MNE-SLAM: Multi-Agent Neural SLAM for Mobile Robots (Supplementary Material)

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

1. Overview

In this supplementary material, we provide the implementation details in Sec. 2. In Sec. 4 we present the extensive experiment results on various datasets. The detail information of our dataset is shown in Sec. 3.

2. Implementation Details

Here we report the detailed settings and hyperparameters used in MNE-SLAM to achieve multi-agent collaboration, high-quality surface reconstruction, accurate camera tracking, and real-time performance. The truncation distance T is set to 6 cm in our method. The coarse feature planes is employed with a resolution of 24 cm. For fine feature planes, we use a resolution of 6 cm. All feature planes have 32 channels, resulting in a 64-channel concatenated feature input for the decoders. The decoders are two-layer MLPs with 32 channels in the hidden layer. The dimension of the geometric feature $\mathbf{z}$ is 15. ReLU activation function is used for the hidden layer, and Tanh and Sigmoid are respectively used for the output layers of TSDF and raw colors. We use 16 bins for One-Blob encoding of each dimension. For Replica [26] dataset, we sample N = 32 points for stratified sampling and $N_{surface}$ = 8 points for importance sampling on each ray. We use 200 iterations for first frame mapping. We perform 10 optimization iterations for mapping and randomly select 4000 rays for each iteration. And for ScanNet [6] dataset, we set N = 48 and $N_{surface}$ = 8. Also, we perform 30 optimization iterations for both mapping and tracking in ScanNet scenes. For the scenes in Apartment dataset [42], we similarly set N = 48 and $N_{surface}$ = 8. For this dataset, We perform 30 optimization iterations for mapping and tracking, and we randomly sample 5000 rays for each iteration.

We employ the pre-trained weights from DROID-SLAM [32] for tracking. We set $N_{\text{local}} = 25$, $r_{\text{local}} = 1$ and $\tau_{\text{co}} = 25$. For the loss coefficients for mapping, we set $\lambda_{fs} = 5$, $\lambda_{sdf_m} = 200$,$\lambda_{sdf_t} = 10$, $\lambda_d = 0.1$, and $\lambda_c = 5$. We use the RAFT [31] feature to select proporiate keyframe for jointly optimizing the feature tri-planes, MLP decoders, and camera poses of the selected keyframes. We add a keyframe when the average flow is greater than or equal to 4 pixels. We use Adam [14] for optimizing all learnable parameters of our method. All experiments are conduct with a desktop PC with NVIDIA RTX 3090 GPU.

Once all input frames are processed, and for evaluation purposes, we build a TSDF volume for each scene and use the marching cubes algorithm [19] to obtain 3D meshes. We use inverse distance weight to fuse our local scene representation into the entire mesh.

Culling Method. In previous NeRF-based SLAM method, all of them use an extra mesh culling step before evaluating the reconstructed mesh. iMAP [28] and NICE-SLAM [42] adopt a frustum culling strategy which removes the mesh vertices outside any of the camera frustum. This culling strategy remove the artifacts outside camera frustum but cannot remove artifacts inside camera frustum. In NeuralRGBD [1] and ESLAM [12], they adopt *frustum+occlusion* culling method. While this strategy could effectively remove some artifacts, their overly aggressive culling strategy results in many holes in the culled mesh. Follow [33], We introduce a modification to the culling strategy used for the quantitative evaluation of the reconstruction accuracy, which leads to a fairer comparison. We use the *frustum+occlusion+virtul camera* culling method. This method simulates virtual camera views that cover the occluded regions.

Evaluation Metrics. After mesh culling, we evaluate the reconstructed mesh with a mixture of 3D (Accuracy [cm]↓, Completion [cm]↓ and Completion Ratio↑) metrics. In Tab. 1, we present the 3D reconstruction metrics. We first uniformly sample two point clouds P and Q from both GT and reconstructed meshes, with $|P| = |Q| = 200000$. Accuracy metric is defined as the average distance between a point on GT mesh to its nearest point on reconstructed mesh. The Completion metric is defined as the average distance between a point on reconstructed mesh to its nearest point on GT mesh. The Completion Ratio metric refers to the pro-

*represents corresponding author.

Reconstruction Metrics	Definition
Depth L1 [cm]	$\frac{1}{N}\sum(\lvert d_i - d_i^* \rvert)/d_i$
Accuracy [cm]	$\sum_{p\in P}\left(\min_{q\in Q}\lVert p-q\rVert\right)/\lvert P\rvert$
Completion [cm]	$\sum_{q\in Q}\left(\min_{p\in P}\lVert p-q\rVert\right)/\lvert Q\rvert$
Completion Ratio [$< 5cm\%$]	$\sum_{q\in Q}\left(\min_{p\in P}\lVert p-q\rVert < 0.05\right)/\lvert Q\rvert$
Completion Ratio [$< 15cm\%$]	$\sum_{q\in Q}\left(\min_{p\in P}\lVert p-q\rVert < 0.15\right)/\lvert Q\rvert$

Table 1. Definitions of scene reconstruction metrics used for evaluation of surface reconstruction quality.

Scene ID	Sequence ID	Scale	Total Length	Total duration
Indoor_corridor	Corridor1,2,3,4	>1000 m^2	1482.75m	23487
Semi_outdoor	Semi_outdoor1,2,3,4	>1500 m^2	1549.60m	25691
Cross_floor	Cross_floor1,2,3,4	>3000 m^2	3078.69m	27832
Room	Room1,2,3,4	35 m^2	183.47m	4375

Table 2. Summary of scene ID, and the sequence ID in each scene. We present the scales of different scenes, as well as the cumulative lengths of the datasets and the total duration (number of frames). For further details of our dataset, refer to the supplementary.

portion of the overall ground truth (GT) where the average distance between a point on the reconstructed mesh and its nearest point on the GT mesh is less than the threshold t. Since the scale of indoor scenes is much larger than that of Replica [26], ScanNet [6], and Apartment [26] datasets, we use different completion ratio thresholds for different datasets. For small-scale scenes, we set the threshold to 5 cm, while for large-scale scenes(INS), it is set to 15 cm.

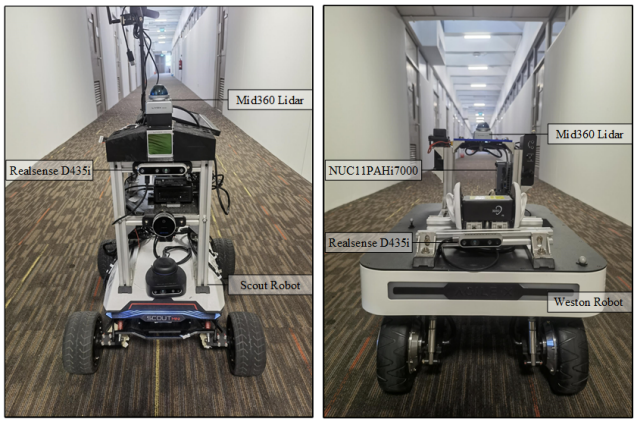
For 2D metrics, we use depth L1[cm] ↓, PSNR↑, SSIM↑, LPIPS↓ metrics. We render RGB and depth images along with the trajectory of camera. The Depth L1 is defined as the average L1 difference between rendered GT depth and rendered depth.

For trajectory metrics, we use ATE RMSE (cm), median and mean. For the camera trajectories generated by CCM-SLAM, we align them with the Ground Truth camera trajectory using Sim(3) Umeyama alignment in the EVO tool. As for the camera trajectories produced by other methods, we align them with the Ground Truth camera trajectory by aligning the origin. Trajectory alignment is crucial for proper drift and loop closure evaluation. To be specific, after aligning the initial poses, we calculate the Absolute Trajectory Error (ATE) for each pose and compute the RMSE, Mean, and Median values.

3. Dataset

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

- Replica Dataset [26]. 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 ensure that there is overlap between the two subsets. We use this dataset to evaluate the reconstruction and localization accuracy in small-scale environments.



Indoor Robot Platform

Figure 1. Visualization of the robot platform in our dataset.

- ScanNet dataset [6]. 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 [26]. Multi-rooms scene (**nearly** $10.8m \times 8.3m \times 3.2m$ with 5000 images). We use this dataset for multi-room environments. The Apartment dataset is a multi-agent collaborative dataset with two distinct agents
- 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 dataset has been partially open-source on Github.

Comparison with other datasets. In Tab. 3, we present a comparison of our dataset with others, focusing on key aspects such as scene types (indoor, outdoor), sensor modalities (camera, depth, LiDAR, IMU), inclusion of pose and 3D map ground truth, their acquisition methods, and a brief description. The upper part of the table lists 3D reconstruction datasets, none of which provide high-precision trajectory ground truth. The lower part lists SLAM datasets, none of which include 3D map ground truth. Currently, most existing neural SLAM methods typically use Replica [26], ScanNet [6], Apartment [26], TUM RGB-D [27]. we find that current datasets are either virtual, such as Replica [26] or only provide trajectory ground truth without 3D ground truth, such as ScanNet [6], and TUM dataset [27]. ScanNet++ [38] is a recently proposed dataset that offers 3D ground truth. However, its poses are

Dataset	Year	Scenes		Senor Modalities				GroundTruth		GT Method		Description
		Ind.	Out.	RGB	Depth	Lidar	IMU	Pose	3D Map	Pose	3D Map	
Tanks and Temples [15]	ToG'17	✓	✗	✓	✗	✗	✗	✓	✓	ICP	Laser Scan	High-quality Object-centric Scenes
RealEstate10k [41]	ToG'18	✓	✓	✓	✗	✗	✗	✓	✗	COLMAP	✗	Video clips on YouTube captured from a moving camera
NeRF-LLFF [21]	SIG'19	✓	✗	✓	✗	✗	✗	✓	✗	COLMAP	✗	Images of various indoor scenes and objects.
ACID [18]	ICCV'21	✗	✓	✓	✗	✗	✗	✓	✗	COLMAP	✗	A dataset of aerial landscape videos
Mip-NeRF 360 [2]	CVPR'22	✓	✓	✓	✗	✗	✗	✓	✗	COLMAP	✗	Outdoor and indoor scenes with 360-degree scene perspectives
LocalRF [20]	CVPR'23	✗	✓	✓	✗	✗	✗	✓	✗	COLMAP	✗	Large-scale outdoor dataset, Static hikes
TUM RGB-D [27]	IROS'12	✓	✗	✓	✓	✗	✗	✓	✗	Motion Capture	✗	Various indoor scenes for mapping and localization
KITTI [9]	IJRR'13	✗	✓	✓	✗	✓	✓	✓	✗	GNSS/INS	✗	A city-scale dataset created for autonomous driving research
ShapeNet [5]	arXiv'15	✓	✗	✓	✗	✗	✗	✓	✓	Simulation	CAD	A richly-annotated, large-scale dataset of 3D shapes
ScanNet [6]	CVPR'17	✗	✗	✓	✓	✗	✗	✓	✗	SLAM	✗	A richly-annotated multiple room scenes with semantic label
Replica [26]	arXiv'19	✓	✗	✓	✓	✗	✗	✓	✗	Simulation	Simulation	Highly photo-realistic 3D indoor scene dataset at room scale
Nuscenes [3]	CVPR'20	✗	✓	✓	✗	✗	✗	✓	✗	GNSS/INS	✗	A city-scale dataset created for autonomous driving research
Waymo [29]	CVPR'20	✗	✓	✓	✗	✗	✗	✓	✗	GNSS/INS	✗	A city-scale dataset created for autonomous driving research
NTU VIRAL [22]	IJRR'22	✗	✓	✓	✗	✗	✗	✓	✗	TLS	✗	A dataset of aerial mapping and localization
ScanNet++ [37]	ICCV'23	✓	✗	✓	✓	✗	✗	✓	✓	SLAM	Laser Scan	A large-scale dataset with high-quality and geometry and color of indoor scenes.
SubT-MRS [40]	CVPR'24	✗	✓	✓	✗	✓	✓	✓	✗	GNSS/INS	✗	Mapping and localization under diverse all-weather conditions
Ours	CVPR'25	✓	✗	✓	✓	✓	✓	✓	✓	MCTR	Laser Scan	A large-scale indoor and outdoor dataset with a widerange of sensing modalities and high-accuracy groundtruth for both single-agent and multi-agent

Table 3. List of commonly used datasets for neural mapping and localization. We provide the first dataset that includes both high-precision trajectory ground truth and 3D mesh ground truth. It covers indoor and outdoor scenes, incorporates multiple sensor modalities, and includes both single-agent and multi-agent sequences.

Modality	Hardware	ROS Topic	Type	Description	Rate
Camera	D435i	/d435i/depth/image_raw	sensor_msgs/Image	848×480 Depth	30HZ
		/d435i/infra1/image_rect_raw		1280×720 Greyscale	
		/d435i/infra2/image_rect_raw		1280×720 Greyscale	
		/d435i/color/image_raw		1280×720 RGB	
	D455	/d455/depth/image_raw		848×480 Depth	
		/d455/infra1/image_rect_raw		1280×720 Greyscale	
		/d455/infra2/image_rect_raw		1280×720 Greyscale	30HZ
		/d455/color/image_raw		1280×720 RGB	
IMU	D435i	/d435i/imu	sensor_msgs/Imu	Bosch BMI055	400HZ
	D455	/d435i/imu			400HZ
	Livox Mid360	/livox/lidar/imu			200HZ
Lidar	Livox Mid360	/livox/lidar	livox_ros_driver/CustomMsg	1 channel. Points per channel: 9984	10HZ

Table 4. Summary of sensing modalities, hardware units, ROS topics, and the nominal rates on each platform in our dataset. All these data have also been directly extracted from the rosbag and saved as individual files.

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 [13]. 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.

Scene ID	Sequence ID	Scale	Total Length	Total duration
Indoor_corridor	Corridor1,2,3,4	$>1000\ m^2$	1482.75m	23487
Semi_outdoor	Semi_outdoor1,2,3,4	$>1500\ m^2$	1549.60m	25691
Cross_floor	Cross_floor1,2,3,4	$>3000\ m^2$	3078.69m	27832
Room	Room1,2,3,4	$35\ m^2$	183.47m	4375

Table 5. Summary of scene ID, and the sequence ID in each scene. We present the scales of different scenes, as well as the cumulative lengths of the datasets and the total duration (number of frames).

Sensor Suit and Robot Platform In Fig. 1, we present the two robotic platforms used for dataset collection, along with detailed sensor information, such as cameras and LiDAR. In Tab. 4, we present the summary of sensing modalities, handware units, ROS topics, and the nominal rates on each platform. All these data have also been directly extracted from the rosbag and saved as individual files.

Calibration The calibration process involves finding the camera intrinsics and the extrinsics of all sensors. First, the Kalibr toolbox is used to find the intrinsic parameters of the D455 cameras. These intrinsic parameters are then used to find the extrinsic of the camera setup on each D455 w.r.t. each IMU available. Finally, we perform a pose-graph optimization over the graph to obtain the final extrinsic of all sensors and summarize them in the calibration report.

3D Groundtruth Map 3D survey-grade mapping of the campuses was done by using various terrestrial laser scanners (TLS). In essence, TLS acquires point cloud data from discrete, static locations by means of ground-based, high-resolution 3D laser scanners. These individual scans are then combined to producea complete 3D point cloud via co-visible landmarks (both man-made and natural) and propriety global scan matching software. point cloud data consists of 3D point coordinates and may also include intensity or RGB channels. Scanning each section took between several days, requiring at least two field operators. Upon completion of the scanning process, all individual scans were registered in the manufacturer's software. Registration of individual scans in each of the sections was performed solely on the basis of the artificial sphere targets recorded in the scans. Cloud-to-cloud registration was then used to align the three sections and create the final point cloud of the entire campus area. Target-based statistics showed a maximum distance error of less than 10 mm for all sphere pairings that were used in the registration optimization process. Building walls and corners were spot-checked for overlap of scans. A total of 15 reference measurements were made with a total station to verify the global accuracy of the point cloud. The results showed a maximum distance error of less than 5 cm over than entire length of the campus, which is below the resolution of the point cloud. We get the 3D mesh use the groundtruth point cloud with Poisson Surface Reconstruction and TSDF fusion. In Tab. 5, we provide information about the collected sequences. Additionally, following the tools provided in Imap [28], users can generate custom camera trajectories along with corresponding RGB and Depth images using the 3D ground truth.

Trajectory Groundtruth Many existing datasets face challenges related to the accuracy of ground truth estimates. GNSS/INS, the most commonly used localization method for automotive vehicles, often exhibits errors at the decimeter level. The Motion Capture system is primarily effective indoors or in low-light conditions. MoCap offer accuracy at the centimeter to millimeter level with clear line-of-sight. Two commonly used methods for trajectory estimation through point cloud matching are Normal Distributions Transform (NDT) and Iterative Closest Point (ICP). These methods often introduce errors in the range of a few decimeters. Simultaneous Localization and Mapping (SLAM) uses onboard LIDAR sensors but may suffer from long-term drift and is generally considered less accurate, with measurement noise ranging from a few decimeters to several meters. The proposed prior map continuous time registration (MCTR) is estimated to provide centimeter-level accuracy without any line-of-sight requirements, making it a promising choice for creating large datasets with centimeter accuracy. To make working with continuous-time trajectory ground truth easy, we have created a python wrapper for the basalt, library called CEVA (Continuous-time Evaluation), which can be installed from the our github website. In many aspects, CEVA exceeds the basalt library in its utilities. Some of the improvements we have made are: CEVA is a Python module, therefore, it can be easily imported into any Python program or Jupyter notebooks instead of having to be included and compiled like the C++ library basalt. Several Jupyter notebooks are provided at scripts to showcase CEVA's utilities.

4. Extensive Experiments

Due to space constraints, we present some extensive results on supplementary. We first present the 2D rendering results (PSNR, SSIM, LPIPS) and depth estimation results (Depth L1) on the Replica dataset [26] in Tab. 6, showing the outcomes for Agent 1, Agent 2, and the overall global sequence. Compared to other multi-agent methods, our approach achieves superior results. In Tab. 7, we present the tracking results (RMSE, Median, Mean) for each agent sequence in the Apartment dataset. It is evident that our method achieves better performance. We also present global tracking performance on TUM RGB-D [27] in Tab. 8. Our method achieves superior reconstruction accuracy compared to other methods.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference*

Methods	Reconstruction											
	Agent 1				Agent 2				Global			
	Depth L1 (cm)	PSNR ↑	SSIM ↑	LPIPS ↓	Depth L1 (cm)	PSNR ↑	SSIM ↑	LPIPS ↓	Depth L1 (cm)	PSNR ↑	SSIM ↑	LPIPS ↓
Single-Agent Methods												
iMAP[28]	3.15	23.19	0.772	0.272	3.21	23.41	0.760	0.278	3.18	23.33	0.769	0.274
NICE-SLAM[42]	3.01	24.28	0.820	0.250	2.96	24.57	0.799	0.257	2.98	24.43	0.809	0.254
Vox-Fusion[36]	2.45	24.09	0.813	0.240	2.49	24.98	0.803	0.253	2.47	24.45	0.791	0.249
ESLAM[12]	1.16	28.36	0.920	0.244	1.21	27.83	0.927	0.248	1.19	28.06	0.923	0.246
Co-SLAM[33]	1.47	26.56	2.153	0.876	1.50	26.39	0.870	0.266	1.51	26.46	0.873	0.269
GO-SLAM[39]	3.01	23.88	0.820	0.298	2.96	23.17	0.799	0.317	3.38	23.53	0.801	0.305
Point-SLAM[23]	0.46	35.27	0.974	0.110	0.53	34.95	0.978	0.116	0.49	35.10	0.980	0.112
PLGSLAM[8]	1.14	28.36	0.919	0.242	1.20	27.83	0.925	0.248	1.17	28.16	0.922	0.244
Loopy-SLAM[17]	0.56	35.53	0.981	0.110	0.59	35.31	0.980	0.116	0.57	35.40	0.981	0.112
Multi-Agent Methods												
CP-SLAM[10]	1.14	28.30	0.915	0.242	1.20	27.69	0.925	0.244	1.16	28.01	0.920	0.243
Ours	0.83	30.79	0.959	0.202	0.90	30.09	0.955	0.204	0.87	30.41	0.957	0.208

Table 6. Quantitative results of our proposed method with existing NeRF-based SLAM systems on the Replica dataset [26]. 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 rendering performance of different agents, as well as the results of the global map. The results are the average of the scenes in the Replica dataset. Our method outperforms existing methods in RGB and depth rendering.

Methods	Apartment-1			Apartment-2			Apartment-0		
	Agent 1	Agent 2	Global	Agent 1	Agent 2	Global	Agent 1	Agent 2	Global
	RMSE[cm]/Mean[cm]/Median[cm]			RMSE[cm]/Mean[cm]/Median[cm]			RMSE[cm]/Mean[cm]/Median[cm]		
Single-Agent Methods									
ORB-SLAM3[4]	4.93/4.65/5.01	4.93/4.04/3.80	4.93/4.35/4.41	1.35/1.05/0.65	1.36/1.24/1.11	1.36/1.15/0.88	0.67/0.58/0.47	1.46/1.11/0.79	1.07/0.85/0.63
NICE-SLAM[42]	55.4/53.4/50.9	21.6/20.3/19.4	38.5/36.4/35.9	5.70/5.53/5.25	2.99/2.79/2.51	4.35/4.15/4.01	2.17/2.01/1.93	2.21/2.03/1.94	2.18/2.05/1.97
Co-SLAM[33]	2.86/2.74/2.58	3.51/3.40/3.24	3.19/2.95/2.77	1.44/1.32/1.27	1.64/1.45/1.39	1.54/1.44/1.38	0.83/0.74/0.67	0.78/0.73/0.0.67	0.83/0.75/0.68
ESLAM[12]	1.38/1.29/1.17	0.95/0.89/0.81	1.17/1.07/0.98	0.84/0.76/0.69	0.75/0.69/0.61	0.79/0.73/0.67	0.58/0.53/0.49	0.95/0.88/0.81	0.76/0.70/0.65
GO-SLAM[39]	1.45/1.38/1.30	1.33/1.21/1.16	1.27/1.12/1.01	0.49/0.45/0.43	0.78/0.74/0.70	0.62/0.58/0.53	0.47/0.44/0.41	0.56/0.52/0.49	0.53/0.50/0.48
Point-SLAM[23]	1.31/1.23/1.11	2.09/1.98/1.77	1.63/1.51/1.38	0.53/0.47/0.42	0.80/0.76/0.73	0.68/0.62/0.57	0.49/0.46/0.42	0.62/0.58/0.55	0.58/0.55/0.53
PLGSLAM[8]	1.33/1.26/1.18	1.06/0.99/0.91	1.13/1.06/0.98	0.82/0.77/0.70	0.73/0.68/0.65	0.79/0.74/0.71	0.56/0.53/0.51	0.93/0.87/0.83	0.74/0.69/0.66
Loopy-SLAM[17]	1.19/1.07/0.98	1.66/1.53/1.44	1.43/1.35/1.26	0.55/0.51/0.48	0.66/0.61/0.53	0.60/0.53/0.48	0.46/0.42/0.40	0.82/0.79/0.76	0.64/0.61/0.58
Multi-Agent Methods									
CCM-SLAM[24]	2.12/1.94/1.74	9.31/6.36/5.57	5.71/4.15/3.66	0.51/0.45/0.40	0.74/0.70/0.68	0.62/0.59/0.57	-/-/-	-/-/-	-/-/-
Swarm-SLAM[16]	4.62/4.17/3.90	6.50/5.27/4.39	5.56/4.72/4.15	2.69/2.48/2.34	8.53/7.59/7.10	5.61/5.04/4.72	1.61/1.33/1.09	1.98/1.48/0.94	1.80/1.41/1.02
CP-SLAM[10]	6.21/5.56/5.27	5.67/5.37/4.67	5.73/5.26/4.77	1.45/1.43/1.39	2.48/2.32/2.27	1.85/1.68/1.73	0.62/0.47/0.30	1.28/1.17/1.37	0.91/0.78/0.80
Ours	1.21/1.09/1.07	0.99/0.87/0.93	1.02/1.01/0.99	0.43/0.41/0.40	0.74/0.72/0.70	0.59/0.58/0.55	0.43/0.41/0.40	0.53/0.50/0.49	0.48/0.46/0.45

Table 7. Two-agent tracking performance in Replica dataset [26]. ATE RMSE(↓), Mean(↓) and Median(↓) (cm) are used as evaluation metrics. Following the setting of [10], we quantitatively evaluated respective trajectories (Agent 1 and Agent 2) and global results of the two agents. ”-” indicates invalid results due to the failure of CCM-SLAM. Our method achieve SOTA performance compared with other existing methods.

on Computer Vision and Pattern Recognition (CVPR), pages 6290–6301, 2022. 1

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 3

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3

[4] 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. 5, 6

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[6] 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. 1, 2, 3

[7] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time

Methods	Loop Closure	fr1/desk	fr1/desk2	fr1/room	fr2/xyz	fr3/office	Avg.
Traditional Methods							
BAD-SLAM [25]	✓	1.7	N/A	N/A	1.1	1.7	N/A
Kintinuous [34]	✓	3.7	7.1	7.5	2.9	3.0	4.84
ORBSLAM3 [4]	✓	1.6	2.2	4.7	0.4	1.0	1.98
ElasticFusion [35]	✓	2.53	6.83	21.49	1.17	2.52	6.91
BundleFusion [7]	✓	1.6	N/A	N/A	1.1	2.2	N/A
Neural-based Methods							
DI-Fusion [11]	✗	4.4	N/A	N/A	2.0	5.8	N/A
NICE-SLAM [42]	✗	4.26	4.99	34.49	6.19	3.87	10.76
Vox-Fusion [36]	✗	3.52	6.00	19.53	1.49	26.01	11.31
MIPS-Fusion [30]	✓	3.0	N/A	N/A	1.4	4.6	N/A
Point-SLAM [23]	✗	4.34	4.54	30.92	1.31	3.48	8.92
ESLAM [12]	✗	2.47	3.69	29.73	1.11	2.42	7.89
Co-SLAM [33]	✗	2.40	N/A	N/A	1.7	2.4	N/A
PLGSLAM [8]	✗	2.45	3.67	29.61	1.10	2.40	7.85
GO-SLAM [39]	✓	1.52	2.78	4.64	0.69	1.39	2.20
Loopy-SLAM [17]	✓	3.79	3.38	7.03	1.62	3.41	3.85
Ours	✓	1.42	2.45	4.50	0.62	1.31	2.05

Table 8. **Tracking Performance on TUM-RGBD [27]** (ATE RMSE ↓ [cm]). shows competitive performance on a variety of scenes. On average outperforms existing dense neural RGBD methods that do not employ loop closure (LC), and is reducing the gap to traditional dense and sparse SLAM methods.

globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(4), 2017. 6

[8] 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. 5, 6

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 3

[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. 5

[11] Jiahui Huang, Shi-Sheng Huang, Haoxuan Song, and Shi-Min Hu. Di-fusion: Online implicit 3d reconstruction with deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8932–8941, 2021. 6

[12] 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. 1, 5, 6

[13] 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. 3

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[15] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 3

[16] 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. 5

[17] 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. 5, 6

[18] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021. 3

[19] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 1

[20] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H. Kim, and Johannes Kopf. Progres-

sively optimized local radiance fields for robust view synthesis. In *CVPR*, 2023. 3

[21] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019. 3

[22] Thien-Minh Nguyen, Shenghai Yuan, Muqing Cao, Yang Lyu, Thien H Nguyen, and Lihua Xie. Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint. *The International Journal of Robotics Research*, 41 (3):270–280, 2022. 3

[23] 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. 5, 6

[24] 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. 5

[25] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[26] 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. 1, 2, 3, 4, 5

[27] 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, 3, 4, 6

[28] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, pages 6229–6238, 2021. 1, 4, 5

[29] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 3

[30] 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. 6

[31] 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. 1

[32] 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. 1

[33] 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. 1, 5, 6

[34] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended kinectfusion. 2012. 6

[35] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: science and systems*, page 3. Rome, Italy, 2015. 6

[36] 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. 5, 6

[37] 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. 3

[38] 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

[39] 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. 5, 6

[40] Shibo Zhao, Yuanjun Gao, Tianhao Wu, Damanpreet Singh, Rushan Jiang, Haoxiang Sun, Mansi Sarawata, Yuheng Qiu, Warren Whittaker, Ian Higgins, Yi Du, Shaoshu Su, Can Xu, John Keller, Jay Karhade, Lucas Nogueira, Sourojit Saha, Ji Zhang, Wenshan Wang, Chen Wang, and Sebastian Scherer. Subt-mrs dataset: Pushing slam towards all-weather environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22647–22657, 2024. 3

[41] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 3

[42] 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. 1, 5, 6