SGS-SLAM: Semantic Gaussian Splatting For Neural Dense SLAM

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

Abstract. We present SGS-SLAM, the first semantic visual SLAM system based on Gaussian Splatting. It incorporates appearance, geometry, and semantic features through multi-channel optimization, addressing the oversmoothing limitations of neural implicit SLAM systems in high-quality rendering, scene understanding, and object-level geometry. We introduce a unique semantic feature loss that effectively compensates for the shortcomings of traditional depth and color losses in object optimization. Through a semantic-guided keyframe selection strategy, we prevent erroneous reconstructions caused by cumulative errors. Extensive experiments demonstrate that SGS-SLAM delivers state-of-the-art performance in camera pose estimation, map reconstruction, precise semantic segmentation, and object-level geometric accuracy, while ensuring real-time rendering capabilities.

Keywords: SLAM · 3D Reconstruction · 3D Segmentation

1 Introduction

Dense Visual Simultaneous Localization and Mapping (SLAM) is a crucial problem in the field of computer vision. It aims to reconstruct a dense 3D map in an unseen environment while simultaneously tracking the camera poses. Traditional visual SLAM systems [6,29,31,34] stand out in sparse mapping using point clouds and voxels, but fall short in dense reconstruction. To extract dense geometric information for high-quality representation, learning-based SLAM methods [1,37] have gained wild attention. They demonstrate proficiency in generating decent 3D global maps meanwhile exhibiting robustness on noises and outliers. Drawing inspiration from advancements in the neural radiance field (NeRF) [27], NeRF-based SLAM approaches [8,19,20,36,39,45,48] have made further progress. They excel in producing accurate and high-fidelity global reconstruction by capturing dense photometric information through differentiable rendering.

However, NeRF-based SLAM methods employ multi-layer perceptrons (MLPs) as the implicit neural representation of scenes, which introduces several challenging limitations. Primarily, MLP models struggle with over-smoothing issues at

* These authors contributed equally to this work.
† Corresponding author. Email: whyu@dlut.edu.cn

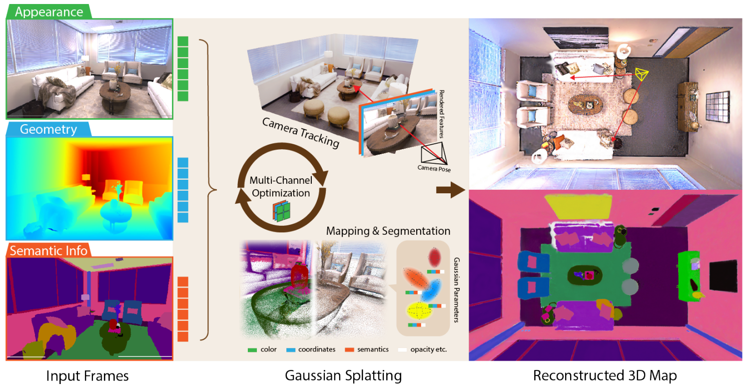

Fig. 1: The illustration of the proposed SGS-SLAM. It employs 2D inputs encompassing appearance, geometry, and semantic information, leveraging Gaussian Splatting and differentiable rendering for multi-channel parameter optimization. During the mapping process, SGS-SLAM maps the 2D semantic prior to the 3D scene, jointly optimizing it via the mapping loss for accurate 3D segmentation outcomes.

the edge of objects, leading to a lack of fine-grained details in the map. This challenge also brings difficulties in disentangling the representation of objects, making it non-trivial to segment, edit, and manipulate objects within the scene. Moreover, when applied to larger scenes, MLP models are prone to catastrophic forgetting. This means that incorporating new scenes can adversely affect the precision of previously learned models, thereby reducing overall performance. Additionally, NeRF-based methods are computationally inefficient. Since the entire scene is modeled through one or several MLPs, it necessitates extensive model tuning for adding or updating scenes.

In this context, as opposite to NeRF-based neural representation, our exploration shifts towards the volumetric representation based on the 3D Gaussian Radiance Field [18]. This approach marks a significant shift and offers notable advantages in the scene representation.

Benefits from its rasterization of 3D primitives, Gaussian Splatting exhibits remarkably fast rendering speeds and allows direct gradient flow to each Gaussian's parameters. This results in an almost linear projection between the dense photometric loss and parameters during optimization [17], unlike the hierarchical pixel sampling and indirect gradient flow through multiple non-linear layers seen in NeRF models. Moreover, the direct projection capability simplifies the addition of new channels to the Gaussian field, thereby enabling dynamic multi-channel feature rendering. Crucially, we integrate a semantic map into the 3D Gaussian field, essential for applications in robotics and mixed reality. This integration allows real-time rendering of appearance, depth, and semantic color.

When compared with neural implicit semantic SLAM systems, such as DNS-SLAM [20] and SNI-SLAM [47], our system demonstrates remarkable superiority in terms of rendering speed, reconstruction quality, and segmentation accuracy. Leveraging these benefits, our method enables precise editing and manipulation of specific scene elements while preserving the high fidelity of the overall rendering. Furthermore, using explicit spatial and semantic information to identify scene content can be instrumental in optimizing camera tracking. Particularly, we incorporate a two-level adjustment based on geometric and semantic criteria for keyframes selection. This process relies on recognizing objects that have been previously seen in the trajectory. Extensive experiments are conducted on both synthetic and real-world scene benchmarks. These experiments compare our method against both implicit NeRF-based approaches [16, 39, 42, 48], and novel 3D-Gaussian-based methods [17], evaluating performance in mapping, tracking, and semantic segmentation.

Overall, our work presents several key contributions, summarized as follows:

- We introduce SGS-SLAM, the first semantic RGB-D SLAM system grounded in 3D Gaussians. SGS-SLAM employs an explicit volumetric representation, enabling swift and real-time camera tracking and scene mapping. More importantly, it utilizes 2D semantic maps to learn 3D semantic representations expressed by Gaussians. Compared with previous NeRF-based methods which offer over-smooth object edges, SGS-SLAM provides high-fidelity reconstruction and optimal segmentation precision.
- In SGS-SLAM, semantic maps provide additional supervision for optimizing parameters and selecting keyframes. We employ a multi-channel parameter optimization strategy where appearance, geometric, and semantic signals collectively contribute to camera tracking and scene reconstruction. Furthermore, SGS-SLAM utilizes these diverse channels for keyframe selection during the tracking phase, concentrating on actively recognizing objects seen earlier in the trajectory.
- Utilizing the semantic map, SGS-SLAM provides a highly accurate disentangled object representation in 3D scenes, laying a solid foundation for downstream tasks such as scene editing and manipulation. SGS-SLAM facilitates the dynamic moving, rotating, or removal of objects that is achieved by grouping Gaussians by specifying the semantic labels of the objects.

2 Related Work

2.1 Semantic SLAM

Semantic information is of great importance for SLAM systems [13, 29, 32, 40], which is a crucial requirement for applications in robotics and VR or AR fields. Real-time dense semantic SLAM systems [1, 33, 34] integrate semantic information into 3D geometric representations. Traditional semantic SLAM systems rely on sparse 3D semantic expressions, such as voxel [14], point cloud [30], and signed distance field [30]. These methods struggle with accurately interpreting complex environments due to limited semantic understanding. This results in a simplified categorization of environmental features, which may not capture the full range of objects and their relationships within a space. Moreover, these methods exhibit limitations regarding reconstruction speed, high-fidelity model acquisition, and memory usage.

2.2 Neural Implicit SLAM

Methods based on NeRF [26], which handle complex topological structures and differentiable scene representation methods, have garnered significant attention, leading to the development of neural implicit SLAM methods [3,9–11,21,22,46]. iMAP [36] uses a single MLP for scene representation, which shows limitations in large-scale scenes. NICE-SLAM [48] uses pre-trained multiple MLPs for hierarchical scene representation. Co-SLAM [39] combines pixel-set-based keyframe tracking with one-blob encoding. Go-SLAM [45] uses Droid-SLAM [38] as the tracking system and multi-resolution hash encoding [28] for mapping. However, these methods cannot utilize semantic information in the map. NIDS-SLAM [12] leverages the tracking system of ORB-SLAM3 [2] and Instant-NGP [28] for mapping but does not optimize joint semantic features for 3D reconstruction. DNS-SLAM [20] proposes a 2D semantic prior system that provides multi-view geometry constraints but does not optimize 3D reconstruction with semantic features. DNS-SLAM [20] and SNI-SLAM [47] introduce semantic loss for geometric supervision but remain limited by the efficiency constraints of NeRF's volume rendering.

2.3 3D Gaussian Splatting SLAM

Utilizing the outstanding performance and fast rasterization capabilities of 3D Gaussian Splatting [18], Gaussian-based SLAM systems offer higher efficiency and accuracy on scene reconstruction [7,15,17,23,25,41,44]. However, existing Gaussian-based SLAM systems lack the ability to recognize semantic information in scenes. To bridge this gap, We utilize the semantic map during keyframe selection and integrate the semantic feature loss in the tracking and mapping process. This allows us to obtain more effective and higher-quality scene segmentation outcomes meanwhile preserving real-time processing performance.

3 Method

SGS-SLAM is a Gaussian-based semantic visual SLAM system. Sec. 3.1 introduces its multi-channel Gaussian representation for joint parameter optimization. Like previous SLAM techniques, our method can be split into two processes: tracking and mapping. The tracking process estimates the camera pose of each frame while keeping the scene parameters fixed. Mapping optimizes the scene representations based on the estimated camera pose. Sec. 3.2 explains the breakdown steps in detail. In addition, Sec. 3.3 presents scene manipulation as a case study for downstream tasks. Fig. 1 shows an overview of our system.

3.1 Multi-Channel Gaussian Representation

The scene is represented using a Gaussian influence function $f(\cdot)$ on the map, For simplicity, these Gaussians are isotropic, as proposed in [17]:

$$f^{\text{3D}}(x) = \sigma \exp\left(-\frac{\|x-\mu\|^2}{2r^2}\right) \tag{1}$$

Here, $\sigma \in [0,1]$ indicates opacity, $\mu \in \mathbb{R}^3$ represents the center position, and r denotes the radius. Each Gaussian also carries RGB colors $c_i = [r_i \; b_i \; g_i]^T$.

In order to optimize the parameters of Gaussians to represent the scene, we need to render the Gaussians into 2D images in a differentiable manner. We use the render method from [24], providing extended functionality of rendering depth in colors. It works by splatting 3D Gaussians into the image plane via approximating the integral projection of the influence function $f(\cdot)$ along the depth dimension in pixel coordinates. The center of the Gaussian μ, radius r, and depth d (in camera coordinates) is splatted using the standard point rendering formula:

$$\mu^{\text{2D}} = K\frac{E_t\mu}{d}, \quad r^{\text{2D}} = \frac{lr}{d}, \quad d = (E_t\mu)_z \tag{2}$$

where K is the camera intrinsic matrix, E_t is the extrinsic matrix capturing the rotation and translation of the camera at frame t, l is the focal length. The influence of all Gaussians on this pixel can be combined by sorting the Gaussians in depth order and performing front-to-back volume rendering:

$$C_{\text{pix}} = \sum_{i=1}^{n} c_i f_{i,\text{pix}}^{\text{2D}} \prod_{j=1}^{i-1}(1 - f_{j,\text{pix}}^{\text{2D}}) \tag{3}$$

The pixel-level rendered color C_{pix} is the sum over the colors of each Gaussian c_i and weighted by the influence function $f_{i,\text{pix}}^{\text{2D}}$ (replace the 3D means and covariance matrices with the 2D splatted versions), multiplied by an occlusion term taking into account the effect of all Gaussians in front of the current Gaussian. Similarly, the depth can be rendered as:

$$D_{\text{pix}} = \sum_{i=1}^{n} d_i f_{i,\text{pix}}^{\text{2D}} \prod_{j=1}^{i-1}(1 - f_{j,\text{pix}}^{\text{2D}}) \tag{4}$$

where d_i denotes the depth of each Gaussian. By setting $d_i = 1$, we can calculate a silhouette, $Sil_{\text{pix}} = D_{\text{pix}}(d_i = 1)$, which assists in determining whether a pixel is visible in the current view [17]. This aspect of visibility is essential for camera pose estimation, as it relies on the current reconstructed map. Additionally, it is also employed in map reconstruction, where new Gaussians are introduced in pixels lacking sufficient information.

While acquiring 3D semantic information is challenging and usually demands extensive manual labeling, the 2D semantic label is more accessible prior. In our approach, we leverage 2D semantic labels, which are often provided in datasets or can be easily obtained using off-the-shelf methods. We assign distinct channels to the parameters of Gaussians to denote their semantic labels and colors. During the rendering process, the 2D semantic map can be rendered from the reconstructed 3D scene as follows:

$$S_{\text{pix}} = \sum_{i=1}^{n} s_i f_{i,\text{pix}}^{\text{2D}} \prod_{j=1}^{i-1} (1 - f_{j,\text{pix}}^{\text{2D}}) \tag{5}$$

where $s_i = [r_i \ b_i \ g_i]^T$ denotes the semantic color associated with the Gaussian. This semantic color is optimized jointly with the appearance color and depth during the mapping process.

The Gaussian representations employed in SGS-SLAM facilitate high-quality reconstructions at high rendering speed, offering exceptional accuracy in capturing complex textures and geometry with remarkable detail and efficiency. Furthermore, the integration of semantic features within our method significantly advances optimal scene interpretation and precise object-level geometry, effectively mitigating the oversmoothing issues prevalent in NeRF models.

3.2 Tracking and Mapping

Camera Pose Estimation Given the first frame, the camera pose is set to identity and used as the reference coordinates for the following tracking and mapping procedure. While assessing the camera pose of an RGB-D view at a new timestep, the initial camera pose is determined by adding a displacement to the previous pose, assuming constant velocity, as $E_{t+1} = E_t + (E_t - E_{t-1})$. Following this, the current pose is iteratively refined by minimizing the tracking loss between the ground truth color (C_{pix}^{GT}), depth images (D_{pix}^{GT}), and semantic map (S_{pix}^{GT}) and their differentiably rendered views:

$$\mathcal{L}_{\text{tracking}} = \sum_{\text{pix}} (Sil_{\text{pix}} > T_{\text{sil}})(\lambda_D |D_{\text{pix}}^{GT} - D_{\text{pix}}| + \lambda_C |C_{\text{pix}}^{GT} - C_{\text{pix}}| + \lambda_S |S_{\text{pix}}^{GT} - S_{\text{pix}}|) \tag{6}$$

Here, only those rendered pixels with a sufficiently large silhouette are factored into the loss calculation. The threshold T_{sil} is designed to make use of the map that has been previously optimized and has high certainty to be visible in the current camera view.

Keyframes Selection and Weighting During the tracking phase of SLAM systems, keyframes are identified and stored simultaneously. These keyframes, providing different views of objects, are critical for mapping to refine 3D scene reconstruction. SGS-SLAM captures and stores keyframes at constant time intervals. Subsequently, keyframes associated with the current frame are chosen based on geometric and semantic constraints. Specifically, we randomly select pixels from the current frame and extract their corresponding Gaussians G_{sample} in the

3D scene. These sampled G_{sample} are then projected onto the camera views of keyframes as G_{proj}, which are evaluated based on the geometric overlap ratio:

$$\eta = \frac{1}{\sum G_{\text{proj}}} \sum_{n=i} \{G_i | 0 \leq width(G_i) \leq W, 0 \leq height(G_i) \leq H\} \tag{7}$$

It represents the proportion of Gaussians captured within the camera view of the keyframes. W and H are the width and height of the camera view. The candidates with η lower than a certain threshold T_{geo} are removed. After the initial geometric-based selection, a second filter is conducted based on semantic criteria. We discard keyframes whose semantic maps S_{pix} are identical to the current frame's semantic map, as indicated by a high mIoU score. This threshold, denoted as T_{sem} intends to enhance map optimization from varying viewpoints, preferring views with low mIoU overlap. The remaining candidates are randomly sampled to serve as the selected keyframes associated with the current frame. In addition, we compute an uncertainty score for each keyframe, defined as $\mathcal{U}(t) = e^{-\tau t}$, with t representing the timestamp of the keyframe and τ being a decay coefficient. This uncertainty score is used to weight the mapping loss $\mathcal{L}_{\text{mapping}}$. The intuition behind this is that keyframes with a later timestamp index carry a higher uncertainty in reconstruction due to the accumulation of camera tracking errors along the trajectory.

Map Reconstruction The scene is modeled using Gaussians across three distinct channels: (1) their mean coordinates represent the geometric information of the scene, (2) their appearance colors depict the scene's visual appearance, and (3) their semantic colors indicate the semantic labels of objects. These parameters across the channels are jointly optimized during the process of Gaussian densification and optimization, whereas the camera pose, ascertained from tracking, remains fixed.

Starting with the first frame, all pixels contribute to initializing the map. In the process of map reconstruction at a new timestep, new Gaussians are introduced to areas of the map that are either insufficiently dense or display new geometry in front of the previously estimated map. The addition of new Gaussians is regulated by applying a mask to the pixels where either (ii) the silhouette value Sil_{pix} falls below a certain threshold, signifying a high uncertainty in visibility, or (ii) the ground-truth depth is much smaller than the estimated depth, suggesting the presence of new geometric entities.

After densification, the parameters of the map are optimized by minimizing the mapping loss:

$$\mathcal{L}_{\text{mapping}} = \mathcal{U}_t \sum_{\text{pix}} \lambda_D |D_{\text{pix}}^{GT} - D_{\text{pix}}| + \lambda_C \mathcal{L}_C + \lambda_S \mathcal{L}_S \tag{8}$$

where $\mathcal{L}_C$ and $\mathcal{L}_S$ are weighted SSIM loss [18] with respect to appearance image and semantic image:

$$\mathcal{L}(I_{\text{pix}}) = \sum_{\text{pix}} \alpha |I_{\text{pix}}^{GT} - I_{\text{pix}}| + (1-\alpha)(1 - ssim(I_{\text{pix}}^{GT}, I_{\text{pix}})) \tag{9}$$

λ_D, λ_C, λ_S, and α are predefined hyperparameters, and $\mathcal{U}_t$ is the uncertainty score defined in Sec. 3.2.

Compared to existing NeRF-based approaches [16,20,47,48] that necessitate complex model architectures and feature fusion strategies, SGS-SLAM adopts explicit Gaussian representation for mapping, resulting in high rendering speeds and optimal reconstruction quality. Compared to recent Gaussian-based methods [17,41], SGS-SLAM incorporates geometric, appearance, and semantic features for multi-channel rendering. This enables the joint optimization of parameters across different channels, remarkably enhancing the efficiency and effectiveness of both mapping and segmentation processes.

3.3 Scene Manipulation via Object-level Geometry

Given that the scene is represented explicitly by Gaussians, it becomes feasible to directly edit and manipulate a targeted group of Gaussians. In our case, Gaussian groups are identified based on their semantic labels. The mapping process generates these Gaussians, as defined in Eq. (1), allowing for further manipulation in the following manner:

$$f_{\text{edit}}^{\text{3D}}(G, \tilde{y}) = M(G, \tilde{y}) \cdot \Phi_T(f^{\text{3D}}(G), \tilde{y}) \tag{10}$$

where the edited Gaussians, $f_{\text{edit}}^{\text{3D}}$, are influenced by the visibility mask M, transition function Φ_T, and the Gaussian's semantic label $\tilde{y}$. The visibility mask M determines if the Gaussians should be retained (1) or removed (0) based on $\tilde{y}$. The transition function Φ_T applies a transformation to the Gaussian's coordinates on selected $\tilde{y}$, enabling spatial manipulation.

4 Experiment

4.1 Experimental Setup

Datasets We evaluate our method on both synthetic and real-world datasets. To compare with other neural implicit SLAM methods, we evaluate synthetic scenes from Replica dataset [35] and real-world scenes from ScanNet [4] and ScanNet++ [43] datasets. The ground-truth camera pose and semantic map of Replica are offered from simulation, and the ground-truth camera pose of ScanNet is generated by BundleFusion [5]. The ground-truth 2D semantic label is provided by the dataset.

Metrics We use PSNR, Depth-L1, SSIM, and LPIPS to evaluate the reconstruction quality. To evaluate the camera pose, we use the average absolute trajectory error (ATE RMSE). For semantic segmentation, we calculate mIoU score.

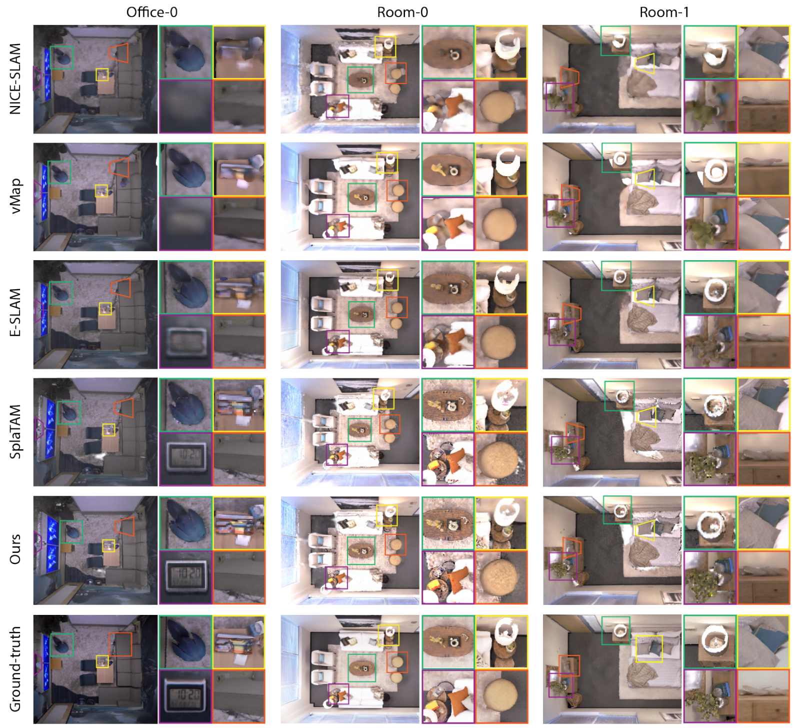

Fig. 2: Qualitative comparison of our method and the baselines for reconstruction across three scenes from the Replica Dataset [35], with key details accentuated using colored boxes. The results demonstrate that our method delivers more high-fidelity and robust reconstructions.

Baselines We compare the tracking and mapping with state-of-the-art methods NICE-SLAM [48], vMap [19], Co-SLAM [39], ESLAM [16], and SplaTAM [17]. For semantic segmentation accuracy, we compare with NIDS-SLAM [12], DNS-SLAM [20], and SNI-SLAM [47].

4.2 Evaluation of Mapping and Localization

We show quantitative measures of reconstruction quality using the Replica dataset [35] in Tab. 1. Our method demonstrates state-of-the-art performance. Compared to other baseline methods, our approach attains notably superior outcomes, outperforming them by a margin of 10dB in PSNR.

In Fig. 2, we present the reconstruction results of three chosen scenes, where regions of interest are accentuated with boxes in various colors. Our method exhibits high-fidelity reconstruction outcomes. Specifically, for small, intricately textured objects like a clock, socket, books on a tea table, and a lamp, our approach shows remarkable accuracy over NeRF-based methods. This is because

Table 1: Quantitative comparison of our method and the baselines in training view rendering on the Replica dataset [35]. Our method demonstrates SOTA performances in most cases among three metrics.

Methods	Metrics	Avg.	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4
NICE-SLAM	PSNR↑	24.42	22.12	22.47	24.52	29.07	30.34	19.66	22.23	24.94
	SSIM↑	0.809	0.689	0.757	0.814	0.874	0.886	0.797	0.801	0.856
	LPIPS↓	0.233	0.330	0.271	0.208	0.229	0.181	0.235	0.209	0.198
Co-SLAM	PSNR↑	30.24	27.27	28.45	29.06	34.14	34.87	28.43	28.76	30.91
	SSIM↑	0.939	0.910	0.909	0.932	0.961	0.969	0.938	0.941	0.955
	LPIPS↓	0.252	0.324	0.294	0.266	0.209	0.196	0.258	0.229	0.236
ESLAM	PSNR↑	29.08	25.32	27.77	29.08	33.71	30.20	28.09	28.77	29.71
	SSIM↑	0.929	0.875	0.902	0.932	0.960	0.923	0.943	0.948	0.945
	LPIPS↓	0.336	0.313	0.298	0.248	0.184	0.228	0.241	0.196	0.204
SplaTAM	PSNR↑	33.98	32.48	33.72	34.96	38.34	39.04	31.90	29.70	31.68
	SSIM↑	0.969	0.975	0.970	**0.982**	0.982	**0.982**	0.965	0.950	0.946
	LPIPS↓	0.099	0.072	0.096	0.074	**0.083**	0.093	**0.100**	0.118	0.155
Ours	PSNR↑	**34.66**	**32.50**	**34.25**	**35.10**	**38.54**	**39.20**	**32.90**	**32.05**	**32.75**
	SSIM↑	**0.973**	**0.976**	**0.978**	0.981	**0.984**	0.980	**0.967**	**0.966**	**0.949**
	LPIPS↓	**0.096**	**0.070**	**0.094**	**0.070**	0.086	**0.087**	0.101	**0.115**	**0.148**

Table 2: Quantitative comparison in terms of Depth L1, ATE, and FPS between our method and the baselines on the Replica dataset [35]. The values represent the average outcomes across eight scenes. The FPS is evaluated by setting the same number of training iterations for all systems for fair comparison. The results of baselines are retrieved from [47]. Our method outperforms the baselines at Depth and ATE evaluations, and performs fairly on FPS metrics. SOTA performances are highlighted.

Methods	Depth L1 [cm]↓	ATE Mean [cm]↓	ATE RMSE [cm]↓	Track. FPS [f/s]↑	Map. FPS [f/s]↑	SLAM FPS [f/s]↑
NICE-SLAM	1.903	1.795	2.503	13.70	0.20	0.20
Co-SLAM	1.513	0.935	1.059	17.24	**10.20**	**6.41**
ESLAM	1.180	0.520	0.630	**18.11**	3.62	3.02
SplaTAM	0.525	0.348	0.454	5.53	3.84	2.26
Ours	**0.356**	**0.327**	**0.412**	5.27	3.52	2.11

Gaussians are capable of representing objects with complex textures and surfaces. Furthermore, NeRF-based methods often struggle with the over-smoothing issue, resulting in blurred edges on objects. In contrast, by utilizing an explicit Gaussian representation, SGS-SLAM precisely captures objects with clear edges, irrespective of their sizes. Compared with SplaTAM [17], which is also a Gaussian-based model, our approach utilizes semantic information for discerning object categories, recognizing visual appearance to determine texture, and applying geometric constraints to preserve accurate shapes. This combination enables our method to achieve thorough modeling of both objects and their surrounding environment. The combination of these constraints allows SGS-SLAM to capture fine-grained details of objects, offering high-fidelity and accurate reconstruction.

Tab. 2 displays the tracking evaluation results on the Replica dataset [35]. Our method excels in achieving the highest level of depth L1 loss (cm) and minimal ATE error, surpassing baseline methods by 70% in terms of depth loss and 34% in terms of ATE RMSE (cm). This exceptional performance can be attributed to our precise scene reconstruction, which provides finely-detailed rendering results. The high-quality rendering, in turn, contributes to accurate camera pose estimation based on the established map by preventing incorrect geometric reconstruction, which could otherwise result in inaccurate tracking outcomes. Additionally, utilizing features from different channels of Gaussians, such as geometry, appearance, and semantic information, provides multiple levels of supervision, resulting in a more robust and accurate tracking capability.

4.3 Evaluation of Semantic Segmentation

Tab. 3 shows a quantitative evaluation of our method in comparison to other neural semantic SLAM approaches. It's worth noting that we only show four scenes because previous NeRF-based semantic models only reported results on these scenes. In comparison to these previous methods, SGS-SLAM demonstrates state-of-the-art performance, outperforming the initial baseline by more than 10%. Substantial enhancement highlights the crucial advantage of explicit Gaussian representation over NeRF-based approaches. Gaussians can precisely isolate object boundaries, resulting in highly accurate 3D scene segmentation. In contrast, NeRF-based methods often struggle to recognize individual objects and typically require complex muti-level model designs and extensive feature fusion. Our approach offers an unparalleled ability to identify 3D objects in decomposed representations, which can serve as 3D priors for tracking and mapping in future time steps, and is well-suited for further downstream tasks.

Table 3: Quantitative comparison of our method against existing semantic NeRF-based SLAM methods on the Replica dataset [35]. The baselines are limited to four scenes as their results are reported only for these. For each scene, we compute the average mIoU score by comparing the rendered and the ground-truth 2D semantic image in the training view. Our method significantly outperforms the NeRF-based approaches, achieving SOTA mIoU scores over 90%.

Methods	**Avg. mIoU↑**	Room0 [%]↑	Room1 [%]↑	Room2 [%]↑	Office0 [%]↑
NIDS-SLAM	82.37	82.45	84.08	76.99	85.94
DNS-SLAM	84.77	88.32	84.90	81.20	84.66
SNI-SLAM	87.41	88.42	87.43	86.16	87.63
Ours	**92.72**	**92.95**	**92.91**	**92.10**	**92.90**

4.4 Evaluation of Keyframe Optimization

In real-world datasets, tracking errors tend to accumulate along a trajectory, making pose estimations at later timestamps less reliable. Such inaccuracies can compromise the quality of map reconstructions, negatively impacting the previously well-established scenes. A case in point is scene0000 from the ScanNet dataset [4], where objects such as bike and guitar are revisited at early and late stages in the trajectory. Keyframes from later sequences, influenced by inaccurate camera poses, can disrupt the previously accurate reconstructions. Fig. 3

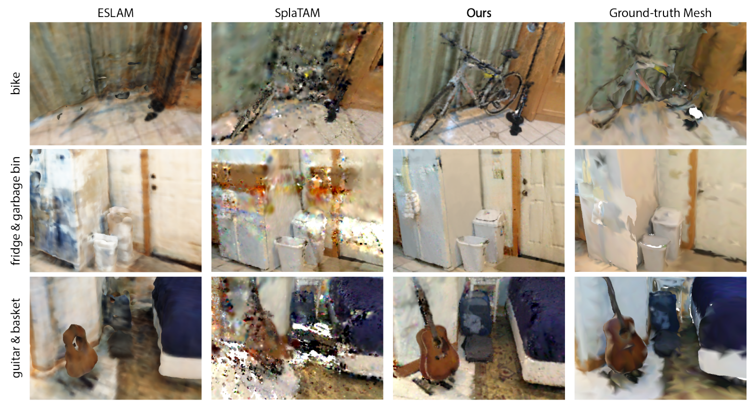

Fig. 3: The selected novel view synthesis of scene0000 from the ScanNet dataset [4]. The rendered views display the reconstructed objects such as bike, fridge, garbage bin, and guitar from novel views. Our method outperforms baselines by a large margin primarily due to the integration of keyframe optimization and semantic constraints. Note that the ground-truth for novel views is captured from the offline-reconstructed mesh provided by the ScanNet dataset.

illustrates the novel view evaluation for scene0000. In comparison to ESLAM [16] and SplaTAM [17], which are based on NeRF and 3D Gaussians, our method delivers more accurate reconstruction outcomes. The bike, garbage bin, and guitar are accurately rendered, meanwhile details are preserved. Our method facilitates the selection of keyframes based on geometric and semantic constraints, incorporating uncertainty weighting during the optimization of selected keyframes. This strategy demonstrates its effectiveness in map optimization from different views meanwhile preventing the unreliable keyframse with high uncertainty to significantly altering the earlier accurately reconstructed map.

4.5 Scene Manipulation

The obtained semantic mask within the 3D scene has a range of applications for subsequent tasks. As an illustrative example, we demonstrate a straightforward but efficient Gaussian editing method defined by Eq. (10), which is crucial for enabling scene manipulation for robotics or mixed reality applications.

Utilizing the decoupled scene representation, in contrast to NeRF-based approaches that demand fine-tuning of the entire network, SGS-SLAM can edit specific objects within the scene while keeping the remainder of the well-trained, irrelevant environment fixed. As shown in Fig. 4, we can directly manipulate the Gaussians associated with the editing target, such as erasing, moving, and rotating the jar and flowers on the table. In addition, we can group objects by selecting their semantic masks and applying a transition, such as rotating both the table and the above objects as shown in the supplementary material. This

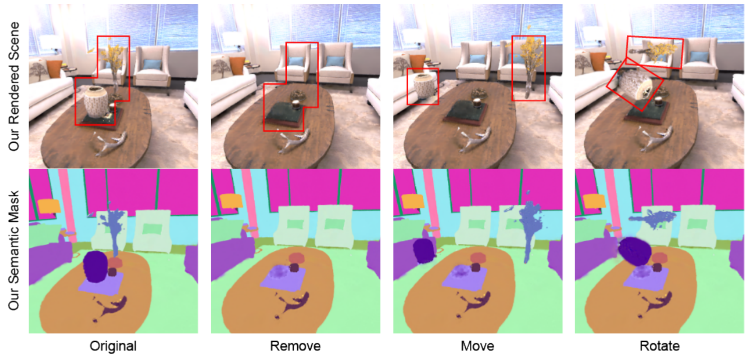

Fig. 4: The case study on scene manipulation in room0 of the Replica dataset [35]. We show the capabilities for object removal and transformation by specifying semantic labels. SGS-SLAM allows manipulation of either individual objects or a group of items, as illustrated by actions that include the removal of a jar and flowers, as well as moving and rotating them.

editing capability requires no training or fine-tuning, making it readily available for downstream applications.

4.6 Ablation Study

We perform the ablation of SGS-SLAM on the scene0000_00 of the ScanNet dataset [4] to evaluate the effectiveness of multi-channel feature supervision, and the keyframe optimization strategies.

Table 4: Ablation study of multi-channel optimization on the scene0000_00 of the ScanNet dataset [4]. The comparison involves settings where appearance, depth, and semantic supervision are removed. ✗ means the metric is inapplicable.

Settings	Depth L1 [cm]↓	ATE RMSE [cm]↓	PSNR [dB]↑	mIoU [%]↑
without color image (C_{pix})	7.44	24.59	✗	68.19
without depth map (D_{pix})	47.66	40.47	15.14	54.52
without semantic map (S_{pix})	9.15	13.81	17.52	✗
without silhouette threshold (Sil_{pix})	29.12	357.48	12.06	28.07
with multi-channel optimization	**6.18**	**11.26**	**19.47**	**70.27**

Effect of Multi-channel Optimization Tab. 4 shows the ablation study on multi-channel parameter optimization. The results reveal that our optimization strategy can significantly improve the localization and mapping performance. Specifically, the system without appearance color cannot provide rendered views, whereas camera pose and depth can still be estimated by leveraging depth and

Table 5: Ablation study of keyframe optimization on the scene0000 of the ScanNet dataset [4]. The comparison involves settings where geometric, semantic, and uncertainty constraints are removed.

Settings	Depth L1 [cm]↓	ATE RMSE [cm]↓	PSNR [dB]↑	mIoU [%]↑
without geometric threshold (T_{geo})	6.66	15.55	19.21	68.93
without semantic threshold (T_{sem})	8.44	12.89	17.84	69.85
without uncertainty weighting ($\mathcal{U}$)	6.87	11.43	18.72	70.12
with keyframe selection	**6.18**	**11.26**	**19.47**	**70.27**

semantic input. The absence of depth data leads to the poorest depth estimation, highlighting the importance of geometric supervision. Furthermore, the absence of an input semantic map disables 3D semantic segmentation and remarkably diminishes the performance of tracking and mapping. Additionally, the silhouette threshold, essential for assessing scene visibility, is crucial for the system stability. Without this threshold, the system shows a significant decline in the effectiveness of tracking and mapping.

Effect of Keyframe Optimization Tab. 5 presents the results of keyframe selection ablation. Our two-level keyframe selection strategy reveals that omitting either geometric or semantic constraints results in a significant drop in both tracking and mapping performance. Additionally, without incorporating uncertainty weighting, the system demonstrates a decrease in performance compared to its full implementation.

5 Conclusion and Limitations

We presented SGS-SLAM, the first semantic dense visual SLAM system based on the 3D Gaussian representation. We propose to leverage multi-channel parameter optimization where appearance, geometric, and semantic constraints are combined to enforce high-accurate 3D semantic segmentation, and high-fidelity dense map reconstruction meanwhile effectively producing a robust camera pose estimation. SGS-SLAM takes advantage of optimal keyframe optimization, resulting in reliable reconstruction quality. Extensive experiments show that our method provides state-of-the-art tracking and mapping results, meanwhile maintaining rapid rendering speeds. Furthermore, the high-quality reconstruction of scenes and precise 3D semantic labeling generated by our system establish a strong foundation for downstream tasks such as scene editing, offering solid prior for robotics or mixed reality applications.

Limitations SGS-SLAM replies on depth and 2D semantic signal inputs for tracking and mapping. In scenarios where this information is scarce or difficult to access, the system's effectiveness will be compromised. Additionally, our method incurs large memory consumption when deployed to large scenes. Addressing these limitations will be an objective for future research.

References

1. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.J.: Codeslam—learning a compact, optimisable representation for dense visual slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2560–2568 (2018) 1, 3
2. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics **37**(6), 1874–1890 (2021) 4
3. Chung, C.M., Tseng, Y.C., Hsu, Y.C., Shi, X.Q., Hua, Y.H., Yeh, J.F., Chen, W.C., Chen, Y.T., Hsu, W.H.: Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 9400–9406. IEEE (2023) 4
4. 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) 8, 11, 12, 13, 14
5. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Transactions on Graphics (ToG) **36**(4), 1 (2017) 8
6. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. IEEE transactions on pattern analysis and machine intelligence **29**(6), 1052–1067 (2007) 1
7. Deng, T., Chen, Y., Zhang, L., Yang, J., Yuan, S., Wang, D., Chen, W.: Compact 3d gaussian splatting for dense visual slam. arXiv preprint arXiv:2403.11247 (2024) 4
8. Deng, T., Shen, G., Qin, T., Wang, J., Zhao, W., Wang, J., Wang, D., Chen, W.: Plgslam: Progressive neural scene represenation with local to global bundle adjustment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19657–19666 (2024) 1
9. Deng, T., Wang, N., Wang, C., Yuan, S., Wang, J., Wang, D., Chen, W.: Incremental joint learning of depth, pose and implicit scene representation on monocular camera in large-scale scenes. arXiv preprint arXiv:2404.06050 (2024) 4
10. Deng, T., Wang, Y., Xie, H., Wang, H., Wang, J., Wang, D., Chen, W.: Neslam: Neural implicit mapping and self-supervised feature tracking with depth completion and denoising. arXiv preprint arXiv:2403.20034 (2024) 4
11. Deng, T., Xie, H., Wang, J., Chen, W.: Long-term visual simultaneous localization and mapping: Using a bayesian persistence filter-based global map prediction. IEEE Robotics & Automation Magazine **30**(1), 36–49 (2023) 4
12. Haghighi, Y., Kumar, S., Thiran, J.P., Van Gool, L.: Neural implicit dense semantic slam. arXiv preprint arXiv:2304.14560 (2023) 4, 9
13. He, J., Li, M., Wang, Y., Wang, H.: Ovd-slam: An online visual slam for dynamic environments. IEEE Sensors Journal (2023) 3
14. Hermans, A., Floros, G., Leibe, B.: Dense 3d semantic mapping of indoor scenes from rgb-d images. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 2631–2638. IEEE (2014) 3
15. Huang, H., Li, L., Cheng, H., Yeung, S.K.: Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras. arXiv preprint arXiv:2311.16728 (2023) 4

16. 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, 8, 9, 12
17. 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) 2, 3, 4, 5, 8, 9, 10, 12
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023) 2, 4, 7
19. Kong, X., Liu, S., Taher, M., Davison, A.J.: vmap: Vectorised object mapping for neural field slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 952–961 (2023) 1, 9
20. Li, K., Niemeyer, M., Navab, N., Tombari, F.: Dns slam: Dense neural semantic-informed slam. arXiv preprint arXiv:2312.00204 (2023) 1, 2, 4, 8, 9
21. Li, M., He, J., Jiang, G., Wang, H.: Ddn-slam: Real-time dense dynamic neural implicit slam with joint semantic encoding. arXiv preprint arXiv:2401.01545 (2024) 4
22. Li, M., He, J., Wang, Y., Wang, H.: End-to-end rgb-d slam with multi-mlps dense neural implicit representations. IEEE Robotics and Automation Letters **8**(11), 7138–7145 (2023) 4
23. Liu, S., Zhou, H., Li, L., Liu, Y., Deng, T., Zhou, Y., Li, M.: Structure gaussian slam with manhattan world hypothesis. arXiv preprint arXiv:2405.20031 (2024) 4
24. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: 3DV (2024) 5
25. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18039–18048 (2024) 4
26. 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) 4
27. 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
28. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022) 4
29. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015) 1, 3
30. Narita, G., Seno, T., Ishikawa, T., Kaji, Y.: Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4205–4212. IEEE (2019) 3
31. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: 2011 international conference on computer vision. pp. 2320–2327. IEEE (2011) 1
32. Qin, T., Li, P., Shen, S.: Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics **34**(4), 1004–1020 (2018) 3

33. 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) 3
34. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H., Davison, A.J.: Slam++: Simultaneous localisation and mapping at the level of objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1352–1359 (2013) 1, 3
35. 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, 9, 10, 11, 13
36. 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) 1, 4
37. Sucar, E., Wada, K., Davison, A.: Nodeslam: Neural object descriptors for multi-view shape reconstruction. In: 2020 International Conference on 3D Vision (3DV). pp. 949–958. IEEE (2020) 1
38. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. Advances in neural information processing systems **34**, 16558–16569 (2021) 4
39. 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) 1, 3, 4, 9
40. Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., Davison, A.: Elasticfusion: Dense slam without a pose graph. In: Proceedings of Robotics: Science and Systems. Robotics: Science and Systems (2015) 3
41. 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) 4, 8
42. 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
43. 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) 8
44. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting. arXiv preprint arXiv:2312.10070 (2023) 4
45. Zhang, Y., Tosi, F., Mattoccia, S., Poggi, M.: Go-slam: Global optimization for consistent 3d instant reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3727–3737 (2023) 1, 4
46. Zhou, H., Guo, Z., Liu, S., Zhang, L., Wang, Q., Ren, Y., Li, M.: Mod-slam: Monocular dense mapping for unbounded 3d scene reconstruction. arXiv preprint arXiv:2402.03762 (2024) 4
47. 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) 2, 4, 8, 9, 10
48. 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) 1, 3, 4, 8, 9