	**0.312**	**0.097**	**0.788**	0.161	0.605	**0.117**	**0.753**	0.183	0.57

Table 7. Scene separation UE results for scene background. "entire views" evaluates on the full GS scene; "background" limits to the scene background. PRIMU* uses direction-dependent coverage and error maps.

for evaluation. After selecting an initial training view, three additional views are chosen by maximizing the Euclidean distance between camera centers and the already selected views. All methods start training with these four views.

Additional views are selected during training according to the AVS scheme, with each selection occurring after a number of iterations proportional to the number of already selected views. This process continues until a total of 20 training views is reached. Specifically, a new view is added after $100 \cdot |\text{selected views}|$ iterations (e.g., the fifth view after 400 iterations, the sixth after an additional 500, and so on). To account for the limited number of training views in early iterations, the spherical harmonics degree is increased every 5000 iterations instead of every 1000, as in the original GS setup. Apart from this adjustment, standard GS training parameters are used [14].

B. UE Scene Separation Study for Background

Here, we are extending our UE scene separation study. We are also analyzing UE on the scene background alone. Therefore, the meta-regression models for our method and the regression-based FisherRF are trained on the scene background. In Tab. 7 we provide the numerical results. For easier comparison, we also include the results for the entire scene again. We observe that the effects on the scene background are analogous to those on the entire scene. For instance, the best-performing configurations are identical. This is to be expected, as the background usually dominates in terms of pixel counts in an image. Typically, this is true when capturing an object in an environment, as in the scene datasets used here. In general, the UE performance on the scene background is slightly worse than in the entire-scene setting. Nevertheless, the overall performance of our method is good for the background, especially for depth UE.

C. UE Out-of-Scene Generalization

We study how well the PRIMU UE meta-regression models generalize to new scenes. If they do, it would suggest that the uncertainty information in our feature maps is at least partially scene-independent. For this study, we use the setup from our UE scene separation study to also investigate generalization in object-centric and background settings. For comparison, we also conduct generalization experiments for the regression-based FisherRF approach. We train the meta-regression models using one view of one scene, multiple views of one scene, or multiple views of multiple scenes from the same dataset. The results are provided in Tab. 8, which also includes a row of standard FisherRF results for comparison. To improve readability, we only report the Pearson correlation of the uncertainty maps to the true error. The rows of the table correspond to the mean correlations of different meta-regression models trained using different sets of views from the training datasets (given by column "training"). The columns correspond to the means of evaluation over scenes in the evaluation datasets (given by row "dataset"). These means never include regression models that are trained and evaluated on the same scene.

Examining the results, we see that our method consistently outperforms the standard FisherRF method, even in settings where we train regression models using holdout views from different scenes. The only noticeable exception is depth UE on the LF dataset in the entire-scene and background settings. In these settings, both our method and the regression-based FisherRF method perform significantly worse than when training on holdout views of the same scene. This is likely due to an inconsistent depth scale across the LF scenes. When we compare to the TUM dataset in the depth UE, entire-scene, or background settings, we see a less pronounced drop in performance, and the TUM dataset has metric depth scales for all scenes. In-

views	method	model	predicted error type / evaluation on dataset / metric / training	rendering error: entire views LF Pears. (↑)	rendering error: entire views TUM Pears. (↑)	rendering error: object-centric LF Pears. (↑)	rendering error: object-centric TUM Pears. (↑)	rendering error: background LF Pears. (↑)	rendering error: background TUM Pears. (↑)	depth error: entire views LF Pears. (↑)	depth error: entire views TUM Pears. (↑)	depth error: object-centric LF Pears. (↑)	depth error: object-centric TUM Pears. (↑)	depth error: background LF Pears. (↑)	depth error: background TUM Pears. (↑)
	FisherRF			-0.224	-0.241	0.129	0.458	-0.265	-0.255	0.31	-0.08	0.05	0.303	**0.231**	-0.169
1 view	reg. FisherRF	grad.	LF	0.153	0.163	0.328	0.42	0.168	0.157	0.24	-0.008	0.284	0.312	0.176	-0.095
			TUM	0.104	0.13	0.273	0.419	0.126	0.137	0.139	0.273	0.245	0.278	0.039	0.245
		lin.	LF	0.115	0.078	0.184	0.386	0.117	0.065	0.19	0.123	0.259	0.253	0.107	0.086
			TUM	0.076	0.099	0.087	0.329	0.081	0.097	0.15	0.203	0.089	0.184	0.066	0.175
	PRIMU (ours)	grad.	LF	0.192	0.168	0.509	0.481	0.199	0.124	0.121	0.153	0.632	0.453	0.017	0.059
			TUM	0.122	0.177	0.398	0.484	0.141	0.158	0.187	0.404	0.514	0.506	0.049	0.335
		lin.	LF	0.215	0.197	0.206	0.013	0.225	0.198	0.25	0.097	0.444	0.128	0.105	0.063
			TUM	0.186	0.239	0.165	0.296	0.204	0.227	**0.312**	0.468	0.265	0.257	0.076	0.392
	PRIMU*(ours)	grad.	LF	0.2	0.167	0.548	0.506	0.196	0.095	0.159	0.104	0.631	0.452	0.031	0.016
			TUM	0.155	0.203	0.424	0.5	0.166	0.18	0.183	0.378	0.513	0.5	0.048	0.3
		lin.	LF	**0.288**	0.202	0.179	0.1	**0.297**	0.18	0.242	0.076	0.384	0.17	0.126	0.059
			TUM	0.248	0.259	0.067	0.262	0.247	0.247	0.281	0.413	0.123	0.166	0.078	0.331
m-view	reg. FisherRF	grad.	LF	0.15	0.16	0.343	0.448	0.17	0.164	0.243	-0.01	0.284	0.315	0.159	-0.106
			TUM	0.145	0.186	0.308	0.445	0.175	0.198	0.193	0.349	0.31	0.304	0.072	0.305
	PRIMU (ours)	grad.	LF	0.194	0.195	0.491	0.513	0.2	0.128	0.13	0.148	0.657	0.464	0.017	0.084
			TUM	0.13	0.226	0.444	0.5	0.155	0.208	0.203	0.477	0.584	0.577	0.057	0.412
	PRIMU*(ours)	grad.	LF	0.206	0.176	0.551	0.497	0.203	0.074	0.153	0.12	0.644	0.448	0.038	-0.014
			TUM	0.199	0.257	0.497	0.524	0.206	0.229	0.247	0.45	0.6	0.585	0.079	0.377
m-scene	reg. FisherRF	grad.	LF	0.248	0.217	0.532	0.503	0.248	0.201	0.083	-0.046	0.261	0.356	0.062	-0.093
			TUM	0.213	0.261	0.434	0.502	0.244	0.268	0.201	0.456	0.419	0.418	0.058	0.423
	PRIMU (ours)	grad.	LF	0.264	0.202	0.575	0.454	0.268	0.135	0.039	0.095	0.687	0.455	0.009	0.036
			TUM	0.165	0.305	0.544	0.559	0.184	0.291	0.217	**0.561**	0.642	0.628	0.1	**0.488**
	PRIMU*(ours)	grad.	LF	0.266	0.112	**0.611**	0.517	0.266	0.045	-0.03	-0.004	0.663	0.463	-0.051	-0.002
			TUM	0.256	**0.333**	0.583	**0.572**	0.269	**0.315**	0.274	0.545	**0.688**	**0.645**	0.133	0.474

Table 8. Numerical results of UE out-of-scene generalization study. Pearson Correlation of predicted to true depth/rendering error for entire/object-centric/background scene evaluation. All rows except the first are the averages when training the regressor on the dataset in the column 'training'. PRIMU* uses direction-dependent coverage and error maps.

terestingly, we also see that, in some cases, linear regression trained on a single hold-out view performs best, indicating weaker nonlinear effects in these settings and suggesting a tendency of gradient boosting to overfit the training scenes. We also notice that, in a few cases, models trained on TUM scenes perform better on LF than models trained on LF. This may be due to the fact that TUM has more scenes and more diverse hold-out views within those scenes than LF, covering a wider range of scenarios. Overall, we observe that meta-regression models can generalize to a certain extent to unseen scenes, and models trained on multiple scenes demonstrate the best generalization capability.

D. Visual UE Examples

Here, we provide some additional qualitative examples and comparisons of uncertainty maps. In Fig. 5, we show more qualitative comparisons of the uncertainty maps of our method against the baselines for rendering error UE, and in Fig. 6, for depth UE. Figs. 7 and 8 show a visual comparison of the uncertainty maps of our method for different choices of regression model, number of training views, and direction-dependent or independent uncertainty feature maps. As we can see, the uncertainty maps produced by linear regression models are less nuanced than those produced by gradient boosting models. This is likely due to the limited capacity of linear regression models compared to gradient boosting models. Using direction-dependent uncertainty feature maps makes the predicted uncertainty maps more visually similar to the actual error maps. For the Teddy2 scene from the TUM dataset (see Fig. 8), we observe that training the regression model with multiple hold-out views also notably improves the visual quality of the uncertainty maps. However, this is not the case for the Africa scene from LF (see Fig. 7), where the uncertainty maps produced by training on one hold-out view are already very close to the actual error maps.

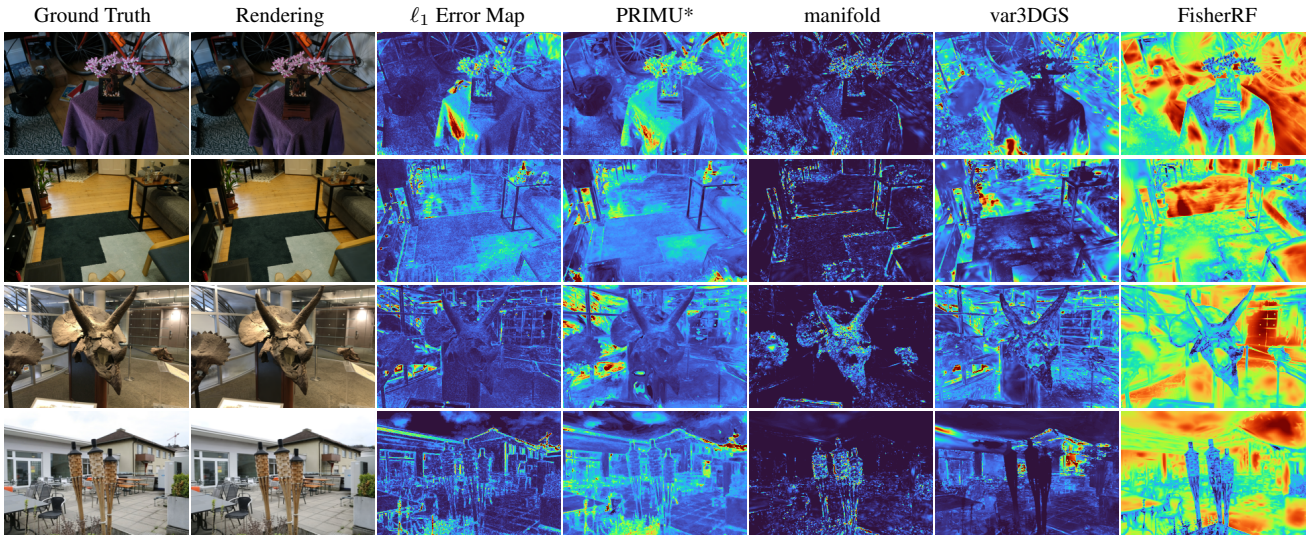

Figure 5. Qualitative comparison of uncertainty feature maps for rendering error UE of FisherRF [11], manifold [19], var3DGS [17] and direction-dependent PRIMU* (ours), to the ℓ_1 rendering error. Scenes from top to bottom: MipNeRF360 bonsai and room, LLFF horns and LF torch.

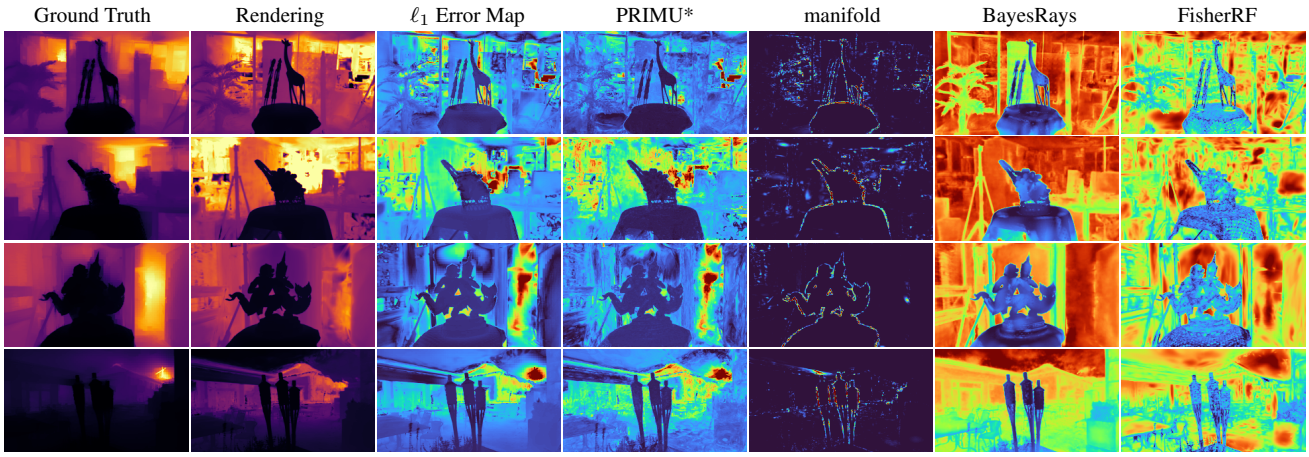

Figure 6. Qualitative comparison of uncertainty feature maps for depth UE of FisherRF [11], manifold [19], BayesRays [5] and direction-dependent PRIMU* (ours), to the ℓ_1 rendering error. Scenes from top to bottom: LF africa, basket, statue and torch.

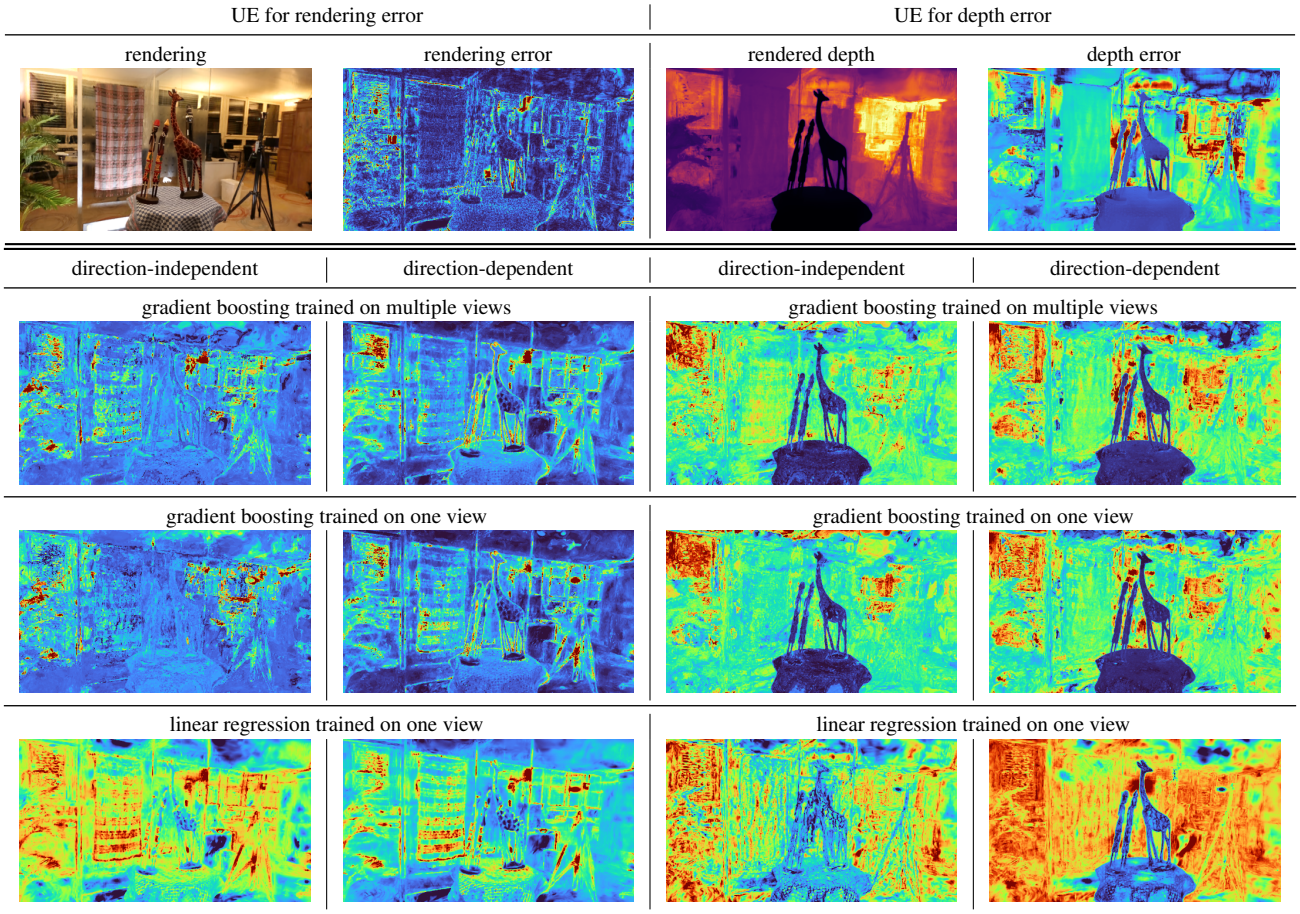

Figure 7. Qualitative example of the UE of our method for the scene africa of the LF dataset. Top row color and depth renderings of view and their error maps. Below predicted error from different regression models trained on direction-independent or direction-dependent (plus FoV counter) uncertainty feature maps.

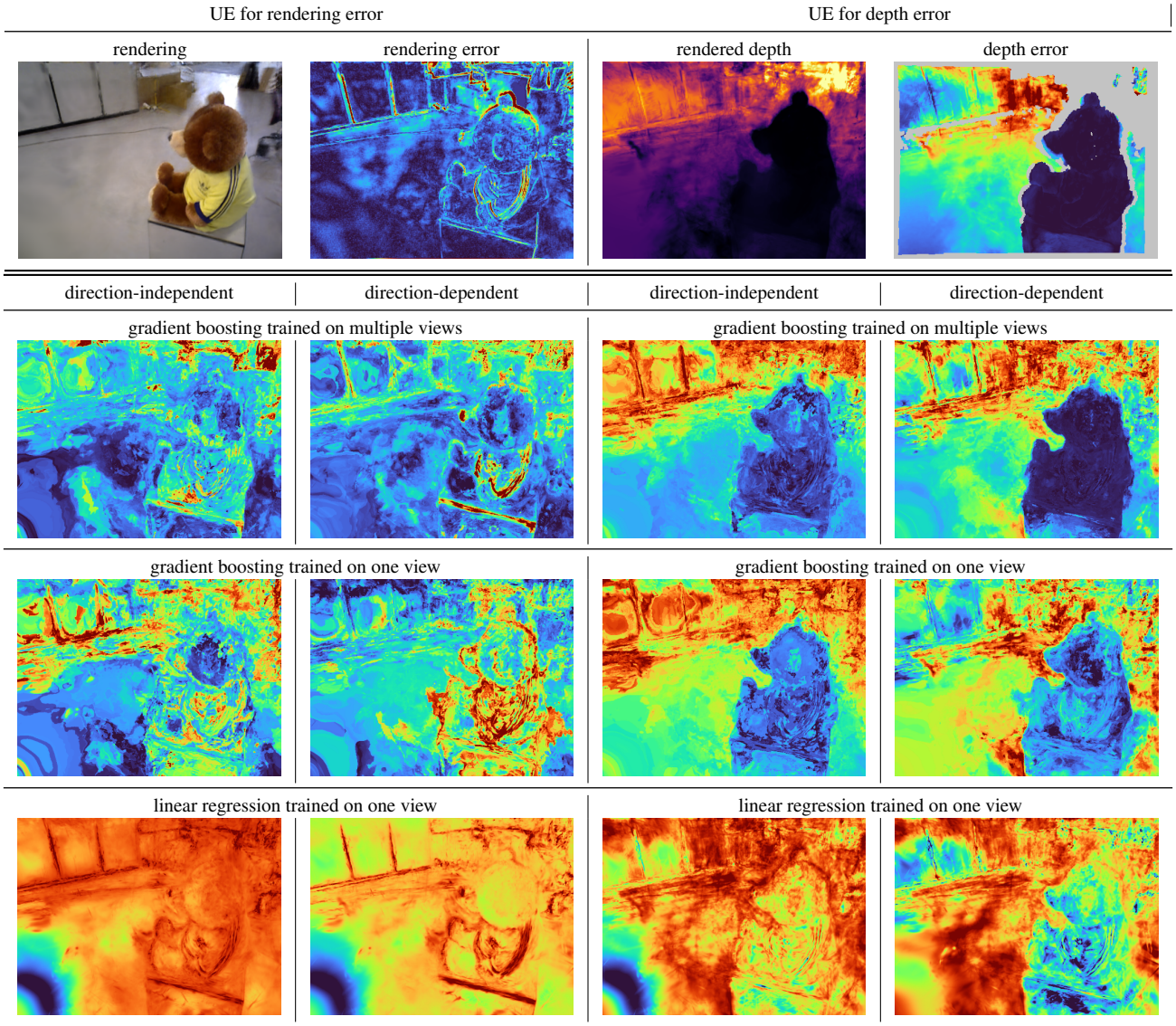

Figure 8. Qualitative example of the UE of our method for the scene teddy2 of the TUM dataset. Top row color and depth renderings of view and their error maps. Below predicted error from different regression models trained on direction-independent or direction-dependent (plus FoV counter) uncertainty feature maps.

E. The Role of Spatial Interaction

So far, we performed pixel-wise regression on uncertainty feature maps. Here, we study whether incorporating uncertainty features from neighboring pixels improves the estimation of rendering and depth uncertainty. We replace the pixel-wise gradient boosting regressor with lightweight fully convolutional neural networks (CNNs) and evaluate their UE on the LF dataset. We train 72 CNNs to predict rendering and depth errors on foreground objects, using all 13 direction-independent primitive representations from the first three hold-out views of a scene and testing on the fourth view. Each CNN has a depth of 4 layers, with kernel sizes in

$$\{(k_1, \ldots, k_4) | \, k_i \in \{1, 3, 5\}, \, k_j + 2 \geq k_i \geq k_j \;\text{ for }\; i < j\} \tag{10}$$

and channel depth in $\{(h, h, h, h) | h \in \{32, 64, 128, 256\}\}$. Notably, CNNs with all kernel sizes set to 1 reduce to pixel-wise fully connected baselines using only 1×1 convolutions. As batch size we found 1 most useful and trained all CNNs for 2000 iterations with a learning rate scheduled to decrease from 10^{-3} to 10^{-5}, along with a weight decay of 10^{-5}. UE performances are provided in terms of Pearson correlation between the predicted and true errors, displayed in Fig. 9.

Overall, the CNNs did not outperform gradient boosting, which, however, is not the aim of this study. Interestingly, CNNs with larger receptive fields consistently outperform the fully-connected baselines in rendering error estimation, demonstrating the usefulness of neighborhood information for this task. For depth error estimation, increased receptive field size improves performance only for CNNs with 32 channels, which may be attributed to increased model capacity instead of neighborhood information itself.

F. Synthetic Camera Baseline Experiment

We conduct an experiment on two simple synthetic scenes to investigate the effects of varying camera baselines and object texture on depth UE for our method. Both scenes consist of a quadratic room containing a cube at its center and a single point light source. They differ only in the texture of the cube: one scene contains an untextured cube, while the other features a cube with an irregular painting texture (see Fig. 10).

cube texture	Angle 22°		Angle 43°		Angle 65°		Angle 86°	
	AUSE	ℓ_1	AUSE	ℓ_1	AUSE	ℓ_1	AUSE	ℓ_1
no	0.163	**2.40**	0.157	**2.46**	0.069	2.89	0.167	**2.89**
yes	**0.088**	2.58	**0.081**	2.90	**0.056**	**2.79**	**0.098**	3.05

Table 9. Depth ℓ_1 error of the reconstruction and AUSE for the synthetic camera baseline experiment for depth UE.

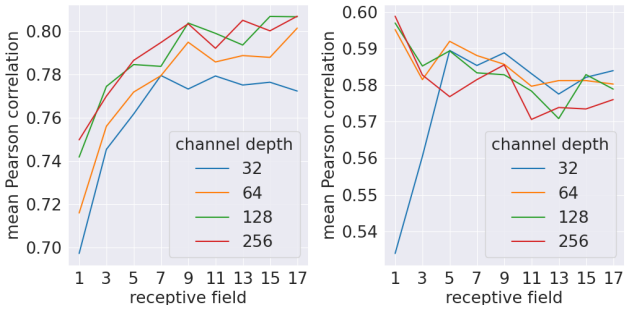

Figure 9. Average in-scene Pearson correlations of predicted and true rendering (left) and depth errors (right) for CNN models with varying receptive field sizes and numbers of uncertainty features.

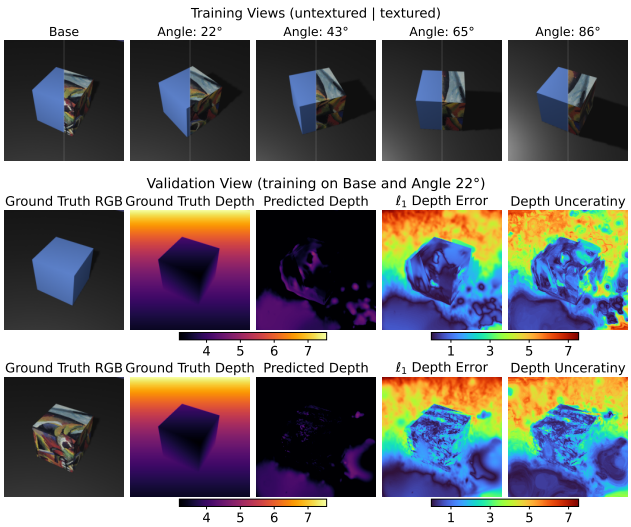

Figure 10. Visualization of the synthetic camera baseline experiment. Top row: *Base* training view and *Angle* training views. GS training is done on *Base* + another *Angle* view. Bottom rows: Qualitative examples of depth UE.

For both scenes, we train a GS model four times, each using only two training views. The two training views consist of a *Base* view and an *Angle* view, resulting in camera baselines (i.e., angles between the two views) of 22°, 43°, 65°, and 86°. Fig. 10 shows the training view candidates in the top row, alongside qualitative examples of PRIMU depth uncertainty maps for the camera baseline angle of 22° in the bottom two rows. Since only two training views are available, we use only direction-independent uncertainty feature maps. We use six holdout views, which are located at angles between the training views, to train the meta-regression model and perform the UE evaluation. In a cross-validation setup, we first train a meta-regression model for each hold-out view, before evaluating its performance on the remaining views.

Tab. 9 reports the mean AUSE scores for depth UE as well as the ℓ_1 depth error for both scenes across

different camera baselines. We observe that uncertainty prediction improves for the textured cube, as indicated by lower AUSE scores. Gaussian Splatting generally produces inaccurate depth predictions for both scenes. The textured cube scene typically exhibits slightly higher ℓ_1 depth errors across the camera baselines. However, the overall depth UE performance remains consistent with the results reported in the main manuscript and is largely independent of the camera baseline.

References

[1] Luca Savant Aira, Diego Valsesia, and Enrico Magli. Modeling uncertainty for gaussian splatting. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11657–11663, 2025. 2, 3

[2] Niki Amini-Naieni, Tomas Jakab, Andrea Vedaldi, and Ronald Clark. Instant uncertainty calibration of nerfs using a meta-calibrator. In *European Conference on Computer Vision*, pages 309–324. Springer, 2025. 2

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 2, 5, 8

[4] Nicholas I Fisher, Toby Lewis, and Brian JJ Embleton. *Statistical analysis of spherical data*. Cambridge university press, 1993. 4

[5] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' rays: Uncertainty quantification for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20061–20070, 2024. 2, 3, 6, 5

[6] Chenyu Han and Corentin Dumery. View-dependent uncertainty estimation of 3d gaussian splatting. *arXiv preprint arXiv:2504.07370*, 2025. 3

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 2

[8] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 2

[9] Letian Huang, Jiayang Bai, Jie Guo, and Yanwen Guo. Gs++: Error analyzing and optimal gaussian splatting. *arXiv preprint arXiv:2402.00752*, 2(3), 2024. 2

[10] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–667, 2018. 5

[11] Wen Jiang, Boshu Lei, and Kostas Daniilidis. Fisherrf: Active view selection and uncertainty quantification for radiance fields using fisher information. *arXiv*, 2023. 2, 3, 5, 6, 8

[12] Jaewoo Jung, Jisang Han, Honggyu An, Jiwon Kang, Seonghoon Park, and Seungryong Kim. Relaxing accurate initialization constraint for 3d gaussian splatting. *arXiv preprint arXiv:2403.09413*, 2024. 2

[13] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 2

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 2, 6, 8, 3

[15] Georgios Kopanas and George Drettakis. Improving NeRF Quality by Progressive Camera Placement for Free-Viewpoint Navigation. In *Vision, Modeling, and Visualization*. The Eurographics Association, 2023. 3

[16] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. WildGaussians: 3D gaussian splatting in the wild. *NeurIPS*, 2024. 2

[17] Ruiqi Li and Yiu-ming Cheung. Variational multi-scale representation for estimating uncertainty in 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:87934–87958, 2024. 2, 3, 5, 6

[18] Kehan Long, Cheng Qian, Jorge Cortés, and Nikolay Atanasov. Learning barrier functions with memory for robust safe navigation. *IEEE Robotics and Automation Letters*, 6(3):4931–4938, 2021. 2

[19] Linjie Lyu, Ayush Tewari, Marc Habermann, Shunsuke Saito, Michael Zollhöfer, Thomas Leimkühler, and Christian Theobalt. Manifold sampling for differentiable uncertainty in radiance fields. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 3, 5, 6, 8

[20] Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009. 4

[21] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021. 2

[22] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024. 2

[23] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2): 99–108, 1995. 2

[24] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019. 5

[25] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2

[26] Claus Müller. *Spherical harmonics*. Springer, 2006. 4

[27] Kiyohiro Nakayama, Mikaela Angelina Uy, Yang You, Ke Li, and Leonidas J Guibas. Provnerf: Modeling per point

provenance in nerfs as a stochastic field. *Advances in Neural Information Processing Systems*, 37:99145–99160, 2024. 2

[28] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013. 2, 5

[29] Jeremy Ocampo, Matthew A Price, and Jason D McEwen. Scalable and equivariant spherical cnns by discrete-continuous (disco) convolutions. *arXiv preprint arXiv:2209.13603*, 2022. 2

[30] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *Computer Vision – ECCV 2022*, pages 230–246, Cham, 2022. Springer Nature Switzerland. 3

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 2

[32] Matthew A. Price and Jason D. McEwen. Differentiable and accelerated spherical harmonic and wigner transforms. *Journal of Computational Physics*, 510:113109, 2024. 4, 2

[33] Fabio Remondino, Ali Karami, Ziyang Yan, Gabriele Mazzacca, Simone Rigon, and Rongjun Qin. A critical analysis of nerf-based 3d reconstruction. *Remote Sensing*, 15(14): 3585, 2023. 2

[34] Seunghyeon Seo, Yeonjin Chang, and Nojun Kwak. Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22883–22893, 2023. 2

[35] Afshar Shamsi, Hamzeh Asgharnezhad, Shirin Shamsi Jokandan, Abbas Khosravi, Parham M Kebria, Darius Nahavandi, Saeid Nahavandi, and Dipti Srinivasan. An uncertainty-aware transfer learning-based framework for covid-19 diagnosis. *IEEE transactions on neural networks and learning systems*, 32(4):1408–1417, 2021. 2

[36] Jianxiong Shen, Antonio Agudo, Francesc Moreno-Noguer, and Adria Ruiz. Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 2, 5

[37] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. Association for Computing Machinery, 2006. 2

[38] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2012. 5

[39] Niko Sünderhauf, Jad Abou-Chakra, and Dimity Miller. Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9370–9376, 2023. 2, 3

[40] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 2

[41] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[43] Frederik Warburg, Ethan Weber, Matthew Tancik, Aleksander Holynski, and Angjoo Kanazawa. Nerfbusters: Removing ghostly artifacts from casually captured nerfs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18120–18130, 2023. 2

[44] Joey Wilson, Marcelino Almeida, Min Sun, Sachit Mahajan, Maani Ghaffari, Parker Ewen, Omid Ghasemalizadeh, Cheng-Hao Kuo, and Arnie Sen. Modeling uncertainty in 3d gaussian splatting through continuous semantic splatting, 2024. 2, 3

[45] Joey Wilson, Marcelino Almeida, Sachit Mahajan, Martin Labrie, Maani Ghaffari, Omid Ghasemalizadeh, Min Sun, Cheng-Hao Kuo, and Arnab Sen. POp-GS: Next Best View in 3D-Gaussian Splatting with P-Optimality. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3646–3655, 2025. 3

[46] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024. 5, 6

[47] 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 Transactions on Graphics*, 35(3), 2016. 5

[48] Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. Cor-gs: sparse-view 3d gaussian splatting via co-regularization. In *European Conference on Computer Vision*, pages 335–352. Springer, 2024. 2

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[50] Yuhang Zhang, Joshua Maraval, Zhengyu Zhang, Nicolas Ramin, Shishun Tian, and Lu Zhang. Evaluating human perception of novel view synthesis: Subjective quality assessment of gaussian splatting and nerf in dynamic scenes. *arXiv preprint arXiv:2501.08072*, 2025. 2

[51] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, pages 29–538. IEEE, 2001. 2, 3