VarSplat: Uncertainty-aware 3D Gaussian Splatting for Robust RGB-D SLAM

Supplementary Material

In Supplementary Material, we provide implementation details with hyperparameters and per-scene results that support the conclusions in the main paper.

- **Implementation.** System hardware and software details, along with changes to the rasterizer to render appearance variance.
- **Additional results.** Per scene PSNR, SSIM, and LPIPS on Replica, TUM RGB-D, and ScanNet
- **Limitations and future work.** Discussion of current limitations and directions for applying appearance variance in dynamic scenarios.

S1. Implementation.

System Details. We implement VarSplat in Python 3.10 and PyTorch 2.4.1 on NVIDIA A100 80 GB GPU with CUDA 12.6. Starting from the original 3DGS rasterizer [5] and its depth-rendering extension with pose [12], we extend the renderer to propagate per-splat appearance variance via the law of total variance (Eq. 9 in the main paper) to obtain a differentiable per-pixel uncertainty map.

Hyperparameters. Defaults follow LoopSplat [13] for fair comparison. We report the settings used in our experiments in Table S1, including λ_c for tracking loss, learning rate l_r for rotation and l_t for translation, and optimization iterations iter_t and iter_m for tracking and mapping processes, τ for variance-based weighting. Specifically, τ controls the sharpness of the uncertainty-aware weighting function. For scenes where uncertainty-based regularization is less critical or requires minimal calibration, we set $\tau = 100$ to effectively maintain nearly uniform weight distribution across pixels and splats, thereby softening the penalty on high-variance regions. Unless noted, $\lambda_{\text{color}}, \lambda_{\text{depth}}, \lambda_{\text{reg}}$ are set to 1 and λ_{var} is set to 0.0001 in mapping loss $\mathcal{L}_{\text{map}}$ for all datasets.

Table S1. Per-Dataset Hyperparameters.

Params	Replica [9]	TUM-RGBD [10]	ScanNet [2]	ScanNet++ [11]
λ_c	0.95	0.6	0.6	0.5
l_r	0.0002	0.002	0.002	0.002
l_t	0.002	0.01	0.01	0.01
iter_t	60	200	200	300
iter_m	100	100	100	500
τ	10	50	5	10

Tracking loss. In the tracking loss, the inlier mask M_{inlier} filters pixels whose depth residual is extreme. Concretely, a pixel is removed if its depth error exceeds 50 times the median depth error in the current frame. Pixels without valid depth are also excluded from pose optimization. For soft alpha masking $M_{\text{alpha}} = \alpha^3$, we follow prior work [12, 13] for loss weighting. On ScanNet++, we reinitialize the current-frame pose with ICP odometry [6] whenever the tracking loss exceeds 50 times the running average.

Submap. Based on motion heristics, new submap is triggered with displacement threshold $d_{\text{thre}} = 0.5[m]$ and rotation threshold $\theta_{\text{thre}} = 50°$[13]. Due to motion blur and far camera poses in ScanNet and ScanNet++, we use a different approach for submap initialization as setting fixed iterations of 50 and 100 frames. For new keyframe, we uniformly sample M_k points to meet alpha value condition or depth discrepancy condition. M_k is limited for per-dataset setting, 30k for TUM-RGBD and ScanNet, while 100K for ScaNet++, and all available points for Replica. The alpha threshold α_{thre} is set to 0.98 for all datasets. The depth discrepancy condition is depth error passes 40 times median depth error of current frame. New Gaussians are initialized with opacity 0.5 and initial scales to the nearest neighbor. After finishing mapping optimization, we prune Gaussians based on fixed opacity threshold with 0.1 for Replica and 0.5 for remaining datasets.

Loop Detection. We use NetVLAD [1] with the VGG16-NetVLAD-Pitts30K weights from HLoc [8], similar to [13]. For each submap, we compute cosine similarities among its keyframes and define a self-similarity score s^i_{self} as the p-th percentile of these values. We set $p = 50$ on Replica, TUM-RGBD, and ScanNet, and $p = 33$ on ScanNet++. After obtaining submap-to-submap similarities, we rescore them with the submap-level reliability ratio $r^{(s)}$ derived from per-splat variances, so $\text{sim}_k = \text{cross\_sim}_k r^{(q)} r^{(k)}$. We then filter candidates by an overlap ratio computed with the front-end poses.

3DGS Registration For each accepted loop, we select the top $k = 2$ overlapping viewpoints using NetVLAD and solve a multi-view pose estimation between the two submaps [13]. We optimize the relative pose and per-view exposure coefficients for the selected viewpoints. The registration objective uses the photometric loss weighted by the per-pixel variance weight $\tilde{w_p}$ and an unweighted depth loss, as defined in the tracking and registration losses earlier. During registration, variance parameters are fixed.

Datasets. The DSLR sequences contain abrupt motions, so we evaluate only the first 250 frames of each sequence. Table S2 reports, for all evaluated scenes, the average frame to frame translation, rotation, and the frame count. Among the benchmarks, ScanNet++ shows about $10\times$ larger motion per frame than the others, which makes pose estimation more challenging and more prone to drift, highlighting the effectiveness of our approach in reducing drift for real-world SLAM.

Table S2. Frame Motion on Replica [9], TUM-RGBD [10], ScanNet [2], and ScanNet++ [11].

Dataset	Replica	TUM	ScanNet	ScanNet++
Translation (cm)	1.07	1.39	1.34	14.77
Rotation (°)	0.50	1.37	0.69	13.43

S2. Additional Results.

In the main paper we report dataset level averages for rendering quality. Tables S3 to S5 list per-scene results for Replica, TUM RGB-D, and ScanNet. Across the three datasets, VarSplat is competitive on most scenes.

S3. Limitations and Future Work

Although VarSplat improves robustness, a depth-based approach for where and when to add Gaussians limits performance when depth is sparse or missing. As seen in the per-scene TUM-RGBD results, future work should explore joint learning of appearance and geometric uncertainty with depth completion. Additionally, learning and rendering variance add computation and memory. Future work includes variance sharing across splats, pruning, and lightweight approximations of the uncertainty map. Fianlly, Our experiments focus on mostly static scenes. Extending appearance variance to handle moving objects with variance-guided motion segmentation and dynamic mapping is a promising direction.

Table S3. Rendering Performance on Replica. * denotes evaluating on submaps instead of a global one.

Method	Metric	Rm0	Rm1	Rm2	Off0	Off1	Off2	Off3	Off4	Avg.
NICE-SLAM [14]	PSNR↑	22.12	22.47	24.52	29.07	30.34	19.66	22.23	24.94	24.42
	SSIM↑	0.689	0.757	0.814	0.874	0.886	0.797	0.801	0.856	0.809
	LPIPS↓	0.330	0.271	0.208	0.229	0.181	0.235	0.209	0.198	0.233
ESLAM [3]	PSNR↑	25.25	27.39	28.09	30.33	27.04	27.99	29.27	29.15	28.06
	SSIM↑	0.874	0.890	0.935	0.934	0.910	0.942	0.953	0.948	0.923
	LPIPS↓	0.315	0.296	0.245	0.213	0.254	0.238	0.186	0.210	0.245
Point-SLAM [7]	PSNR↑	32.40	34.08	35.50	38.26	39.16	33.99	33.48	33.49	35.17
	SSIM↑	0.974	0.977	0.982	0.983	0.986	0.960	0.960	0.979	0.975
	LPIPS↓	0.113	0.116	0.111	0.100	0.118	0.156	0.132	0.142	0.124
SplaTAM [4]	PSNR↑	32.86	33.89	35.25	38.26	39.17	31.97	29.70	31.81	34.11
	SSIM↑	0.980	0.970	0.980	0.980	0.980	0.970	0.950	0.950	0.970
	LPIPS↓	0.070	0.100	0.080	0.090	0.090	0.100	0.120	0.150	0.100
* Gaussian-SLAM [12]	PSNR↑	38.88	41.80	42.44	46.40	45.29	40.10	39.06	42.65	42.08
	SSIM↑	0.993	0.996	0.996	0.998	0.997	0.997	0.997	0.997	0.996
	LPIPS↓	0.017	0.018	0.019	0.015	0.016	0.020	0.020	0.020	0.018
LoopSplat [13]	PSNR↑	33.07	35.32	36.16	40.82	40.21	34.67	35.67	37.10	36.63
	SSIM↑	0.973	0.978	0.985	0.992	0.990	0.985	0.990	0.989	0.985
	LPIPS↓	0.116	0.122	0.111	0.085	0.123	0.140	0.096	0.106	0.112
VarSplat	PSNR↑	**33.93**	**35.82**	**36.50**	**41.06**	**41.12**	**35.61**	**36.05**	**37.49**	**37.16**
	SSIM↑	0.978	**0.981**	**0.986**	**0.992**	**0.991**	**0.986**	**0.990**	**0.990**	**0.986**
	LPIPS↓	0.105	0.117	0.109	**0.082**	0.120	0.137	**0.093**	**0.100**	0.109

Table S4. Rendering Performance on TUM RGB-D. * denotes evaluating on submaps instead of a global one.

Method	Metric	fr1/desk	fr2/xyz	fr3/office	Avg.
NICE-SLAM [14]	PSNR↑	13.83	17.87	12.89	14.86
	SSIM↑	0.569	0.718	0.554	0.614
	LPIPS↓	0.482	0.344	0.498	0.441
ESLAM [3]	PSNR↑	11.29	17.46	17.02	15.26
	SSIM↑	0.666	0.310	0.457	0.478
	LPIPS↓	0.358	0.698	0.652	0.569
Point-SLAM [7]	PSNR↑	13.87	17.56	18.43	16.62
	SSIM↑	0.627	0.708	0.754	0.696
	LPIPS↓	0.544	0.585	0.448	0.526
SplaTAM [4]	PSNR↑	22.00	24.50	21.90	22.80
	SSIM↑	0.857	0.947	0.876	0.893
	LPIPS↓	0.232	0.100	0.202	0.178
* Gaussian-SLAM [12]	PSNR↑	24.01	25.02	26.13	25.05
	SSIM↑	0.924	0.924	0.939	0.929
	LPIPS↓	0.178	0.186	0.141	0.168
LoopSplat [13]	PSNR↑	22.03	22.68	23.47	22.72
	SSIM↑	0.849	0.892	0.879	0.873
	LPIPS↓	0.307	0.217	0.253	0.259
VarSplat	PSNR↑	**22.03**	23.85	**23.53**	**23.14**
	SSIM↑	0.847	0.920	**0.882**	0.883
	LPIPS↓	0.311	0.189	0.248	0.248

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 1

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2

[3] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. ESLAM: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17408–17419, 2023. 2, 3

[4] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplaTAM: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 2, 3

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[6] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 143–

Table S5. Rendering Performance on ScanNet. * denotes evaluating on submaps instead of a global one.

Method	Metric	0000	0059	0106	0169	0181	0207	Avg.
NICE-SLAM [14]	PSNR↑	18.71	16.55	17.29	18.75	15.56	18.38	17.54
	SSIM↑	0.641	0.605	0.646	0.629	0.562	0.646	0.621
	LPIPS↓	0.561	0.534	0.510	0.534	0.602	0.552	0.548
ESLAM [3]	PSNR↑	15.70	14.48	15.44	14.56	14.22	17.32	15.29
	SSIM↑	0.687	0.632	0.628	0.656	0.696	0.653	0.658
	LPIPS↓	0.449	0.450	0.529	0.486	0.482	0.534	0.488
Point-SLAM [7]	PSNR↑	21.30	19.48	16.80	18.53	18.23	20.56	19.16
	SSIM↑	0.806	0.765	0.676	0.688	0.823	0.750	0.751
	LPIPS↓	0.485	0.499	0.544	0.542	0.471	0.544	0.514
SplaTAM [4]	PSNR↑	19.33	19.27	17.73	21.97	16.76	19.80	19.14
	SSIM↑	0.660	0.792	0.690	0.776	0.683	0.696	0.716
	LPIPS↓	0.438	0.289	0.376	0.281	0.402	0.341	0.358
* Gaussian-SLAM [12]	PSNR↑	28.54	26.21	23.27	28.60	27.79	28.63	27.67
	SSIM↑	0.926	0.934	0.926	0.917	0.923	0.913	0.923
	LPIPS↓	0.271	0.211	0.217	0.226	0.277	0.288	0.248
LoopSplat [13]	PSNR↑	24.99	23.23	23.35	26.80	24.82	26.33	24.92
	SSIM↑	0.840	0.831	0.846	0.877	0.824	0.854	0.845
	LPIPS↓	0.450	0.400	0.409	0.346	0.514	0.430	0.425
VarSplat	PSNR↑	**25.12**	23.16	**23.52**	**26.82**	24.67	26.20	**24.92**
	SSIM↑	**0.850**	**0.834**	**0.852**	**0.879**	0.820	**0.854**	**0.848**
	LPIPS↓	0.444	0.399	0.404	0.340	0.518	0.425	0.422

152, 2017. 1

[7] Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. Point-SLAM dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18433–18444, 2023. 2, 3

[8] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12716–12725, 2019. 1

[9] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 2

[10] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 1, 2

[11] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 1, 2

[12] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-SLAM: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. 1, 2, 3

[13] Liyuan Zhu, Yue Li, Erik Sandström, Shengyu Huang, Konrad Schindler, and Iro Armeni. LoopSplat: Loop closure by registering 3d gaussian splats. In *2025 International Conference on 3D Vision (3DV)*, pages 156–167. IEEE, 2025. 1, 2, 3

[14] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-SLAM: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022. 2, 3