SLC^2-SLAM: Semantic-Guided Loop Closure Using Shared Latent Code for NeRF SLAM

Yuhang Ming, *Member, IEEE*, Di Ma, Weichen Dai, Han Yang, Rui Fan, *Senior Member, IEEE*, Guofeng Zhang, and Wanzeng Kong, *Senior Member, IEEE*

***Abstract*—Targeting the notorious cumulative drift errors in NeRF SLAM, we propose a Semantic-guided Loop Closure using Shared Latent Code, dubbed SLC^2-SLAM. We argue that latent codes stored in many NeRF SLAM systems are not fully exploited, as they are only used for better reconstruction. In this letter, we propose a simple yet effective way to detect potential loops using the same latent codes as local features. To further improve the loop detection performance, we use the semantic information, which are also decoded from the same latent codes to guide the aggregation of local features. Finally, with the potential loops detected, we close them with a graph optimization followed by bundle adjustment to refine both the estimated poses and the reconstructed scene. To evaluate the performance of our SLC^2-SLAM, we conduct extensive experiments on Replica and ScanNet datasets. Our proposed semantic-guided loop closure significantly outperforms the pre-trained NetVLAD and ORB combined with Bag-of-Words, which are used in all the other NeRF SLAM with loop closure. As a result, our SLC^2-SLAM also demonstrated better tracking and reconstruction performance, especially in larger scenes with more loops, like ScanNet.**

***Index Terms*—SLAM, loop detection, localization, semantic scene understanding.**

I. Introduction

USING a RGB-D camera as the primary sensor, dense simultaneous localization and mapping (SLAM) aims at estimating the self-motion, *i.e.* poses, of an agent while recovering the dense 3D reconstruction of its surrounding environment. Dense SLAM is the core to a wide range of spatial artificial intelligence (AI) applications, including autonomous robots and systems, embodied AI, and metaverse applications. Thus, it has been a popular research area in the robotics and computer vision communities.

Over the past decade, the field has seen remarkable advancements in dense SLAM, alongside a growing integration of SLAM systems with neural networks. Early dense SLAM systems, such as KinectFusion [3] and ElasticFusion [4], prioritized precise geometrical reconstructions of environments, enabling detailed spatial modeling. Then, incorporating pre-trained neural networks, dense SLAM systems have evolved to provide enhanced scene comprehension [5] and increased resilience against cumulative drift errors [6]. This synergy has expanded the scope of SLAM, transforming it from purely geometric mapping to a more semantically aware, robust system capable of more accurate and stable performance in complex environments.

More recently, the introduction of neural radiance fields (NeRF) [7] has showcased the powerful scene representation capabilities of multi-layer perceptrons (MLP). By encoding 3D scenes implicitly within the weights of an MLP, it generates compact neural implicit maps, which not only reduce the storage requirements of large-scale reconstructed scenes but also allow for efficient bundle adjustment of both estimated poses and the reconstructed map. Due to these advantages, NeRF has garnered substantial interest for developing dense SLAM systems that leverage neural implicit representations [8]. Pioneered by iMAP [9] and NICE-SLAM [10], a series of NeRF SLAM systems have emerged, showing notable advances in reconstruction quality [11], tracking precision [12], and overall system efficiency [2]. These developments represent a promising shift toward more accurate, storage-efficient, and computationally feasible dense SLAM solutions.

Comparatively, much less attention has been paid to address the cumulative drift errors in NeRF SLAM systems. Existing implementations of loop closure in NeRF SLAM generally follow one of three main approaches: (1) employing handcrafted local features with global descriptor aggregation, such as ORB features [13] paired with Bag-of-Words (BoW) descriptors [14]; (2) utilizing pre-trained place recognition models, like NetVLAD [15]; and (3) applying a simple covisibility score-based method. However, none of these techniques offer an optimal solution for NeRF SLAM. Covisibility score-based methods are too simple to close the loops with large drifts, while the other approaches require additional efforts for local feature extraction. These added steps not only increase computational overhead but also risk losing relevant information unique to NeRF representations, highlighting the need for a more specialized loop closure strategy tailored to the NeRF SLAM framework.

Received 22 February 2025; accepted 16 March 2025. Date of publication 20 March 2025; date of current version 10 April 2025. This article was recommended for publication by Associate Editor T. Fischer and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by National Natural Science Foundation of China under Grant 62401188 and Grant 62473288, in part by the Fundamental Research Funds for the Central Universities, in part by NIO University Programme (NIO UP), and in part by Xiaomi Young Talents Program. *(Corresponding author: Wanzeng Kong.)*

Yuhang Ming, Di Ma, Weichen Dai, Han Yang, and Wanzeng Kong are with the School of Computer Science and Key Laboratory of Brain Machine Collaborative Intelligence of Zhejiang Province, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: yuhang.ming@hdu.edu.edu; 231050008@hdu.edu.edu; weichendai@hdu.edu.edu; yhan_hdu@hdu.edu.edu; kongwanzeng@hdu.edu.cn).

Rui Fan is with the College of Electronics and Information Engineering, Shanghai Institute of Intelligent Science and Technology, Shanghai Research Institute for Intelligent Autonomous Systems, the State Key Laboratory of Intelligent Autonomous Systems, Tongji University, Shanghai 201804, China, and also with the Frontiers Science Center for Intelligent Autonomous Systems, Tongji University, Shanghai 201804, China (e-mail: rui.fan@ieee.org).

Guofeng Zhang is with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China (e-mail: zhangguofeng@zju.edu.cn).

Digital Object Identifier 10.1109/LRA.2025.3553352

Recognizing that recent NeRF SLAM systems commonly utilize InstantNGP-style [16] mapping for efficiency, where latent codes are learned on-the-fly and stored throughout operation, we observe that these latent codes' potential as local features for loop detection has been underutilized. In this letter, we introduce Semantic-guided Loop Closure using Shared Latent Code for NeRF SLAM (SLC^2-SLAM), a simple yet effective approach specifically designed to leverage these latent codes for effective loop detection within NeRF SLAM systems. In particular, our method uniquely repurposes these latent codes, initially intended solely for scene reconstruction, as local geometric features which are then aggregated into a global descriptor. To enhance this aggregation process, we incorporate semantic information, also decoded from the latent codes, guiding the selection of local latent codes for better aggregation. After identifying potential loops, we close the loop with a pose graph optimization, followed by bundle adjustment, to refine both the estimated pose and the reconstructed map.

We rigorously evaluate the performance of our SLC^2-SLAM with extensive experiments on Replica [17] and ScanNet [1] datasets. Our method shows significant improvement in loop detection capabilities, achieving an average recall rate of 0.662—outperforming the closest competing approach, which achieves only 0.277. This enhanced loop detection also contributes to superior tracking accuracy and reconstruction quality, particularly evident in the larger scenes from the ScanNet dataset, as illustrated in Fig. 1.

Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to exploit the latent codes stored in many NeRF SLAM system not only for scene reconstruction but also for semantic segmentation and loop detection.
- We conduct extensive experiments on publicly available datasets and our SLC^2-SLAM consistently outperforms existing methods, achieving state-of-the-art performance in loop detection, reconstruction quality, and competitive performance in tracking accuracy.

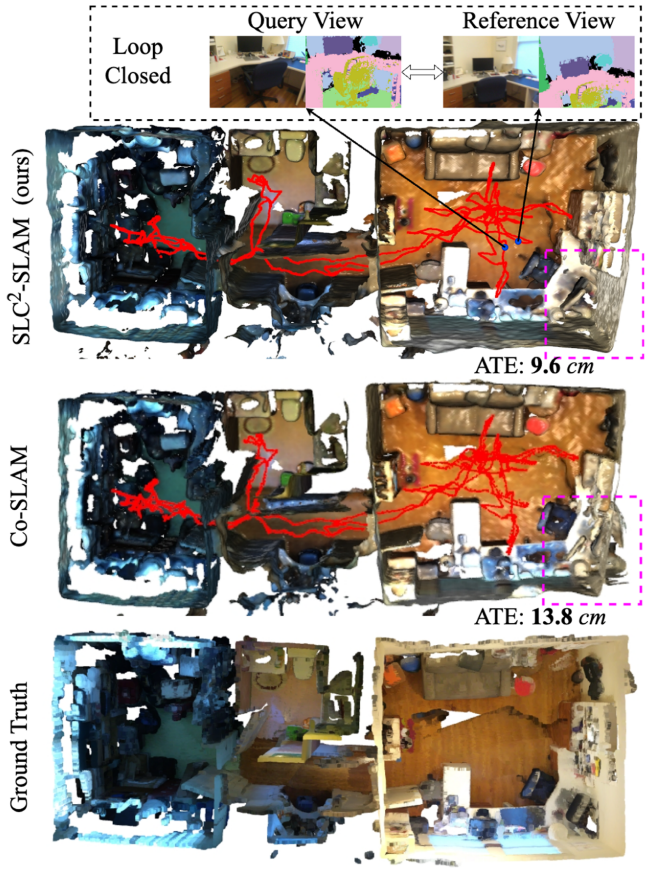

Fig. 1. Tracking and reconstruction results on the scene0054 of ScanNet [1]. With semantic-guided loop closure, our SLC^2-SLAM achieved better tracking and reconstruction performance. In contrast, our base system Co-SLAM [2] exhibits obvious misalignment, especially evident in the areas in the pink bounding boxes.

II. Related Work

A. NeRF SLAM With Latent Codes

To enhance the reconstruction quality of NeRF SLAM systems, many approaches leverage latent codes to capture local scene structures, reducing the burden on the MLP for detailed map representation. While various terms such as features, embeddings, or latent codes are used across the literature, we refer to them collectively as latent codes here for consistency.

Vox-Fusion [11] pioneered the integration of neural implicit maps with explicit voxel structures by attaching on-the-fly learned latent codes to voxel vertices and utilizing an octree for efficient voxel indexing. Similar concepts also appear in systems like Co-SLAM [2], ESLAM [18], and VPE-SLAM [19]. Both Co-SLAM and VPE-SLAM followed voxel representations, but with distinct encoding design. Co-SLAM [2] builds on the InstantNGP [16] framework, introducing a joint coordinate and parametric encoding with multi-resolution hashing and One-blob encoding. VPE-SLAM, alternatively, presents a voxel-permutohedral encoding that merges sparse voxels with multi-resolution permutohedral tetrahedral. Contrasting with voxel-centric approaches, ESLAM [18] specifically favors a plane-based representation to retain latent codes.

Building on this hybrid map representation, various works have been published to improve the systems' performance from different aspects. For a richer scene understanding, both SNI-SLAM [20] and NIS-SLAM [21] expand NeRF SLAM by generating semantic maps, allowing for detailed scene labeling. Regarding the accuracy of the reconstructed geometry, Hu et al. [22] address issues related to incomplete depth data by introducing attentive depth fusion priors into the volume rendering process. In terms of robustness, HERO-SLAM [23] tackles abrupt viewpoint changes by implementing a hybrid enhanced robust optimization, while RoDyn-SLAM [24] improves dynamic object handling by removing dynamic rays from the reconstruction process with motion masks generated from optical flow and semantic information.

B. NeRF SLAM With Loop Closures

Loop closures are necessary to all SLAM systems to ensure robust operation in larger-scale environments. As outlined in the previous section, current loop closure approaches typically use one of three strategies: (1) covisibility scores, (2) the pre-trained NetVLAD [15] model, or (3) ORB [13] features in combination with BoW [14] descriptors. Additionally, NeRF SLAM systems can be broadly categorized by their approach to camera pose estimation: *Coupled* NeRF SLAM, which estimates poses directly through inverse NeRF optimization, and *Decoupled*

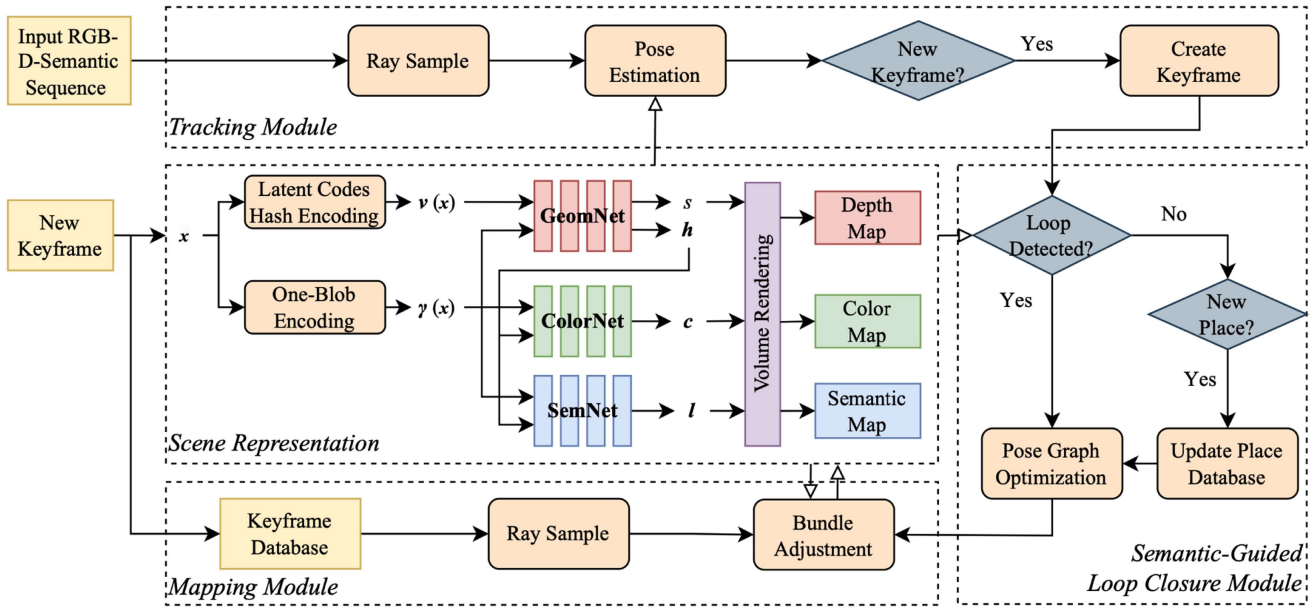

Fig. 2. System Overview. Our proposed SLC^2-SLAM consists of four main components. At its core, there is a hybrid scene representation with latent code voxel hashing and three MLPs. Then, we have tracking module, mapping module, and semantic-guided loop closure module that interact with the hybrid scene representation to perform tracking, mapping, and loop closure.

NeRF SLAM, which leverages external SLAM systems for tracking.

For *Decoupled* NeRF SLAM systems, their choice of loop closure heavily rely on the type of tracker utilized. When employing DROID-SLAM [25], as seen in systems like Go-SLAM [26] and HI-SLAM [27], the covisibility score—derived from the mean rigid flow—becomes the preferred option for loop detection due to its compatibility with DROID-SLAM's tracking mechanism. Alternatively, ORB-SLAM [28], [29] is also widely used, featuring in systems such as Orbeez-SLAM [30], NEWTON [31], NGEL-SLAM [32], and the system by Bruns et al. [33]. These systems inherit ORB-SLAM's loop closure capabilities, relying on ORB [13] features paired with BoW [14] descriptors for robust loop detection. Despite their strong tracking performance, these systems frequently encounter challenges in achieving high-quality reconstructions, as their focus on loop closure methods does not fully address limitations in fine-grained scene detail.

In *Coupled* NeRF SLAM systems, where tracking and reconstruction are tightly integrated, various approaches have been explored for loop closure. For instance, MIPS-Fusion [34] introduced multi-implicit-submaps and performed submap-level loop closure by computing the covisibility between current frame and inactive submaps. However, this loop detection approach has limitations, primarily being effective for correcting only small drifts. Vox-Fusion++[35] instead relied on a pre-trained NetVLAD [15] model for loop detection and implemented a hierarchical pose optimization for robust loop closure. Similarly, Gaussian splatting SLAM systems such as GLC-SLAM [36] and LoopSplat [37] also employed NetVLAD for loop detection. Another approach, Loopy-SLAM [38] favors the combination of ORB [13] features and BoW [14] descriptors for loop detection, despite that these features were not part of the tracking or reconstruction process. It can be seen that all these systems require additional feature extraction steps to achieve loop closure.

We argue that existing NeRF SLAM systems have not fully leveraged the latent codes inherent in their maps. By focusing solely on using these codes for reconstruction, they overlook the valuable potential of these latent codes in aiding loop detection directly, an oversight that our proposed approach seeks to address.

III. System Design

As shown in Fig. 2, our SLC^2-SLAM comprises 4 components. At its core, we use a hybrid *scene representation* with voxel-centric latent codes and three shallow MLPs. Interacting with this scene representation, we have a *tracking module* that estimates the 6 °-of-free (DoF) poses of the input frame, a *mapping module* that is in charge of the keyframe management and scene representation optimization, and a *semantic-guided loop closure module* that detects loops by aggregating on-the-fly learned latent codes and closes them with pose graph optimization.

A. Hybrid Scene Representation

Although our SLC^2-SLAM is able to work with any NeRF SLAM systems with latent codes, as we have reviewed above, we base our system on the Co-SLAM [2], modify it to incorporate semantic information, and carry out all the experiments.

Following Co-SLAM, we use a sparse set of voxels, that are indexed by a hash table, with a learnable compact latent code attached to each voxel center/vertices for coordinate encoding, and employ OneBlob encoding for parametric encoding. Regarding the shallow MLPs, our scene representation contains three: the GeomNet, ColorNet and SemNet. In particular, the GeomNet

takes in the latent code $\boldsymbol{v}(\boldsymbol{x})$ and the parametrically encoded position $\gamma(\boldsymbol{x})$, and outputs the scene geometry s, in terms of signed distance function (SDF), and a hidden feature vector $\boldsymbol{h}$. Connected to the GeomNet, ColorNet and SemMet are placed in parallel, both of which take the hidden feature vector $\boldsymbol{h}$ and the parametrically encoded position $\gamma(\boldsymbol{x})$ as input, and produce the RGB color $\boldsymbol{c}$ and semantic label l respectively.

Then, given the input RGB-D frames and semantic masks, both the latent codes and the weights of the three MLPs can be learned with the following loss function:

$$\mathcal{L} = \lambda_{rgb}\mathcal{L}_{rgb} + \lambda_d\mathcal{L}_d + \lambda_{sdf}\mathcal{L}_{sdf} + \lambda_{fs}\mathcal{L}_{fs} + \lambda_{sem}\mathcal{L}_{sem} + \lambda_{smooth}\mathcal{L}_{smooth}, \quad (1)$$

where each λ represents a per-loss weight. $\mathcal{L}_{rgb}$, $\mathcal{L}_d$, $\mathcal{L}_{sdf}$, $\mathcal{L}_{fs}$, and $\mathcal{L}_{smooth}$ denote the loss terms for color, depth, SDF, free space, and smooth regularization, respectively, following the formulation in Co-SLAM [2]. The remaining, $\mathcal{L}_{sem}$, computes the cross-entropy loss for the semantic labels as follow:

$$\mathcal{L}_{sem} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} l_{i,j}\log(\hat{l}_{i,j}). \quad (2)$$

B. Keyframe Management

To strike a balance between map update frequency, loop detection efficiency, and runtime performance, our SLC^2-SLAM introduces a hierarchical keyframe management strategy, consisting of keyframes, covisible frames, and place frames.

For keyframes, we follow Co-SLAM [2] and add a new keyframe every 5 frames. This high rate of keyframe addition enables frequent map optimization iterations, ensuring high reconstruction quality. However, this density of keyframes introduces redundancy, which can be inefficient for loop detection and pose graph optimization. To address this, we introduce place frames, a sparser subset of frames within the keyframe set, to streamline the loop detection process and enhance the efficiency of pose graph optimization.

We determine whether a keyframe should be added as a new place frame based on point cloud overlap. Specifically, we calculate the overlap between the point cloud from the most recent keyframe and those from all previously stored place frames. If the overlap is below a user-defined threshold, τ_{place}, the keyframe is accepted as a new place frame. In practice, setting τ_{place} to a relatively low value ensures minimal overlap, resulting in only a few place frames per indoor room, which efficiently covers the scene with a reduced number of frames.

In contrast to the spatially dense keyframes, we find that, in practice, place frames are too sparse to effectively distribute accumulative errors detected during loop closures. To balance between these extremes, we further introduce covisible frames, which have an intermediate spatial density. This density is also managed by using point cloud overlap, but with a higher threshold, τ_{covis}, than that of place frames. Importantly, all place frames are also designated as covisible frames. When a loop is detected at a keyframe, the keyframe, along with all covisible frames, are used to construct the pose graph for optimization. The optimization process will be discussed in detail in Section III-D.

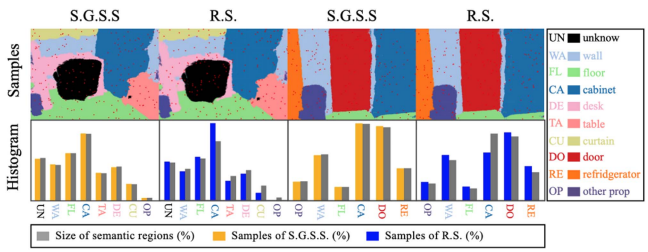

Fig. 3. Examples of semantic-guided stratified sampling (S.G.S.S) versus random sampling (R.S.).

C. Semantic-Guided Loop Detection

To perform loop detection, we formulate it as a retrieval task and solve it with a two-step process: generating global descriptors from local features and matching these descriptors with those in a database of known locations.

Unlike previous loop detection methods relying on handcrafted point features [13], [38] or neural features extracted by pre-trained convolutional neural networks [15], [35], we propose directly leveraging the latent codes stored within the NeRF SLAM map as local features. Since these latent codes are shared across tracking, mapping, and semantic segmentation tasks, this approach not only removes the need for external feature extractors but also enhances the overall efficiency of the system.

Given the high resolution of the input images, aggregating latent codes for every pixel is computationally intractable. Therefore, we aim to select M representative pixels that best describe the image. To achieve this, we introduce a semantic-guided stratified sampling method that utilizes the semantic masks predicted by SemNet and sets the number of samples to be proportional to the size of each semantic region. Although naive random sampling could be used, Fig. 3 illustrates that semantic-guided stratified sampling provides a more accurate view representation by identifying the easily-overlooked small semantic regions and reducing oversampling of the dominant semantic region. The results in Table I further demonstrate the advantages of our semantic-guided stratified sampling over random sampling.

After gathering latent codes to represent local features, we apply the vector of locally aggregated descriptors (VLAD) [39] to construct global descriptors for the current keyframe and all stored place frames. We then match these descriptors to identify the closest place frame, forming a loop hypothesis for the current keyframe.

To prevent catastrophic system failures caused by incorrect loop closures, we subject each loop hypothesis to further validation using both geometric and semantic information. Specifically, we calculate the overlap between the point clouds and their semantic labels. The loop hypothesis is only accepted if both overlaps exceed pre-set thresholds.

D. Pose Graph Optimization

Given a pose graph comprising the covisible frames discussed in the previous section, the current keyframe as the loop frame, and its matched place frame, we can now incorporate a loop edge into the pose graph. This loop edge represents the relative transformation between the loop frame and its matched place frame, calculated using a standard point-to-plane iterative closest point (ICP) [40] algorithm.

TABLE I
LOOP DETECTION RESULTS ON SCANNET

Methods	Metric	scene0000	scene0059	scene0106	scene0169	scene0181	scene0207	Avg.
NetVLAD [15]	**Precision↑**	0.041	0.056	0.089	0.089	0.046	0.067	0.063
	F1-score↑	0.028	0.083	0.108	0.114	0.085	0.121	0.084
	recall@1↑	0.022	0.158	0.136	0.158	0.597	**0.635**	0.277
ORB [13] + BoW [14]	**Precision↑**	0.175	**0.246**	**0.645**	0.298	0.273	0.335	0.329
	F1-score↑	0.229	**0.494**	0.142	0.330	0.292	0.344	0.305
	recall@1↑	0.331	0.167	0.080	0.369	0.314	0.353	0.269
Ours (w/o semantic)	**Precision↑**	0.306	0.147	0.377	**0.547**	**0.328**	**0.493**	**0.366**
	F1-score↑	0.336	0.182	0.462	**0.605**	0.352	0.506	0.407
	recall@1↑	0.371	0.238	0.597	0.677	0.378	0.520	0.464
Ours	**Precision↑**	**0.317**	0.244	0.379	0.320	0.286	0.461	0.335
	F1-score↑	**0.414**	0.325	**0.492**	0.462	**0.416**	**0.521**	**0.438**
	recall@1↑	**0.598**	**0.489**	**0.701**	**0.828**	**0.756**	0.598	**0.662**

The best and second best results are marked with bold and underline.

TABLE II
AVERAGE RECALL RATE

Methods	Recall@1	Recall@2	Recall@3
NetVLAD [15]	0.113	0.164	0.206
ORB [13] + BoW [14]	0.165	0.195	0.220
AnyLoc [46]	0.208	0.284	0.327
SALAD [47]	0.322	0.417	0.462
Ours	**0.596**	**0.664**	**0.689**

The best and second best results are marked with bold and underline.

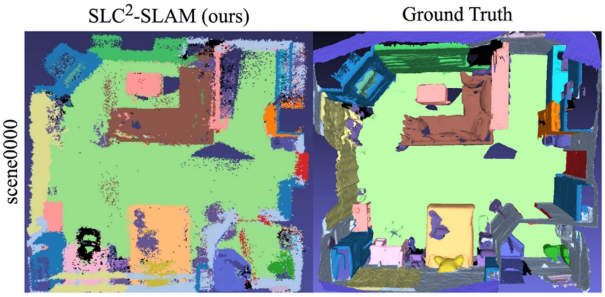

Fig. 4. Semantic segmentation examples on ScanNet.

TABLE III
TRACKING RESULTS ON SCANNET (ATE RMSE [CM]↓)

Methods	0000	0059	0106	0169	0181	0207	Avg.
Co-SLAM [2]	7.13	11.14	9.36	5.90	11.81	7.14	8.75
Loopy-SLAM [38][1]	**4.2**	7.5	8.3	7.5	**10.6**	7.9	7.7
Orbeez-SLAM [30]	7.22	**7.15**	8.05	6.58	12.77	7.16	8.66
MIPS-Fusion [34][1]	7.9	10.7	9.7	9.7	14.2	7.8	10.0
SplaTAM [44]	12.83	10.10	17.12	12.08	11.10	7.46	11.88
GLC-SLAM [36][1]	12.9	7.9	**6.3**	10.5	11.0	6.3	9.2
SLC²-SLAM (ours)	5.83	9.45	8.00	**5.29**	11.26	**6.10**	**7.66**

[1] Loopy-SLAM, MIPS-Fusion, and GLC-SLAM papers only reported tracking results in one decimal place.
The best and second best results are marked with bold and underline.

Once the graph is constructed, we proceed with optimization using standard pose graph toolkits. The recent release of PyPose [41] enables seamless integration of geometry-based optimization with learning-based loop detection, all within the PyTorch framework. Specifically, we employ the Levenberg-Marquardt optimizer, establishing a trust-region strategy to dynamically adjust the learning rate. For a particular edge in the pose graph, we define the loss function as follows:

$$\mathbf{e}_i = \log\left(\mathbf{T}_{\text{edge}_i}^{-1}\mathbf{T}_{\text{node}_0}\mathbf{T}_{\text{node}_1}^{-1}\right), \tag{3}$$

where $\mathbf{T}_{\text{edges}_i}$ is the relative transformation of between two frames connected by the edge, $\mathbf{T}_{\text{node0}}$ and$\mathbf{T}_{\text{node1}}$ are poses of the two frames respectively. Then, the overall loss to be optimized can be formulated as:

$$\mathcal{L}_{pg} = \sum_i \|\mathbf{e}_i\|^2. \tag{4}$$

After optimizing all covisible frames upon loop closure, we use them to update the poses of the keyframes. These updated keyframe poses are then included in an additional bundle adjustment step, as in Co-SLAM [2], to jointly refine both the keyframe poses and the map, which is represented by the MLPs and latent codes.

IV. EXPERIMENTS

A. Experiment Setup

We evaluate our proposed SLC²-SLAM on two widely-used indoor datasets: Replica [17] and ScanNetv1 [1]. Replica is a synthetic dataset containing 18 high-fidelity replicates of different indoor rooms, offering ground-truth dense reconstruction, semantic and instance annotations, among other resources. In line with other NeRF SLAM studies, we use the subset provided in NICE-SLAM [10], comprising 2000-frame sequences from 8 out of the 18 indoor rooms. ScanNet, by contrast, is a large-scale dataset of 1,513 sequences collected from 707 real-world indoor rooms. Also to align with other NeRF SLAM letters, we use scenes *0000, 0059, 0106, 0169, 0181*, and *0207* for both qualitative and quantitative evaluations.

To quantitatively assess the loop closure performance, we collect a database of place frames and a set of query frames from the sequences of the 6 ScanNet scenes. In particular, all the place frames and query frames are the keyframes generated from the sequence at intervals of every 5 frames. Then, we use the overlap between frame-pairs as the criteria and select a keyframe as a place frame if the overlap is lower than 0.3. All other keyframes are used as query frame for evaluation. In total, from the 6 ScanNet scenes, we gathered 96 place frames and 3,218 query frames.

Pushing it to the limit, we further evaluated our method on the complete ScanNetv1 [1] under the place recognition setup.

TABLE IV
TRACKING RESULTS ON REPLICA (ATE RMSE [CM]↓)

Methods	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4	Avg.
NICE-SLAM [10]	1.69	2.04	1.55	0.99	0.90	1.39	3.97	3.08	1.95
Co-SLAM [2][1]	0.77	1.04	1.09	0.58	0.53	2.05	1.49	0.84	0.99
Hu *et al.* [22]	1.39	1.55	2.60	1.09	1.23	1.61	1.61	1.42	1.81
MIPS-Fusion [34][2]	1.1	1.2	1.1	0.7	0.8	**1.3**	2.2	1.1	1.2
SLC^2-SLAM (ours)	**0.58**	**0.63**	**0.88**	**0.49**	**0.49**	1.53	**1.37**	**0.66**	**0.83**

[1] The results for Co-SLAM are generated using their official implementation;
[2] MIPS-Fusion paper only reported tracking results in one decimal place.
The best and second best results are marked with bold and underline.

Following the protocols established in recent indoor place recognition studies [42], [43], we utilized the test split of ScanNetv1 with spatial sparsification, resulting in 4,313 frames from 142 indoor rooms. From this set, we selected 294 frames as our place frames for retrieval by enforcing a minimum spatial separation of 3 meters, and used the remaining 4,019 frames as query frames. It is important to note that, compared to the loop detection setup, the place recognition setup uses significantly sparser place frames, making the task considerably more challenging.

Evaluation Metrics: To evaluate tracking and reconstruction performance, we adopt standard evaluation metrics: root mean square error (RMSE) of the absolute trajectory error (ATE) for tracking, and accuracy (Acc.), completion (Comp.), and completion ratio (Comp. Ratio) for reconstruction. Note that we follow Co-SLAM [2] to perform mesh culling before evaluation, and we refer readers to the survey letter [8] for more details on metric computation. For loop detection, a loop candidate is accepted if the translational pose difference with the place frame is less than 1 m and the rotational difference is under 35 degrees. We evaluate loop detection with three metrics—precision, recall, and F1 score—all based on top-1 retrieval results. As for place recognition, we follow the protocols used in [42], [43] and compute the average recall rates for Top-K retrievals.

Baselines: We choose Co-SLAM [2] as our primary baseline, as it serves as the foundation for our proposed SLC^2-SLAM. We then compare SLC^2-SLAM to three recent NeRF SLAM systems that support loop closure. Of particular interest are systems with a similar mapping setup, maintaining a single, global NeRF-based map. Therefore, we select Loopy-SLAM [38] and Orbeez-SLAM [30] for comparison. To broaden the scope, we also include two NeRF SLAM systems [10], [22], two loop closure-enabled NeRF SLAM systems that utilize submaps [33], [34], a Gaussian splatting SLAM [44], and a loop closure-enabled Gaussian splatting SLAM [36].

Regarding loop detection and place recognition, we compare our semantic-guided loop detection method with two commonly used approaches: NetVLAD [15] and the combination of ORB [13] and BoW [14]. Following the setup in GLC-SLAM [36] and Loopy-SLAM [38], we used the NetVLAD model pre-trained on the Pitts30 K dataset [45] and the BoW vocabulary provided by ORB-SLAM2 [28]. Although not yet integrated in SLAM systems, two state-of-the-art (SoTA) Dinov2-based models, AnyLoc [46] and SALAD [47], are also compared in the place recognition task to highlight the superiority of our method.

Implementation Details: We keep most of our system and training parameters in line with our backbone system, Co-SLAM [2]. For the additional modules, we design our SemNet as a 4-layer MLP with 32 hidden neurons, setting its learning rate 0.05, and assigning a weight of $\lambda_{sem} = 10$ for the semantic loss. Moreover, the thresholds for generating place frames and covisible frames are set to $\tau_{place} = 0.3$, $\tau_{covis} = 0.45$, respectively. When a loop candidate is validated, we perform another 10 iterations of bundle adjustment following the pose graph optimization. All of our experiments are performed on a desktop PC with AMD Ryzen 9 5950X CPU and NVIDIA GeForce RTX 4090 GPU.

B. Results and Discussions

Loop Detection: We present the quantitative results in Tables I and II, all recorded prior to the loop validation step. Our semantic-guided approach shows a significant performance improvement over these baseline and SoTA methods across all scenes and nearly all metrics in both loop detection and place recognition setups.

We also conducted an ablation study by removing semantic guidance and substituting semantic-guided stratified sampling with naive random sampling. This modification led to a noticeable performance decline in recall and F1 score; however, our method still outperformed NetVLAD [15] and the ORB [13] and BoW [14] combination.

In addition, though the semantic segmentation task is not the focus of our work but merely an assistant in the loop detection task, we provide some qualitative results shown in Fig. 4. Quantitatively, our SLC^2-SLAM achieves a mean intersection over union (mIoU) of 0.6795 on the six test scenes of ScanNet [1]. In comparison, a recent semantic SLAM system, SGS-SLAM [48], achieves 0.6980 on the same six scenes. Although our system's performance is slightly lower, it is sufficient to effectively guide the loop detection process.

Tracking: The tracking results, shown in Tables IV and III, reveal that our approach outperforms the baseline system Co-SLAM [2], with tracking accuracy gains of 16.16% on Replica [17] and 12.46% on ScanNet [1]. These substantial improvements confirm the effectiveness of our loop closure method. Additionally, compared to other recent systems, both with and without loop closure capabilities, our SLC^2-SLAM surpasses thess methods across most of the test scenes. Particularly in larger indoor rooms in ScanNet, our system shows superior average tracking performance. We attribute this to our system's ability to detect more loops, enabling additional pose graph optimizations that enhance tracking accuracy.

Reconstruction: The reconstruction results, both quantitative and qualitative, are presented in Table V and Fig. 5. While we aimed to compare our system with other loop-closure-enabled methods, only the one by Bruns et al. [33] provided these metrics. Thus, we compared SLC^2-SLAM against three more systems

TABLE V
RECONSTRUCTION RESULTS ON REPLICA

Methods	Metric	Room-0	Room-1	Room-2	Office-0	Office-1	Office-2	Office-3	Office-4	Avg.
Co-SLAM [2]	**Acc.**[cm]↓	2.11	1.68	1.99	1.57	1.31	2.84	3.06	2.23	2.10
	Comp.[cm]↓	2.02	1.81	1.96	1.56	1.59	2.43	2.72	2.52	2.08
	Comp. Ratio[< 5cm %]↑	95.26	95.19	93.58	96.09	94.65	91.63	90.72	90.44	93.44
Hu *et al.* [22]	**Acc.**[cm]↓	2.54	2.70	2.25	2.14	2.80	3.58	3.46	2.68	2.77
	Comp.[cm]↓	2.41	2.26	2.46	1.76	1.94	2.56	2.93	3.27	2.45
	Comp. Ratio[< 5cm %]↑	93.22	94.75	93.02	96.04	94.77	91.89	90.17	88.46	92.79
Bruns *et al.* [33]	**Acc.**[cm]↓	2.63	2.25	2.86	1.88	2.07	3.45	4.92	2.98	2.88
	Comp.[cm]↓	2.25	1.86	3.57	1.67	1.79	2.34	**2.69**	2.67	2.36
	Comp. Ratio[< 5cm %]↑	93.23	94.98	89.62	95.59	93.34	91.35	89.40	89.34	92.11
SLC^2-SLAM (ours)	**Acc.**[cm]↓	**1.42**	**1.31**	**1.29**	**1.19**	**1.09**	**2.67**	**2.56**	**1.54**	**1.63**
	Comp.[cm]↓	**1.41**	**1.24**	**1.37**	**1.16**	**1.04**	**1.51**	3.57	**1.57**	**1.61**
	Comp. Ratio[< 5cm %]↑	**99.36**	**99.96**	**98.58**	**99.42**	**99.59**	**95.85**	**91.11**	**98.15**	**97.75**

We mark the best results with bold and second best with underline.

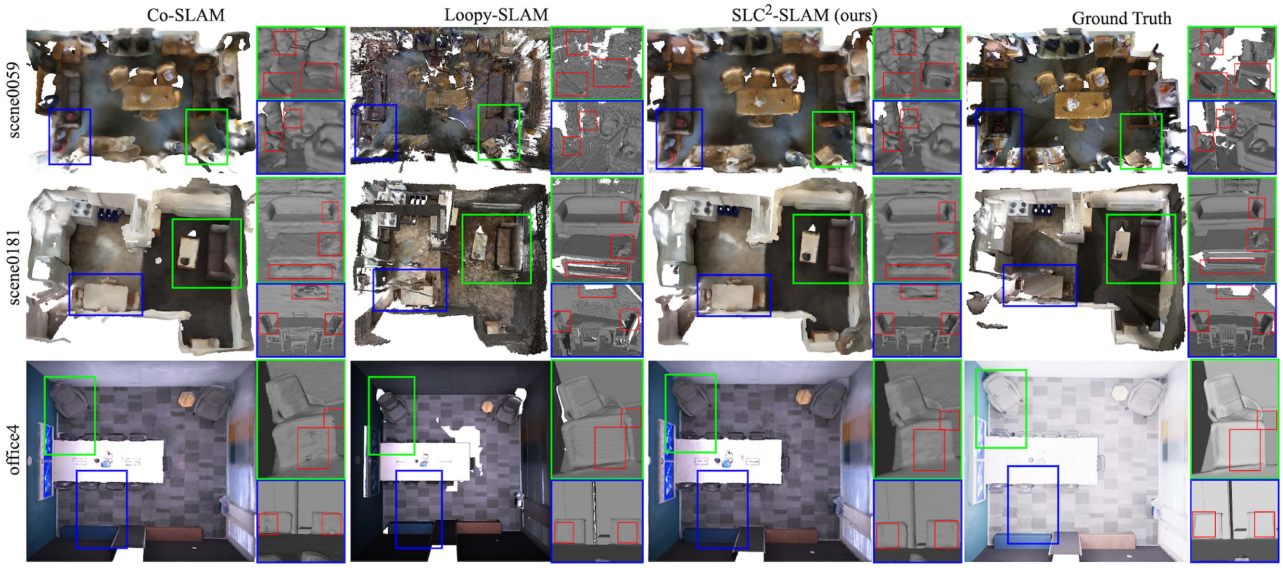

Fig. 5. Reconstruction examples of Co-SLAM [2], Loopy-SLAM [38], and our SLC^2-SLAM on the ScanNet and Replica datasets. Compared to Loopy-SLAM, our reconstructions are more complete for Replica scenes and better aligned and less noisy for ScanNet scenes. Compared to Co-SLAM, ours are more complete and less noisy for both datasets. Zoomed in views are provided with highlights for better visualization.

without loop closure, including our baseline Co-SLAM [2]. As shown, our SLC^2-SLAM significantly outperforms all other methods across all three metrics, with a considerable margin in each, underscoring the impact of our loop closure on reconstruction quality.

Memory and Runtime: Our SLC^2 -SLAM operates efficiently, consuming only 2GB video memory. Although all experiments were conducted on a NVIDIA 4090 GPU, the system can run on any GPU with a minimum of 4GB memory. In contrast, other loop-implemented NeRF SLAM, such as Loopy-SLAM [38], require GPUs with at least 12GB of memory. For runtime, the tracking and mapping processes of SLC^2-SLAM are on par with Co-SLAM [2]. Our loop detection and pose graph optimization achieve, on average, 1.6 seconds and 0.5 seconds per loop, respectively. Comparatively, Loopy-SLAM [38] needs 12 seconds.

V. CONCLUSION

In this letter, we present SLC^2-SLAM, a NeRF-SLAM system featuring a simple yet highly effective loop closure method. Our approach leverages on-the-fly learned latent codes, originally introduced to assist 3D scene reconstruction, and repurposes them as local features for global descriptor aggregation. To ensure these sampled latent codes accurately represent the current view, we introduce a semantic-guided stratified sampling, drawing on semantic information also decoded from the latent codes. We evaluate our SLC^2-SLAM on two publicly available datasets, comparing it to various NeRF-SLAM systems, both with and without loop closure, and demonstrate its superior performance.

REFERENCES

[1] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2432–2443.

[2] H. Wang, J. Wang, and L. Agapito, "Co-SLAM: Joint coordinate and sparse parametric encodings for neural real-time SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13293–13302.

[3] R. A. Newcombe et al., "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.

[4] T. Whelan et al., "ElasticFusion: Dense SLAM without a pose graph," in *Proc. Robot.: Sci. Syst. XI*, Sapienza Univ. Rome, Rome, Italy, Jul. 13-17, 2015, doi: 10.15607/RSS.2015.XI.001.

[5] Y. Ming, X. Yang, and A. Calway, "Object-augmented RGB-D SLAM for wide-disparity relocalisation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 2203–2209.

[6] X. Yang, Y. Ming, Z. Cui, and A. Calway, "FD-SLAM: 3-D reconstruction using features and dense matching," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 8040–8046.

[7] B. Mildenhall et al., "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[8] Y. Ming et al., "Benchmarking neural radiance fields for autonomous robots: An overview," *Eng. Appl. Artif. Intell.*, vol. 140, 2025, Art. no. 109685.

[9] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6209–6218.

[10] Z. Zhu et al., "NICE-SLAM: Neural implicit scalable encoding for SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12776–12786.

[11] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2022, pp. 499–507.

[12] Y. Ming, W. Ye, and A. Calway, "iDF-SLAM: End-to-end RGB-D SLAM with neural implicit mapping and deep feature tracking," 2022, *arXiv:2209.07919*.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[14] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 846–853.

[15] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetvLAD: CNN architecture for weakly supervised place recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, Jun. 2018.

[16] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, 2022.

[17] J. Straub et al., "The replica dataset: A digital replica of indoor spaces," 2019, *arXiv:1906.05797*.

[18] M. M. Johari, C. Carta, and F. Fleuret, "ESLAM: Efficient dense SLAM system based on hybrid representation of signed distance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17408–17419.

[19] Z. Zhang et al., "VPE-SLAM: Neural implicit voxel-permutohedral encoding for SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 5104–5110.

[20] S. Zhu et al., "SNI-SLAM: Semantic neural implicit SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21167–21177.

[21] H. Zhai, G. Huang, Q. Hu, G. Li, H. Bao, and G. Zhang, "NIS-SLAM: Neural implicit semantic RGB-D SLAM for 3D consistent scene understanding," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 11, pp. 7129–7139, Nov. 2024.

[22] P. Hu and Z. Han, "Learning neural implicit through volume rendering with attentive depth fusion priors," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2023, pp. 33012–33026.

[23] Z. Xin, Y. Yue, L. Zhang, and C. Wu, "HERO-SLAM: Hybrid enhanced robust optimization of neural SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 8610–8616.

[24] H. Jiang, Y. Xu, K. Li, J. Feng, and L. Zhang, "RoDyn-SLAM: Robust dynamic dense RGB-D SLAM with neural radiance fields," *IEEE Robot. Autom. Lett.*, vol. 9, no. 9, pp. 7509–7516, Sep. 2024.

[25] Z. Teed and J. Deng, "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 16558–16569.

[26] P. Pham et al., "Go-SLAM: Grounded object segmentation and localization with Gaussian splatting SLAM," 2024, *arXiv:2409.16944*.

[27] W. Zhang, T. Sun, S. Wang, Q. Cheng, and N. Haala, "HI-SLAM: Monocular real-time dense mapping with hybrid implicit fields," *IEEE Robot. Autom. Lett.*, vol. 9, no. 2, pp. 1548–1555, Feb. 2024.

[28] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[29] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[30] C. Chung et al., "Orbeez-SLAM: A real-time monocular visual SLAM with ORB features and NeRF-realized mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 9400–9406.

[31] H. Matsuki, K. Tateno, M. Niemeyer, and F. Tombari, "NEWTON: Neural view-centric mapping for on-the-fly large-scale SLAM," *IEEE Robot. Autom. Lett.*, vol. 9, no. 4, pp. 3704–3711, Apr. 2024.

[32] Y. Mao et al., "NGEL-SLAM: Neural implicit representation-based global consistent low-latency SLAM system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 6952–6958.

[33] L. Bruns, J. Zhang, and P. Jensfelt, "Neural graph mapping for dense SLAM with efficient loop closure," 2024, *arXiv:2405.03633*.

[34] Y. Tang et al., "MIPS-fusion: Multi-implicit-submaps for scalable and robust online neural RGB-D reconstruction," *ACM Trans. Graph.*, vol. 42, no. 6, pp. 246:1–246:16, 2023.

[35] H. Zhai et al., "Vox-fusion++: Voxel-based neural implicit dense tracking and mapping with multi-maps," 2024, *arXiv:2403.12536*

[36] Z. Xu, Q. Li, C. Chen, X. Liu, and J. Niu, "GLC-SLAM: Gaussian splatting slam with efficient loop closure," 2024, *arXiv:2409.10982*.

[37] L. Zhu, Y. Li, E. Sandström, S. Huang, K. Schindler, and I. Armeni, "LoopSplat: Loop closure by registering 3D Gaussian splats," 2024, *arXiv:2408.10154*.

[38] L. Liso, E. Sandström, V. Yugay, L. Van Gool, and M. R. Oswald, "Loopy-SLAM: Dense neural SLAM with loop closures," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 20363–20373.

[39] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3304–3311.

[40] Y. Chen and G. G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1991, pp. 2724–2729.

[41] C. Wang et al., "PyPose: A library for robot learning with physics-based optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22024–22034.

[42] Y. Ming, X. Yang, G. Zhang, and A. Calway, "CGiS-NEt: Aggregating colour, geometry and implicit semantic features for indoor place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 6991–6997.

[43] Y. Ming, J. Ma, X. Yang, W. Dai, Y. Peng, and W. Kong, "AEGIS-Net: Attention-guided multi-level feature aggregation for indoor place recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2024, pp. 4030–4034.

[44] N. V. Keetha et al., "SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21357–21366.

[45] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 883–890.

[46] N. V. Keetha et al., "AnyLoc: Towards universal visual place recognition," *IEEE Robot. Autom. Lett.*, vol. 9, no. 2, pp. 1286–1293, Feb. 2024.

[47] S. Izquierdo and J. Civera, "Optimal transport aggregation for visual place recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 17658–17668.

[48] M. Li et al., "SGS-SLAM: Semantic Gaussian splatting for neural dense SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 163–179.