

LEMON-Mapping: Loop-Enhanced Large-Scale Multi-Session Point Cloud Merging and Optimization for Globally Consistent Mapping

Lijie Wang¹, Xiaoyi Zhong¹, Ziyi Xu¹, Kaixin Chai², Anke Zhao¹, Tianyu Zhao¹, Changjian Jiang³, Qianhao Wang^{1,†}, Xieyuanli Chen^{4,†}, Fei Gao^{1,†}

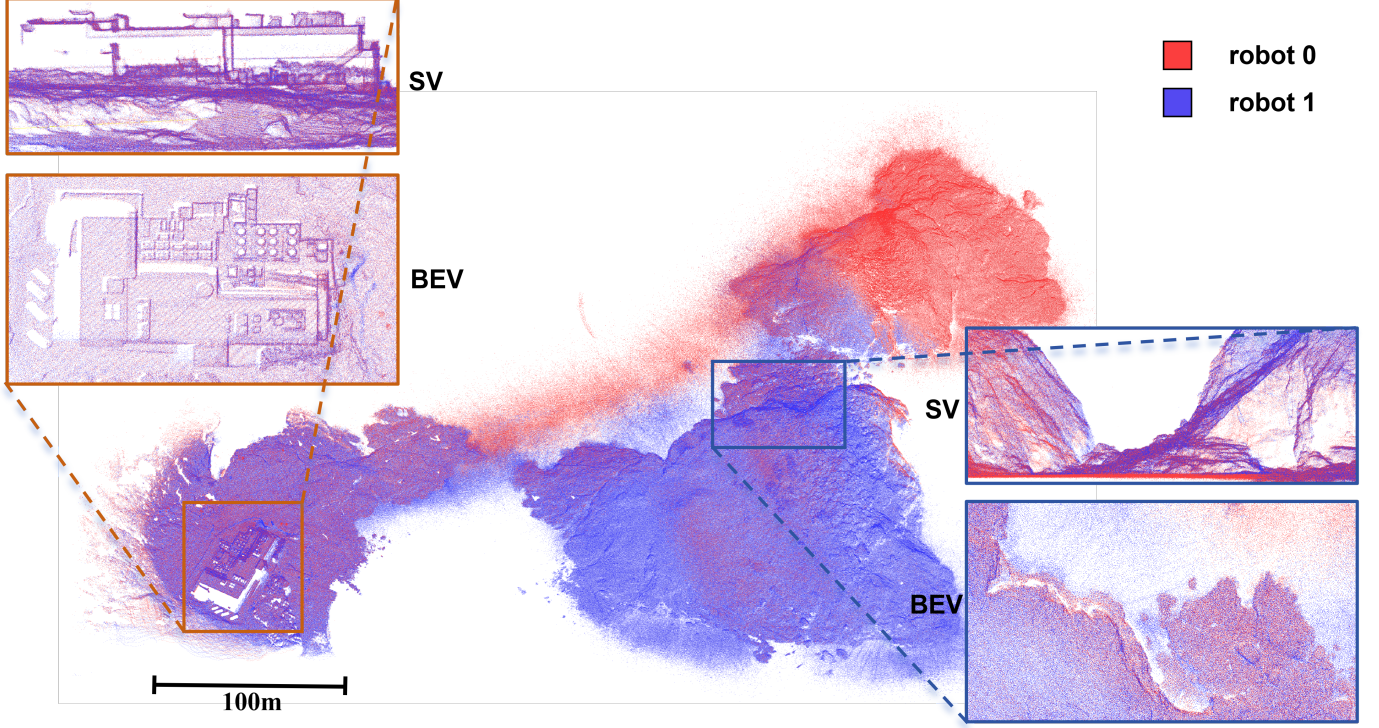


Fig. 1: The merging map of our framework in island sequence of MARS-LVIG [1] dataset, the details in the figure are framed and shown in two forms: side view (SV) and bird's eye view (BEV).

Abstract—Multi-robot collaboration is critical but challenging in building globally consistent maps. Traditional multi-robot pose graph optimization (PGO) methods ensure basic global consistency but ignore the geometric structure of the map, and only use loop closures as constraints between pose nodes, leading to divergence and blurring in overlapping regions. To address this, we propose LEMON-Mapping, a loop-enhanced framework for large-scale, multi-session point cloud fusion and optimization. We re-examine the role of loops in multi-robot mapping and present three key innovations. First, we develop a robust loop processing mechanism with outlier rejection and a recall strategy to recover valid loops. Second, we introduce spatial bundle adjustment to reduce divergence and eliminate blurring in overlapping areas. Third, we design a PGO-based optimization that integrates refined bundle adjustment constraints to propagate local accuracy globally. Experiments on public and self-collected datasets demonstrate that LEMON-Mapping achieves superior

accuracy, consistency, and scalability over traditional approaches in large-scale multi-robot scenarios.

Note to Practitioners—In this paper, we address inaccurate and inconsistent large-scale multi-robot map fusion, where conventional multi-robot SLAM ignores geometric structures, causing divergence and blurring in overlapping regions. In real-world exploration, multi-robot mapping is designed to extend coverage, but inconsistent and inaccurate map fusion degrades map quality and affects downstream applications like re-localization, navigation, and robotic operation, potentially leading to navigation failures and unsafe behaviors. LEMON-Mapping integrates robust loop processing, spatial bundle adjustment, and pose graph optimization to reconstruct accurate and globally consistent multi-robot maps, providing reliable support for multi-robot collaboration, localization, and autonomous navigation.

Index Terms—Localization and Mapping, Multi-Robot SLAM, Swarms

I. INTRODUCTION

LARGE-SCALE 3D mapping is a fundamental capability in modern robotics, providing rich geometric information that supports tasks such as drone-based inspection [2]–[4], autonomous driving [5], and long-term exploration with ground robots [6]. Furthermore, large-scale 3D maps are critical for

¹Institute of CyberSystems and Control, Zhejiang University, Hangzhou 310027, China. (e-mail: 3210101760@zju.edu.cn)

²The Huzhou Institute, Zhejiang University, Huzhou 313000, China.

³The State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China.

⁴The College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410007, China.

[†]Corresponding author: Qianhao Wang and Fei Gao. Email: fgaoaa@zju.edu.cn.

emerging fields such as embodied AI, where agents interact with complex environments based on spatial understanding [7]. They also play a key role in end-to-end visuomotor navigation systems [8], which rely on accurate environmental priors to enhance generalization and robustness. Especially, multi-robot 3D mapping is essential for large-scale and complex tasks, where robot teams provide greater coverage and robustness compared to a single agent. In scenarios such as search and rescue [9], forest monitoring [10], and subterranean exploration [11], accurate point cloud fusion is vital to realize cooperation among multiple robots under complex and GPS-denied conditions.

Rapid development of LiDAR-based Simultaneous Localization and Mapping (SLAM) techniques [12]–[17] has enabled individual robots to perform real-time localization and point cloud map construction. However, most of the existing lightweight SLAM systems [14]–[16] for onboard computers are based on direct LiDAR-Inertial-Odometry methods. These methods prioritize pose estimation over map construction to achieve real-time performance. However, the lack of feature extraction and matching in these approaches results in maps with blurred details and poor geometric accuracy. Therefore, the quality of the point cloud maps generated by onboard computers based on lightweight methods requires further improvement.

To address mapping challenges and enhance the geometric accuracy of point cloud maps, various strategies have been proposed. Among them, one category of approaches leverages pose graph optimization (PGO) [18] to handle loop scenarios. Loop closure constraints detected by loop detection methods [19]–[22] are added to the pose graph along with odometry constraints to eliminate accumulated drift and improve global consistency of the map. However, PGO-based methods only use the point cloud map to extract descriptors during loop detection and ignore the structure and geometric information. ***Therefore, although PGO-based methods can improve global consistency based on loop closures, it cannot guarantee an accurate map without divergence and blurring.*** (We provide examples in Fig. 9 (b) and (c).) In contrast, another type of map refinement method based on bundle adjustment (BA) has demonstrated impressive capabilities to improve the accuracy of point cloud maps in recent years [23]–[26]. These approaches utilize the structural characteristics of the point cloud map and improve its quality by minimizing the geometric residuals between feature points and feature structure. ***Although BA-based methods are capable of generating high-quality maps, their reliance on temporal sequences may prevent them from fully leveraging loop closure information, potentially resulting in poor map consistency.*** (Examples are illustrated in Fig. 8 (b) and (c).) The dependence on the temporal ordering of poses also limits the applicability of BA-based methods to multi-robot systems.

In addition to the above mapping approaches, to achieve larger-scale map reconstruction and multi-robot collaboration, several multi-robot SLAM systems [27], [28] and offline map fusion techniques [29]–[34] have shown promising results in recent years. In general, these methods focus on two aspects: removing outliers in loop closure data and achieving

map fusion through multi-robot PGO. However, PGO-based methods are fundamentally the same in single-robot and multi-robot applications as they both use odometry constraints and loop closure constraints for map refinement. As discussed in the previous paragraph, these approaches neither directly utilize the geometric information in multi-robot maps nor fully leverage loop closure data. Consequently, they only ensure the basic alignment and consistency of the global map but fail to generate high-quality point cloud maps. In particular, they often suffer from misalignment in regions with overlap between robots, as demonstrated in Fig. 9 (b) and (c).

In general, PGO-based and BA-based methods cannot guarantee both map accuracy and global consistency in multi-robot scenarios with a large number of loops. Building upon the strengths of existing methods and aiming to address their limitations, we perform an in-depth analysis of previous multi-robot SLAM systems and map merging techniques. The key issue of multi-robot mapping is to solve the ***divergence*** and ***blurring*** problems of submaps, while maintaining the ***global consistency*** of the whole map. We argue that the merging of point cloud maps is essentially a map-driven local point cloud registration problem rather than a pose-dominated pose optimization problem. Building upon this perspective, we identify two essential challenges for achieving effective multi-robot point cloud map fusion: **(1)** Map fusion should be treated as a map-driven registration problem rather than a pose-centric optimization task such as PGO. **(2)** Loop closure information should be fully exploited for local registration, not only to ensure global consistency but also to improve local accuracy.

To overcome the above challenge, we propose **LEMON-Mapping**, a **Loop-Enhanced large-scale Multi-session map merging and Optimization** framework that achieves globally consistent and geometrically accurate point cloud mapping. Our framework re-examines the ability of loop closure and reasonably enhances its utilization by two aspects. First, we innovatively propose a loop recall mechanism, providing more sufficient information and optimization starting points for subsequent spatial BA. (We provide an example in Fig. 14.) Second, by leveraging sufficient valid loop closures, our proposed spatial BA can effectively deal with multi-robot maps, whereas traditional BA methods fail to address. The spatial BA is performed within a local spatial window around the loop, effectively leveraging multiple observations and abundant constraints from different robots. It directly optimizes the poses from different robots equally within the spatial window; thus, the poses of multiple robots matching the same geometric features (planes, lines) are adjusted simultaneously, generating a locally accurate and consistent map. Although spatial BA improves local map accuracy at loop closures, it lacks the capability for global map fusion. To address this problem, we propose a reasonable PGO-based method which effectively combines local BA constraints and odometry constraints to transfer the local alignment achieved by our spatial BA to the whole map, thereby achieving global consistency and accuracy. We conduct a series of extensive experiments, and the results demonstrate the high mapping accuracy and strong scalability of our multi-session map fusion approach, as shown in Fig. 1 and Fig. 17.

In general, the main contributions of this work can be summarized as follows:

- A scalable multi-session point cloud map merging and optimization framework is designed, which integrates two-step PGO with spatial BA. The method achieves high-precision 3D mapping in large-scale and multi-robot scenarios.
- A novel spatial BA that can be used for multi-robot mapping is introduced, which operates on loop-based spatial windows to fully utilize loop constraints and jointly optimize multi-robot poses equally. This improves the local accuracy and eliminates the serious map divergence.
- A pose graph optimization scheme with reasonable sparsification of BA constraints is developed which effectively propagates local accuracy to the global scale, improving both consistency and efficiency.
- A robust loop closure processing pipeline is designed, including outlier rejection and false negative recall, which enhances the reliability and completeness of loop constraints used for map fusion.

II. RELATED WORKS

A. Single Map Maintenance and Optimization

Single map maintenance and optimization have been extensively studied, with existing approaches falling broadly into two categories: pose graph optimization (PGO) and bundle adjustment (BA) based methods. PGO remains a widely adopted back-end in LiDAR SLAM systems [35]–[37]. A single-robot pose graph is constructed to refine initial pose estimates from odometry by utilizing the constraints of loop closures. However, traditional PGO frameworks focus on pose consistency rather than directly optimizing point cloud structure, leading to map with poor geometric quality, which limits their effectiveness in generating globally consistent maps for precision-demanding applications.

In contrast, BA-based methods jointly optimize scan poses and scene features by minimizing the geometric residuals of matched primitives, offering improved map accuracy. BALM [23] introduces a closed-form solution for feature parameters to reduce computational complexity, while BALM2 [24] extends this with point clustering and a more efficient second-order solver. HBA [25] adopts a hierarchical BA strategy followed by top-down PGO, significantly improving the quality and consistency of the overall map. PSS-BA [26] introduces quadratic surface modeling for point cloud maps and uses progressive smoothing iteration to optimize map quality. RSO-BA [38] improves robustness via a second-order estimator integrated with a robust kernel function. However, traditional bundle adjustment approaches rely on time-based sliding windows and fail to incorporate long-range spatial constraints introduced by loop closures. These constraints of loop closures, though temporally distant, are spatially adjacent with sharing of the same geometric features and crucial for improving local consistency. Generally, the existing BA methods mentioned above do not solve this problem well.

B. Multi-Session Map Merging

Multi-session map merging aims to integrate submaps from multiple agents either with or without initial pose estimates into a unified, globally consistent map. Some multi-robot exploration frameworks such as SMMR-Explore [39] and MR-GMMEExplore [40] address multi-robot exploration and map fusion under communication limits. However, the former is limited to 2D point cloud submaps while the latter heavily suffers from loss of geometric details caused by Gaussian Mixture Models (GMMs) based submaps. And their lack of pose graph optimization (PGO) prevents them from achieving global consistency. Therefore, these multi-robot exploration frameworks are both unsuitable for generation of large-scale geometric accurate point cloud map. SegMap [41] extracts semantic features from 3D point clouds to estimate 6-DoF transformations and utilizes incremental pose graph optimization to achieve map fusion. However, it heavily depends on accurate semantic segmentation, limiting its applicability to structured urban environments. AutoMerge [30] introduces a city-scale merging framework that avoids relying on initial alignment, but its performance degrades in complex environments due to the lack of elimination of loop outliers. PCM [31]–[33] selects reliable loop sequences through pairwise consistency maximization, but struggles on datasets with high loop error rates. It is also severely affected by the accumulated drift of odometry in large-scale environments. LAMM [29] enhances place recognition by removing dynamic objects through M-Detector [42] and using the robust loop detection method BTC [20], yet it only merges map by pose graph optimization, which may suffer from local divergence in the overlapping areas of the multi-robot map.

Outlier rejection constitutes another critical concern in multi-session map merging system. This is because large-scale environments often contain repetitive structures, increasing the risk of incorrect data association and false positive loop closures. Thus, a robust rejection of outliers is essential. RANSAC [43] remains a standard solution for fitting models despite outliers, but its effectiveness diminishes under high outlier ratios and in cases lacking strong priors. PCM [31], adopted in systems such as DCL-SLAM [27] and Disco-SLAM [28], leverages pairwise geometric consistency for loop validation. Although PCM is more robust under uncertain initial estimates, its accuracy can be compromised by unreliable odometry, resulting in over-rejection of valid loop closures.

III. SYSTEM OVERVIEW

A. Problem Formulation

Our objective is to construct an accurate and globally consistent map by merging multiple submaps generated from different robots. In LiDAR-based SLAM systems [12]–[14], [16], each scanned point cloud is considered rigidly connected to the LiDAR, and the global point cloud map is constructed by transforming each scan into the world coordinate system utilizing its corresponding pose. Therefore, the quality of the point cloud map is strictly associated with the sensor pose at the time of acquisition.

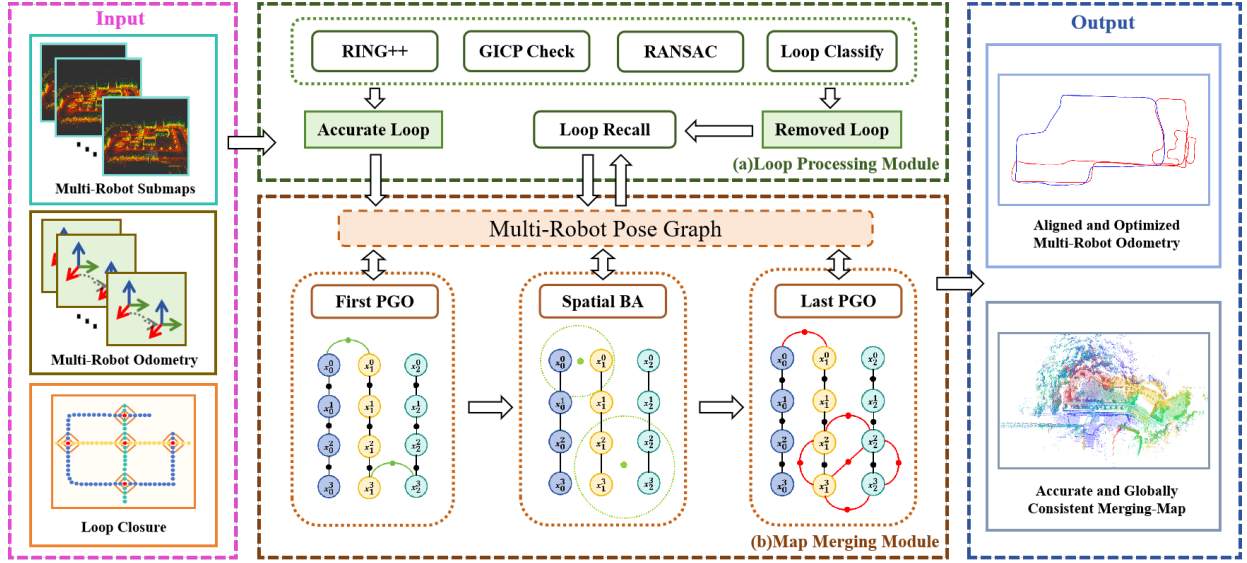


Fig. 2: The framework of our method. Our framework takes multi-robot submaps, odometry and loop closures as input, and generates accurate and globally consistent merged map. (a) The Loop Processing Module, which removes outliers and recall false negative loops, providing correct and sufficient loops for the spatial BA. (b) The Map Merging Module, which achieves accurate and globally consistent multi-robot map merging through two-step PGO and spatial BA. Each of the three steps in map merging module interacts closely with the multi-robot pose graph.

In a multi-robot system, the set of pose sequences can be denoted as $S_N = \{s_1, s_2, \dots, s_N\}$, where each sequence s_k corresponds to a separate robot. Different sequences such as s_i and s_j , which have different starting points and initial orientations, lack prior information on relative pose transformation. To achieve accurate and effective map fusion, it is essential to estimate the relative transformations between these trajectories and optimize the poses using constraints in overlapping regions to improve geometric consistency.

The problem can therefore be formulated as finding a set of optimized pose sequences $S_N^* = \{s_1^*, s_2^*, \dots, s_N^*\}$, such that all trajectories are aligned to a common coordinate system (typically s_1^*), and the resulting global map exhibits minimal divergence in overlapping regions while maintaining overall consistency.

B. Framework of LEMON-Mapping

An overview of the proposed framework is illustrated in Fig. 2, which is designed for multi-robot point cloud map merging and optimization. Our system consists of two main components: the Loop Processing Module (Fig. 2 (a)) detailed in Section IV and the Map Merging Module (Fig. 2 (b)) described in Section V and Section VI.

The Loop Processing Module takes the odometry and submaps from multiple agents along with raw loop closure data (including both self-loops and inter-loops) as input. It removes erroneous loops to output the correct loops. The Map Merging Module then uses the correct loops to optimize the multi-robot trajectories through three steps: spatial bundle adjustment (BA) and two pose graph optimization (PGO) steps. First, a preliminary global pose estimate is obtained using a conventional multi-robot PGO (FPGO), which serves as an initial guess and prior information for loop recall in the Loop

Processing Module. Next, a spatial BA is performed within local regions around the loops to fully exploit the geometric constraints from overlapping observations. Finally, the last PGO (LPGO) step refines the trajectories using constraints of our spatial BA and constraints of odometry. As the two PGO steps share a similar formulation, they are jointly described in Section VI, while the proposed spatial BA is discussed separately in Section V.

IV. LOOP PROCESSING MODULE

The Loop Processing Module is a fundamental component that supports the spatial bundle adjustment described in Section V and the two-step pose graph optimization in Section VI. It consists of three submodules: Outlier Rejection, Loop Classification, and Loop Recall. This module processes both self-loops and inter-loops from multiple robots, which are initially detected by the robust and state-of-the-art (SOTA) loop detection method RING++ [21].

A. Outlier Rejection

Loop closure data may contain incorrect constraints, which seriously affects the accuracy of coordinate alignment and pose graph optimization. To solve this problem, we design an outlier rejection method using the Generalized Iterative Closest Point (GICP) [45] combined with Random Sample Consensus (RANSAC) [43].

The process begins with Statistical Outlier Removal (SOR) [46], which filters the noise from the raw point clouds by analyzing the statistical distribution of point-to-neighbor distances. GICP is then employed to align the filtered point clouds, initialized using the transformation estimated by RING++. To validate the alignment, correspondences between

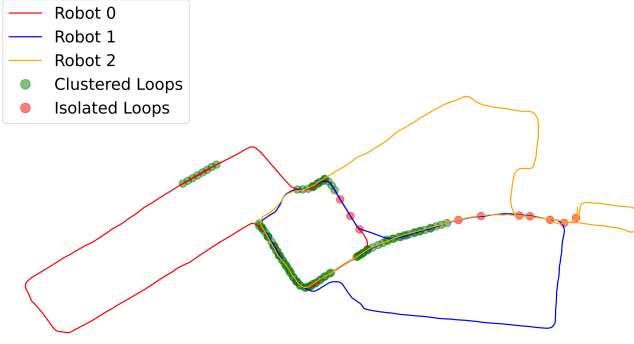


Fig. 3: An example of loop classification in S3E Campus_1 dataset [44], the trajectories of multi-robot and the two types of loops are shown in the figure.

the two point clouds are refined using a RANSAC-based correspondence rejection method, which discards outliers inconsistent with rigid transformations. The loop is accepted only if the number of inlier correspondences exceeds a predefined threshold. Additionally, the GICP fitness score is used as a quantitative metric for alignment quality, and alignments with poor scores or non-convergence are discarded. This two-stage filtering approach significantly enhances the reliability of loop closures by ensuring that only geometrically valid alignments are retained.

B. Loop Classification

Following the outlier rejection process, the remaining valid loop closures are classified to facilitate efficient spatial bundle adjustment. Specifically, we categorize loops into two types based on their spatial distribution: clustered loops and isolated loops.

We implement a region-growing algorithm based on Breadth-First Search (BFS). Starting from the spatial center of each loop, the algorithm incrementally searches for nearby loops within a predefined radius. If neighboring loops are found, the search expands from their centers, continuing recursively until no additional nearby loops are detected. Loops that form such spatial clusters are labeled as clustered loops, while those without adjacent loops are marked as isolated loops. An illustrative example of this classification process in the S3E Campus_1 dataset is shown in Fig. 3.

C. Loop Recall

Since the outlier rejection step adopts strict criteria to ensure robustness and accuracy, it may inadvertently discard valid loop closures, thereby resulting in false negatives. These missing constraints between robot trajectories hinder the correction of divergence in overlapping areas, potentially degrading the consistency of the fused multi-robot map.

To address this issue, we propose a loop recall mechanism to recover previously discarded valid loops. After the first pose graph optimization (Section VI) aligns all robot trajectories into the same coordinate system using valid loops, the updated

poses are used to re-evaluate previously rejected loops. If the Euclidean distance between the associated poses is below a threshold (2 meters in our system), the loop is recalled for further optimization. This lightweight and distance-based strategy effectively recovers useful constraints and enhances subsequent optimization. Fig. 14 demonstrates the impact of loop recall on the merging of point cloud maps.

V. SPATIAL BUNDLE ADJUSTMENT

Traditional BA methods [23]–[25] jointly optimize sequential poses within a sliding window of temporally ordered data. However, they struggle with scenarios involving revisiting the same location over long time spans, making them unsuitable for multi-robot systems. Different from previous BA methods, our spatial BA jointly optimizes poses from different robots in local spatial windows simultaneously. This design effectively reduces map divergence across sessions, as demonstrated in Fig. 4 and Fig. 9. Our approach specifically focuses on loop closure regions for two reasons: (1) these regions often exhibit significant spatial overlap among submaps and suffer from high divergence; (2) they contain multiple observations of the same geometric structure from different times or agents, offering rich constraints for accurate joint optimization.

The spatial BA is performed on all correct loops after the FPGO and the loop recall step. All available loops are divided into two categories based on the principle of loop classification in Section IV. The poses of all robots are built into a kd-tree using their spatial positions for faster searching. For each loop, we define a spherical spatial window centered at the midpoint between the two poses involved in the loop closure. We employ a radius-based search using the pose kd-tree to efficiently retrieve the poses in the spherical region. These selected poses form the local optimization window for the spatial BA, ensuring that only the spatially correlated data around the loop are refined. Depending on the type of loop closure classified in Section IV, we propose two forms of spatial bundle adjustment designed for their respective characteristics to refine multi-robot maps. For isolated loops, we apply our developed spatial diffusion bundle adjustment (DBA). For clustered loops, a spatial variant of hierarchical bundle adjustment (HBA) is proposed.

A. DBA for Isolated Loops

Our DBA is built upon BALM2 [24]. However, it relies on accurate initial plane estimation, which may fail to converge reliably when the same plane observed by different robots appears misaligned due to map divergence. In such cases, the same structure may be voxelized as separate multi-layer planes (shown in Fig. 4 (a)), making direct optimization invalid. To address this, we partition the poses within the spatial window by robot into distinct clusters, each with its own point cloud, as illustrated in the left part of Fig. 4. We then apply GICP [45] to roughly align these clusters, reducing plane stratification and enabling reliable plane estimation for subsequent optimization. However, GICP only provides a coarse alignment of multi-robot submaps rather than true fine fusion. It simply adjusts the relative transformation of the assembled point cloud submaps

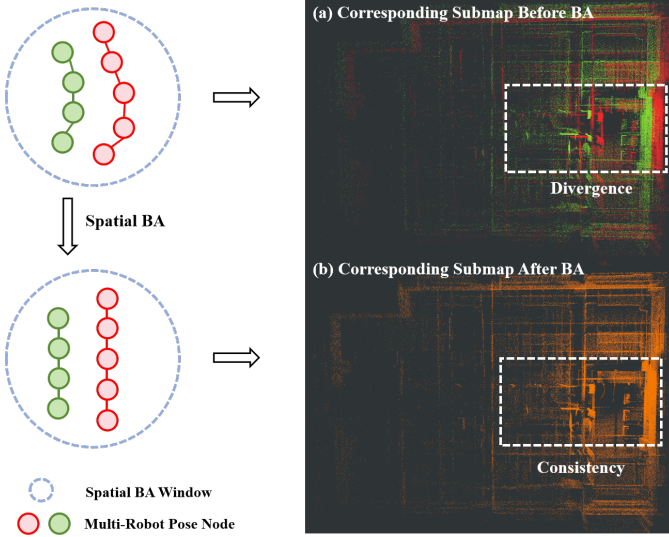


Fig. 4: The red and green nodes in the left part show the trajectories of different robots at certain loop. (a) and (b) show the submaps before BA optimization and after BA optimization respectively. Our spatial bundle adjustment significantly reduces divergence and generates locally consistent map.

of different pose sequence, but it does not optimize multi-robot poses at the frame level or refine point cloud at the single scan level. After the initial pose graph optimization incorporating an isolated loop, the poses near the loop center are effectively aligned and optimized. However, as the distance from the loop center increases, the influence of loop constraints in FPGO gradually weakens, leading to less accurate optimization and alignment of distant poses. Therefore, for an isolated loop, the key point is to figure out how large its influence range is and align the surrounding affected poses. We develop DBA to tackle this issue.

We begin with a set of LiDAR poses confined to a small range near an isolated loop. We then gradually expand the set by diffusely incorporating additional LiDAR poses from increasingly wider ranges. The key principle is that, within the loop influence region, poses located further from the loop center contribute less accurately to the representation of the loop's features. The details of DBA are as follows.

For simplicity, we use the following notation, and we refer the reader to [24] for more detailed information. In this paper, feature is seen as *point clusters*, the point cluster for the i -th feature is denoted by set $\mathcal{C}_i = \{\mathbf{p}_{ijk} \in \mathbb{R}^3 | j = 1, \dots, M_p, k = 1, \dots, N_{ij}\}$, where M_p is the number of poses, and the corresponding *point cluster coordinate* $\mathfrak{R}(\mathcal{C})$ is defined as:

$$\begin{aligned} \mathfrak{R}(\mathcal{C}_i) &\triangleq \sum_{j=1}^{M_p} \sum_{k=1}^{N_{ij}} \begin{bmatrix} \mathbf{p}_{ijk} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{ijk}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_i & \mathbf{v}_i \\ \mathbf{v}_i^T & N_i \end{bmatrix} \in \mathbb{S}^{4 \times 4} \\ \mathbf{P}_i &= \sum_{j=1}^{M_p} \sum_{k=1}^{N_{ij}} \mathbf{p}_{ijk} \mathbf{p}_{ijk}^T, \quad \mathbf{v}_i = \sum_{j=1}^{M_p} \sum_{k=1}^{N_{ij}} \mathbf{p}_{ijk} \end{aligned} \quad (1)$$

A BA formulation with determining LiDAR poses $\mathbf{T} = (\mathbf{T}_1, \dots, \mathbf{T}_{M_p})$, and feature parameters $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_{M_f})$,

where M_f is the number of features, starts with the optimization problem:

$$\min_{\mathbf{T}, \boldsymbol{\pi}} \left(\sum_{i=1}^{M_f} c(\boldsymbol{\pi}_i, \mathbf{T}) \right). \quad (2)$$

When using plane features, each cost item standards for the squared Euclidean distance from a plane, which can be proved (BALM2 [24]) that it is in the form of

$$\begin{aligned} c_i(\mathbf{T}) &\triangleq c(\boldsymbol{\pi}_i, \mathbf{T}) = \lambda_3 \left(\mathbf{A} \left(\sum_{j=1}^{M_p} \mathbf{T}_j \mathbf{C}_{f_{ij}} \mathbf{T}_j^T \right) \right) \\ \mathbf{A}(\mathbf{C}_i) &\triangleq \frac{1}{N_i} \mathbf{P}_i - \frac{1}{N_i^2} \mathbf{v}_i \mathbf{v}_i^T, \quad \mathbf{C}_i = \begin{bmatrix} \mathbf{P}_i & \mathbf{v}_i \\ \mathbf{v}_i^T & N_i \end{bmatrix} \in \mathbb{S}^{4 \times 4} \end{aligned} \quad (3)$$

with $\lambda_3(\mathbf{A})$ as the 3rd largest eigen value of matrix function $\mathbf{A}$, $\mathbf{C}_{f_{ij}} \in \mathbb{R}^{4 \times 4}$ being a pre-computed matrix where

$$\begin{aligned} \mathbf{C}_{f_{ij}} &= \begin{bmatrix} \mathbf{P}_{f_{ij}} & \mathbf{v}_{f_{ij}} \\ \mathbf{v}_{f_{ij}}^T & N_{ij} \end{bmatrix} \\ \mathbf{P}_{f_{ij}} &= \sum_{k=1}^{N_{ij}} \mathbf{p}_{f_{ijk}} \mathbf{p}_{f_{ijk}}^T, \quad \mathbf{v}_{f_{ij}} = \sum_{k=1}^{N_{ij}} \mathbf{p}_{f_{ijk}} \end{aligned} \quad (4)$$

Leveraging the above definition, where *point clusters* forming plane features, we could solve the problem with an optimal update $\Delta \mathbf{T}^*$ using the Levenberg-Marquardt (LM) algorithm:

$$\Delta \mathbf{T}^* = -(\mathbf{H} + \mu \mathbf{I})^{-1} \mathbf{J}^T, \quad (5)$$

Here μ is the damping parameter, while $\mathbf{J}$ and $\mathbf{H}$ are the Jacobian and Hessian of the cost function, respectively.

In DBA, we categorize the poses into D groups based on their distances from the loop. Each group contains a set of poses. The number of poses in the d_i -th group is denoted as $d_i M_p$ (for $i = 0, \dots, D$). In the following part, we first demonstrate that in this *incremental* form of BA, the optimal update formulation remains the same as performing BA jointly on all features. We then prove that the algorithmic complexity is smaller compared to the traditional BA methods, and our approach achieves a relatively higher confidence level which should perform better in practice.

In the d_i -th diffusion process, we treat all poses that participated in the previous i diffusion process as accurate and freeze their gradients, meaning they are not optimized furthermore. Consequently, when optimizing the d_i -th process, we can categorize the poses into two groups ${}^0\mathbf{T}$ and ${}^1\mathbf{T}$ with numbers of $M_{p_0} \triangleq \sum_{k=0}^{i-1} d_k M_p$ and $M_{p_1} \triangleq d_i M_p$.

Assumption 1. Small range LiDAR poses M_{p_0} that participate in the BA have significantly smaller measurement noise covariance $\Sigma_{c_{f_{ij}}}$ than that associated with the outer pose M_{p_1} .

Lemma 1 (Continuity in Eigenvalue Functions). If $\mathbf{A}(x)$ is a continuous real symmetric matrix function (i.e., $\mathbf{A}(x)$ is continuous and is symmetric for every x), then its eigenvalue function $\lambda_i(x)$ is also continuous.

Proof. See Appendix (VIII-A). $\square$

According to the definition of matrix $\mathbf{A}$, we could obviously get $\mathbf{A}(\mathbf{C}) \in \mathbb{S}^{3 \times 3}$. Hence, based on Lemma 1, we could easily

partition conformably the LM Jacobian and Hessian for DBA as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{00} & \mathbf{H}_{01} \\ \mathbf{H}_{10} & \mathbf{H}_{11} \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_0 \\ \mathbf{J}_1 \end{bmatrix}$$

Using LM optimization in Eq. 5, we could respectively derive the update of pose for group M_{p_1} for the joint BA and DBA:

$$\begin{aligned} \Delta^1 \mathbf{T}_{\text{joint}} &= -\mathbf{S}^{-1}(\mathbf{J}_1 - \mathbf{H}_{10}(\mathbf{H}_{00} + \mu\mathbf{I}_0)^{-1}\mathbf{J}_0) \\ \Delta^1 \mathbf{T}_{\text{DBA}} &= -(\mathbf{H}_{11} + \mu\mathbf{I}_1)^{-1}\mathbf{J}_1 \end{aligned} \quad (6)$$

where $\mathbf{S} = \mathbf{H}_{11} + \mu\mathbf{I}_1 - \mathbf{H}_{10}(\mathbf{H}_{00} + \mu\mathbf{I}_0)^{-1}\mathbf{H}_{01}$ is the Schur's complement.

The joint optimization can be viewed as an update method that refines the coupled terms $\mathbf{H}_{10} = \mathbf{H}_{01}^T$ for the DBA method. Hence from Eq. 7, we could define two refinement rates about the Hessian ($r_{\mathbf{H}}$) and the Jacobian ($r_{\mathbf{J}}$) as ratio of the coupling contribution to the main term:

$$\begin{aligned} r_{\mathbf{H}} &= \frac{\|\mathbf{H}_{10}(\mathbf{H}_{00} + \mu\mathbf{I}_0)^{-1}\mathbf{H}_{01}\|}{\|\mathbf{H}_{11} + \mu\mathbf{I}_1\|}, \\ r_{\mathbf{J}} &= \frac{\|\mathbf{H}_{10}(\mathbf{H}_{00} + \mu\mathbf{I}_0)^{-1}\mathbf{J}_0\|}{\|\mathbf{J}_1\|} \end{aligned} \quad (7)$$

The smaller these two refinement rates are (i.e., closer to 0), the less significant the coupled terms become relative to the main terms. According to Assumption 1, consequently, the information matrix (Hessian approximation) $\mathbf{H}_{00}$, which represents the certainty of the inner poses M_{p_0} , is assumed to be strong (i.e., possessing large singular values), formally implying that the norm of its pseudo-inverse is relatively small:

$$\|(\mathbf{H}_{00} + \mu\mathbf{I}_0)^{-1}\| \approx 0.$$

which in turn drives $r_{\mathbf{H}}$ and $r_{\mathbf{J}}$ towards zero. Consequently, the bias of the two method is minimized, and the high-speed DBA method closely approximates the high-accuracy joint optimization of traditional BA.

Lemma 2 (Computational Complexity Reduction of DBA). Let the total number of poses be $M = \sum_{i=0}^{D-1} m_i$, where $m_i \triangleq {}^{d_i}M_p$ denotes the number of poses in the i -th diffusion group, for simplicity. A joint BA optimization over all poses has complexity

$$O(M_f M + M_f M^2 + M^3).$$

In contrast, DBA performs BA incrementally and its total complexity is

$$O\left(M_f \sum_i m_i + M_f \sum_i m_i^2 + \sum_i m_i^3\right),$$

which is strictly no larger and generally substantially smaller. In particular, when the diffusion groups are of comparable size, i.e., $m_i \approx M/D$, the quadratic and cubic terms are reduced by factors of approximately D and D^2 , respectively.

Proof. See Appendix (VIII-B). $\square$

The above proof uses a maximum of D diffusion steps, while in practice, DBA is used when having a strong confi-

dence of inner group poses for a small number of times far below D .

Let us estimate the confidence level of the estimated pose using the covariance estimation. Denote $\mathbf{C}_f = \{\mathbf{C}_{f_{ij}}\}$, $\delta\mathbf{C}_f = \{\delta\mathbf{C}_{f_{ij}}\}$, and $\mathbf{T}^*$ as the converged solution using the measured cluster $\mathbf{C}_f$, according to BALM2, we get:

$$\delta\mathbf{T}^* = \mathbf{H}^{-1} \frac{\partial \mathbf{J}^T(\mathbf{T}^*, \mathbf{C}_f)}{\partial \mathbf{C}_f} \delta\mathbf{C}_f \sim \mathcal{N}(\mathbf{0}, \Sigma_{\delta\mathbf{T}^*}), \quad (8)$$

$$\begin{aligned} \Sigma_{\delta\mathbf{T}^*} &= \mathbf{H}^{-1} \frac{\partial \mathbf{J}^T(\mathbf{T}^*, \mathbf{C}_f)}{\partial \mathbf{C}_f} \Sigma_{\delta\mathbf{C}_f} \frac{\mathbf{J}(\mathbf{T}^*, \mathbf{C}_f)}{\partial \mathbf{C}_f} \mathbf{H}^{-T} \\ &= \mathbf{H}^{-1} \left(\sum_{i=1}^{M_f} \sum_{j=1}^{M_p} \mathbf{L}_{ij} \Sigma_{c_{f_{ij}}} \mathbf{L}_{ij}^T \right) \mathbf{H}^{-T}. \end{aligned} \quad (9)$$

where

$$\Sigma_{c_{f_{ij}}} = \sum_{k=1}^{N_{ij}} \mathbf{B}_{f_{ijk}} \Sigma_{p_{f_{ijk}}} \mathbf{B}_{f_{ijk}}^T \succeq 0, \quad \mathbf{B}_{f_{ijk}} \in \mathbb{R}^{9 \times 3} \quad (10)$$

$$\frac{\partial \mathbf{J}^T}{\partial c_{f_{ij}}} = \begin{bmatrix} \vdots \\ \frac{\partial(\mathbf{J}^p)^T}{\partial c_{f_{ij}}} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \mathbf{L}_{ij}^p \\ \vdots \end{bmatrix} \triangleq \mathbf{L}_{ij} \in \mathbb{R}^{6M_p \times 9} \quad (11)$$

Lemma 3 (Covariance ordering: DBA vs joint BA). Similar to the partition above, let the Jacobian-derivative blocks w.r.t. feature clusters be

$$\mathbf{L}_{ij} = \begin{bmatrix} {}^0\mathbf{L}_{ij} \\ {}^1\mathbf{L}_{ij} \end{bmatrix}, \quad \Sigma_{c_{f_{ij}}} \succeq 0.$$

Define the joint-pose covariance perturbation due to cluster noise as

$$\Sigma_{\delta\mathbf{T}^*}^{\text{joint}} = \mathbf{H}^{-1} \left(\sum_{i=1}^{M_f} \sum_{j=1}^{M_p} \mathbf{L}_{ij} \Sigma_{c_{f_{ij}}} \mathbf{L}_{ij}^T \right) \mathbf{H}^{-T},$$

and let $\Sigma_{11}^{\text{joint}}$ be its bottom-right block (covariance of block 1 under joint BA). For DBA (block 0 frozen, optimizing only block 1) define

$$\Sigma_1^{\text{DBA}} = \mathbf{H}_{11}^{-1} \left(\sum_{i=1}^{M_f} \sum_{j \in \text{group1}} {}^1\mathbf{L}_{ij} \Sigma_{c_{f_{ij}}} {}^1\mathbf{L}_{ij}^T \right) \mathbf{H}_{11}^{-T}.$$

Then, under the usual invertibility assumptions on $\mathbf{H}$ and $\mathbf{H}_{00}$, the following PSD ordering holds:

$$\Sigma_1^{\text{DBA}} \preceq \Sigma_{11}^{\text{joint}}.$$

Consequently, freezing well-measured inner poses (block 0) yields a covariance for the active poses that is no larger (and typically smaller) than the covariance those same poses would have under a full joint BA. Note that the inequality follows without assuming inner poses are better measured.

Proof. See Appendix (VIII-C). $\square$

B. HBA for Clustered Loops

For two trajectories from different robots with significant spatial overlap, loop closures tend to appear densely, and the associated point cloud maps of adjacent loop closures

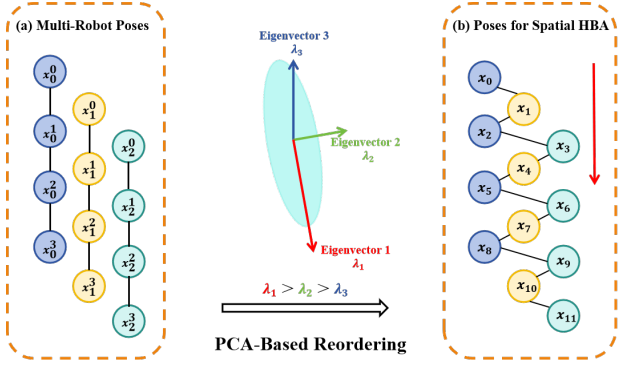


Fig. 5: (a) shows the multi-robot poses in a spatial window. (b) shows the reordered poses using PCA for spatial HBA.

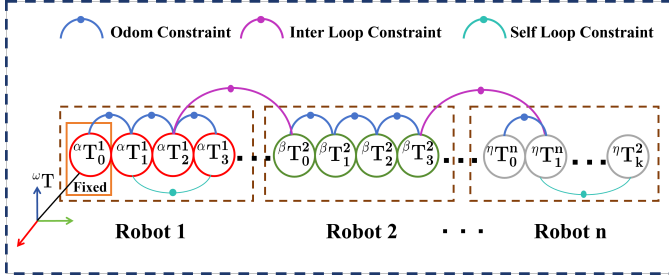


Fig. 6: The first pose graph optimization, with constraints including odometry and loop closure. The first pose of the first robot is fixed as the world coordinate.

often share large common areas, thus forming a clustered loop groups. In such clustered loop groups, it is crucial to jointly process the loops within the entire overlapping region to maintain local consistency. If we optimize the poses by processing loop closures sequentially and independently, this may introduce inconsistency or even divergence in overlapping regions due to contradictory constraints. To address this, we adapt and extend HBA into a multi-robot framework for clustered loop scenarios.

The original HBA relies on temporally ordered poses from a single robot, assuming that adjacent poses share common point cloud features. However, in our multi-robot setting, poses selected via kd-tree radius search within a loop cluster are unordered and from different robots. To address this, we use Principal Component Analysis (PCA) [47] to analyze the spatial distribution of selected poses and reorder them along the principal axis defined by the largest eigenvalue, as illustrated in Fig. 5. This spatial reordering ensures that neighboring poses in the optimization sequence are also spatially close, promoting shared features and effective constraints. Once reordered, all poses are processed consistently in a unified optimization window, allowing the enhanced HBA to accurately align the geometry across overlapping regions from multiple robots.

VI. TWO-STEP POSE GRAPH OPTIMIZATION

The two-step pose graph optimization is used to align and refine the global structure of multi-robot trajectories. The details of each step are as follows.

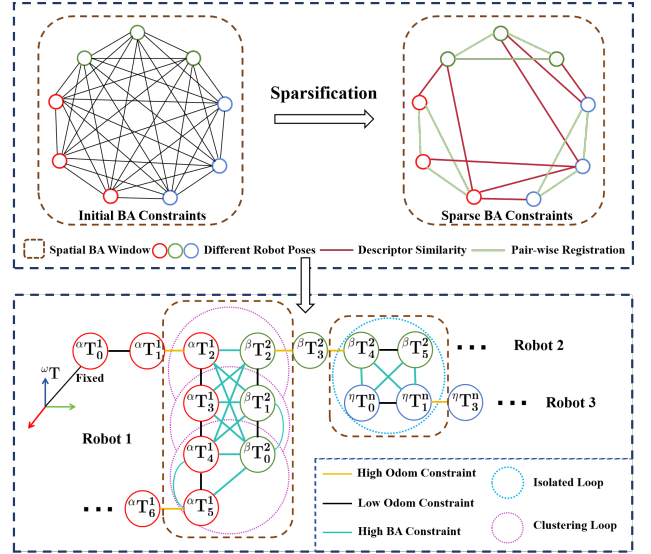


Fig. 7: The upper part of the picture shows the sparsification of BA constraints in a spatial window at loop. The sparse BA constraints are added to pose graph below along with odometry constraints with different weights.

A. First Pose Graph Optimization

Following loop processing, we perform the first pose graph optimization (FPGO) to coarsely align the trajectories of all robots. We construct a centralized pose graph shown in Fig. 6, in which each robot's pose sequence is included as a subgraph. The graph incorporates three types of constraint: odometry, self-loops, and inter-loops. The first pose of the first robot is fixed to define the origin point of the world coordinate, ensuring that all other robot trajectories are aligned to this global reference frame. With this initial alignment established, we acquire a rough estimate of the relative positions among all robot trajectories, which enables us to recover the valid but previously rejected loops (in Section IV). The recalled loops are added back to the correct set of loops and used as constraints in the FPGO. This process containing optimization and recall is repeated until no new loops are recalled.

B. Last Pose Graph Optimization

Although spatial BA (Section V) improves the accuracy of local pose near loop closures, it does not propagate this accuracy globally and may break the continuity of odometry, resulting in even worse global consistency. To address this issue, we develop the last pose graph optimization (LPGO), and the details are as follows.

We design the LPGO with two types of constraints: odometry constraints to maintain trajectory smoothness, and sparsified BA-based constraints to preserve refined local structures (see Fig. 7). For each robot, odometry constraints are added between adjacent poses to preserve temporal continuity. In cases where a BA-optimized pose is adjacent to an unoptimized one, a high-weight odometry constraint is applied to mitigate potential discontinuities introduced by local optimization. In contrast, lower weights are assigned between two unoptimized poses to allow greater global adjustment flexibility.

Additionally, pose pairs within the spatial BA window that exhibit strong geometric overlap and reliable correspondence are assigned high-weight BA constraints to preserve local accuracy. The specific selection criteria for these constraints are detailed in the following paragraphs.

We use two different metrics for constraint selection between pose pairs of different robots and pose pairs of the same robot. For pose pairs across different robots, we evaluate the similarity of their descriptors obtained from RING++ [21] and add BA constraints only when the similarity exceeds a specified threshold. For pose pairs within the same robot, we apply pair-wise registration techniques from [48] to selectively retain constraints between poses that significantly constrain each other. The sparsification process using pair-wise registration is as follows.

For an edge $\mathbf{E}_k$ connecting two pose nodes belonging to the same robot, let $\mathbf{E}_k^0$ and $\mathbf{E}_k^1$ denote the indices of the corresponding poses. The residual ϵ_k and its associated covariance matrix Ω_k between the two poses can be estimated using a pair-wise registration approach. For clarity, we denote the poses at $\mathbf{E}_k^0$ and $\mathbf{E}_k^1$ as $(\mathbf{R}_{L_0}, \mathbf{t}_{L_0})$ and $(\mathbf{R}_{L_1}, \mathbf{t}_{L_1})$, respectively. The relative transformation from frame L_0 to L_1 is computed as $\mathbf{R}_{L_1}^{L_0} = \mathbf{R}_{L_0}^\top \mathbf{R}_{L_1}$ and $\mathbf{t}_{L_1}^{L_0} = \mathbf{R}_{L_0}^\top (\mathbf{t}_{L_1} - \mathbf{t}_{L_0})$. Based on nearest neighbor matching, we obtain point-to-point correspondences $\{(\mathbf{P}_u^{L_0}, \mathbf{P}_u^{L_1})\}_{u=1}^U$, where U denotes the total number of matched pairs. These correspondences are used to formulate the registration residual function ϵ_k^{reg} associated with edge $\mathbf{E}_k$.

$$\epsilon_k^{\text{reg}} = \sum_{u=1}^U (\mathbf{R}_{L_1}^{L_0} \mathbf{P}_u^{L_1} + \mathbf{t}_{L_1}^{L_0} - \mathbf{P}_u^{L_0}), \quad (12)$$

The Jacobian of ϵ_k^{reg} respect to the two relative poses of the same robot $\mathbf{R}_{L_1}^{L_0}, \mathbf{t}_{L_1}^{L_0}$ is calculated by (13).

$$\mathbf{J}_k^{\text{reg}} = \sum_{u=1}^U \begin{bmatrix} -[\mathbf{P}_u^{L_0}]_{\times} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \quad (13)$$

The covariance Ω_k of the registration function ϵ_k^{reg} can be calculated as follows.

$$\Omega_k = \mathbf{J}_k^{\text{reg}\top} \mathbf{J}_k^{\text{reg}}. \quad (14)$$

The minimal eigen value of the covariance $\lambda_k^{\min} = \lambda_{\min}(\Omega_k)$ can be used to represent the constraint ability of the point cloud from $\mathbf{E}_k^0$ -th pose and $\mathbf{E}_k^1$ -th pose. When the $\lambda_{\min}(\Omega_k)$ of two node is small enough, the BA constraint between them will be retained and added to the pose graph in Fig. 7. On the contrary, no constraints from BA will be added between $\mathbf{E}_k^0$ -th pose and $\mathbf{E}_k^1$ -th pose with a relatively large $\lambda_{\min}(\Omega_k)$ value.

VII. EXPERIMENTS

A. Experimental Setup

All algorithms are implemented in C++ using the Robot Operating System (ROS) [49]. To evaluate the performance of the proposed multi-session map merging framework, we conduct experiments on several datasets, including the publicly

TABLE I: Overview of Self-Collect Dataset

Sequence	Environment	Trajectory Length (m)	LiDAR
Garage	Indoor	334	Mid360
Library	Outdoor	519	Mid360
Yard	Indoor & Outdoor	232	Mid360
Laboratory	Outdoor	98	Mid360
Flying Arena	Indoor	280	Mid360

TABLE II: Overview of Public Dataset

Dataset	Sequence	Robot Number	Trajectory Length (m)	LiDAR
S3E	Campus_1	3	2989	Velodyne
	Campus_3	3	2938	Velodyne
	Dormitory	3	2168	Velodyne
	Library	3	1524	Velodyne
	Teaching_Building	3	1983	Velodyne
	Tunnel	3	1525	Velodyne
GEODE	Inlandwaterways	3	1209	Avia
	Tunnelingtunnel	3	270	Avia
	Stairs	2	161	Ouster
	Offroad	2	542	Ouster
MARS-LVIG	Airport	2	4135	Avia
	Valley	3	6697	Avia
	Town	4	6117	Avia
	Island	2	2139	Avia
R ³ LIVE	HKU Park	5	356	Avia
	HKU Campus	10	338	Avia
	HKUST Campus	20	945	Avia

available S3E [44], GEODE [50], MARS-LVIG [1], R³LIVE [51] dataset, as well as a self-collected dataset. The GEODE, MARS-LVIG and R³LIVE datasets are segmented into multiple sessions with overlapping areas, but the starting point of each part is different and the relative transformation is unknown. The parameters of our self-collected dataset and the public datasets are shown in Table I and Table II, respectively. For all public datasets, initial odometry is obtained using the LiDAR-Inertial-Odometry method recommended by the respective dataset, while the initial odometry for our own dataset is generated using FAST-LIO2 [14].

We first evaluate our proposed spatial bundle adjustment by comparing it with the conventional LiDAR-based bundle adjustment methods BALM2 [24] and HBA [25] (Section VII-B). Next, we assess the localization performance of our complete multi-session merging framework against DCL-SLAM [27] and LAMM [29] (Section VII-C). Moreover, the mapping quality evaluation is detailed in VII-D. An ablation study is then conducted to analyze the contributions of individual components within our framework (Section VII-E). Finally, we test the scalability of our method in handling a large number of sessions (Section VII-F).

B. Single-Robot Study

To evaluate the effectiveness of our spatial bundle adjustment (BA), we utilize our self-collected single-robot dataset consisting of several indoor and outdoor scenes (Table I). During data acquisition, the z -axis value remains largely stable across each scene, allowing it to serve as a reference to evaluate vertical drift.

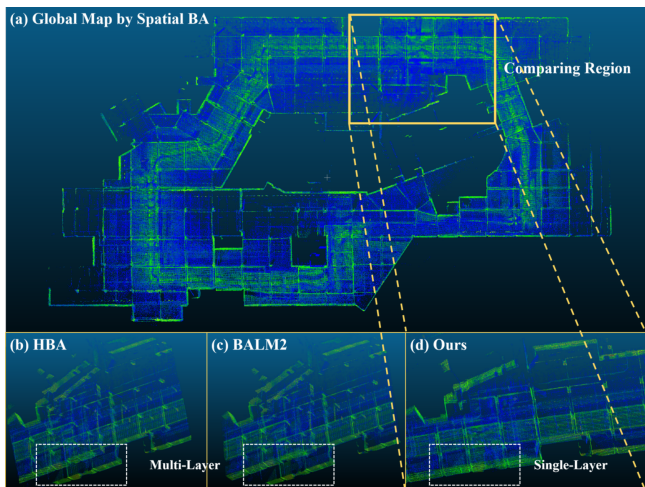


Fig. 8: Result of single-robot study in our Garage dataset. (a) shows the global map generated by our method. (b)-(d) show the details of the local maps near the loop closure for different methods.

TABLE III: MME, z -Drift AND z -RMSE of Single-Robot Study

Sequence	Method	MME	z -Drift	z -RMSE
Garage	HBA	-6.83	5.98	7.14
	BALM2	-6.93	5.95	7.17
	Ours	-6.86	4.87	5.35
Library	HBA	-5.95	6.17	6.78
	BALM2	-6.20	6.51	7.31
	Ours	-6.23	3.68	4.53
Yard	HBA	-6.39	2.33	2.70
	BALM2	-5.93	1.86	2.12
	Ours	-6.40	1.12	1.25
Laboratory	HBA	-6.26	3.24	3.72
	BALM2	-6.04	3.22	3.70
	Ours	-6.20	3.17	3.64

Each sequence in our dataset includes loop closures that enable spatial BA (Section V). We compare our spatial BA with BALM2 [24] and HBA [25], using average z -axis drift (z -DRIFT) and z -axis Root Mean Square Error (z -RMSE) as primary evaluation metrics. Both metrics are computed with respect to the z -value of the first frame as a reference, allowing a consistent evaluation of vertical alignment over time. Due to the absence of ground truth, we also compute the Mean Map Entropy (MME) using MapEval [52], where lower values indicate better map consistency and reduced clutter. To ensure a fair comparison of the performance of the three BA methods, all methods are evaluated using raw odometry trajectories without any prior loop-based refinement. For computational feasibility in large-scale scenes, BALM2 is executed in a sliding-window configuration.

Table III presents the results. Our method achieves the best performance across all metrics in the Library and Yard scenarios. Furthermore, it consistently outperforms BALM2 and HBA in z -DRIFT and z -RMSE, demonstrating its superior capability in mitigating global drift.

Fig. 8 shows the mapping performance of HBA, BALM2 and our proposed method in the Garage sequence, with the framed area indicating the region where loop closures occur.

TABLE IV: RMSE of The ATE(m) of Localization Study

Sequence	DCL-SLAM	LAMM	Ours
Campus_3	17.89	12.51	3.51
Dormitory	3.52	28.67	3.46
Library	6.37	4.58	1.48
Tunnel	3.09	×	0.98
Inlandwaterways	×	×	7.44
Offroad	×	25.45	1.30
Tunnelingtunnel	×	0.62	0.16
Airport	×	8.41	1.30
Town	×	×	1.49
Island	×	×	0.801

It can be seen that both HBA and BALM2 exhibit significant layering in this area, whereas our method achieves superior alignment. This result clearly demonstrates the strong capability of our spatial BA approach in handling loop closure regions.

C. Multi-Robot Localization Study

To further evaluate the proposed framework, we perform comparative multi-robot experiments using the MARS-LVIG [1], GEODE [50], and multi-robot S3E [44] datasets. The parameters of the multi-robot datasets used here are summarized in Table II. Our framework is compared with the multi-robot SLAM system DCL-SLAM [27] and the multi-session map merging method LAMM [29].

The accuracy of map merging is evaluated using the Root Mean Square Error (RMSE) of the Absolute Trajectory Error (ATE). A failure is defined as any sequence with an RMSE greater than 30 meters and is indicated by “×” in the table. As shown in Table IV, the best values are highlighted in bold. Our framework successfully merges all sequences, while LAMM and DCL-SLAM fail in four and six sequences, respectively. In successful cases, our framework achieves significantly lower RMSE values, indicating higher merging accuracy.

We select a scene where all three methods successfully merge the multi-robot point cloud for display. Fig. 9 visualizes the results from the S3E Library sequence. The left image shows our globally consistent and accurate merged map, while the right part shows the local map fusion of the three methods and the comparison of their respective trajectories. Our method yields tightly aligned local maps, whereas LAMM and DCL-SLAM exhibit severe local divergences.

The comparison of trajectory and ground truth further confirms the superior accuracy of our method. Fig. 10 shows optimized trajectories and error plots for different scenes, the multi-robot trajectories generated by our framework are basically consistent with the ground truth, and the error is stable and small, further demonstrating the superior performance and stability of our method. Fig. 11 and Fig. 12 present multi-robot trajectories comparisons between our method, LAMM, and DCL-SLAM. The trajectories estimated by our method closely follow the ground truth for each robot, exhibiting low and stable errors. In contrast, the benchmark methods show significant deviations from the ground truth in some regions and minor local inconsistencies in others. The large deviations are likely caused by incorrect loop closures, while the local

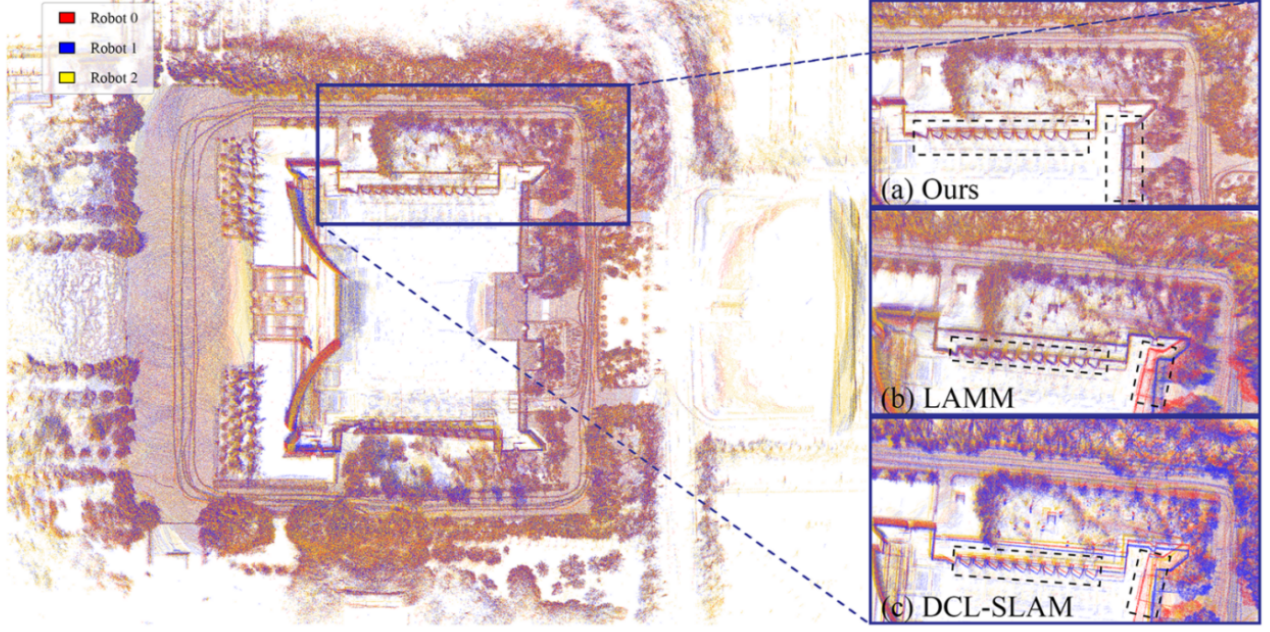


Fig. 9: The map merging result of S3E library. The local maps generated by three methods are selected for comparison. Our method almost eliminates the divergence of local multi-session point cloud maps. While LAMM [29] and DCL-SLAM [27] suffer from serious inconsistencies.

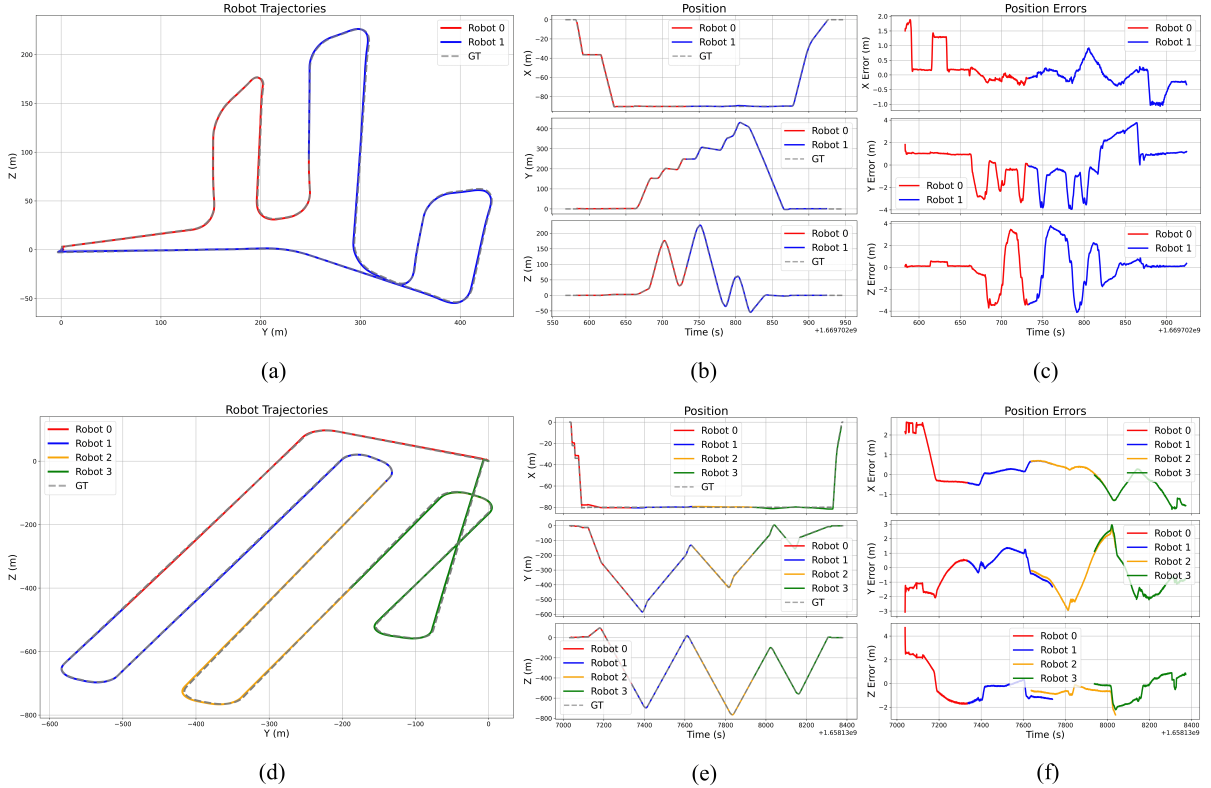


Fig. 10: The optimized trajectory, position, and position error for proposed method in MARS-LVIG Island ((a)-(c)) and Town ((d)-(f)).

inconsistencies may result from the lack of bundle adjustment, which fails to eliminate the divergence of submaps.

The performance disparity can be attributed to LAMM and DCL-SLAM relying solely on PGO with loop closure

constraints. Although this achieves general global consistency, it often neglects the refinement of overlapping local regions. Lack of attention to the geometric structure of point cloud maps leads to serious local multi-robot map divergence.

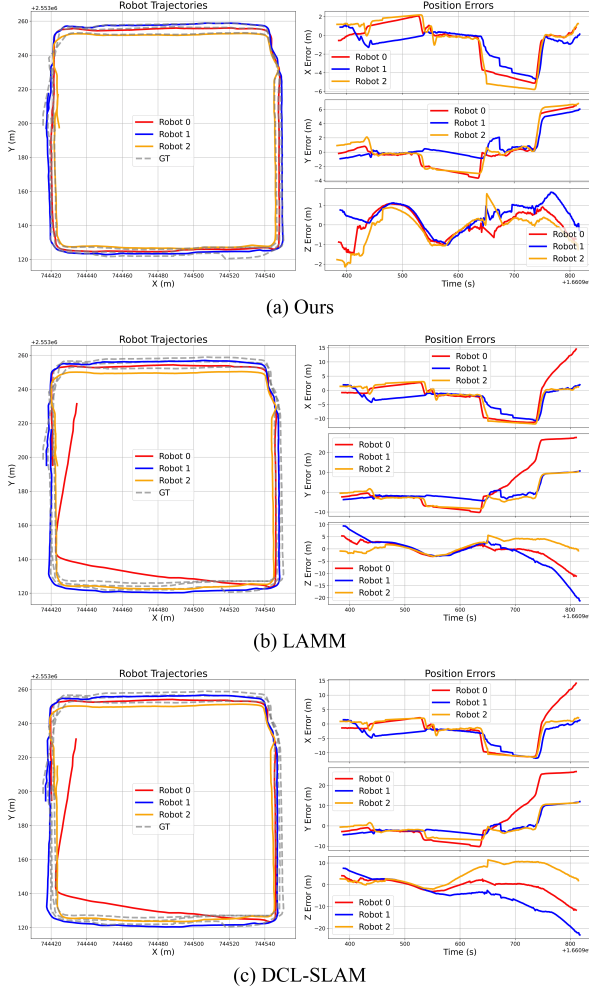


Fig. 11: Comparison of multi-robot trajectories and error between our method, LAMM, and DCL-SLAM on S3E Library. The ground truth is shown as a dashed line in the figure.

Our framework explicitly re-examines the role of loop and enhances its utilization by performing spatial BA in local regions, while propagating the refined results globally via the last pose graph optimization. This process effectively reduces divergence and ensures consistent multi-session map fusion.

D. Multi-Robot Mapping Quality Evaluation

In this section, we evaluate the mapping quality of multi-robot using the MapEval [52]. For the MARS-LVIG dataset, the ground-truth map is a high-precision point cloud generated by the DJI L1 LiDAR sensor and processed with the DJI Terra system. For the S3E dataset, since no ground-truth map is available and the ground-truth poses do not contain rotational information, we evaluate the geometric quality of the reconstructed maps using the average plane thickness and planarity metrics. Detailed definitions of the evaluation metrics are provided in [52].

Table V presents the quantitative comparison of mapping quality between our method and LAMM. Our approach achieves the best performance across all datasets, including lower average Wasserstein distance (AWD), lower Chamfer

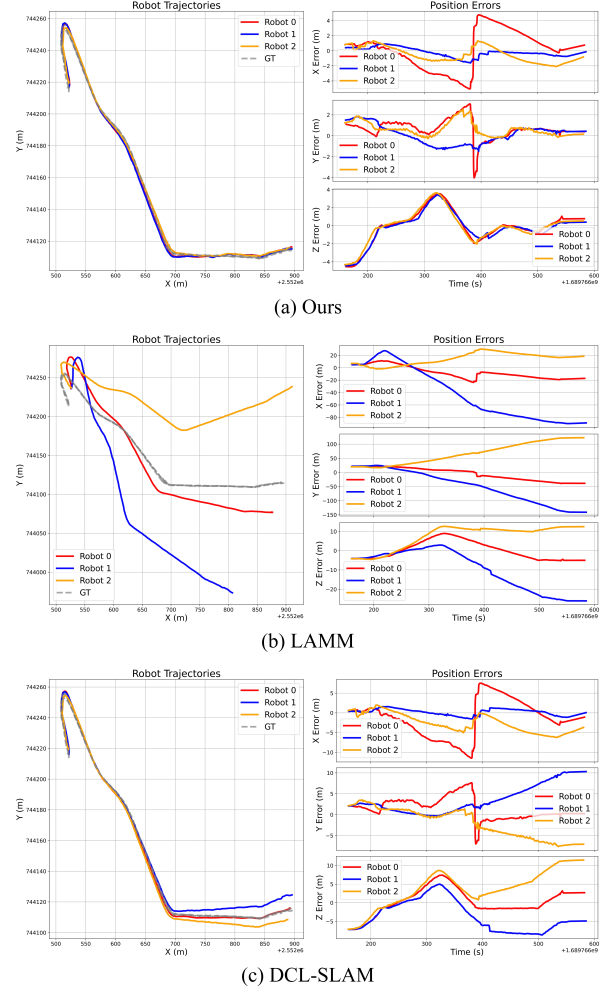


Fig. 12: The same comparison of multi-robot trajectories on S3E Tunnel as Fig. 11.

distance (CD), smaller Spatial Consistency Score (SCS), and lower Mean Map Entropy (MME). These results indicate that our method produces higher-quality maps with better alignment to the ground-truth maps. In these three large-scale environments covering areas of over 100,000 square meters, our AWD ranges from 0.25 m to 0.65 m, and the CD ranges from 0.43 m to 1.11 m, demonstrating strong consistency with the ground-truth maps. Furthermore, the lower SCS and MME values suggest that our method achieves better spatial consistency than LAMM, which can be attributed to the proposed spatial BA and the last PGO for improving local accuracy and global consistency. Furthermore, Fig 13 further visualizes the Wasserstein distance distribution on the Island dataset. As shown in (a), the errors are concentrated within the 3σ bound (0.93 m), with most voxels around 0.25 m. (b) shows that almost all voxels have Wasserstein distance values below 0.5 m and exhibit a spatially consistent distribution, indicating that the reconstructed multi-robot map is well aligned with the ground-truth map and maintains high spatial consistency.

Table VI reports the averaged plane thickness and planarity evaluation results over all sequences in the S3E dataset. Our method achieves the best performance compared to DCL-

TABLE V: Mapping Quality Evaluation

Data	Method	Metrics			
		AWD(m) ↓	CD(m) ↓	SCS ↓	MME ↓
Island	LAMM	1.30	6.44	0.50	-5.51
	Ours	0.30	0.43	0.49	-6.13
Town	LAMM	1.77	10.64	0.57	-6.37
	Ours	0.65	1.11	0.47	-6.52
Airport	LAMM	0.75	36.05	0.62	-6.17
	Ours	0.25	0.47	0.57	-6.28

TABLE VI: Geometry Accuracy Comparison

Metric	DCL-SLAM	LAMM	Ours
Thickness(m) ↓	0.12	0.14	0.08
Planarity ↑	0.61	0.59	0.86

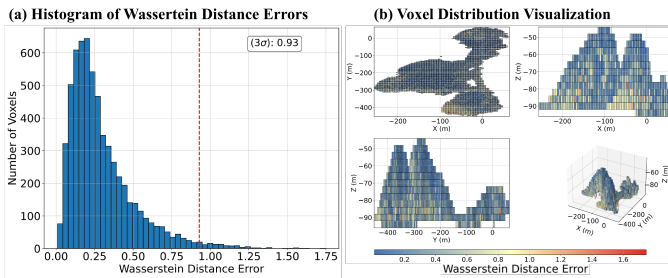


Fig. 13: Visualization of Wasserstein distance errors on the Island dataset. (a) Histogram of errors over all voxels, where the dashed line indicates the 3σ bound (0.93 m). (b) Spatial distribution of voxel-wise errors visualized from multiple coordinates, showing the consistency of reconstruction quality across the environment.

SLAM and LAMM, indicating that the proposed spatial bundle adjustment significantly improves the geometric accuracy of multi-robot mapping and effectively mitigates the multi-robot map divergence commonly observed in PGO-based methods.

E. Ablation Study

To understand the contribution of each component in our framework, we perform ablation experiments on the S3E, GEODE, MARS-LVIG datasets, as well as our self-collected dataset. All sequences of our self-collected dataset are divided into two sessions. Our ablation experiment consists of the following two parts.

First, we conduct an ablation experiment on the loop recall in Section IV. We compare the performance of the first pose graph optimization with and without loop recall (LR) using RMSE of the ATE (m), reported in the table as “RMSE *w/* LR” and “RMSE *w/o* LR,” respectively. Because the S3E dataset contains a large number of complex scenes that combine unstructured and structured data, as well as many scenes with high similarity, it is easy to have a large number of false loops, so we use it as the ablation dataset for loop recall.

Table VII shows the number of loops in different loop processing stages, as well as the results of the first pose graph optimization with and without loop recall. The experimental

TABLE VII: Loop Closure Recall Comparison

Sequence	Initial	After Rejection	After Recalling	RMSE <i>w/o</i> LR	RMSE <i>w/</i> LR
Campus_1	225	185	190	9.98	9.97
Dormitory	106	76	77	4.15	4.12
Library	286	235	238	1.41	1.40
Tunnel	208	182	194	1.40	1.29

TABLE VIII: RMSE of The ATE(m) of Ablation Study

Sequence	FPGO	FPGO + BA	LEMON Full
Campus_1	9.97	10.05	9.94
Campus_3	3.82	4.26	3.51
Dormitory	4.12	4.40	3.46
Tunnel	1.29	1.52	0.98
Stairs	0.194	0.191	0.142
Valley	7.84	7.39	6.76
Island	0.963	1.076	0.801

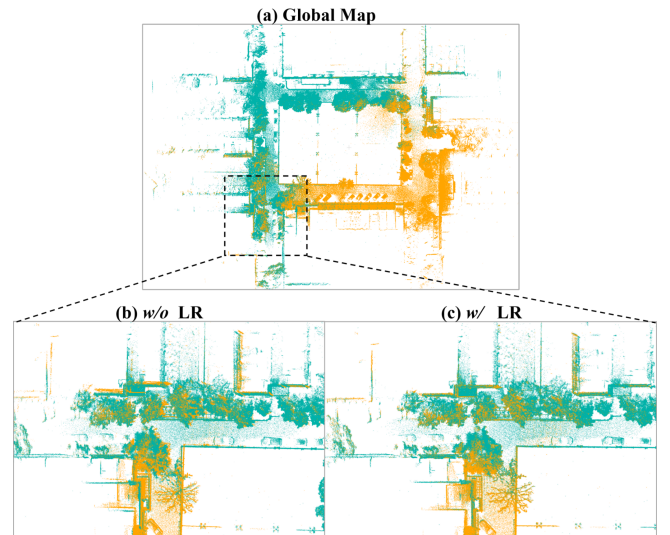


Fig. 14: (a) shows the global map in Laboratory with loop recall. (b) and (c) present enlarged views of the same region without and with loop closure, respectively. In (c), the recalled loop information contributes to improved alignment and consistency across multi-robot maps.

results show that the first PGO achieves smaller errors and better performances when mistakenly rejected loops are recalled. Beyond its impact on PGO, loop recall also introduces additional constraints that benefit subsequent bundle adjustment.

Fig. 14 illustrates the results of the first ablation experiment in our laboratory scene. (a) presents the global map generated with the loop recall step. (b) and (c) compare the local mapping results without and with loop recall, respectively. It is evident that the loop recall introduces additional constraints that significantly improve point cloud alignment within the selected region.

Second, we evaluate the performance of three variants in our map merging module: (1) First Pose Graph Optimization only (FPGO), (2) First Pose Graph Optimization and Spatial Bundle Adjustment (FPGO + BA) and (3) the full LEMON-Mapping system (LEMON Full). RMSE of the ATE (m) is used for quantitative comparison.

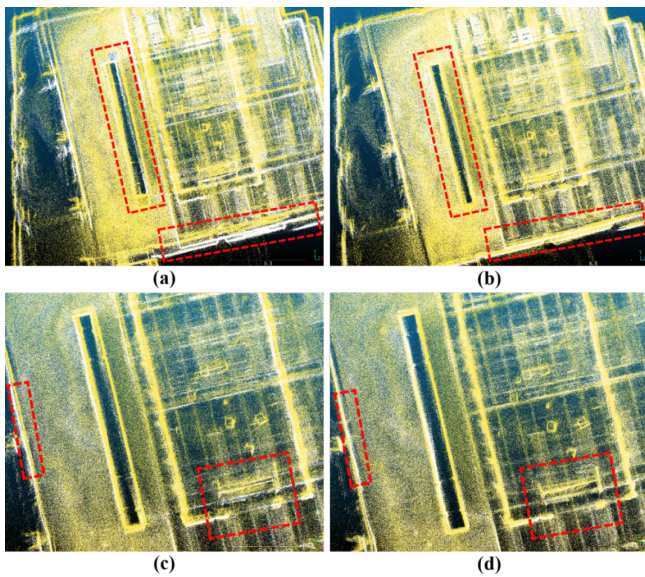


Fig. 15: Result of ablation study in our Flying Arena dataset. (a) and (b) show the merged map of FPGO and FPGO + BA. (c) and (d) show the merged map of FPGO + BA and LEMON-mapping full model. The comparison regions are framed in red.

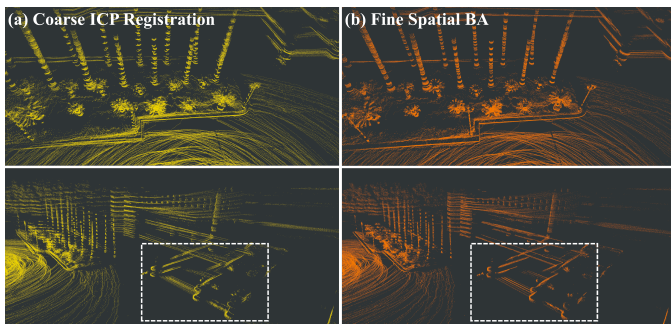


Fig. 16: The map comparison of coarse registration (a) and fine multi-robot spatial BA (b) in S3E Library. It can be clearly seen that GICP only achieves basic alignment of multi-robot submaps. In the upper image, tree trunks and signs show obvious misalignment, and road edges are noticeably blurred. In the lower image, steps exhibit severe diversity. In contrast, after fine spatial BA, the submaps are completely consistent, with excellent geometric quality and clear edge structures.

Table VIII summarizes the results, with the best values highlighted in bold. The complete framework consistently achieves the lowest RMSE, validating the effectiveness of combining spatial BA and two PGO steps. For the variant of FPGO + BA, although local BA improves relative accuracy, it may disrupt the continuity of odometry since it only refines loop regions. Consequently, its standalone use can increase RMSE compared to FPGO alone. However, the last PGO integrates both local BA constraints and odometry continuity, propagating local accuracy to the global map and significantly reducing RMSE.

Fig. 15 shows the map merging results of three variants above in our Flying Arena scene. (a) and (b) demonstrate the function of spatial BA to improve local consistency and accuracy compared to PGO only. (c) and (d) in Fig. 15 further illustrates that maps refined with both BA and last pose graph

TABLE IX: Scalability Study Results

Sequence	Robot Number	Merged Number	Success Rate
HKU Park	5	5	100%
HKU Campus	10	10	100%
HKUST Campus	20	20	100%

optimization (LPGO) have better global consistency than using BA alone, confirming the role of LPGO in maintaining global map structure.

In particular, we visualize the comparison between coarse GICP registration and the proposed fine spatial BA. Fig. 16 (a) shows the result of GICP-based coarse alignment, where only basic submap alignment is achieved, and noticeable misalignments appear in tree trunks, traffic signs, and road boundaries. Fig. 16 (b) shows the result after applying the proposed spatial BA, where submaps become fully consistent with clear edge structures and significantly improved geometric quality. This confirms that spatial BA is a critical component for high-precision multi-robot mapping. While GICP only ensures coarse submap-level alignment, the proposed spatial BA performs fine-grained joint optimization at the pose level, enabling globally consistent and locally accurate map reconstruction. As a result, the originally divergent multi-robot point clouds are refined to achieve highly consistent and precise reconstruction, approaching the geometric fidelity typically observed in short-range single-robot mapping.

F. Scalability Study

Most existing map fusion frameworks are limited to scenarios that involve only a few robots (less than five [27]–[29], [41]), and their scalability to large-scale deployments is not yet proven. To assess the scalability of our proposed framework, we perform experiments on the R³LIVE dataset, which we subdivide into groups of 5, 10, and 20 sessions. A successful fusion is defined as the correct alignment of each session with all its adjacent sessions that shares sufficient map overlap.

Table IX summarizes the corresponding results of three cases. Our framework achieves a success rate 100% in all experiments, including the five-session, ten-session, and twenty-session scenarios. These results demonstrate the scalability and robustness of the LEMON-Mapping system when dealing with large numbers of multi-robot map merging.

Fig. 17 visualizes the fused map for the five-session case of HKU Park in R³LIVE dataset, including a global point cloud map from a bird’s eye view (BEV) and 6 enlarged map details. The five-session map shows good fusion effect and local accuracy in both structured and unstructured environments.

VIII. CONCLUSION

This paper presents **LEMON-Mapping**, a Loop-Enhanced Large-Scale Multi-Session Point Cloud Map Merging and Optimization framework for Globally Consistent Mapping. LEMON-Mapping is a scalable framework for a large number of robots. Unlike existing methods that rely solely on pose sequences, it enhances the role of loop closures by directly utilizing geometric constraints from overlapping point clouds

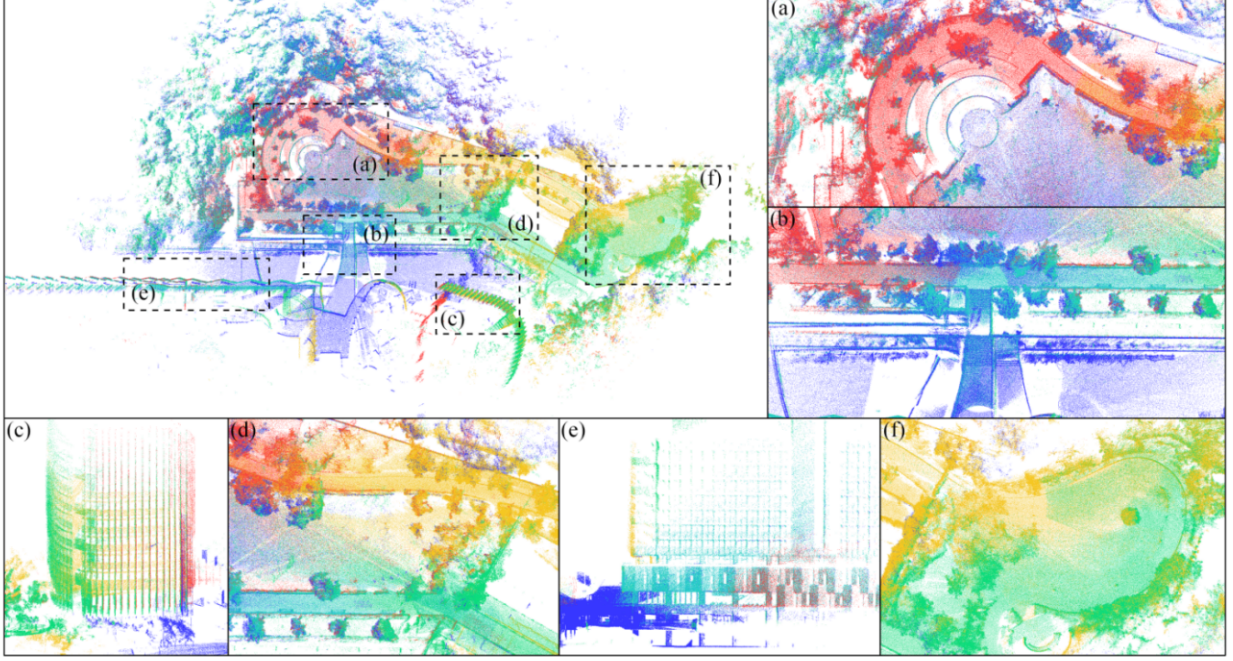


Fig. 17: The map merging result of the five-session dataset in R³LIVE HKU Park. The local maps of (a)-(f) are enlarged to show the details.

and reasonably transferring the local accuracy to the global map. The framework includes a loop processing module for reliable loop selection and innovate loop recall, and a map merging module that combines two-step pose graph optimization with a spatial bundle adjustment tailored for multi-robot systems. The spatial bundle adjustment operates on loop-based spatial windows, allowing geometry-aware pose optimization across unordered and cross-robot trajectories. To ensure global consistency, we sparsify the BA constraints and propagate local accuracy across the entire map using a refined pose graph optimization strategy. Experimental results in S3E, MARS-LVIG, GEODE, R³LIVE and our self-collected dataset demonstrate that our approach maintains high map merging success rates and robustness as the number of sessions increases, while outperforming traditional methods in geometric accuracy and scalability.

APPENDIX: PROOF OF LEMMA

A. Proof of Lemma 1

Proof. Since each $\mathbf{A}(x)$ is real symmetric, the spectral theorem guarantees an orthogonal matrix $\mathbf{Q}(x)$ and a real diagonal matrix

$$\mathbf{\Lambda}(x) = \text{diag}(\lambda_1(x), \dots, \lambda_n(x))$$

such that

$$\mathbf{A}(x) = \mathbf{Q}(x) \mathbf{\Lambda}(x) \mathbf{Q}(x)^\top.$$

In particular, the eigenvalues $\lambda_i(x)$ are all real.

Consider the characteristic polynomial

$$p_x(t) = \det(\mathbf{A}(x) - t\mathbf{I}).$$

On one hand, by orthogonal diagonalization,

$$p_x(t) = \det(\mathbf{A}(x) - t\mathbf{I}) = \prod_{i=1}^n (\lambda_i(x) - t).$$

On the other hand, expanding the determinant in terms of the entries of $\mathbf{A}(x)$ shows

$$p_x(t) = t^n + c_{n-1}(x)t^{n-1} + \dots + c_1(x)t + c_0(x),$$

where each coefficient $c_k(x)$ is a finite sum of products of the entries of $\mathbf{A}(x)$. Since $\mathbf{A}(x)$ is continuous in x , each $c_k(x)$ is a continuous function of x .

By the fundamental theorem of algebra and the fact that all roots of p_x are real, we may label them in nondecreasing order

$$\lambda_1(x) \leq \lambda_2(x) \leq \dots \leq \lambda_n(x).$$

A standard result on the continuous dependence of polynomial roots on coefficients now implies that each ordered root $\lambda_i(x)$ varies continuously with the coefficients $\{c_k(x)\}$, and hence with x itself. Therefore, every eigenvalue function $x \mapsto \lambda_i(x)$ is continuous. $\square$

B. Proof of Lemma 2

Proof. The complexity of joint BA over all M poses is

$$C_{\text{joint}} = O(M_f M + M_f M^2 + M^3),$$

where the three terms correspond to evaluating feature residuals, forming the Hessian, and solving the LM linear system.

Expanding the squared and cubic terms via group sizes m_i yields

$$M^2 = \left(\sum_i m_i \right)^2 = \sum_i m_i^2 + 2 \sum_{i < j} m_i m_j,$$

$$M^3 = \left(\sum_i m_i \right)^3 = \sum_i m_i^3 + (\text{mixed cross-group terms}).$$

Thus joint BA includes not only the per-group contributions $\sum_i m_i^2$ and $\sum_i m_i^3$, but also all cross-group interaction terms such as $m_i m_j$, $m_i^2 m_j$, and $m_i m_j m_k$, which dominate when multiple groups have comparable size.

In DBA, only the poses in the current diffusion group i remain active while all inner groups are frozen. The BA performed in diffusion step i therefore has complexity

$$O(M_f m_i + M_f m_i^2 + m_i^3).$$

Summing across all diffusion steps gives the total DBA cost

$$\mathcal{C}_{\text{DBA}} = O\left(M_f \sum_i m_i + M_f \sum_i m_i^2 + \sum_i m_i^3\right),$$

which eliminates all cross-group terms present in $\mathcal{C}_{\text{joint}}$. Since

$$\sum_i m_i^2 \leq M^2, \quad \sum_i m_i^3 \leq M^3,$$

DBA is never more expensive than joint BA.

Moreover, when groups are approximately balanced with $m_i \approx M/D$, we obtain

$$\sum_i m_i^2 \approx \frac{M^2}{D}, \quad \sum_i m_i^3 \approx \frac{M^3}{D^2},$$

showing that the dominating quadratic and cubic terms are reduced by factors of order D and D^2 , respectively. This proves that DBA yields a strictly lower computational cost and achieves substantial practical speedups. $\square$

C. Proof of Lemma 3

Proof. We proceed in two steps.

1) *Block-extraction lower bound for the measurement noise term.*: For each measurement cluster the per-measurement contribution is

$$\mathbf{L}_{ij} \Sigma_{c_{fij}} \mathbf{L}_{ij}^\top = \begin{bmatrix} {}^0\mathbf{L}_{ij} \Sigma^0 \mathbf{L}_{ij}^\top & {}^0\mathbf{L}_{ij} \Sigma^1 \mathbf{L}_{ij}^\top \\ {}^1\mathbf{L}_{ij} \Sigma^0 \mathbf{L}_{ij}^\top & {}^1\mathbf{L}_{ij} \Sigma^1 \mathbf{L}_{ij}^\top \end{bmatrix},$$

which is positive semidefinite because it is of the form $A \Sigma A^\top$. Subtracting the block-diagonal matrix that keeps only the bottom-right block yields

$$\mathbf{L}_{ij} \Sigma \mathbf{L}_{ij}^\top - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^1\mathbf{L}_{ij} \Sigma^1 \mathbf{L}_{ij}^\top \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{L}_{ij} \Sigma^0 \mathbf{L}_{ij}^\top & {}^0\mathbf{L}_{ij} \Sigma^1 \mathbf{L}_{ij}^\top \\ {}^1\mathbf{L}_{ij} \Sigma^0 \mathbf{L}_{ij}^\top & \mathbf{0} \end{bmatrix}.$$

The right-hand side can be written as

$$\begin{bmatrix} {}^0\mathbf{L}_{ij} \\ \mathbf{0} \end{bmatrix} \Sigma \begin{bmatrix} {}^0\mathbf{L}_{ij} \\ \mathbf{0} \end{bmatrix}^\top \succeq 0,$$

hence

$$\mathbf{L}_{ij} \Sigma \mathbf{L}_{ij}^\top \succeq \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^1\mathbf{L}_{ij} \Sigma^1 \mathbf{L}_{ij}^\top \end{bmatrix}.$$

Summing over all measurements preserves the PSD ordering, and extracting the bottom-right block gives the key inequality

$$\left[\sum_{i,j} \mathbf{L}_{ij} \Sigma_{c_{fij}} \mathbf{L}_{ij}^\top \right]_{11} \succeq \sum_{i,j \in \text{group1}} {}^1\mathbf{L}_{ij} \Sigma_{c_{fij}} {}^1\mathbf{L}_{ij}^\top. \quad (15)$$

2) *Schur-complement ordering for Hessian inverses.*: Write the Schur complement of $\mathbf{H}_{00}$ in $\mathbf{H}$:

$$\mathbf{S} \triangleq \mathbf{H}_{11} - \mathbf{H}_{10} \mathbf{H}_{00}^{-1} \mathbf{H}_{01}.$$

Because $\mathbf{H}_{10} \mathbf{H}_{00}^{-1} \mathbf{H}_{01} \succeq 0$, we have

$$\mathbf{S} \preceq \mathbf{H}_{11}.$$

For two strict positive definite matrices $\mathbf{A} \preceq \mathbf{B}$ it follows that $\mathbf{B}^{-1} \preceq \mathbf{A}^{-1}$. Applying this to $\mathbf{S} \preceq \mathbf{H}_{11}$ yields

$$\mathbf{H}_{11}^{-1} \preceq \mathbf{S}^{-1}.$$

3) *Combine VIII-C1 and VIII-C2.*: The bottom-right block of the joint covariance can be written using the Schur complement inverse $\mathbf{S}^{-1}$:

$$\Sigma_{11}^{\text{joint}} = \mathbf{S}^{-1} \left[\sum_{i,j} \mathbf{L}_{ij} \Sigma_{c_{fij}} \mathbf{L}_{ij}^\top \right]_{11} \mathbf{S}^{-T}.$$

Using (15) and the inverse-ordering $\mathbf{H}_{11}^{-1} \preceq \mathbf{S}^{-1}$ we obtain

$$\begin{aligned} \Sigma_{11}^{\text{joint}} &= \mathbf{S}^{-1} \left[\sum_{i,j} \mathbf{L}_{ij} \Sigma \mathbf{L}_{ij}^\top \right]_{11} \mathbf{S}^{-T} \\ &\succeq \mathbf{S}^{-1} \left(\sum_{i,j \in \text{group1}} {}^1\mathbf{L}_{ij} \Sigma^1 \mathbf{L}_{ij}^\top \right) \mathbf{S}^{-T} \\ &\succeq \mathbf{H}_{11}^{-1} \left(\sum_{i,j \in \text{group1}} {}^1\mathbf{L}_{ij} \Sigma^1 \mathbf{L}_{ij}^\top \right) \mathbf{H}_{11}^{-T} \\ &= \Sigma_1^{\text{DBA}}, \end{aligned}$$

where the second PSD inequality follows from left- and right-multiplying the PSD matrix $\sum_{i,j \in \text{group1}} {}^1\mathbf{L}_{ij} \Sigma^1 \mathbf{L}_{ij}^\top$ by $\mathbf{S}^{-1}$ and noting $\mathbf{H}_{11}^{-1} \preceq \mathbf{S}^{-1}$.

Thus $\Sigma_1^{\text{DBA}} \preceq \Sigma_{11}^{\text{joint}}$, which proves the stated ordering. $\square$

REFERENCES

- [1] H. Li, Y. Zou, N. Chen, J. Lin, X. Liu, W. Xu, C. Zheng, R. Li, D. He, F. Kong *et al.*, “Mars-lvig dataset: A multi-sensor aerial robots slam dataset for lidar-visual-inertial-gnss fusion,” *The International Journal of Robotics Research*, vol. 43, no. 8, pp. 1114–1127, 2024.
- [2] Y. Ren, F. Zhu, G. Lu, Y. Cai, L. Yin, F. Kong, J. Lin, N. Chen, and F. Zhang, “Safety-assured high-speed navigation for mavs,” *Science Robotics*, vol. 10, no. 98, p. eado6187, 2025.
- [3] F. Zhu, Y. Ren, F. Kong, H. Wu, S. Liang, N. Chen, W. Xu, and F. Zhang, “Swarm-liv: Decentralized swarm lidar-inertial odometry,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [4] F. Zhu, Y. Ren, L. Yin, F. Kong, Q. Liu, R. Xue, W. Liu, Y. Cai, G. Lu, H. Li, and F. Zhang, “Swarm-liv2: Decentralized efficient lidar-inertial odometry for aerial swarm systems,” *IEEE Transactions on Robotics*, 2025.
- [5] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li, and D. Cao, “Deep learning for image and point cloud fusion in autonomous driving: A review,” *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [6] Y. Ren, Y. Cai, F. Zhu, S. Liang, and F. Zhang, “Rog-map: An efficient robocentric occupancy grid map for large-scene and high-resolution lidar-based motion planning,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.
- [7] F. Yang, C. Wang, C. Cadena, and M. Hutter, “iplanner: Imperative path planning,” *Proceedings of Robotics: Science and System XIX*, p. 064, 2023.
- [8] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter, “Viplanner: Visual semantic imperative learning for local navigation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5243–5249.
- [9] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, “An autonomous multi-uav system for search and rescue,” in *Proceedings of the first workshop on micro aerial vehicle networks, systems, and applications for civilian use*, 2015, pp. 33–38.
- [10] A. G. Araújo, C. A. Pizzino, M. S. Couceiro, and R. P. Rocha, “A multi-drone system proof of concept for forestry applications,” *Drones*, vol. 9, no. 2, p. 80, 2025.
- [11] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, D. Heřt, M. Petrlik, T. Bába, V. Spurný *et al.*, “Darpa subterranean challenge: Multi-robotic exploration of underground environments,” in *Modelling and Simulation for Autonomous Systems: 6th International Conference, MESAS 2019, Palermo, Italy, October 29–31, 2019, Revised Selected Papers 6*. Springer, 2020, pp. 274–290.
- [12] J. Zhang, S. Singh *et al.*, “Loam: Lidar odometry and mapping in real-time,” in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [13] T.-M. Nguyen, D. Duberg, P. Jensfelt, S. Yuan, and L. Xie, “Slic: Multi-input multi-scale surfel-based lidar-inertial continuous-time odometry and mapping,” *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2102–2109, 2023.
- [14] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “Fast-liv2: Fast direct lidar-inertial odometry,” *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [15] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, “Faster-liv: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.
- [16] K. Chen, R. Nemirow, and B. T. Lopez, “Direct lidar-inertial odometry: Lightweight liv with continuous-time motion correction,” in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 3983–3989.
- [17] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, “Point-liv: Robust high-bandwidth light detection and ranging inertial odometry,” *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200459, 2023.
- [18] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.
- [19] C. Yuan, J. Lin, Z. Zou, X. Hong, and F. Zhang, “Std: Stable triangle descriptor for 3d place recognition,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1897–1903.
- [20] C. Yuan, J. Lin, Z. Liu, H. Wei, X. Hong, and F. Zhang, “Btc: A binary and triangle combined descriptor for 3-d place recognition,” *IEEE Transactions on Robotics*, vol. 40, pp. 1580–1599, 2024.
- [21] X. Xu, S. Lu, J. Wu, H. Lu, Q. Zhu, Y. Liao, R. Xiong, and Y. Wang, “Ring++: Roto-translation invariant gram for global localization on a sparse scan map,” *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4616–4635, 2023.
- [22] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss, “Overlapnet: Loop closing for lidar-based slam,” *arXiv preprint arXiv:2105.11344*, 2021.
- [23] Z. Liu and F. Zhang, “Balm: Bundle adjustment for lidar mapping,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.
- [24] Z. Liu, X. Liu, and F. Zhang, “Efficient and consistent bundle adjustment on lidar point clouds,” *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4366–4386, 2023.
- [25] X. Liu, Z. Liu, F. Kong, and F. Zhang, “Large-scale lidar consistent mapping using hierarchical lidar bundle adjustment,” *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1523–1530, 2023.
- [26] J. Li, T.-M. Nguyen, S. Yuan, and L. Xie, “Pss-ba: Lidar bundle adjustment with progressive spatial smoothing,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 1124–1129.
- [27] S. Zhong, Y. Qi, Z. Chen, J. Wu, H. Chen, and M. Liu, “Dcl-slam: A distributed collaborative lidar slam framework for a robotic swarm,” *IEEE Sensors Journal*, vol. 24, no. 4, pp. 4786–4797, 2024.
- [28] Y. Huang, T. Shan, F. Chen, and B. Englot, “Disco-slam: Distributed scan context-enabled multi-robot lidar slam with two-stage global-local graph optimization,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1150–1157, 2021.
- [29] H. Wei, R. Li, Y. Cai, C. Yuan, Y. Ren, Z. Zou, H. Wu, C. Zheng, S. Zhou, K. Xue, and F. Zhang, “Large-scale multi-session point-cloud map merging,” *IEEE Robotics and Automation Letters*, 2025.
- [30] P. Yin, S. Zhao, H. Lai, R. Ge, J. Zhang, H. Choset, and S. Scherer, “Automerger: A framework for map assembling and smoothing in city-scale environments,” *IEEE Transactions on Robotics*, 2023.
- [31] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, “Pairwise consistent measurement set maximization for robust multi-robot map merging,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [32] B. Forsgren, R. Vasudevan, M. Kaess, T. W. McLain, and J. G. Mangelson, “Group-k consistent measurement set maximization for robust outlier detection,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [33] B. Forsgren, M. Kaess, R. Vasudevan, T. W. McLain, and J. G. Mangelson, “Group-k consistent measurement set maximization via maximum clique over k-uniform hypergraphs for robust multi-robot map merging,” *The International Journal of Robotics Research*, vol. 43, no. 14, pp. 2245–2273, 2024.
- [34] L. Wang, L. Guo, Z. Xu, Q. Wang, F. Gao, and X. Chen, “Lidar-vgg: Cross-modal coarse-to-fine fusion for globally consistent and metric-scale dense mapping,” *arXiv preprint arXiv:2511.01186*, 2025.
- [35] N. Sünderhauf and P. Protzel, “Towards a robust back-end for pose graph slam,” in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 1254–1261.
- [36] E. Mendes, P. Koch, and S. Lacroix, “Icp-based pose-graph slam,” in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 195–200.
- [37] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “Liv-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.
- [38] T. Ma, B. Xia, Y. Ou, J. Wang, and S. Xu, “Robust second-order lidar bundle adjustment algorithm using mean squared group metric,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–1, 2025.
- [39] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, “Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8779–8785.
- [40] Y. Wu, Q. Gu, J. Yu, G. Ge, J. Wang, Q. Liao, C. Zhang, and Y. Wang, “Mr-gmmexplore: Multi-robot exploration system in unknown environments based on gaussian mixture model,” in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2022, pp. 1198–1203.
- [41] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, “Segmap: 3d segment mapping using data-driven descriptors,” in *Robotics: Science and Systems XIV*, ser. RSS2018. Robotics: Science and Systems Foundation, Jun. 2018. [Online]. Available: <http://dx.doi.org/10.15607/RSS.2018.XIV.003>

- [42] H. Wu, Y. Li, W. Xu, F. Kong, and F. Zhang, "Moving event detection from lidar point streams," *nature communications*, vol. 15, no. 1, p. 345, 2024.
- [43] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, 1981.
- [44] D. Feng, Y. Qi, S. Zhong, Z. Chen, Q. Chen, H. Chen, J. Wu, and J. Ma, "S3e: A multi-robot multimodal dataset for collaborative slam," *IEEE Robotics and Automation Letters*, 2024.
- [45] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.
- [46] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.
- [47] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.
- [48] J. Li, T.-M. Nguyen, M. Cao, S. Yuan, T.-Y. Hung, and L. Xie, "Graph optimality-aware stochastic lidar bundle adjustment with progressive spatial smoothing," *arXiv preprint arXiv:2410.14565*, 2024.
- [49] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.
- [50] Z. Chen, Y. Qi, D. Feng, X. Zhuang, H. Chen, X. Hu, J. Wu, K. Peng, and P. Lu, "Heterogeneous lidar dataset for benchmarking robust localization in diverse degenerate scenarios," *arXiv preprint arXiv:2409.04961*, 2024.
- [51] J. Lin and F. Zhang, "R3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 672–10 678.
- [52] X. Hu, J. Wu, M. Jia, H. Yan, Y. Jiang, B. Jiang, W. Zhang, W. He, and P. Tan, "Mapeval: Towards unified, robust and efficient slam map evaluation framework," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4228–4235, 2025.