IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Distributed Pose Graph Optimization via Contractive Belief Sharing

Xiangyu Liu, *Graduate Student Member, IEEE*, and Margarita Chli

***Abstract*—Following the relative maturity of single-robot Simultaneous Localization And Mapping (SLAM) techniques, works addressing collaborative SLAM have started emerging lately. Driven by the need for robust and scalable multi-robot systems, the community has been targeting Distributed Pose Graph Optimization (DPGO), with current DPGO methods falling into two categories: optimization-based methods providing favorable convergence properties at the expense of excessive communication rounds among participants, and belief-propagation methods that exhibit better scalability and faster computation, albeit risking divergence on loopy and noisy graphs. Inspired by the need for more effective DPGO techniques, this work introduces Contractive Belief Sharing (CBS), a two-stage message-passing algorithm that combines Maximum-A-Posteriori (MAP) optimization with belief propagation with a Hellinger-distance-based damping rule. In this way, CBS ensures fast and reliable convergence while maintaining lly distributed computation and communication with neighbors only. Experiments on benchmarks show that CBS reaches convergence substantially faster and more efficient and scalable than the state-of-the-art methods while maintaining high trajectory accuracy, opening up new capabilities for collaborative SLAM.**

***Index Terms*—Multi-robot SLAM, localization, SLAM.**

I. Introduction

MULTI-ROBOT SLAM is shifting from centralized to distributed systems to meet growing demands for scalability and robustness. Distributed Pose Graph Optimization (DPGO) is a key technique, enabling robots to estimate trajectories while preserving global consistency. A practical DPGO solver should converge reliably even on loopy, noisy graphs that arise in real environments, limiting the number of communication rounds to preserve bandwidth, while avoiding centralized syncing bottlenecks that inherently hinder scalability. Despite rapid progress, current methods still struggle to meet these goals simultaneously. Optimization-based methods converge reliably, but often need hundreds of communication rounds to stabilize. Belief propagation methods exchange beliefs with neighbor-only messages, offering great scalability but risking divergence or oscillatory behavior on loopy or noisy graphs without convergence guarantees.

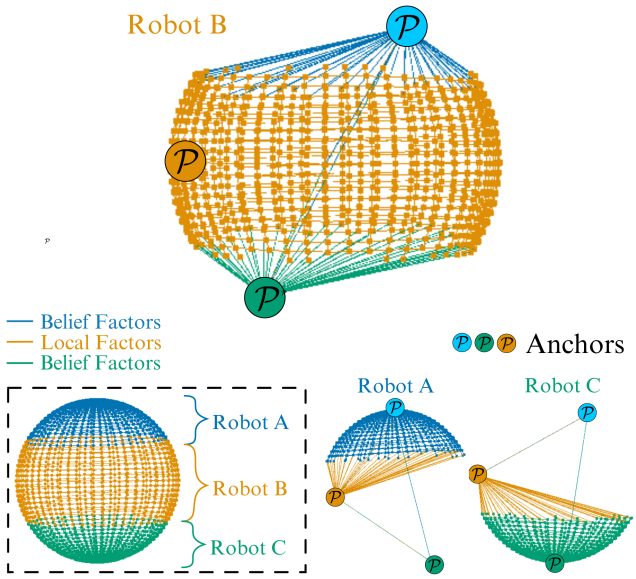


Fig. 1. Example of the factor graphs of CBS. Each robot optimizes its local pose graph, updates its pose beliefs before sharing them with its neighbors. The received beliefs are incorporated as belief factors connecting the advertised poses and the senders' anchors, ensuring globally consistent trajectory estimation and fast convergence.

In this work, we propose Contractive Belief Sharing (CBS, Fig. 1), a hybrid DPGO framework that combines the reliable convergence of MAP optimization with the scalability of belief propagation. CBS augments message passing with a Hellinger-distance-based damping rule that contracts belief updates, ensuring fast and stable convergence even on loopy graphs. Robots also estimate and share anchor-node beliefs to improve global consistency and accelerate convergence. To the best of our knowledge, this is the first DPGO method to regulate step size via belief divergence and to provide a formal convergence analysis, achieving an unprecedented balance of accuracy, speed, and communication efficiency.

II. Related Work

DPGO has been studied from multiple perspectives, with various methods addressing the challenges of accuracy and efficiency in different ways. Methods based on the paradigm of Alternating Direction Method of Multipliers (ADMM) [1], [2], such as the most recent MESA [3], [4] enforce consensus through multiplier updates, ensuring gradual convergence even

Received 15 August 2025; accepted 13 October 2025. Date of publication 14 November 2025; date of current version 25 November 2025. This article was recommended for publication by Associate Editor A. Vora and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported by the European Research Council (ERC) grant "SkEyes" under Grant 101089328. *(Corresponding author: Xiangyu Liu.)*

Xiangyu Liu and Margarita Chli are with the Vision for Robotics Lab, Department of Electrical and Computer Engineering, University of Cyprus, 1678 Aglantzia, Cyprus, and also with ETH Zurich, 8092 Zürich, Switzerland (e-mail: liu.xiangyu@ucy.ac.cy; chli.margarita@ucy.ac.cy).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3632725, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3632725

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

under communication constraints. DC2-PGO [5] and its asynchronous extension ASAPP [6], on the other hand, avoid explicit consensus constraints by employing a Riemannian Block Coordinate Descent (RBCD) solver combined with semi-definite relaxation techniques, improving robustness while maintaining distributed computation. Another established method DGS [7] adopts a Gauss-Seidel update scheme with chordal relaxation, partitioning optimization into sequential sub-problems to improve computational efficiency. Belief propagation methods offer an alternative by directly performing distributed probabilistic inference through local message passing. Gaussian Belief Propagation (GBP) methods [8], [9], [10] in particular, allows robots to update their estimates from neighbor's beliefs. This makes them well-suited for large-scale estimation problems as they only require local communication. However, a fundamental challenge in belief propagation is its convergence behavior, especially in graphs with loops or inconsistent measurements. Although GBP is guaranteed to converge in tree-structured graphs [11], [12], real-world SLAM problems involve cyclic, nonlinear and nonconvex constraints that can lead to oscillations and divergence. Apart from methods based on belief propagation, hybrid approaches such as DDF-SAM2 [13], [14] focus on sharing information on variables' marginals between robots while carefully avoiding double counting. These methods integrate marginal sharing mechanisms within MAP optimization, ensuring that beliefs are exchanged in a controlled manner. However, they often suffer from the inconsistency in solutions and convergence instability similar to belief propagation. This is an inherent challenge in marginal sharing approaches, particularly in scenarios with inconsistent constraints or delayed information exchange.

III. Problem Formulation

A DPGO problem involves a team of robots $\mathcal{R}$ collaboratively estimating their trajectories while navigating through an environment. Each robot $i \in \mathcal{R}$ collects intra-robot measurements Z_{ii} between poses along its own trajectory, and inter-robot measurements Z_{ij} with robots $j \in \mathcal{E}_i$, where $\mathcal{E}_i \subseteq \mathcal{R}$ is the set of robots that robot i observes. Each robot fixes its gauge freedom by adding a prior to a pose that fixes it as the origin of the local frame. Such a pose is referred to as its anchor $\mathcal{P}_{ii} = \mathbf{0}_d$ and each robot forms a local *anchored pose graph* [15], and maintains all poses in its local frame, similar to [16]. With the trajectory in each robot's local frame, the estimation of transformation between anchors is critical for the alignment of robots' trajectories, especially when global initialization is not available. For example, robot i estimates the anchor of neighbor j in its local frame as the variable $\mathcal{P}_{ij}$.

Robot i maintains a set of variables X_i in its local optimization state, such that

$$X_i = \mathcal{X}_{ii} \cup \{\mathcal{X}_{ij} \mid j \in \mathcal{E}_i\} \cup \{\mathcal{P}_{ik} \mid k \in \mathcal{R}\},$$

i.e. its own poses $\mathcal{X}_{ii}$, local estimates of other robots' poses $\mathcal{X}_{ij}$ it observed, and local estimates of all robots' anchors $\mathcal{P}_{ik}$, which includes its own anchor. All variables are estimated in the local frame of i. For convenience, we denote all pose variables i holds as $\mathcal{X}_i$ and all anchor variables $\mathcal{P}_i$.

All of the above variables lie on the Lie group $\mathrm{SE}(d)$, and the solver therefore uses the manifold operations $\oplus$ and $\ominus$ defined in [17]. For every measurement $z \in Z_i$ we create a non-linear least-square factor $\phi_z : \mathrm{SE}(d) \times \mathrm{SE}(d) \to \mathbb{R}^{\dim z}$, $(\mathcal{X}^1_{i(z)}, \mathcal{X}^2_{i(z)}) \mapsto \left(\mathcal{X}^{1\,-1}_{i(z)} \oplus \mathcal{X}^2_{i(z)}\right) \ominus z$ weighted by its covariance C_z, where $i(z)$ denotes the indices of the two poses involved in measurement z. Therefore, the set of factors that i holds locally is $F_i = \{\phi_z \mid z \in Z_i\} \cup \{F(\mathcal{P}_{ii})\}$, where $F(\mathcal{P}_{ii})$ is the anchor prior.

If all measurements are pooled centrally, the global MAP estimate is obtained by optimizing:

$$X^\star = \operatorname{argmin}_{\{X_i\}_{i\in\mathcal{R}}} \sum_{i\in\mathcal{R}} \sum_{z\in Z_i} \left\|\phi_z(X)\right\|^2_{C_z}.$$

In a distributed setting, each robot can access only its local measurements F_i and a subset of variables X_i. Consequently, distributed solvers aim to achieve the same MAP optimum that would be obtained centrally while maintaining consistency in variables shared across different robots. Specifically, in the setup of multiple anchored pose graphs as in this work and [16], each robot must optimize its trajectory and estimate the anchor transformations. The global solution can then be obtained by aligning the optimized trajectories based on the anchor estimates of each robot. Unlike [16], our method optimize both the trajectories and anchors in a distributed manner through belief sharing.

Inspired by belief propagation, in CBS robot i encodes its information on any public state variable x by a Gaussian belief,

$$b_{\mathrm{x}} = \mathcal{N}(\mathrm{x},\, C_{\mathrm{x}}), \quad B_i = \{\, b_{\mathrm{x}} \mid \mathrm{x} \in X_i \}.$$

where $\mathrm{x}, C_{\mathrm{x}}$ are the mean and covariance of the belief, and B_i denotes the complete set of beliefs maintained by robot i. With local graphs anchored, b_{x} is the marginal belief of x in the local frame defined by $\mathcal{P}_{ii}$, which represents the probability $P_i(\mathrm{x})$. Our CBS framework shows that these marginal beliefs can be fused with local factors F_i inside standard MAP optimizers, producing highly accurate estimates while maintaining neighbor-only communication.

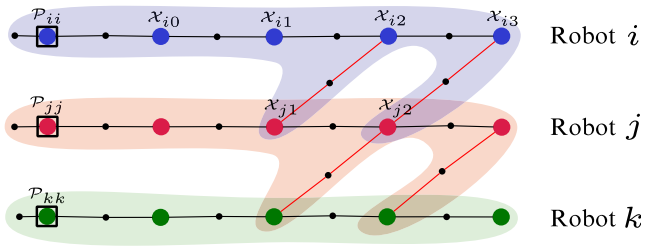
IV. Methodology

A. CBS Algorithm

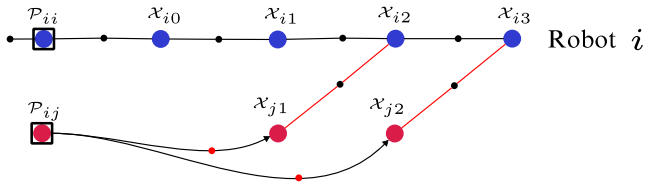
For solving the DPGO problem with belief sharing, the generation of beliefs and regulating the belief updates is the key to the convergence. As shown in Algorithm 1, the belief sharing process in CBS involves two stages; (1) the pose-belief sharing, (2) and the anchor-belief sharing stages. In the pose-belief sharing stage, robots share marginal Gaussian beliefs of their poses with their neighbors; recipients convert each belief as a factor within its own graph, binding the received pose to its estimate of the sender's anchor (Section IV-B), thereby maintaining individual robot's graphs as in Fig. 2 without the need for global initialization or exchange of raw measurements between the robots. The resulting factor graph is then optimized (Section IV-B), marginalized (Section IV-C) to update beliefs of the new linearization points by each robot. To improve convergence, before the updated beliefs are published to neighbors, they get damped using the Hellinger distance (Section IV-D), essentially turning the updates into a contraction. As soon as the residuals are stabilized when the absolute or relative change falls below the threshold $\varepsilon_{\mathcal{X}}$ for every robot, CBS switches to the anchor–belief stage, where robots exchange and refine anchor beliefs, in order to prove the consistency of frame alignment. Because the protocol follows the same neighbor-to-neighbor message pattern as in GBP, the communication and computation per robot remain bounded by the size of each robot's local graph, making CBS

Algorithm 1: Distributed Pose Graph Optimization via Contractive Belief Sharing (CBS).

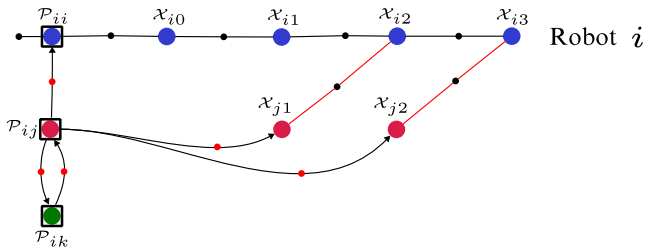
Input: Robots $\mathcal{R}$, decay factor $q \in (0,1)$; divergence limit $D_{\max}$; tolerances $\varepsilon_{\mathcal{X}}, \varepsilon_{\mathcal{P}}$. For each robot $i \in \mathcal{R}$: local factors F_i; neighbor set $\mathcal{E}_i$
Output: converged pose $\mathcal{X}_i$, estimates on all robots' anchors $\mathcal{P}_{ij}, j \in \mathcal{R}$ for each robot i.
Initialize $\mathcal{X}_i, \mathcal{P}_i, B_i$, by solving and marginalizing on F_i anchored by $\mathcal{P}_{ii}$.
▷**Stage 1 — Pose-belief sharing**
for $i \in \mathcal{R}$ **do** *(parallel or sequential)*
  repeat
    Receive neighbor pose beliefs $B_j^{\mathcal{X}}$ for $j \in \mathcal{E}_i$
    $F_{\mathcal{X}} \leftarrow \textsc{BeliefsToFactors}\big(B_j^{\mathcal{X}}\big)$ ▷Eq. (1)
    $\bar{X}_i \leftarrow \arg\min\big(F_i + F_{\mathcal{X}}\big)$ ▷Eq. (3)
    $\bar{B}_i \leftarrow \textsc{Marginalize}(\bar{X}_i, F_i)$ ▷Eq. (5)
    $\bar{B}_i^{\mathcal{X}} \leftarrow \{b_{\mathrm{x}} \in \bar{B}_i \mid \mathrm{x} \in \mathcal{X}_i\}$
    $B_i^{\mathcal{X}} \leftarrow \textsc{Damp}(\bar{B}_i^{\mathcal{X}})$ ▷Alg. 2
    $\textsc{Publish}(B_i^{\mathcal{X}})$ ▷Alg. 2
  until $\Delta\|F_i + F_{\mathcal{X}}\| < \varepsilon_{\mathcal{X}}$
end for
▷**Stage 2 — Anchor-belief sharing**
for $i \in \mathcal{R}$ **do** *(parallel or sequential)*
  repeat
    Receive neighbor anchor beliefs $B_j^{\mathcal{P}}$ for $j \in \mathcal{E}_i$
    $F_{\mathcal{P}} \leftarrow \textsc{BeliefsToFactors}\big(B_j^{\mathcal{P}}\big)$ ▷Eq. (2)
    $\bar{X}_i \leftarrow \arg\min\big(F_i + F_{\mathcal{X}} + F_{\mathcal{P}}\big)$ ▷Eq. (4)
    $\bar{B}_i \leftarrow \textsc{Marginalize}(\bar{X}_i, F_i + F_{\mathcal{X}})$ ▷Eq. (7)
    $\bar{B}_i^{\mathcal{P}} \leftarrow \{b_{\mathrm{x}} \in \bar{B}_i \mid \mathrm{x} \in \mathcal{P}_i\}$
    $B_i^{\mathcal{P}} \leftarrow \textsc{Damp}(\bar{B}_i^{\mathcal{P}})$ ▷Alg. 2
    $\textsc{Publish}(B_i^{\mathcal{P}})$ ▷Alg. 2
  until $\Delta\|F_i + F_{\mathcal{X}} + F_{\mathcal{P}}\| < \varepsilon_{\mathcal{P}}$
end for



(a) The global pose graph of three robots with inter-robot loop-closures (red lines). Each trajectory is anchored by a prior anchor pose $\mathcal{P}_{rr}$. Each robot's local graph is shaded by different colors.



(b) In pose-belief sharing stage, where pose beliefs get shared, each robot i receives each neighbor's beliefs on the shared variables, and converts them into factors (red dots) in its own graph, connecting the pose estimates and the estimation of the corresponding neighbors' anchors (here only j).



(c) In anchor-belief sharing stage, robots switch to share their anchor beliefs to their neighbors. Anchors from robots unobserved by i (e.g. k), via relayed beliefs from neighbors, ensuring all robots eventually hold anchors for every other robot and enabling consistent global alignment.

Fig. 2. The evolution of factor graph structures in CBS. (a) A global view of all robots' maps holding local and inter-robot factors. (b) In the pose-sharing stage, robots optimize and share pose beliefs with each other. (c) After all robots reach convergence on the poses within their local graphs, they optimize and share anchor beliefs. This structure of factor graphs generalizes to arbitrary numbers of robots, and all robots converge to accurate global solutions after the anchor-sharing stage is finished.

inherently scalable with the number of robots in the team. In this way, the proposed CBS algorithm preserves the scalability of GBP while embedding it in an MAP loop employing damping of updates, significantly improving its convergence.

B. Belief Factorization & Optimization

At each iteration, robot i receives a set of Gaussian beliefs B_j from every neighbor $j \in \mathcal{E}_i$. These beliefs are turned into *belief factors* ((1) and (2)) that tie the corresponding state variables in i's graph to its local estimate of the sender's anchor, using a robust lost $\rho(\cdot)$.

In the pose-belief stage, B_j contains only beliefs of pose variables, i.e., $B_j = \{b_{\mathrm{x}} | \mathrm{x} \in \mathcal{X}_j\}$. For every pose belief $b_{\mathrm{x}} \in B_j$, the residual is formed between robot i's local estimate of the received pose $\mathcal{X}_{i(\mathrm{x})}$ and the anchor estimate $\mathcal{P}_{ij}$:

$$F_{\mathcal{X}} = \sum_{j \in \mathcal{E}_i} \sum_{b_{\mathrm{x}} \in B_j} \rho\Big(\big\| (\mathcal{P}_{ij}^{-1} \oplus \mathcal{X}_{i(\mathrm{x})}) \ominus \mathrm{x} \big\|^2_{C_{\mathrm{x}}} \Big),$$

$$b_{\mathrm{x}} = \mathcal{N}(\mathrm{x}, C_{\mathrm{x}}) \in B_j, \quad \mathrm{x} \in \mathcal{X}_j, \tag{1}$$

where $i(\mathrm{x})$ gives the index of the variable x.

In the anchor–belief stage, after the pose beliefs have converged, the same construction is applied to the anchor beliefs, although relating the robot anchors, i.e. $B_j = \{b_{\mathcal{P}} | \mathcal{P} \in \mathcal{P}_j\}$. For example, if j's belief on its neighbor k's anchor $\mathcal{P}_{jk}, k \in \mathcal{E}_j$ is sent to i, it is converted to a factor connecting $\mathcal{P}_{ij}, \mathcal{P}_{ik}$, with the residual $\|(\mathcal{P}_{ij}^{-1} \oplus \mathcal{P}_{ik}) \ominus \mathcal{P}_{jk}\|^2_{C_{\mathcal{P}_{jk}}}$. If k has not been observed by i, $\mathcal{P}_{ik}$ is created as the message is received, and robots relay the anchor beliefs from their neighbors. Therefore, after messages are fully propagated through the team, each robot has the anchor estimations of all other robots. Formally, the conversion is as follows.

$$F_{\mathcal{P}} = \sum_{j \in \mathcal{R} \setminus i} \sum_{\substack{k \in \mathcal{R} \setminus i \\ k \neq j}} \rho\Big(\big\| (\mathcal{P}_{ij}^{-1} \oplus \mathcal{P}_{ik}) \ominus \mathcal{P}_{jk} \big\|^2_{C_{\mathcal{P}_{jk}}} \Big),$$

$$b_{\mathcal{P}} = \mathcal{N}(\mathcal{P}_{jk}, C_{\mathcal{P}_{jk}}) \in B_j, \quad \mathcal{P}_{jk} \in \mathcal{P}_j. \tag{2}$$

By anchoring beliefs to $\mathcal{P}_{ij}$, we effectively remove their local prior, transforming the received belief from the joint distribution $P_j(\mathrm{x})$ into the conditional form $P(\mathrm{x}|\mathcal{P}_{ij})$. Following the factorization of received beliefs, each robot i runs an MAP solver to

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

optimize its local estimates X_i on local factors F_i, the belief factors $F_{\mathcal{X}}$ and $F_{\mathcal{P}}$:

$$\text{pose-belief stage}: \quad \bar{X}_i = \arg\min_{X_i}\{F_i + F_{\mathcal{X}}\}, \tag{3}$$

$$\text{anchor-belief stage}: \quad \bar{X}_i = \arg\min_{X_i}\{F_i + F_{\mathcal{X}} + F_{\mathcal{P}}\}, \tag{4}$$

where in the anchor-belief stage, the converged pose beliefs are kept in the graph. We use the over-lined notation $\bar{X}_i$ to represent the raw optimized values before damping. Let $\Theta : B_j \mapsto \bar{X}_i$ be the MAP solution map for later convenience.

C. Local Marginalization

After optimizing the problems in (3) or (4), robot i computes the Jacobian and Hessian matrices at the linearization points $\bar{X}_i$, then use Schur complement to compute the local marginal covariances of poses or anchors. In the two stages of belief sharing, we linearize different factors. In particular, in the pose-belief stage, we linearize only local factors F_i, excluding pose and anchor belief factors,

$$J_{i\mathcal{X}} = F_i'(\bar{X}_i), \quad H_{i\mathcal{X}} = J_{i\mathcal{X}}^\top J_{i\mathcal{X}}, \tag{5}$$

$$C_{\bar{\mathrm{x}}} = \left(Schur_{\mathrm{x}}(H_{i\mathcal{X}})\right)^{-1}, \quad \mathrm{x} \in \mathcal{X}_i. \tag{6}$$

In the anchor-belief stage, we exclude anchor beliefs but keep the pose beliefs,

$$J_{i\mathcal{P}} = \begin{pmatrix} F_i'(\bar{X}_i) \\ F_{\mathcal{X}}'(\bar{X}_i) \end{pmatrix}, \quad H_{i\mathcal{P}} = J_{i\mathcal{P}}^\top J_{i\mathcal{P}}, \tag{7}$$

$$C_{\bar{\mathrm{x}}} = \left(Schur_{\mathrm{x}}(H_{i\mathcal{P}})\right)^{-1}, \quad \mathrm{x} \in \mathcal{P}_i. \tag{8}$$

By marginalization, each robot i forms new beliefs of variables as $\bar{B}_i = \{\bar{b}_{\mathrm{x}} = \mathcal{N}(\bar{\mathrm{x}}, C_{\bar{\mathrm{x}}}) \mid \bar{\mathrm{x}} \in \bar{X}_i\}$. Similarly, we use the over-lined symbols to represent the raw beliefs formed before applying damping. This marginalization process is denoted as the marginalization map $\mathcal{M} : \bar{X}_i \mapsto \bar{B}_i$ in the remainder of the paper.

D. Belief Damping With Hellinger Distance

To improve the convergence of CBS, we introduce a damping mechanism to control the step size based on the Hellinger distance [18], which provides a natural way to quantify belief divergence. Formally, let $\mathcal{U} : \bar{B}_i \mapsto B_i$ be the Hellinger-damping mapping, which takes raw beliefs $\bar{B}_i$ from marginalization, regulates the divergence between the previous iteration, and outputs the damped belief B_i. We compute the damping factor η_{x} such that the regulated belief iterates B_i with the decaying Hellinger distance using a decay factor q between consecutive updates; that is, each damped belief $b_{\mathrm{x}} \in B_i$ must satisfy,

$$\|b_{\mathrm{x}} - b'_{\mathrm{x}}\|_H \le qD'_{\mathrm{x}} = q\|b'_{\mathrm{x}} - b''_{\mathrm{x}}\|_H, \tag{9}$$

where $\|\cdot\|_H$ denotes the Hellinger distance between two beliefs, $b'_{\mathrm{x}} = \mathcal{N}(\mathrm{x}', C_{\mathrm{x}'})$ and b''_{x} correspond to the last and second to last iterations, respectively, and $\bar{b}_{\mathrm{x}} \in \bar{B}_i$ are the raw beliefs of the marginalization step. The Hellinger-damping process for each $b_{\mathrm{x}} \in B_i$ is defined as follows,

$$\tau_{\mathrm{x}} = \bar{\mathrm{x}} \ominus \mathrm{x}', \quad \tau_C = {}^{\mathrm{x}'}C_{\bar{\mathrm{x}}} - C_{\mathrm{x}'} \tag{10}$$

$$\mathrm{x} = \mathrm{x}' \oplus \eta_{\mathrm{x}}\tau_{\mathrm{x}}, \quad C_{\mathrm{x}} = C_{\mathrm{x}'} + \eta_{\mathrm{x}}^2\tau_C, \tag{11}$$

Algorithm 2: Hellinger-Damped Belief Update.

Require: Raw beliefs $\bar{B}_i = \{\bar{b}_{\mathrm{x}}\}$; previous iterates $B'_i = \{b'_{\mathrm{x}}\}$; previous Hellinger distance $\{D'_{\mathrm{x}}\}$; decay factor $q \in (0,1)$; divergence limit $D_{\max}$

$B_i \leftarrow \varnothing$
for all $\bar{b}_{\mathrm{x}} = \mathcal{N}(\bar{\mathrm{x}}, C_{\bar{\mathrm{x}}}) \in \bar{B}_i$ **do**
  Retrieve $b'_{\mathrm{x}} = \mathcal{N}(\mathrm{x}', C_{\mathrm{x}'})$ and D'_{x}
  $D_{\mathrm{x}} \leftarrow \|\bar{b}_{\mathrm{x}}, b'_{\mathrm{x}}\|_H$
  if $D_{\mathrm{x}} > D_{\max}$ **then**
    $B_i \leftarrow B_i \cup \{\bar{b}_{\mathrm{x}}\}$
    $D'_{\mathrm{x}} = D_{\text{raw}}$
  else
    $\tau_{\mathrm{x}} \leftarrow \bar{\mathrm{x}} \ominus \mathrm{x}', \tau_C \leftarrow {}^{\mathrm{x}'}C_{\bar{\mathrm{x}}} - C_{\mathrm{x}'}$ ▷Eq. (11)
    $\eta_{\mathrm{x}} \leftarrow \mathcal{H}(qD'_{\mathrm{x}})$ ▷Eq. (14)
    $\mathrm{x} \leftarrow \mathrm{x}' \oplus \eta_{\mathrm{x}}\tau_{\mathrm{x}}, C_{\mathrm{x}} \leftarrow C_{\mathrm{x}'} + \eta_{\mathrm{x}}^2\tau_C$ ▷Eq. (10)
    $b_{\mathrm{x}} \leftarrow \mathcal{N}(\mathrm{x}, C_{\mathrm{x}})$
    publish b_{x} to all neighbors
    $B_i \leftarrow B_i \cup \{b_x\}$
    $D'_{\mathrm{x}} = qD'_{\mathrm{x}}$
  end if
end for
$B'_i \leftarrow B_i$

therefore, the damping mapping $\mathcal{U}$ is,

$$\begin{aligned}\mathcal{U} : \bar{B}_i \mapsto B_i &:= \mathrm{Damp}(\bar{B}_i, B'_i) \\ &:= \{\, b_{\mathrm{x}} = \mathcal{N}(\mathrm{x}, C_{\mathrm{x}}) \mid \mathrm{x} \in X_i \,\}.\end{aligned} \tag{12}$$

As indicated in (9), η_{x} needs to be computed such that $\|b_{\mathrm{x}} - b'_{\mathrm{x}}\|_H \le qD'_{\mathrm{x}}$. The Hellinger distance between two Gaussian distributions $b_1 = \mathcal{N}(\mu_1, C_1), b_2 = \mathcal{N}(\mu_2, C_2)$ is formulated as follows:

$$\|b_1 - b_2\|_H = Hell(b_1, b_2) =$$

$$\sqrt{1 - \frac{\det C_1^{1/4} \det C_2^{1/4}}{\det \bar{C}^{1/2}} \cdot \exp\left\{-\frac{1}{8}\Delta\mu^\top \bar{C}^{-1}\Delta\mu\right\}}, \tag{13}$$

where $\bar{C} = (C_1 + C_2)/2, \Delta\mu = (\mu_1 - \mu_2)$. Substituting C_1 with $C_{\mathrm{x}'}$, C_2 with $C_{\mathrm{x}'} + \eta_{\mathrm{x}}^2\tau_C$, $\Delta\mu$ with $\eta_{\mathrm{x}}\tau_{\mathrm{x}}$, and setting $Hell(b_1, b_2) = qD'_{\mathrm{x}}$, we can get the approximate closed form of η_{x} with the auxiliary variables α, β, D_*, y, as

$$\alpha = -\frac{1}{8}\tau_{\mathrm{x}}^\top C_{\mathrm{x}'}^{-1}\tau_{\mathrm{x}}, \; \beta = -\frac{1}{16}\,\mathrm{tr}[(C_{\mathrm{x}'}^{-1}\tau_C)]^2,$$

$$D_* = \log\left(1 - (qD'_{\mathrm{x}})^2\right), \; y = \frac{-\alpha - \sqrt{\alpha^2 + 4\beta D_*}}{2\beta}.$$

$$\eta_{\mathrm{x}} = \mathcal{H}(qD'_{\mathrm{x}}) = \begin{cases} \sqrt{y}, & \text{if } \beta \neq 0 \\ \sqrt{D_*/\alpha}, & \text{if } \beta = 0 \end{cases} \tag{14}$$

If a belief update diverges by a Hellinger distance greater than $D_{\max}$ from the previous iteration, it is stored locally for the next iteration but not published to neighbors. Formally,

$$D_{\mathrm{x}} = \|\bar{b}_{\mathrm{x}} - b'_{\mathrm{x}}\|_H,$$

$$b_{\mathrm{x}} = \begin{cases} \bar{b}_{\mathrm{x}} & \text{if } \; D_{\mathrm{x}} \ge D_{\max}, \\ \mathrm{Damp}(\{\bar{b}_{\mathrm{x}}\}, \{b'_{\mathrm{x}}\}) & \text{if } \; D_{\mathrm{x}} < D_{\max}. \end{cases}$$

The process of this damping and publishing procedure is detailed in Algorithm 2. The decay factor q and the divergence limit $D_{\max}$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

are critical to the convergence of our method, which is discussed in the following section.

E. Convergence Analysis

We establish the global convergence of our method based on the composition-of-contractions theorem [19]. Specifically, if each mapping T_i is individually a contraction mapping, the composition of multiple such mappings is also contractive. To analyze the belief update process within a team of robots in this framework, we model the entire mapping of each robot as follows:

$$T_i = \Theta \circ \mathcal{M} \circ \mathcal{U}, \quad T_i : B_j \mapsto B_i, \tag{15}$$

where in particular,

$$\Theta : B_j \mapsto \bar{X}_i = \arg\min_{X_i} F(X_i, B_j),$$
$$\mathcal{M} : \bar{X}_i \mapsto \bar{B}_i = \text{Marginalization}(\bar{X}_i),$$
$$\mathcal{U} : \bar{B}_i \mapsto B_i = \text{Damp}(\bar{B}_i, B_i').$$

To prove the local Lipschitz conditions and contraction of the mappings, in Lemma 1 below, we first show that the map Θ is Lipschitz-continuous in the input beliefs B_j, if B_j stays in a small local region defined by Hellinger distance. Then Lemma 2 shows that the complete map T_i also has local Lipschitz continuity by chaining the Lipschitz condition of 3 individual maps. Finally, Lemma 1 proves by picking small values of η_{x} and $D_{\max}$, T_i achieves local Banach contraction.

Lemma 1: Let us fix a reference input belief set $B_j^\star$ with its local MAP minimizer $X_i^\star$, i.e. $\nabla_X F(X_i^\star, B_j^\star) = 0$. Assuming that in a neighborhood of $(X_i^\star, B_j^\star)$ defined by γ_X, γ_B, the residual is small enough, such that the exact Hessian $H_{XX}(X_i, B_j) = \nabla_X^2 F(X_i, B_j)$ remains uniformly positive definite (PD):

$$\exists\, a > 0, \text{ such that } \quad H_{XX}(X_i, B_j) \succeq aI$$
$$\forall (X_i, B_j) \text{ with } \|X_i - X_i^\star\| \le \gamma_X,\ \|B_j - B_j^\star\|_H \le \gamma_B.$$

Then, shrinking γ_B such that the supremum M below is finite, the MAP solution map $\Theta(B_j) := \arg\min_{X_i} F(X_i, B_j)$ is locally Lipschitz-continuous on the belief neighborhood $\mathcal{B}_b := \{B_j : \|B_j - B_j^*\|_H \le \gamma_B\}$:

$$\|\Theta(B_j^1) - \Theta(B_j^2)\| \le L_\Theta \|B_j^1 - B_j^2\|_H, \quad L_\Theta = \frac{M}{a}, \tag{16}$$

$$M = \sup_{\substack{\|X_i - X_i^\star\| \le \gamma_X \\ \|B_j - B_j^\star\|_H \le \gamma_B}} \|\partial_B \nabla_X F(X_i, B_j)\| < \infty.$$

Proof: Let $g(X_i, B_j) := \nabla_X F(X_i, B_j) \in \mathbb{R}^{n_X}$. Because F is C^2 in a neighborhood of $(X_i^\star, B_j^\star)$, the mapping g is C^1. At the reference point $g(X_i^\star, B_j^\star) = 0$ and its X–Jacobian is the exact Hessian $H_{XX}^\star = \nabla_X^2 F(X_i^\star, B_j^\star) \succ 0$ by assumption. Hence $|H_{XX}^\star| \neq 0$, and the *Implicit-Function Theorem* yields,

$$\exists\, U_B \subset \mathbb{R}^{n_B},\ U_X \subset \mathbb{R}^{n_X} :$$
$$\Theta \in C^1(U_B, U_X), g\big(\Theta(B_j), B_j\big) = 0.$$

Choose $\gamma_B > 0$ so $\mathcal{B}_b$ lies inside U_B. Continuity of Θ implies bounded γ_X. If $\mathcal{B}_{\text{x}} := \{X_i : \|X_i - X_i^\star\| \le \gamma_X\}$ and the product set is defined as $K := \mathcal{B}_{\text{x}} \times \mathcal{B}_b$, then K is compact. By assumption then, the exact Hessian remains uniformly PD on K:

$$H_{XX}(X_i, B_j) \succeq aI \quad \forall (X_i, B_j) \in K. \tag{17}$$

The continuity of $\partial_B g = \partial_B \nabla_X F$ on the compact K implies

$$M := \sup_{(X_i, B_j) \in K} \big\|\partial_B \nabla_X F(X_i, B_j)\big\| < \infty. \tag{18}$$

Differentiating the stationarity condition $g(\Theta(B_j), B_j) = 0$ with respect to B_j:

$$\partial_X g \frac{d\Theta}{dB_j} + \partial_B g = 0 \implies$$
$$\frac{d\Theta}{dB_j} = -\big[\partial_X g\big]^{-1} \partial_B g = -H_{XX}^{-1}\, \partial_B \nabla_X F. \tag{19}$$

Using (17) and (18), we get an upper bound for (19):

$$\Big\|\frac{d\Theta}{dB_j}\Big\| \le \|H_{XX}^{-1}\|\, \big\|\partial_B \nabla_X F\big\| \le \frac{1}{a} M, \quad \forall (X_i, B_j) \in K.$$

Integrating between B_j^1 and B_j^2 on $\mathcal{B}_b$, for any $B_j^1, B_j^2 \in \mathcal{B}_b$, gives:

$$\|\Theta(B_j^1) - \Theta(B_j^2)\| = \|\bar{X}_i^1 - \bar{X}_i^2\|$$
$$\le \sup_{B_b} \Big\|\frac{d\Theta}{dB_j}\Big\| \|B_j^1 - B_j^2\|_H \le \frac{M}{a} \|B_j^1 - B_j^2\|_H.$$

Setting $L_\Theta := M/a$ completes the proof. ■

After we establish the local Lipschitz condition of Θ, we show the the Lipschitz conditions of $\mathcal{M}$,$\mathcal{U}$ and the robot mapping T_i in the following lemma.

Lemma 2: Under the conditions of Lemma 1, the three mappings $\Theta, \mathcal{M}, \mathcal{U}$, are all locally Lipschitz. More precisely, for inputs $B_j^1, B_j^2 \in \mathcal{B}_b$,

$$\|\Theta(B_j^1) - \Theta(B_j^2)\| \le L_\Theta \|B_j^1 - B_j^2\|_H, \tag{20}$$
$$\|\mathcal{M}(\bar{X}_i^1) - \mathcal{M}(\bar{X}_i^2)\|_H \le L_{\mathcal{M}} \|\bar{X}_i^1 - \bar{X}_i^2\|, \tag{21}$$
$$\|\mathcal{U}(\bar{B}_i^1) - \mathcal{U}(\bar{B}_i^2)\|_H \le L_{\mathcal{U}1} \|B_i'^1 - B_i'^2\|_H + L_{\mathcal{U}2} \|\bar{B}_i^1 - \bar{B}_i^2\|_H. \tag{22}$$

Thus, the complete mapping T_i satisfies the joint Lipschitz condition on B_i' and B_j, under the same conditions as in Lemma 1.

Proof: For the solution mapping Θ, the Lipschitz constant L_Θ is exactly the M/a derived in Lemma 1. For the marginalization mapping $\mathcal{M}$, for each $\bar{X}_i$, inside the convergence ball $\mathcal{B}_{GN}$ we have, $\|J(\bar{X}_i^1) - J(\bar{X}_i^2)\| \le L_J \|\bar{X}_i^1 - \bar{X}_i^2\|, J(\bar{X}_j) \le M_J$, and $H(\bar{X}_i) \succeq \lambda_{\min} I$, [20]. Due to these Lipschitz and boundedness conditions, within the neighborhood $\mathcal{B}_{\text{x}} \subseteq \mathcal{B}_{GN}$, the Gauss–Newton Hessian $H(\bar{X}_i) = J(\bar{X}_i)^\top J(\bar{X}_i)$ is itself L_H–Lipschitz with, $\|H(\bar{X}_i^1) - H(\bar{X}_i^2)\| \le L_H \|\bar{X}_i^1 - \bar{X}_i^2\|$. Thus the updated beliefs $\bar{B}_i$, composed by $\bar{\text{x}} \in X_i$ and $C_{\bar{\text{x}}} = \text{Schur}_{\bar{\text{x}}}(H(\bar{X}_i))^{-1}$, preserves the Lipschitz condition:

$$\|\mathcal{M}(\bar{X}_i^1) - \mathcal{M}(\bar{X}_i^2)\|_H =$$
$$\|\bar{B}_j^1 - \bar{B}_j^2\|_H \le L_{\mathcal{M}} \|\bar{X}_i^1 - \bar{X}_i^2\|,$$

The damping mapping $\mathcal{U}$ is effectively a mixing combination. In the metric space formed by Gaussian beliefs and Hellinger

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

distance, this mixing is Lipschitz-continuous with constants in (22) as,

$$L_{\mathcal{U}1} = 1 - \min_{\mathrm{x}\in X_i} \eta_{\mathrm{x}}, \quad L_{\mathcal{U}2} = \eta_{\max} = \max_{\mathrm{x}\in X_i} \eta_{\mathrm{x}}.$$

The local Lipschitz conditions above can be chained to show that T_i is locally jointly Lipschitz-continuous, i.e.,

$$\begin{aligned} &\|T_i(B_j^1; B_i'^1) - T_i(B_j^2; B_i'^2)\|_H = \\ &\|B_i^1 - B_i^2\|_H \le L_{\mathcal{U}1}\|B_i'^1 - B_i'^2\|_H + L_{\mathcal{U}2}\|B_j^1 - B_j^2\|_H \\ &\quad \le L_{\mathcal{U}1}\|B_i'^1 - B_i'^2\|_H + L_{\mathcal{U}2}L_{\mathcal{M}}L_{\Theta}\|B_j^1 - B_j^2\|_H \end{aligned} \tag{23}$$

■

From the Lipschitz condition of T_i, we can finally establish the local Banach contraction condition as in the following theorem.

Theorem 1: Let the assumptions of Lemma 1 hold, such that the constants L_Θ and $L_\mathcal{M}$ are valid inside $\mathcal{B}_b$. Let us assume that the damping factor η_{x} and the divergence limit $D_{\max}$ satisfy, $\eta_{\max} := \max_{b\in B_i} \eta_{\mathrm{x}} < \frac{1}{L_\mathcal{M} L_\Theta}$, and $D_{\max} \le (1 - q_T)\,\gamma_B$. Then, for every fixed B_i', the mapping T_i is a contraction on B_b, *i.e.*,

$$\|T_i(B_j^1; B_i') - T_i(B_j^2; B_i')\|_H \le q_T \|B_j^1 - B_j^2\|_H, q_T \in [0, 1),$$

Hence there exists a unique fixed point $(B_i^\star, B_j^*) \in B_b$ satisfying $T_i(B_j^\star; B_i^\star) = B_i^\star$, and the successive outputs generated by the CBS iteration obey, $\|B_i^k - B_i^\star\|_H \le q_T^k \|B_j^0 - B_j^\star\|_H$.

Proof: With B_i' fixed in each iteration, the damping term that multiplies $\|B_i' - B_i'\|_H$ vanishes, so

$$\begin{aligned} &\|T_i(B_j^1; B_i') - T_i(B_j^2; B_i')\|_H \\ &\quad \le \eta_{\max} L_\mathcal{M} L_\Theta \|B_j^1 - B_j^2\|_H = q_T \|B_j^1 - B_j^2\|_H, \end{aligned}$$

where $q_T \in [0, 1)$ by choice of $\eta_{\max}$. Because the first update satisfies $\|B_j^1 - B_j^0\|_H \le D_{\max}$, the bounded divergence (9) ensures iterates remain in $\mathcal{B}_b$, and Banach's theorem yields a unique fixed point.

■

Lemma 1 proves each robot's mapping T_i is a contraction mapping under certain conditions. The conclusion of the composition-of-contractions theorem follows immediately that the DPGO problem as a composition of robots' mappings $\{T_i\}_{i\in\mathcal{R}}$ is also a contraction and converges to a global unique solution.

Remark 1 (Choose the decay factor q instead of η_{x}): Fixing a small η_{x} offers no direct control over update divergence and may violate the $D_{\max}$ bound. Since η_{x} from (14) decreases monotonically with smaller q, adjusting q controls both the divergence qD_{x}' and the step size η_{x} per iteration, ensuring Lemma 1 holds. Thus, q is used as the convergence parameter in our experiments.

Remark 2 (Practical meaning of convergence): After k iterations, two observations hold. First, each robot has received the marginalized posteriors of shared poses from all neighbors and anchor beliefs of all robots. Combined with its local factors, these form an approximate joint posterior $\tilde{P}(\{\mathcal{X}_i\}, \{\mathcal{P}_i\} \mid \{Z_i\})$ matching the centralized posterior up to marginalization and linearization errors. The fixed point of CBS is thus the MAP estimate of $\tilde{P}$, coinciding with the centralized solution when beliefs are consistent.

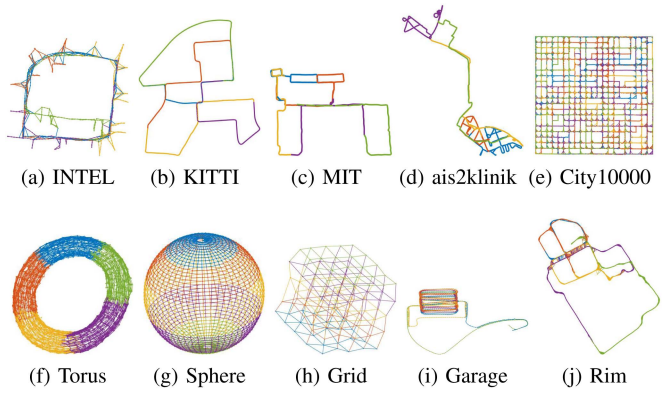


Fig. 3. Distributed pose graphs optimized with the CBS on 10 benchmarking datasets.

Second, belief-factor residuals decay geometrically: $\|r_{ij}^k - \bar{r}_{ij}\| = \mathcal{O}(q_T^k)$, i.e., at the same rate as belief updates. Hence, even without strict consensus, shared variables become numerically close, providing near-MAP solutions in few iterations.

Remark 3 (Relationship to GBP and DDF-SAM2): Although CBS is inspired by recent advances in GBP methods [8], [9], the resemblance stops with the use of beliefs as a representation of messages. Conceptually, CBS is closer to marginal-sharing strategies such as DDF-SAM2, but it differs in two crucial ways: estimation and sharing of the beliefs of anchor nodes improves the consistency of solutions, and also the contractive step size mechanism is introduced to ensure convergence.

V. Experiments

We evaluate CBS through extensive experiments on benchmarking datasets, assessing trajectory accuracy, communication load, parameter sensitivity, and scalability against state-of-the-art distributed optimization methods for DPGO.

A. Trajectory Accuracy and Communication Burden

We compare CBS with ASAPP [6], DGS [7], and MESA [3] on the datasets[1] from [21], [22]. Each dataset is sequentially partitioned into five portions to create a five-robot setup for distributed estimation (as visible in Fig. 3). We adopt the experimental setup of MESA[2], and test against MESA's Geodesic variant with $\beta_{(i,j)}^0 = 2$ and $\alpha = 1.05$ (as in their paper). For ASAPP, we set the RGD step size to 0.01, and 100 RGD steps per iteration. While for DGS, we use the default parameters[3]. For CBS, we set the decay factor $q = 0.1$, the divergence limit $D_{\max} = 0.1$, and the residual thresholds $\varepsilon_\mathcal{X} = 1e^{-3}$. The Huber loss function is applied on (1) and (2), while the chordal initialization and Gauss-Newton optimizer are used to solve (3) and (4). We report the worst-case trajectory errors of CBS by aligning trajectories by the anchor estimates from different robots. We let all methods run $600 * |\mathcal{R}|$ iterations on all robots in total. The Absolute Position Errors (APE) and Absolute Rotation Errors (ARE) as recorded in Table I, are computed against the

[1] https://lucacarlone.mit.edu/datasets
[2] https://github.com/rpl-cmu/mesa.git
[3] https://github.com/CogRob/distributed-mapper

TABLE I
ABSOLUTE POSITION (T) AND ROTATION (R) ERRORS RECORDED TOGETHER WITH THE NUMBERS OF COMMUNICATION ROUNDS REQUIRED BY OUR METHOD AND PRIOR WORKS ON PGO BENCHMARKING DATASETS. EACH DATASET PARTITIONED SEQUENTIALLY INTO FIVE ROBOTS. THE BEST RESULTS FOR EACH DATASET ARE HIGHLIGHTED IN BOLD, WHILE THE SECOND-BEST ARE UNDERLINED.

Dim.	Dataset	ASAPP			DGS			MESA			CBS		
		t (m)	R (°)	comm.	t (m)	R (°)	comm.	t (m)	R (°)	comm.	t (m)	R (°)	comm.
2D	CSAIL	0.16	0.37	4908	0.13	0.45	240	**0.11**	**0.42**	870	**0.11**	0.54	**205**
	KITTI	1.50	0.86	4891	1.86	1.40	1600	4.25	2.48	360	**1.44**	**0.47**	**220**
	M3500	0.45	2.54	5188	0.47	3.00	12080	0.50	3.02	650	**0.34**	**0.82**	**185**
	MIT	8.08	10.26	4047	8.26	11.40	5430	36.87	41.25	380	**2.58**	**2.11**	**230**
3D	Sphere	0.20	1.88	1921	0.22	1.96	2080	**0.06**	**0.21**	430	0.12	0.43	**225**
	Grid	0.05	2.17	382	0.23	23.27	240	**0.002**	**0.15**	330	0.09	5.30	**230**
	Torus	0.015	0.20	2835	0.19	7.28	850	**0.002**	**0.04**	880	0.07	1.82	**240**
	Garage	1.17	1.70	2349	1.19	1.77	180	1.21	1.71	170	**0.88**	**1.09**	**150**

centralized solution computed by Levenberg–Marquardt optimizer with chordal initialization.

Communication is considered the biggest bottleneck in DPGO problems. In this experiment, we also record the number of communication rounds used by each method, as defined in [3], i.e. a pair of robots transmitting data bidirectionally is considered as one round of communication. To account for different convergence criteria, we report the rounds of communications used when each method reaches within 1% of its final errors.

Table I highlights the trade-off between accuracy and communication rounds for all algorithms. ASAPP, as a certifiably correct method, remains accurate under severe measurement noise, but incurs the largest communication cost. DGS, on the other hand, requires less communication, but provides less accurate solutions, due to its chordal relaxation. MESA converges faster than ASAPP and DGS, however, its accuracy drops on some noisy sequences. In constrast, CBS converges with same q and $D_{\max}$ values on all datasets, using fewer than 250 communication rounds, meanwhile consistently achieving high accuracy, especially on noisy datasets such as KITTI and MIT. More examples of optimized trajectories are visualized in Fig. 3.

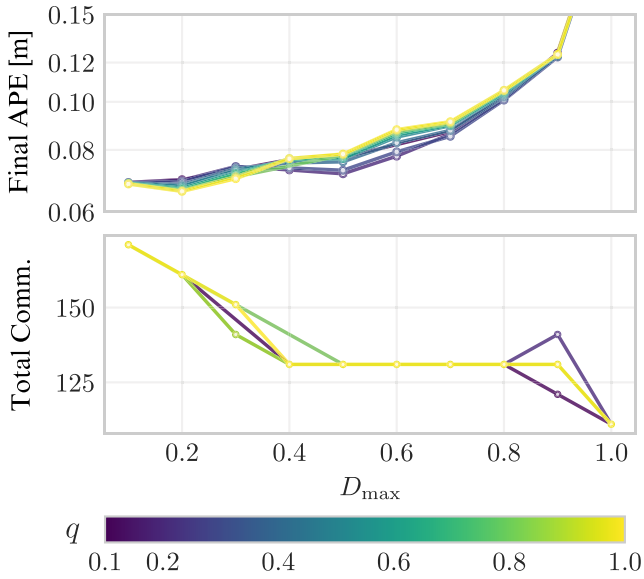


Fig. 4. Final APE and the rounds of communications until convergence of CBS with different values of q and $D_{\max}$.

B. Parameter Sensitivity

We also analyse the effect of q and $D_{\max}$ on the Torus dataset, to provide some insights on the parameter sensitivity in a less noisy 3D pose graph. Fig. 4 reports the final APEs and rounds of communication until the convergence of CBS for a grid of $(q, D_{\max})$ values. As $D_{\max}$ increases, CBS converges in fewer rounds but at the cost of higher errors. Conversely, achieving lower errors with larger $D_{\max}$ requires smaller q, while larger q can be tolerated when $D_{\max}$ is small. These trends are consistent with our theoretical analysis, confirming that the Hellinger-damping mechanism and contraction conditions jointly regulate convergence and stabilize the optimization-then-belief-sharing process.

C. Scalability With Increasing Number of Robots

To assess scalability, we generate synthetic 3D grid-world datasets with 5–15 robots, each following a 100-pose trajectory. Odometry and loop closure measurements have Gaussian noise $\sigma = 0.05$ m and 2°; intra-robot loop closures are added with probability 0.1, and inter-robot loops with probability 0.3 when poses are nearby. Methods are configured as in Section V-A, except ASAPP uses an RGD step size of 0.001 for stability.

In Fig. 5 we report per-robot resources required by each method to reach the same final errors of CBS: communication rounds, optimizer iterations, and computation time, all normalized by the number of robots. If a method does not achieve CBS's accuracy within the iteration budget, it is marked with 'x' and its resource usage until reaching 1% of its final error is reported.

As shown in Fig. 5, the trade-offs are consistent with those observed in Section V-A. ASAPP in general consumes the most resources across all metrics. DGS achieves better scalability among baseline works, but it produces less accurate results. MESA is able to produce the highest accuracy, and is more scalable than ASAPP. In contrast, CBS demonstrates two key advantages: firstly, all three metrics increase only modestly as the team size grows; and secondly, it achieves consistently significantly lower resource usage than the other methods across all metrics. These results highlight the efficiency of our approach: CBS retains the fully local communication pattern of belief propagation while maintaining fast convergence, making it well-suited for large-scale multi-robot applications.

VI. CONCLUSION

We revisited Distributed Pose Graph Optimization (DPGO) from the perspective of belief-sharing frameworks, and proposed *Contractive Belief Sharing* (CBS), a hybrid method that combines the communication efficiency of belief propagation with

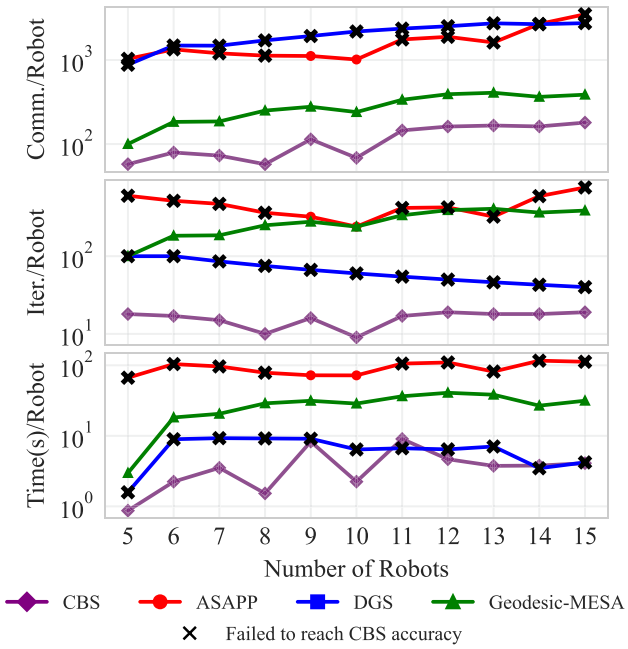


Fig. 5. The normalized numbers of normalized communication rounds, optimization iterations and computation time, used by each method until reaching CBS's final errors with increasing numbers of robots.

the robustness of MAP optimization. A key innovation of CBS is its Hellinger-distance-based damping rule, which enforces contraction in the belief updates and enables fast, stable convergence even in loopy and noisy graphs. Through extensive evaluation, CBS achieved state-of-the-art accuracy and scalability, while requiring substantially less computational and communication resources.

Future works can focus on solving two major limitations; First, marginalization can become computationally expensive in very dense graphs. Second, the current stage-switching mechanism relies on a global convergence check, which constrains its use in fully asynchronous settings. Addressing these challenges will further extend the applicability of CBS to real-time, large-scale multi-robot deployments in the field.

REFERENCES

[1] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. IEEE Conf. Decis. Control*, 2012, pp. 5445–5450.
[2] O. Shorinwa, T. Halsted, J. Yu, and M. Schwager, "Distributed optimization methods for multi-robot systems: Part 2—A survey," *IEEE Robot. & Automat. Mag.*, vol. 31 no. 3, pp. 154–169, 2024.
[3] D. McGann, K. Lassak, and M. Kaess, "Asynchronous distributed smoothing and mapping via on-manifold consensus ADMM," in *IEEE Int. Conf. Robot. Automat.*, 2024, pp. 4577–4583.
[4] D. McGann and M. Kaess, "iMESA: Incremental distributed optimization for collaborative simultaneous localization and mapping," in *Proc. Robot., Sci. Syst.*, Delft, Netherlands, Jul. 2024.
[5] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 2137–2156, Jun. 2021.
[6] Y. Tian, A. Koppel, A. S. Bedi, and J. P. How, "Asynchronous and parallel distributed pose graph optimization," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5819–5826, Oct. 2020.
[7] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *Int. J. Robot. Res.*, vol. 36, no. 12, pp. 1286–1311, 2017.
[8] A. J. Davison and J. Ortiz, "FutureMapping 2: Gaussian belief propagation for spatial AI," 2019, *arXiv:1910.14139*.
[9] R. Murai, J. Ortiz, S. Saeedi, P. H. J. Kelly, and A. J. Davison, "A robot web for distributed many-device localization," *IEEE Trans. Robot.*, vol. 40, pp. 121–138, 2024.
[10] A. Ranganathan, M. Kaess, and F. Dellaert, "Loopy SAM," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 2191–2196.
[11] Y. Weiss and W. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," in *Proc. Adv. Neural Inf. Process. Syst.*, S. T. Solla Leen and K. Müller, Eds., vol. 12, 1999, pp. 673–679.
[12] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *J. Mach. Learn. Res.*, vol. 7, pp. 2031–2064, 2006.
[13] A. Cunningham, M. Paluri, and F. Dellaert, "DDF-SAM: Fully distributed SLAM using constrained factor graphs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 3025–3030.
[14] A. Cunningham, V. Indelman, and F. Dellaert, "DDF-SAM 2.0: Consistent distributed smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 5220–5227.
[15] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 545–565, Jun. 2016.
[16] B. Kim et al., "Multiple relative pose graphs for robust cooperative mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 3185–3192.
[17] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," 2018, *arXiv:1812.01537*.
[18] L. M. Le Cam and G. L. Yang, *Asymptotics in Statistics: Some Basic Concepts*. Berlin, Germany: Springer, 2000.
[19] L. Lorentzen, "Compositions of contractions," *J. Comput. Appl. Math.*, vol. 32, no. 1-2, pp. 169–178, 1990.
[20] L. Carlone, "A convergence analysis for pose graph optimization via Gauss-Newton methods," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 965–972.
[21] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 475–492, Apr. 2014.
[22] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, "A fast and accurate approximation for planar pose graph optimization," *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 965–987, 2014.