MNE-SLAM: Multi-Agent Neural SLAM for Mobile Robots

Tianchen Deng[1], Guole Shen[1], Chen Xun[2], Shenghai Yuan[2], Tongxin Jin[2], Hongming Shen[2], Yanbo Wang[1], Jingchuan Wang*[1], Hesheng Wang[1], Danwei Wang[2], Weidong Chen*[1]
[1] Shanghai Jiao Tong University [2] Nanyang Technological University

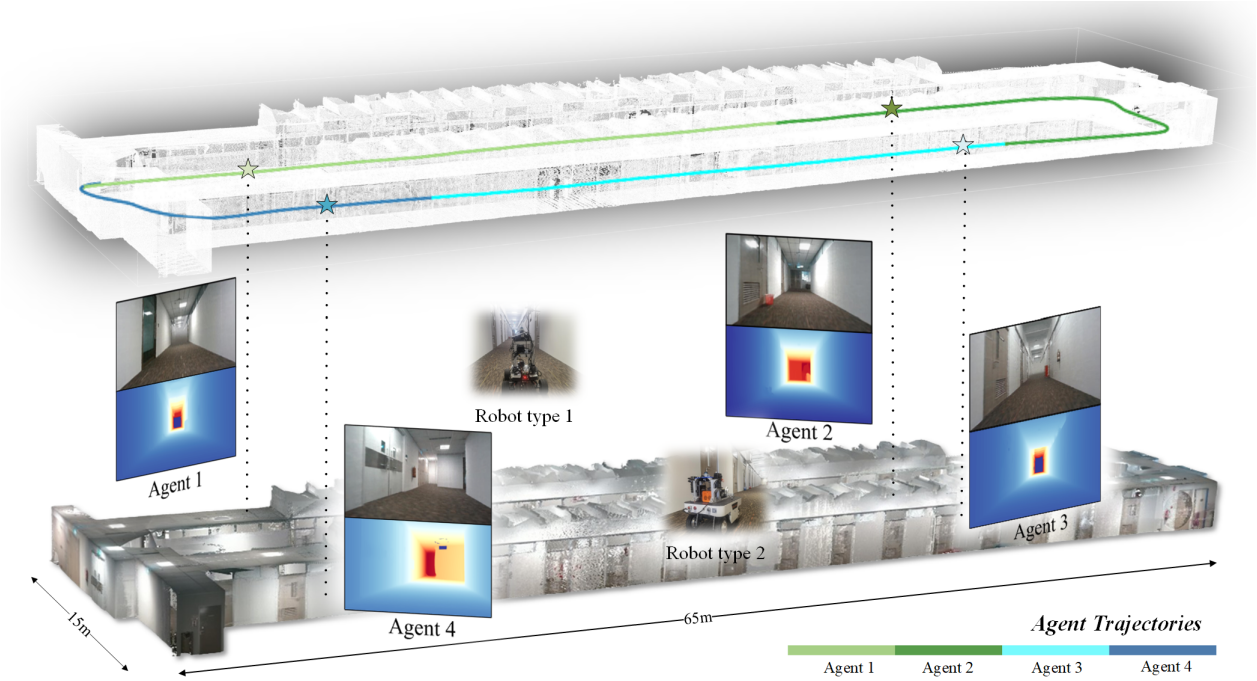


Figure 1. We present MNE-SLAM, the first distributed multi-agent collaborative SLAM system with distributed mapping and camera tracking, joint implicit neural scene representation, intra-to-inter loop closure, and multiple submap fusion. Depicted at the bottom, we demonstrate the real-world, large-scale long-corridor scenes (≈1000 m^2) with high-accuracy 3D groundtruth mesh. This scene is collected through various industrial laser scans. We display the rendered depth and color image of different type of agents around the corridor. The trajectory of each agent is marked in a unique color for clarity.

Abstract

Neural implicit scene representations have recently shown promising results in dense visual SLAM. However, existing implicit SLAM algorithms are constrained to single-agent scenarios, and fall difficulty in large indoor scenes and long sequences. Existing multi-agent SLAM frameworks cannot meet the constraints of communication bandwidth. To this end, we propose the first distributed multi-agent collaborative SLAM framework with distributed mapping and camera tracking, joint scene representation, intra-to-inter loop closure, and multi-submap fusion. Specifically, our proposed distributed neural mapping and tracking framework only needs peer-to-peer communication, which can greatly improve multi-agent cooperation and communication efficiency. A novel intra-to-inter loop closure method is designed to achieve local (single-agent) and global (multi-agent) consistency. Furthermore, to the best of our knowledge, there is no real-world dataset for NeRF-based/GS-based SLAM that provides both continuous-time trajectories groundtruth and high-accuracy 3D meshes groundtruth. To this end, we propose the first real-world indoor neural slam (INS) dataset covering both single-agent and multi-agent scenarios, ranging from small room to large-scale scenes, with high-accuracy ground truth for both 3D mesh and continuous-time camera trajectory. This dataset can advance the development of the community. Experiments on various datasets demonstrate the superiority of the proposed method in both mapping, tracking, and communication. The dataset and code will be open-source on https://github.com/dtc111111/MNESLAM.

*represents corresponding author. The first two authors contribute equally to this paper.

1. Introduction

Dense Visual Simultaneous Localization and Mapping has been a fundamental challenge in robotics and computer vision, which aims to achieve 3D reconstruction and localization [21, 22]. Traditional visual SLAM has witnessed continuous development, achieving accurate mapping and tracking in various scenarios, such as DTAM [26], Kinectfusion [12]. They reconstruct meaningful 3D global maps. Nowadays, with the proposal of Neural Radiance Fields (NeRF) [25] and 3DGS [16], it has wide applications such as autonomous driving [3, 8], remote sensing, and VR/AR [19, 27]. Many following works combine implicit scene representation with SLAM framework. iMAP [32] is the first work to use a single multi-layer perceptron (MLP) to reconstruct the scene in an online mapping framework. Some works [6, 13, 37, 43] further improves the scene representation with hybrid feature grids, axis-aligned feature planes, joint coordinate-parametric encoding, and other scene representation methods. They can achieve promising reconstruction quality in a small indoor room.

However, current research on NeRF-based SLAM frameworks primarily focuses on the single-robot setting. Compared to a single robot, multi-robot systems can perform tasks more efficiently and robustly. Moreover, the multi-agent framework enhances the system's resilience to individual agent failures, ensuring stability even in the presence of anomalies. We outline the key challenges for multi agents NeRF-based SLAM in large-scale scenes: **a) Communication bandwidth limitation.** The multi-agent system needs to operate under limited communication bandwidth. Agents can only exchange limited data instead of raw data in a timely manner to meet communication bandwidth constraints. **b) Accumulation of errors and pose drifts.** Pose error and drift will accumulate continuously as the robot operates, and this issue becomes particularly severe in large-scale environments. Existing frameworks struggle with accumulating errors in large-scale scenes. **c) Consensus on scene reconstruction among all agents.** The sensor observations of multiple agents are local and different, which leads to discrepancies of the final map. Furthermore, compared to traditional explicit sparse point cloud maps, the implicit map representation makes multi agent submap fusion more challenging.

CP-SLAM [10] designs a distributed-to-centralized learning framework for scene representation. However, it needs to train an extra global network with all the keyframe raw data in the centralized stage, which cannot meet the communication bandwidth restriction. Given the constraints of privacy and communication bandwidth, we design the first fully distributed multi-agents neural SLAM framework, which only relies on local (peer-to-peer) communication, achieving accurate scene reconstruction, robust pose estimation, and efficient multi-agent communication. The agents in our framework only exchange their network parameters and features instead of raw data when the system detects a loop. For each agent, it builds local scene representation and camera tracking.

Furthermore, we propose a novel intra-to-inter loop closure method for a multi-agent neural SLAM framework to mitigate the accumulation of pose errors and achieve consensus scene reconstruction among multiple agents. Intra-loop closure is designed to detect the pre-visit place of single robot, which can effectively eliminate the cumulative error with the global keyframe database. In a distributed neural SLAM system, all these agents optimize their scene representation independently with local and different observations. As a result, the final maps generated by individual agents only represent specific portions of the scene, leading to discrepancies. So, we propose inter-loop closure to detect the same scenario visited by multiple agents and register submaps generated from different agents. A global consistency loss is introduced to optimize and register the pose of different submaps. A peer-to-peer distillation method is designed for the fusion of different neural implicit submaps. When communication is available, we perform intra-to-inter loop closure, exchange the neural network parameters of each agent, and perform implicit submap fusion to achieve global consistency across multiple agents.

Furthermore, to the best of our knowledge, we find that current datasets are either virtual, such as Replica [30] or only provide trajectory ground truth without 3D ground truth, such as ScanNet [2], and TUM dataset [31]. ScanNet++ [40] is a recently proposed dataset that offers 3D ground truth. However, its poses are obtained through COLMAP, making them unsuitable as accurate SLAM ground truth. In addition, Scannet++ is primarily intended as a NeRF Training & Novel View Synthesis dataset, as stated in their paper and SplaTAM [15]. ScanNet++ contains non-time-continuous trajectories with numerous abrupt jumps and teleportations, which makes it unsuitable for SLAM systems. To this end, we propose a real-world dataset ranging from small-room scenarios to large-scale corridors. Our dataset provides high-accuracy and time-continuous trajectory and 3D mesh groudtruth, which is suitable for all neural SLAM systems, such as NeRF-based SLAM and 3DGS-based SLAM systems [4, 15, 16, 18, 23, 24, 38, 41]. Overall, our contributions are shown as follows:

- We design the first distributed multi-agent neural SLAM framework, MNE-SLAM, with joint neural scene representation, and robust camera tracking, achieving accurate collaborative scene reconstruction, robust camera tracking, and efficient multi-agent communication.
- A novel intra-to-inter loop closure method is proposed to detect and eliminate the cumulative pose drift and register the neural sub-maps for global map consistency.
- We propose the first real-world indoor neural SLAM

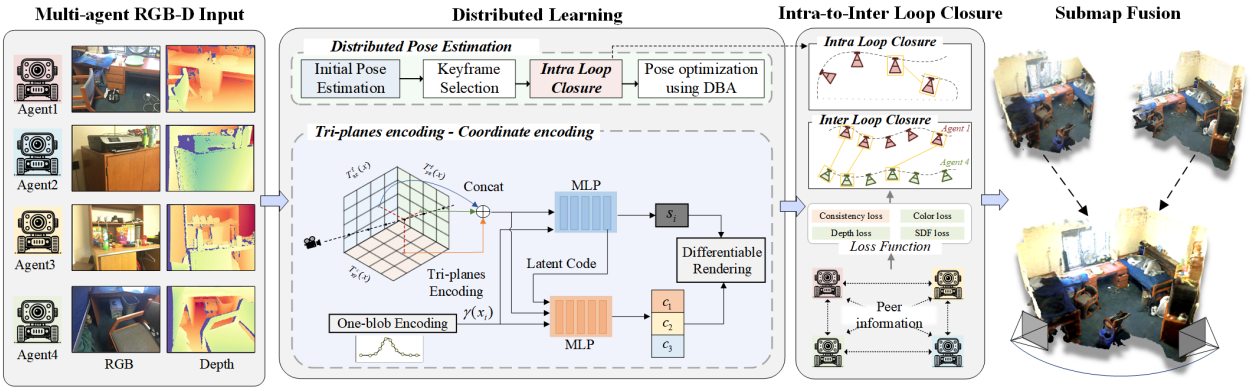


Figure 2. **System Overview.** Our system is a distributed multi-agent collaborative SLAM system that consists of joint scene representation, distributed tracking, intra-to-inter loop closure, and submap-fusion. Our framework takes the color images and depth images as input. Then, our framework performs distributed mapping and camera tracking. When communication is available, we perform intra-to-inter loop closure, exchange the neural network parameters of each agent, and perform implicit submap fusion to achieve global consistency across multiple agents. Each agent can successively perform individual scene mapping and collaborative mapping and tracking to generate the final neural implicit map with submap-fusion.

dataset (INS) for all kinds of neural SLAM systems with high-accuracy ground-truth for both camera trajectory and 3D reconstruction mesh, which can foster the development of the community.

2. Related work

Dense visual SLAM Dense visual SLAM technologies have significantly advanced in the past two decades. DTAM [26] first introduces a dense SLAM system that leverages photometric consistency for tracking handheld cameras and depicting scenes as cost volumes. KinectFusion [12] employs the iterative-closest-point method for camera tracking and updates the scene using TSDF-Fusion. DROID-SLAM [35] leverages neural networks to extract richer contextual information from images, significantly enhancing trajectory estimation.

Implicit Neural SLAM Recent advancements in neural implicit networks have revolutionized 3D scene reconstruction and camera pose estimation. Many efforts have focused on combining NeRF with SLAM systems to enhance both accuracy and efficiency. iMAP [32] and NICE-SLAM [43] are two noteworthy works, with iMAP pioneering the use of a single MLP for scene reconstruction and NICE-SLAM enhancing detail capture through a learnable hierarchical feature grid. Building on these foundations, ESLAM [13] reduces memory demands with axis-aligned feature planes. Co-SLAM [37] improves efficiency through its unique joint coordinate-parametric encoding strategy. To adapt to larger environments, PLGSLAM [5] and MIPS-Fusion [33] enhances scalability with its progressive scene representation method. Point-SLAM [28] uses neural points to represent the scene. GO-SLAM [42], Loopy-SLAM [20] use loop closure to enhance camera tracking performance. Our method expands from a single-agent to a multi-agent framework, achieving a globally coherent 3D reconstruction in large-scale scenarios.

Collaborative visual SLAM The goal of collaborative visual SLAM is to estimate the poses of multiple agents while reconstructing a consistent global map. There are two mainstream multi-agent SLAM frameworks: centralized and distributed. In terms of centralized systems, CVI-SLAM [14] is the first method to introduce full visual-inertial collaborative SLAM. CCM-SLAM [29] and CoSLAM [44] optimizes server use by offloading heavy computations while maintaining agent autonomy with onboard visual odometry. Within the distributed approach, Kimera-Multi [36] is a multi-robot Visual-Inertial Odometry (VIO) system with distributed mapping and tracking. MACIM [7] introduces a multi-agent online mapping algorithm with implicit ESDF scene representation. CP-SLAM [10] designs a distributed-to-centralized multi-agent SLAM framework which is not efficient for robot communication. Our proposed multi-agent SLAM framework is the first fully distributed neural SLAM method, improving global map accuracy, consistency, and robot communication efficiency.

3. Methods

The pipeline of our system is shown in Fig. 2. We propose MNE-SLAM, a distributed collaborative multi-agent SLAM system. The input of this multi-agent system is RGB-D frames $\{\mathbf{I_i}, \mathbf{D_i}\}_{i=1}^{M}$ with known camera intrinsic $K \in R_{3\times3}$. Our model predicts multi-agent camera poses $\{R_i|t_i\}_{i=1}^{M}$, color $\mathbf{c}$, and implicit truncated signed distance function (TSDF) representation. The system consists of four main modules: (i) distributed local mapping (Sec. 3.1), (ii) distributed camera tracking via PGO (Sec. 3.2), (iii) intra-to-inter loop closure (Sec. 3.3) (iv) sub-map fusion (Sec. 3.4). Among these modules, intra-loop closure and submap fusion are the ones that involve communication between different robots. We elaborate on the entire pipeline

of our system in the following subsections.

3.1. Joint Scene Representation

Voxel grid-based architectures [9, 43] are the mainstream in NeRF-based SLAM system. However, they face challenges with cubic memory growth and real-time performance in large-scale environments. We design a parametric-coordinate joint encoding method. The parametric encoding employs tri-plane encoding, while the coordinate encoding uses one-blob encoding with an MLP. This joint scene representation architecture enables high-fidelity, smooth scene reconstruction with enhanced hole-filling capabilities. Specifically, we employ a two-scale tri-plane representation, consisting of both coarse and fine levels, to model the scene. The tri-planes feature $\boldsymbol{T}(x)$ can be formulated as:

$$\begin{aligned} t^c(x) &= T^c_{xy}(x) + T^c_{xz}(x) + T^c_{yz}(x) \\ t^f(x) &= T^f_{xy}(x) + T^f_{xz}(x) + T^f_{yz}(x) \\ \boldsymbol{T}(x) &= Concat\left(t^c(x); t^f(x)\right) \end{aligned} \tag{1}$$

where $t^c(x), t^f(x)$ denote the coarse and fine feature form tri-planes. x is the world coordinate. $\{T^c_{xy}, T^c_{xz}, T^c_{yz}\}$ represent the three coarse geometry feature planes, and $\{T^f_{xy}, T^f_{xz}, T^f_{yz}\}$ represent the three fine geometry feature planes. For a sample point x, we use bilinearly interpolating the nearest neighbors on each feature plane. Then, we sum the interpolated coarse features and the fine, respectively, into the coarse output and fine output. Finally, we concatenate the outputs together as tri-plane features. The geometry decoder outputs the predicted SDF value and a feature vector. We use one-blob encoding [37] instead of embedding spatial coordinates into multiple frequency bands. Finally, the color decoder predicts the RGB value. Combining the MLP with the tri-planes scene representation, our architecture achieves accurate and smooth surface reconstruction, efficient memory use, and hole-filling performance.

3.2. Distributed Camera Tracking

To address the communication constraints of robots, we propose a distributed camera tracking method. We formulate the problem to optimize all pose variables:

$$\begin{aligned} \min_{\substack{\{R|t\}_{\alpha_i} \in \mathrm{SE}(3) \\ \forall \alpha \in \mathcal{R}, \forall i}} & \underbrace{\sum_{\alpha \in \mathcal{R}} \sum_{i=1}^{n_\alpha - 1} \mathcal{L}\left(\{R|t\}_{\alpha_i}, \{R|t\}_{\alpha_{i+1}}\right)}_{\text{odometry}} + \\ & \underbrace{\sum_{(\alpha_i, \beta_j) \in L} \mathcal{L}\left(r^{\alpha_i}_{\beta_i}\left(\{R|t\}_{\alpha_i}, \{R|t\}_{\beta_j}\right)\right)}_{\text{loop closures}}, \end{aligned} \tag{2}$$

where $\{R|t\}_{\alpha_i}$ denotes the ith pose of robot α in the global frame. $\mathcal{R} = \{\alpha, \beta, \dots\}$ denotes the set of robots, n_α is the total number of poses of robot α. For each agent, our distributed camera tracking framework takes a live RGB-D video stream. It first applies a recurrent update operator based on RAFT [34] to compute the optical flow of each new frame compared to the last keyframe. If the average flow is larger than a pre-defined τ_k, a new keyframe is created and added to the global keyframe database for further refinement. We use the set of keyframes $\{KF_i\}_{i=1}^N$ to create a keyframe-graph structure $(\mathcal{V}, \mathcal{E})$, which represents the co-visibility between frames. An edge $(i, j) \in \mathcal{E}$ means $\mathbf{I_i}, \mathbf{I_j}$ have overlapping fields of view which share points. The keyframe graph is built dynamically with the system operation.

Afterward, we use the differentiable Dense Bundle Adjustment (DBA) layer to solve a non-linear squares optimization problem to correct the camera pose $\{R_i|t_i\}_{i=1}^M$.

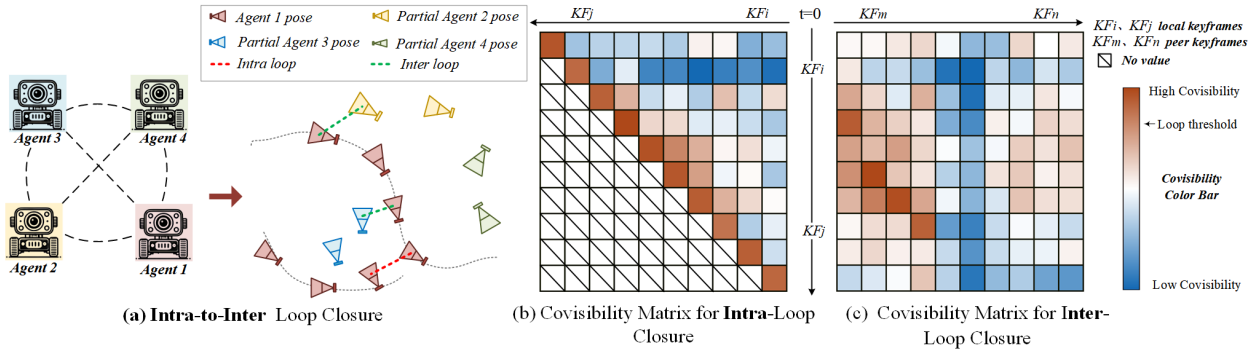
$$\begin{aligned} \mathbf{E}\left(\{\mathbf{R}|\mathbf{t}\}, \mathbf{d}\right) = \sum_{(i,j) \in \mathcal{E}} & \left\| \mathbf{p}^*_{ij} - \Pi_c(\{\mathbf{R}|\mathbf{t}\}_{ij} \circ \Pi_c^{-1} \right. \\ & \left. (\mathbf{p}_i, \mathbf{d}_i)) \right\|^2_{\Sigma_{ij}} + \alpha \sum \left\| \hat{\mathbf{d}}_\mathbf{i} - \mathbf{D_i} \right\|^2 \end{aligned} \tag{3}$$

where Π_c, Π_c^{-1} are the projection and back-projection functions, $\| \cdot \|_\Sigma$ and $\Sigma_{ij} = \operatorname{diag} \mathbf{w}_{ij}$ denotes the Mahalanobis distance which weights the error terms based on the confidence weights $\mathbf{w}_{ij}$. $\mathbf{p_i}$ is the 2D pixel position from keyframe KF_i. $\{\mathbf{R}|\mathbf{t}\}_{ij}$ is the pose transformation from KF_i to KF_j. $\hat{d}_i$ denotes the estimated depth. We use $\mathbf{p}^*_{ij}$ to present the estimated flow. α denotes the weight of depth residual. This loss function states that we want to optimize the camera pose and per-pixel depth to maximize the compatibility with flow $\mathbf{p}^*_{ij}$ predicted by the recurrent update operator.

3.3. Intra-to-Inter Loop Closure

Most neural implicit SLAM frameworks suffer from the accumulation of pose drifts and distortion in the reconstruction. With the growing cumulative error ε of pose estimation, those methods result in failure in large-scale indoor scenes and long videos. Concurrently, during multi-agent collaborative mapping in large-scale environments, the accumulation pose error of multiple agents results in rapid degradation of the submaps. To address these problems, we propose a novel intra-to-inter loop closure method, which can eliminate pose drift and register multiple submaps into the global map to achieve global consistency. We illustrate the schematic diagram of the method in Fig. 3.

Intra-Loop Closure For single agent (intra) loop closure, we build the keyframe-graph $(\mathcal{V}, \mathcal{E})$ with two steps: (i) detect and select keyframe pairs with high covisibility τ_{cov} in the local most recent keyframes N_{local} (ii) detect loop closure between local keyframes and historical keyframes outside the local window. Accordingly, we compute a co-visibility matrix of local keyframes of size $N_{local} \times N_{KF}$ for local loop closure. We also compute a covisibility ma-



(a) **Intra-to-Inter** Loop Closure (b) Covisibility Matrix for **Intra**-Loop Closure (c) Covisibility Matrix for **Inter**-Loop Closure

Figure 3. (a) presents the reconstruction of a multi-agent pose graph with intra-to-inter loop closure. (b) and (c) present the covisibility matrix of intra -loop closure and inter-loop closure. In Figure (b), the horizontal and vertical axes represent the keyframes of the local (single-agent) system, while the horizontal and vertical axes in Figure (c) represent the keyframes of the local and peer agents.

trix of all historical keyframes of size $N_{KF} \times N_{KF}$ for global loop closure, as shown in Fig. 3 (b). The covisibility is represented by the mean rigid flow between keyframe pairs using efficient back-projection. Those keyframe pairs with low covisibility (mean flow higher than threshold (τ_{cov}) are filtered out. We build edges for these keyframe pairs. We suppress the possible neighboring edges between $\{KF_k\}_{k=i-r_{local}}^{i+r_{local}} \rightarrow \{KF_k\}_{k=i-r_{local}}^{j+r_{local}}$, where r_{local} denotes a temporal radius. For local loop closure, the number of edges in the graph is linear to N_{local} with an upper bound $N_{local} \times N_{local}$. The number of edges of the global loop closure is linear to N_{KF} with an upper bound $N_{KF} \times N_{KF}$. We also suppress the redundant neighboring edges with radius r_{global}. By constructing a pose graph using global historical frames, we can leverage all information to optimize the pose of the current frame. Intra-loop closure effectively integrates local and global information, which greatly improves the accuracy of camera pose optimization in large-scale indoor scenes.

Inter-Loop Closure For multi-agent loop closure, we detect keyframe pairs from global keyframe database $\mathcal{G}$ of different agents. Once the covisibility value of the keyframe pairs is lower than the thershold τ_{cov}, we perform inter loop closure on these frames. We present the covisibility matrix in Fig. 3(c). For the robot α, β and the detected frames A, B, we set the odometric estimates of pose i and pose j as $\{\mathbf{R}|\mathbf{t}\}_{\alpha_i}^{A}, \{\mathbf{R}|\mathbf{t}\}_{\beta_j}^{B} \in \mathbb{SE}(3)$. Then, the relative transformation between frames A, B can be formulate as:

$$\{\mathbf{R}|\mathbf{t}\}_{B_{ij}}^{A} \triangleq \{\mathbf{R}|\mathbf{t}\}_{\alpha_i}^{A} \{\mathbf{R}|\mathbf{t}\}_{\beta_j}^{\alpha_i} \left(\{\mathbf{R}|\mathbf{t}\}_{\beta_j}^{B}\right)^{-1} \tag{4}$$

where the subscript of $\{\mathbf{R}_i|\mathbf{t}_i\}_{B_{ij}}^{A}$ indicates that this estimate is computed using inter loop closure (i,j). In order to obtain a reliable estimate of the true relative transformation, we formulate the pose average problem:

$$\{\mathbf{R}|\mathbf{t}\}_{B_{ij}}^{A} \in \underset{\{\mathbf{R}|\mathbf{t}\} \in \mathrm{SE}(3)}{\arg\min} \sum_{(i,j) \in L_{\alpha,\beta}} \mathcal{L}\left(r_{ij}(\{\mathbf{R}|\mathbf{t}\})\right) \tag{5}$$

where $\mathcal{L}$ denotes the loss function. $L_{\alpha,\beta}$ is the set of inter-robot loop closure between robot α and β. We adopt consistency loss to optimize the relative pose of frame A and frame B:

$$\mathcal{L}_{lc}(\{\mathbf{R}|\mathbf{t}\}) = \frac{1}{n} \sum_{\{A,B\}}^{n} \left(\hat{\boldsymbol{c}_A} - \hat{\boldsymbol{c}_B}\right)^2 \tag{6}$$

$$\mathcal{L}_{ld}(\{\mathbf{R}|\mathbf{t}\}) = \frac{1}{n} \sum_{\{A,B\}}^{n} \left(\hat{\boldsymbol{d}_A} - \hat{\boldsymbol{d}_B}\right)^2 \tag{7}$$

where $\hat{\boldsymbol{c}_A}, \hat{\boldsymbol{c}_B}$ are the rendered color image from agent α, β, while $\hat{\boldsymbol{d}_A}, \hat{\boldsymbol{d}_B}$ denote the rendered depth. This consistency loss effectively aligns the trajectories of agent α and agent β by comparing the rendering results of two images.

3.4. Distributed Optimization and Submap-Fusion

In our distributed multi-agent method, there is no global keyframe database across all agents. Each agent constructs a local keyframe database, where each keyframe extracts visual RAFT features. During the multi-agent communication process, high-dimensional features of each keyframe are transmitted when they arrive, and loop closure is detected based on the similarity of these feature vectors. When a loop closure is detected, a signal is sent to the two corresponding agents, triggering information exchange and fusion. This includes transmitting the keyframe poses and network parameters of two agents, optimizing the relative positions between the agents. Our mapping thread is performed via minimizing our objective functions with respect to network parameters θ and camera parameters $\{\mathbf{R}|\mathbf{t}\}$. The color and depth rendering losses are used in our mapping thread:

$$\mathcal{L}_c = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{\mathbf{c}}_i - \mathbf{C}_\mathbf{i}\right)^2, \quad \mathcal{L}_d = \frac{1}{|R_i|} \sum_{i \in R_i} \left(\hat{\mathbf{d}}_\mathbf{i} - \mathbf{D}_\mathbf{i}\right)^2 \tag{8}$$

where R_i is the set of rays that have a valid depth observation. In addition, we design SDF loss and free space loss for our mapping thread. Specifically, for samples within the

truncation region, we leverage the depth sensor measurement to approximate the signed distance field:

$$\mathcal{L}_{sdf} = \frac{1}{|R_i|} \sum_{r \in R_i} \frac{1}{|X_r^{tr}|} \sum_{x \in X_r^{tr}} \left(\phi_{\boldsymbol{g}}(x) \cdot T - (\mathbf{D_i} - \mathbf{d})\right)^2 \tag{9}$$

where X_r^{tr} is a set of points on the ray r that lie in the truncation region, $|\mathbf{D_i} - \mathbf{d}| \leq tr$. We differentiate the weights of points that are closer to the surface $X_r^{tm} = \{x | x \in |\mathbf{D_i} - \mathbf{d}| \leq 0.4tr\}$ from those that are at the tail of the truncation region X_r^{tt} in our SDF loss.

$$\mathcal{L}_{sdf_m} = \mathcal{L}_{sdf}\left(X_r^{tm}\right), \quad \mathcal{L}_{sdf_t} = \mathcal{L}_{sdf}\left(X_r^{tt}\right) \tag{10}$$

For sample points that are far from the surface $|D_i - d| \geq T$:

$$\mathcal{L}_{fs} = \frac{1}{|R_i|} \sum_{r \in R_i} \frac{1}{\left|X_r^{fs}\right|} \sum_{x \in X_r^{fs}} \left(\phi_{\boldsymbol{g}}(x) - 1\right)^2 \tag{11}$$

This loss can force the SDF prediction value to be the truncated distance tr.

Submap Fusion The sensor observations of multiple agents are local and different, which leads to discrepancies of the final map. In order to achieve global consistency and consensus on scene reconstruction among all agents, two critical components are required: (1) aligning the poses of submaps to ensure global consistency. We use inter-to-intra loop closure to align the poses of different submaps. (2) fusing the implicit submaps. Unlike traditional point cloud-based maps, implicit map fusion is particularly challenging, especially in distributed systems.

Due to communication bandwidth constraints, our peer-to-peer communication only exchanges network parameters. We propose a multi-agent distillation method to fuse submaps into a global map. By using the aligned poses and the keyframe depths, we can obtain the boundaries of the two agents' maps and compute the overlap region. Then, we use the poses of the overlapping region keyframe $\{KF\}_{i=1}^{m}$ to render the RGB and depth keyframe images from different agents. This allows the networks of two different agents to merge their parameters solely by comparing the rendered RGB $\hat{\mathbf{c}}$, depth images $\hat{\mathbf{d}}$, and sdf value $\hat{\mathbf{s}}$ with distillation loss $\mathcal{L}_{\alpha-\beta}$:

$$\mathcal{L}_{\alpha-\beta} = \frac{1}{m} \sum_{\alpha,\beta \in \{KF\}}^{m} \left(\left(\hat{\mathbf{c}}_\alpha - \hat{\mathbf{c}}_\beta\right)^2 + \left(\hat{\mathbf{d}}_\alpha - \hat{\mathbf{d}}_\beta\right)^2 + \left(\hat{\mathbf{s}_\alpha} - \hat{\mathbf{s}}_\beta\right)^2 \right) \tag{12}$$

We use this distillation loss to optimize the network parameters of agent α and β. We learn the scene parameters to ensure that the overall map scene representation parameters remain globally consistent. The maps are then quickly fused using the distillation method. The different agents work simultaneously and merge submaps in real-time.

4. Experiments

We validate that our method outperforms existing implicit representation-based methods in surface reconstruction, pose estimation, and real-time performance.

Datasets. We evaluate MNE-SLAM on a variety of scenes from different datasets.

- Replica Dataset [30]. 8 small room scenes (**nearly** $6.5m \times 4.2m \times 2.7m$ with 2000 images). We partition the dataset into two subsets, each corresponding to the trajectory of one agent, and ensured that there is overlap between the two subsets. We use this dataset to evaluate the reconstruction and localization accuracy in small-scale environments.
- ScanNet dataset [2]. Real-world scenes with long sequences (more than 5000 images) and large-scale indoor scenarios. (**nearly** $7.5m \times 6.6m \times 3.5m$). We also partition the dataset into two subsets for two agents. We use this dataset for real-world indoor environments.
- Apartment dataset from [30]. Multi-rooms scene (**nearly** $10.8m \times 8.3m \times 3.2m$ with 5000 images). We use this dataset for multi-room environments.
- Our **I**ndoor dataset for **N**eural **S**LAM systems (**INS** dataset). This is the first real-world dataset for all kinds of neural SLAM systems with high-accuracy ground-truth for both camera trajectory and 3D reconstruction mesh. We collected single-agent and multi-agent datasets from various indoor environments, ranging from small room scenes (**nearly** $35m^2$) to large-scale scenes (**>1000** m^2), accumulating more than 100,000 camera frames. The details are listed in supplementary material. The dataset has been partially open-source on Github.

4.1. Experimental Results

Scene reconstruction and tracking. We evaluate our method on Replica [30], ScanNet [2], Apartment [30] and our own dataset **INS**. As shown in Tab. 1, we present the two single-agent reconstruction and tracking results (Agent1, Agent2) and the global scene reconstruction and tracking performance in small room scenarios on the Replica dataset. Best results are highlighted as first, second, and third. Our multi-agent method achieves accurate and consistent global map reconstruction through multi-map fusion. In Tab. 2, we provide the single-agent and multi-agent camera tracking performance on Apartment dataset [30]. Although the tracking performance of a single agent does not reach the best, we achieve the SOTA performance in global camera tracking. This is attributed to our multi-agent collaborative tracking, which leverages intra-to-inter loop closure to eliminate pose error and achieve consistency across multiple agents. In Tab. 3, we present the global camera tracking performance on ScanNet [2]. Our method outperforms existing multi-agent and single-agent methods. We show the qualitative comparison on **INS** dataset of our method with

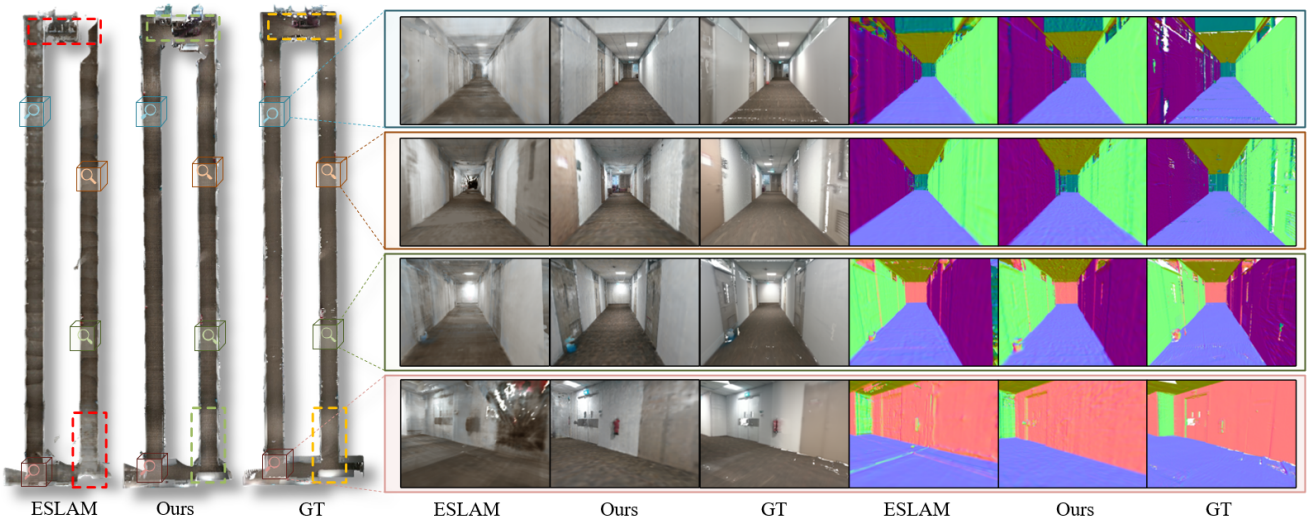


Figure 4. Qualitative comparison of our proposed method's surface reconstruction with existing SOTA method: ESLAM [13] on the long corridor sequence. On the right is the zoomed-in view of the reconstructed mesh. The region outlined on the image is marked in red to signify lower reconstruction accuracy, in green to signify higher accuracy, and in yellow to represent the ground truth results.

Methods	Reconstruction									Localization		
	Agent 1			Agent 2			Global			Agent 1	Agent 2	Global
	Acc. ↓	Comp. ↓	Comp.Ratio(%) ↑	Acc. ↓	Comp. ↓	Comp.Ratio(%) ↑	Acc. ↓	Comp. ↓	Comp.Ratio(%) ↑	RMSE ↓	RMSE ↓	RMSE ↓
Single-Agent Methods												
iMAP[32]	4.812	4.519	76.378	3.663	3.582	81.352	2.881	4.934	80.515	3.481	3.773	4.153
NICE-SLAM[43]	3.035	3.907	85.894	2.163	2.881	90.792	2.373	2.645	91.137	1.657	1.767	2.503
Vox-Fusion[39]	2.975	3.757	83.599	1.991	2.609	91.015	2.881	2.563	90.936	0.977	1.432	1.453
ESLAM[13]	1.838	1.979	93.317	1.857	1.884	94.531	2.082	1.754	95.977	0.593	0.579	0.657
Co-SLAM[37]	1.758	2.153	92.894	1.949	2.033	93.413	2.104	2.082	93.435	1.001	0.934	1.059
GO-SLAM[42]	2.931	3.759	86.484	2.245	2.678	90.571	2.091	2.415	93.174	0.334	0.451	0.404
Point-SLAM[28]	2.324	3.455	86.811	1.925	3.102	90.975	1.935	2.779	90.479	0.534	0.461	0.559
PLGSLAM[5]	1.793	2.012	93.415	1.983	1.773	94.669	1.884	1.543	96.329	0.584	0.607	0.631
Loopy-SLAM[20]	1.935	3.107	89.894	1.683	2.783	91.792	1.593	2.645	91.137	0.257	0.362	0.369
Multi-Agent Methods												
CP-SLAM[10]	2.271	3.457	86.429	1.834	2.901	90.115	1.794	2.455	90.394	0.613	0.639	0.673
Ours	1.731	1.932	93.692	1.797	1.685	95.458	1.681	1.509	96.957	0.281	0.358	0.398

Table 1. Quantitative results of our proposed method with existing NeRF-based SLAM system on the Replica dataset [30]. We evaluate the reconstruction and localization performance of Agent1, Agent2, and global(global scene reconstruction and camera tracking). We present the evaluation results of the reconstruction and trajectory outcomes for different agents, as well as the scene reconstruction and trajectory results of the global map. The results are the average of the scenes in the Replica dataset. Our method outperforms existing methods in surface reconstruction and pose estimation. Best results are highlighted as first , second , and third .

the SOTA method ESLAM. in Fig. 4. Our method successfully achieves consistent completion as well as high-fidelity surface reconstruction results in large-scale, real-world indoor scenes. In Tab. 4, We present the performance on the INS dataset, showing that most neural SLAM methods experience significant pose drift in real-world scenarios, ultimately leading to reconstruction failure. Our multi-agent approach, however, successfully achieves accurate reconstruction and pose estimation in large-scale environments.

Time analysis We analyze the speed and memory usage of our method compared with other SOTA methods in Replica datasets [30] and ScanNet datasets [2]. We evaluate the time consumption of tracking and mapping every iteration and every frame in Tab. 5. We also evaluate the memory and communication consumption of existing methods. Compared to CP-SLAM, our method improves communication efficiency by seven times (**7** $\times$). Our distributed method significantly enhances communication efficiency, increasing the effectiveness in real-world operations.

4.2. Ablation Study

In this section, we conduct various experiments to verify the effectiveness of our method. Tab. 6 illustrates a quantitative evaluation with different settings.

Intra-Loop Closure We remove our intra loop closure in this experiment. Our full model leads to higher scene reconstruction accuracy and better completion. The intra loop closure can significantly reduce the growing cumulative error and improve the robustness and accuracy of the single-

Methods	Apartment-1			Apartment-2			Apartment-0		
	Ag. 1	Ag. 2	Global	Ag. 1	Ag. 2	Global	Ag. 1	Ag. 2	Global
Single-Agent Methods									
ORB-SLAM3[1]	4.93	4.93	4.93	1.35	1.36	1.36	0.67	1.46	1.07
NICE-SLAM[43]	55.46	21.65	38.56	5.7	2.99	4.35	2.17	2.2	2.18
Co-SLAM[37]	2.86	3.51	3.19	1.44	1.64	1.54	0.83	0.78	0.83
ESLAM[13]	1.38	0.95	1.17	0.84	0.75	0.79	0.58	0.95	0.76
GO-SLAM[42]	1.45	1.33	1.27	0.49	0.78	0.62	0.47	0.56	0.53
Point-SLAM[28]	1.31	2.09	1.63	0.53	0.80	0.68	0.49	0.62	0.58
PLGSLAM[5]	1.33	1.06	1.13	0.82	0.73	0.79	0.56	0.93	0.74
Loopy-SLAM[20]	1.19	1.66	1.43	0.55	0.66	0.60	0.46	0.82	0.64
Multi-Agent Methods									
CCM-SLAM[29]	2.12	9.31	5.71	0.51	0.74	0.62	-	-	-
Swarm-SLAM[17]	4.62	6.50	5.56	2.69	8.53	5.61	1.61	1.98	1.80
CP-SLAM[10]	6.21	5.67	5.73	1.45	2.48	1.85	0.62	1.28	0.91
Ours	1.21	0.99	1.02	0.43	0.74	0.59	0.43	0.53	0.48

Table 2. Two-agent tracking performance in Replica dataset [30]. Following the dataset partitioning setting of [10], we quantitatively evaluate respective trajectories and global results of the two agents. "Ag.1" and "Ag.2" denote Agent 1 and Agent 2. "-" indicates invalid results due to the failure of CCM-SLAM.

Methods	0000	0059	0106	0169	0181	0207	0054	0233	Avg.
Single-Agent Methods									
NICE-SLAM[43]	12.0	14.0	7.9	10.9	13.4	6.5	20.9	9.0	13.0
ESLAM[13]	7.3	8.5	7.5	7.5	9.0	6.0	36.3	4.6	10.6
Co-SLAM[37]	7.1	11.1	9.4	7.1	11.8	7.1	14.8	-	-
Point-SLAM[28]	10.2	7.8	8.7	22.2	14.8	9.5	28.0	6.1	14.3
GO-SLAM[42]	5.4	7.5	7.0	7.7	6.8	6.9	8.8	4.8	6.9
PLGSLAM[5]	7.3	8.2	7.4	7.5	9.2	5.8	30.8	4.2	9.9
Loopy-SLAM[20]	5.2	7.9	8.5	7.7	10.6	7.9	14.5	5.2	7.7
Multi-Agent Methods									
Swarm-SLAM[17]	10.7	8.7	9.5	12.5	13.8	8.4	14.5	6.5	10.5
CP-SLAM[10]	7.6	8.9	8.8	8.1	10.8	8.3	20.1	5.6	9.7
Ours	4.9	7.2	6.6	7.3	6.4	6.4	8.5	4.3	6.6

Table 3. **Camera Tracking Performance on ScanNet dataset [2].** We use ATE RMSE(cm) as the metric. We select 8 sequences follow the setting from previous methods.

Methods	Reconstruction[cm]			Localization[m]	
	Acc.↓	Comp.↓	Comp.Ratio(%)↑	Mean ↓	RMSE ↓
Single-Agent Methods					
ORB-SLAM3[1]	-	-	-	0.76	0.80
Co-SLAM[37]	-	-	-	16.84	18.80
ESLAM[13]	73.78	17.60	58.49	1.10	1.23
GO-SLAM[42]	109.124	133.152	10.53	3.76	4.36
Point-SLAM[28]	-	-	-	2.83	2.95
PLG-SLAM[5]	70.42	15.49	59.37	1.01	1.12
Photo-SLAM[11]	60.61	139.89	10.75	0.43	0.45
Loopy-SLAM[20]	-	-	-	1.95	2.01
Multi-Agent Methods					
CP-SLAM[10]	-	-	-	9.50	10.67
Ours	**43.96**	**14.69**	**64.00**	**0.38**	**0.40**

Table 4. **Scene Reconstruction and Camera Tracking Performance on INS dataset**. "-" indicates invalid results due to the failure or inability of these methods.

agent camera tracking.

Inter-Loop Closure We remove the inter-loop closure in this experiment. Inter-loop closure can effectively register different submaps into the global map and improve the tracking performance across different agents.

Submap Fusion Submap fusion can significantly improve the integration of information among different maps, ensuring the consistency of the global map; otherwise, the map may exhibit numerous discontinuities or floating artifacts. This also suggests that handling smooth transitions between two submaps, especially when revisiting an inactive submap, is critical to the overall reconstruction quality.

	Methods	Map/Fr.	Track/Fr.	Memory	Communication
Replica[30]	ESLAM[13]	0.29s	68.5ms	20.73MB	-
	Co-SLAM[37]	0.15s	54.8ms	26.39MB	-
	Point-SLAM[28]	10.47s	29.7ms	31.42MB	-
	CP-SLAM[10]	10.45s	34.7ms	39.75MB	429.78MB
	Ours	0.14s	22.1ms	24.45MB	60.58MB

Table 5. Runtime, Scene representation memory, and multi-robot communication performance evaluation on Replica [30] dataset. "Memory" refers to the storage consumption of the scene representation and the decoder. "Communication" refers to the amount of data transmitted during the robot's operation.

Methods	Reconstruction[cm]			Localization[cm]		
	Acc.↓	Comp.↓	Comp.Ratio(%)↑	Mean↓	Median↓	RMSE↓
w/o intra-loop.	1.831	1.592	95.779	0.397	0.403	0.432
w/o inter-loop	1.903	1.597	95.679	0.401	0.409	0.441
w/o submap	1.949	1.607	95.494	0.359	0.365	0.401
Ours	**1.681**	**1.509**	**96.957**	**0.357**	**0.362**	**0.398**

Table 6. Ablation study. We conduct experiments on Replica dataset [30] to verify the effectiveness of our method. Our full model achieves better completion reconstructions and more accurate pose estimation results.

5. Conclusion

In this paper, we propose a novel distributed multi-agent neural SLAM system, MNE-SLAM, which achieves accurate collaborative surface reconstruction and pose estimation in both small and large-scale indoor scenes. Our distributed mapping and tracking framework enables our system to represent large-scale indoor scenes under communication bandwidth restriction. The intra-to-inter loop closure can effectively eliminate pose drifts and achieve global map consistency across multiple agents. Our extensive experiments demonstrate the effectiveness and accuracy of our system in both scene reconstruction and pose estimation. The proposed neural SLAM dataset with high-accuracy 3D mesh and time-continuous camera trajectory can greatly advance the development of the community.

Acknowledgement This work is supported by the National Key R&D Program of China (Grant 2020YFC2007500), and the Science and Technology Commission of Shanghai Municipality (Grant 20DZ2220400). This research is supported by the National Research Foundation, Singapore, under the NRF Medium Sized Centre scheme (CARTIN), Maritime and Port Authority of Singapore under its Maritime Transformation Programme (Project No. SMI-2022-MTP-04), ASTAR under National Robotics Programme with Grant No.M22NBK0109.

References

[1] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 8

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6, 7, 8

[3] Tianchen Deng, Siyang Liu, Xuan Wang, Yejia Liu, Danwei Wang, and Weidong Chen. Prosgnerf: Progressive dynamic neural scene graph with frequency modulated auto-encoder in urban scenes. *arXiv preprint arXiv:2312.09076*, 2023. 2

[4] Tianchen Deng, Yaohui Chen, Leyan Zhang, Jianfei Yang, Shenghai Yuan, Danwei Wang, and Weidong Chen. Compact 3d gaussian splatting for dense visual slam. *arXiv preprint arXiv:2403.11247*, 2024. 2

[5] Tianchen Deng, Guole Shen, Tong Qin, Jianyu Wang, Wentao Zhao, Jingchuan Wang, Danwei Wang, and Weidong Chen. Plgslam: Progressive neural scene represenation with local to global bundle adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19657–19666, 2024. 3, 7, 8

[6] Tianchen Deng, Yanbo Wang, Hongle Xie, Hesheng Wang, Rui Guo, Jingchuan Wang, Danwei Wang, and Weidong Chen. Neslam: Neural implicit mapping and self-supervised feature tracking with depth completion and denoising. *IEEE Transactions on Automation Science and Engineering*, pages 1–1, 2025. 2

[7] Yinan Deng, Yujie Tang, Yi Yang, Danwei Wang, and Yufeng Yue. Macim: Multi-agent collaborative implicit mapping. *IEEE Robotics and Automation Letters*, 2024. 3

[8] Zhiheng Feng, Wenhua Wu, and Hesheng Wang. Rogs: Large scale road surface reconstruction based on 2d gaussian splatting. *arXiv e-prints*, pages arXiv–2405, 2024. 2

[9] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 4

[10] Jiarui Hu, Mao Mao, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Cp-slam: Collaborative neural point-based slam system. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3, 7, 8

[11] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21584–21593, 2024. 8

[12] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 2, 3

[13] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. 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*, pages 17408–17419, 2023. 2, 3, 7, 8

[14] Marco Karrer, Patrik Schmuck, and Margarita Chli. Cvi-slam—collaborative visual-inertial slam. *IEEE Robotics and Automation Letters*, 3(4):2762–2769, 2018. 3

[15] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 2

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2

[17] Pierre-Yves Lajoie and Giovanni Beltrame. Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. *IEEE Robotics and Automation Letters*, 9(1):475–482, 2023. 8

[18] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pages 163–179. Springer, 2024. 2

[19] Mingrui Li, Zhetao Guo, Tianchen Deng, Yiming Zhou, Yuxiang Ren, and Hongyu Wang. Ddn-slam: Real time dense dynamic neural implicit slam. *IEEE Robotics and Automation Letters*, 2025. 2

[20] Lorenzo Liso, Erik Sandström, Vladimir Yugay, Luc Van Gool, and Martin R Oswald. Loopy-slam: Dense neural slam with loop closures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20363–20373, 2024. 3, 7, 8

[21] Jiuming Liu, Guangming Wang, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Translo: A window-based masked point transformer framework for large-scale lidar odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1683–1691, 2023. 2

[22] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang. Regformer: An efficient projection-aware transformer network for large-scale point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8451–8460, 2023. 2

[23] Shuhong Liu, Heng Zhou, Liuzhuozheng Li, Yun Liu, Tianchen Deng, Yiming Zhou, and Mingrui Li. Structure gaussian slam with manhattan world hypothesis. *arXiv preprint arXiv:2405.20031*, 2024. 2

[24] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. *arXiv preprint arXiv:2312.06741*, 2023. 2

[25] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[26] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. 2, 3

[27] Zhenghao Qi, Shenghai Yuan, Fen Liu, Haozhi Cao, Tianchen Deng, Jianfei Yang, and Lihua Xie. Air-embodied: An efficient active 3dgs-based interaction and reconstruction framework with embodied large language model. *arXiv preprint arXiv:2409.16019*, 2024. 2

[28] Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18433–18444, 2023. 3, 7, 8

[29] Patrik Schmuck and Margarita Chli. Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *Journal of Field Robotics*, 36(4):763–781, 2019. 3, 8

[30] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 6, 7, 8

[31] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012. 2

[32] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, pages 6229–6238, 2021. 2, 3, 7

[33] Yijie Tang, Jiazhao Zhang, Zhinan Yu, He Wang, and Kai Xu. Mips-fusion: Multi-implicit-submaps for scalable and robust online neural rgb-d reconstruction. *ACM Transactions on Graphics (TOG)*, 42(6):1–16, 2023. 3

[34] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 4

[35] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 3

[36] Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P How, and Luca Carlone. Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Transactions on Robotics*, 38(4), 2022. 3

[37] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. 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*, pages 13293–13302, 2023. 2, 3, 4, 7, 8

[38] Chi Yan, Delin Qu, Dong Wang, Dan Xu, Zhigang Wang, Bin Zhao, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. *arXiv preprint arXiv:2311.11700*, 2023. 2

[39] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507, 2022. 7

[40] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 2

[41] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. 2

[42] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3727–3737, 2023. 3, 7, 8

[43] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, pages 12786–12796, 2022. 2, 3, 4, 7, 8

[44] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):354–366, 2012. 3