Distributed 3-D Multi-Robot Cooperative Localization: An Efficient and Consistent Approach

Yizhi Zhou, Yufan Liu, and Xuan Wang, *Member, IEEE*

***Abstract*—This letter studies the problem of Cooperative Localization (CL) for multi-robot systems in 3-D environments, where a group of mobile robots jointly localize themselves by using measurements from onboard sensors and shared information from other robots. To ensure the efficiency of information fusion and observability consistency in a distributed CL system, we propose a distributed multi-robot CL method based on Lie groups, well-suited for 3-D scenarios with full 3-D rotational dynamics and generic nonlinear inter-robot measurement models. Unlike most existing distributed CL algorithms that operate in vector space and are only applicable to simple 2-D environments, the proposed algorithm performs distributed information fusion directly on the manifold that inherently accounts for the non-Euclidean nature of 3-D rotations and translations. By leveraging the nice property of invariant errors, we analytically prove that the proposed algorithm naturally preserves the observability consistency of the CL system. This ensures that the system maintains the correct structure of unobservable directions throughout the estimation process. The effectiveness of the proposed algorithm is validated by several numerical experiments conducted to rigorously investigate the effects of relative information fusion in the distributed CL system.**

***Index Terms*—Cooperative localization, distributed estimation, consistency.**

I. Introduction

COOPERATIVE localization (CL) is widely employed in networked multi-robot systems. Unlike single-robot localization, which relies solely on a robot's own measurements, CL for multi-robot systems leverages shared or relative information from neighboring robots through inter-robot communication, significantly enhancing the localization accuracy of the ego robot. CL methods are generally categorized into two types: centralized and distributed. Centralized methods fuse all robots' states into one vector to achieve optimal accuracy [1], but they are vulnerable to single-point failures. In contrast, distributed methods use only each robot's local information [2], making them more robust to node failures and dynamic network topologies.

Developing a fully distributed CL algorithm for multi-robot systems in complex *3-D* environments is challenging, particularly in fusing inter-robot measurements without explicit knowledge of cross-correlations between robots [3]. As robots exchange information to refine their localization, intricate correlations emerge between their state estimates, which are typically unknown and intractable in a fully distributed framework [2]. Improperly handling these correlations during fusion can lead to inaccurate results, a phenomenon known as inconsistency. Most existing studies have addressed this problem in 2-D scenarios [2], [3] or simplified 3-D settings that neglect complex rotations to achieve consistent fusion [4]. However, these approaches operate in vector spaces, making them ill-suited for capturing the non-Euclidean nature of robot states in general 3-D environments with complex rotations. Unlike the 2-D scenario, which operates in a vector space, addressing the distributed CL problem in complex 3-D environments requires handling non-Euclidean state representations and incorporating 3-D rotational dynamics. Some studies addressed the 3-D fusion problem by representing the robot's state using a Lie group $SE(3)$ [5] or an augmented quaternion, and fuse relative measurements indirectly by converting them into state estimates of the observed robot to ensure a consistent fusion. However, these methods are restricted to relative pose measurements and cannot handle general models.

Besides the improper handling of correlations between robots, another common cause of inconsistency is the failure to preserve system observability. Observability inconsistency occurs when the estimated states fail to correctly capture the system's inherent unobservable directions, potentially leading to erroneous estimates and degraded localization performance [6]. One approach to solving this problem is to modify the system's observable directions to enforce proper observability properties. This is a technique widely adopted in 3-D single robot localization [7] and some 2-D CL scenarios [1], [6]. Another approach is the invariant Kalman filter (InEKF), which represents the robot state as an $SE(3)$ matrix Lie group [8], [9] and naturally preserves system observability due to the properties of the invariant error. Nevertheless, existing InEKF-based methods are primarily designed for single-robot localization or multi-robot systems that do not incorporate relative information between robots [10].

Motivated by these challenges, this letter proposes a novel distributed invariant Kalman filter (InEKF) for 3-D multi-robot CL on Lie groups, which ensures estimation consistency. Compared to most existing CL methods that operate solely in vector spaces, our approach leverages the Lie group structure to better handle the non-Euclidean nature of 3-D robot states while preserving system consistency. In addition, our method directly fuses relative information, which we refer to as *tightly-coupled fusion*, without requiring the transformation of relative measurements before fusion. Thus, the proposed method does not require relative measurements to be pose-only observations [5],

Received 2 July 2025; accepted 8 December 2025. Date of publication 12 January 2026; date of current version 19 January 2026. This article was recommended for publication by Associate Editor Subhrajit Bhattacharya and Editor M. Ani Hsieh upon evaluation of the reviewers' comments. *(Corresponding author: Xuan Wang.)*

Yizhi Zhou and Xuan Wang are with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA (e-mail: yzhou@gmu.edu; xwang64@gmu.edu).

Yufan Liu is with Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: yufanliu@andrew.cmu.edu).

Digital Object Identifier 10.1109/LRA.2026.3653287

making it suitable for general measurement models. The major contributions of this work are:

- We propose a distributed InEKF on Lie groups for 3-D multi-robot CL that tightly fuses relative information in a fully distributed manner, while naturally handling the non-Euclidean geometry of 3-D poses and supporting general nonlinear measurements.
- We analytically derived the observability properties of the multi-robot CL system on Lie groups and proved the observability consistency of the proposed method. Please refer to the **supplementary material** [11].
- We conducted extensive simulations as well as large-scale real-world experiments to investigate the impact of relative information fusion on the observability properties of the CL system. The results demonstrate that the proposed method rigorously preserves observability consistency while achieving higher state estimation accuracy, thereby validating its effectiveness and scalability in practical multi-robot settings.

II. Related Works

A. Cooperative Localization

Effectively fusing relative information or measurements between robots is essential for consistent distributed multi-robot CL [12]. A Covariance Intersection (CI)-based method is presented in [3] to fuse relative poses between robots. The method in [13] incorporates a generalized maximum likelihood estimation to further handle the biased sensor measurements. Zhu et al. [2] introduces a tightly-coupled fusion method for fusing the relative measurements suitable for any types of relative measurements. However, all those methods only consider 2-D environments where the robot states reside in the vector space. Fang et al. [14] introduces an adaptive cubature split CI filter to fuse the relative information in 3-D environments. In [5], a distributed fusion method for 3-D CL is proposed to fuse the relative poses between robots. Nevertheless, those methods are specifically designed for relative pose measurements and may not generalize to other types of relative measurements such as distance-based measurements [15]. Meanwhile, cooperative visual-inertial odometry approaches are proposed in [16], [17] for distributed CL in 3-D environments, leveraging common observations to enhance ego-robot localization. However, these methods [16], [17] do not ensure observability consistency and do not account for relative measurements between robots.

B. Observability Consistency of Localization System

The problem of observability inconsistency of an estimator was first examined in [7]. To address this problem, methods such as the first-estimates Jacobian (FEJ) [7] and the observability-constrained (OC) approach [18], [19] have been proposed. These methods ensure consistency by preserving the correct observable directions of the estimator. Both approaches have been successfully applied in single-robot [7], [20] and multi-robot localization scenarios [18]. A Kalman decomposition-based approach is proposed in [1] to ensure the observability consistency of the CL system by eliminating the error discrepancy items. Instead of manually modifying the estimators' observable space for consistency, studies such as [21], [22] employ Lie group representations for robot poses to achieve consistent localization in single robot scenarios by leveraging the properties of invariant observers [8]. Building on this, works such as [13], [23] extend these approaches to multi-agent estimation, enabling consistent localization across collaborative systems. However, these methods [13], [23] do not consider relative measurements between robots and therefore cannot be applied to multi-robot CL scenarios.

III. Problem Statement and Analysis

A. Problem Statement and Model

Consider a networked multi-robot system consisting of m robots in a 3-D environment, where the poses of each robot are initially unknown and need to be estimated. Each robot is equipped with proprioceptive (e.g., IMU or odometries) to measure its ego-motion, and exteroceptive sensors (e.g., camera or Ultra-Wideband sensor) to observe static environmental features and obtain relative measurements to other robots. In a distributed CL setting, each robot may use its ego-motion measurements and relative information received from other robots to estimate its own state.

System State: We define the state of each robot i to be estimated at timestep k as $\mathbf{X}_{i,k} \in SE_{2+L}(3) \times \mathbb{R}^6$, where L denotes the number of environmental feature points, and the subscript represents augmented $SE(3)$ group [9].

$$\mathbf{X}_{i,k} = (\mathbf{T}_{i,k}, \mathbf{B}_{i,k}), \quad \mathbf{B}_{i,k} = \begin{bmatrix} \mathbf{b}_{a_{i,k}}^\top & \mathbf{b}_{\omega_{i,k}}^\top \end{bmatrix}^\top$$
$$\mathbf{T}_{i,k} = \left[\begin{array}{c|ccc} {}^G_{I_{i,k}}\mathbf{R} & {}^G\mathbf{v}_{I_{i,k}} & {}^G\mathbf{p}_{I_{i,k}} & {}^G\mathbf{p}_f \\ \hline \mathbf{0}_3 & & \mathbf{I}_3 & \end{array}\right], \tag{1}$$

where $\mathbf{T}_{i,k} \in SE_{2+L}(3)$. The terms $\mathbf{b}_{\omega_i,k}, \mathbf{b}_{a_i,k} \in \mathbb{R}^3$ denote the gyroscope and accelerometer biases, respectively. ${}^G_{I_{i,k}}\mathbf{R} \in SO(3)$ represents the rotation from robot i's inertial frame $I_{i,k}$ to global frame G at timestep k. ${}^G\mathbf{p}_{I_{i,k}}$ and ${}^G\mathbf{v}_{I_{i,k}}$ denote IMU's 3-D position and velocity in global frame G, respectively. ${}^G\mathbf{p}_f$ represents the coordinate of an environmental feature point. For simplicity, we assume the system $\mathbf{X}_{i,k}$ only contains a single feature, i.e. $L = 1$, which is easily generalizable to the multiple features case by augmenting matrix $\mathbf{T}_{i,k}$. We define $\hat{\mathbf{X}}_{i,k} = (\hat{\mathbf{T}}_{i,k}, \hat{\mathbf{B}}_{i,k})$ as the state estimate of $\mathbf{X}_{i,k}$, then formulate the estimation error as

$$\widetilde{\mathbf{X}}_{i,k} = \left(\hat{\mathbf{T}}_{i,k}\mathbf{T}_{i,k}^{-1}, \hat{\mathbf{B}}_{i,k} - \mathbf{B}_{i,k}\right) \triangleq \left(\boldsymbol{\eta}_{i,k}, \widetilde{\mathbf{B}}_{i,k}\right), \tag{2}$$

where $\boldsymbol{\eta}_{i,k} \in SE_3(3)$ denotes the right invariant error:

$$\boldsymbol{\eta}_{i,k} = \left[\begin{array}{c|ccc} {}^G_{I_{i,k}}\widetilde{\mathbf{R}} & \boldsymbol{\Gamma}_1 & \boldsymbol{\Gamma}_2 & \boldsymbol{\Gamma}_3 \\ \hline \mathbf{0}_3 & & \mathbf{I}_3 & \end{array}\right]$$
$${}^G_{I_{i,k}}\widetilde{\mathbf{R}} = {}^G_{I_{i,k}}\hat{\mathbf{R}}({}^G_{I_{i,k}}\mathbf{R})^{-1}, \quad \boldsymbol{\Gamma}_1 = {}^G\hat{\mathbf{v}}_{I_{i,k}} - {}^G_{I_{i,k}}\widetilde{\mathbf{R}}{}^G\mathbf{v}_{I_{i,k}}$$
$$\boldsymbol{\Gamma}_2 = {}^G\hat{\mathbf{p}}_{I_{i,k}} - {}^G_{I_{i,k}}\widetilde{\mathbf{R}}{}^G\mathbf{p}_{I_{i,k}}, \quad \boldsymbol{\Gamma}_3 = {}^G\hat{\mathbf{p}}_f - {}^G_{I_{i,k}}\widetilde{\mathbf{R}}{}^G\mathbf{p}_f, \tag{3}$$

and $\widetilde{\mathbf{B}}_{i,k} = [(\hat{\mathbf{b}}_{a_{i,k}} - \mathbf{b}_{a_{i,k}})^\top, (\hat{\mathbf{b}}_{\omega_{i,k}} - \mathbf{b}_{\omega_{i,k}})^\top]^\top$. By applying the log-linear property of the invariant error [9], $\boldsymbol{\eta}_{i,k}$ can be approximated by the following first-order approximation[1]

$$\boldsymbol{\eta}_{i,k} = \exp_{\mathcal{G}}(\boldsymbol{\xi}_{i,k}) \approx \mathbf{I}_6 + \boldsymbol{\xi}_{\boldsymbol{i,k}}{}^\wedge \in \mathbb{R}^{6\times 6}, \tag{4}$$

[1] $\exp_{\mathcal{G}}(\cdot)$ represents the exponential map that maps elements in Lie algebra to the matrix Lie group. Details are given in the supplementary material.

where $(\cdot)^\wedge : \mathbb{R}^{\dim \mathfrak{g}} \to \mathfrak{g}$ is the linear map that transforms the error vector $\boldsymbol{\xi}_{i,k}$ defined in the Lie algebra to its corresponding matrix representation [9], and

$$\begin{aligned}
\boldsymbol{\xi}_{i,k} &\triangleq \left[(\boldsymbol{\xi}_{\theta_{i,k}})^\top \quad (\boldsymbol{\xi}_{v_{i,k}})^\top \quad (\boldsymbol{\xi}_{p_{i,k}})^\top \quad (\boldsymbol{\xi}_{f_k})^\top\right]^\top \in \mathbb{R}^{12} \\
\boldsymbol{\xi}_{\theta_{i,k}} &= \widetilde{\boldsymbol{\theta}}_{i,k} = \log({}^G_{I_{i,k}}\widetilde{\mathbf{R}}) \in \mathbb{R}^3 \\
\boldsymbol{\xi}_{v_{i,k}} &= {}^G\hat{\mathbf{v}}_{I_{i,k}} - (\mathbf{I}_3 + \lfloor \widetilde{\boldsymbol{\theta}}_{i,k} \times \rfloor){}^G\mathbf{v}_{I_{i,k}} \in \mathbb{R}^3 \\
\boldsymbol{\xi}_{p_{i,k}} &= {}^G\hat{\mathbf{p}}_{I_{i,k}} - (\mathbf{I}_3 + \lfloor \widetilde{\boldsymbol{\theta}}_{i,k} \times \rfloor){}^G\mathbf{p}_{I_{i,k}} \in \mathbb{R}^3 \\
\boldsymbol{\xi}_{f_k} &= {}^G\hat{\mathbf{p}}_f - (\mathbf{I}_3 + \lfloor \widetilde{\boldsymbol{\theta}}_{i,k} \times \rfloor){}^G\mathbf{p}_f \in \mathbb{R}^3.
\end{aligned} \tag{5}$$

We define the overall error state of each robot i at time t_k as

$$\widetilde{\mathbf{x}}_{i,k} = \begin{bmatrix} \boldsymbol{\xi}_{i,k}^\top & \widetilde{\mathbf{B}}_{i,k}^\top \end{bmatrix}^\top \in \mathbb{R}^{18}. \tag{6}$$

System motion model: The motion model for each robot i in the network adheres to a general 3-D rigid body continuous-time kinematics described as follows:

$$\begin{aligned}
&{}^G_{I_{i,k}}\dot{\mathbf{R}} = {}^G_{I_{i,k}}\mathbf{R}\lfloor {}^{I_{i,k}}\boldsymbol{\omega} \times \rfloor, \quad {}^G\dot{\mathbf{v}}_{I_{i,k}} = {}^G_{I_{i,k}}\mathbf{R}({}^{I_{i,k}}\mathbf{a}) + {}^G\mathbf{g} \\
&{}^G\dot{\mathbf{p}}_{I_{i,k}} = {}^G\mathbf{v}_{I_{i,k}}, \quad {}^G\dot{\mathbf{p}}_f = \mathbf{0}, \quad \dot{\mathbf{b}}_{\omega_{i,k}} = \mathbf{w}_{\omega_i}, \quad \dot{\mathbf{b}}_{a_{i,k}} = \mathbf{w}_{a_i}
\end{aligned} \tag{7}$$

where ${}^G\mathbf{g} = \begin{bmatrix} 0 & 0 & -9.8 \end{bmatrix}^\top$ denotes the gravity vector. ${}^{I_{i,k}}\mathbf{a}$ and ${}^{I_{i,k}}\boldsymbol{\omega}$ represent the acceleration and angular velocity in the robot i's own frame, respectively, which can be measured by the IMU. The measurement model is given as

$$\begin{aligned}
{}^{I_{i,k}}\mathbf{a}_m &= {}^{I_{i,k}}\mathbf{a} - {}^G_{I_{i,k}}\mathbf{R}^\top {}^G\mathbf{g} + \mathbf{n}_{a_{i,k}} + \mathbf{b}_{a_{i,k}} \\
{}^{I_{i,k}}\boldsymbol{\omega}_m &= {}^{I_{i,k}}\boldsymbol{\omega} + \mathbf{n}_{\omega_{i,k}} + \mathbf{b}_{\omega_{i,k}},
\end{aligned} \tag{8}$$

where $\mathbf{n}_{a_i,k}$ and $\mathbf{n}_{\omega_i,k}$ are assumed to be zero mean Gaussian noises. $\mathbf{b}_{\omega_{i,k}}$ and $\mathbf{b}_{a_{i,k}}$ are the biases which are modeled as random walk process, where their time derivatives follow a Gaussian distribution.

Measurement models: We consider two types of exteroceptive sensor measurement models. The first type of measurement is the visual measurement model, denoted as $\mathbf{z}_{i,k}$. When a static environmental feature ${}^G\mathbf{p}_f$ is tracked by the robot's camera, the corresponding measurement model is given by

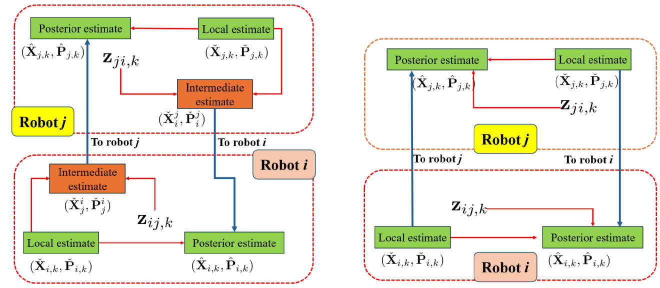
$$\begin{aligned}
\mathbf{z}_{i,k} &= \boldsymbol{\Pi}({}^I\mathbf{p}_f) + \mathbf{n}_{c_{i,k}} \\
{}^I\mathbf{p}_f &\triangleq \begin{bmatrix} x & y & c \end{bmatrix}^\top = {}^G_I\mathbf{R}^\top \left({}^G\mathbf{p}_f - {}^G\mathbf{p}_{I_{i,k}}\right)
\end{aligned} \tag{9}$$

where $\boldsymbol{\Pi}(\begin{bmatrix} x & y & z \end{bmatrix}^\top) = \begin{bmatrix} x/z & y/z \end{bmatrix}^\top$ denotes the projection function, and $\mathbf{n}_{c_{i,k}} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_i)$ represents the measurement noise assumed to be zero-mean Gaussian. The second type involves relative measurements between two robots i and j, which is typically a function of the relative pose between the two robots, expressed as follows

$$\mathbf{z}_{ij,k} = \mathbf{h}_{ij}(\mathbf{X}_{i,k}, \mathbf{X}_{j,k}, \mathbf{n}_{ij,k}) \tag{10}$$

where $\mathbf{n}_{ij,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{ij})$ denotes the relative measurement noise, which is also assumed to be zero-mean white Gaussian.

Remark 1: In the algorithm derivation, the relative measurement $\mathbf{h}_{ij}(\cdot)$ is modeled as a general nonlinear measurement model without specifying the exact measurement type. In Section IV-C, we present two representative examples of measurement types (the relative distances and relative positions between robots) to analyze how different types of relative measurements influence the system's performance.



(a) Loosely-coupled fusion (b) Tightly-coupled fusion

Fig. 1. Comparison of Distributed Fusion Strategies: Loosely-coupled vs. Tightly-coupled approaches. Red arrows indicate the transmission of a robot's own information, while blue arrows represent the use of a neighbor's information.

B. Problem Analysis

To achieve efficient and consistent multi-robot CL, we found two major challenges: how to efficiently fuse the inter-robot information in a fully distributed manner, and how to ensure the observability consistency of the system.

1) Distributed Fusion of Inter-Robot Information: In multi-robot CL, relative measurements introduce implicit correlations between robots, which are generally unknown in distributed settings. To handle this, one can adopt either loosely- or tightly-coupled fusion strategies as shown in Fig. 1.

The differences between Loosely and Tightly-coupled fusions are detailed as follows:

Loosely-coupled fusion: As shown in Fig. 1(a), robot i first estimates its own state using onboard sensors as $(\check{\mathbf{X}}_{i,k}, \check{\mathbf{P}}_{i,k})$, and then uses the relative measurement $\mathbf{z}_{ij,k}$ to infer an estimate of robot j's state, denoted as $(\check{\mathbf{X}}^i_{j,k}, \check{\mathbf{P}}^i_{j,k})$. This estimate is transmitted to robot j, which fuses it with its own local estimate. Since this approach indirectly incorporates relative information at the state level, it requires the neighbor's state to be fully observable from the measurement, and is thus only applicable to certain types of relative observations [5].

Tightly-coupled fusion: This paper considers tightly-coupled fusion (Fig. 1(b)), which directly incorporates relative measurements and neighbor information into the local state update. For instance, robot i uses $\mathbf{z}_{ij,k}$ from neighbor j to update its estimate $(\check{\mathbf{X}}_{i,k}, \check{\mathbf{P}}_{i,k})$. This enables more flexible use of relative information, as full observability is not required. However, performing such fusion on Lie groups while ensuring that the state remains on the manifold is nontrivial.

2) Observability Consistency of CL: Observability reflects how well the system states can be recovered from measurements and is critical for consistent CL. While the observability of single-robot 3-D localization is well understood, the introduction of inter-robot fusion in 3-D CL systems makes the analysis more complex. It is therefore essential to examine how relative information fusion affects the observability directions and whether it preserves consistency.

Based on the above analysis, we define **the problem of interest** as follows: Given the process and measurement models described in (7), (9), and (10), each robot aims to estimate their states and fuse the information in a tightly-coupled and distributed manner, while ensuring the observability consistency of the entire CL system throughout the estimation process.

IV. Algorithm Design

In this section, we propose a consistent distributed CL algorithm for 3-D multi-robot systems, consisting of a local estimation and a tightly coupled fusion step.

A. Local Estimation

The local estimation step follows the general procedure of the InEKF, where each robot i uses its own sensor measurements to estimate its own state independently. Given the posterior estimate $(\hat{\mathbf{X}}_{i,k-1}, \hat{\mathbf{P}}_{i,k-1})$ of the previous timestep $k-1$, each robot i first performs the propagation to compute a prior estimate, denoted as $(\bar{\mathbf{X}}_{i,k}, \bar{\mathbf{P}}_{i,k})$. In the propagation step, the state $\bar{\mathbf{X}}_{i,k}$ of the i's robot is propagated forward using its own IMU measurements ${}^{I_{i,k}}\mathbf{a}_m$ and ${}^{I_{i,k}}\boldsymbol{\omega}_m$ based on kinematics (7). To propagate the state covariance, we first linearize the continuous-time kinematics (7) to compute a linearized error-state model as

$$\frac{d}{dt}\widetilde{\mathbf{x}}_{i,k} \triangleq \mathbf{F}_{i,k}\widetilde{\mathbf{x}}_{i,k} + \mathbf{G}_{i,k}\mathbf{n}_i, \tag{11}$$

where $\widetilde{\mathbf{x}}_{i,k}$ is the overall error state of each robot i defined in (6); $\mathbf{F}_{i,k}$ and $\mathbf{G}_{i,k}$ are corresponding Jacobians. Given the linearized system in (11), the error covariance can be propagated as

$$\bar{\mathbf{P}}_{i,k} = \boldsymbol{\Phi}_{i,k}\hat{\mathbf{P}}_{i,k-1}\boldsymbol{\Phi}_{i,k}^{\top} + \mathbf{O}_i, \tag{12}$$

where $\mathbf{O}_i$ denotes the discrete noise covariance matrix; $\boldsymbol{\Phi}_{i,k}$ represents the discrete-time state transition matrix from time t_k to t_{k+1} which can be simply computed as $\boldsymbol{\Phi}_{i,k} = \exp(\mathbf{F}_{i,k}\delta t)$, $\delta t = t_k - t_{k-1}$. We provide more detailed derivations in *Section 2* of the supplementary material.

Once the robot's camera captures the environmental features, the robot will update the prior estimate for a local estimate, denoted as $(\check{\mathbf{X}}_{i,k}, \check{\mathbf{P}}_{i,k})$. Specifically, with the definition of error states in (5), we linearize the camera measurements at the prior estimate and compute the Jacobians as

$$\begin{aligned} \widetilde{\mathbf{z}}_{i,k} &= \mathbf{H}_{c,i}\widetilde{\mathbf{x}}_{i,k} + \mathbf{n}_{c_{i,k}} \\ \mathbf{H}_{c,i} &= \mathbf{H}_{p_i}\, {}^{G}_{I_{i,k}}\bar{\mathbf{R}}^{\top}\begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 6}\end{bmatrix} \end{aligned} \tag{13}$$

to perform the measurement update (detailed derivation is given in *Section 3.1* of the supplementary material).

B. Tightly-Coupled Fusion for Relative Information

If robot i has access to the relative robot-to-robot measurement $\mathbf{z}_{ij,k}$ at the current timestep k, the robot will fuse this relative information with its local estimate to compute a posterior estimate, denoted as $(\hat{\mathbf{X}}_{i,k}, \hat{\mathbf{P}}_{i,k})$. As we analyzed before, the relative measurement $\mathbf{z}_{ij,k}$ depends on both the states of related robot i and j. To update the local estimate with the relative measurements, we first build a linear residual system of the relative measurement by linearizing the measurement $\mathbf{z}_{ij,k}$ in (10) as

$$\widetilde{\mathbf{z}}_{ij,k} = \mathbf{H}_{r,i}\widetilde{\mathbf{x}}_{i,k} + \mathbf{H}_{r,j}\widetilde{\mathbf{x}}_{j,k} + \mathbf{G}_{ij}\mathbf{n}_{ij,k}, \tag{14}$$

where $\mathbf{H}_{r,i}$ and $\mathbf{H}_{r,j}$ are the measurement Jacobian with respect to the state of robot i and j at time k, respectively. Although these Jacobians are inherently time-dependent, we omit the explicit notation of k for simplicity and readability. $\widetilde{\mathbf{x}}_{i,k}$ and $\widetilde{\mathbf{x}}_{j,k}$ are the corresponding error state vector in consistent with (6). $\mathbf{G}_{ij}$ is the jacobian with respect to the measurement noise $\mathbf{n}_{ij,k}$. Let $\mathcal{N}_{i,k}$ denotes the neighbor set of robot i where the relative measurements $\mathbf{z}_{ij,k}$ are accessible by robot i, given as $\mathcal{N}_{i,k} = \{j_1, \ldots, j_L\}$. By using all available relative measurements $\mathbf{z}_{ij,k}$, we obtain a stacked linear residual systems as

$$\begin{aligned} \widetilde{\mathbf{z}}_{i,k} &= \begin{bmatrix}\widetilde{\mathbf{z}}_{ij_1,k}\\ \widetilde{\mathbf{z}}_{ij_2,k}\\ \vdots \\ \widetilde{\mathbf{z}}_{ij_L,k}\end{bmatrix} = \widetilde{\mathbf{H}}_i \begin{bmatrix}\widetilde{\mathbf{x}}_{i,k}\\ \widetilde{\mathbf{x}}_{j_1,k}\\ \vdots \\ \widetilde{\mathbf{x}}_{j_L,k}\end{bmatrix} + \begin{bmatrix}\mathbf{G}_{ij_1}\mathbf{n}_{ij_1,k}\\ \mathbf{G}_{ij_2}\mathbf{n}_{ij_2,k}\\ \vdots \\ \mathbf{G}_{ij_L}\mathbf{n}_{ij_L,k}\end{bmatrix} \\ \widetilde{\mathbf{H}}_i &= \begin{bmatrix} \mathbf{H}_{r,i}^{ij_1} & \mathbf{H}_{r,j_1}^{ij_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{H}_{r,i}^{ij_2} & \mathbf{0} & \mathbf{H}_{r,j_2}^{ij_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{r,i}^{ij_L} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{H}_{r,j_L}^{ij_L} \end{bmatrix} \end{aligned} \tag{15}$$

For simplicity, we reorganize the structure $\widetilde{\mathbf{H}}_i$ in (15), where each column of $\widetilde{\mathbf{H}}_i$ is rewritten as a block matrix:

$$\begin{bmatrix}\widetilde{\mathbf{z}}_{ij_1,k}\\ \widetilde{\mathbf{z}}_{ij_2,k}\\ \vdots \\ \widetilde{\mathbf{z}}_{ij_L,k}\end{bmatrix} \triangleq \begin{bmatrix}\bar{\mathbf{H}}_{r,i} & \bar{\mathbf{H}}_{r,j_1} & \cdots & \bar{\mathbf{H}}_{r,j_L}\end{bmatrix} \begin{bmatrix}\widetilde{\mathbf{x}}_{i,k}\\ \widetilde{\mathbf{x}}_{j_1,k}\\ \vdots \\ \widetilde{\mathbf{x}}_{j_L,k}\end{bmatrix} + \widetilde{\mathbf{n}}_i, \tag{16}$$

and $\widetilde{\mathbf{n}}_i$ denotes the stacked noise vector.

It is important to note that if robot i has used the relative measurement $\mathbf{z}_{ij,k}$, which depends on both robot i's and j's states, to update its own state $\mathbf{X}_{i,k}$, these two robots' states will become correlated afterwards. In other words, the cross covariances between each local filter are non-zero, which is generally unknown and nontrivial to handle in a fully distributed setting. Thus, (16) cannot be directly used to perform the InEKF update.

To properly handle the unknown correlation term and ensure the consistency of the estimator, we leverage the covariance intersection (CI) algorithm [16] that satisfies the following condition to approximate the covariance matrix:

$$\mathrm{diag}\left(\frac{1}{\boldsymbol{\omega}_{i,k}}\check{\mathbf{P}}_{i,k}, \ldots, \frac{1}{\boldsymbol{\omega}_{j_L,k}}\check{\mathbf{P}}_{j_L,k}\right) \succ \begin{bmatrix}\check{\mathbf{P}}_{i,k} & \cdots & \check{\mathbf{P}}_{ij_L,k}\\ \vdots & \ddots & \vdots \\ \check{\mathbf{P}}_{ij_L,k}^{\top} & \cdots & \check{\mathbf{P}}_{j_L,k}\end{bmatrix} \tag{17}$$

where the right-hand side denotes the true covariance matrix with cross-covariance terms and the left-hand side is the approximated covariance matrix by CI [16]. ω_i, ω_{j_l} are the weights which satisfy

$$\boldsymbol{\omega}_{i,k} + \sum_{j\in\mathcal{N}_{i,k}} \boldsymbol{\omega}_{j,k} = 1, \boldsymbol{\omega}_{i,k} > 0, \boldsymbol{\omega}_{j,k} > 0. \tag{18}$$

Algorithm 1: Cooperative Localization for Robot i.

Input: IMU measurements ${}^{I_{i,k}}\mathbf{a}_m$, ${}^{I_{i,k}}\boldsymbol{\omega}_m$, camera measurements $\mathbf{z}_{i,k}$, relative measurements $\mathbf{z}_{ij,k}$ from $j \in \mathcal{N}_{i,k}$ at each timestep k, and the initial estimate $(\hat{\mathbf{X}}_{i,0}, \hat{\mathbf{P}}_{i,0})$.

```
for each timestep k = 1, 2, ... do
    /*Local Estimation*/
    Obtain the prior estimate (X̄_{i,k}, P̄_{i,k}) by
    propagating the previous estimate (X̂_{i,k-1}, P̂_{i,k-1})
    using the IMU kinematics (7).
    if a camera observation z_{i,k} is available then
        Compute (X̌_{i,k}, P̌_{i,k}) through the local update.
    else
        (X̌_{i,k}, P̌_{i,k}) ← (X̄_{i,k}, P̄_{i,k}).
    end if
    /*Tightly-Coupled Fusion*/
    if relative measurements z_{ij,k} are available then
        Communicate with neighbors j ∈ N_{i,k} to obtain
        their estimates (X̌_{j,k}, P̌_{j,k}) and build the stacked
        linearized system (16).
        Determine the optimal CI weights.
        Compute the posterior estimate (X̂_{i,k}, P̂_{i,k}) by
        updating the local estimate using (19) and (20).
    else
        (X̂_{i,k}, P̂_{i,k}) ← (X̌_{i,k}, P̌_{i,k}).
    end if
end for
```

It is worth noting that the weight terms for each robot can either be optimized online during the estimation process based on various performance criteria [24] or fixed to enhance computational efficiency [16]. We discuss the trade-offs between these two strategies in the supplementary material. By substituting the approximated covariance (17) into the update step of InEKF, the relative measurements can be used to update the local estimate of each robot. In particular, robot i's covariance can be updated as

$$\hat{\mathbf{P}}_{i,k} = \frac{1}{\boldsymbol{\omega}_{i,k}}\check{\mathbf{P}}_{i,k} - \frac{1}{\boldsymbol{\omega}_{i,k}^2}\check{\mathbf{P}}_{i,k}\bar{\mathbf{H}}_{r,i}^{\top}\mathbf{S}_k^{-1}\bar{\mathbf{H}}_{r,i}\check{\mathbf{P}}_{i,k}$$

$$\mathbf{S}_{i,k} = \sum_{j\in(\{i\}\cup\mathcal{N}_{i,k})} \frac{1}{\boldsymbol{\omega}_{j,k}}\bar{\mathbf{H}}_{r,j}\check{\mathbf{P}}_{j,k}\bar{\mathbf{H}}_{r,j}^{\top} + \mathbf{R}_i, \tag{19}$$

where $\mathbf{R}_i$ denotes the covariance of the stacked noise vector $\tilde{\mathbf{n}}_i$; and the state is updated as

$$\hat{\mathbf{X}}_{i,k} = \check{\mathbf{X}}_{i,k} \boxplus \boldsymbol{\varepsilon}_{i,k} = \begin{bmatrix} \exp(\boldsymbol{\varepsilon}_{A_{i,k}})\check{\mathbf{T}}_{i,k} \\ \check{\mathbf{B}}_{i,k} + \boldsymbol{\varepsilon}_{B_{i,k}} \end{bmatrix}$$

$$\boldsymbol{\varepsilon}_{i,k} \triangleq \begin{bmatrix} \boldsymbol{\varepsilon}_{A_{i,k}} \\ \boldsymbol{\varepsilon}_{B_{i,k}} \end{bmatrix} = \frac{1}{\boldsymbol{\omega}_{i,k}}\hat{\mathbf{P}}_{i,k|k-1}\bar{\mathbf{H}}_{i,k}^{\top}\mathbf{S}_{i,k}^{-1}\tilde{\mathbf{z}}_{i,k} \tag{20}$$

Remark 2: The correction term $\boldsymbol{\varepsilon}_{i,k} = \begin{bmatrix} \boldsymbol{\varepsilon}_{A_{i,k}}^{\top} & \boldsymbol{\varepsilon}_{B_{i,k}}^{\top} \end{bmatrix}^{\top}$ is applied to the state estimate $\check{\mathbf{X}}_{i,k}$. It consists of two components: $\boldsymbol{\varepsilon}_{A_{i,k}}$ and $\boldsymbol{\varepsilon}_{B_{i,k}}$, which are used to correct the Lie group component $\check{\mathbf{T}}_{i,k}$ and the vector component $\check{\mathbf{B}}_{i,k}$ of the state $\check{\mathbf{X}}_{i,k}$, respectively.

C. Observability Analysis

Observability plays a crucial role in analyzing system consistency, as it determines whether the system can recover its initial state using all the available measurements. Following the proof of Theorem 1 of [6], the local observability matrix for the time-varying error state of the whole system is defined as

$$\mathcal{O} = \begin{bmatrix} \mathcal{O}_0 \\ \mathcal{O}_1 \\ \vdots \\ \mathcal{O}_k \end{bmatrix} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1\bar{\boldsymbol{\Phi}}_{1|0} \\ \vdots \\ \mathbf{H}_k\bar{\boldsymbol{\Phi}}_{k|0} \end{bmatrix}, \tag{21}$$

where $\bar{\boldsymbol{\Phi}}_{k,0} = \mathrm{diag}(\boldsymbol{\Phi}_{1,k|0}, \ldots, \boldsymbol{\Phi}_{m,k|0})$ is the joint state transition matrix of all the robots; $\mathbf{H}_k$ denotes the measurement Jacobian. Since the derivation is formulated for a general nonlinear measurement model, we illustrate the exact observability by analyzing two common cases: relative-position and distance-only measurements.

Distance-only Measurement: The distance measurement between two robots i and j can be modeled as a function of their positions with noise $\mathbf{n}_{ij,k}$:

$$\mathbf{z}_{ij,k} = \|{}^G\mathbf{p}_{I_{i,k}} - {}^G\mathbf{p}_{I_{j,k}}\|_2 + \mathbf{n}_{ij,k} \tag{22}$$

and the corresponding measurement Jacobian in (14) at the local estimate $\check{\mathbf{X}}_{i,k}$ is computed as

$$\mathbf{H}_{r,i} = \frac{{}^G\check{\mathbf{p}}_{ij}^{\top}}{\|{}^G\check{\mathbf{p}}_{ij}\|_2}\begin{bmatrix} -\lfloor {}^G\check{\mathbf{p}}_{I_{i,k}}\times\rfloor & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 9} \end{bmatrix}$$

$$\mathbf{H}_{r,j} = \frac{{}^G\check{\mathbf{p}}_{ij}^{\top}}{\|{}^G\check{\mathbf{p}}_{ij}\|_2}\begin{bmatrix} \lfloor {}^G\check{\mathbf{p}}_{I_{j,k}}\times\rfloor & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_{3\times 9} \end{bmatrix} \tag{23}$$

Theorem 1: If each robot undergoes general motion in the 3-D environment, the multi-robot system has unobservable directions, denoted as $\mathbf{N}$, given by

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1^{\top} & \cdots & \mathbf{N}_i^{\top} & \cdots & \mathbf{N}_m^{\top} \end{bmatrix}^{\top}$$

$$\mathbf{N}_i = \begin{bmatrix} {}^G\mathbf{g}^{\top} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 6} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 6} \end{bmatrix},$$

$$\text{for } i = \{1, \ldots, m\} \tag{24}$$

where $\mathcal{O}_k\mathbf{N} = \mathbf{0}$. The detailed derivation is given in *Section 4* of the supplementary material.

Theorem 1 shows that the multi-robot CL system remains unobservable with four directions, corresponding to the global yaw and position. In general, with m robots running independent visual–inertial localization (without relative measurements), the system exhibits $4m$ unobservable directions, since each visual–inertial system inherently has four [20]. Incorporating relative measurements reduces these to four, as global yaw and position remain unobservable while inter-robot relative poses become observable. Thus, relative measurements enhance localization by increasing the number of observable directions.

Lemma 1: As the unobservable space $\mathbf{N}$ is unrelated to the system state and only consists of constant values, the system unobservable space will not be affected by system linearization change. Thus, the correct observability of the overall CL system is preserved naturally with our proposed fusion algorithm.

TABLE I
SIMULATION PARAMETERS. ALL THE ALGORITHMS AND ROBOTS USE THE EXACT SAME PARAMETERS AS THE TABLE (REL MEAS. NOISE MEANS RELATIVE MEASUREMENT NOISE)

Parameter	Value	Parameter	Value
Cam Noise(pixel)	1	Rel Meas. Noise(m)	0.10
Gyro Noise(rad/(s$\sqrt{\text{Hz}}$))	2.0e-3	Gyro Bias	3.0e-4
Accel Noise (m/(s$^2\sqrt{\text{Hz}}$))	3.0e-3	Accel Bias	3.0e-4

Relative-position Measurement: The relative position measurement between two robots i and j can be modeled as:

$$\mathbf{z}_{ij,k} = \left({}^G\mathbf{p}_{I_{i,k}} - {}^G\mathbf{p}_{I_{j,k}}\right) + \mathbf{n}_{ij,k} \tag{25}$$

Following the similar line as the previous analysis, we have the following Theorem:

Theorem 2: If each robot undergoes general motion in the 3-D environment, the unobservable directions of the CL system, denoted as $\mathbf{N}$, are given as

$$\mathbf{N} = \begin{bmatrix}\mathbf{N}_1^\top & \cdots & \mathbf{N}_i^\top & \cdots & \mathbf{N}_m^\top\end{bmatrix}^\top$$
$$\mathbf{N}_i = \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 6}\end{bmatrix}, \text{ for } i = \{1, \ldots, m\} \tag{26}$$

where $\mathcal{O}_k\mathbf{N} = \mathbf{0}$. Similarly, by fusing the relative position measurements as additional geometric constraints between robots, the unobservable directions of the CL system has reduced from $4m$ to 3. Due to the space limitations, we provided a more detailed proof and analysis in the supplementary material.

V. SIMULATION EXPERIMENTS

A. Simulation Setup

In this section, we implement the proposed algorithm in a nontrivial 3-D CL scenarios with six robots following distinct pre-designed trajectories. For each trajectory, the algorithm is evaluated using two types of relative measurements: relative-pose and relative-distance-only observations, both sampled at 10 Hz. Each robot is equipped with an IMU (100 Hz) for ego-motion sensing and a monocular camera (10 Hz) that intermittently observes six randomly placed static features. To simulate limited inter-robot communication, relative measurements from neighbors are available with a detection probability of 60% at each timestep. All robots are initialized with the ground-truth state, and the initial state covariance is set as

$$\hat{\mathbf{P}}_{i,0} = \mathbf{diag}\left(0.02^2\mathbf{I}_3, 0.05^2\mathbf{I}_3, 0.01^2\mathbf{I}_9\right) \tag{27}$$

We conduct 100 Monte Carlo trials using the simulation parameters in Table I. All algorithms share identical parameter settings and are tested on the same simulated datasets. In each trial, robots simultaneously perform ego-motion estimation, inter-robot communication, and local sensing. Performance is evaluated using root mean squared error (RMSE) for accuracy and normalized estimation error squared (NEES) for consistency. For a consistent estimator, NEES values for position and orientation should remain below their respective degrees of freedom (3 each). Final results are averaged over all runs. We compare the performance of the proposed algorithm with several baseline methods. These include the standard Extended Kalman Filter (EKF) and the Lie group-based Invariant EKF (InEKF) [21], both of which are widely used for single-robot localization but cannot incorporate inter-robot relative measurements. We also consider a naïve fusion method that fuses relative measurements without accounting for cross-correlations. Although such a method can exploit relative information, it typically leads to inconsistency. While consistent methods such as the loosely coupled approach proposed in [5] exist, they are designed specifically for full relative-pose measurements and are not applicable to partial inter-robot observations such as relative distances, as illustrated in Fig. 1(b).

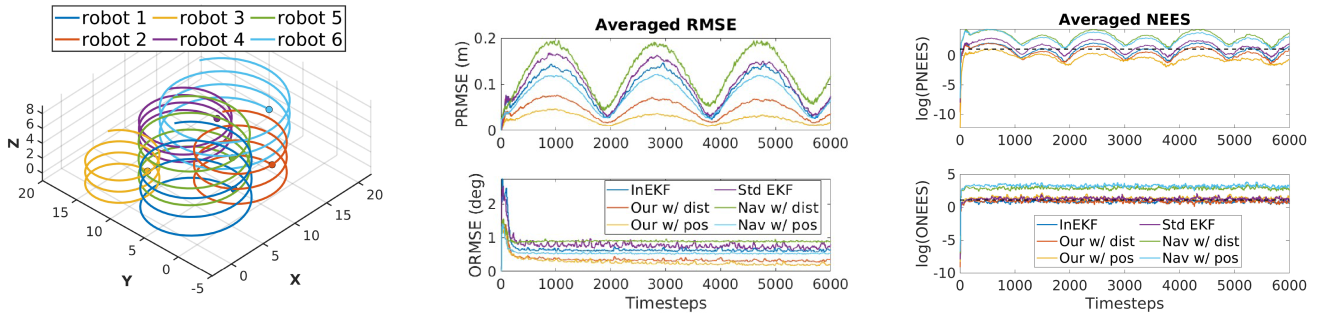
TABLE II
AVERAGED RMSE AND NEES ERRORS OVER 100 MONTE-CARLO RUNS

	Method	PRMSE	ORMSE	PNEES	ONEES
Traj. 1	**Our w/ pos**	0.092	1.013	3.122	3.043
	Our w/ dist	0.128	1.475	3.255	2.877
	EKF	0.165	1.506	3.899	3.675
	InEKF	0.143	1.402	3.096	3.012
	Nai. w/ dist	0.139	1.672	37.088	29.803
	Nai. w/ pos	0.098	1.875	32.124	31.116

B. Simulation Results

1) Experiment 1 (fusion with relative distances): In this case, we consider that each robot is equipped with an Ultra-Wideband (UWB) sensor to measure inter-robot distances as relative information between robots. Fig. 2 shows the NEES and RMSE of the robot team over time, averaged across 100 Monte Carlo runs for each algorithm. Table II summarizes the average RMSE and NEES across 100 Monte Carlo runs. The results show that the proposed method significantly improves both position and orientation accuracy compared with single-robot localization algorithms (standard EKF and InEKF) by effectively fusing inter-robot distance measurements. Unlike the loosely-coupled approach, which requires full relative-pose observations, our method can incorporate partial observations, making it more general and applicable to diverse sensor configurations while still achieving superior accuracy. Moreover, the proposed algorithm exhibits the slowest RMSE growth, highlighting its improved long-term robustness. The table further shows that our method maintains consistency in both position and orientation estimates by properly fusing relative information while preserving the system's observability properties, as established in Lemma 1. By contrast, the naive fusion method and standard EKF are inconsistent: the former neglects inter-robot correlations, while the latter suffers from linearization errors that distort observability. The InEKF remains consistent but is less accurate than our approach. Finally, results reported in the supplementary material [11] for trajectory set 2 show that the position NEES of our method falls below the expected value of 3, indicating consistent yet slightly conservative estimation. This behavior stems from the use of the CI algorithm, which yields conservative estimates under unknown cross-correlations [5]. In 3-D Lie group settings, where the nonlinear manifold structure introduces additional complexity, CI tends to be even more conservative in order to ensure consistency. Despite this conservativeness, our method consistently achieves accurate and robust localization performance.

2) Experiment 2 (fusion with relative positions): In this case, each robot can measure the relative position between other robots as relative information, which can provide more information compared with the distance-only measurements. As shown



(a) Trajectories set 1

(b) Averaged RMSE for both positions and orientations for trajectories 1.

(c) Averaged NEES for both positions and orientations for trajectories 1.

Fig. 2. Averaged estimation performance of the multi-robot system over 100 Monte Carlo trials using the proposed algorithm for two types of relative measurements (relative position and relative distance) on traj set 1, compared against the naive fusion method, InEKF, and the standard EKF.

TABLE III
PERFORMANCE UNDER VARIOUS RELATIVE-MEASUREMENT NOISES

Rel Meas. Noise(m)	PRMSE	ORMSE	PNEES	ONEES
0.05	0.037	0.601	3.012	2.788
0.12	0.052	0.687	2.365	2.404
0.15	0.054	0.682	2.076	2.401

in Figs. 2(b) and (c), incorporating relative position measurements significantly improves localization accuracy compared to both single-robot localization and fusion using only relative distance. This result is consistent with Theorem 2, which states that fusing relative position measurements improves observability by introducing more informative constraints than relative distance measurements, thereby enhancing overall localization accuracy. Due to space limitations, additional simulation results are provided in the supplementary material (see the Simulation section).

3) Experiment 3 (fusion under different noise level): To further evaluate the robustness and consistency of our fusion method, we test its performance on trajectory set 1 using relative-distance measurements under three noise levels. As shown in Table III, our distributed InEKF achieves low localization errors across all cases, even under high measurement noise. This demonstrates that the proposed method can effectively incorporate relative measurements despite significant uncertainty. We also observe that under severe noise, the NEES falls below its nominal value, indicating that the fusion algorithm becomes conservative. This conservativeness arises from CI fusion: by upper-bounding unknown cross-covariances, it inflates the covariance to ensure consistency. Moreover, since our approach operates on the SE(3) manifold in 3-D, the uncertainty propagation is more complex than in many 2-D methods, which further amplifies the CI inflation effect. These results highlight that CI fusion in high-dimensional spaces requires careful tuning to balance consistency and covariance tightness.

VI. Real-World Experiments

In this section, we evaluate our algorithm on the publicly available *UTIAS* 2-D multi-robot cooperative localization dataset [25]. Although the method is designed for 3-D scenarios, real-world 3-D experiments are highly challenging,

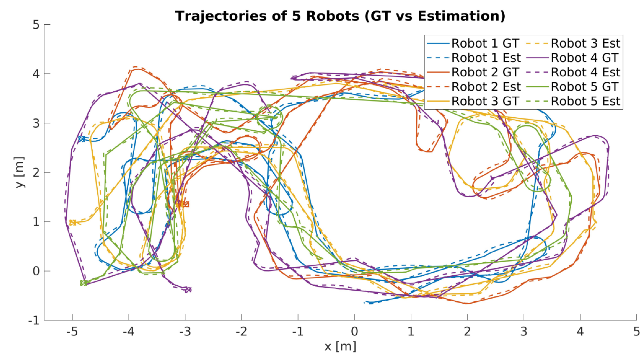


Fig. 3. Estimated trajectories versus ground truth of five robots in sub-dataset 1 of the *UTIAS* dataset.

and, to the best of our knowledge, no publicly available 3-D multi-robot dataset exists. The dataset was collected using five differential-drive robots operating in a $15\mathrm{m} \times 8\mathrm{m}$ indoor environment with 15 static landmarks. Each robot is equipped with a monocular camera with a 60° field of view, from which relative range and bearing measurements are obtained whenever another robot or landmark enters the view. Ego-motion data, including linear and angular velocities, is also recorded. Ground-truth robot trajectories and landmark positions in the global frame are provided by a Vicon motion capture system. The dataset comprises nine subsets with varying robot motion profiles, offering diverse and challenging scenarios for evaluation. Fig. 3 illustrates the estimated trajectories of all robots in sub-dataset 1, demonstrating the effectiveness of the proposed CL algorithm. To further evaluate its performance, we compare our method with a centralized approach and two state-of-the-art decentralized estimation methods: the decentralized dual fusion (DDF) method [26] and the distributed extended information filter (DEIF) [2]. It should be emphasized that these baselines are restricted to two-dimensional vector spaces and thus cannot be extended to the three-dimensional non-vector formulation considered in this work. This highlights the advantage of our approach, which directly operates on the 3-D Lie group structure, while the comparisons with vector-space baselines are provided only as a point of reference. In these experiments, the initial robot poses are generated by perturbing the ground truth with a $\pm 0.2\,\mathrm{m}$ offset in position and a $\pm 5^\circ$ offset in orientation, rather than using

TABLE IV
AVERAGED RMSE FOR SUB-DATASET 1

	Our w/ dist	DDF w/ dist	DEIF w/ dist	Centralized
PRMSE	1.280	1.272	1.495	1.125
ORMSE	0.675	0.669	0.690	0.671

the exact poses as in the previous simulation setup. As shown in Table IV, our algorithm achieves comparable performance. Considering that DDF requires two communication rounds per iteration, whereas our method needs only one, this highlights the superior communication efficiency of our approach while maintaining comparable accuracy. Additional experimental results are provided in *Section 5.2* of the supplementary material.

VII. CONCLUSION

In this letter, we proposed a novel distributed invariant Extended Kalman Filter (InEKF) for multi-robot CL in 3-D environments. Our method is fully distributed and accommodates general nonlinear inter-robot measurements by fusing the relative information in a tightly-coupled manner. Additionally, unlike most existing distributed CL approaches that operate in vector space and often fail to maintain observability consistency, our method directly fuses relative information on the Lie group, ensuring both consistency and efficiency for distributed CL in 3-D environments. We also experimentally validated the effectiveness of the proposed algorithms. For future work, we plan to extend our approach to multi-robot SLAM, leveraging common observations as additional relative information to further enhance estimation accuracy.

REFERENCES

[1] N. Hao, F. He, C. Tian, Y. Yao, and W. Xia, "KD-EKF: A consistent cooperative localization estimator based on Kalman decomposition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 11064–11070.

[2] P. Zhu and W. Ren, "Fully distributed joint localization and target tracking with mobile robot networks," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 4, pp. 1519–1532, Jul. 2021.

[3] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1412–1417.

[4] J. Yan, F. Zhu, Y. Huang, and Y. Zhang, "Robust decentralized cooperative localization for multi-robot system against measurement outliers," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 17934–17947, May 2024.

[5] H. Li, X. Li, S. Huang, C. Yang, and J. Wu, "Covariance intersection-based invariant Kalman filtering (DInCIKF) for distributed pose estimation," 2024, *arXiv:2409.07933*.

[6] G. Huang, M. Kaess, and J. J. Leonard, "Consistent unscented incremental smoothing for multi-robot cooperative target tracking," *Robot. Auton. Syst.*, vol. 69, pp. 52–67, 2015.

[7] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A first-estimates Jacobian EKF for improving SLAM consistency," in *Proc. 11th Int. Symp. Exp. Robot.*, O. Khatib, V. Kumar, and G. J. Pappas, Eds., 2009, pp. 373–382.

[8] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.

[9] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended Kalman filtering for robot state estimation," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 402–430, 2020.

[10] J. Xu, P. Zhu, Y. Zhang, and W. Ren, "Moving target estimation and active tracking in multi-robot systems," in *Proc. 62nd IEEE Conf. Decis. Control*, 2023, pp. 6972–6977.

[11] "Supplementary material of toward consistent and efficient fusion for 3-D distributed multi-robot systems: Problems, analysis, and algorithms," 2025. [Online]. Available: https://bit.ly/4jnhRgo

[12] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic SLAM for multi-robot systems," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2022–2038, Aug. 2022.

[13] R. He, Y. Shan, and K. Huang, "Robust cooperative localization with failed communication and biased measurements," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2997–3004, Mar. 2024.

[14] S. Fang, H. Li, and M. Yang, "Adaptive cubature split covariance intersection filter for multi-vehicle cooperative localization," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1158–1165, Apr. 2022.

[15] T.-M. Nguyen, Z. Qiu, T. H. Nguyen, M. Cao, and L. Xie, "Distance-based cooperative relative localization for leader-following control of MAVs," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3641–3648, Oct. 2019.

[16] P. Zhu, Y. Yang, W. Ren, and G. Huang, "Cooperative visual-inertial odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13135–13141.

[17] J. Xu, P. Zhu, and W. Ren, "Cooperative 3-D active multi-robot multi-target tracking," in *Proc. Amer. Control Conf.*, 2024, pp. 1317–1322.

[18] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based consistent EKF estimators for multi-robot cooperative localization," *Auton. Robots*, vol. 30, no. 1, pp. 99–122, 2011.

[19] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "An observability-constrained sliding window filter for SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 65–72.

[20] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, 2013.

[21] Y. Yang, C. Chen, W. Lee, and G. Huang, "Decoupled right invariant error states for consistent visual-inertial navigation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1627–1634, Apr. 2022.

[22] K. Wu, T. Zhang, D. Su, S. Huang, and G. Dissanayake, "An invariant-EKF vins algorithm for improving consistency," in *Proc. 2017 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1578–1585.

[23] J. Xu, P. Zhu, Y. Zhou, and W. Ren, "Distributed invariant extended Kalman filter using lie groups: Algorithm and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 6, pp. 2777–2789, Nov. 2023.

[24] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. Amer. Control Conf.*, 1997, vol. 4, pp. 2369–2373.

[25] K. Y. Leung, Y. Halpern, T. D. Barfoot, and H. H. Liu, "The utias multi-robot cooperative localization and mapping dataset," *Int. J. Robot. Res.*, vol. 30, no. 8, pp. 969–974, 2011.

[26] N. Hao, F. He, Y. Hou, W. Song, D. Xu, and Y. Yao, "Decentralized cooperative localization: A communication-efficient dual-fusion consistent approach," *IEEE Robot. Automat. Lett.*, vol. 10, no. 1, pp. 636–643, Jan. 2025.