

# LoopSplat: Loop Closure by Registering 3D Gaussian Splats

## Supplementary Material

### Abstract

*This supplementary material includes a video of LoopSplat running on a multi-room scene, showcasing the effectiveness of the online loop closure module of LoopSplat. Furthermore, we provide the implementation details and statistics on loop closure and pose graph optimization (PGO). We also present more qualitative results and ablation studies. Lastly, we discuss the limitations and future work.*

optimization iterations  $\text{iter}_t$  for tracking and  $\text{iter}_m$  for mapping on the reported Replica [15], TUM-RGBD [16], ScanNet [3], and ScanNet++ [19] datasets. Additionally we set  $\lambda_{\text{color}}$ ,  $\lambda_{\text{depth}}$ , and  $\lambda_{\text{reg}}$  to 1 in the mapping loss  $\mathcal{L}_{\text{render}}$  for all datasets.

Params	Replica	TUM-RGBD	ScanNet	ScanNet++
$\lambda_c$	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
$\text{iter}_t$	60	200	200	300
$\text{iter}_m$	100	100	100	500

Table B.1. Per-dataset Hyperparameters.

### Contents

A Video .....	1
B. Implementation Details .....	1
C Datasets .....	2
D Novel View Synthesis .....	3
E. Additional Analysis .....	3
F. Additional Qualitative Results .....	4
G Limitations and Future Work .....	4

### A. Video

We submit a video `loopsplat_0054.mp4`, demonstrating LoopSplat’s online tracking and reconstruction capabilities on ScanNet [3] `scene0054`. This video showcases the effectiveness of our globally consistent reconstruction process. The visualization includes the reconstructed mesh, the colorized camera trajectory that denotes average translation error from the ground truth trajectory – see heatmap legend on the right, and the point cloud observed from the current frame colored in blue. As the camera completes its scan of the first room, one can clearly observe the significant improvements achieved through loop closure. While substantial drift occurs in the bathroom and storage room (the leftmost room), the online loop closure (LC) module in LoopSplat successfully corrects the accumulated error when the camera revisits the first room at the end of the video. This correction highlights the robustness of our method in maintaining global consistency throughout the reconstruction process.

### B. Implementation Details

**Hyperparameters.** Tab. B.1 lists the hyperparameters used in our system, including  $\lambda_c$  in the tracking loss, learning rates  $l_r$  for rotation and  $l_t$  for translation, and the number of

**Submap Initialization.** A new submap is triggered based on motion heuristics with the displacement threshold  $d_{\text{thre}} = 0.5 [m]$  and rotation threshold  $\theta_{\text{thre}} = 50^\circ$ . For the ScanNet and ScanNet++ datasets, we adopted a different approach to submap initialization. Motion heuristics were not employed, primarily due to two factors: significant motion blur in ScanNet and substantial per-frame motion in ScanNet++ (cf. Tab. C.1). Instead, we implemented a fixed interval system for triggering new submaps. Specifically, we set intervals of 50 frames for ScanNet and 100 for ScanNet++.

**Frame-to-model Tracking.** Instead of estimating the current camera pose  $\mathbf{T}_j$  directly, we optimize the relative camera pose  $\mathbf{T}_{j-1,j}$  of frame  $j$  with respect to  $j-1$ . To achieve the equivalent of rendering at the current pose  $\mathbf{T}_j$ , we transform the submap with the relative transformation  $\mathbf{T}_{j-1,j}^{-1}$  and render from the last camera pose  $\mathbf{T}_{j-1}$  to get the rendered color  $\hat{\mathbf{I}}_j$  and depth  $\hat{\mathbf{D}}_j$ .

**Tracking Loss.** The inlier mask  $M_{\text{inlier}}$  in the tracking loss filters out pixels with depth errors 50 times larger than the median depth error of the current re-rendered depth map. Pixels without valid depth input are also excluded as the inconsistent re-rendering in those areas can hinder the pose optimization. For the soft alpha mask, we adopt  $M_{\text{alpha}} = \alpha^3$  for per-pixel loss weighting. On the ScanNet++ dataset, if at the initialized pose the re-rendering loss is 50 times larger than the running average during tracking optimization, we use ICP odometry [11] to re-initialize the pose for the current frame.

**Submap Expansion.** When selecting candidates to add to the submap at a new keyframe, we uniformly sample  $M_k$  points from pixels that meet either the alpha value condition or the depth discrepancy condition.  $M_k$  is set to  $30K$  for TUM-RGBD and ScanNet datasets,  $100K$  for ScanNet++, and all available points that meet either condition for Replica. The alpha threshold  $\alpha_{\text{thre}}$  is set to 0.98 across all datasets. The depth discrepancy condition masks pixels

where the depth error exceeds 40 times the median depth error of the current frame.

**Submap Update.** The radius  $\rho$  for the neighborhood check when adding new Gaussians to the submap is set to  $1cm$ . Newly added Gaussians are initialized with opacity values 0.5 and their initial scales are set to the nearest neighbor distances within the submap. As mentioned in the main paper, the Gaussians are not pruned until optimization finishes. After the mapping optimization for the new keyframe, we prune Gaussians that have opacity values lower than a threshold  $o_{thre}$ . We set  $o_{thre} = 0.1$  for Replica and 0.5 for all other datasets.

**Submap Merging.** Upon completing the mapping and tracking of all frames for the input sequence, we merge the saved submaps into a global map. The mesh is extracted by TSDF fusion [2] using the rendered depth maps and estimated poses from the submaps. Then we use the reconstructed mesh vertices to initialize the Gaussian centers of the global map, providing a good starting point as they represent the scene geometry. We perform color refinement on the global map for 30K iterations using the same hyperparameters as in [8]. The Gaussian parameters of the global map are optimized from scratch using the RGB-D input and our estimated camera poses.

Params	Replica	TUM RGB-D	ScanNet	ScanNet++
$lr_{rotation}$	0.003	0.015	0.015	0.015
$lr_{translation}$	0.001	0.005	0.005	0.005
$lr_{exposure}$	0.1	0.1	0.1	0.1
$overlap_{min}$	0.1	0.2	0.2	0.2
$interval_{min}$	2	4	3	1

Table B.2. Per-dataset Hyperparameters on Loop Closure.

**Loop Detection.** For NetVLAD [1], we use the pretrained weights VGG16-NetVLAD-Pitts30K from HLoc [13]. We compute the cosine similarities of all keyframes within the  $i$ -th submap and determine the self-similarity score  $s_{self}^i$  corresponding to their  $p$ -th percentile. We set  $p = 50$  on Replica, TUM RGB-D, and ScanNet and  $p = 33$  on ScanNet++. After getting the initial loops from the visual similarity between submaps, we further filter detected loops by computing their overlap ratio (OR) using the pose estimated from the front-end. The overlapping region between  $\mathbf{P}$  and  $\mathbf{Q}$  is defined as in [5]:

$$OR = \frac{1}{|\mathcal{K}_{ij}|} \sum_{(\mathbf{p}, \mathbf{q}) \in \mathcal{K}_{ij}} [\|\mathbf{T}_{\mathbf{P} \rightarrow \mathbf{Q}}(\mathbf{p}) - \mathbf{q}\|_2 < \tau_1], \quad (1)$$

with  $[\cdot]$  the Iverson bracket and  $(\mathbf{p} \in \mathbf{P}, \mathbf{q} \in \mathbf{Q}) \in \mathcal{K}_{ij}$  the set of putative correspondences found by reciprocal matching the closest point between  $\mathbf{P}$  and  $\mathbf{Q}$ . We set  $\tau_1 = 0.1m$  on Replica and  $\tau_1 = 0.2m$  on TUM RGB-D, ScanNet, and ScanNet++. The selected thresholds are quite loose compared to standard point cloud registration, as we only need

to ensure that two submaps have a spatial overlap for the next step. We also remove the loops where two submaps are too temporally close to each other to avoid redundant computations. We set the minimum submap id interval ( $interval_{min}$ ) (cf. Tab. B.2) and remove the loop edges whose submap id distances are smaller than  $interval_{min}$ .

**3DGS Registration.** We first find the overlapping viewpoints between two submaps using NetVLAD, as discussed in the main paper. For all datasets, we select the top- $k$  pairs as the overlapping viewpoints,  $k = 2$ . In multi-view pose estimation, we optimize the camera pose parameters (*i.e.* translation and rotation) and the exposure coefficients for selected viewpoints because the exposure of renders in different submaps can differ. We set different learning rates of parameters in Tab. B.2. The learning rates of camera pose parameters are significantly smaller because Replica is a synthetic dataset with high-quality RGB-D measurements from rendering; thus, the step size for optimization should be smaller. The learning rates on the three real-world datasets are consistent with each other.

**Number of LCs.** We report the number of frames, submaps, and loop closures (LCs) for each scene in our LoopSplat system. On Replica scenes, LCs occur on average every 500 frames, about 4 times per scene (Tab. B.3a). The relatively low frequency of LCs in Replica is due to its single-room layouts and shorter sequences (approximately 2000 frames). In contrast, ScanNet [3] scenes feature longer sequences, averaging 4000 frames per scene (cf. Tab. B.3b). More challenging scenes like Scene 00, 54, and 233 require LoopSplat to create over 100 submaps and perform more than 30 pose graph optimizations (PGOs) per scene, which is attributed to their high sequence lengths. The TUM RGB-D dataset presents a mix of long and short sequences (cf. Tab. B.3c), resulting in varied numbers of submaps and PGOs across its scenes. This diversity in scene complexity and sequence length across datasets showcases the adaptability of LoopSplat to different scene capturing scenarios.

## C. Datasets

We first specify the ScanNet++ sequences used throughout our evaluation: (a) b20a261fdf, (b) 8b5caf3398, (c) fb05e13ad1, (d) 2e74812d00, (e) 281bc17764. Some sudden large motions occur in the DSLR-captured sequences. To avoid this, we only use the first 250 frames of each sequence. Tab. C.1 shows the average ground truth frame translation distance and rotation degree per dataset on the scenes (and frame length) we evaluated. The average motion on ScanNet++ is about  $10\times$  larger than in other datasets, making it a challenging dataset for accurate pose estimation and, hence, highlighting the robustness of LoopSplat given its superior performance on it.

Method	r0	r1	r2	o0	o1	o2	o3	o4	Avg.
# Frames	2000	2000	2000	2000	2000	2000	2000	2000	2000
# Submaps	38	25	33	27	11	39	45	39	32
# LCs	2	8	4	3	4	1	2	6	4

(a) Replica [14]

Method	00	54	59	106	169	181	207	233	Avg.
# Frames	5578	6629	1807	2324	2034	2349	1988	7643	4073
# Submaps	112	132	36	47	41	47	39	153	76
# LCs	48	36	17	4	11	15	19	55	26

(b) ScanNet [3]

Method	frl/desk1	frl/desk2	frl/room	fr2/xyz	fr3/office	Avg.
# Frames	595	640	1362	3669	2585	1770
# Submaps	14	15	24	6	39	20
# LCs	7	7	6	2	5	5

(c) TUM-RGBD [17]

Table B.3. Number of Submaps and PGOs Across Different Datasets.

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

Table C.1. Average Frame Motion Across Datasets.

## D. Novel View Synthesis

We evaluate the novel view synthesis (NVS) performance using the test set of the ScanNet++ sequences, where the test views are held-out and distant from training views. PSNR is evaluated on all test views after 10K iterations of global map refinement using the image resolution of  $876 \times 584$ . Tab. D.1 shows that ours yields the best NVS results. For the baselines, we implement the evaluation using their open-sourced code.

Method	a	b	c	d	e	Avg.
ESLAM [10]	13.63	11.86	11.83	10.59	10.64	11.71
SplaTAM [7]	23.95	22.66	13.95	8.47	20.06	17.82
Gaussian-SLAM [21]	<b>26.66</b>	<b>24.42</b>	15.01	18.35	21.91	21.27
LoopSplat (Ours)	25.60	23.65	<b>15.87</b>	<b>18.86</b>	<b>22.51</b>	<b>21.30</b>

Table D.1. Novel View Synthesis on ScanNet++ [19] (PSNR  $\uparrow$  [dB]). For the baselines, results were obtained using the open-sourced code with our implementation for the NVS evaluation. PSNR calculations include all pixels, regardless of whether they have valid depth input. LoopSplat yields the best results.

## E. Additional Analysis

**Rendering Performance at Scene Level.** In the main paper, we only report the average rendering performance on each dataset. Tab. E.1, Tab. E.2, and Tab. E.3 report the per-scene rendering performance on Replica, TUM RGB-D, and ScanNet, respectively. On Replica and ScanNet, LoopSplat has the best performance on most of the scenes and on TUM RGB-D, LoopSplat is only second to SplaTAM [7].

Method	Metric	Rm0	Rm1	Rm2	Off0	Off1	Off2	Off3	Off4	Avg.
NICE-SLAM [23]	PSNR $\uparrow$	22.12	22.47	24.52	29.07	30.34	19.66	22.23	24.94	24.42
	SSIM $\uparrow$	0.689	0.757	0.814	0.874	0.886	0.797	0.801	0.856	0.809
	LPIPS $\downarrow$	0.330	0.271	0.208	0.229	0.181	0.235	0.209	0.198	0.233
Vox-Fusion [18]	PSNR $\uparrow$	22.39	22.36	23.92	27.79	29.83	20.33	23.47	25.21	24.41
	SSIM $\uparrow$	0.683	0.751	0.798	0.857	0.876	0.794	0.803	0.847	0.801
	LPIPS $\downarrow$	0.303	0.269	0.234	0.241	0.184	0.243	0.213	0.199	0.236
ESLAM [6]	PSNR $\uparrow$	25.25	27.39	28.09	30.33	27.04	27.99	29.27	29.15	28.06
	SSIM $\uparrow$	0.874	0.89	0.935	0.934	0.910	0.942	0.953	0.948	0.923
	LPIPS $\downarrow$	0.315	0.296	0.245	0.213	0.254	0.238	0.186	0.210	0.245
Point-SLAM [12]	PSNR $\uparrow$	32.40	34.08	35.50	38.26	39.16	33.99	33.48	33.49	35.17
	SSIM $\uparrow$	0.974	0.977	0.982	0.983	0.986	0.960	0.960	0.979	0.975
	LPIPS $\downarrow$	0.113	0.116	0.111	0.100	0.118	0.156	0.132	0.142	0.124
SplaTAM [7]	PSNR $\uparrow$	32.86	33.89	35.25	38.26	39.17	31.97	29.70	31.81	34.11
	SSIM $\uparrow$	0.98	0.97	0.98	0.98	0.98	0.97	0.95	0.95	0.97
	LPIPS $\downarrow$	0.07	0.10	0.08	0.09	0.09	0.10	0.12	0.15	0.10
* Gaussian-SLAM [21]	PSNR $\uparrow$	38.88	41.80	42.44	46.40	45.29	40.10	39.06	42.65	42.08
	SSIM $\uparrow$	0.993	0.996	0.996	0.998	0.997	0.997	0.997	0.997	0.996
	LPIPS $\downarrow$	0.017	0.018	0.019	0.015	0.016	0.020	0.020	0.020	0.018
LoopSplat	PSNR $\uparrow$	<b>33.07</b>	<b>35.32</b>	<b>36.16</b>	<b>40.82</b>	<b>40.21</b>	<b>34.67</b>	<b>35.67</b>	<b>37.10</b>	<b>36.63</b>
	SSIM $\uparrow$	<b>0.973</b>	<b>0.978</b>	<b>0.985</b>	<b>0.992</b>	<b>0.990</b>	<b>0.985</b>	<b>0.990</b>	<b>0.989</b>	<b>0.985</b>
	LPIPS $\downarrow$	<b>0.116</b>	<b>0.122</b>	<b>0.111</b>	<b>0.085</b>	<b>0.123</b>	<b>0.140</b>	<b>0.096</b>	<b>0.106</b>	<b>0.112</b>

Table E.1. Rendering Performance on Replica [14]. \* denotes evaluating on submaps instead of a global one.

Method	Metric	frl/desk	fr2/xyz	fr3/office	Avg.
NICE-SLAM [23]	PSNR $\uparrow$	13.83	17.87	12.890	14.86
	SSIM $\uparrow$	0.569	0.718	0.554	0.614
	LPIPS $\downarrow$	0.482	0.344	0.498	0.441
Vox-Fusion [18]	PSNR $\uparrow$	15.79	16.32	17.27	16.46
	SSIM $\uparrow$	0.647	0.706	0.677	0.677
	LPIPS $\downarrow$	0.523	0.433	0.456	0.471
ESLAM [6]	PSNR $\uparrow$	11.29	17.46	17.02	15.26
	SSIM $\uparrow$	0.666	0.310	0.457	0.478
	LPIPS $\downarrow$	0.358	0.698	0.652	0.569
Point-SLAM [12]	PSNR $\uparrow$	13.87	17.56	18.43	16.62
	SSIM $\uparrow$	0.627	0.708	0.754	0.696
	LPIPS $\downarrow$	0.544	0.585	0.448	0.526
SplaTAM [7]	PSNR $\uparrow$	22.00	24.50	21.90	22.80
	SSIM $\uparrow$	0.857	0.947	0.876	0.893
	LPIPS $\downarrow$	0.232	0.100	0.202	0.178
* Gaussian-SLAM [21]	PSNR $\uparrow$	24.01	25.02	26.13	25.05
	SSIM $\uparrow$	0.924	0.924	0.939	0.929
	LPIPS $\downarrow$	0.178	0.186	0.141	0.168
LoopSplat	PSNR $\uparrow$	<b>22.03</b>	<b>22.68</b>	<b>23.47</b>	<b>22.72</b>
	SSIM $\uparrow$	<b>0.849</b>	<b>0.892</b>	<b>0.879</b>	<b>0.873</b>
	LPIPS $\downarrow$	<b>0.307</b>	<b>0.217</b>	<b>0.253</b>	<b>0.259</b>

Table E.2. Rendering Performance on TUM RGB-D [17]. \* denotes evaluating on submaps instead of a global one.

**Online LC.** We investigate the significance of applying LC and PGO online in LoopSplat, as opposed to applying them only after the entire run concludes. The online mode, as presented in our main paper, continuously performs LC and PGO during the SLAM process. In contrast, the offline mode delays these operations until the input stream ends, applying them only once. Results in Tab. E.4 reveal that for smaller scenes, such as those in Replica, online LC does not significantly improve performance due to the limited number of loops. However, in more complex environments like ScanNet and TUM RGB-D, online LC proves crucial to LoopSplat’s superior performance. This is because it con-

Method	Metric	0000	0059	0106	0169	0181	0207	Avg.
NICE-SLAM [23]	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
Vox-Fusion [18]	PSNR↑	19.06	16.38	18.46	18.69	16.75	19.66	18.17
	SSIM↑	0.662	0.615	0.753	0.650	0.666	0.696	0.673
	LPIPS↓	0.515	0.528	0.439	0.513	0.532	0.500	0.504
ESLAM [6]	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 [12]	PSNR↑	21.30	19.48	16.80	18.53	22.27	20.56	19.82
	SSIM↑	0.806	0.765	0.676	0.686	0.823	0.750	0.751
	LPIPS↓	0.485	0.499	0.544	0.542	0.471	0.544	0.514
SplaTAM [7]	PSNR↑	19.33	19.27	17.73	21.97	16.76	19.8	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	0.420	0.341	0.358
*Gaussian-SLAM [21]	PSNR↑	28.539	26.208	26.258	28.604	27.789	28.627	27.67
	SSIM↑	0.926	0.9336	0.9259	0.917	0.9223	0.9135	0.923
	LPIPS↓	0.271	0.211	0.217	0.226	0.277	0.288	0.248
LoopSplat (Ours)	PSNR↑	<b>24.99</b>	<b>23.23</b>	<b>23.35</b>	<b>26.80</b>	<b>24.82</b>	<b>26.33</b>	<b>24.92</b>
	SSIM↑	<b>0.840</b>	<b>0.831</b>	<b>0.846</b>	<b>0.877</b>	<b>0.824</b>	<b>0.854</b>	<b>0.845</b>
	LPIPS↓	0.450	0.400	0.409	0.346	0.514	0.430	0.425

Table E.3. **Rendering Performance on ScanNet [3].** \* denotes evaluating on submaps instead of a global one. We exclude these results from the comparison for not being fair and for evaluating an easier setting.

LC Mode	Replica	ScanNet	TUM RGB-D
Offline	0.26	15.27	12.54
Online	0.26	8.39	3.33

Table E.4. **Ablation Study on Offline LC.** (ATE [cm],↓)

Dataset	Replica	TUM-RGBD	ScanNet	ScanNet++
# Gaussians	295K	219K	331K	330K

Table E.5. **Average Number of Gaussians Per-scene.**

stantly corrects map drift, preventing cumulative errors that would otherwise degrade accuracy over time.

**Average Number of Gaussians Per Scene.** Tab. E.5 reports the average number of Gaussians after global map refinement for each dataset. For a room-sized scene, we obtain on average around 300K Gaussian splats, which is a reasonable number. The number of Gaussians is dependent on the scale of the scenes, the number of vertices used to initialize the Gaussians, and the number of densification iterations during the optimization of 3DGS.

## F. Additional Qualitative Results

In this section, we present additional qualitative results.

**Overlap Ratio.** We first illustrate the overlap ratio we adopt to determine if a detected loop is added to the pose graph. In Fig. G.1, we showcase three representative ScanNet submap pairs with descending overlap ratios.

**3DGS Registration.** Fig. G.2 presents more registration results on the submaps. The red arrows highlight the differences between odometry, ours, and ground truth. The odometry results have the most misalignment, whereas estimates from LoopSplat are closer to, or even better than, the ground truth through visual inspection.

**Mesh Reconstruction.** We present additional qualitative results for mesh reconstruction on ScanNet scenes 0059 and 0207 in Fig. G.3. Our analysis concentrates on regions with high geometric complexity. As evident from the results, LoopSplat consistently produces higher-quality and more consistent reconstructions compared to baseline methods, particularly in these challenging areas.

## G. Limitations and Future Work

**Limitations.** LoopSplat still faces certain limitations. As the number of submaps exceeds 100, the computational demands for pairwise registrations during pose graph optimization increase significantly, reducing the efficiency of the loop closure module. While LoopSplat demonstrates competitive performance and achieves the lowest peak GPU usage among all compared methods, there remains significant room to improve the system’s overall efficiency. The iterative nature of optimizing 3D Gaussians and camera poses limits the speed of the system. The pose initialization is based on the constant speed assumption, which can be improved with Kalman Filters. In terms of submap construction, we use different hyperparameters for different datasets, which is a standard practice in the SLAM community, but we believe it hinders the generalization ability of the system to in-the-wild data.

**Future Work.** Several promising avenues for future research emerge from this work. First, employing advanced mesh extraction methods that directly operate on 3DGS, such as SuGAR [4] or GOF [20], can improve the reconstruction performance. Second, integrating uncertainty estimates for each viewpoint could improve both overlap estimation and multi-view optimization in 3DGS registration. Additionally, exploring techniques to refine 3DGS reconstruction in overlapping regions between submaps presents another intriguing direction.



## References

- [1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 2
- [2] Brian Curless and Marc Levoy. Volumetric method for building complex models from range images. In *SIGGRAPH Conference on Computer Graphics*, 1996. 2
- [3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 1, 2, 3, 4, 8
- [4] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. 4
- [5] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021. 2
- [6] M. M. Johari, C. Carta, and F. Fleuret. ESLAM: Efficient dense slam system based on hybrid representation of signed distance fields. In *CVPR*, 2023. 3, 4
- [7] 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. *CVPR*, 2024. 3, 4
- [8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 2
- [9] Lorenzo Liso, Erik Sandström, Vladimir Yugay, Luc Van Gool, and Martin R. Oswald. Loopy-slam: Dense neural slam with loop closures. In *CVPR*, 2024. 8
- [10] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. pages arXiv-2211, 2022. 3
- [11] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *ICCV*, pages 143–152, 2017. 1
- [12] Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. Point-slam: Dense neural point cloud-based slam. In *ICCV*, 2023. 3, 4
- [13] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2
- [14] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 3
- [15] 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
- [16] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *International Conference on Intelligent Robots and Systems (IROS)*, 2012. 1
- [17] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 3
- [18] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2022. 3, 4
- [19] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 1, 3
- [20] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient high-quality compact surface reconstruction in unbounded scenes. *arXiv:2404.10772*, 2024. 4
- [21] 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. 3, 4, 8
- [22] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *ICCV*, 2023. 8
- [23] 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 *CVPR*, 2022. 3, 4

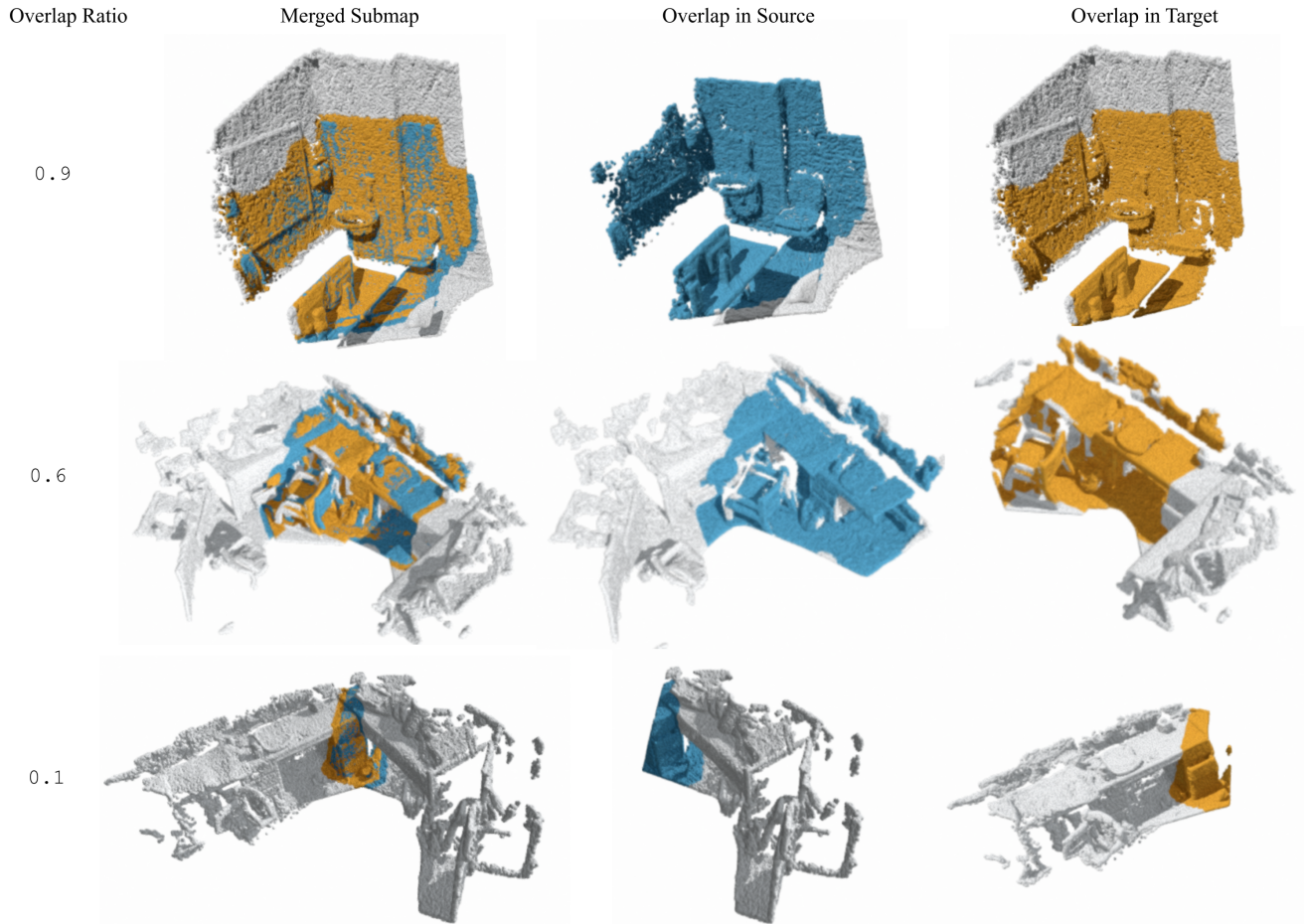


Figure G.1. **Qualitative Results of Overlap Ratio between Submaps.** We visualize the centers of 3D Gaussians as point clouds, with two submaps only colorized in the overlapping region. The top row demonstrates a large overlap between submaps with  $OR = 0.9$ . The middle row showcases a medium overlap of  $OR = 0.6$ , while the bottom row exhibits an extremely low overlap of  $OR = 0.1$ . This last case was rejected as a loop due to its insufficient overlap, which typically leads to low-accuracy registration or even complete failure.

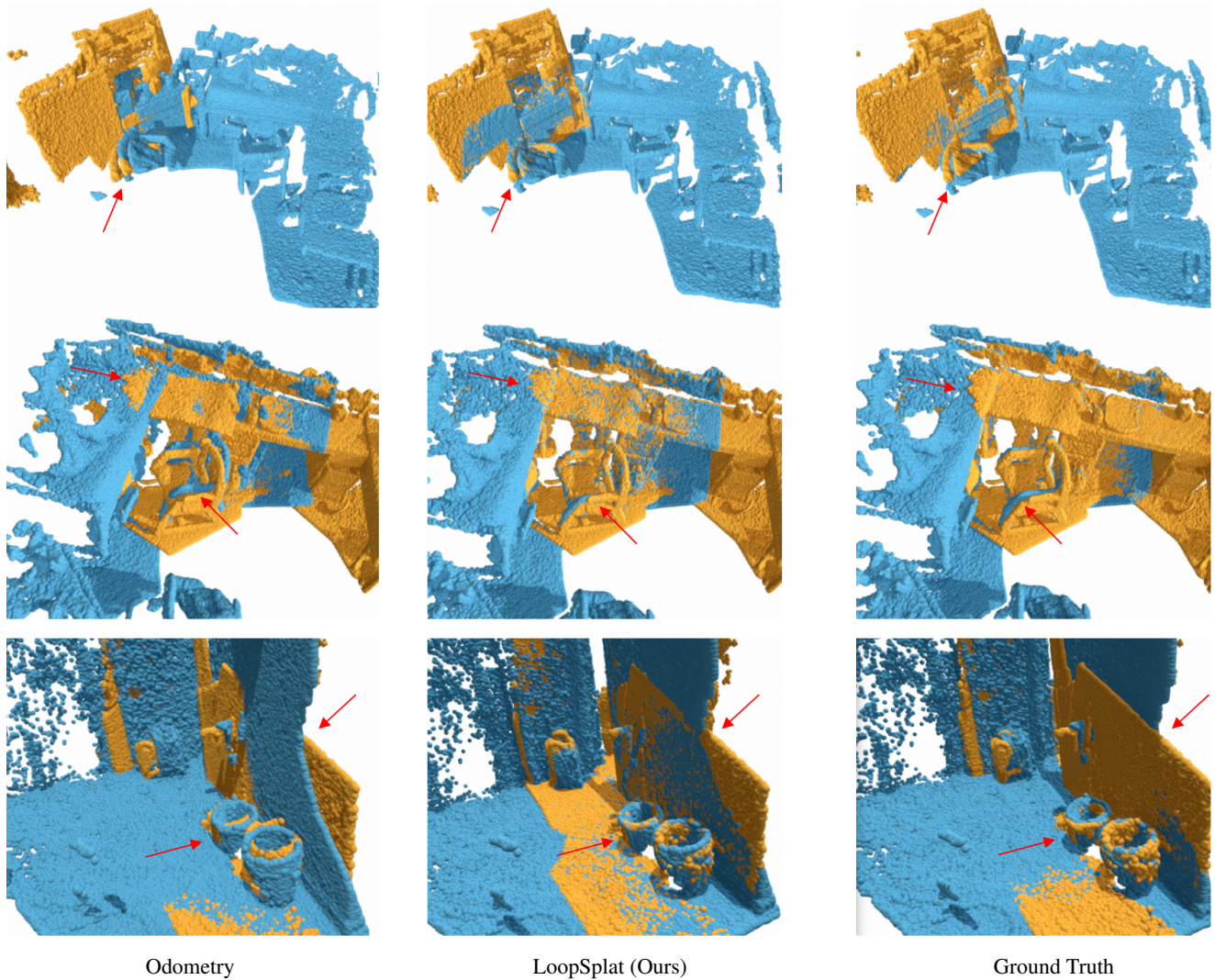
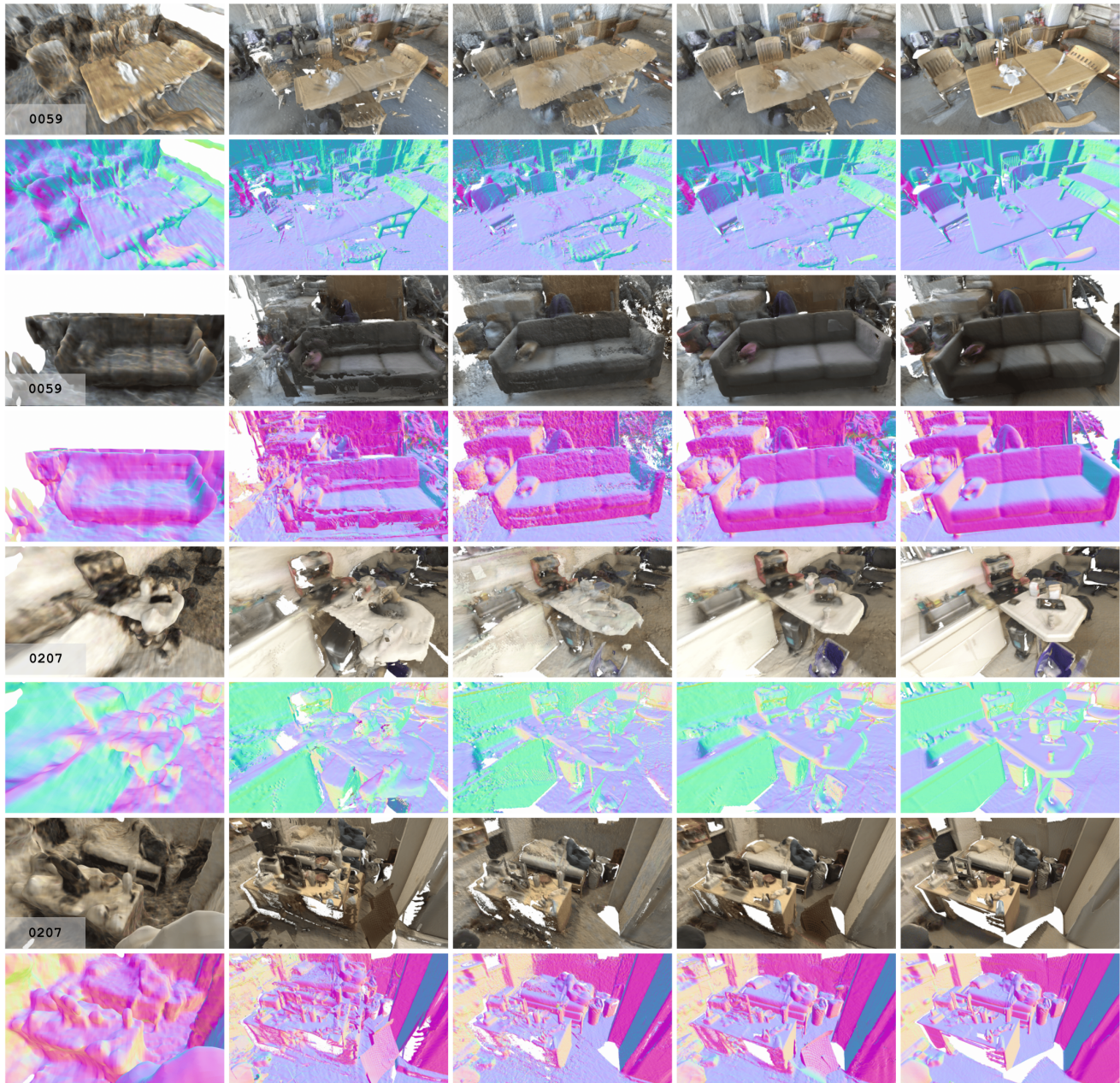


Figure G.2. **Qualitative Results on Submap Registration.** We visualize the centers of 3D Gaussians as point clouds, with two submaps colorized differently. LoopSplat consistently improves upon the initial odometry-based alignment and outperforms the pseudo ground truth. In the first row, LoopSplat (middle) achieves better alignment of the chair’s back compared to both odometry and ground truth. Similar improvements are observed in the second row. The last row demonstrates LoopSplat’s superior alignment of walls and trash cans. These results, representative of ScanNet and not cherry-picked, consistently showcase the method’s effectiveness across various scenes.





GO-SLAM [22]

Gaussian-SLAM [21]

Loopy-SLAM [9]

LoopSplat (Ours)

Ground Truth

Figure G.3. **Mesh Reconstruction on ScanNet [3] scenes 0059 and 0207.** Per example, the first row displays the colored mesh, while the second row shows the corresponding normals. LoopSplat demonstrates superior performance compared to baseline methods, excelling in both texture fidelity and geometric detail. Notably, our approach yields smoother and more complete mesh reconstructions than the strongest baseline, Loopy-SLAM.