SGS-SLAM: Semantic Gaussian Splatting For Neural Dense SLAM — Supplementary Material —

Mingrui Li[1⋆], Shuhong Liu[2*], Heng Zhou[3], Guohao Zhu[2], Na Cheng[1], Tianchen Deng[4], and Hongyu Wang[1]

[1] Dalian University of Technology [2] The University of Tokyo
[3] Columbia University [4] Shanghai Jiao Tong University

A Experiment Settings

In this section, we outline the experimental setup and hyperparameters applied in our studies. The experiments were conducted on a server with NVIDIA A100-40GB GPU. However, our method typically takes less than 12 GB of memory for the scenes presented in this study, making it compatible with any GPU that has more than this amount of memory. The ground-truth results we compared, particularly for the novel view rendering, were obtained from the ground-truth mesh provided in the dataset, which was generated in an offline manner. Therefore, some defects can be observed in the ground-truth results.

SGS-SLAM By default, both mapping and tracking operations are conducted for each frame. During the tracking phase, we set the silhouette visibility threshold, T_{sil}, to 0.99. The multi-channel optimization involves three parameters: $\lambda_D = 1.0$ for depth, $\lambda_C = 0.5$ for colors, and $\lambda_S = 0.05$ for semantic loss, with the semantic loss weight being comparatively low due to the typical noisiness of real-world semantic labels. Throughout the tracking, the multi-channel Gaussian parameters remain constant, adjusting only the camera parameters with a learning rate of 2e-3 for transition. Key-frames are initially chosen at intervals of every 5 frames, then refined based on geometric and semantic criteria. The geometric overlap threshold, η, is defined at 0.05, and the semantic mean Intersection over Union (mIoU) threshold, T_{sem}, at 0.7. The maximum number of keyframes per frame is limited to 25, considering the computation speed. The uncertainty decay coefficient, τ scales with the length of the input frame series. In the mapping process, the silhouette threshold T_{sil} is adjusted to 0.5. The weights of photometric loss are set to $\lambda_D = 1.0$, $\lambda_C = 0.5$, and $\lambda_S = 0.1$. Here, camera parameters are fixed, and Gaussian parameters are optimized, with specific learning rates for 3D position at 1e-4, color 2.5e-3, Gaussian rotation at 1e-3, logit opacity at 0.05, and log scale at 1e-3. Performance metrics of tracking and mapping are assessed every 5 frames, with mIoU scores evaluated at the same frequency.

⋆ These authors contributed equally to this work.

The mapping and tracking iteration steps are specific to each dataset, In the case of the Replica dataset [7], the number of iterations for tracking and mapping are set to 40 and 60. For the ScanNet dataset [2], tracking and mapping are set to 120 and 40. In the enhanced ScanNet++ dataset [11], where the camera transition is large between each frame, the tracking and mapping iterations are adjusted to 220 and 50.

Baselines We adhere to the default configurations for each baseline as reported in their papers. The evaluation metrics for tracking and mapping are consistent with those applied to our method. For baselines whose implementations are not publicly available, we present the results as reported in their papers.

B Additional Experiment Results

We provide additional quantitative analysis of camera tracking in Sec. B.1. The visualization of semantic segmentation compared with NeRF-based method is presented in Sec. B.2. More qualitative novel view rendering results are illustrated in Sec. B.3. We compared our method with Vox-Fusion [9], NICE-SLAM [12], Co-SLAM [8], ESLAM [4], and Point-SLAM [6] for ATE RMSE evaluation. For 3D semantic segmentation, we visualized the comparison with DNS-SLAM [5].

B.1 Camera Tracking

In this section, we break down the quantitative analysis on ATE RMSE [cm] on Replica [7], ScanNet [2], and ScanNet++ [11] datasets. Tab. 1, Tab. 2, and Tab. 3 present the evaluation our SGS-SLAM against baseline models on each dataset. Our method of estimating camera poses by directly optimizing the gradient on dense photometric loss achieves state-of-the-art tracking performance on datasets with high-quality RGB-D images. In particular, on the ScanNet++ dataset [11], where there is a large camera transition between successive frames, NeRF-based methods like ESLAM failed to track. Conversely, SGS-SLAM demonstrated robust and accurate tracking capability.

Table 1: Quantitative comparison of ATE RMSE [cm] between our method and the baselines for each scene of the Replica dataset [7]. Our method demonstrates SOTA performances.

Methods	Avg.	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4
Vox-Fusion	3.09	1.37	4.70	1.47	8.48	2.04	2.58	1.11	2.94
NICE-SLAM	2.50	2.25	2.86	2.34	1.98	2.12	2.83	2.68	2.96
Co-SLAM	0.86	0.65	1.13	1.43	0.55	0.50	0.46	1.40	0.77
ESLAM	0.63	0.71	0.70	0.52	0.57	0.55	0.58	0.72	0.63
Point-SLAM	0.52	0.61	0.41	0.37	0.38	0.48	0.54	0.69	0.72
Ours	**0.41**	**0.46**	**0.45**	**0.29**	**0.46**	**0.23**	**0.45**	**0.42**	**0.55**

Table 2: Quantitative comparison of ATE RMSE [cm] between our method and the baselines for the selected scenes on the ScanNet dataset [2].

Methods	Avg.	0000	0059	0106	0169	0181	0207
Vox-Fusion	26.90	68.84	24.18	8.41	27.28	23.30	9.41
NICE-SLAM	10.70	12.00	14.00	7.90	10.90	13.40	6.20
Co-SLAM	9.73	12.29	9.57	**6.62**	13.43	**7.13**	9.37
ESLAM	**7.88**	**8.47**	8.70	7.58	**7.45**	8.87	6.20
Point-SLAM	12.19	10.24	**7.81**	8.65	22.16	14.77	9.54
Ours	9.87	11.15	9.54	10.43	10.70	11.28	**6.11**

Table 3: Quantitative comparison of ATE RMSE [cm] between our method and the baseline for the selected scenes on the ScanNet++ dataset [11].

Methods	Avg. [cm]↓	8b5caf3398 [cm]↓	b20a261fdf [cm]↓
ESLAM	170.06	185.15	156.96
Ours	**1.62**	**0.65**	**2.34**

B.2 Semantic Segmentation

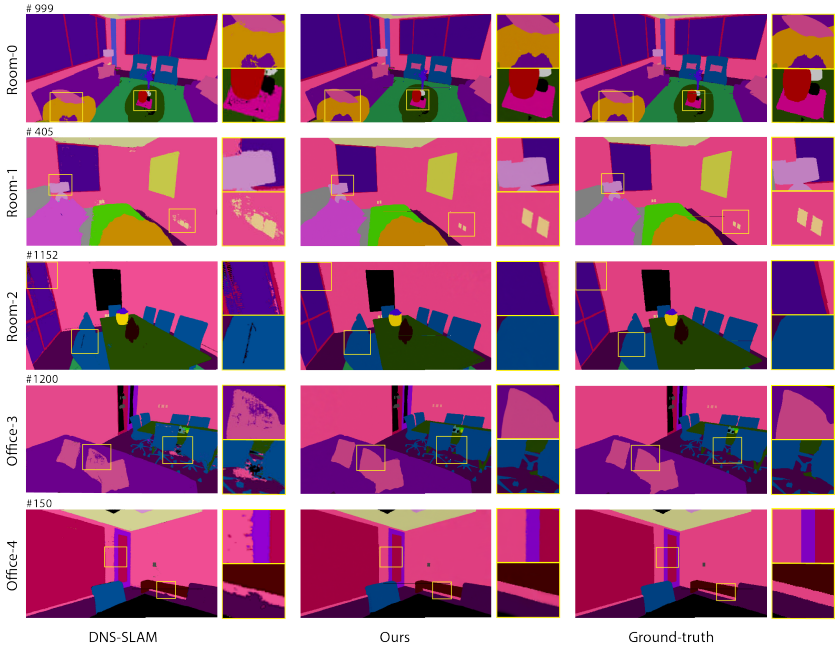

Fig. 1: Qualitative comparison of our method and DNS-SLAM [5] for semantic segmentation from the Replica dataset [7]. The visualization outcomes of DNS-SLAM [5] are obtained from its paper. The frames of the training view are chosen based on the results presented in DNS-SLAM. Compared to NeRF-based models, our approach delivers segmentation results with higher accuracy.

In this section, the outcomes of semantic segmentation on the Replica dataset [7] are visualized and compared with DNS-SLAM [5], a NeRF-based approach. As illustrated, our method offered accurate and detailed segmentation, whereas DNS-SLAM faces challenges in edges due to the over-smoothing issue of NeRF.

B.3 Novel View Rendering

We present additional results of novel view rendering using our method across the Replica [7], ScanNet [2], and ScanNet++ [11] datasets, with comparisons to ESLAM [4]. Visualizations are provided in Fig. 2, Fig. 3, Fig. 4, and Fig. 5 with semantic segmentation outcomes. Our method consistently delivers high-quality rendering results for both synthesized and real-world datasets. Notably, on the challenging real-world ScanNet++ dataset, ESLAM [4] struggled to reconstruct the scene. By contrast, SGS-SLAM provides accurate high-fidelity scene reconstructions along with precise segmentation outcomes. Note that the ground-truth segmentation labels are retrieved from the ground-truth mesh at the instance level, and therefore, our results also show instance-level segmentation.

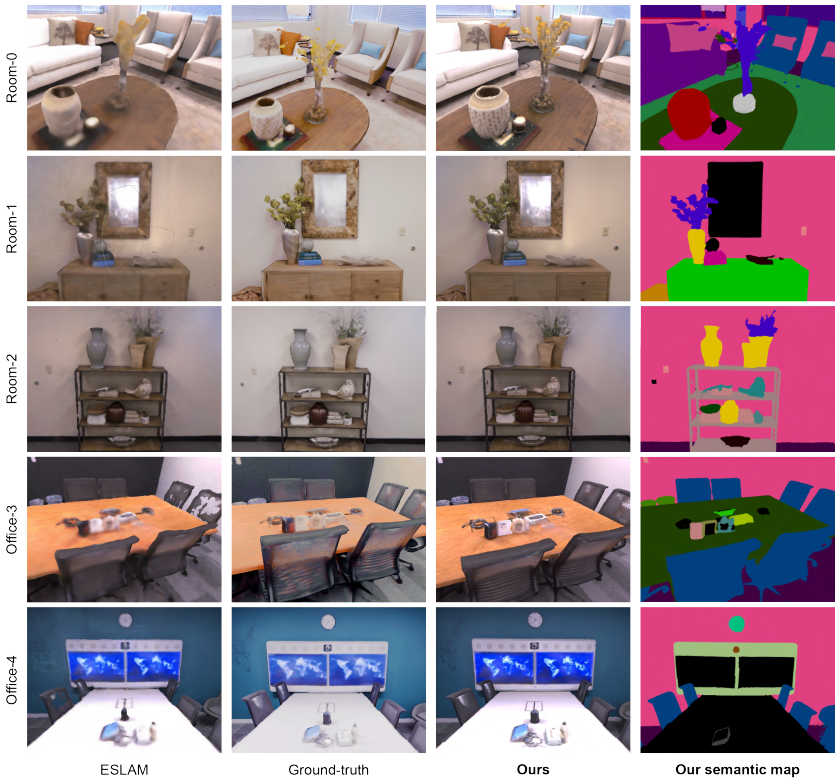

Fig. 2: The visualization of novel view rendering between the ESLAM [4] and our method on the Replica dataset [7]. The ground-truth novel views are captured from meshes provided by the dataset.

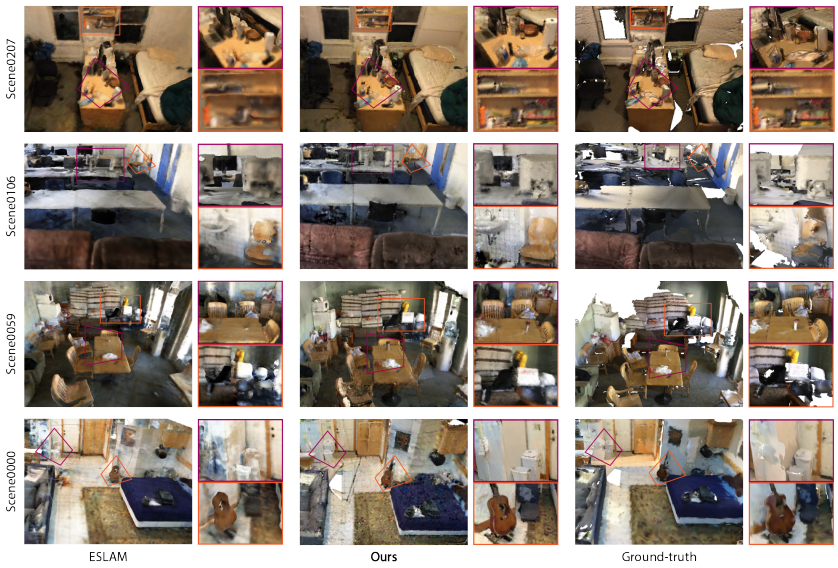

Fig. 3: The visualization of novel view rendering between the baseline and our method using the ScanNet dataset [2]. The ground-truth novel views are captured from meshes. SGS-SLAM exhibits rendering of high fidelity and outperforms the NeRF-based ESLAM [4]. In contrast to the ground-truth mesh, our method demonstrates robust mapping in areas where the ground-truth mesh presents holes.

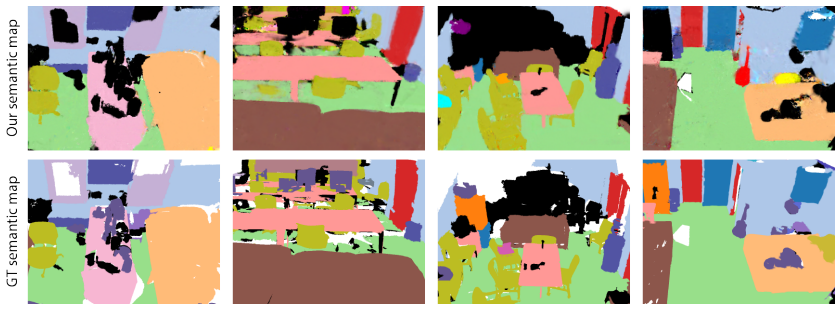

Fig. 4: The visualization of 3D semantic segmentation results of SGS-SLAM, as applied to the novel views selected in Fig. 3. Note that the rendering results exhibit minor variations in scene objects due to the use of a modified semantic dataset from ScantNet. For our method, the training data is processed from the filtered semantic labels using the nyu40-class, where certain objects are not distinctly labeled and are assigned as background (depicted in black). Furthermore, we introduce extra labels, like guitar, bag, and basket, to enhance the quality of scene reconstruction.

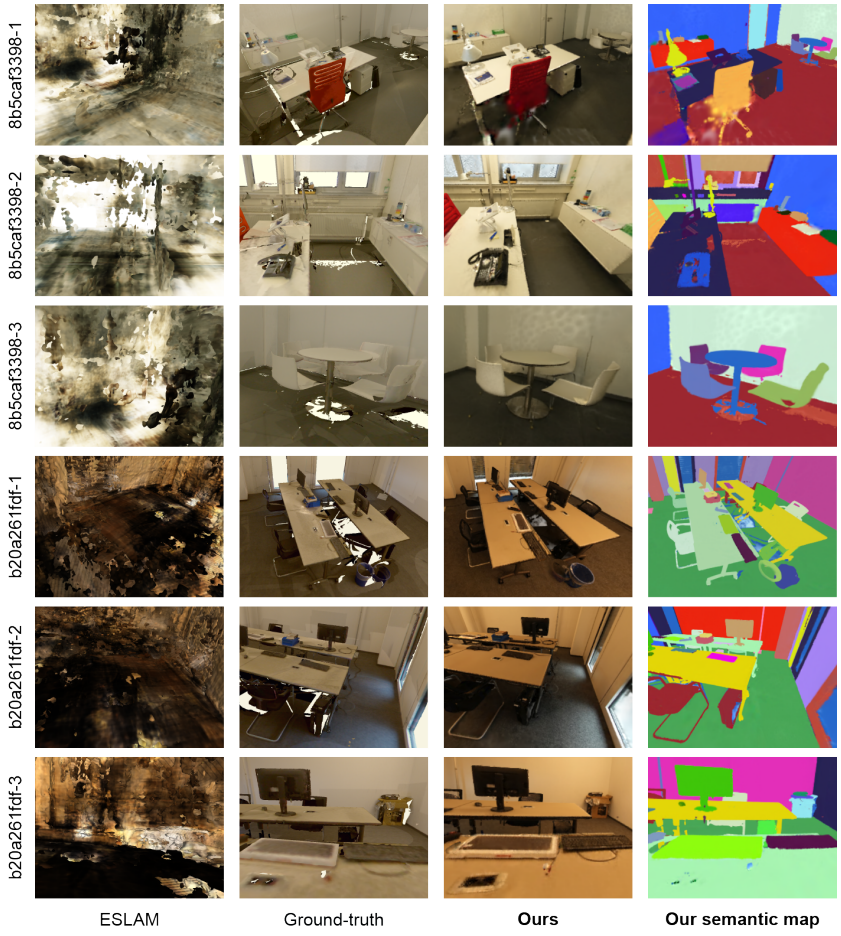

Fig. 5: The visualization of novel view rendering between the baseline and our method using the ScanNet++ dataset [11]. The ground-truth novel views are captured from meshes. SGS-SLAM demonstrates superior rendering quality, while ESLAM [4] suffers from significant tracking errors and fails to reconstruct the map. In addition, our method also offers accurate instance-level segmentation outcomes.

B.4 Semantic Segmentation without Ground-truth Mask

As mentioned in the limitation section, our system leverages the ground-truth 2D semantic masks as the segmentation prior. To justify the robustness of our system without utilizing ground-truth masks, we carried out further experiments using the real-world ScanNet dataset [2]. For these experiments, we used the semantic masks predicted by Lang-SAM [1], without any fine-tuning. These masks were adapted to the simplified NYU40 categories, which involved merging similar categories like table and counter. As shown in Figure 6, segmentation outcomes using Lang-SAM [2] show comparable or even more precise segmen-

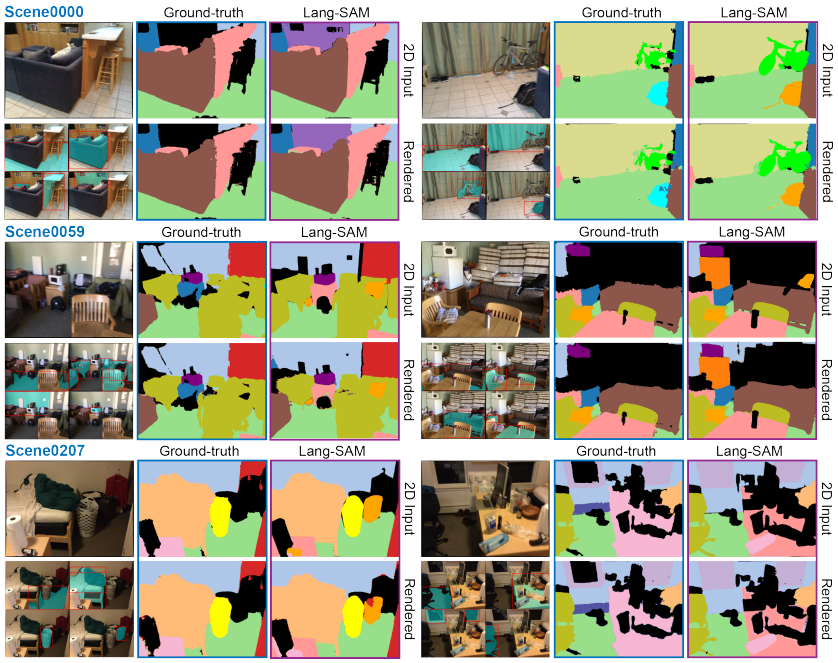

Fig. 6: Visualization of our segmentation results on the ScanNet dataset [2]. The results include inputting the ground-truth 2D labels and Lang-SAM [1] predicted masks prompted by the NYU40 categories. Additionally, masks of major objects in each scene are displayed separately beneath the RGB images.

tation than the ground-truth label captured from the mesh surface. Moreover, SGS-SLAM demonstrates robustness against noisy labels, which are inconsistent across frames, by employing multi-view geometry optimization. This approach remains effective as long as the noisy labels do not dominate. This robustness allows our system to successfully segment objects that are not recognized in the current view by leveraging information from other better frames of different view angles. An example of this can be seen with the wall at the right of scene0059_00 in Figure 6.

B.5 Scene Manipulation

In this section, we visualize scene manipulation results by grouping the Gaussians using the semantic mask. As shown in Fig. 7, for object removal, we can directly erase the Gaussians associated with the editing target, such as removing the table while preserving all the items on it. In addition, we can group objects by selecting their semantic masks and applying translation and rotation, such as moving and rotating both the table and the above objects to a different place.

It is worth noting that we can observe holes left in the place when removing or transitioning the objects. Such as the hole left on the ground when we removed the table. This is due to the explicit scene representation using 3D

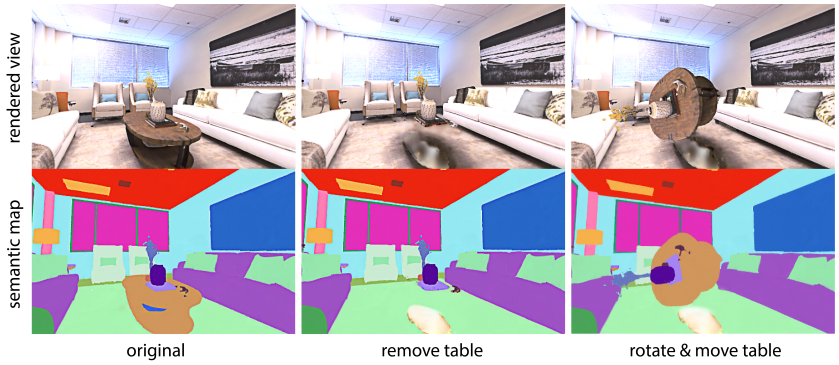

Fig. 7: The visualization of scene manipulation by grouping Gaussians via semantic labels. SGS-SLAM allows manipulation of either individual objects or a group of items, as illustrated by actions that include the removal of a table, as well as moving and rotating the table together with all objects on it.

Gaussians where the unobserved geometry in the multi-views from the trajectory are inevitably missing. This defect, stemming from the characteristics of the 3D Gaussian representation, poses a challenging problem. It is identified as an area for future research, with the potential solution through the use of 3D geometry priors [3] or scene inpainting [10] techniques.

References

1. Language segment-anything. `https://github.com/paulguerrero/lang-sam`
2. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5828–5839 (2017)
3. Freda, L.: Plvs: A slam system with points, lines, volumetric mapping, and 3d incremental segmentation. arXiv preprint arXiv:2309.10896 (2023)
4. Johari, M.M., Carta, C., Fleuret, F.: 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. pp. 17408–17419 (2023)
5. Li, K., Niemeyer, M., Navab, N., Tombari, F.: Dns slam: Dense neural semantic-informed slam. arXiv preprint arXiv:2312.00204 (2023)
6. 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)
7. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
8. Wang, H., Wang, J., Agapito, L.: Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13293–13302 (2023)
9. Yang, X., Li, H., Zhai, H., Ming, Y., Liu, Y., Zhang, G.: Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In: 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 499–507. IEEE (2022)
10. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023)
11. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the International Conference on Computer Vision (ICCV) (2023)
12. 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. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12786–12796 (2022)