SemGauss-SLAM: Dense Semantic Gaussian Splatting SLAM

Siting Zhu[1], Renjie Qin[1], Guangming Wang[2], Jiuming Liu[1],Hesheng Wang[1]

[1] Shanghai Jiao Tong University
[2] University of Cambridge

Abstract. We propose SemGauss-SLAM, the first semantic SLAM system utilizing 3D Gaussian representation, that enables accurate 3D semantic mapping, robust camera tracking, and high-quality rendering in real-time. In this system, we incorporate semantic feature embedding into 3D Gaussian representation, which effectively encodes semantic information within the spatial layout of the environment for precise semantic scene representation. Furthermore, we propose feature-level loss for updating 3D Gaussian representation, enabling higher-level guidance for 3D Gaussian optimization. In addition, to reduce cumulative drift and improve reconstruction accuracy, we introduce semantic-informed bundle adjustment leveraging semantic associations for joint optimization of 3D Gaussian representation and camera poses, leading to more robust tracking and consistent mapping. Our SemGauss-SLAM method demonstrates superior performance over existing dense semantic SLAM methods in terms of mapping and tracking accuracy on Replica and ScanNet datasets, while also showing excellent capabilities in novel-view semantic synthesis and 3D semantic mapping. The source code will be released upon acceptance.

Keywords: Dense Semantic SLAM · Semantic-informed Bundle Adjustment · 3D Gaussian Semantic Representation · 3D Reconstruction

1 Introduction

Dense semantic Simultaneous Localization and Mapping (SLAM) is a fundamental challenge for robotic systems [15,18] and autonomous driving [1,10]. It integrates semantic understanding of the environment into dense map reconstruction and performs pose estimation simultaneously. Traditional semantic SLAM has limitations including its inability to predict unknown areas and the demand for considerable map storage [11]. Subsequent semantic SLAM based on Neural Radiance Fields (NeRF) [16] methods [9,32] have addressed these drawbacks, but suffers from inefficient per-pixel raycasting rendering.

Recently, a novel radiance field based on 3D Gaussian [8] has demonstrated remarkable capability in scene representation, enabling high-quality and efficient rendering via splatting. Following the advantages of 3D Gaussian representation, 3D Gaussian SLAM methods [7,13,27,31] have been developed to achieve photo-realistic mapping. However, existing 3D Gaussian SLAM systems focus on visual

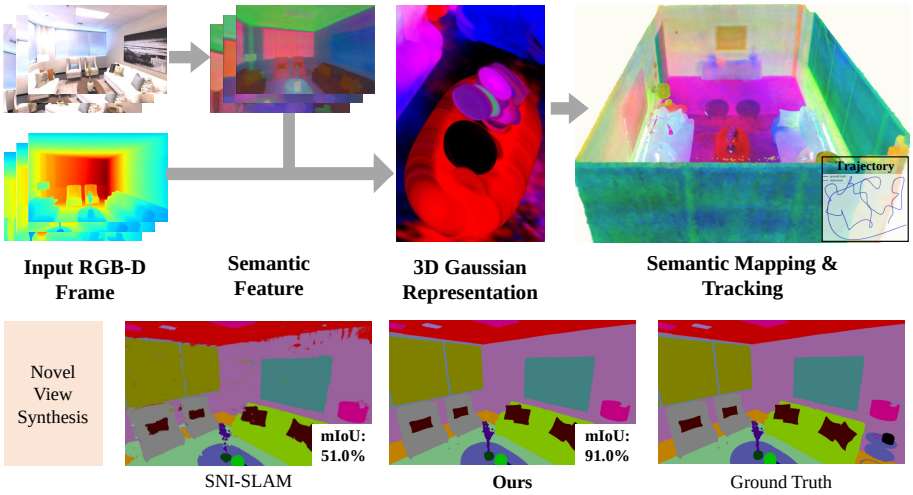


Fig. 1: Our SemGauss-SLAM incorporates semantic feature embedding into 3D Gaussian representation to perform dense semantic SLAM. This modeling strategy not only achieves accurate semantic mapping, but also enables high-precision semantic novel view synthesis compared with other radiance field-based semantic SLAM. We visualize 3D Gaussian blobs with semantic embedding, showing the spatial layout of semantic Gaussian representation. Moreover, semantic mapping is visualized using semantic feature embedding, showing 3D semantic modeling of the scene.

mapping to obtain RGB maps, where color information alone is insufficient for downstream tasks such as navigation. Meanwhile, current semantic SLAM methods based on NeRF are prone to cumulative drift, leading to degraded SLAM accuracy. Therefore, developing a low-drift semantic SLAM system based on radiance field is essential and challenging.

For radiance field-based semantic SLAM, there are two challenges: 1) Existing semantic SLAM methods struggle to achieve real-time 3D semantic mapping from 2D semantic information without predefined bounds, which is a fundamental requirement for semantic SLAM system. 2) As the progression of tracking, pose estimation is prone to cumulative drift, leading to a decrease in SLAM accuracy.

For the first challenge, SNI-SLAM [32] utilizes feature collaboration to achieve semantic mapping but requires specific scene bounds. Moreover, solely utilizing rendering results for semantic representation optimization without initial value is computationally inefficient. In this paper, we leverage the explicit structure of 3D Gaussian representation and propagate extracted 2D semantic features directly to 3D Gaussian as the initial value for achieving unbounded mapping and efficient 3D semantic scene optimization. This design enables faster transfer from 2D semantic information to 3D semantic map. In addition, we introduce 2D feature-level loss to guide scene representation optimization at a higher-level, thereby accelerating the convergence of 3D Gaussian optimization.

For the second challenge, NICE-SLAM [33] performs local bundle adjustment (BA) for joint optimization of pose and scene representation to reduce accumulated drift. However, it only utilizes appearance and geometry constraints, lacking adequate semantic constraints for semantic SLAM optimization. To address this challenge, we leverage semantic associations among co-visible frames and propose semantic-informed bundle adjustment for joint optimization of camera poses and 3D Gaussian representation. This design exploits the consistency of multi-view semantics for establishing constraints, which enables the reduction of cumulative drift in tracking and enhanced semantic mapping precision.

Overall, we provide the following contributions:

- We present SemGauss-SLAM, the first 3D Gaussian semantic SLAM system, which can achieve accurate semantic mapping and photo-realistic reconstruction. We incorporate semantic feature embedding into 3D Gaussian for precisely constructing semantic maps. Moreover, feature-level loss is introduced to provide higher-level guidance for 3D Gaussian optimization.
- We perform semantic-informed bundle adjustment by leveraging multi-view semantic constraints for joint optimization of camera poses and 3D Gaussian representation, achieving low-drift tracking and accurate semantic mapping.
- We conduct extensive evaluations on two challenging datasets, Replica [21] and ScanNet [2], to demonstrate our method achieves state-of-the-art performance compared with existing radiance field-based SLAM in mapping, tracking, semantic segmentation and novel-view synthesis.

2 Related Work

Semantic SLAM can be divided into two main categories based on the form of scene representation: traditional semantic SLAM and neural implicit semantic SLAM. Traditional semantic SLAM employs explicit 3D representation such as surfels [15], mesh [5, 18, 24] and Truncated Signed Distance Fields (TSDF) [3, 14, 20]. Neural implicit semantic SLAM [4, 9, 32] utilizes implicit representation, such as feature grid [9] and feature plane [32].

Traditional Semantic SLAM. SemanticFusion [15] uses surfel representation and employs Conditional Random Field (CRF) for updating class probability distribution incrementally. Fusion++ [14] performs object-level SLAM where each object is reconstructed within its own TSDF volume and segmented based on estimated foreground probability. Kimera [18] utilizes visual-inertial odometry for pose estimation and generates dense semantic mesh maps. However, these explicit scene modeling methods not only require high storage space but also fail to achieve high-fidelity and complete reconstruction.

Neural Implicit SLAM. iMAP [23] first achieves real-time mapping and tracking utilizing a single MLP network for scene representation. To overcome oversmoothed scene reconstruction and improve scalability, NICE-SLAM [33] adopts hierarchical feature grid representation. Following this, several works [6, 19, 25, 29] introduce more efficient scene representation, such as hash-based feature grid and feature plane, to achieve more accurate SLAM performance.

For neural implicit semantic SLAM, DNS SLAM [9] utilizes 2D semantic priors and integrates multi-view geometry constraints for semantic reconstruction. SNI-SLAM [32] introduces feature collaboration and one-way correlation decoder for improved scene representation in semantic mapping. However, these methods require specific scene bounds for mapping and suffer from cumulative drift in pose estimation. In this paper, we leverage the explicit structure of 3D Gaussian for unbounded mapping, and perform semantic-informed bundle adjustment utilizing multi-frame semantic constraints for conducting low-drift, high-quality dense semantic SLAM.

3D Gaussian Splatting SLAM. 3D Gaussian [8] emerges as a promising 3D scene representation using a set of 3D Gaussians with position, anisotropic covariance, opacity, and color. This representation is capable of quick differential rendering through splatting and has a wide range of applications in dynamic scene modeling [12, 26, 28] and scene editing [30, 34].

Our main focus is on 3D Gaussian SLAM [7, 13, 27, 31]. These works emerge concurrently and all perform dense visual SLAM by leveraging scene geometry and appearance modeling capabilities of 3D Gaussian representation. SplaTAM [7] introduces silhouette-guided optimization to facilitate structured map expansion for dense mapping of visual SLAM. Gaussian Splatting SLAM [13] performs novel Gaussian insertion and pruning for monocular SLAM. However, the capability of 3D Gaussian representation extends well beyond appearance and geometry modeling of scene, as it can be augmented to perform semantic scene understanding. In this paper, we incorporate semantic feature embedding into 3D Gaussian for 3D semantic scene modeling to achieve high-precision dense semantic SLAM.

3 Method

SemGauss-SLAM is the first RGB-D dense semantic SLAM based on 3D Gaussian splatting. The overview of SemGauss-SLAM is shown in Fig. 2. Sec. 3.1 introduces semantic Gaussian representation for dense semantic mapping, as well as the process of 3D Gaussian-based semantic SLAM. Sec. 3.2 introduces the loss functions in the mapping and tracking process. Sec. 3.3 presents semantic-informed bundle adjustment to reduce accumulated drift in tracking and improve semantic mapping accuracy.

3.1 3D Gaussian Semantic Mapping and Tracking

Semantic Gaussian Representation. We utilize a set of Gaussians with specific properties for scene representation. To simplify scene representation for the SLAM process, we use isotropy Gaussians. Considering semantic representation, which has never been explored in other Gaussian SLAM, we introduce a new parameter, semantic feature embedding, to each Gaussian for semantic representation. To achieve real-time semantic mapping, it is crucial for 3D Gaussian representation, augmented by semantic feature embedding, to converge

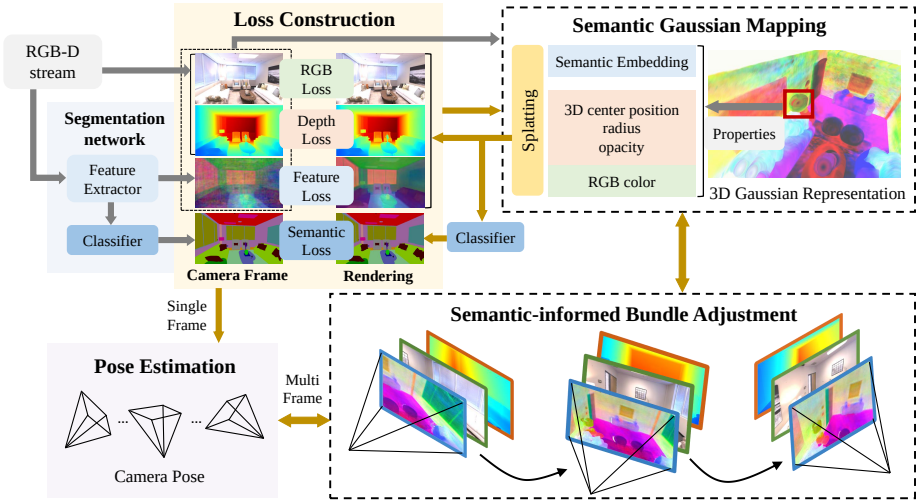
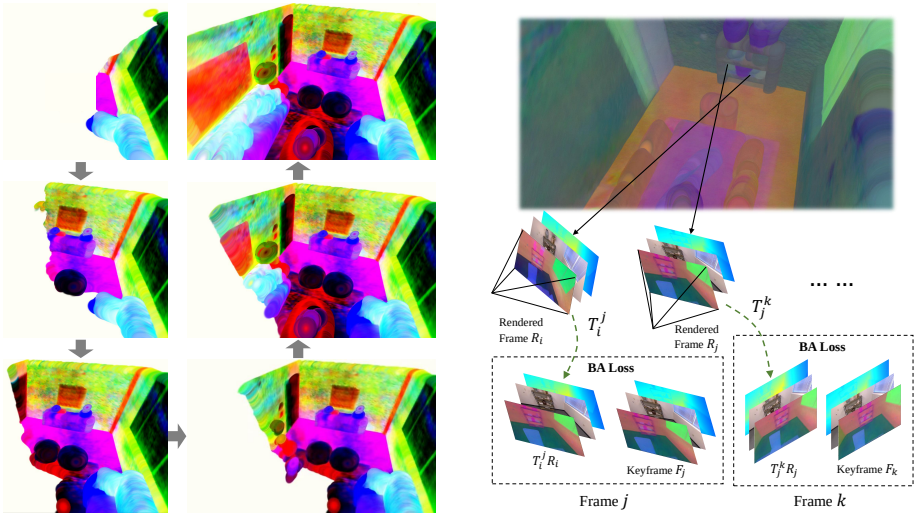


Fig. 2: An overview of SemGauss-SLAM. Our method takes an RGB-D stream as input. RGB images are fed into feature extractor to obtain semantic features. These features are then categorized by pretrained classifier to attain semantic labels. Then, semantic features, semantic labels, along with the input RGB and depth data serve as supervision signals. In the meantime, semantic features and input RGB-D data propagate to 3D Gaussian blobs as initial properties of Gaussian representation. Rendered semantic feature, RGB, and depth are obtained from 3D Gaussian splatting, while rendered semantic label is attained by classifying rendered feature. Supervision and rendered information are utilized for loss construction to optimize camera poses and 3D Gaussian representation. During the SLAM process, we utilize semantic-informed bundle adjustment based on multi-frame constraints for joint optimization of poses and 3D Gaussian representation.

rapidly during the mapping optimization process. Therefore, instead of initializing these feature embeddings randomly, we propagate 2D semantic features extracted from images to 3D Gaussian as the initial values to achieve faster convergence of Gaussian semantic feature optimization. Moreover, integrating feature embedding into 3D Gaussian makes Gaussian semantic representation compact and efficient for capturing the spatial semantic information of the environment. Overall, each Gaussian includes 3D center position μ, radius r, color $c = (r, g, b)$, opacity α, 16-channel semantic feature embedding e, and is defined as standard Gaussian equation multiplied by opacity α:

$$g(x) = \alpha \exp\left(-\frac{\|x-\mu\|^2}{2r^2}\right). \tag{1}$$

3D Gaussian Rendering. Following Gaussian splatting [8], we project 3D Gaussians to 2D splats for high-fidelity differentiable rendering, allowing for explicit gradient flow for Gaussian scene optimization and pose estimation. Specifically, for splatting of semantic feature embedding, we first sort all Gaussians from



(a) 3D Gaussian Semantic Mapping **(b)** Semantic-informed Bundle Adjustment

Fig. 3: (a) 3D Gaussian semantic mapping process that utilizes 3D Gaussian blobs with semantic feature embedding for visualization. (b) Illustration of semantic-informed bundle adjustment.

front-to-back and then blend N ordered points projecting to pixel $p = (u, v)$, obtaining 2D semantic feature maps $E(p)$:

$$E(p) = \sum_{i \in N} e_i g_i(p) \prod_{j=1}^{i-1} (1 - g_j(p)), \tag{2}$$

where $g_i(p)$ is computed as shown in Eq.(1) with μ and r representing the rendered 2D Gaussians in pixel plane:

$$\mu_{\text{pix}} = K_c \frac{T_k \mu}{d}, \; r_{\text{pix}} = \frac{fr}{d}, \; \text{where } d = (T_k \mu)_z. \tag{3}$$

K_c is calibrated camera intrinsic, T_k is estimated camera pose at frame k, f is known focal length, d is the depth of the i-th Gaussian in camera coordinates.

For RGB and depth rendering, we follow the similar approach in Eq.(2):

$$C(p) = \sum_{i \in N} c_i g_i(p) \prod_{j=1}^{i-1} (1 - g_j(p)), \quad D(p) = \sum_{i \in N} d_i g_i(p) \prod_{j=1}^{i-1} (1 - g_j(p)), \tag{4}$$

where $C(p)$ and $D(p)$ represents splatted 2D color and depth images respectively. Moreover, given camera pose, visibility information of Gaussian is required for mapping and tracking process, such as adding new Gaussian and loss construction. Therefore, following [7], we render a silhouette image to determine

visibility:

$$Sil(p) = \sum_{i \in N} g_i(p) \prod_{j=1}^{i-1} (1 - g_j(p)). \tag{5}$$

Tracking Process. During tracking, we keep 3D Gaussian parameters fixed and only optimize camera pose T of current frame. Since the proximity between adjacent frames is relatively small, we assume a constant velocity model to obtain an initial pose estimation for new frame. The camera pose is then iteratively refined by optimization through loss construction between differentiable rendering from 3D Gaussian and camera observation within the visible silhouette.

Mapping Process. Our system performs RGB mapping and semantic mapping simultaneously. Mapping process begins with the initialization of the scene representation, which is achieved by inverse transforming all pixels of the first frame to 3D coordinates and obtaining the initial 3D Gaussian representation. Then, when the overlap of coming frame with existing map rendering is less than half, we add a new Gaussian for incremental mapping. Figure 3a demonstrates 3D Gaussian semantic mapping process.

3.2 Loss Functions

For optimization of semantic scene representation, we utilize cross-entropy loss for constructing semantic loss $\mathcal{L}_s$. Furthermore, instead of using semantic loss solely, we introduce a feature-level loss $\mathcal{L}_f$ for higher-level semantic optimization guidance. Feature-level loss is obtained by constructing L1 loss between extracted features generated by Dinov2 [17]-based feature extractor and splatted features obtained from 3D Gaussian representation. Compared with semantic loss, feature loss provides direct guidance on intermediate features, leading to more robust and accurate semantic scene understanding.

We employ RGB loss $\mathcal{L}_c$ and depth loss $\mathcal{L}_d$ for optimization of scene color and geometry representation. $\mathcal{L}_c$ and $\mathcal{L}_d$ are both L1 loss constructed by comparing the RGB and depth splats with the input RGB-D frame. These loss functions are then utilized for mapping and pose estimation.

In mapping process, we construct loss over all rendered pixels for 3D Gaussian scene optimization. Moreover, we add SSIM term to RGB loss following [8]. The complete loss function for mapping is a weighted sum of the above losses:

$$\mathcal{L}_{\text{mapping}} = \sum_{p \in P_M} (\lambda_{f_m} \mathcal{L}_f(p) + \lambda_{s_m} \mathcal{L}_s(p) + \lambda_{c_m} \mathcal{L}_c(p) + \lambda_{d_m} \mathcal{L}_d(p)), \tag{6}$$

where P_M represents the collection of all pixels in rendered image. λ_{f_m}, λ_{s_m}, λ_{c_m}, λ_{d_m} are weighting coefficients.

During tracking, utilizing an overly constrained loss function for pose estimation can lead to less precise camera pose and increased processing time, which is unsuitable for real-time tracking. Therefore, loss function for tracking is constructed based only on a weighted sum of RGB loss and depth loss:

$$\mathcal{L}_{\text{tracking}} = \sum_{p \in P_T} (\lambda_{c_t} \mathcal{L}_c(p) + \lambda_{d_t} \mathcal{L}_d(p)), \tag{7}$$

where λ_{c_t}, λ_{d_t} are weighting coefficients in tracking process. P_T represents pixels that are rendered from well-optimized part of 3D Gaussian map, which is area that rendered visibility silhouette $Sil(p)$ is greater than 0.99.

3.3 Semantic-informed Bundle Adjustment

Currently, existing radiance field-based semantic SLAM systems utilize the latest input RGB-D frame to construct RGB and depth loss for pose estimation, while optimization of scene representation is then performed using estimated camera pose and the latest frame. However, relying solely on single-frame constraint for pose optimization can lead to cumulative drift in the tracking process due to the lack of global consistency. Furthermore, using only single-frame information for scene representation optimization can result in updates to the scene that are inconsistent on a global semantic level. To address this problem, we propose semantic-informed bundle adjustment, as illustrated in Fig. 3b, to achieve joint optimization of 3D Gaussian representation and camera poses by leveraging multi-view constraints and semantic associations.

In semantic-informed bundle adjustment, we leverage the consistency of multi-view semantics to establish constraints. Specifically, rendered semantic feature is warped to its co-visible keyframe j using estimated relative pose T_i^j, and constructs loss with extracted semantic feature F_j^e of keyframe j to obtain $\mathcal{L}_{\text{BA-sem}}$:

$$\mathcal{L}_{\text{BA-sem}} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (|T_i^j \cdot \mathcal{G}(T_i, e) - F_j^e|), \tag{8}$$

where $\mathcal{G}(T_i, e)$ represents splatted semantic embedding from 3D Gaussian $\mathcal{G}$ using camera pose T_i. Moreover, to achieve geometry and appearance consistency, we also warp rendered RGB and depth to co-visible keyframes to construct loss with corresponding keyframes following the similar approach in Eq.(8):

$$\begin{aligned} \mathcal{L}_{\text{BA-rgb}} &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (|T_i^j \cdot \mathcal{G}(T_i, c) - F_j^c|), \\ \mathcal{L}_{\text{BA-depth}} &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (|T_i^j \cdot \mathcal{G}(T_i, d) - F_j^d|), \end{aligned} \tag{9}$$

where F_j^c and F_j^d are color and depth of keyframe j respectively. $\mathcal{G}(T_i, c)$ and $\mathcal{G}(T_i, d)$ represents rendered RGB and depth. Therefore, overall loss function $\mathcal{L}_{\text{BA}}$ for joint optimization of corresponding keyframe poses and 3D Gaussian scene representation is the weighted sum of the above losses:

$$\mathcal{L}_{\text{BA}} = \lambda_e \mathcal{L}_{\text{BA-sem}} + \lambda_c \mathcal{L}_{\text{BA-rgb}} + \lambda_d \mathcal{L}_{\text{BA-depth}}, \tag{10}$$

where λ_e, λ_c, λ_d are weighting coefficients. This design leverages fast rendering capability of 3D Gaussian to achieve joint optimization of scene representation and camera poses in real time. Furthermore, semantic-informed BA integrates consistency and correlation of multi-perspective semantic, geometry, and appearance information, resulting in low-drift tracking and consistent mapping.

Table 1: Quantitative comparison of tracking accuracy for our proposed SemGauss-SLAM with other dense visual SLAM and semantic SLAM methods on Replica dataset [21]. We utilize RMSE (cm) metric. To ensure more objectivity in the results, each scene is tested and averaged with three independent runs. Our work outperforms previous radiance field-based semantic SLAM methods across all scenes. Best results are highlighted as **first**, second.

	Methods	room0	room1	room2	office0	office1	office2	office3	office4	Avg.
Visual SLAM	iMAP [23]	6.33	3.46	2.65	3.31	1.42	7.17	6.32	2.55	4.15
	NICE-SLAM [33]	1.86	2.37	2.26	1.50	1.01	1.85	5.67	3.53	2.51
	Vox-Fusion [29]	1.37	1.90	1.47	1.35	1.76	1.18	1.11	1.64	1.47
	Co-SLAM [25]	0.72	0.85	1.02	0.69	0.56	2.12	1.62	0.87	1.06
	ESLAM [6]	0.76	0.71	0.56	0.53	0.49	0.58	0.74	0.64	0.62
	Point-SLAM [19]	0.61	0.41	0.37	**0.38**	0.48	0.54	0.69	0.72	0.52
	SplaTAM [7]	**0.31**	**0.40**	**0.29**	0.47	**0.27**	**0.29**	**0.32**	**0.55**	**0.36**
Semantic SLAM	SNI-SLAM [32]	0.50	0.55	0.45	0.35	0.41	0.33	0.62	0.50	0.46
	DNS SLAM [9]	0.49	0.46	0.38	**0.34**	0.35	0.39	0.62	0.60	0.45
	SemGauss-SLAM (Ours)	**0.26**	**0.42**	**0.27**	**0.34**	**0.17**	**0.32**	**0.36**	**0.49**	**0.33**

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate the performance of SemGauss-SLAM on two datasets with semantic ground truth annotations, including 8 scenes on simulated dataset Replica [21] and 5 scenes on real-world dataset ScanNet [2].

Metrics. To evaluate the SLAM system and rendering quality, we follow metrics from [19]. For mapping evaluation, we use Depth L1 (cm). For tracking accuracy evaluation, we utilize ATE RMSE (cm) [22]. Moreover, we use PSNR (dB), SSIM, and LPIPS for evaluating RGB image rendering performance. Semantic segmentation is evaluated with respect to mIoU accuracy.

Baselines. We compare our method with the existing state of-the-art dense visual SLAM, including NeRF-based SLAM [6,19,23,25,29,33] and 3D Gaussian SLAM [7]. For comparison of dense semantic SLAM performance, we consider NeRF-based semantic SLAM including SNI-SLAM [32], DNS SLAM [9], and NIDS-SLAM [4] as baseline.

Implementation Details. We run SemGauss-SLAM on NVIDIA RTX 4090 GPU. For experimental settings, we perform mapping every 8 frames. The weighting coefficients of each loss are $\lambda_{f_m} = 0.01$, $\lambda_{s_m} = 0.01$, $\lambda_{c_m} = 0.5$, $\lambda_{d_m} = 1$ in mapping, $\lambda_{c_t} = 0.5$ and $\lambda_{d_t} = 1$ in tracking. Moreover, we set $\lambda_e = 0.004$, $\lambda_c = 0.5$, $\lambda_c = 1$ in semantic-informed bundle adjustment. Please refer to the supplementary for further details of our implementation.

4.2 Experimental Results

Tracking Results. As shown in Tab. 1 and 2, our method achieves up to 47% relative increase in tracking accuracy compared with other dense semantic SLAM

Table 2: We compare our proposed SemGauss-SLAM with other existing radiance field-based SLAM methods on ScanNet dataset [2] for tracking metric RMSE (cm). The results are an average of three independent runs. Best results are highlighted as **first**, second, third.

Methods	scene0000	scene0059	scene0169	scene0181	scene0207	Avg.
NICE-SLAM [33]	12.00	14.00	10.90	13.40	**6.20**	11.30
Vox-Fusion [29]	68.84	24.18	27.28	23.30	9.41	30.60
Point-SLAM [19]	**10.24**	**7.81**	22.16	14.77	9.54	12.90
SplaTAM [7]	12.83	10.10	12.08	11.10	7.46	10.71
SemGauss-SLAM (Ours)	12.56	7.97	**9.05**	**9.78**	8.97	**9.67**

Table 3: Quantitative comparison of training view rendering performance on Replica dataset [21]. Our work outperforms other radiance field-based semantic SLAM methods on all three metrics across all scenes.

	Methods	Metrics	room0	room1	room2	office0	office1	office2	office3	office4	Avg.
Visual SLAM	NICE-SLAM [33]	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
	Vox-Fusion [29]	PSNR ↑	22.39	22.36	23.92	27.79	29.83	20.33	23.47	25.21	24.41
		SSIM ↑	0.683	0.751	0.798	0.857	0.876	0.794	0.803	0.847	0.801
		LPIPS ↓	0.303	0.269	0.234	0.241	0.184	0.243	0.213	0.199	0.236
	Co-SLAM [25]	PSNR ↑	27.27	28.45	29.06	34.14	34.87	28.43	28.76	30.91	30.24
		SSIM ↑	0.910	0.909	0.932	0.961	0.969	0.938	0.941	0.955	0.939
		LPIPS ↓	0.324	0.294	0.266	0.209	0.196	0.258	0.229	0.236	0.252
	ESLAM [6]	PSNR ↑	25.32	27.77	29.08	33.71	30.20	28.09	28.77	29.71	29.08
		SSIM ↑	0.875	0.902	0.932	0.960	0.923	0.943	0.948	0.945	0.929
		LPIPS ↓	0.313	0.298	0.248	0.184	0.228	0.241	0.196	0.204	0.239
	SplaTAM [7]	PSNR ↑	**32.86**	**33.89**	**35.25**	**38.26**	**39.17**	**31.97**	**29.70**	**31.81**	**34.11**
		SSIM ↑	**0.978**	**0.969**	**0.979**	**0.977**	**0.978**	**0.968**	**0.949**	**0.949**	**0.968**
		LPIPS ↓	**0.072**	**0.103**	**0.081**	**0.092**	**0.093**	**0.102**	**0.121**	**0.152**	**0.102**
Semantic SLAM	SNI-SLAM [32]	PSNR ↑	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
	SemGauss-SLAM (Ours)	PSNR ↑	**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**

methods. This improvement is attributed to semantic-informed bundle adjustment, which provides multi-view constraint leveraging consistency of semantic information to reduce accumulated drift in tracking process.

Rendering Quality Results. Tab. 3 shows rendering quality on the input views of Replica dataset [21]. Our method achieves the best performance in PSNR, SSIM, and LPIPS compared with other dense semantic SLAM.

Reconstruction Results. As shown in Tab. 4, our method achieves up to 77.3% increase in reconstruction accuracy compared with other radiance field-based methods. This enhancement in performance is due to the incorporation of semantic-informed BA process, achieving joint optimization for both camera poses and scene representation. Specifically, our method exploits the geometric consistency of co-visible frames to construct geometry constraints for more precise reconstruction.

Table 4: We compare our proposed SemGauss-SLAM with other existing radiance field-based SLAM methods on Replica [21] for reconstruction metric Depth L1 (cm). The results are an average of three independent runs. Our method outperforms other methods by up to 77.3%.

	Methods	room0	room1	room2	office0	office1	office2	office3	office4	Avg.
Visual SLAM	NICE-SLAM [33]	1.81	1.44	2.04	1.39	1.76	8.33	4.99	2.01	2.97
	Vox-Fusion [29]	1.09	1.90	2.21	2.32	3.40	4.19	2.96	1.61	2.46
	Co-SLAM [25]	1.05	0.85	2.37	1.24	1.48	1.86	1.66	1.54	1.51
	ESLAM [6]	**0.73**	**0.74**	**1.26**	**0.71**	**1.02**	**0.93**	**1.03**	**1.18**	**0.95**
Semantic SLAM	SNI-SLAM [32]	0.55	0.58	0.87	0.55	0.97	0.89	**0.75**	0.97	0.77
	SemGauss-SLAM (Ours)	**0.54**	**0.46**	**0.43**	**0.29**	**0.22**	**0.51**	0.98	**0.56**	**0.50**

Table 5: Quantitative comparison of input views semantic segmentation performance on Replica [21] for semantic metric mIoU(%). For one scene, we calculate mIoU between splatted semantic maps and ground truth labels, which is calculated every 4 frames, to obtain average mIoU. Our method achieves highest semantic accuracy across all scenes.

Methods	room0	room1	room2	office0	office1	office2	office3	office4
NIDS-SLAM [4]	82.45	84.08	76.99	85.94	–	–	–	–
DNS SLAM [9]	88.32	84.90	81.20	84.66	–	–	–	–
SNI-SLAM [32]	88.42	87.43	86.16	87.63	78.63	86.49	74.01	80.22
SemGauss-SLAM (Ours)	**92.81**	**94.10**	**94.72**	**95.23**	**90.11**	**94.93**	**92.93**	**94.82**

Semantic Segmentation Results. As shown in Tab. 5, our work outperforms other dense semantic SLAM methods by up to 26% on mIoU metric and achieves 95% mIoU accuracy. Such enhancement attributes to the integration of semantic feature embedding into 3D Gaussian for enriched semantic representation, and semantic feature-level loss for direct guidance of semantic optimization. Moreover, our proposed semantic-informed BA also contributes to high semantic precision, as it leverages multiple co-visible frames to construct a globally consistent semantic map, thereby achieving high-precision semantic representation.

Novel View Semantic Evaluation Results. As shown in Tab. 6, our method achieves up to 75% increase in semantic segmentation accuracy for semantic novel view synthesis compared with existing dense semantic SLAM. By introducing 3D Gaussian feature embedding for semantic representation, our method enables continuous semantic modeling. This modeling is crucial for generating

Table 6: Quantitative comparison of semantic novel view synthesis performance on Replica [21] for semantic metric mIoU(%). For one scene, we randomly choose 100 new viewpoints for semantic novel view synthesis evaluation. Our method outperforms SNI-SLAM [32] across all scenes, showing excellent 3D semantic mapping accuracy.

Methods	room0	room1	room2	office0	office1	office2	office3	office4	Avg.
SNI-SLAM [32]	51.20	50.10	54.80	70.21	63.47	58.91	64.31	71.04	60.51
SemGauss-SLAM (Ours)	**89.63**	**84.72**	**86.51**	**93.60**	**89.57**	**92.80**	**92.41**	**92.10**	**90.17**

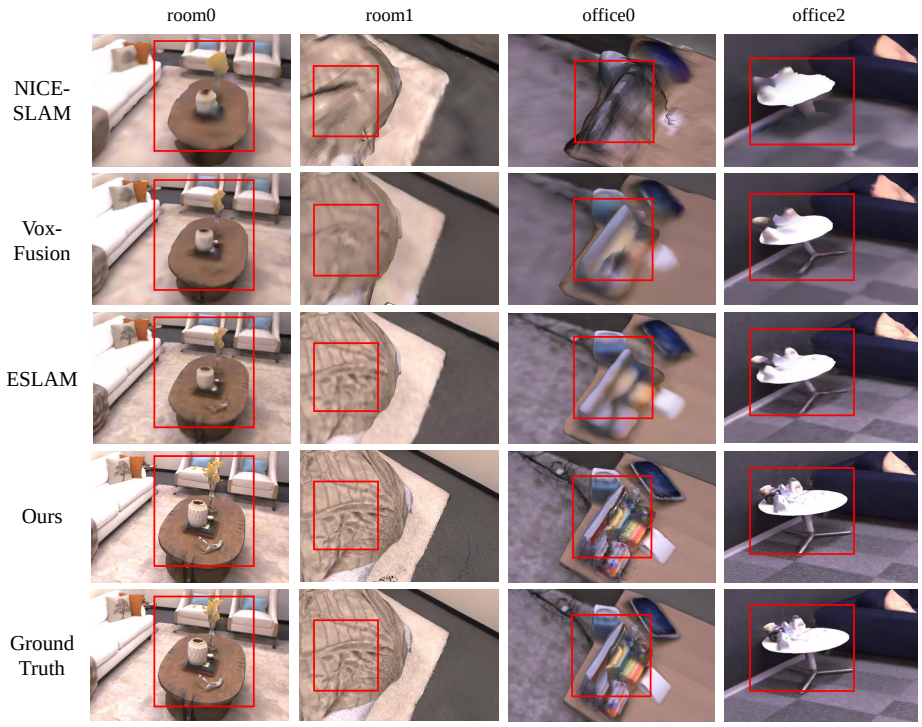


Fig. 4: Qualitative comparison on rendering quality of our method and baseline. We visualize 4 selected scenes of Replica dataset [21] and details are highlighted with red color boxes. Our method achieves photo-realistic rendering quality compared with other methods, especially in areas with rich textural information.

semantically coherent scenes, as it reduces the occurrence of sharp transitions that can cause inconsistencies, ensuring semantic representation accuracy from different viewpoints for high-precision novel view semantic synthesis.

Visualization. Fig. 4 shows rendering quality comparison of 4 scenes with interesting regions highlighted with colored boxes. For small objects, such as vases and items on the table, other methods fail to reconstruct them clearly. Our method leverages the high-quality rendering capability of 3D Gaussian representation and introduces semantic-informed BA to achieve detailed geometric reconstruction results. Specifically, our method enhances multi-view geometry consistency by BA to ensure that the reconstructed geometry aligns accurately across all observed viewpoints, leading to precise geometry reconstruction. As shown in Fig. 5, our method achieves superior novel view semantic segmentation accuracy compared with the baseline SNI-SLAM [32]. It can be observed that SNI-SLAM struggles with segmenting ceilings in novel view synthesis as they are less frequently observed during the mapping process. Consequently, the ceiling features are poorly modeled in the semantic scene representation. Our method introduces semantic-informed BA to construct multi-view constraints, enabling

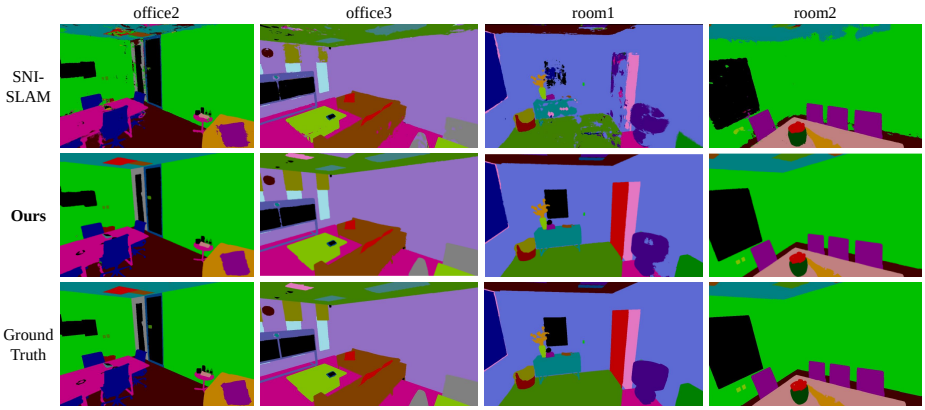


Fig. 5: Qualitative comparison on semantic novel view synthesis of our method and baseline semantic SLAM method SNI-SLAM [32] on 4 scenes of Replica [21].

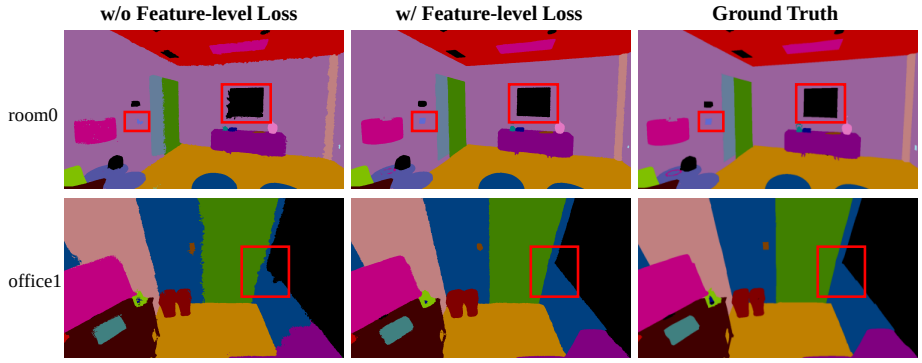


Fig. 6: Semantic rendering results and ground truth labels of feature-level loss ablation on two scenes of Replica [21].

areas that are sparsely observed to effectively utilize the limited information from co-visible frames, thus establishing sufficient constraints for accurate semantic reconstruction.

4.3 Ablation Study

We perform ablation study on three scenes of Replica dataset [21] in Tab. 7 to validate the effectiveness of feature-level loss and semantic-informed BA in SemGauss-SLAM. Moreover, ablation of semantic-informed BA component is conducted in Tab. 8.

Feature-level Loss. Tab. 7 shows that incorporating feature-level loss significantly enhances semantic segmentation performance, while having little effect on tracking and geometric reconstruction. This result occurs because feature-level loss influences only the optimization of semantic features, without affecting

Table 7: Ablation study of our contributions on three scenes of Replica [21].

Methods	room0			room1			office1		
	RMSE ↓	Depth L1 ↓	mIoU ↑	RMSE ↓	Depth L1 ↓	mIoU ↑	RMSE ↓	Depth L1 ↓	mIoU ↑
w/o feature-level loss	**0.26**	**0.54**	83.60	**0.42**	**0.46**	86.10	**0.17**	**0.22**	80.13
w/o semantic-informed BA	0.35	0.70	90.10	0.51	0.60	91.08	0.21	0.30	87.73
SemGauss-SLAM (Ours)	**0.26**	**0.54**	**92.81**	**0.42**	**0.46**	**94.10**	**0.17**	**0.22**	**90.11**

Table 8: Ablation study of semantic-informed BA on three scenes of Replica [21]. (w/o semantic) without adding semantic constraint loss $\mathcal{L}_{\text{BA-sem}}$; (w/o RGB and depth) without adding RGB and depth constraint loss $\mathcal{L}_{\text{BA-rgb}}$ and $\mathcal{L}_{\text{BA-depth}}$.

Methods	room0			room1			office1		
	RMSE ↓	Depth L1 ↓	mIoU ↑	RMSE ↓	Depth L1 ↓	mIoU ↑	RMSE ↓	Depth L1 ↓	mIoU ↑
w/o semantic	0.35	0.66	90.15	0.50	0.49	91.50	0.20	0.24	87.95
w/o RGB and depth	0.31	0.70	92.01	0.48	0.59	93.90	0.18	0.28	89.60
SemGauss-SLAM (Ours)	**0.26**	**0.54**	**92.81**	**0.42**	**0.46**	**94.10**	**0.17**	**0.22**	**90.11**

the optimization of geometry and pose estimation. As shown in Fig. 6, utilizing feature-level loss can achieve improved boundary segmentation and finer segmentation of small objects. This enhancement occurs because feature loss compels the scene representation to capture high-dimensional and direct information within the feature space, leading to the capability of distinguishing intricate details and subtle variations within the scene.

Semantic-informed BA. Tab. 7 shows that introducing semantic-informed BA leads to enhancements in tracking, reconstruction, and semantic segmentation. This improvement is due to the joint optimization of camera poses and scene representation, which is informed by multi-view constraints. As shown in Tab. 8, lacking semantic constraint leads to a significant decrease in tracking performance compared with the absence of color and depth constraints. Such result suggests that multi-view semantic constraints provide more comprehensive and accurate information, due to the consistency of semantics across multiple perspectives. Moreover, the absence of semantic constraint can lead to reduced semantic precision, while lacking color and depth constraints results in decreased reconstruction accuracy.

5 Conclusion

We propose SemGauss-SLAM, a novel dense semantic SLAM system utilizing 3D Gaussian representation that enables dense visual mapping, robust camera tracking, and 3D semantic mapping of the whole scene. We incorporate semantic feature embedding into 3D Gaussian for Gaussian semantic representation, leading to real-time dense semantic mapping. Moreover, we propose feature-level loss for 3D Gaussian scene optimization to achieve accurate semantic representation. In addition, we introduce semantic-informed BA that enables joint optimization of camera poses and 3D Gaussian representation by establishing multi-view semantic constraints, resulting in low-drift tracking and precise mapping.

References

1. Bao, Y., Yang, Z., Pan, Y., Huan, R.: Semantic-direct visual odometry. IEEE Robotics and Automation Letters **7**(3), 6718–6725 (2022) 1
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 conference on computer vision and pattern recognition. pp. 5828–5839 (2017) 3, 9, 10
3. Grinvald, M., Furrer, F., Novkovic, T., Chung, J.J., Cadena, C., Siegwart, R., Nieto, J.: Volumetric instance-aware semantic mapping and 3d object discovery. IEEE Robotics and Automation Letters **4**(3), 3037–3044 (2019) 3
4. Haghighi, Y., Kumar, S., Thiran, J.P., Van Gool, L.: Neural implicit dense semantic slam. arXiv preprint arXiv:2304.14560 (2023) 3, 9, 11
5. Hughes, N., Chang, Y., Carlone, L.: Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. arXiv preprint arXiv:2201.13360 (2022) 3
6. 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) 3, 9, 10, 11
7. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. arXiv preprint arXiv:2312.02126 (2023) 1, 4, 6, 9, 10
8. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023) 1, 4, 5, 7
9. Li, K., Niemeyer, M., Navab, N., Tombari, F.: Dns slam: Dense neural semantic-informed slam. arXiv preprint arXiv:2312.00204 (2023) 1, 3, 4, 9, 11
10. Lianos, K.N., Schonberger, J.L., Pollefeys, M., Sattler, T.: Vso: Visual semantic odometry. In: Proceedings of the European conference on computer vision (ECCV). pp. 234–250 (2018) 1
11. Liu, Z., Milano, F., Frey, J., Siegwart, R., Blum, H., Cadena, C.: Unsupervised continual semantic adaptation through neural rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3031–3040 (2023) 1
12. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: 3DV (2024) 4
13. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. arXiv preprint arXiv:2312.06741 (2023) 1, 4
14. McCormac, J., Clark, R., Bloesch, M., Davison, A., Leutenegger, S.: Fusion++: Volumetric object-level slam. In: 2018 international conference on 3D vision (3DV). pp. 32–41. IEEE (2018) 3
15. McCormac, J., Handa, A., Davison, A., Leutenegger, S.: Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In: 2017 IEEE International Conference on Robotics and automation (ICRA). pp. 4628–4635. IEEE (2017) 1, 3
16. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 1

17. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) 7
18. Rosinol, A., Abate, M., Chang, Y., Carlone, L.: Kimera: an open-source library for real-time metric-semantic localization and mapping. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 1689–1696. IEEE (2020) 1, 3
19. 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) 3, 9, 10
20. Schmid, L., Delmerico, J., Schönberger, J.L., Nieto, J., Pollefeys, M., Siegwart, R., Cadena, C.: Panoptic multi-tsdfs: a flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 8018–8024. IEEE (2022) 3
21. 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) 3, 9, 10, 11, 12, 13, 14
22. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. pp. 573–580. IEEE (2012) 9
23. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6229–6238 (2021) 3, 9
24. Tian, Y., Chang, Y., Arias, F.H., Nieto-Granda, C., How, J.P., Carlone, L.: Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. IEEE Transactions on Robotics **38**(4) (2022) 3
25. 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) 3, 9, 10, 11
26. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. arXiv preprint arXiv:2310.08528 (2023) 4
27. Yan, C., Qu, D., Wang, D., Xu, D., Wang, Z., Zhao, B., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. arXiv preprint arXiv:2311.11700 (2023) 1, 4
28. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians for modeling dynamic urban scenes. arXiv preprint arXiv:2401.01339 (2024) 4
29. 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) 3, 9, 10, 11
30. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023) 4
31. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting. arXiv preprint arXiv:2312.10070 (2023) 1, 4
32. Zhu, S., Wang, G., Blum, H., Liu, J., Song, L., Pollefeys, M., Wang, H.: Sni-slam: Semantic neural implicit slam. arXiv preprint arXiv:2311.11016 (2023) 1, 2, 3, 4, 9, 10, 11, 12, 13

33. 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) 3, 9, 10, 11
34. Zhuang, J., Kang, D., Cao, Y.P., Li, G., Lin, L., Shan, Y.: Tip-editor: An accurate 3d editor following both text-prompts and image-prompts. arXiv preprint arXiv:2401.14828 (2024) 4