

# GS<sup>3</sup>LAM: Gaussian Semantic Splatting SLAM

## - Supplementary Materials -

Anonymous Authors

### 1 MORE RELATED WORK

In order to help understand the characteristics of our GS<sup>3</sup>LAM, this section introduces some related work on semantic feature embedding. It is noteworthy that unlike GS<sup>3</sup>LAM, which simultaneously estimates camera poses and constructs semantic maps, these studies require accurate camera poses.

**Pose-Known Feature Embedded Field.** To enhance the perception and comprehension of 3D scenes, the embedding of high-dimensional features into these scenes has been extensively investigated within the domains of pose-aware NeRF [12] and 3DGS [5]. Early endeavors such as Semantic-NeRF [31] and Panoptic Lifting [21] aim to embed semantics into 3D space by optimizing a 3D feature radiance field to effectively reconstruct 2D features rendered from volumes. Building upon this foundation, Distilled Feature Fields [7] and LERF [6] extend this approach by incorporating high-dimensional feature vectors derived from models like DINOv2 [16] and CLIP [18] into the NeRF framework. In parallel with NeRF advancements, methodologies like LangSplat [17] and LEGaussians [20] endeavor to quantize high-dimensional CLIP features into 3D Gaussians, leveraging them for tasks pertaining to open-vocabulary scene understanding. Concurrently, techniques like Feature 3DGS [32] and Gaussian Grouping [30] embed semantic features into 3D Gaussians for application in tasks related to 3D group analysis. Although these methods can effectively lift 2D features into 3D field, the optimization for these features usually demands several hours of offline training, which presents challenges for SLAM systems requiring real-time pose and field optimization.

In GS<sup>3</sup>LAM framework, we embed the semantic features with our proposed Semantic Gaussian Field (SG-Field), in which high-dimensional semantic labels are encoded as low-dimensional implicit features, and subsequently decoded by a decoder into semantic labels, thereby facilitating an efficient conversion between 3D implicit features and 2D semantic labels.

### 2 METHOD DETAILS

This section provides the theoretical foundation of frame-to-model tracking in our GS<sup>3</sup>LAM, specifically focusing on the Jacobian of SG-Field with respect to camera poses. Furthermore, we investigate the optimization bias problem in Model-based (NeRF/3DGS) SLAM, a topic that has not yet been explored in the existing literature.

#### 2.1 Analytical Jacobian of Camera Pose

According to the ‘‘Semantic Splatting-Rendering and Decoding’’ pipeline outlined in Sec. 3.2.2, it is observed that the gradient of the camera pose  $\mathbf{T}_{CW}$  is associated with three intermediary variables: the 2D covariance matrix  $\Sigma^{2D}$ , the camera intrinsic parameter  $\mathbf{K}$ , and the center of the projected 2D Gaussian  $\mathbf{g}_i = (\mathbf{K}\mathbf{T}_{CW}\boldsymbol{\mu}_i)/d_i$ , where  $\boldsymbol{\mu}_i$  is the centroid (mean) of the  $i$ -th 3D Gaussian,  $d_i$  is the depth of the  $i$ -th 3D Gaussian centroid with respect to the camera coordinate system. By applying the chain rule of differentiation,

the analytical Jacobian of the semantic loss function  $\mathcal{L}_{sem}$  with respect to the camera pose  $\mathbf{T}_{CW}$  is derived as follows,

$$\begin{aligned} \frac{\partial \mathcal{L}_{sem}}{\partial \mathbf{T}_{CW}} &= \frac{\partial \mathcal{L}_{sem}}{\partial \hat{\mathbf{s}}_{pix}} \frac{\partial \hat{\mathbf{s}}_{pix}}{\partial \hat{\mathbf{f}}_{pix}} \frac{\partial \hat{\mathbf{f}}_{pix}}{\partial \mathbf{T}_{CW}} \\ &= \frac{\partial \mathcal{L}_{sem}}{\partial \hat{\mathbf{s}}_{pix}} \frac{\partial \hat{\mathbf{s}}_{pix}}{\partial \hat{\mathbf{f}}_{pix}} \left( \frac{\partial \hat{\mathbf{f}}_{pix}}{\partial \mathbf{f}_i} \frac{\partial \mathbf{f}_i}{\partial \mathbf{T}_{CW}} + \frac{\partial \hat{\mathbf{f}}_{pix}}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial \mathbf{T}_{CW}} \right). \end{aligned} \quad (1)$$

If the features dependent on viewpoint ( $\frac{\partial \hat{\mathbf{f}}_{pix}}{\partial \mathbf{f}_i} \frac{\partial \mathbf{f}_i}{\partial \mathbf{T}_{CW}}$ ) are disregarded, then,

$$\begin{aligned} \frac{\partial \mathcal{L}_{sem}}{\partial \mathbf{T}_{CW}} &= \frac{\partial \mathcal{L}_{sem}}{\partial \hat{\mathbf{s}}_{pix}} \frac{\partial \hat{\mathbf{s}}_{pix}}{\partial \hat{\mathbf{f}}_{pix}} \frac{\partial \hat{\mathbf{f}}_{pix}}{\partial \alpha_i} \left( \frac{\partial \alpha_i}{\partial \Sigma_i^{2D}} \frac{\partial \Sigma_i^{2D}}{\partial \mathbf{T}_{CW}} + \frac{\partial \alpha_i}{\partial \mathbf{g}_i} \frac{\partial \mathbf{g}_i}{\partial \mathbf{T}_{CW}} \right) \\ &= \frac{\partial \mathcal{L}_{sem}}{\partial \hat{\mathbf{s}}_{pix}} \frac{\partial \hat{\mathbf{s}}_{pix}}{\partial \hat{\mathbf{f}}_{pix}} \frac{\partial \hat{\mathbf{f}}_{pix}}{\partial \alpha_i} \left( \frac{\partial \alpha_i}{\partial \Sigma_i^{2D}} \frac{\partial (\mathbf{J}_i \mathbf{R}_{CW} \Sigma_i \mathbf{R}_{CW}^T \mathbf{J}_i^T)}{\partial \mathbf{T}_{CW}} + \frac{\partial \alpha_i}{\partial \mathbf{g}_i} \frac{\partial (\mathbf{K} \mathbf{T}_{CW} \boldsymbol{\mu}_i)}{\partial \mathbf{T}_{CW} d_i} \right). \end{aligned} \quad (2)$$

At this juncture, each component of the equation can be resolved through direct expansion. Similarly, the Jacobian of the color loss function  $\mathcal{L}_{color}$  with respect to the camera pose  $\mathbf{T}_{CW}$  is also derived as,

$$\begin{aligned} \frac{\partial \mathcal{L}_{color}}{\partial \mathbf{T}_{CW}} &= \frac{\partial \mathcal{L}_{color}}{\partial \hat{\mathbf{c}}_{pix}} \frac{\partial \hat{\mathbf{c}}_{pix}}{\partial \alpha_i} \left( \frac{\partial \alpha_i}{\partial \Sigma_i^{2D}} \frac{\partial \Sigma_i^{2D}}{\partial \mathbf{T}_{CW}} + \frac{\partial \alpha_i}{\partial \mathbf{g}_i} \frac{\partial \mathbf{g}_i}{\partial \mathbf{T}_{CW}} \right) \\ &= \frac{\partial \mathcal{L}_{color}}{\partial \hat{\mathbf{c}}_{pix}} \frac{\partial \hat{\mathbf{c}}_{pix}}{\partial \alpha_i} \left( \frac{\partial \alpha_i}{\partial \Sigma_i^{2D}} \frac{\partial (\mathbf{J}_i \mathbf{R}_{CW} \Sigma_i \mathbf{R}_{CW}^T \mathbf{J}_i^T)}{\partial \mathbf{T}_{CW}} + \frac{\partial \alpha_i}{\partial \mathbf{g}_i} \frac{\partial (\mathbf{K} \mathbf{T}_{CW} \boldsymbol{\mu}_i)}{\partial \mathbf{T}_{CW} d_i} \right). \end{aligned} \quad (3)$$

#### 2.2 Dive into Optimization Bias of Model-based SLAM

**Forgetting Phenomenon in Model-based SLAM.** In contrast to traditional SLAM methods based on points, surfels[13, 23, 26, 27], grids[14], or voxels[8, 10, 15] for scene representation, contemporary Model-based (NeRF/3DGS) SLAM systems [4, 11, 19, 25, 29, 36] typically employ implicit volumetric functions or dense Gaussian clouds to represent scenes. While these excel in rendering quality and novel view synthesis, they often exhibit a tendency to forget previously learned information in large scenes or long-term video sequences. This is mainly because Model-based SLAM systems rely on a single neural network with fixed capacity [19, 24, 35] or a global Gaussian model [4, 11, 28], which are susceptible to global changes during the incremental optimization. In NeRF-based SLAM systems, a common method to alleviate this problem is to use sparse ray sampling to train the network in the current observation frame, such as optimizing with randomly sampled 1024 pixels. However, in large-scale incremental mapping, this strategy necessitates complex resampling strategies to maintain lower memory efficiency as data increases. Conversely, in 3DGS-based SLAM systems, the sparse sampling strategy for explicit representation of Gaussian clouds leads to inefficient sampling of information across

3D space, resulting in uneven model updates and significant variations in rendering quality across different viewpoints. Despite the existence of forgetting phenomena in existing Model-based SLAM approaches, they still demonstrate relatively high mean rendering quality (PSNR), leading to the oversight of global map consistency, that is, the variance of the rendering quality (PSNR), in the current literature.

**Optimization Bias.** To evaluate the impact of optimization strategies on the global consistency of the map, we propose a scheme that incorporates camera trajectories, optimization iterations, and rendering quality. As illustrated in Fig. 1, the covisibility relationships between frames can be discerned from the camera trajectories: regions with dense camera trajectories indicate more co-view frames, while areas with sparse camera trajectories indicate fewer covisible frames. The size of each point’s radius indicates the number of optimizations undergone by the current frame during the model optimization process, with larger radii indicating a greater number of optimization iterations. Furthermore, the rendering quality of each frame can be inferred from the color of each point, with darker colors indicative of lower rendering quality (measured by PSNR). Additionally, the figure also provides the mean and variance of PSNR.

If an optimization strategy results in lower PSNR values in regions with high covisibility and frequent optimization iterations, it means that the strategy impedes the convergence of the model. Conversely, if lower PSNR values are observed in areas with low covisibility and fewer optimization iterations, it indicates under-optimization of the model due to the strategy. We refer to this phenomenon as **optimization bias**.

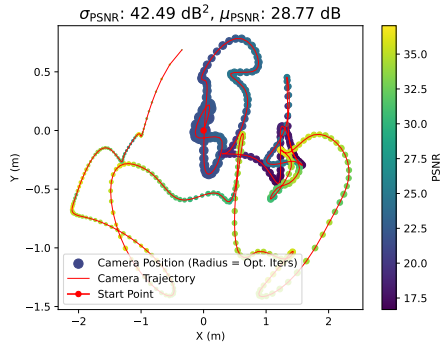


Figure 1: Illustration of optimization bias.

Based on the observations aforementioned, we aim for an optimization strategy that fosters a higher mean PSNR and lower PSNR variance in the model, thereby yielding a map with enhanced global consistency. Therefore, we propose the Random Sampling-based Keyframe Mapping (RSKM) strategy, which integrates random sampling techniques applied in NeRF-based SLAM into 3DGS-based SLAM. In comparison to the Local Covisibility Keyframe Mapping (LCKM) strategy employed in SplatAM [4], RSKM not only enhances rendering quality (resulting in a higher mean PSNR) but also augments the global consistency of the map (resulting in a lower PSNR variance). Further experimental results are presented in Fig. 2.

### 3 MORE EXPERIMENTS

#### 3.1 Further Implementation Details

**Learning rate setting.** During the tracking phase, the learning rate for the rotation quaternion of the pose was set to 0.0004, while for the translation vector of the pose, it was set to 0.002. In the mapping phase, the learning rates of semantic Gaussians are as follows: position-0.0001, color-0.0025, rotation matrix-0.001, opacity-0.05, scaling matrix-0.001, and semantic feature-0.0025. **Objective function weight setting.** In the tracking phase, color term weight  $\lambda_c^t$  was set to 0.5, depth term weight  $\lambda_d^t$  to 1.0, and semantic feature term weight  $\lambda_s^t$  to 0.001. During the mapping state, color term weight  $\lambda_c^m$  was set to 0.5, depth term weight  $\lambda_d^m$  to 1.0, semantic feature term weight  $\lambda_s^m$  to 0.01, big scale term weight  $\lambda_{big}^m$  to 0.01, and small scale term weight  $\lambda_{small}^m$  to 0.001. **Optimization iteration setting.** For Replica [22], the tracking process underwent 40 iterations, while the mapping phase was subjected to 60 iterations. Conversely, for ScanNet [2], the tracking phase involved 100 iterations, whereas the mapping phase underwent 30 iterations.

#### 3.2 Runtime Analysis

As depicted in Table 1, we present a comparative analysis of the runtime performance of GS<sup>3</sup>LAM against SOTA methods on the Replica “Office 0” [22]. Leveraging the efficient representation of SG-Field and the tile-based rasterization technique, GS<sup>3</sup>LAM demonstrates expedited mapping capabilities. Furthermore, its rendering speed surpasses that of 3DGS-based SplatAM [4] by a factor of 1.78 and outperforms NeRF-based Point-SLAM [19] by a significant margin of 36 times.

Table 1: Runtime analysis on Replica “Office 0”.

Method	Mapping /Iteration(ms)	Mapping /Frame(s)	Tracking /Iteration(ms)	Tracking /Frame(s)	Rendering (FPS)
NICE-SLAM[36]	89	<b>1.15</b>	<b>27</b>	<b>1.06</b>	2.64
Vox-Fusion[29]	98	1.47	64	1.92	1.63
Point-SLAM[19]	57	3.52	<b>27</b>	1.11	2.96
SplatAM[4]	83	4.94	70	2.82	59.91
GS <sup>3</sup> LAM (Ours)	<b>55</b>	4.29	89	3.01	<b>106.56</b>

Table 2: Rendering speed (FPS ↑) on Replica [22].

Method	R0	R1	R2	O0	O1	O2	O3	O4	Avg.
SplatAM [4]	71.30	65.90	52.75	59.91	57.77	82.18	63.43	84.49	67.22
GS <sup>3</sup> LAM (Ours)	121.21	93.46	75.55	97.32	95.33	135.69	101.27	153.09	<b>109.12</b>

#### 3.3 More Tracking Evaluations

As shown in Table 4, we present a comparative assessment of the tracking performance of GS<sup>3</sup>LAM against other state-of-the-art methodologies on the ScanNet dataset [2]. Due to the inherent inaccuracies in depth measurements in ScanNet, explicit 3DGS-based SLAM systems encounter challenges. Unlike implicit NeRF-based approaches, which leverage Signed Distance Function (SDF) Multi-Layer Perceptron (MLP) branches to overfit the effects of depth errors effectively, explicit 3DGS-based SLAM methods demonstrate slightly inferior tracking precision. A potential solution to this issue involves integrating MLPs to optimize Gaussian attributes, thereby

**Table 3: The comparative analysis of our GS<sup>3</sup>LAM on Replica [22] with concurrent semantic SLAM-related studies. Under competitive tracking accuracy, GS<sup>3</sup>LAM achieves state-of-the-art rendering quality and semantic reconstruction.**

Methods	Metrics	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4	Avg.
SNI-SLAM [34] CVPR 24	PSNR [dB] ↑	25.91	28.17	29.15	33.86	30.34	29.10	29.02	29.87	29.43
	SSIM ↑	0.885	0.910	0.938	0.965	0.927	0.950	0.950	0.952	0.935
	LPIPS ↓	0.307	0.292	0.245	0.182	0.225	0.238	0.192	0.198	0.235
	mIoU [%] ↑	88.42	87.43	86.16	87.63	78.63	86.49	74.01	80.22	83.62
	ATE RMSE [cm] ↓	0.50	0.55	0.45	0.35	0.41	0.33	0.62	0.50	0.46
SGS-SLAM [9] arXiv 24	PSNR [dB] ↑	32.50	34.25	35.10	38.54	39.20	32.90	32.05	32.75	34.66
	SSIM ↑	0.976	0.978	0.981	0.984	0.980	0.967	0.966	0.949	0.973
	LPIPS ↓	0.070	0.094	0.070	0.086	0.087	0.101	0.115	0.148	0.096
	mIoU [%] ↑	92.95	92.91	92.10	92.90	-	-	-	-	92.72
	ATE RMSE [cm] ↓	0.46	0.45	0.29	0.46	0.23	0.45	0.42	0.55	0.41
SemGauss-SLAM [33] arXiv 24	PSNR [dB] ↑	32.55	33.92	35.15	39.18	39.87	32.97	31.60	35.00	35.03
	SSIM ↑	0.979	0.979	0.987	0.989	0.990	0.979	0.972	0.978	0.982
	LPIPS ↓	0.055	0.054	0.045	0.048	0.050	0.069	0.078	0.093	0.062
	mIoU [%] ↑	92.81	94.10	94.72	95.23	90.11	94.93	92.93	94.82	93.71
	ATE RMSE [cm] ↓	0.26	0.42	0.27	0.34	0.17	0.32	0.36	0.49	0.33
NEDS-SLAM [3] arXiv 24	PSNR [dB] ↑	35.23	34.86	35.16	37.53	39.71	32.68	31.07	31.82	34.76
	SSIM ↑	0.979	0.862	0.983	0.981	0.979	0.973	0.968	0.973	0.962
	LPIPS ↓	0.082	0.075	0.071	0.091	0.087	0.079	0.103	0.113	0.088
	mIoU [%] ↑	90.73	91.20	-	90.42	-	-	-	-	90.78
	ATE RMSE [cm] ↓	0.37	0.40	0.33	0.35	0.28	0.30	0.32	0.47	0.35
GS <sup>3</sup> LAM (Ours)	PSNR [dB] ↑	33.67	35.80	35.96	40.28	41.21	34.30	34.27	34.59	36.26
	SSIM ↑	0.986	0.989	0.990	0.993	0.994	0.988	0.990	0.983	0.989
	LPIPS ↓	0.051	0.039	0.046	0.040	0.030	0.065	0.061	0.081	0.052
	mIoU [%] ↑	96.83	96.68	96.40	96.61	97.35	96.83	96.10	95.73	96.57
	ATE RMSE [cm] ↓	0.27	0.25	0.28	0.67	0.21	0.33	0.30	0.65	0.37

**Table 4: Tracking performance on ScanNet [2] (ATE RMSE ↓ [cm]).**

Method	0000	0059	0106	0169	0181	0207	Avg.
NICE-SLAM[36]	12.00	14.00	7.90	10.90	13.40	6.20	10.70
Vox-Fusion[29]	68.84	24.18	8.41	27.28	23.30	9.41	26.90
Point-SLAM[19]	10.24	7.81	8.65	22.16	14.77	9.54	12.19
SplaTAM[4]	12.83	10.10	17.72	12.08	11.10	7.46	11.88
GS <sup>3</sup> LAM (Ours)	11.34	10.78	17.00	11.35	10.57	6.39	11.24

enhancing the robustness of 3DGS-based SLAM in real-world scenarios [1, 37].

### 3.4 More Rendering Evaluations

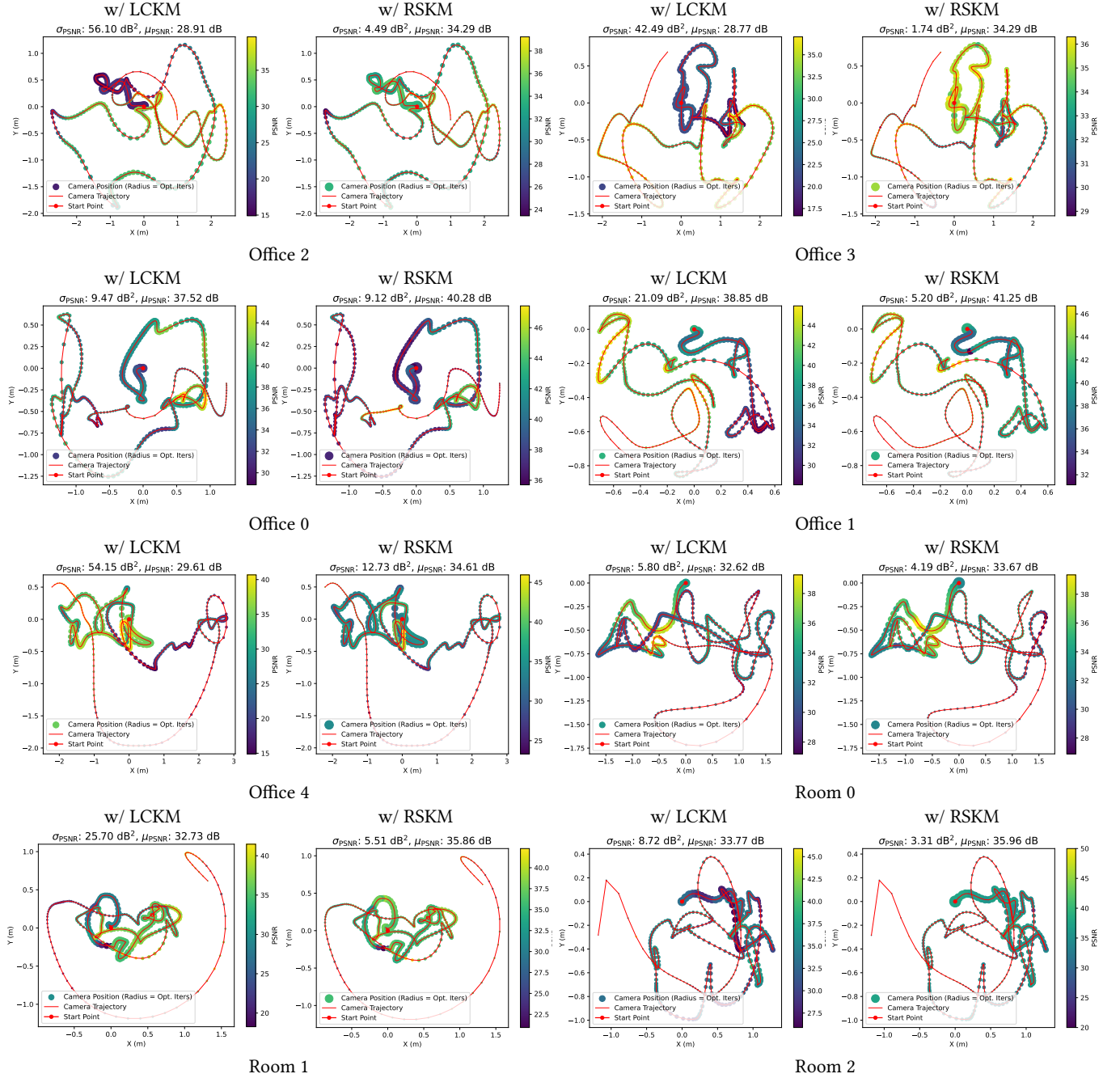
In Fig. 5 and Fig. 6, we present additional comparative analyses of rendering quality between GS<sup>3</sup>LAM and state-of-the-art methods on the Replica [22] and ScanNet [2] datasets, respectively.

### 3.5 Semantic Reconstruction Results

In Fig. 3 and Fig. 4, we present the semantic Gaussian fields reconstructed by our GS<sup>3</sup>LAM, along with the decoupled geometric, appearance, and semantic maps derived therefrom, respectively. Furthermore, in Fig. 7, we present the visual results of semantic segmentation achieved by GS<sup>3</sup>LAM. From these illustrations, it is discernible that our approach yields more precise segmentation along object boundaries, particularly evident on the ScanNet dataset [2] characterized by imprecise semantic labels.

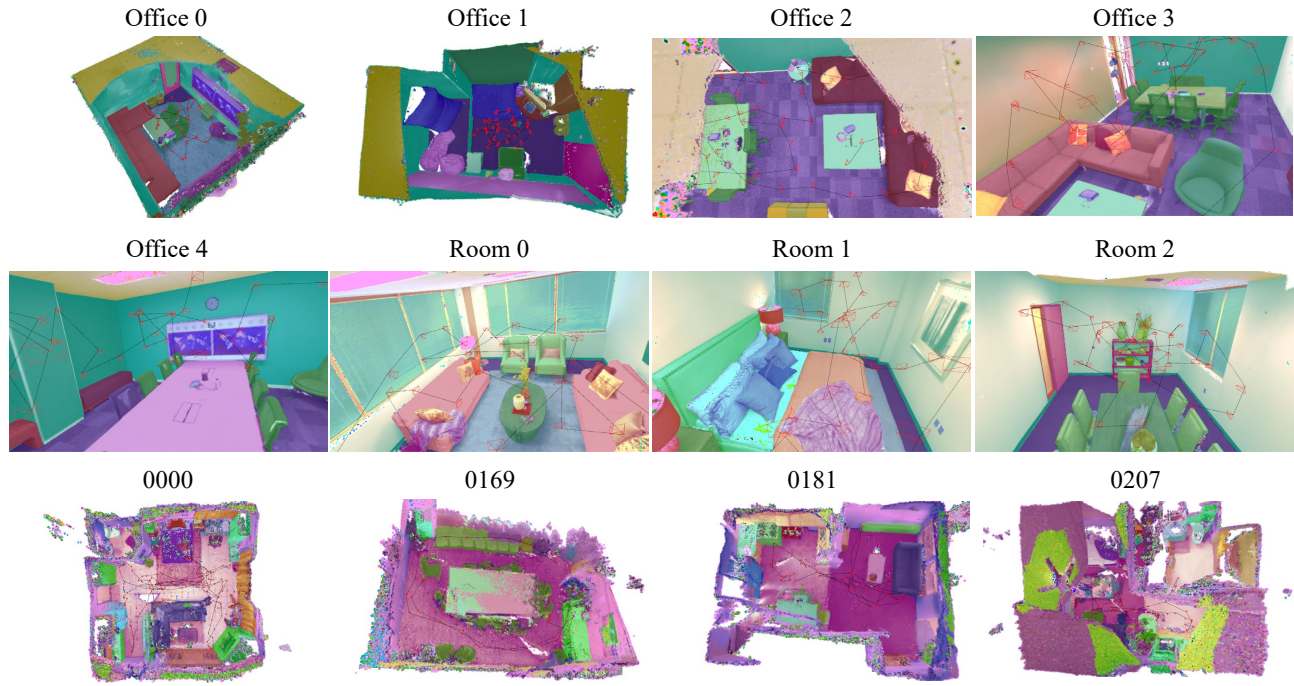
### 3.6 Comparison with Contemporary Studies

As of the submission deadline, several concurrent, non-open-source semantic SLAM endeavors have been identified on arXiv. Our comparative analysis with these studies is presented in Table 3, indicating that our GS<sup>3</sup>LAM achieves state-of-the-art rendering quality and semantic reconstruction while maintaining competitive tracking accuracy.

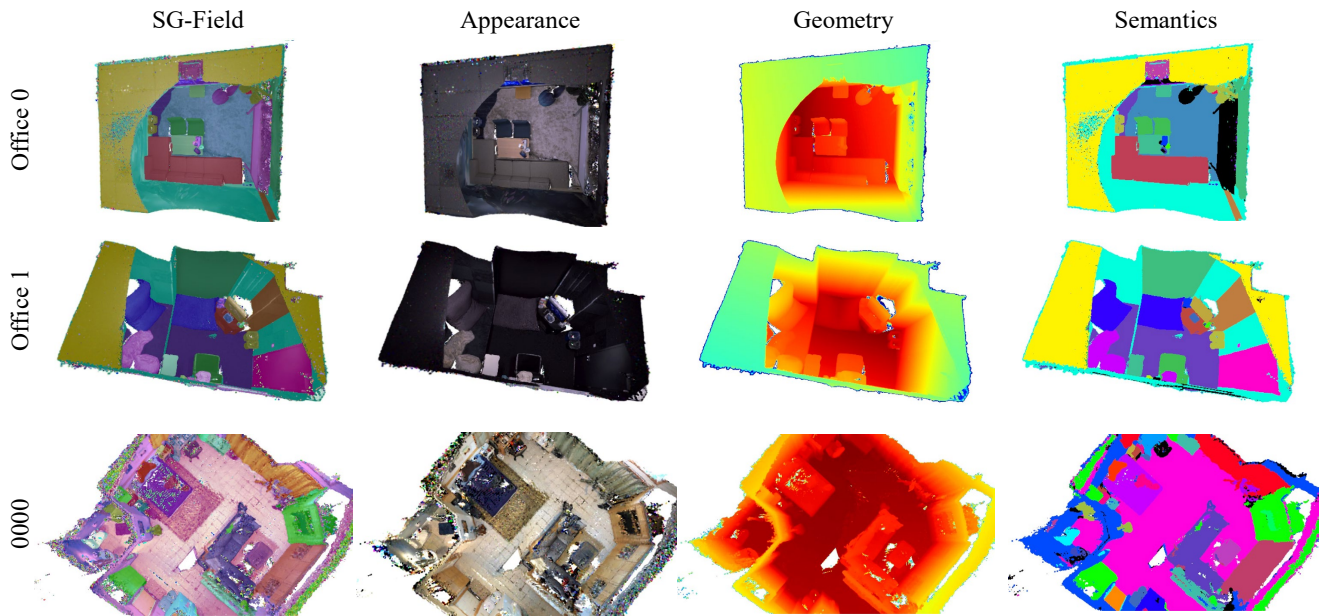


**Figure 2: Optimization bias on Replica [22].** Our proposed RSKM strategy not only improves rendering quality (higher mean PSNR  $\mu_{PSNR}$ ) but also enhances the global consistency of the map (lower PSNR variance  $\sigma_{PSNR}$ ). The LCKM strategy employed in SplatAM [4] exhibits lower PSNR in regions with high covisibility and frequent optimization iterations, thereby hindering model convergence in these areas. Conversely, in regions with fewer covisible frames, the reduced optimization iterations lead to under-optimized model, resulting in decreased PSNR.





**Figure 3: The visualization of the semantic Gaussian fields constructed by our GS<sup>3</sup>LAM on the Replica [22] and ScanNet [2] datasets. GS<sup>3</sup>LAM demonstrates robust tracking capabilities and achieves real-time high-quality rendering at 109 FPS, along with precise 3D semantic reconstruction.**



**Figure 4: Semantic Gaussian field decoupling by our GS<sup>3</sup>LAM. GS<sup>3</sup>LAM is capable of real-time construction of 3D semantic maps that exhibit geometric, appearance, and semantic consistency, thereby enabling potential downstream real-time tasks.**



Figure 5: More rendering results on Replica [22].



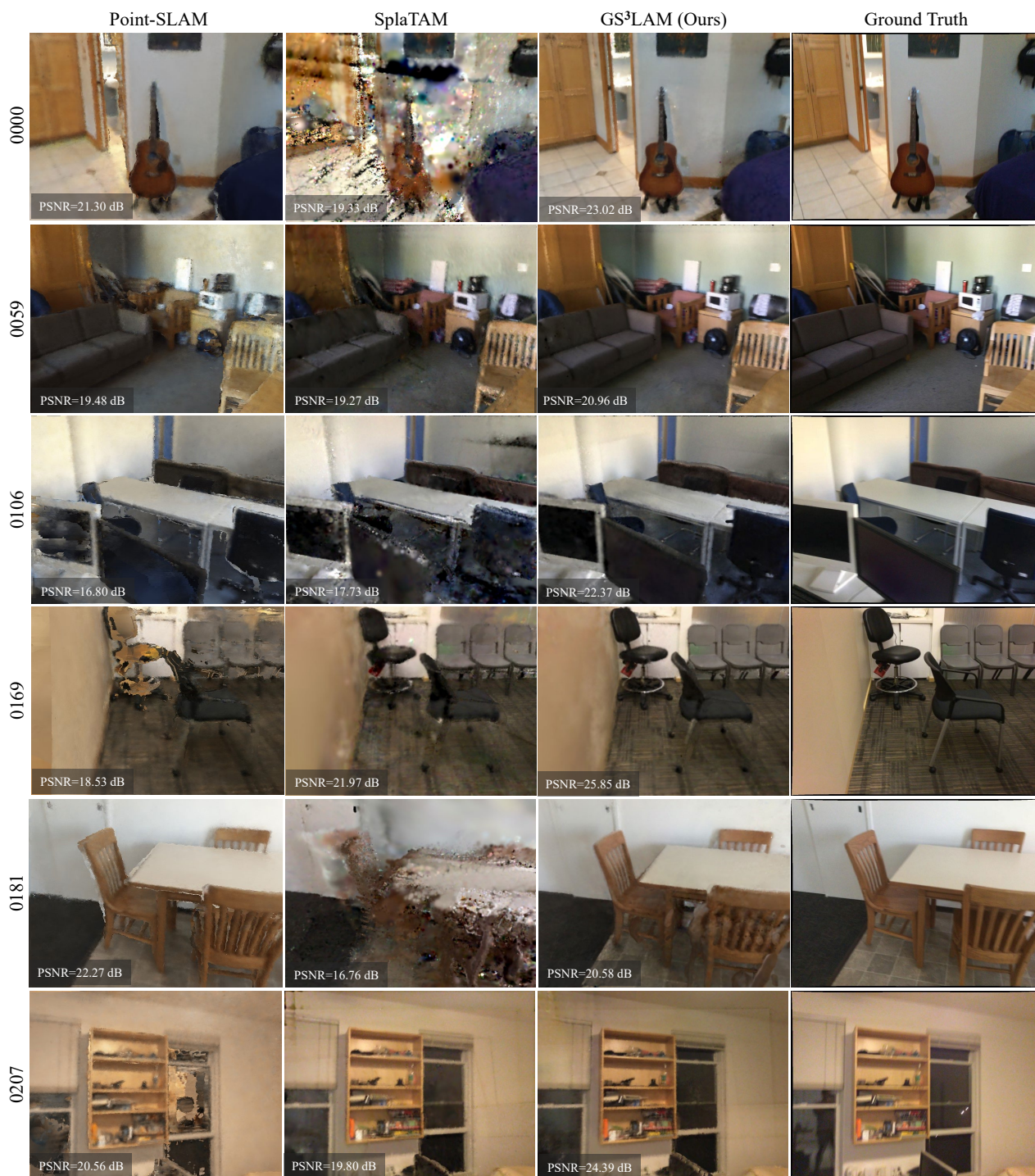
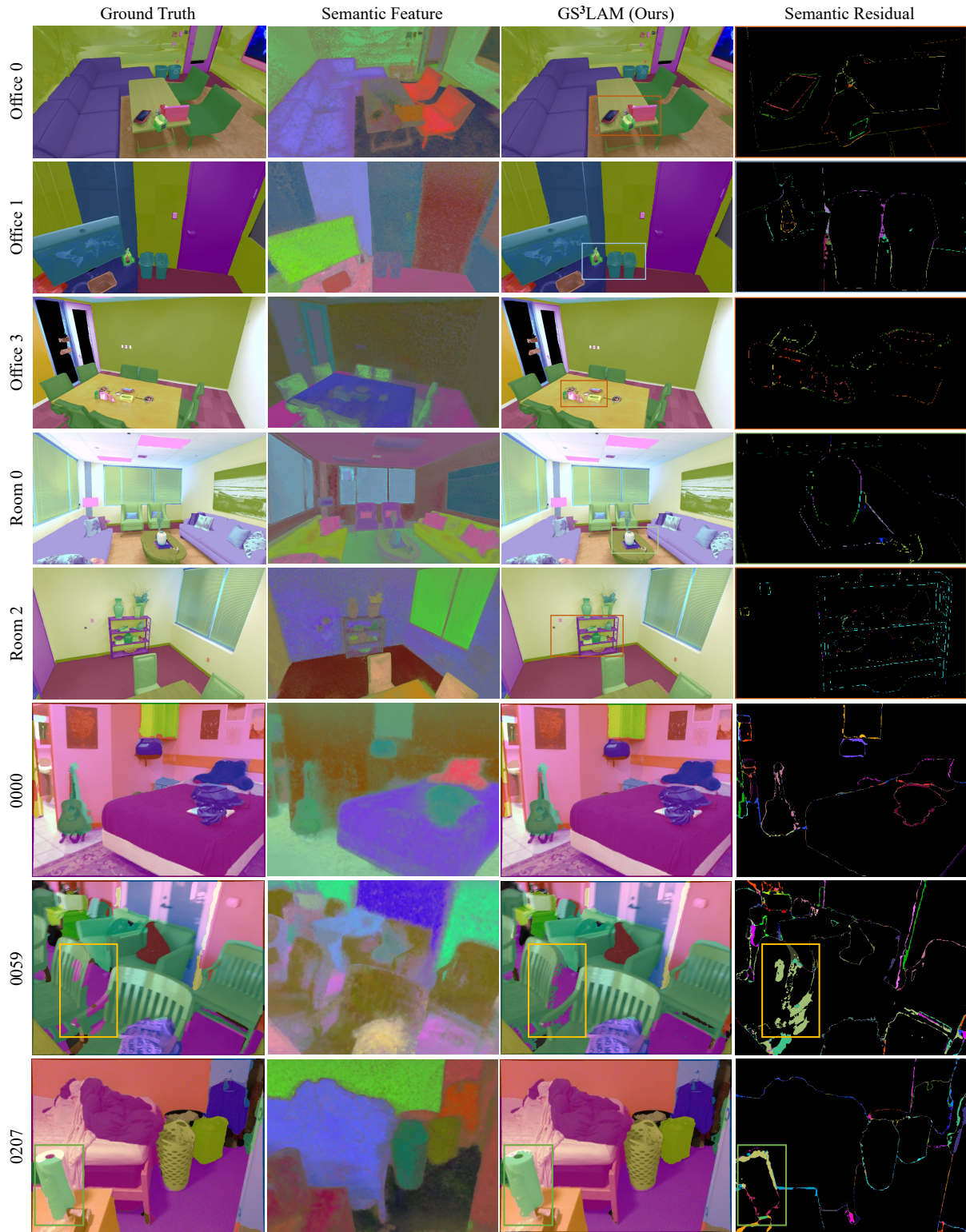


Figure 6: More rendering results on ScanNet [2].





**Figure 7: Semantic rendering on Replica [22] and ScanNet [2]. It is noteworthy that on the ScanNet dataset, which contains real data with inaccurately annotated semantic labels, our GS³LAM exhibits the capability to achieve more precise segmentation results at object boundaries.**



## REFERENCES

- [1] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. 2023. pixelsplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. *arXiv:2312.12337*.
- [2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA, 2432–2443.
- [3] Yiming Ji, Yang Liu, Guanghu Xie, Boyu Ma, and Zongwu Xie. 2024. NEDS-SLAM: A Novel Neural Explicit Dense Semantic SLAM Framework using 3D Gaussian Splatting. *arXiv:2403.11679*.
- [4] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scher, Deva Ramanan, and Jonathon Luiten. 2024. SplatAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM. *arXiv:2312.02126*.
- [5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023), 1–14.
- [6] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. LERF: Language Embedded Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Paris, France, 19729–19739.
- [7] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. 2022. Decomposing NeRF for Editing via Feature Field Distillation. In *Advances in Neural Information Processing Systems*, Vol. 35. 23311–23330.
- [8] Olaf Kähler, Victor Prisacariu, Julien Valentin, and David Murray. 2016. Hierarchical Voxel Block Hashing for Efficient Integration of Depth Images. *IEEE Robotics and Automation Letters* 1, 1 (2016), 192–197.
- [9] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. 2024. SGS-SLAM: Semantic Gaussian Splatting For Neural Dense SLAM. *arXiv:2402.03246*.
- [10] Robert Maier, Raphael Schaller, and Daniel Cremers. 2017. Efficient Online Surface Correction for Real-time Large-Scale 3D Reconstruction. *arXiv:1709.03763*.
- [11] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. 2023. Gaussian Splatting SLAM. *arXiv:2312.06741*.
- [12] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision*. Glasgow, United Kingdom, 405–421.
- [13] Raúl Mur-Artal and Juan D. Tardós. 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics* 33, 5 (2017), 1255–1262.
- [14] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality*. Basel, Switzerland, 127–136.
- [15] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. 2013. Real-time 3D Reconstruction at Scale Using Voxel Hashing. *ACM Transactions on Graphics* 32, 6 (2013), 1–11.
- [16] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. *arXiv:2304.07193*.
- [17] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. 2023. LangSplat: 3D Language Gaussian Splatting. *arXiv:2312.16084*.
- [18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139. 8748–8763.
- [19] Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. 2023. Point-SLAM: Dense Neural Point Cloud-based SLAM. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Los Alamitos, CA, USA, 18387–18398.
- [20] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. 2023. Language Embedded 3D Gaussians for Open-Vocabulary Scene Understanding. *arXiv:2311.18482*.
- [21] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kotschieder. 2023. Panoptic Lifting for 3D Scene Understanding with Neural Fields. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Vancouver, BC, Canada, 9043–9052.
- [22] 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. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. *arXiv:1906.05797*.
- [23] Jörg Stückler and Sven Behnke. 2014. Multi-resolution Surfel Maps for Efficient Dense 3D Modeling and Tracking. *Journal of Visual Communication and Image Representation* 25, 1 (2014), 137–147.
- [24] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. 2021. iMAP: Implicit Mapping and Positioning in Real-Time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Montreal, QC, Canada, 6209–6218.
- [25] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. 2023. Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Vancouver, BC, Canada, 13293–13302.
- [26] Kaixuan Wang, Fei Gao, and Shaojie Shen. 2019. Real-time Scalable Dense Surfel Mapping. In *Proceedings of IEEE International Conference on Robotics and Automation*. Montreal, QC, Canada, 6919–6925.
- [27] Thomas Whelan, Stefan Leutenegger, Rafael F. Salas-Moreno, Ben Glocker, and Andrew J. Davison. 2015. ElasticFusion: Dense SLAM Without A Pose Graph. In *Proceedings of Robotics: Science and Systems*.
- [28] Chi Yan, Delin Qu, Dong Wang, Dan Xu, Zhigang Wang, Bin Zhao, and Xuelong Li. 2024. GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting. *arXiv:2311.11700*.
- [29] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. 2022. Vox-Fusion: Dense Tracking and Mapping with Voxel-based Neural Implicit Representation. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality*. Los Alamitos, CA, USA, 499–507.
- [30] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. 2023. Gaussian Grouping: Segment and Edit Anything in 3D Scenes. *arXiv:2312.00732*.
- [31] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. 2021. In-Place Scene Labelling and Understanding with Implicit Scene Representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Montreal, QC, Canada, 15818–15827.
- [32] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suyu You, Zhangyang Wang, and Achuta Kadambi. 2023. Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields. *arXiv:2312.03203*.
- [33] Siting Zhu, Renjie Qin, Guangming Wang, Jiuming Liu, and Hesheng Wang. 2024. SemGauss-SLAM: Dense Semantic Gaussian Splatting SLAM. *arXiv:2403.07494*.
- [34] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. 2024. SNI-SLAM: Semantic Neural Implicit SLAM. *arXiv:2311.11016*.
- [35] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. 2024. NICER-SLAM: Neural Implicit Scene Encoding for RGB SLAM. In *Proceedings of the International Conference on 3D Vision*. Davos, Switzerland.
- [36] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. 2022. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. New Orleans, LA, USA, 12776–12786.
- [37] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. 2023. Triplane Meets Gaussian Splatting: Fast and Generalizable Single-View 3D Reconstruction with Transformers. *arXiv:2312.09147*.