Majorization Minimization Methods for Distributed Pose Graph Optimization

Taosha Fan and Todd D. Murphey

***Abstract*—We consider the problem of distributed pose graph optimization (PGO) that has important applications in multi-robot simultaneous localization and mapping (SLAM). We propose the majorization minimization (MM) method for distributed PGO (MM–PGO) that applies to a broad class of robust loss kernels. The MM–PGO method is guaranteed to converge to first-order critical points under mild conditions. Furthermore, noting that the MM–PGO method is reminiscent of proximal methods, we leverage Nesterov's method and adopt adaptive restarts to accelerate convergence. The resulting accelerated MM methods for distributed PGO—both with a master node in the network (AMM–PGO*) and without (AMM–PGO$^\#$)—have faster convergence in contrast to the MM–PGO method without sacrificing theoretical guarantees. In particular, the AMM–PGO$^\#$ method, which needs no master node and is fully decentralized, features a novel adaptive restart scheme and has a rate of convergence comparable to that of the AMM–PGO* method using a master node to aggregate information from all the nodes. The efficacy of this work is validated through extensive applications to 2D and 3D SLAM benchmark datasets and comprehensive comparisons against existing state-of-the-art methods, indicating that our MM methods converge faster and result in better solutions to distributed PGO. The code is available at https://github.com/MurpheyLab/DPGO.**

I. Introduction

Pose graph optimization (PGO) is a nonlinear and non-convex optimization problem estimating unknown poses from noisy relative pose measurements. PGO associates each pose with a vertex and each relative pose measurement with an edge such that the optimization problem is well represented through a graph. PGO has important applications in a number of areas, including but not limited to robotics [1]–[3], autonomous driving [4], and computational biology [5], [6]. Recent advances [7]–[16] suggest that PGO can be well solved using iterative optimization. Nevertheless, the aforementioned techniques [7]–[16] are difficult to distribute across a network due to communication and computational limitations, and are only applicable to small- and medium-sized problems with at most tens of thousands poses. In addition, their centralized pipelines are equivalent to using a master node to aggregate information from the entire network, and thus, fail to meet privacy requirements one may wish to impose [17], [18].

In multi-robot simultaneous localization and mapping (SLAM) [19]–[28], each robot estimates not only its own poses but those of the others as well to build an environment map. Even though such a problem can be solved by PGO, communication between robots is restricted and multi-robot SLAM has more unknown poses than single-robot SLAM. Thus, instead of using centralized PGO [7]–[16], it is more reasonable to formulate this large-sized estimation problem involving multiple robots as distributed PGO—each robot in multi-robot SLAM is represented as a node and two nodes (robots) are said to be neighbors if there exists a noisy relative pose measurement between them (a more detailed description of distributed PGO can be found in Section IV). In most cases, it is assumed that inter-node communication only occurs between neighboring nodes and most of these iterative optimization methods are infeasible due to the expensive communication cost of solving linear system and performing line search [7]–[16], which renders distributed PGO more challenging than centralized PGO.

In this paper, we propose majorization minimization (MM) methods [29], [30] for distributed PGO. As the name would suggest, MM methods have two steps. First, in the majorization step, we construct a surrogate function that majorizes the objective function, i.e., the surrogate function is greater to the objective function except for the current iterate where both of them attain the same value. Then, in the minimization step, we minimize the surrogate function instead of the original objective function to improve the iterate. MM methods remain difficult, albeit straightforward, in practical use, e.g., the surrogate function, whose construction and minimization can not be more difficult than solving the optimization problem itself, is usually unknown, and MM methods might fail to converge and suffer from slow convergence. Thus, the implementation of MM methods on large-scale, complicated and nonconvex optimization problems like distributed PGO is nontrivial, and inter-node communication requirements impose extra restrictions making it more so. All of these issues are addressed both theoretically and empirically in the rest of this paper.

This paper extends the preliminary results in [31], [32], where we developed MM methods for centralized and distributed PGO that are guaranteed to converge to first-order critical points. In [31], [32], we also introduced and elaborated on the use of Nesterov's method [33], [34] and adaptive restart [35] for the first time to accelerate the convergence of PGO. Beyond the initial results in [31], [32], this paper presents completely redesigned MM methods for distributed PGO and provides more comprehensive theoretical and empirical results. In particular, our MM methods in this paper are capable of handling a broad class of robust loss kernels, no longer require each iteration to attain a local optimal solution to the surrogate function for the convergence guarantees, and adopt a novel adaptive restart scheme for distributed PGO without a master node to make full use of Nesterov's acceleration.

T. Fan is with Meta AI, Pittsburgh, PA 15213, USA and T. D. Murphey is with the Department of Mechanical Engineering, Northwestern University, Evanston, IL 60201, USA. E-mail: taoshaf@meta.com, t-murphey@northwestern.edu

This material is partially based upon work supported by the National Science Foundation under awards 1662233 and 1837515.

The authors thank Yulun Tian for sharing the code of Riemannian block coordinate descent method (RBCD) for distributed PGO.

In summary, the contributions of this paper are the follows:

1) We derive a class of surrogate functions that suit well with MM methods for distributed PGO. These surrogate functions apply to a broad class of robust loss kernels in robotics and computer vision.
2) We develop MM methods for distributed PGO that are guaranteed to converge to first-order critical points under mild conditions. Our MM methods for distributed PGO implement a novel update rule such that each iteration does not have to minimize the surrogate function to a local optimal solution.
3) We leverage Nesterov's methods and adaptive restart to accelerate MM methods for distributed PGO and achieve significant improvement in convergence without any compromise of theoretical guarantees.
4) We present a decentralized adaptive restart scheme to make full use of Nesterov's acceleration such that accelerated MM methods for distributed PGO without a master node are almost as fast as those requiring a master node.

The rest of this paper is organized as follows. Section II reviews the state-of-the-art methods for distributed PGO. Section III introduces mathematical notation and preliminaries that are used in this paper. Section IV formulates the problem of distributed PGO. Sections V and VI present surrogate functions for individual loss terms and the overall distributed PGO, respectively, which are fundamental to our MM methods. Sections VII to IX present unaccelerated and accelerated MM methods for distributed PGO that are guaranteed to converge to first-order critical points, which are the major contributions of this paper. Section X implements our MM methods for distributed PGO on a number of simulated and real-world SLAM datasets and make extensive comparisons against existing state-of-the-art methods [36], [37]. Section XI concludes this paper and discusses future work.

II. Related Work

In the last decade, multi-robot SLAM has been becoming increasingly popular, which promotes the development of distributed PGO [36]–[39].

Choudhary et al. [36] present a two-stage algorithm that implements either Jacobi Over-Relaxation or Successive Over-Relaxation as distributed linear system solvers. Similar to centralized methods, [36] first evaluates the chordal initialization [40] and then improves the initial guess with a single Gauss-Newton step. However, one step of Gauss-Newton method in most cases can not lead to sufficient convergence for distributed PGO. In addition, no line search is performed in [36] due to the communication limitation, and thus, the behaviors of the single Gauss-Newton step is totally unpredictable and might result in bad solutions.

Tian et al. [37] present the distributed certifiably correct PGO using Riemannian block coordinate descent method, which is later generalized to asynchronous and parallel distributed PGO [41]. Specially, their method makes use of Riemannian staircase optimization to solve the semidefinite relaxation of distributed PGO and is guaranteed to converge to global optimal solutions under moderate measurement noise. Following our previous works [31], [32], they implement Nesterov's method for acceleration as well. Contrary to our MM methods, a major drawback of [37] is that their method has to precompute red-black coloring assignment for block aggregation and keep part of the blocks in idle for estimate updates. In addition, although several strategies for block selection (e.g., greedy/importance sampling) and Nesterov's acceleration (e.g., adaptive/fixed restarts) are adopted in [37] to improve the convergence, most of them are either inapplicable without a master node or at the sacrifice of computational efficiency and theoretical guarantees. In contrast, our MM methods are much faster (see Section X) but have no such restrictions for acceleration. More recently, Tian et al. further apply Riemannian block coordinate descent method to distributed PGO with robust loss kernels [28]. However, they solve robust distributed PGO by trivially updating the weights using graduated nonconvexity [42] and no formal proofs of convergence are provided. Again, this is in contrast to the work presented here that has provable convergence to first-order critical points for a broad class of robust loss kernels.

Tron and Vidal [38] present a consensus-based method for distributed PGO using Riemannian gradient. The authors derive a condition for convergence guarantees related with the stepsize of the method and the degree of the pose graph. Nonetheless, their method estimates rotation and translation separately, fails to handle robust loss kernels, and needs extra computation to find the convergence-guaranteed stepsize.

Cristofalo et al. [39] present a novel distributed PGO method using Lyapunov theory and multi-agent consensus. Their method is guaranteed to converge if the pose graph has certain topological structures. However, [39] updates rotations without exploiting the translational measurements and only applies to pairwise consistent PGO with nonrobust loss kernels.

In comparison to these aforementioned techniques, our MM methods have the mildest conditions (not requiring any specific pose graph structures, any extra computation for preprocessing, any master nodes for information aggregation, etc.) to converge to first-order critical points, apply to a broad class of robust loss kernels in robotics and computer vision, and manage to implement decentralized acceleration with convergence guarantees. Most importantly, as is shown in Section X, our MM methods outperform existing state-of-the-art methods in terms of both efficiency and accuracy on a variety of SLAM benchmark datasets.

III. Notation[1]

Miscellaneous Sets. $\mathbb{R}$ denotes the sets of real numbers; $\mathbb{R}^+$ denotes the sets of nonnegative real numbers; $\mathbb{R}^{m\times n}$ and $\mathbb{R}^n$ denote the sets of $m \times n$ matrices and $n \times 1$ vectors, respectively. $SO(d)$ denotes the set of special orthogonal groups and $SE(d)$ denotes the set of special Euclidean groups. $|\cdot|$ denotes the cardinality of a set.

Matrices. For a matrix $X \in \mathbb{R}^{m\times n}$, $[X]_{ij}$ denotes the (i, j)-th entry or (i, j)-th block of X, and $[X]_i$ denotes the i-th entry or i-th block of X. For symmetric matrices $X, Y \in \mathbb{R}^{n\times n}$, $X \succeq Y$ (or $Y \preceq X$) and $X \succ Y$ (or $Y \prec X$) mean

[1] A more complete summary of the notation is given in [43, Appendix A].

that $X - Y$ is positive (or negative) semidefinite and definite, respectively.

Inner Products and Norms. For a matrix $M \in \mathbb{R}^{n\times n}$, $\langle \cdot, \cdot \rangle_M : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}$ denotes the function

$$\langle X, Y \rangle_M \triangleq \mathrm{trace}(XMY^\top) \tag{1}$$

where $X, Y \in \mathbb{R}^{m\times n}$. If M is the identity matrix, $\langle \cdot, \cdot \rangle_M$ is also represented as $\langle \cdot, \cdot \rangle : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}$ such that

$$\langle X, Y \rangle \triangleq \mathrm{trace}(XY^\top). \tag{2}$$

For a positive semidefinite matrix $M \in \mathbb{R}^{n\times n}$, $\|\cdot\|_M : \mathbb{R}^{m\times n} \to \mathbb{R}^+$ denotes the function

$$\|X\|_M \triangleq \sqrt{\mathrm{trace}(XMX^\top)} \tag{3}$$

where $X \in \mathbb{R}^{m\times n}$. Also, $\|\cdot\|$ denotes the Frobenius norm of matrices and vectors, and $\|\cdot\|_2$ denotes the induced 2-norms of matrices and linear operators.

Riemannian Geometry. If $F(\cdot) : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a function, $\mathcal{M} \subset \mathbb{R}^{m\times n}$ is a Riemannian manifold and $X \in \mathcal{M}$, then $\nabla F(X)$ and $\mathrm{grad}\, F(X)$ denote the Euclidean and Riemannian gradients, respectively.

Graph Theory. PGO is represented as a directed graph $\vec{\mathcal{G}} = (\mathcal{V}, \vec{\mathcal{E}})$ where $\mathcal{V}$ and $\mathcal{E}$ are the sets of vertices and edges, respectively [8]. In distributed PGO, each vertex is described as an ordered pair $(\alpha, i) \in \mathcal{V}$ where α is the node index and i the local index of the vertex within node α. For any nodes α and β in distributed PGO, $\vec{\mathcal{E}}^{\alpha\beta}$ denotes the set of edges between nodes α and β:

$$\vec{\mathcal{E}}^{\alpha\beta} \triangleq \{(i, j) | ((\alpha, i), (\beta, j)) \in \vec{\mathcal{E}}\}; \tag{4}$$

and $\mathcal{N}_-^\alpha$ denotes the set of nodes with edges from node α:

$$\mathcal{N}_-^\alpha \triangleq \{\beta | \vec{\mathcal{E}}^{\alpha\beta} \neq \emptyset \text{ and } \alpha \neq \beta\}; \tag{5}$$

and $\mathcal{N}_+^\alpha$ denotes the set of nodes with edges to node α:

$$\mathcal{N}_+^\alpha \triangleq \{\beta | \vec{\mathcal{E}}^{\beta\alpha} \neq \emptyset \text{ and } \alpha \neq \beta\}; \tag{6}$$

and $\mathcal{N}^\alpha$ denotes the set of nodes with edges from or to node α:

$$\mathcal{N}^\alpha \triangleq \mathcal{N}_-^\alpha \cup \mathcal{N}_+^\alpha \triangleq \{\beta | \vec{\mathcal{E}}^{\alpha\beta} \neq \emptyset \text{ or } \vec{\mathcal{E}}^{\beta\alpha} \neq \emptyset \text{ and } \alpha \neq \beta\}. \tag{7}$$

Optimization. For optimization variables X, X^α, R^α, t^α, etc., the notation $X^{(\mathsf{k})}$, $X^{\alpha(\mathsf{k})}$, $R^{\alpha(\mathsf{k})}$, $t^{\alpha(\mathsf{k})}$, etc. denotes the k-th iterate of corresponding optimization variables.

IV. Problem Formulation

A. Distributed Pose Graph Optimization

In distributed PGO [36]–[38], we are given $|\mathcal{A}|$ nodes $\mathcal{A} \triangleq \{1, 2, \cdots, |\mathcal{A}|\}$ and each node $\alpha \in \mathcal{A}$ has n_α poses $g_1^\alpha, g_2^\alpha, \cdots, g_{n_\alpha}^\alpha \in SE(d)$. Let $g_{(\cdot)}^\alpha \triangleq (t_{(\cdot)}^\alpha, R_{(\cdot)}^\alpha)$ where $t_{(\cdot)}^\alpha \in \mathbb{R}^d$ is the translation and $R_{(\cdot)}^\alpha \in SO(d)$ the rotation. We consider the problem of estimating unknown poses $g_1^\alpha, g_2^\alpha, \cdots, g_{n_\alpha}^\alpha \in SE(d)$ for all the nodes $\alpha \in \mathcal{A}$ given intra-node noisy measurements $\tilde{g}_{ij}^{\alpha\alpha} \triangleq (\tilde{t}_{ij}^{\alpha\alpha}, \tilde{R}_{ij}^{\alpha\alpha}) \in SE(d)$ of the relative pose

$$g_{ij}^{\alpha\alpha} \triangleq \left(g_i^\alpha\right)^{-1} g_j^\alpha \in SE(d) \tag{8}$$

within a single node α, and inter-node noisy measurements $\tilde{g}_{ij}^{\alpha\beta} \triangleq (\tilde{t}_{ij}^{\alpha\beta}, \tilde{R}_{ij}^{\alpha\beta}) \in SE(d)$ of the relative pose

$$g_{ij}^{\alpha\beta} \triangleq \left(g_i^\alpha\right)^{-1} g_j^\beta \in SE(d) \tag{9}$$

between different nodes $\alpha \neq \beta$. In Eqs. (8) and (9), note that $\tilde{t}_{ij}^{\alpha\alpha}$ and $\tilde{t}_{ij}^{\alpha\beta} \in \mathbb{R}^d$ are translational measurements, and $\tilde{R}_{ij}^{\alpha\alpha}$ and $\tilde{R}_{ij}^{\alpha\beta} \in SO(d)$ are rotational measurements.

Following Eqs. (4) to (7), we represent distributed PGO as a directed graph $\vec{\mathcal{G}} = (\mathcal{V}, \vec{\mathcal{E}})$ such that unknown pose $g_i^\alpha \in SE(d)$ and noisy measurement $\tilde{g}_{ij}^{\alpha\beta} \in SE(d)$ have one-to-one correspondence to vertex $(\alpha, i) \in \mathcal{V}$ and directed edge $((\alpha, i), (\beta, j)) \in \vec{\mathcal{E}}$, respectively. We refer nodes α and $\beta \in \mathcal{A}$ as neighbors as long as either $\vec{\mathcal{E}}^{\alpha\beta} \neq \emptyset$ or $\vec{\mathcal{E}}^{\beta\alpha} \neq \emptyset$. Then, $\mathcal{N}_-^\alpha$ and $\mathcal{N}_+^\alpha$ are the sets of neighbors with a directed edge from and to node α, respectively, and $\mathcal{N}^\alpha$ is the set of neighbors with a directed edge connected to node α.

In the rest of this paper, we make the following assumption that each node can communicate with its neighbors and the network topology is unchanged during optimization. These assumptions are common in distributed PGO [36]–[39].

Assumption 1. *Each node α can communicate with its neighbors $\beta \in \mathcal{N}^\alpha$ and the network topology is fixed.*

B. Loss Kernels

In practice, it is inevitable that there exist inter-node measurements that are outliers resulting from false loop closures. These outliers adversely affect the overall performance of distributed PGO. To address this issue, it is popular to use non-trivial loss kernels—e.g., Huber and Welsch losses—to enhance the robustness of distributed PGO [44]–[46].

In this paper, we make the following assumption that applies to a broad class of loss kernels $\rho(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ in robotics and computer vision.

Assumption 2. *The loss kernel $\rho(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ satisfies the following properties:*

- *(a) $\rho(s) \geq 0$ for any $s \in \mathbb{R}^+$ and the equality "=" holds if and only if $s = 0$;*
- *(b) $\rho(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ is continuously differentiable;*
- *(c) $\rho(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ is a concave function;*
- *(d) $0 \leq \nabla\rho(s) \leq 1$ for any $s \in \mathbb{R}^+$ and $\nabla\rho(0) = 1$;*
- *(e) $\varphi(\cdot) : \mathbb{R}^{m\times n} \to \mathbb{R}$ with $\varphi(X) \triangleq \rho(\|X\|^2)$ has Lipschitz continuous gradient, i.e., there exists $\mu > 0$ such that $\|\nabla\varphi(X) - \nabla\varphi(X')\| \leq \mu \cdot \|X - X'\|$ for any $X, X' \in \mathbb{R}^{m\times n}$.*

In the following, we present some examples of loss kernels (see Fig. 1) satisfying Assumption 2.

Example 1 (Trivial Loss)**.**

$$\rho(s) = s. \tag{10}$$

Example 2 (Huber Loss)**.**

$$\rho(s) = \begin{cases} s, & |s| \leq a, \\ 2\sqrt{a|s|} - a, & |s| \geq a \end{cases} \tag{11}$$

where $a > 0$.

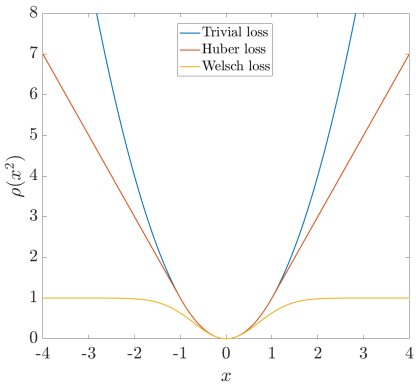


Fig. 1: $\rho(x^2)$ for the trivial, Huber, Welsch loss kernels.

Example 3 (Welsch Loss)**.**

$$\rho(s) = a - a\exp\left(-\frac{s}{a}\right) \tag{12}$$

where $a > 0$.

C. Objective Function

Recall that each node $\alpha \in \mathcal{A}$ has n_α unknown poses g_1^α, g_2^α, $\cdots$, $g_{n_\alpha}^\alpha \in SE(d)$. For notational simplicity, we define $\mathcal{X}^\alpha$ and $\mathcal{X}$ as

$$\mathcal{X}^\alpha \triangleq \mathbb{R}^{d\times n_\alpha} \times SO(d)^{n_\alpha}$$

and

$$\mathcal{X} \triangleq \mathcal{X}^1 \times \cdots \times \mathcal{X}^{|\mathcal{A}|} \subset \mathbb{R}^{d\times(d+1)n},$$

respectively, where $n \triangleq \sum_{\alpha\in\mathcal{A}} n_\alpha$. Furthermore, we represent $g_i^\alpha \in SE(d)$, i.e., the i-th pose of node $\alpha \in \mathcal{A}$, as a $d\times(d+1)$ matrix

$$X_i^\alpha \triangleq \begin{bmatrix} t_i^\alpha & R_i^\alpha \end{bmatrix} \in SE(d) \subset \mathbb{R}^{d\times(d+1)}, \tag{13}$$

represent $(g_1^\alpha, g_2^\alpha, \cdots, g_{n_\alpha}^\alpha) \in SE(d)^{n_\alpha}$, i.e., all the poses of node $\alpha \in \mathcal{A}$, as an element of $\mathcal{X}^\alpha$ as well as a $d \times (d+1)n_\alpha$ matrix

$$X^\alpha \triangleq \begin{bmatrix} t^\alpha & R^\alpha \end{bmatrix} \in \mathcal{X}^\alpha \subset \mathbb{R}^{d\times(d+1)n_\alpha}, \tag{14}$$

where

$$t^\alpha \triangleq \begin{bmatrix} t_1^\alpha & \cdots & t_{n_\alpha}^\alpha \end{bmatrix} \in \mathbb{R}^{d\times n_\alpha}$$

and

$$R^\alpha \triangleq \begin{bmatrix} R_1^\alpha & \cdots & R_{n_\alpha}^\alpha \end{bmatrix} \in SO(d)^{n_\alpha} \subset \mathbb{R}^{d\times dn_\alpha},$$

and represent $\{(g_1^\alpha, g_2^\alpha, \cdots, g_{n_\alpha}^\alpha)\}_{\alpha\in\mathcal{A}} \in SE(d)^n$, i.e., all the poses of distributed PGO, as an element of $\mathcal{X}$ as well as a $d \times (d+1)n$ matrix

$$X \triangleq \begin{bmatrix} X^1 & \cdots & X^{|\mathcal{A}|} \end{bmatrix} \in \mathcal{X} \subset \mathbb{R}^{d\times(d+1)n}. \tag{15}$$

Remark 1. *$\mathcal{X}^\alpha$ and $\mathcal{X}$ are by definition homeomorphic to $SE(d)^{n_\alpha}$ and $SE(d)^n$, respectively. Thus, $X^\alpha \in \mathcal{X}^\alpha$ and $X \in \mathcal{X}$ are sufficient to represent elements of $SE(d)^{n_\alpha}$ and $SE(d)^n$.*

Following [8], [31], [32], distributed PGO can be formulated as an optimization problem on $X = \begin{bmatrix} X^1 & \cdots & X^{|\mathcal{A}|} \end{bmatrix} \in \mathcal{X}$:

Problem 1 (Distributed Pose Graph Optimization)**.**

$$\min_{X\in\mathcal{X}} F(X). \tag{16}$$

The objective function $F(X)$ in Eq. (16) is defined as

$$\begin{aligned} F(X) \triangleq \sum_{\alpha\in\mathcal{A}} \sum_{(i,j)\in\vec{\mathcal{E}}^{\alpha\alpha}} \frac{1}{2}\Big[\kappa_{ij}^{\alpha\alpha}\|R_i^\alpha \widetilde{R}_{ij}^{\alpha\alpha} - R_j^\alpha\|^2 + \\ \tau_{ij}^{\alpha\alpha}\|R_i^\alpha \tilde{t}_{ij}^{\alpha\alpha} + t_i^\alpha - t_j^\alpha\|^2\Big] + \\ \sum_{\substack{\alpha,\beta\in\mathcal{A},\\ \alpha\neq\beta}} \sum_{(i,j)\in\vec{\mathcal{E}}^{\alpha\beta}} \frac{1}{2}\Big[\rho\left(\kappa_{ij}^{\alpha\beta}\|R_i^\alpha \widetilde{R}_{ij}^{\alpha\beta} - R_j^\beta\|^2 + \right. \\ \left. \tau_{ij}^{\alpha\beta}\|R_i^\alpha \tilde{t}_{ij}^{\alpha\beta} + t_i^\alpha - t_j^\beta\|^2\right)\Big], \end{aligned} \tag{17}$$

where $\kappa_{ij}^{\alpha\alpha}$, $\tau_{ij}^{\alpha\alpha}$, $\kappa_{ij}^{\alpha\beta}$, $\tau_{ij}^{\alpha\beta}$ are the weights and $\rho(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ is the loss kernel.

For notational simplicity, $F(X)$ in Eq. (17) can be also rewritten as

$$\begin{aligned} F(X) = \sum_{\alpha\in\mathcal{A}} \sum_{(i,j)\in\vec{\mathcal{E}}^{\alpha\alpha}} F_{ij}^{\alpha\alpha}(X) + \\ \sum_{\substack{\alpha,\beta\in\mathcal{A},\\ \alpha\neq\beta}} \sum_{(i,j)\in\vec{\mathcal{E}}^{\alpha\beta}} F_{ij}^{\alpha\beta}(X) \end{aligned} \tag{18}$$

where

$$\begin{aligned} F_{ij}^{\alpha\alpha}(X) \triangleq \frac{1}{2}\kappa_{ij}^{\alpha\alpha}\|R_i^\alpha \widetilde{R}_{ij}^{\alpha\alpha} - R_j^\alpha\|^2 + \\ \frac{1}{2}\tau_{ij}^{\alpha\alpha}\|R_i^\alpha \tilde{t}_{ij}^{\alpha\alpha} + t_i^\alpha - t_j^\alpha\|^2, \end{aligned} \tag{19a}$$

$$\begin{aligned} F_{ij}^{\alpha\beta}(X) \triangleq \frac{1}{2}\rho\Big(\kappa_{ij}^{\alpha\beta}\|R_i^\alpha \widetilde{R}_{ij}^{\alpha\beta} - R_j^\beta\|^2 + \\ \frac{1}{2}\tau_{ij}^{\alpha\beta}\|R_i^\alpha \tilde{t}_{ij}^{\alpha\beta} + t_i^\alpha - t_j^\beta\|^2\Big). \end{aligned} \tag{19b}$$

Note that $F_{ij}^{\alpha\alpha}(X)$ and $F_{ij}^{\alpha\beta}$ corresponds to intra- and inter-node measurements, respectively.

In the next sections, we will present MM methods for distributed PGO, which is the major contribution of this paper.

V. The Majorization of Loss Kernels

In this section, we present surrogate functions majorizing the loss kernels $\rho(\cdot)$. The resulting surrogate functions lead to an intermediate upper bound of distributed PGO while attaining the same value as the original objective function at each iterate.

It is straightforward to show that there exists sparse and positive semidefinite matrices $M_{ij}^{\alpha\beta} \in \mathbb{R}^{(d+1)n\times(d+1)n}$ for either $\alpha = \beta$ or $\alpha \neq \beta$ such that

$$\begin{aligned} \frac{1}{2}\|X\|_{M_{ij}^{\alpha\beta}}^2 = \frac{1}{2}\kappa_{ij}^{\alpha\beta}\|R_i^\alpha \widetilde{R}_{ij}^{\alpha\beta} - R_j^\beta\|^2 + \\ \frac{1}{2}\tau_{ij}^{\alpha\beta}\|R_i^\alpha \tilde{t}_{ij}^{\alpha\beta} + t_i^\alpha - t_j^\beta\|^2. \end{aligned} \tag{20}$$

Then, in terms of intra-node measurements with $\alpha = \beta$ and inter-node measurements with $\alpha \neq \beta$, $F_{ij}^{\alpha\alpha}(X)$ and $F_{ij}^{\alpha\beta}$ take the form of

$$F_{ij}^{\alpha\alpha}(X) \triangleq \frac{1}{2}\|X\|_{M_{ij}^{\alpha\alpha}}^2, \tag{21a}$$

$$F_{ij}^{\alpha\beta}(X) \triangleq \frac{1}{2}\rho\big(\|X\|^2_{M_{ij}^{\alpha\beta}}\big). \tag{21b}$$

From Eqs. (19a) and (19b), we obtain an upper bound of $F_{ij}^{\alpha\alpha}(X)$ and $F_{ij}^{\alpha\beta}(X)$ as the following proposition states.

Proposition 1. *Let $X^{(\mathsf{k})} = \begin{bmatrix} X^{1(\mathsf{k})} & \cdots & X^{|\mathcal{A}|(\mathsf{k})} \end{bmatrix} \in \mathcal{X}$ with $X^{\alpha(\mathsf{k})} \in \mathcal{X}^\alpha$ be an iterate of Eq. (16). If $\rho(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ is a loss kernel that satisfies Assumption 2, then we obtain*

$$\frac{1}{2}\omega_{ij}^{\alpha\beta(\mathsf{k})}\|X - X^{(\mathsf{k})}\|^2_{M_{ij}^{\alpha\beta}} + \big\langle \nabla F_{ij}^{\alpha\beta}(X^{(\mathsf{k})}), X - X^{(\mathsf{k})}\big\rangle + F_{ij}^{\alpha\beta}(X^{(\mathsf{k})}) \geq F_{ij}^{\alpha\beta}(X) \tag{22}$$

for any X and $X^{(\mathsf{k})} \in \mathbb{R}^{d\times(d+1)n}$, in which $\omega_{ij}^{\alpha\beta(\mathsf{k})} \in \mathbb{R}$ is defined as

$$\omega_{ij}^{\alpha\beta(\mathsf{k})} \triangleq \begin{cases} 1, & \alpha = \beta, \\ \nabla\rho\big(\|X^{(\mathsf{k})}\|^2_{M_{ij}^{\alpha\beta}}\big), & \alpha \neq \beta. \end{cases} \tag{23}$$

In Eq. (22), the equality "=" holds as long as $X = X^{(\mathsf{k})}$.

Proof. See [43, Appendix B]. □

Note that $F(X)$, as is shown in Eq. (18), is equivalent to the sum of all $F_{ij}^{\alpha\alpha}(X)$ and $F_{ij}^{\alpha\beta}(X)$. Then, an immediate upper bound of $F(X)$ resulting from Proposition 1 is

$$\frac{1}{2}\big\|X - X^{(\mathsf{k})}\big\|^2_{M^{(\mathsf{k})}} + \big\langle \nabla F(X^{(\mathsf{k})}), X - X^{(\mathsf{k})}\big\rangle + F(X^{(\mathsf{k})}) \geq F(X) \tag{24}$$

in which $M^{(\mathsf{k})} \in \mathbb{R}^{(d+1)n\times(d+1)n}$ is a positive semidefinite matrix that is defined as

$$M^{(\mathsf{k})} \triangleq \sum_{\alpha\in\mathcal{A}} \sum_{(i,j)\in\vec{\mathcal{E}}^{\alpha\alpha}} M_{ij}^{\alpha\alpha} + \sum_{\substack{\alpha,\beta\in\mathcal{A},\\ \alpha\neq\beta}} \sum_{(i,j)\in\vec{\mathcal{E}}^{\alpha\beta}} \omega_{ij}^{\alpha\beta(\mathsf{k})} \cdot M_{ij}^{\alpha\beta} \in \mathbb{R}^{(d+1)n\times(d+1)n}. \tag{25}$$

In addition, the equality "=" in Eq. (24) holds as long as $X = X^{(\mathsf{k})}$.

Remark 2. *If the loss kernel $\rho(\cdot)$ is non-trivial, $\omega_{ij}^{\alpha\beta(\mathsf{k})}$ is a function of $X^{(\mathsf{k})}$ as defined in Eq. (23), and $M^{(\mathsf{k})}$ is a positive semidefinite matrix depending on $X^{(\mathsf{k})}$ as well.*

It is obvious that Eq. (24) has $X^\alpha \in \mathcal{X}^\alpha$ of different nodes coupled with each other, and as a result, is difficult to be used for distributed PGO. In spite of that, as is shown in the next sections, Eq. (24) is still useful for the development and analysis of our MM methods for distributed PGO.

VI. The Majorization of Distributed Pose Graph Optimization

In this section, following a similar procedure to our previous works [31], [32], we present surrogate functions $G(X|X^{(\mathsf{k})})$ and $H(X|X^{(\mathsf{k})})$ that majorize the objective function $F(X)$. The surrogate functions $G(X|X^{(\mathsf{k})})$ and $H(X|X^{(\mathsf{k})})$ decouple unknown poses of different nodes, and thus, are critical to our MM methods for distributed PGO.

A. The Majorization of $F_{ij}^{\alpha\beta}(X)$

For any matrices B, C and $P \in \mathbb{R}^{m\times n}$, it can be shown that

$$\frac{1}{2}\|B - C\|^2_{M_{ij}^{\alpha\beta}} \leq \|B - P\|^2_{M_{ij}^{\alpha\beta}} + \|C - P\|^2_{M_{ij}^{\alpha\beta}} \tag{26}$$

as long as $M_{ij}^{\alpha\beta} \in \mathbb{R}^{n\times n}$ is positive semidefinite, where "=" holds if

$$P = \frac{1}{2}B + \frac{1}{2}C.$$

If we let $P = \mathbf{0}$, Eq. (26) becomes

$$\frac{1}{2}\|B - C\|^2_{M_{ij}^{\alpha\beta}} \leq \|B\|^2_{M_{ij}^{\alpha\beta}} + \|C\|^2_{M_{ij}^{\alpha\beta}}, \tag{27}$$

which holds for any B and $C \in \mathbb{R}^{m\times n}$. Applying Eq. (27) on the right-hand side of Eq. (20), we obtain

$$\begin{aligned} \frac{1}{2}\|X\|^2_{M_{ij}^{\alpha\beta}} \leq{}& \kappa_{ij}^{\alpha\beta}\|R_i^\alpha \widetilde{R}_{ij}^{\alpha\beta}\|^2 + \kappa_{ij}^{\alpha\beta}\|R_j^\beta\|^2 + \\ & \tau_{ij}^{\alpha\beta}\|R_i^\alpha \tilde{t}_{ij}^{\alpha\beta} + t_i^\alpha\|^2 + \tau_{ij}^{\alpha\beta}\|t_j^\beta\|^2 \\ ={}& \kappa_{ij}^{\alpha\beta}\|R_i^\alpha\|^2 + \kappa_{ij}^{\alpha\beta}\|R_j^\beta\|^2 + \\ & \tau_{ij}^{\alpha\beta}\|R_i^\alpha \tilde{t}_{ij}^{\alpha\beta} + t_i^\alpha\|^2 + \tau_{ij}^{\alpha\beta}\|t_j^\beta\|^2, \end{aligned} \tag{28}$$

where the last equality is due to $\big(\widetilde{R}_{ij}^{\alpha\beta}\big)^\top \widetilde{R}_{ij}^{\alpha\beta} = \widetilde{R}_{ij}^{\alpha\beta}\big(\widetilde{R}_{ij}^{\alpha\beta}\big)^\top = \mathbf{I}$. Furthermore, there exists a positive semidefinite matrix $\Omega_{ij}^{\alpha\beta} \in \mathbb{R}^{(d+1)n\times(d+1)n}$ such that the right-hand side of Eq. (28) can be rewritten as

$$\begin{aligned} \frac{1}{2}\|X\|^2_{\Omega_{ij}^{\alpha\beta}} ={}& \kappa_{ij}^{\alpha\beta}\|R_i^\alpha\|^2 + \kappa_{ij}^{\alpha\beta}\|R_j^\beta\|^2 + \\ & \tau_{ij}^{\alpha\beta}\|R_i^\alpha \tilde{t}_{ij}^{\alpha\beta} + t_i^\alpha\|^2 + \tau_{ij}^{\alpha\beta}\|t_j^\beta\|^2, \end{aligned} \tag{29}$$

where $\Omega_{ij}^{\alpha\beta}$ is a block diagonal matrix decoupling unknown poses of different nodes. Replacing the right-hand side of Eq. (28) with Eq. (29) results in

$$\frac{1}{2}\|X\|^2_{M_{ij}^{\alpha\beta}} \leq \frac{1}{2}\|X\|^2_{\Omega_{ij}^{\alpha\beta}}$$

for any $X \in \mathbb{R}^{d\times(d+1)n}$, which suggests

$$\Omega_{ij}^{\alpha\beta} \succeq M_{ij}^{\alpha\beta}. \tag{30}$$

With $\Omega_{ij}^{\alpha\beta} \in \mathbb{R}^{(d+1)n\times(d+1)n}$ in Eqs. (29) and (30), we define $E_{ij}^{\alpha\beta}(\cdot|X^{(\mathsf{k})}) : \mathbb{R}^{d\times(d+1)n} \to \mathbb{R}$:

$$E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})}) \triangleq \frac{1}{2}\omega_{ij}^{\alpha\beta(\mathsf{k})}\|X - X^{(\mathsf{k})}\|^2_{\Omega_{ij}^{\alpha\beta}} + \big\langle \nabla F_{ij}^{\alpha\beta}(X^{(\mathsf{k})}), X - X^{(\mathsf{k})}\big\rangle + F_{ij}^{\alpha\beta}(X^{(\mathsf{k})}), \tag{31}$$

where $\omega_{ij}^{\alpha\beta(\mathsf{k})}$ is given in Eq. (23). From the equation above, it can be concluded that $E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})})$ majorizes $F_{ij}^{\alpha\beta}(X)$ as the following proposition states, which is important for the construction of surrogate functions for distributed PGO.

Proposition 2. *Given any nodes α, $\beta \in \mathcal{A}$ with either $\alpha = \beta$ or $\alpha \neq \beta$, if $\rho(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ is a loss kernel that satisfies Assumption 2, then we obtain*

$$E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})}) \geq F_{ij}^{\alpha\beta}(X). \tag{32}$$

for any $X \in \mathbb{R}^{d\times(d+1)n}$. In the equation above, the equality "=" holds if $X = X^{(\mathsf{k})}$.

Proof. See [43, Appendix C]. □

B. The Majorization of $F(X)$

From Proposition 2, it is straightforward to construct surrogate functions majorizing $F(X)$ in Eq. (17) as the following proposition states.

Proposition 3. *Let $X^{(\mathsf{k})} = \begin{bmatrix} X^{1(\mathsf{k})} & \cdots & X^{|\mathcal{A}|(\mathsf{k})} \end{bmatrix} \in \mathcal{X}$ with $X^{\alpha(\mathsf{k})} \in \mathcal{X}^\alpha$ be an iterate of $X \in \mathcal{X}$ in Eq. (16). Suppose $G(\cdot|X^{(\mathsf{k})}) : \mathbb{R}^{d\times(d+1)n} \to \mathbb{R}$ is a function:*

$$G(X|X^{(\mathsf{k})}) \triangleq \sum_{\alpha\in\mathcal{A}} \sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\alpha\alpha}} F_{ij}^{\alpha\alpha}(X) + \sum_{\substack{\alpha,\beta\in\mathcal{A},\\ \alpha\neq\beta}} \sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\alpha\beta}} E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})}) + \frac{\xi}{2}\left\|X - X^{(\mathsf{k})}\right\|^2 \quad (33)$$

where $\xi \in \mathbb{R}$ and $\xi \geq 0$. Then, we have the following results:

(a) For any $X \in \mathbb{R}^{d\times(d+1)}$ and $X^{(\mathsf{k})} \in \mathcal{X}$,

$$G(X|X^{(\mathsf{k})}) \geq F(X) \quad (34)$$

where the equality "=" holds if $X = X^{(\mathsf{k})}$.

(b) $G(X|X^{(\mathsf{k})})$ is equivalent to

$$G(X|X^{(\mathsf{k})}) = \sum_{\alpha\in\mathcal{A}} G^\alpha(X^\alpha|X^{(\mathsf{k})}) + F(X^{(\mathsf{k})}), \quad (35)$$

where $G^\alpha(X^\alpha|X^{(\mathsf{k})})$ is a function of $X^\alpha \in \mathcal{X}^\alpha$ within a single node α.

(c) For any node $\alpha \in \mathcal{A}$, there exists positive-semidefinite matrices $\Gamma^{\alpha(\mathsf{k})} \in \mathbb{R}^{(d+1)n_\alpha\times(d+1)n_\alpha}$ such that

$$G^\alpha(X^\alpha|X^{(\mathsf{k})}) \triangleq \frac{1}{2}\|X^\alpha - X^{\alpha(\mathsf{k})}\|^2_{\Gamma^{\alpha(\mathsf{k})}} + \left\langle \nabla_{X^\alpha} F(X^{(\mathsf{k})}),\, X^\alpha - X^{\alpha(\mathsf{k})} \right\rangle \quad (36)$$

where $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$ is the Euclidean gradient of $F(X)$ with respect to $X^\alpha \in \mathcal{X}^\alpha$ at $X^{(\mathsf{k})} \in \mathcal{X}$.

Proof. See [43, Appendix D]. □

Following Proposition 3, we might further majorize $F(X)$ as well as $G(X|X^{(\mathsf{k})})$ by applying Eq. (32) to Eq. (33) and replacing $F_{ij}^{\alpha\alpha}(X|X^{(\mathsf{k})})$ with $E_{ij}^{\alpha\alpha}(X|X^{(\mathsf{k})})$, which results in the following proposition.

Proposition 4. *Let $X^{(\mathsf{k})} = \begin{bmatrix} X^{1(\mathsf{k})} & \cdots & X^{|\mathcal{A}|(\mathsf{k})} \end{bmatrix} \in \mathcal{X}$ with $X^{\alpha(\mathsf{k})} \in \mathcal{X}^\alpha$ be an iterate of $X \in \mathcal{X}$ in Eq. (16), and $X_i^{\alpha(\mathsf{k})} = \begin{bmatrix} t^{\alpha(\mathsf{k})} & R^{\alpha(\mathsf{k})} \end{bmatrix} \in SE(d)$ the corresponding iterate of $X_i^\alpha \in SE(d)$. Suppose $H(\cdot|X^{(\mathsf{k})}) : \mathbb{R}^{d\times(d+1)n} \to \mathbb{R}$ is a function:*

$$H(X|X^{(\mathsf{k})}) = \sum_{\alpha\in\mathcal{A}} \sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\alpha\alpha}} E_{ij}^{\alpha\alpha}(X|X^{(\mathsf{k})}) + \sum_{\substack{\alpha,\beta\in\mathcal{A},\\ \alpha\neq\beta}} \sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\alpha\beta}} E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})}) + \frac{\zeta}{2}\left\|X - X^{(\mathsf{k})}\right\|^2 \quad (37)$$

where $\zeta \in \mathbb{R}$ and $\zeta \geq \xi \geq 0$. Note that $\xi \in \mathbb{R}$ is given in Eq. (33). Then, we have the following results:

(a) For any $X \in \mathbb{R}^{d\times(d+1)}$ and $X^{(\mathsf{k})} \in \mathcal{X}$,

$$H(X|X^{(\mathsf{k})}) \geq G(X|X^{(\mathsf{k})}) \geq F(X) \quad (38)$$

where the equality "=" holds if $X = X^{(\mathsf{k})}$.

(b) $H(X|X^{(\mathsf{k})})$ is equivalent to

$$\begin{aligned} H(X|X^{(\mathsf{k})}) &= \sum_{\alpha\in\mathcal{A}} H^\alpha(X^\alpha|X^{(\mathsf{k})}) + F(X^{(\mathsf{k})}) \\ &= \sum_{\alpha\in\mathcal{A}} \sum_{i=1}^{n_\alpha} H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})}) + F(X^{(\mathsf{k})}) \end{aligned} \quad (39)$$

where $H^\alpha(X^\alpha|X^{(\mathsf{k})})$ is a function of $X^\alpha \in \mathcal{X}^\alpha$ within a single node α, and $H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})})$ is a function of a single pose $X_i^\alpha \in SE(d) \subset \mathbb{R}^{d\times(d+1)}$.

(c) For any node $\alpha \in \mathcal{A}$ and $i \in \{1, \cdots, n_\alpha\}$, there exists positive-semidefinite matrices $\Pi^{\alpha(\mathsf{k})} \in \mathbb{R}^{(d+1)n_\alpha\times(d+1)n_\alpha}$ and $\Pi_i^{\alpha(\mathsf{k})} \in \mathbb{R}^{(d+1)\times(d+1)}$ such that

$$H^\alpha(X^\alpha|X^{(\mathsf{k})}) = \frac{1}{2}\|X^\alpha - X^{\alpha(\mathsf{k})}\|^2_{\Pi^{\alpha(\mathsf{k})}} + \left\langle \nabla_{X^\alpha} F(X^{(\mathsf{k})}),\, X^\alpha - X^{\alpha(\mathsf{k})} \right\rangle, \quad (40)$$

$$H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})}) = \frac{1}{2}\|X_i^\alpha - X_i^{\alpha(\mathsf{k})}\|^2_{\Pi_i^{\alpha(\mathsf{k})}} + \left\langle \nabla_{X_i^\alpha} F(X^{(\mathsf{k})}),\, X_i^\alpha - X_i^{\alpha(\mathsf{k})} \right\rangle \quad (41)$$

where $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$ and $\nabla_{X_i^\alpha} F(X^{(\mathsf{k})})$ are the Euclidean gradients of $F(X)$ with respect to $X^\alpha \in \mathcal{X}^\alpha$ and $X_i^\alpha \in SE(d)$ at $X^{(\mathsf{k})} \in \mathcal{X}$, respectively.

Proof. The proof is similar to that of Proposition 3. □

Remark 3. *As a result of Eq. (39), $H^\alpha(X^\alpha|X^{(\mathsf{k})})$ can be rewritten as the sum of $H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})})$, i.e.,*

$$H^\alpha(X^\alpha|X^{(\mathsf{k})}) = \sum_{i=1}^{n_\alpha} H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})}). \quad (42)$$

Note that $H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})})$ in Eqs. (39) and (42) relies on a single pose $X_i^\alpha \in SE(d) \subset \mathbb{R}^{d\times(d+1)}$. This will be later exploited in Sections VII to IX to improve the computational efficiency of distributed PGO.

Remark 4. *Propositions Propositions 3 and 4 indicate that $G(X|X^{(\mathsf{k})})$ and $H(X|X^{(\mathsf{k})})$ not only majorize $F(X)$ but also decouple poses from different nodes through $G^\alpha(X^\alpha|X^{(\mathsf{k})})$ and $H^\alpha(X^\alpha|X^{(\mathsf{k})})$, making it possible to implement majorization minimization methods for distributed PGO.*

Remark 5. *ζ and ξ in Eqs. (33) and (37) are important for the convergence analysis of MM methods for distributed PGO in Sections VII to IX . It is recommended to be set $\zeta > \xi > 0$ but close to zero such that $G(X|X^{(\mathsf{k})})$ and $H(X|X^{(\mathsf{k})})$ are tighter upper bounds of $F(X)$ and yield faster convergence.*

Recall that $G^\alpha(X^\alpha|X^{(\mathsf{k})})$ and $H^\alpha(X^\alpha|X^{(\mathsf{k})})$ in Eqs. (36) and (40) rely on $\Gamma^{\alpha(\mathsf{k})}$, $\Pi^{\alpha(\mathsf{k})}$, $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$, which—according to Eqs. (18) and (19)—are only related to $F_{ij}^{\alpha\alpha}(X)$, $F_{ij}^{\alpha\beta}(X)$, $F_{ji}^{\beta\alpha}(X)$. Also, Eq. (19) indicates that $F_{ij}^{\alpha\alpha}(X)$ depends on X_i^α and X_j^α, while $F_{ij}^{\alpha\beta}(X)$ and $F_{ji}^{\beta\alpha}(X)$ on X_i^α and X_j^β. Therefore, $\Gamma^{\alpha(\mathsf{k})}$, $\Pi^{\alpha(\mathsf{k})}$, $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$ can be evaluated as long as node α have access to X^β from its neighbor β. Furthermore, $G^\alpha(X^\alpha|X^{(\mathsf{k})})$ and $H^\alpha(X^\alpha|X^{(\mathsf{k})})$ can be constructed in a distributed setting with one inter-node communication round between neighboring nodes α and β.

In the next sections, we will present MM methods for distributed PGO using $G(X|X^{(\mathsf{k})})$ and $H(X|X^{(\mathsf{k})})$ that are guaranteed to converge to first-order critical points.

VII. The Majorization Minimization Method for Distributed Pose Graph Optimization

In distributed optimization, MM methods are one of the most popular first-order optimization methods [29], [30]. As mentioned before, MM methods solve an optimization problem by iteratively minimizing an upper bound of the objective function such that the objective value is nonincreasing. Recall that $G(X|X^{(\mathsf{k})})$ and $H(X|X^{(\mathsf{k})})$ majorize $F(X)$ and decouple poses from different nodes; see Propositions 3 and 4, respectively. Therefore, we might make use of MM methods where distributed PGO is reduced to independent optimization problems that can be solved in parallel. In Section VII-A, we propose MM methods for distributed PGO using $G(X|X^{(\mathsf{k})})$ and $H(X|X^{(\mathsf{k})})$. Then, in Section VII-B, we present the algorithm and prove that the proposed method is guaranteed to converge to first-order critical points.

A. Update Rule

According to Propositions 3 and 4, $G(X|X^{(\mathsf{k})})$ and $H(X|X^{(\mathsf{k})})$ are surrogate functions majorizing $F(X)$:

$$H(X|X^{(\mathsf{k})}) \geq G(X|X^{(\mathsf{k})}) \geq F(X), \tag{43}$$

$$H(X^{(\mathsf{k})}|X^{(\mathsf{k})}) = G(X^{(\mathsf{k})}|X^{(\mathsf{k})}) = F(X^{(\mathsf{k})}). \tag{44}$$

Following the notion of MM methods [29], we propose the following update rule:

$$X^{(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X\in\mathcal{X}} H(X|X^{(\mathsf{k})}), \tag{45}$$

$$X^{(\mathsf{k}+1)} \leftarrow \arg\min_{X\in\mathcal{X}} G(X|X^{(\mathsf{k})}). \tag{46}$$

Here, $X^{(\mathsf{k}+\frac{1}{2})}$ in Eq. (45) is first solved and used to initialize $X^{(\mathsf{k}+1)}$ in Eq. (46). Also, Eqs. (35) and (39) indicate that Eqs. (45) and (46) are equivalent to $|\mathcal{A}|$ independent optimization problems of $X^\alpha \in \mathcal{X}^\alpha$ within a single node α:

$$X^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X^\alpha\in\mathcal{X}^\alpha} H^\alpha(X^\alpha|X^{(\mathsf{k})}), \quad \forall\alpha\in\mathcal{A}, \tag{47}$$

$$X^{\alpha(\mathsf{k}+1)} \leftarrow \arg\min_{X^\alpha\in\mathcal{X}^\alpha} G^\alpha(X^\alpha|X^{(\mathsf{k})}), \quad \forall\alpha\in\mathcal{A}, \tag{48}$$

where $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ in Eq. (47) is the initial guess to solve $X^{\alpha(\mathsf{k}+1)}$ in Eq. (48). We remark that Eqs. (47) and (48) can be solved within a single node $\alpha \in \mathcal{A}$. Recalling from Eq. (42) that $H^\alpha(X^\alpha|X^{(\mathsf{k})}) = \sum_{i=1}^{n_\alpha} H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})})$, we further reduce Eq. (47) to $n \triangleq \sum_{\alpha\in\mathcal{A}} n_\alpha$ independent optimization problems on a single pose $X_i^\alpha \in SE(d)$:

$$X_i^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X_i^\alpha\in SE(d)} H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})}), \quad \forall\alpha\in\mathcal{A} \text{ and } i\in\{1,\cdots,n_\alpha\}. \tag{49}$$

In [43, Appendix K], we have shown that Eq. (49) admits an efficient closed-form solution involving only matrix multiplication and singular value decomposition [47].

Algorithm 1 The MM–PGO Method

1: **Input**: An initial iterate $X^{(0)} \in \mathcal{X}$ and $\zeta \geq \xi \geq 0$.
2: **Output**: A sequence of iterates $\{X^{(\mathsf{k})}\}$ and $\{X^{(\mathsf{k}+\frac{1}{2})}\}$.
3: **for** $\mathsf{k} \leftarrow 0, 1, 2, \cdots$ **do**
4: **for** node $\alpha \leftarrow 1, \cdots, |\mathcal{A}|$ **do**
5: retrieve $X^{\beta(\mathsf{k})}$ from $\beta \in \mathcal{N}_\alpha$
6: evaluate $\Gamma^{\alpha(\mathsf{k})}$, $\Pi^{\alpha(\mathsf{k})}$, $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$
7: $X^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X^\alpha\in\mathcal{X}^\alpha} H^\alpha(X^\alpha|X^{(\mathsf{k})})$ using Algorithm 2
8: $X^{\alpha(\mathsf{k}+1)} \leftarrow$ improve $\arg\min_{X^\alpha\in\mathcal{X}^\alpha} G^\alpha(X^\alpha|X^{(\mathsf{k})})$ with $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ as the initial guess
9: **end for**
10: **end for**

Algorithm 2 Solve $X^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X^\alpha\in\mathcal{X}^\alpha} H^\alpha(X^\alpha|X^{(\mathsf{k})})$

1: **Input**: $X^{\alpha(\mathsf{k})}$, $\Pi^{\alpha(\mathsf{k})}$, $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$.
2: **Output**: $X^{\alpha(\mathsf{k}+\frac{1}{2})}$.
3: **for** $i \leftarrow 1, \cdots, n_\alpha$ **do**
4: $X_i^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X_i^\alpha\in SE(d)} H_i^\alpha(X_i^\alpha|X^{(\mathsf{k})})$ using [43, Appendix K]
5: **end for**
6: retrieve $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ from $X_i^{\alpha(\mathsf{k}+\frac{1}{2})}$ in which $i = 1, \cdots, n_\alpha$

From Eqs. (43) and (44), we conclude that Eqs. (45) to (48) result in

$$F(X^{(\mathsf{k})}) = H(X^{(\mathsf{k})}|X^{(\mathsf{k})}) \geq H(X^{(\mathsf{k}+\frac{1}{2})}|X^{(\mathsf{k})}) \geq F(X^{(\mathsf{k}+\frac{1}{2})}), \tag{50a}$$

$$F(X^{(\mathsf{k})}) = G(X^{(\mathsf{k})}|X^{(\mathsf{k})}) \geq G(X^{(\mathsf{k}+1)}|X^{(\mathsf{k})}) \geq F(X^{(\mathsf{k}+1)}) \tag{50b}$$

which indicate $F(X^{(\mathsf{k}+\frac{1}{2})}) \leq F(X^{(\mathsf{k})})$ and $F(X^{(\mathsf{k}+1)}) \leq F(X^{(\mathsf{k})})$. Therefore, Eqs. (45) to (48) are a reasonable update rule for distributed PGO. In particular, we remark that Eqs. (45) to (48) combine the strengths of our previous work [31], [32]. Even though Eqs. (45) and (47) are motivated by [31], Eqs. (46) and (48) make better use of the information within a single node, and thus, take fewer iterations. In contrast to [32], since Eqs. (45) and (47) have an efficient closed-form solution to Eq. (49) that yields sufficient improvement, the time-consuming local minimization of Eqs. (46) and (48) is avoided as long as $X^{(\mathsf{k}+1)}$ and $X^{\alpha(\mathsf{k}+1)}$ are initialized with $X^{(\mathsf{k}+\frac{1}{2})}$ and $X^{\alpha(\mathsf{k}+\frac{1}{2})}$. Most importantly, as is shown later in Proposition 5, the proposed updated rule of Eqs. (45) to (48) has provable convergence to first-order critical points.

B. Algorithm

The proposed update rule results in the MM–PGO method for distributed PGO (Algorithm 1). The outline of MM–PGO is as follows:

1) In line 5 of Algorithm 1, each node α performs one inter-node communication round to retrieve $X^{\beta(\mathsf{k})}$ from its neighbors $\beta \in \mathcal{N}_\alpha$. Note that that no other inter-node communication is required.

2) In lines 6 of Algorithm 1, each node α evaluates Γ, Π, $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$ with $X^{\alpha(\mathsf{k})}$ and $X^{\beta(\mathsf{k})}$ where $\beta \in \mathcal{N}^\alpha$ are neighbors of node α.
3) In line 7 of Algorithm 1, we obtain the intermediate solution $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ using Algorithm 2. We have proved that the resulting $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ is already sufficient to guarantee the convergence to first-order critical points.
4) In line 4 of Algorithm 2, there exists an exact and efficient closed-form solution to $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ using [43, Appendix K].
5) In line 8 of Algorithm 1, we use $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ to initialize Eq. (48), and improve the final solution $X^{\alpha(\mathsf{k}+1)}$ through iterative optimization such that $G^\alpha(X^{\alpha(\mathsf{k}+1)}|X^{(\mathsf{k})}) \leq G^\alpha(X^{\alpha(\mathsf{k}+\frac{1}{2})}|X^{(\mathsf{k})})$. Note that $X^{\alpha(\mathsf{k}+1)}$ does not have to be a local optimal solution to Eq. (48), nevertheless, $X^{\alpha(\mathsf{k}+1)}$ is still expected to have a faster convergence than $X^{\alpha(\mathsf{k}+\frac{1}{2})}$.

Remark 6. *In line 5 of Algorithm 1, node α does not retrieve all the poses in $X^{\beta(\mathsf{k})}$—only poses related to inter-node measurements are needed and exchanged. This also applies to line 2 of Algorithm 4, and lines 5 and 13 of Algorithm 5 in the following sections.*

Remark 7. MM–PGO *(Algorithm 1) requires no inter-node communication except for line 5 of Algorithm 1 that is used to evaluate $\Gamma^{\alpha(\mathsf{k})}$, $\Pi^{\alpha(\mathsf{k})}$, $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$, which, as mentioned before, can be distributed with one inter-node communication round between neighboring nodes α and β without introducing any additional computation.*

Since $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ in Eq. (47) has a closed-form solution that can be efficiently computed, and Eq. (48) does not require $X^{\alpha(\mathsf{k}+1)}$ to be a local optimal solution, the overall computational efficiency of the MM–PGO method is significantly improved in contrast to [31], [32]. More importantly, the MM–PGO method still converges to first-order critical points as long as the following assumption holds.

Assumption 3. *For $X^{\alpha(\mathsf{k}+1)}$ and $X^{\alpha(\mathsf{k}+\frac{1}{2})}$, it is assumed that*

$$G^\alpha(X^{\alpha(\mathsf{k}+1)}|X^{(\mathsf{k})}) \leq G^\alpha(X^{\alpha(\mathsf{k}+\frac{1}{2})}|X^{(\mathsf{k})}) \tag{51}$$

for each node $\alpha = 1, 2 \cdots, |\mathcal{A}|$.

Note that Assumption 3 can be satisfied with ease as long as line 8 of Algorithm 1 is initialized with $X^{\alpha(\mathsf{k}+\frac{1}{2})}$. Then, we have the following proposition about the convergence of MM–PGO (Algorithm 1).

Proposition 5. *If Assumptions 1 to 3 hold, then for a sequence of $\{X^{(\mathsf{k})}\}$ and $\{X^{(\mathsf{k}+\frac{1}{2})}\}$ generated by Algorithm 1, we have*

(a) $F(X^{(\mathsf{k})})$ is nonincreasing as $\mathsf{k} \to \infty$;
(b) $F(X^{(\mathsf{k})}) \to F^\infty$ as $\mathsf{k} \to \infty$;
(c) $\|X^{(\mathsf{k}+1)} - X^{(\mathsf{k})}\| \to 0$ as $\mathsf{k} \to \infty$ if $\xi > 0$;
(d) $\|X^{(\mathsf{k}+\frac{1}{2})} - X^{(\mathsf{k})}\| \to 0$ as $\mathsf{k} \to \infty$ if $\zeta > \xi > 0$;
(e) if $\zeta > \xi > 0$, then there exists $\epsilon > 0$ such that

$$\min_{0\leq \mathsf{k} < \mathsf{K}} \|\mathrm{grad}\, F(X^{(\mathsf{k}+\frac{1}{2})})\| \leq \sqrt{\frac{2}{\epsilon} \cdot \frac{F(X^{(0)}) - F^\infty}{\mathsf{K}+1}}$$

for any $\mathsf{K} \geq 0$;
(f) if $\zeta > \xi > 0$, then $\mathrm{grad}\, F(X^{(\mathsf{k})}) \to \mathbf{0}$ and $\mathrm{grad}\, F(X^{(\mathsf{k}+\frac{1}{2})}) \to \mathbf{0}$ as $\mathsf{k} \to \infty$.

Proof. See [43, Appendix E]. □

Remark 8. *In contrast to other distributed PGO algorithms [36]–[39],* MM–PGO *has the mildest conditions for convergence and apply to a broad class of loss kernels.*

VIII. The Accelerated Majorization Minimization Method for Distributed Pose Graph Optimization with Master Node

In the last several decades, a number of accelerated first-order optimization methods have been proposed [33], [34]. Even though most of them were originally developed for convex optimization, it has been recently found that these accelerated methods also work well for nonconvex optimization [48]–[50]. In our previous works [31], [32], we proposed to use Nesterov's method to accelerate distributed PGO, which yield much faster convergence. Since MM–PGO is a first-order optimization method, it is possible to exploit Nesterov's method for acceleration. In Section VIII-A, we implement Nesterov's method to accelerate MM methods for distributed PGO. Then, in Section VIII-B, we introduce the adaptive restart scheme [35] to guarantee the convergence if a master node exists. Lastly, in Section VIII-C, we propose the accelerated MM method for distributed PGO with master and prove that such a method converges to first-order critical points.

A. Nesterov's Method

According to Propositions 3 and 4, Eqs. (45) and (46) are proximal operators of $F(X)$, making it possible to implement Nesterov's method [33], [34] for acceleration and resulting in the following update rule for $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ and $X^{\alpha(\mathsf{k}+1)}$:

$$s^{\alpha(\mathsf{k}+1)} = \frac{\sqrt{4 s^{\alpha(\mathsf{k})2} + 1} + 1}{2}, \tag{52}$$

$$\lambda^{\alpha(\mathsf{k})} = \frac{s^{\alpha(\mathsf{k})} - 1}{s^{\alpha(\mathsf{k}+1)}}, \tag{53}$$

$$Y^{\alpha(\mathsf{k})} = X^{\alpha(\mathsf{k})} + \lambda^{\alpha(\mathsf{k})} \cdot \left(X^{\alpha(\mathsf{k})} - X^{\alpha(\mathsf{k}-1)}\right), \tag{54}$$

$$X^{\alpha(\mathsf{k}+\frac{1}{2})} = \arg\min_{X^\alpha \in \mathcal{X}^\alpha} H^\alpha(X^\alpha|Y^{(\mathsf{k})}), \tag{55}$$

$$X^{\alpha(\mathsf{k}+1)} = \arg\min_{X^\alpha \in \mathcal{X}^\alpha} G^\alpha(X^\alpha|Y^{(\mathsf{k})}). \tag{56}$$

In Eqs. (55) and (56), $G^\alpha(\cdot|Y^{(\mathsf{k})}) : \mathcal{X}^\alpha \to \mathbb{R}$ and $H^\alpha(\cdot|Y^{(\mathsf{k})}) : \mathcal{X}^\alpha \to \mathbb{R}$ are surrogate functions at $Y^{\alpha(\mathsf{k})}$:

$$G^\alpha(X^\alpha|Y^{(\mathsf{k})}) = \frac{1}{2}\|X^\alpha - Y^{\alpha(\mathsf{k})}\|^2_{\Gamma^{\alpha(\mathsf{k})}} + \left\langle \nabla_{X^\alpha} F(Y^{(\mathsf{k})}),\, X^\alpha - Y^{\alpha(\mathsf{k})} \right\rangle, \tag{57}$$

$$H^\alpha(X^\alpha|Y^{(\mathsf{k})}) = \frac{1}{2}\|X^\alpha - Y^{\alpha(\mathsf{k})}\|^2_{\Pi^{\alpha(\mathsf{k})}} + \left\langle \nabla_{X^\alpha} F(Y^{(\mathsf{k})}),\, X^\alpha - Y^{\alpha(\mathsf{k})} \right\rangle, \tag{58}$$

where $\Gamma^{\alpha(\mathsf{k})}$ and $\Pi^{\alpha(\mathsf{k})}$ are the same as these in $G^\alpha(\cdot|X^{(\mathsf{k})})$ and $H^\alpha(\cdot|X^{(\mathsf{k})})$ in Eqs. (36) and (40).

The key insight of Nesterov's method is to exploit the momentum $X^{\alpha(\mathsf{k})} - X^{\alpha(\mathsf{k}-1)}$ for acceleration, which is essentially governed by Eqs. (52) to (54). Note that Nesterov's method using Eqs. (52) to (56) is equivalent to Eqs. (47) and (48) when $s^{\alpha(\mathsf{k})} = 1$ and $\lambda^{\alpha(\mathsf{k})} = 0$, and then increasingly affected by the momentum as $s^{\alpha(\mathsf{k})}$ and $\lambda^{\alpha(\mathsf{k})}$ increase.

Nesterov's method is known to converge quadratically for convex optimization while the unaccelerated MM method only has linear convergence [33], [34]. Even though distributed PGO is a nonconvex optimization problem, similar to [31], [32], Eqs. (52) to (56) using Nesterov's method for acceleration empirically have significant speedup while introducing almost no extra computation or communication compared to the MM–PGO method. Thus, it is preferable to adopt Nesterov's method to accelerate distributed PGO.

B. Adaptive Restart

In spite of faster convergence, Nesterov's accelerated distributed PGO using Eqs. (52) to (56) is no longer nonincreasing, and might fail to converge due to the nonconvexity of PGO. Fortunately, such a problem can be remedied with an adaptive restart scheme [31], [32], [35] as the following.

Let $\overline{F}^{(\mathsf{k})}$ be an exponential moving averaging of $F(X^{(0)})$, $F(X^{(1)})$, $\cdots$, $F(X^{(\mathsf{k})})$:

$$\overline{F}^{(\mathsf{k})} \triangleq \begin{cases} F(X^{(0)}), & \mathsf{k}=0, \\ (1-\eta)\cdot \overline{F}^{(\mathsf{k}-1)} + \eta \cdot F(X^{(\mathsf{k})}), & \text{otherwise} \end{cases} \tag{59}$$

where $\eta \in (0,\, 1]$. Following [31], [50], [51], the adaptive restart scheme guarantees the convergence by keeping $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$. Even though it is not obvious, $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$ can be achieved with the following steps:

1) Update $X^{(\mathsf{k}+\frac{1}{2})}$ and $X^{(\mathsf{k}+1)}$ by solving Eqs. (55) and (56) for each node $\alpha \in \mathcal{A}$;
2) If $F(X^{(\mathsf{k}+\frac{1}{2})}) > \overline{F}^{(\mathsf{k})}$, update $X^{(\mathsf{k}+\frac{1}{2})}$ again by solving Eq. (47) for each node $\alpha \in \mathcal{A}$;
3) If $F(X^{(\mathsf{k}+1)}) > \overline{F}^{(\mathsf{k})}$, update $X^{(\mathsf{k}+1)}$ again by solving Eq. (48) and reduce $s^{\alpha(\mathsf{k}+1)}$ for each node $\alpha \in \mathcal{A}$.

Due to space limitation, the complete analysis of the adaptive restart scheme is left in [43, Appendix F] where more details are presented.

Remark 9. *Since $\eta \in (0,\, 1]$ in Eq. (59), then $\overline{F}^{(\mathsf{k}+1)} \leq \overline{F}^{(\mathsf{k})}$ as long as $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$. Note that Eqs. (47) and (48) lead to $F(X^{(\mathsf{k}+\frac{1}{2})}) \leq F(X^{(\mathsf{k})})$ and $F(X^{(\mathsf{k}+1)}) \leq F(X^{(\mathsf{k})})$, and we have proved in [43, Appendix F] that $F(X^{(\mathsf{k})}) \leq \overline{F}^{(\mathsf{k})}$. This suggests $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$ if $X^{(\mathsf{k}+1)}$ is updated from Eq. (48), and thus, satisfies the restart conditions. Then, the adaptive restart scheme above results in a nonincreasing sequence of $\overline{F}^{(\mathsf{k})}$. Furthermore, [43, Appendix F] indicates that such an adaptive restart scheme is sufficient to guarantee the convergence to first-order critical points under mild conditions.*

Note that one has to aggregate information across the network to evaluate and compare $\overline{F}^{(\mathsf{k})}$, $F(X^{(\mathsf{k}+\frac{1}{2})})$, $F(X^{(\mathsf{k}+1)})$ using Eqs. (18) and (59). Thus, a master node capable of communicating with each node $\alpha \in \mathcal{A}$ is required. In the rest of this section, we make the following assumption about the existence of such a master node.

Assumption 4. *There is a master node to retrieve $X^{\alpha(\mathsf{k})}$ and $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ from each node $\alpha \in \mathcal{A}$ and evaluate $\overline{F}^{(\mathsf{k})}$, $F(X^{(\mathsf{k}+\frac{1}{2})})$, $F(X^{(\mathsf{k}+1)})$.*

Algorithm 3 The AMM–PGO* Method

1: **Input**: An initial iterate $X^{(0)} \in \mathcal{X}$, and $\zeta \geq \xi \geq 0$, and $\eta \in (0,\, 1]$, and $\psi > 0$, and $\phi > 0$.
2: **Output**: A sequence of iterates $\{X^{(\mathsf{k})}\}$ and $\{X^{(\mathsf{k}+\frac{1}{2})}\}$.
3: **for** node $\alpha \leftarrow 1, \cdots, |\mathcal{A}|$ **do**
4: $\quad X^{\alpha(-1)} \leftarrow X^{\alpha(0)}$ and $s^{\alpha(0)} \leftarrow 1$
5: $\quad$ send $X^{\alpha(0)}$ to the master node
6: **end for**
7: evaluate $F(X^{(0)})$ using Eq. (17) at the master node
8: $\overline{F}^{(-1)} \leftarrow F(X^{(0)})$ at the master node
9: **for** $\mathsf{k} \leftarrow 0,\, 1,\, 2, \cdots$ **do**
10: $\quad$ **for** node $\alpha \leftarrow 1, \cdots, |\mathcal{A}|$ **do**
11: $\quad\quad s^{\alpha(\mathsf{k}+1)} \leftarrow \frac{\sqrt{4s^{\alpha(\mathsf{k})2}+1}+1}{2}$, $\lambda^{\alpha(\mathsf{k})} \leftarrow \frac{s^{\alpha(\mathsf{k})}-1}{s^{\alpha(\mathsf{k}+1)}}$
12: $\quad\quad Y^{\alpha(\mathsf{k})} \leftarrow X^{\alpha(\mathsf{k})} + \lambda^{\alpha(\mathsf{k})} \cdot \left(X^{\alpha(\mathsf{k})} - X^{\alpha(\mathsf{k}-1)}\right)$
13: $\quad$ **end for**
14: $\quad \overline{F}^{(\mathsf{k})} \leftarrow (1-\eta)\cdot \overline{F}^{(\mathsf{k}-1)} + \eta \cdot F(X^{(\mathsf{k})})$ at the master node
15: $\quad$ update $X^{(\mathsf{k}+\frac{1}{2})}$ and $X^{(\mathsf{k}+1)}$ using Algorithm 4
16: **end for**

C. Algorithm

Implementing Nesterov's method and the adaptive restart scheme, we obtain the AMM–PGO* method for distributed PGO (Algorithm 3), where "$*$" indicates the existence of a master node.

The outline of AMM–PGO* is as follows:

1) In lines 11, 12 of Algorithm 3, each node α computes $Y^{(\mathsf{k})}$ for Nesterov's acceleration that is related with $s^{\alpha(\mathsf{k})} \in [1,\, \infty)$ and $\lambda^{\alpha(\mathsf{k})} \in [0,\, 1)$.
2) In line 2 of Algorithm 4, each node α performs one inter-node communication round to retrieve $X^{\beta(\mathsf{k})}$ and $Y^{\beta(\mathsf{k})}$ from its neighbors $\beta \in \mathcal{N}^{\alpha}$.
3) In line 5 of Algorithm 3 and lines 6, 12, 20 of Algorithm 4, each node α performs one inter-node communication round to send $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ and $X^{\alpha(\mathsf{k}+1)}$ to the master node.
4) In lines 3 of Algorithm 4, each node α evaluates $\Gamma^{(\mathsf{k})}$, $\Pi^{\alpha(\mathsf{k})}$, $\nabla_{X^{\alpha}} F(X^{(\mathsf{k})})$, $\nabla_{X^{\alpha}} F(Y^{(\mathsf{k})})$ using $X^{\alpha(\mathsf{k})}$, $Y^{\alpha(\mathsf{k})}$, $X^{\beta(\mathsf{k})}$, $Y^{\beta(\mathsf{k})}$ where $\beta \in \mathcal{N}^{\alpha}$ are neighbors of node α.
5) In lines 8, 14 of Algorithm 3 and lines 8, 14, 22 of Algorithm 4, the master node evaluates $\overline{F}^{(\mathsf{k})}$, $F(X^{(\mathsf{k}+\frac{1}{2})})$, $F(X^{(\mathsf{k}+1)})$ that are used for adaptive restart.
6) In lines 9 to 23 of Algorithm 4, the master node performs adaptive restart to keep $F(X^{(\mathsf{k}+\frac{1}{2})}) \leq \overline{F}^{(\mathsf{k})}$ and $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$, which yields a nonincreasing sequence of $\overline{F}^{(\mathsf{k})}$ to guarantee the convergence.
7) In lines 5, 18 of Algorithm 4, note that $X^{\alpha(\mathsf{k}+1)}$ does not have to be a local optimal solution to Eq. (48).
8) In lines 24 to 26 of Algorithm 4, $F(X^{(\mathsf{k}+1)})$ is guaranteed to yield sufficient improvement over $\overline{F}^{(\mathsf{k})}$ compared to $F(X^{(\mathsf{k}+\frac{1}{2})})$.

In spite of acceleration, AMM–PGO* converges to first-order critical points as the following proposition states.

Proposition 6. *If Assumptions 1 to 4 hold, $\psi > 0$ and $\phi > 0$, then for a sequence of $\{X^{(\mathsf{k})}\}$ and $\{X^{(\mathsf{k}+\frac{1}{2})}\}$ generated by*

Algorithm 4 Updates for the AMM–PGO* Method

for node $\alpha \leftarrow 1, \cdots, |\mathcal{A}|$ **do**
  retrieve $X^{\beta(\mathsf{k})}$ and $Y^{\beta(\mathsf{k})}$ from $\beta \in \mathcal{N}_\alpha$
  evaluate $\Gamma^{\alpha(\mathsf{k})}$, $\Pi^{\alpha(\mathsf{k})}$, $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$, $\nabla_{X^\alpha} F(Y^{(\mathsf{k})})$
  $X^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X^\alpha \in \mathcal{X}^\alpha} H^\alpha(X^\alpha | Y^{(\mathsf{k})})$ using Algorithm 2
  $X^{\alpha(\mathsf{k}+1)} \leftarrow$ improve $\arg\min_{X^\alpha \in \mathcal{X}^\alpha} G^\alpha(X^\alpha | Y^{(\mathsf{k})})$ with $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ as the initial guess
  send $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ and $X^{\alpha(\mathsf{k}+1)}$ to the master node
end for
evaluate $F(X^{(\mathsf{k}+\frac{1}{2})})$ and $F(X^{(\mathsf{k}+1)})$ using Eq. (17) at the master node
if $F(X^{(\mathsf{k}+\frac{1}{2})}) > \overline{F}^{(\mathsf{k})} - \psi \cdot \|X^{(\mathsf{k}+\frac{1}{2})} - X^{(\mathsf{k})}\|^2$ **then**
  for node $\alpha \leftarrow 1, \cdots, |\mathcal{A}|$ **do**
    $X^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X^\alpha \in \mathcal{X}^\alpha} H^\alpha(X^\alpha | X^{(\mathsf{k})})$ using Algorithm 2
    send $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ to the master node
  end for
  evaluate $F(X^{(\mathsf{k}+\frac{1}{2})})$ using Eq. (17) at the master node
end if
if $F(X^{(\mathsf{k}+1)}) > \overline{F}^{(\mathsf{k})} - \psi \cdot \|X^{(\mathsf{k}+1)} - X^{(\mathsf{k})}\|^2$ **then**
  for node $\alpha \leftarrow 1, \cdots, |\mathcal{A}|$ **do**
    $X^{\alpha(\mathsf{k}+1)} \leftarrow$ improve $\arg\min_{X^\alpha \in \mathcal{X}^\alpha} G^\alpha(X^\alpha | X^{(\mathsf{k})})$ with $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ as the initial guess
    $s^{\alpha(\mathsf{k}+1)} \leftarrow \max\{\frac{1}{2} s^{\alpha(\mathsf{k}+1)}, 1\}$
    send $X^{\alpha(\mathsf{k}+1)}$ to the master node
  end for
  evaluate $F(X^{(\mathsf{k}+1)})$ using Eq. (17) at the master node
end if
if $\overline{F}^{(\mathsf{k})} - F(X^{(\mathsf{k}+1)}) < \phi \cdot \left(\overline{F}^{(\mathsf{k})} - F(X^{(\mathsf{k}+\frac{1}{2})})\right)$ **then**
  $X^{(\mathsf{k}+1)} \leftarrow X^{(\mathsf{k}+\frac{1}{2})}$ and $F(X^{(\mathsf{k}+1)}) \leftarrow F(X^{(\mathsf{k}+\frac{1}{2})})$
end if

Algorithm 3, we have

(a) $\overline{F}^{(\mathsf{k})}$ *is nonincreasing;*
(b) $F(X^{(\mathsf{k})}) \to F^\infty$ *and* $\overline{F}^{(k)} \to F^\infty$ *as* $\mathsf{k} \to \infty$*;*
(c) $\|X^{(\mathsf{k}+1)} - X^{(\mathsf{k})}\| \to 0$ *as* $\mathsf{k} \to \infty$ *if* $\xi > 0$ *and* $\zeta > 0$*;*
(d) $\|X^{(\mathsf{k}+\frac{1}{2})} - X^{(\mathsf{k})}\| \to 0$ *as* $\mathsf{k} \to \infty$ *if* $\zeta \geq \xi > 0$*;*
(e) if $\zeta \geq \xi > 0$*, then there exists* $\epsilon > 0$ *such that*

$$\min_{0 \leq \mathsf{k} < \mathsf{K}} \|\mathrm{grad}\, F(X^{(\mathsf{k}+\frac{1}{2})})\| \leq 2\sqrt{\frac{1}{\epsilon} \cdot \frac{F(X^{(0)}) - F^\infty}{\mathsf{K}+1}}$$

for any $\mathsf{K} \geq 0$*;*
(f) if $\zeta > \xi > 0$*, then* $\mathrm{grad}\, F(X^{(\mathsf{k})}) \to \mathbf{0}$ *and* $\mathrm{grad}\, F(X^{(\mathsf{k}+\frac{1}{2})}) \to \mathbf{0}$ *as* $\mathsf{k} \to \infty$.

Proof. See [43, Appendix F]. □

Remark 10. *If* $\eta = 1$ *in Eq. (59),* $F(X^{(\mathsf{k})}) = \overline{F}^{(\mathsf{k})}$*, and* $F(X^{(\mathsf{k})})$ *is also nonincreasing according to Proposition 6(a). While* $F(X^{(\mathsf{k})})$ *might fail to be nonincreasing, we still recommend to choose* $\eta \ll 1$ *that empirically yields fewer adaptive restarts and faster convergence for distributed PGO.*

Remark 11. $\psi > 0$ *and* $\phi > 0$ *in Algorithm 4 guarantee that* $F(X^{(\mathsf{k}+\frac{1}{2})})$ *and* $F(X^{(\mathsf{k}+1)})$ *yield sufficient improvement over* $\overline{F}^{(\mathsf{k})}$ *in terms of* $\|X^{(\mathsf{k}+\frac{1}{2})} - X^{(\mathsf{k})}\|$ *and* $\|X^{(\mathsf{k}+1)} - X^{(\mathsf{k})}\|$*, and are recommended to set close to zero to avoid unnecessary adaptive restarts and make full use of Nesterov's acceleration.*

IX. The Accelerated Majorization Minimization Method for Distributed Pose Graph Optimization without Master Node

The adaptive restart is essential for the convergence of accelerated MM methods. In AMM–PGO* (Algorithm 3), the adaptive restart scheme needs a master node to evaluate $F(X^{(\mathsf{k}+1)})$ and $\overline{F}^{(\mathsf{k})}$ and guarantee the convergence. On the other hand, if there is no master node, the adaptive restart scheme requires substantial amount of inter-node communication, making AMM–PGO* unscalable for large-scale distributed PGO. Recently, we developed an adaptive restart scheme for distributed PGO that does not require a master node while generating convergent iterates [32]. Nevertheless, the adaptive restart scheme in [32] is conservative and suffers from unnecessary restarts that hinder acceleration and yield slower convergence. Thus, we need to redesign the adaptive restart scheme for distributed PGO without master node to maximize the performance of accelerated MM methods. To address this issue, in Section IX-A, we develop a novel adaptive restart scheme that requires no master node and is fully decentralized. Then, in Section IX-B, we propose the accelerated MM method for distributed PGO without master that has provable convergence to first-order critical points. In particular, the resulting accelerated MM method, which needs no master node and is fully decentralized, empirically has no loss of computational efficiency in contrast to AMM–PGO* with master node; see Section X for more details.

A. Adaptive Restart

Recall that AMM–PGO*'s adaptive restart scheme guarantees the convergence by keeping $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$, where the master node only evaluates and compares $F(X^{(\mathsf{k}+1)})$ and $\overline{F}^{(\mathsf{k})}$. This suggests that if we could achieve $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$ without evaluating and comparing $F(X^{(\mathsf{k}+1)})$ and $\overline{F}^{(\mathsf{k})}$, no master node will be needed. We also note that if there is a sequence of $\{F^{\alpha(\mathsf{k})}\}$ and $\{\overline{F}^{\alpha(\mathsf{k})}\}$ for each node α such that

$$F(X^{(\mathsf{k})}) = \sum_{\alpha \in \mathcal{A}} F^{\alpha(\mathsf{k})}, \tag{60}$$

$$\overline{F}^{(\mathsf{k})} = \sum_{\alpha \in \mathcal{A}} \overline{F}^{\alpha(\mathsf{k})}, \tag{61}$$

$$F^{\alpha(\mathsf{k}+1)} \leq \overline{F}^{\alpha(\mathsf{k})}, \tag{62}$$

then $F(X^{(\mathsf{k}+1)}) = \sum_{\alpha \in \mathcal{A}} F^{\alpha(\mathsf{k}+1)} \leq \sum_{\alpha \in \mathcal{A}} \overline{F}^{\alpha(\mathsf{k})} = \overline{F}^{(\mathsf{k})}$. Therefore, the sequence above of $\{F^{\alpha(\mathsf{k})}\}$ and $\{\overline{F}^{\alpha(\mathsf{k})}\}$ is sufficient to keep $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$ despite that $F(X^{(\mathsf{k}+1)})$ and $\overline{F}^{(\mathsf{k})}$ are not explicitly evaluated and compared. More importantly, an adaptive restart scheme without master node can be developed with the sequence. In rest of this section,

we will construct $\{F^{\alpha(\mathsf{k})}\}$ and $\{\overline{F}^{\alpha(\mathsf{k})}\}$ satisfying Eqs. (60) to (62), which further results in the adaptive restart scheme for distributed PGO without a master node.

For notational simplicity, we define $\Delta G^\alpha(X|X^{(\mathsf{k})}) : \mathcal{X} \to \mathbb{R}$ related to the majorization gap of $G(X|X^{(\mathsf{k})})$ over $F(X)$:

$$\Delta G^\alpha(X|X^{(\mathsf{k})}) \triangleq -\frac{\xi}{2}\|X^\alpha - X^{\alpha(\mathsf{k})}\|^2 + \frac{1}{2}\sum_{\beta\in\mathcal{N}_-^\alpha}\sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\alpha\beta}}\Big(F_{ij}^{\alpha\beta}(X) - E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})})\Big) + \frac{1}{2}\sum_{\beta\in\mathcal{N}_+^\alpha}\sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\beta\alpha}}\Big(F_{ij}^{\beta\alpha}(X) - E_{ij}^{\beta\alpha}(X|X^{(\mathsf{k})})\Big) \quad (63)$$

where $F_{ij}^{\alpha\beta}(X)$, $F_{ij}^{\beta\alpha}(X)$, $E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})})$, $E_{ij}^{\beta\alpha}(X|X^{(\mathsf{k})})$ are given in Eqs. (19) and (31). From $\Delta G^\alpha(X|X^{(\mathsf{k})})$ in Eq. (63), we recursively define $F^{\alpha(\mathsf{k})}$, $\overline{F}^{\alpha(\mathsf{k})}$, $G^{\alpha(\mathsf{k})}$ according to:

1) If $\mathsf{k} = -1$, each node α initializes $F^{\alpha(-1)}$ and $\overline{F}^{\alpha(-1)}$ with

$$F^{\alpha(-1)} \triangleq \sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\alpha\alpha}} F_{ij}^{\alpha\alpha}(X^{(0)}) + \frac{1}{2}\sum_{\beta\in\mathcal{N}_-^\alpha}\sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\alpha\beta}} F_{ij}^{\alpha\beta}(X^{(0)}) + \frac{1}{2}\sum_{\beta\in\mathcal{N}_+^\alpha}\sum_{(i,j)\in\overrightarrow{\mathcal{E}}^{\beta\alpha}} F_{ij}^{\beta\alpha}(X^{(0)}), \quad (64)$$

$$\overline{F}^{\alpha(-1)} \triangleq F^{\alpha(-1)}. \quad (65)$$

2) If $\mathsf{k} \geq 0$, each node α recursively updates $G^{\alpha(\mathsf{k})}$, $F^{\alpha(\mathsf{k})}$ and $\overline{F}^{\alpha(\mathsf{k})}$ according to

$$G^{\alpha(\mathsf{k})} \triangleq G^\alpha(X^{\alpha(\mathsf{k})}|X^{(\mathsf{k}-1)}) + F^{\alpha(\mathsf{k}-1)}, \quad (66)$$

$$F^{\alpha(\mathsf{k})} \triangleq G^{\alpha(\mathsf{k})} + \Delta G^\alpha(X^{(\mathsf{k})}|X^{(\mathsf{k}-1)}), \quad (67)$$

$$\overline{F}^{\alpha(\mathsf{k})} \triangleq (1-\eta)\cdot\overline{F}^{\alpha(\mathsf{k}-1)} + \eta\cdot F^{\alpha(\mathsf{k})} \quad (68)$$

where $\eta \in (0,\, 1]$.

In [43, Appendix G], we have proved that such a sequence of $\{F^{\alpha(\mathsf{k})}\}$ and $\{\overline{F}^{\alpha(\mathsf{k})}\}$ satisfies Eqs. (60) to (62) as long as $G^{\alpha(\mathsf{k}+1)} \leq \overline{F}^{\alpha(\mathsf{k})}$, which yields the following proposition.

Proposition 7. *For $G^{\alpha(\mathsf{k})}$, $F^{\alpha(\mathsf{k})}$, $\overline{F}^{\alpha(\mathsf{k})}$ in Eqs. (64) to (68), we have*

(a) $F(X^{(\mathsf{k})}) = \sum_{\alpha\in\mathcal{A}} F^{\alpha(\mathsf{k})}$ where $F(X^{(\mathsf{k})})$ is given in Eq. (17);

(b) $\overline{F}^{(\mathsf{k})} = \sum_{\alpha\in\mathcal{A}} \overline{F}^{\alpha(\mathsf{k})}$ where $\overline{F}^{(\mathsf{k})}$ is given in Eq. (59);

(c) $F^{\alpha(\mathsf{k}+1)} \leq \overline{F}^{\alpha(\mathsf{k}+1)} \leq \overline{F}^{\alpha(\mathsf{k})}$ if $G^{\alpha(\mathsf{k}+1)} \leq \overline{F}^{\alpha(\mathsf{k})}$.

Proof. See [43, Appendix G]. □

It can be concluded from Propositions 7(a) and 7(b) that the resulting $\{F^{\alpha(\mathsf{k})}\}$ and $\{\overline{F}^{\alpha(\mathsf{k})}\}$ satisfies Eqs. (60) and (61), and Proposition 7(c) indicates that Eq. (62) holds if $G^{\alpha(\mathsf{k}+1)} \leq \overline{F}^{\alpha(\mathsf{k})}$. In [43, Appendix H], we have also proved that the following steps are sufficient to lead to $G^{\alpha(\mathsf{k}+1)} \leq \overline{F}^{\alpha(\mathsf{k})}$:

1) Update $X^{\alpha(\mathsf{k}+1)}$ by solving Eq. (56) at node α;
2) Compute $G^{\alpha(\mathsf{k}+1)}$ with Eq. (66) at node α;
3) If $G^{\alpha(\mathsf{k}+1)} > \overline{F}^{\alpha(\mathsf{k}+1)}$, update $X^{\alpha(\mathsf{k}+1)}$ again by solving Eq. (48) and reduce $s^{\alpha(\mathsf{k}+1))}$ at node $\alpha \in \mathcal{A}$.

Algorithm 5 The AMM–PGO$^\#$ Method

Input: An initial iterate $X^{(0)} \in \mathcal{X}$, and $\eta \in (0,\, 1]$, and $\zeta \geq \xi \geq 0$, and $\psi > 0$, and $\phi > 0$.
Output: A sequence of iterates $\{X^{(\mathsf{k})}\}$ and $\{X^{(\mathsf{k}+\frac{1}{2})}\}$.
for node $\alpha \leftarrow 1, \cdots, |\mathcal{A}|$ **do**
  $X^{\alpha(-1)} \leftarrow X^{\alpha(0)}$ and $s^{\alpha(0)} \leftarrow 1$
  retrieve $X^{\beta(-1)}$ and $X^{\beta(0)}$ from $\beta \in \mathcal{N}_\alpha$
  evaluate $F^{\alpha(-1)}$, $\overline{F}^{\alpha(-1)}$ using Eqs. (64) and (65)
  $G^{\alpha(0)} \leftarrow G^\alpha(X^{\alpha(0)}|X^{(-1)}) + F^{\alpha(-1)}$
end for
for $\mathsf{k} \leftarrow 0,\, 1,\, 2,\, \cdots$ **do**
  for node $\alpha \leftarrow 1, \cdots, |\mathcal{A}|$ **do**
    $s^{\alpha(\mathsf{k}+1)} \leftarrow \frac{\sqrt{4s^{\alpha(\mathsf{k})2}+1}+1}{2}$, $\lambda^{\alpha(\mathsf{k})} \leftarrow \frac{s^{\alpha(\mathsf{k})}-1}{s^{\alpha(\mathsf{k}+1)}}$
    $Y^{\alpha(\mathsf{k})} \leftarrow X^{\alpha(\mathsf{k})} + \lambda^{\alpha(\mathsf{k})}\cdot\big(X^{\alpha(\mathsf{k})} - X^{\alpha(\mathsf{k}-1)}\big)$
    retrieve $X^{\beta(\mathsf{k})}$ and $Y^{\beta(\mathsf{k})}$ from $\beta \in \mathcal{N}_\alpha$
    $F^{\alpha(\mathsf{k})} \leftarrow G^{\alpha(\mathsf{k})} + \Delta G^\alpha(X^{(\mathsf{k})}|X^{(\mathsf{k}-1)})$ using Eqs. (63) and (66)
    $\overline{F}^{\alpha(\mathsf{k})} \leftarrow (1-\eta)\cdot\overline{F}^{\alpha(\mathsf{k}-1)} + \eta\cdot F^{\alpha(\mathsf{k})}$
    update $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ and $X^{\alpha(\mathsf{k}+1)}$ using Algorithm 6
  end for
end for

Then, we not only obtain a sequence of $\{F^{\alpha(\mathsf{k})}\}$ and $\{\overline{F}^{\alpha(\mathsf{k})}\}$ satisfying Eqs. (60) to (62), but also an adaptive restart scheme using $G^{\alpha(\mathsf{k})}$, $F^{\alpha(\mathsf{k})}$, $\overline{F}^{\alpha(\mathsf{k})}$ to keep $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$. Note that $F(X^{(\mathsf{k}+1)})$ and $\overline{F}^{(\mathsf{k})}$ are neither evaluated nor compared. Instead, we evaluate and compare $G^{\alpha(\mathsf{k}+1)}$ and $\overline{F}^{\alpha(\mathsf{k})}$ independently at each node α. Moreover, according to Eqs. (19), (36) and (63), it is tedious but straightforward to show that $G^{\alpha(\mathsf{k})}$, $F^{\alpha(\mathsf{k})}$, $\overline{F}^{\alpha(\mathsf{k})}$ in Eqs. (64) to (68) can be computed with one inter-node communication round between node α and its neighbors $\beta \in \mathcal{N}_\alpha$. We emphasize that such an adaptive restart scheme differs from those in AMM–PGO* and [31], [50], [51] that have a master node to evaluate and compare $F(X^{(\mathsf{k}+1)})$ and $\overline{F}^{(\mathsf{k})}$. In contrast, the resulting adaptive restart scheme keeps $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$ but needs no master node, and thus, is well-suited for distributed PGO without master node.

Remark 12. *$F_{ij}^{\alpha\beta}(X) - E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})})$ and $F_{ji}^{\beta\alpha}(X) - E_{ji}^{\beta\alpha}(X|X^{(\mathsf{k})})$ are majorization gaps of inter-node measurements related to nodes α and β. According to Eq. (63), $\Delta G^\alpha(X^\alpha|X^{(\mathsf{k})})$ takes half of these majorization gaps of inter-node measurements for node α. Then, Eq. (67) uses $\Delta G^\alpha(X^\alpha|X^{(\mathsf{k})})$ to compute $F^{\alpha(\mathsf{k})}$ with majorization gaps $F_{ij}^{\alpha\beta}(X) - E_{ij}^{\alpha\beta}(X|X^{(\mathsf{k})})$ and $F_{ji}^{\beta\alpha}(X) - E_{ji}^{\beta\alpha}(X|X^{(\mathsf{k})})$ evenly redistributed between nodes α and β. However, $F_{ij}^{\alpha\beta}(X)$ and $F_{ji}^{\beta\alpha}(X)$ of inter-node measurements might fail to be evenly distributed between nodes α and β for $\mathsf{k} > 0$.*

B. Algorithm

With the adaptive restart scheme using $G^{\alpha(\mathsf{k})}$, $F^{\alpha(\mathsf{k})}$, $\overline{F}^{\alpha(\mathsf{k})}$ to keep $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$, we obtain the AMM–PGO$^\#$ method (Algorithm 5) for distributed PGO, where "#" indicates that no master node is needed.

Algorithm 6 Updates for the AMM–PGO$^{\#}$ Method

1: evaluate $\omega_{ij}^{\alpha\beta(\mathsf{k})}$ and $\omega_{ji}^{\beta\alpha(\mathsf{k})}$ using Eq. (23)
2: evaluate $\Gamma^{\alpha(\mathsf{k})}$, $\Pi^{\alpha(\mathsf{k})}$, $\nabla_{X^\alpha} F(X^{(\mathsf{k})})$, $\nabla_{X^\alpha} F(Y^{(\mathsf{k})})$ in Eqs. (36) and (40)
3: $X^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X^\alpha \in \mathcal{X}^\alpha} H^\alpha(X^\alpha | Y^{(\mathsf{k})})$ using Algorithm 2
4: $G^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow G^\alpha(X^{\alpha(\mathsf{k}+\frac{1}{2})} | X^{(\mathsf{k})}) + F^{\alpha(\mathsf{k})}$
5: $X^{\alpha(\mathsf{k}+1)} \leftarrow$ improve $\arg\min_{X^\alpha \in \mathcal{X}^\alpha} G^\alpha(X^\alpha | Y^{(\mathsf{k})})$ with $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ as the initial guess
6: $G^{\alpha(\mathsf{k}+1)} \leftarrow G^\alpha(X^{\alpha(\mathsf{k}+1)} | X^{(\mathsf{k})}) + F^{\alpha(\mathsf{k})}$
7: **if** $G^{\alpha(\mathsf{k}+\frac{1}{2})} > \overline{F}^{\alpha(\mathsf{k})} - \psi \cdot \|X^{\alpha(\mathsf{k}+\frac{1}{2})} - X^{\alpha(\mathsf{k})}\|^2$ **then**
8: $X^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow \arg\min_{X^\alpha \in \mathcal{X}^\alpha} H^\alpha(X^\alpha | X^{(\mathsf{k})})$ using Algorithm 2
9: $G^{\alpha(\mathsf{k}+\frac{1}{2})} \leftarrow G^\alpha(X^{\alpha(\mathsf{k}+\frac{1}{2})} | X^{(\mathsf{k})}) + F^{\alpha(\mathsf{k})}$
10: **end if**
11: **if** $G^{\alpha(\mathsf{k}+1)} > \overline{F}^{\alpha(\mathsf{k})}$ **then**
12: $X^{\alpha(\mathsf{k}+1)} \leftarrow$ improve $\arg\min_{X^\alpha \in \mathcal{X}^\alpha} G^\alpha(X^\alpha | X^{(\mathsf{k})})$ with $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ as the initial guess
13: $G^{\alpha(\mathsf{k}+1)} \leftarrow G^\alpha(X^{\alpha(\mathsf{k}+1)} | X^{(\mathsf{k})}) + F^{\alpha(\mathsf{k})}$
14: $s^{\alpha(\mathsf{k}+1)} \leftarrow \max\{\frac{1}{2} s^{\alpha(\mathsf{k}+1)}, 1\}$
15: **end if**
16: **if** $\overline{F}^{\alpha(\mathsf{k})} - G^{\alpha(\mathsf{k}+1)} < \phi \cdot \left(\overline{F}^{\alpha(\mathsf{k})} - G^{\alpha(\mathsf{k}+\frac{1}{2})}\right)$ **then**
17: $X^{\alpha(\mathsf{k}+1)} \leftarrow X^{\alpha(\mathsf{k}+\frac{1}{2})}$ and $G^{\alpha(\mathsf{k}+1)} \leftarrow G^{\alpha(\mathsf{k}+\frac{1}{2})}$
18: **end if**

The outline of AMM–PGO$^{\#}$ is similar to AMM–PGO* and the key difference is the adaptive restart scheme:

1) In lines 5, 13 of Algorithm 5, each node α performs one inter-node communication round to retrieve $X^{\beta(\mathsf{k})}$ and $Y^{\beta(\mathsf{k})}$ from its neighbors $\beta \in \mathcal{N}^\alpha$. We remark that no other inter-node communication is required.
2) In lines 6, 14, 15 of Algorithm 5 and lines 4, 6, 9, 13 of Algorithm 6, each node α evaluates $F^{\alpha(\mathsf{k})}$, $\overline{F}^{\alpha(\mathsf{k})}$, $G^{\alpha(\mathsf{k}+\frac{1}{2})}$, $G^{\alpha(\mathsf{k}+1)}$ that are used for adaptive restart. Note that $X^{\beta(\mathsf{k})}$ and $X^{\beta(\mathsf{k}-1)}$ from node α's neighbors $\beta \in \mathcal{N}^\alpha$ are needed.
3) In lines 7 to 15 of Algorithm 6, each node α performs independent adaptive restart such that $G^{\alpha(\mathsf{k}+\frac{1}{2})} \leq \overline{F}^{\alpha(\mathsf{k})}$ and $G^{\alpha(\mathsf{k}+1)} \leq \overline{F}^{\alpha(\mathsf{k})}$, which also results in $F(X^{(\mathsf{k}+1)}) \leq \overline{F}^{(\mathsf{k})}$ and a nonincreasing sequence of $\overline{F}^{(\mathsf{k})}$ for distributed PGO without master node.
4) In lines 16 to 18 of Algorithm 6, $G^{\alpha(\mathsf{k}+1)}$ is guaranteed to yield sufficient improvement over $\overline{F}^{\alpha(\mathsf{k})}$ compared to $G^{\alpha(\mathsf{k}+\frac{1}{2})}$.

Furthermore, AMM–PGO$^{\#}$ converges to first-order critical points as the following propositions states.

Proposition 8. *If Assumptions 1 to 3 hold, $\psi > 0$ and $\phi > 0$, then for a sequence of $\{X^{(\mathsf{k})}\}$ and $\{X^{(\mathsf{k}+\frac{1}{2})}\}$ generated by Algorithm 5, we have*

(a) $\overline{F}^{(\mathsf{k})}$ is nonincreasing;
(b) $F(X^{(\mathsf{k})}) \to F^\infty$ and $\overline{F}^{(k)} \to F^\infty$ as $\mathsf{k} \to \infty$;
(c) $\|X^{(\mathsf{k}+1)} - X^{(\mathsf{k})}\| \to 0$ as $\mathsf{k} \to \infty$ if $\zeta > \xi > 0$;
(d) $\|X^{(\mathsf{k}+\frac{1}{2})} - X^{(\mathsf{k})}\| \to 0$ as $\mathsf{k} \to \infty$ if $\zeta > \xi > 0$;
(e) if $\zeta \geq \xi > 0$, then there exists $\epsilon > 0$ such that

$$\min_{0 \leq \mathsf{k} < \mathsf{K}} \|\mathrm{grad}\, F(X^{(\mathsf{k}+\frac{1}{2})})\| \leq 2\sqrt{\frac{1}{\epsilon} \cdot \frac{F(X^{(0)}) - F^\infty}{\mathsf{K}+1}}$$

for any $\mathsf{K} \geq 0$;
(f) if $\zeta \geq \xi > 0$, then $\mathrm{grad}\, F(X^{(\mathsf{k})}) \to \mathbf{0}$ and $\mathrm{grad}\, F(X^{(\mathsf{k}+\frac{1}{2})}) \to \mathbf{0}$ as $\mathsf{k} \to \infty$.

Proof. See [43, Appendix H]. □

In spite of no master node, Proposition 8 indicates that AMM–PGO$^{\#}$ has provable convergence as long as each node $\alpha \in \mathcal{A}$ can communicate with its neighbors $\beta \in \mathcal{N}^\alpha$. Thus, AMM–PGO$^{\#}$ eliminates the bottleneck of communication for distributed PGO without master node. In addition, AMM–PGO$^{\#}$ also reduces unnecessary adaptive restarts compared to [32], and thus makes better of Nesterov's acceleration and has faster convergence.

X. Experiments

In this section, we evaluate the performance of our MM methods (MM–PGO, AMM–PGO* and AMM–PGO$^{\#}$) for distributed PGO on the simulated Cube datasets and a number of 2D and 3D SLAM benchmark datasets [8]. In terms of MM–PGO, AMM–PGO* and AMM–PGO$^{\#}$, η, ξ, ζ, ψ and ϕ in Algorithms 1, 3 and 5 are 5×10^{-4}, 1×10^{-10}, 1.5×10^{-10}, 1×10^{-10} and 1×10^{-6}, respectively, for all the experiments. In addition, MM–PGO, AMM–PGO* and AMM–PGO$^{\#}$ can take at most one iteration when solving Eqs. (48) and (56) to improve the estimates. All the experiments have been performed on a laptop with an Intel Xeon(R) CPU E3-1535M v6 and 64GB of RAM running Ubuntu 20.04.

A. Cube Datasets

In this section, we test and evaluate our MM methods for distributed PGO on 20 simulated Cube datasets (see Fig. 2) with 5, 10 and 50 robots. In the experiment, a simulated Cube dataset has $12 \times 12 \times 12$ cube grids with 1 m side length, a path of 3600 poses along the rectilinear edge of the cube grid, odometric measurements between all the pairs of sequential poses, and loop-closure measurements between nearby but non-sequential poses that are randomly available with a probability of 0.1. We generate the odometric and loop-closure measurements according to the noise models in [8] with an expected translational RMSE of 0.02 m and an expected angular RMSE of 0.02π rad. The centralized chordal

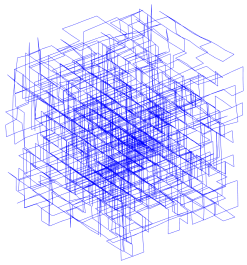
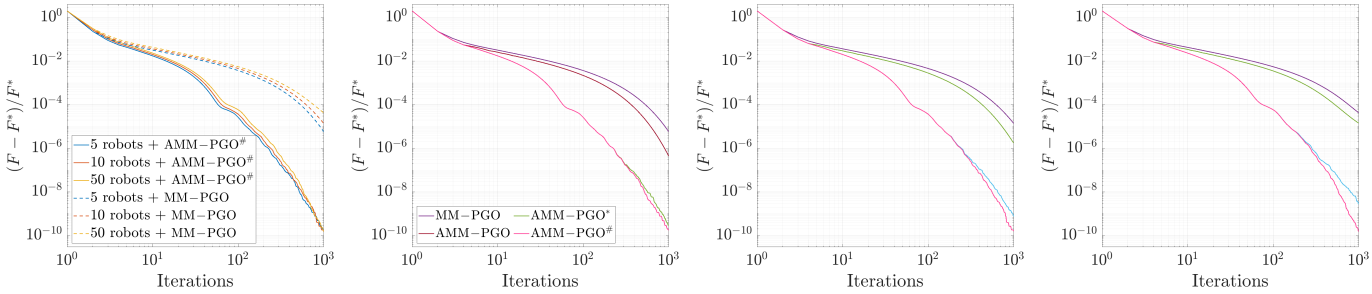
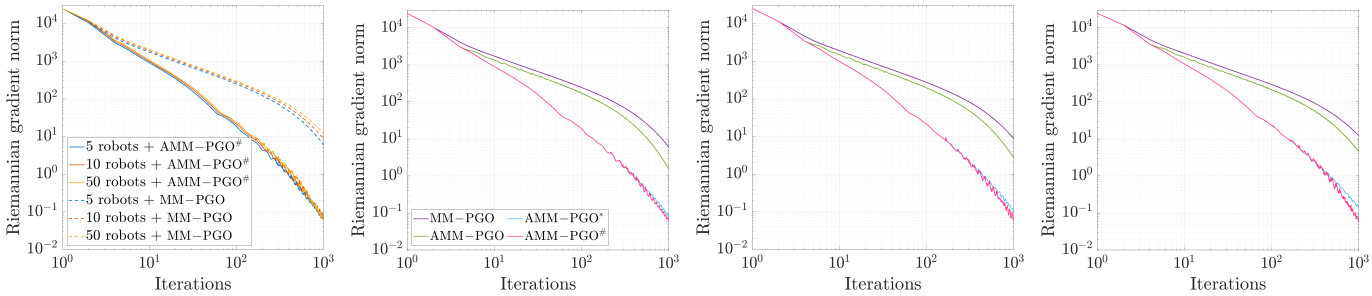


Fig. 2: A Cube dataset has $12\times12\times12$ grids of side length of 1 m, 3600 poses, loop closure probability of 0.1, an translational RSME of 0.02 m and an angular RSME of 0.02π rad.



(a) AMM–PGO$^{\#}$ vs. MM–PGO (b) 5 robots (c) 10 robots (d) 50 robots

Fig. 3: The relative suboptimality gaps of MM–PGO, AMM–PGO*, AMM–PGO$^{\#}$ and AMM–PGO [32] for distributed PGO with the **trivial loss kernel** on 5, 10 and 50 robots. The results are averaged over 20 Monte Carlo runs.



(a) AMM–PGO$^{\#}$ vs. MM–PGO (b) 5 robots (c) 10 robots (d) 50 robots

Fig. 4: The Riemannian gradient norms of MM–PGO, AMM–PGO*, AMM–PGO$^{\#}$ and AMM–PGO [32] for distributed PGO with the **trivial loss kernel** on 5, 10 and 50 robots. The results are averaged over 20 Monte Carlo runs.

initialization [40] is implemented such that distributed PGO with different number of robots have the same initial estimate. The maximum number of iterations is 1000.

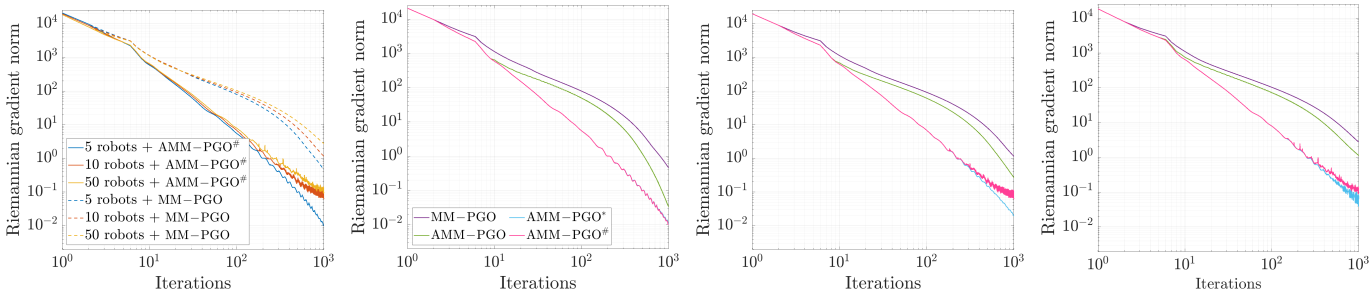
We evaluate the convergence of MM–PGO, AMM–PGO* and AMM–PGO$^{\#}$ in terms of the relative suboptimality gap and Riemannian gradient norm. For reference, we also make comparisons against AMM–PGO [32]. Note that AMM–PGO is the original accelerated MM method for distributed PGO whose adaptive restart scheme is conservative and might prohibit Nesterov's acceleration.

Relative Suboptimality Gap. We implement the certifiably-correct SE−Sync [8] to get the globally optimal objective value F^* of distributed PGO with the trivial loss kernel (Example 1), making it possible to compute the relative suboptimality gap $(F-F^*)/F^*$ where F is the objective value for each iteration. The results are in Fig. 3.

Riemannian Gradient Norm. We also compute the Riemannian gradient norm of distributed PGO with the trivial, Huber and Welsch loss kernels in Examples 1 to 3 for evaluation. Note that it is difficult to find globally optimal solutions to distributed PGO if Huber and Welsch loss kernels are used. The results are in Figs. 4 to 6.

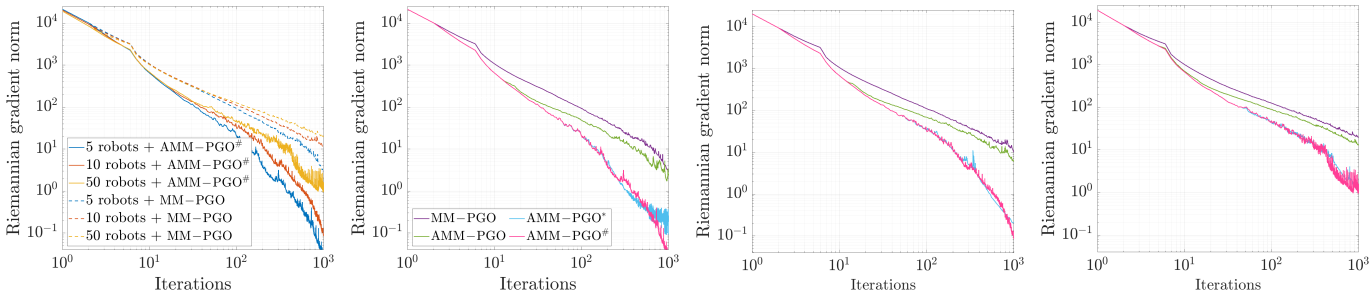
In Figs. 3 to 6, it can be seen that MM–PGO, AMM–PGO*, AMM–PGO$^{\#}$ and AMM–PGO have a faster convergence if the number of robots (nodes) decreases. This is expected since $G(X|X^{(k)})$ and $H(X|X^{(k)})$ in Eqs. (33) and (37) result in tighter approximations for distributed PGO with fewer robots (nodes). In addition, Figs. 4 to 6 suggest that the convergence rate of MM–PGO, AMM–PGO*, AMM–PGO$^{\#}$ and AMM–PGO also relies on the type of loss kernels. Nevertheless, AMM–PGO*, AMM–PGO$^{\#}$ and AMM–PGO accelerated by Nesterov's method outperform unaccelerated MM–PGO by a large margin for any number of robots and any types of loss kernels, which means that Nesterov's method improves the convergence of distributed PGO. In particular, Figs. 3(a), 4(a), 5(a), 6(a) indicate that AMM–PGO$^{\#}$ with 50 robot still converges faster than MM–PGO with 5 robots despite that the later has a much smaller number of robots. Therefore, we conclude that Nesterov's method accelerates the convergence of distributed PGO.

We emphasize the convergence comparisons of Nesterov's accelerated AMM–PGO*, AMM–PGO$^{\#}$ and AMM–PGO that merely differ from each other by the adaptive restart schemes—AMM–PGO* has an additional master node to aggregate information from all the robots (nodes), whereas AMM–PGO$^{\#}$ and AMM–PGO are restricted to one inter-node communication round per iteration among neighboring robots (nodes). Notwithstanding limited local communication, as is shown in Figs. 3, 5 and 6, AMM–PGO$^{\#}$ has a convergence rate comparable to that of AMM–PGO* using a master node while being significantly faster than AMM–PGO. In particular, AMM–PGO$^{\#}$ reduces adaptive restarts by 80% to 95% compared to AMM–PGO on the Cube datasets, and thus, is expected to make better use of Nesterov's acceleration. Since AMM–PGO$^{\#}$ and AMM–PGO differ in the adaptive restart schemes, we attribute the faster convergence of AMM–PGO$^{\#}$ to its redesigned adaptive restart scheme. These results suggest that AMM–PGO$^{\#}$ is advantageous over other methods for very large-scale distributed PGO where computational and communicational efficiency are equally important.



(a) AMM–PGO$^{\#}$ vs. MM–PGO (b) 5 robots (c) 10 robots (d) 50 robots

Fig. 5: The Riemannian gradient norms of MM–PGO, AMM–PGO*, AMM–PGO$^{\#}$ and AMM–PGO [32] for distributed PGO with the **Huber loss kernel** on 5, 10 and 50 robots. The results are averaged over 20 Monte Carlo runs.



(a) AMM–PGO$^{\#}$ vs. MM–PGO (b) 5 robots (c) 10 robots (d) 50 robots

Fig. 6: The Riemannian gradient norms of MM–PGO, AMM–PGO*, AMM–PGO$^{\#}$ and AMM–PGO [32] for distributed PGO with the **Welsch loss kernel** on 5, 10 and 50 robots. The results are averaged over 20 Monte Carlo runs.

B. Benchmark Datasets

In this section, we evaluate our MM methods (MM–PGO, AMM–PGO* and AMM–PGO$^{\#}$) for distributed PGO on a number of 2D and 3D SLAM benchmark datasets [8] (see [43, Appendix L]). We use the trivial loss kernel and there are no outliers such that the globally optimal solution can be exactly computed with SE−Sync [8]. For each dataset, we also make comparisons against SE−Sync [8], distributed Gauss-Seidel (DGS) [36] and the Riemannian block coordinate descent (RBCD) [37] method[2], all of which are the state-of-the-art algorithms for centralized and distributed PGO. The SE−Sync and DGS methods use the recommended settings in [8], [36]. We implement two Nesterov's accelerated variants of RBCD [37], i.e., one with greedy selection rule and adaptive restart (RBCD++*) and the other with uniform selection rule and fixed restart (RBCD++$^{\#}$)[3]. As mentioned before, AMM–PGO* and AMM–PGO$^{\#}$ can take at most one iteration when updating $X^{\alpha(k+1)}$ using Eqs. (48) and (56), which is similar to RBCD++* and RBCD++$^{\#}$. An overview of the aforementioned methods is given in Table I.

Number of Iterations. First, we examine the convergence of MM–PGO, AMM–PGO*, AMM–PGO$^{\#}$, DGS [36], RBCD++* [37] and RBCD++$^{\#}$ [37] w.r.t. the number of iterations. The distributed PGO has 10 robots and all the methods are initialized with distributed Nesterov's accelerated chordal initialization [32].

TABLE I: An overview of the state-of-the-art algorithms for distributed and centralized PGO. Note that AMM–PGO* and RBCD++* require a master node for distributed PGO, and AMM–PGO$^{\#}$ is the only accelerated method with provable convergence for distributed PGO without master node.

Method	Distributed	Accelerated	Masterless	Converged
SE−Sync [8]	×	N/A	N/A	YES
DGS [36]	YES	×	YES	×
RBCD++* [37]	YES	YES	×	YES
RBCD++$^{\#}$ [37]	YES	YES	YES	×
MM–PGO	YES	×	YES	YES
AMM–PGO*	YES	YES	×	YES
AMM–PGO$^{\#}$	YES	YES	YES	YES

The objective values of each method with 100, 250 and 1000 iterations are reported in Table II and the reconstruction results using AMM–PGO$^{\#}$ are shown in Figs. 7 and 8. For almost all the benchmark datasets, AMM–PGO* and AMM–PGO$^{\#}$ outperform the other methods (MM–PGO, DGS, RBCD++* and RBCD++$^{\#}$). While RBCD++* and RBCD++$^{\#}$ have similar performances in four relatively easy datasets—CSAIL, sphere, torus and grid—AMM–PGO* and AMM–PGO$^{\#}$ achieve much better results in the other more challenging datasets in particular if there are no more than 250 iterations. As discussed later, AMM–PGO* and AMM–PGO$^{\#}$ have faster convergence to more accurate estimates without any extra computation and communication in contrast to RBCD++* and RBCD++$^{\#}$. Last but not the least, Table II demonstrates that the accelerated AMM–PGO* and AMM–PGO$^{\#}$ converge

[2]RBCD [37] solves the lifted problem which usually results in slightly smaller objective values than the original problem.

[3]In the experiments, we run RBCD++$^{\#}$ [37] with fixed restart frequencies of 30, 50 and 100 iterations for each dataset and report the best results.

TABLE II: Results of distributed PGO on 2D and 3D SLAM benchmark datasets (see [43, Appendix L]). The distributed PGO has 10 robots and is initialized with distributed Nesterov's accelerated chordal initialization [32]. We report the objective values of each method with 100, 250 and 1000 iterations. $F^{(\mathsf{k})}$ and F^* are the objective value at iteration k and globally optimal objective value, respectively. The best results are colored in red and the second best in blue if no methods tie for the best.

Dataset	$F^{(0)}$	F^*	k	$F^{(\mathsf{k})}$					
				Methods w/ Master Node		Methods w/o Master Node			
				AMM–PGO*	RBCD++* [37]	MM–PGO	AMM–PGO#	DGS [36]	RBCD++# [37]
2D SLAM Benchmark Datasets									
ais2klinik	3.8375×10^2	1.8850×10^2	100	2.0372×10^2	2.1079×10^2	2.1914×10^2	2.0371×10^2	8.4701×10^2	2.1715×10^2
			250	1.9447×10^2	2.0077×10^2	2.1371×10^2	1.9446×10^2	9.1623×10^1	2.1084×10^2
			1000	1.8973×10^2	1.9074×10^2	2.0585×10^2	1.8936×10^2	3.8968×10^2	2.0253×10^2
city	7.0404×10^2	6.3862×10^2	100	6.4327×10^2	6.5138×10^2	6.5061×10^2	6.4327×10^2	7.7745×10^2	6.5396×10^2
			250	6.3899×10^2	6.4732×10^2	6.4850×10^2	6.3899×10^2	7.0063×10^2	6.5122×10^2
			1000	6.3862×10^2	6.3935×10^2	6.4461×10^2	6.3863×10^2	6.5583×10^2	6.4768×10^2
CSAIL	3.1719×10^1	3.1704×10^1	100	3.1704×10^1	3.1704×10^1	3.1706×10^1	3.1704×10^1	3.2479×10^1	3.1705×10^1
			250	3.1704×10^1	3.1704×10^1	3.1706×10^1	3.1704×10^1	3.1792×10^1	3.1704×10^1
			1000	3.1704×10^1	3.1704×10^1	3.1705×10^1	3.1704×10^1	3.1712×10^1	3.1704×10^1
M3500	2.2311×10^2	1.9386×10^2	100	1.9446×10^2	1.9511×10^2	1.9560×10^2	1.9447×10^2	1.9557×10^2	1.9551×10^2
			250	1.9414×10^2	1.9443×10^2	1.9516×10^2	1.9414×10^2	1.9445×10^2	1.9511×10^2
			1000	1.9388×10^2	1.9392×10^2	1.9461×10^2	1.9388×10^2	1.9415×10^2	1.9455×10^2
intel	5.3269×10^1	5.2348×10^1	100	5.2397×10^1	5.2496×10^1	5.2517×10^1	5.2397×10^1	5.2541×10^1	5.2526×10^1
			250	5.2352×10^1	5.2415×10^1	5.2483×10^1	5.2351×10^1	5.2441×10^1	5.2489×10^1
			1000	5.2348×10^1	5.2349×10^1	5.2421×10^1	5.2348×10^1	5.2381×10^1	5.2425×10^1
MITb	8.8430×10^1	6.1154×10^1	100	6.1331×10^1	6.1518×10^1	6.3657×10^1	6.1330×10^1	9.5460×10^1	6.1997×10^1
			250	6.1157×10^1	6.1187×10^1	6.2335×10^1	6.1165×10^1	7.8273×10^1	6.1599×10^1
			1000	6.1154×10^1	6.1154×10^1	6.1454×10^1	6.1154×10^1	7.2450×10^1	6.1209×10^1
3D SLAM Benchmark Datasets									
sphere	1.9704×10^3	1.6870×10^3	100	1.6870×10^3	1.6870×10^3	1.6901×10^3	1.6870×10^3	1.6875×10^3	1.6870×10^3
			250	1.6870×10^3	1.6870×10^3	1.6874×10^3	1.6870×10^3	1.6872×10^3	1.6870×10^3
			1000	1.6870×10^3	1.6870×10^3	1.6870×10^3	1.6870×10^3	1.6872×10^3	1.6870×10^3
torus	2.4654×10^4	2.4227×10^4	100	2.4227×10^4	2.4227×10^4	2.4234×10^4	2.4227×10^4	2.4248×10^4	2.4227×10^4
			250	2.4227×10^4	2.4227×10^4	2.4227×10^4	2.4227×10^4	2.4243×10^4	2.4227×10^4
			1000	2.4227×10^4	2.4227×10^4	2.4227×10^4	2.4227×10^4	2.4236×10^4	2.4227×10^4
grid	2.8218×10^5	8.4319×10^4	100	8.4323×10^4	8.4320×10^4	1.0830×10^5	8.4399×10^4	1.4847×10^5	8.4920×10^4
			250	8.4319×10^4	8.4319×10^4	8.6054×10^4	8.4321×10^4	1.4066×10^5	8.4319×10^4
			1000	8.4319×10^4	8.4319×10^4	8.4319×10^4	8.4319×10^4	1.4654×10^5	8.4319×10^4
garage	1.5470×10^0	1.2625×10^0	100	1.3105×10^0	1.3282×10^0	1.3396×10^0	1.3105×10^0	1.3170×10^0	1.3364×10^0
			250	1.2872×10^0	1.3094×10^0	1.3288×10^0	1.2872×10^0	1.2867×10^0	1.3276×10^0
			1000	1.2636×10^0	1.2681×10^0	1.3145×10^0	1.2636×10^0	1.2722×10^0	1.3124×10^0
cubicle	8.3514×10^2	7.1713×10^2	100	7.1812×10^2	7.2048×10^2	7.2300×10^2	7.1812×10^2	7.3185×10^2	7.2210×10^2
			250	7.1714×10^2	7.1794×10^2	7.2082×10^2	7.1715×10^2	7.2308×10^2	7.2081×10^2
			1000	7.1713×10^2	7.1713×10^2	7.2082×10^2	7.1713×10^2	7.2044×10^2	7.1845×10^2
rim	8.1406×10^4	5.4609×10^3	100	5.5044×10^3	5.7184×10^3	5.8138×10^3	5.5044×10^3	6.1840×10^3	5.7810×10^3
			250	5.4648×10^3	5.5050×10^3	5.7197×10^3	5.4648×10^3	6.1184×10^3	5.7195×10^3
			1000	5.4609×10^3	5.4617×10^3	5.5509×10^3	5.4609×10^3	6.0258×10^3	5.5373×10^3

significantly faster than the unaccelerated MM–PGO, which further validates the usefulness of Nesterov's method.

We also compute the performance profiles [52] based on the number of iterations. Given a tolerance $\Delta \in (0, 1]$, the objective value threshold $F_\Delta(p)$ of a PGO problem p is

$$F_\Delta(p) = F^* + \Delta \cdot \left(F^{(0)} - F^*\right) \tag{69}$$

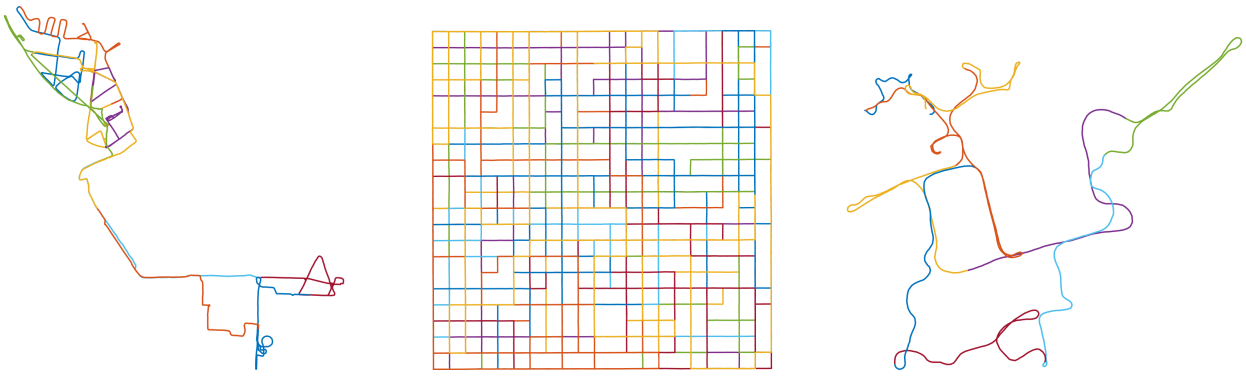
where $F^{(0)}$ and F^* are the initial and globally optimal objective values, respectively. Let $I_\Delta(p)$ denote the minimum number of iterations that a PGO method takes to reduce the objective value to $F_\Delta(p)$, i.e.,

$$I_\Delta(p) \triangleq \min_{\mathsf{k}} \left\{\mathsf{k} \geq 0 | F^{(\mathsf{k})} \leq F_\Delta(p)\right\}$$

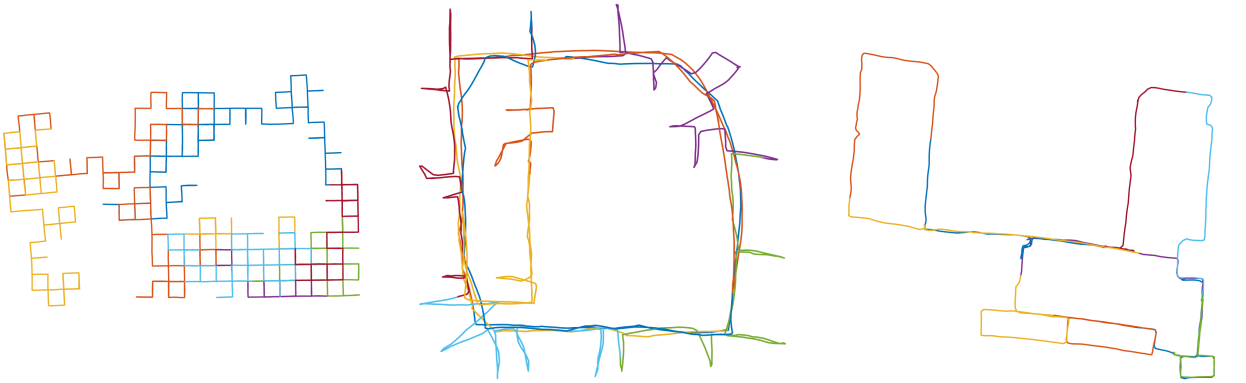
where $F^{(\mathsf{k})}$ is the objective value at iteration k. Then, for a problem set $\mathcal{P}$, the performance profiles of a PGO method is the percentage of problems solved w.r.t. the number of iterations k:

$$\underset{\text{at iteration } \mathsf{k}}{\text{percentage of problems solved}} \triangleq \frac{\left|\{p \in \mathcal{P} | I_\Delta(p) \leq \mathsf{k}\}\right|}{|\mathcal{P}|}.$$

The performance profiles based on the number of iterations over a variety of 2D and 3D SLAM benchmark datasets (see [43, Appendix L]) are shown in Fig. 9. The tolerances evaluated are $\Delta = 1 \times 10^{-2}$, 5×10^{-3}, 1×10^{-3} and 1×10^{-4}. We report the performance of MM–PGO, AMM–PGO*, AMM–PGO#, DGS [36], RBCD++* [37] and RBCD++# [37] for distributed PGO with 10 robots (nodes). As expected, AMM–PGO* and AMM–PGO* dominates the other methods

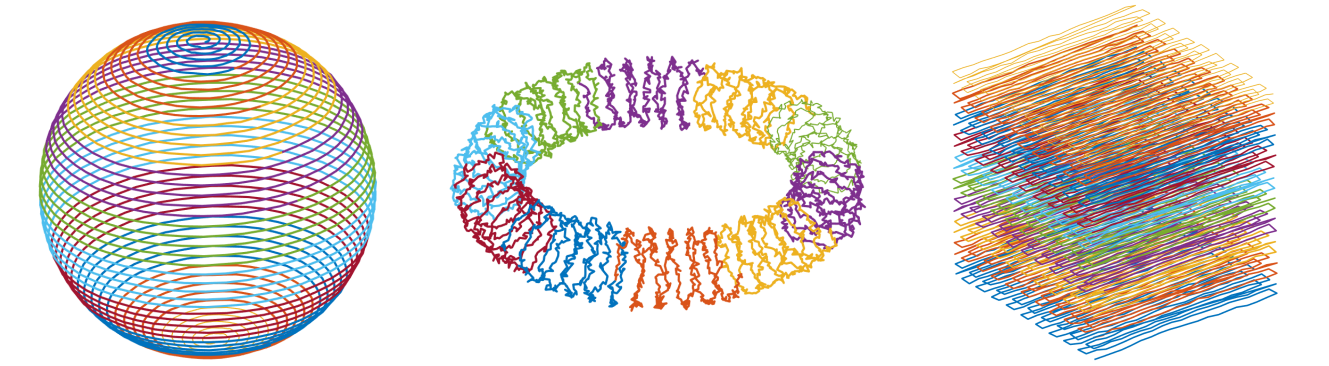


(a) ais2klinik (b) city (c) CSAIL

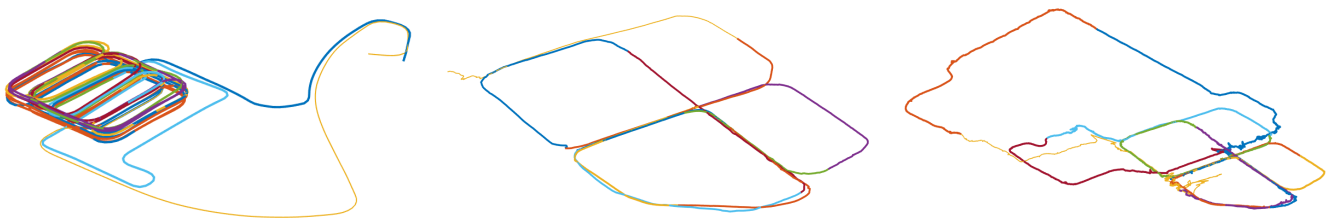


(d) M3500 (e) intel (f) MITb

Fig. 7: AMM–PGO$^{\#}$ results on the 2D SLAM benchmark datasets where the different colors denote the odometries of different robots. The distributed PGO has 10 robots and is initialized with the distributed Nesterov's accelerated chordal initialization [32]. The number of iterations is 1000.

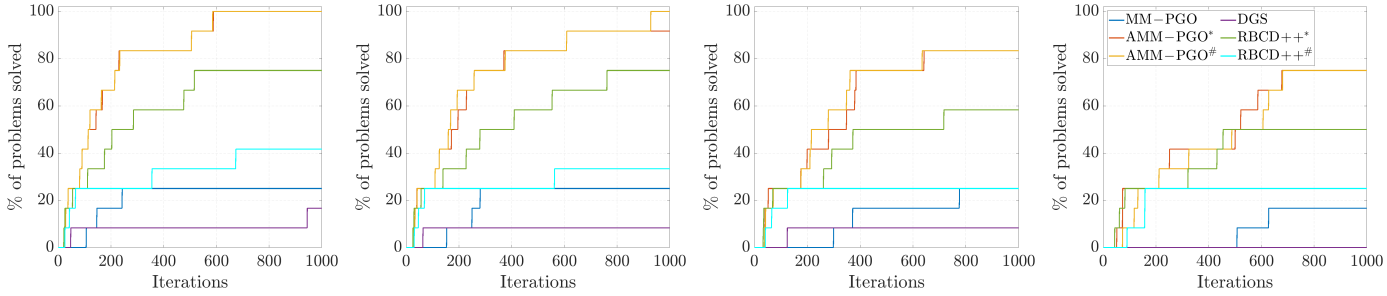


(a) sphere (b) torus (c) grid



(d) garage (e) cubicle (f) rim

Fig. 8: AMM–PGO$^{\#}$ results on the 3D SLAM benchmark datasets where the different colors denote the odometries of different robots. The distributed PGO has 10 robots and is initialized with the distributed Nesterov's accelerated chordal initialization [32]. The number of iterations is 1000.



(a) $\Delta = 1 \times 10^{-2}$ (b) $\Delta = 5 \times 10^{-3}$ (c) $\Delta = 1 \times 10^{-3}$ (d) $\Delta = 1 \times 10^{-4}$

Fig. 9: Performance profiles for MM–PGO, AMM–PGO*, AMM–PGO$^\#$, DGS [36], RBCD++* [37] and RBCD++$^\#$ [37] on 2D and 3D SLAM Benchmark datasets (see [43, Appendix L]). The performance is based on the number of iterations k and the evaluation tolerances are $\Delta = 1 \times 10^{-2}, 5 \times 10^{-3}, 1 \times 10^{-3}, 1 \times 10^{-4}$. The distributed PGO has 10 robots (nodes) and is initialized with distributed Nesterov's accelerated chordal initialization [32]. Note that AMM–PGO* and RBCD++* require a master node, whereas MM–PGO, AMM–PGO$^\#$, DGS and RBCD++$^\#$ do not.

(MM–PGO, DGS, RBCD++* and RBCD++$^\#$) in terms of the convergence for all the tolerances Δ, which means that both of them are better choices for distributed PGO.

In Table II and Fig. 9, we emphasize that AMM–PGO$^\#$ requiring no master node achieves comparable performance to that of AMM–PGO* using a master node, and is a lot better than all the other methods with a master node (RBCD++*) and without (MM–PGO, DGS and RBCD++$^\#$). Even though RBCD++* and RBCD++$^\#$ are similarly accelerated with Nesterov's method, we remark that RBCD++$^\#$ without a master node suffers a great performance drop compared to RBCD++*, and more importantly, RBCD++$^\#$ has no convergence guarantees to first-order critical points. These results reverify that AMM–PGO$^\#$ is more suitable for very large-scale distributed PGO with limited local communication.

Note that all of MM–PGO, AMM–PGO*, AMM–PGO$^\#$, DGS [36], RBCD++* [37] and RBCD++$^\#$ [37] have to exchange poses of inter-node measurements with the neighbors, and thus, need almost the same amount of communication per iteration. However, Fig. 9 indicates that AMM–PGO* and AMM–PGO$^\#$ have much faster convergence in terms of the number of iterations, which also means less communication for the same level of accuracy. In addition, RBCD++* and RBCD++$^\#$ have to keep part of the nodes in idle during optimization and rely on red-black coloring for block aggregation and random sampling for block selection, which induce additional computation and communication. In contrast, neither AMM–PGO* nor AMM–PGO$^\#$ has any extra practical restrictions except Assumptions 1 to 4.

Optimization Time. We evaluate the optimization time of AMM–PGO* and AMM–PGO$^\#$ with different numbers of robots (nodes) against the centralized baseline SE−Sync [8]. To improve the time efficiency of our methods, $X^{\alpha(\mathsf{k}+1)}$ in Eqs. (48) and (56) uses the same rotation as $X^{\alpha(\mathsf{k}+\frac{1}{2})}$ in Eqs. (47) and (55) and merely updates the translation. Due to the different numbers of robots (nodes), the centralized chordal initialization [40] is used for all the runs.

Similar to the number of iterations, we use the performance profiles to evaluate AMM–PGO* and AMM–PGO$^\#$ in terms of the optimization time. Recall from Eq. (69) the objective value threshold $F_\Delta(p)$ where p is the PGO problem and $\Delta \in (0, 1]$ is the tolerance. Since the average optimization time per node is directly related with the speedup, we measure the efficiency of a distributed PGO method with N nodes by computing the average optimization time $T_\Delta(p, N)$ that each node takes to reduce the objective value to $F_\Delta(p)$:

$$T_\Delta(p, N) = \frac{T_\Delta(p)}{N} \tag{70}$$

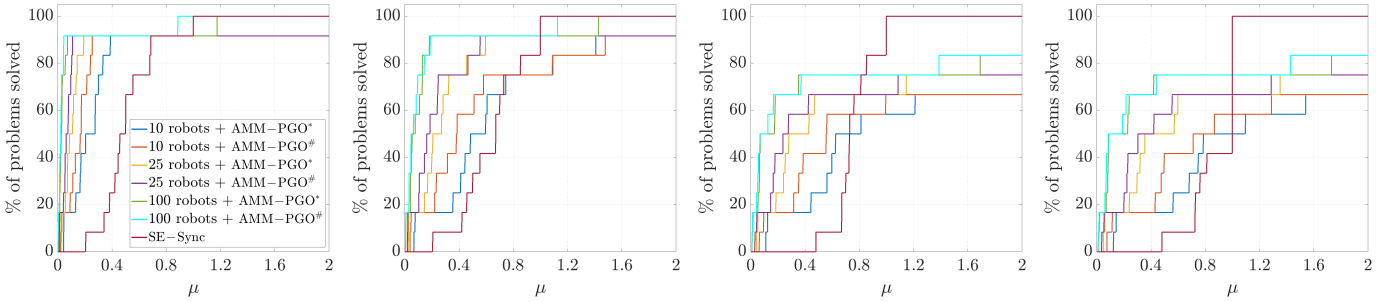
where $T_\Delta(p)$ denotes the total optimization time of all the N nodes. We remark that the centralized optimization method has $N = 1$ node and $T_\Delta(p, N) = T_\Delta(p)$. Let $T_{\mathsf{SE\text{–}Sync}}$ denote the optimization time that SE−Sync needs to find the globally optimal solution. The performance profiles assume a distributed PGO method solves problem p for some $\mu \in [0, +\infty)$ if $T_\Delta(p, N) \leq \mu \cdot T_{\mathsf{SE\text{–}Sync}}$. Note that μ is the scaled average optimization time per node and SE−Sync solves problem p globally at $\mu = 1$.

As a result of [52], the performance profiles evaluate the speedup of distributed PGO methods for a given optimization problem set $\mathcal{P}$ using the percentage of problems solved w.r.t. the scaled average optimization time per node $\mu \in [0, +\infty)$:

$$\underset{\text{solved at } \mu}{\text{percentage of problems}} \triangleq \frac{\left|\{p \in \mathcal{P} | T_\Delta(p, N) \leq \mu \cdot T_{\mathsf{SE\text{–}Sync}}\}\right|}{|\mathcal{P}|}.$$

Our method, due to the optimization method taking distribution into account, can be parallelized while retaining guarantees on convergence and computation. A comparison using $T_{\mathsf{SE\text{–}Sync}}$ assumes there is no value in parallelization, and indeed in that setting SE−Sync would be competitive with our method. But parallelization is valuable, and the $T_\Delta(p, N) = \frac{T_\Delta(p)}{N}$ metric in Eq. (70) captures that value, and shows that when we distribute the optimization across agents we get performance that is both superior in accuracy and faster.

Fig. 10 shows the performance profiles based on the scaled average optimization time per node. The tolerances evaluated are $\Delta = 1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}$ and 1×10^{-5}. We report the performance of AMM–PGO* and AMM–PGO$^\#$ with 10, 25 and 100 robots (nodes). For reference, we also evaluate the performance profile of the centralized PGO baseline SE−Sync [8]. As the results demonstrate, AMM–PGO*



(a) $\Delta = 1 \times 10^{-2}$ (b) $\Delta = 1 \times 10^{-3}$ (c) $\Delta = 1 \times 10^{-4}$ (d) $\Delta = 1 \times 10^{-5}$

Fig. 10: Performance profiles for AMM–PGO*, AMM–PGO$^\#$ and SE−Sync [8] on 2D and 3D SLAM benchmark datasets (see [43, Appendix L]). The performance is based on the scaled average optimization time per node $\mu \in [0, +\infty)$ with tolerances $\Delta = 1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}$. The distributed PGO has 10, 25 and 100 robots (nodes) and is initialized with the centralized chordal initialization [40]. Note that SE−Sync solves all the PGO problems globally at $\mu = 1$.

and AMM–PGO$^\#$ are significantly faster than SE−Sync [8] in most cases for modest accuracies of $\Delta = 1 \times 10^{-2}$ and $\Delta = 1 \times 10^{-3}$, for which the only challenging case is the CSAIL dataset, whose chordal initialization is already very close to the globally optimal solution. In spite of the performance decline for smaller tolerances of $\Delta = 1 \times 10^{-4}$ and $\Delta = 1 \times 10^{-5}$, AMM–PGO* and AMM–PGO$^\#$ with 100 robots (nodes) still achieve a $2.5 \sim 20$x speedup of optimization time over SE−Sync for more than 70% of the benchmark datasets, not to mention that the average optimization time per node of AMM–PGO* and AMM–PGO$^\#$ decreases with the number of robots (nodes). Note that the communication overhead is not considered in the experiments. Nevertheless Fig. 10 indicates that AMM–PGO* and AMM–PGO$^\#$ are promising as fast parallel backends for very large-scale PGO and real-time multi-robot SLAM.

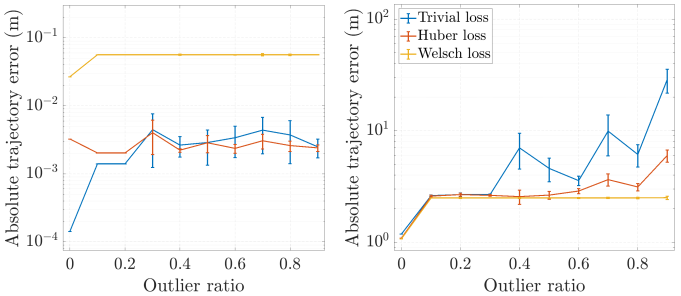
In summary, AMM–PGO* and AMM–PGO$^\#$ achieve the state-of-the-art performance for distributed PGO and enjoy a significant multi-node speedup compared to the centralized baseline [8] for modestly but sufficiently accurate estimates.

C. Robust Distributed PGO

In this section, we evaluate the robustness of AMM–PGO$^\#$ against the outlier inter-node loop closures. Similar to [24], [27], we first use the distributed pairwise consistent measurement set maximization algorithm (PCM) [53] to reject spurious inter-node loop closures and then solve the resulting distributed PGO using AMM–PGO$^\#$ with the trivial, Huber and Welsch loss kernels in Examples 1 to 3 .

We implement AMM–PGO$^\#$ on the 2D intel and 3D garage datasets (see [43, Appendix L]) with 10 robots (nodes). For each dataset, we add false inter-node loop closures with uniformly random rotation and translation errors in the range of $[0, \pi]$ rad and $[0, 5]$ m, respectively. In addition, after the initial outlier rejection using the PCM algorithm [53], we initialize AMM–PGO$^\#$ with distributed Nesterov's accelerated chordal initialization [32] for all the loss kernels.

The absolute trajectory errors (ATE) of AMM–PGO$^\#$ w.r.t. different outlier ratios of inter-node loop closures are in Fig. 11. The ATEs are computed against the outlier-free results of SE−Sync [8] and averaged over 10 Monte Carlo runs.

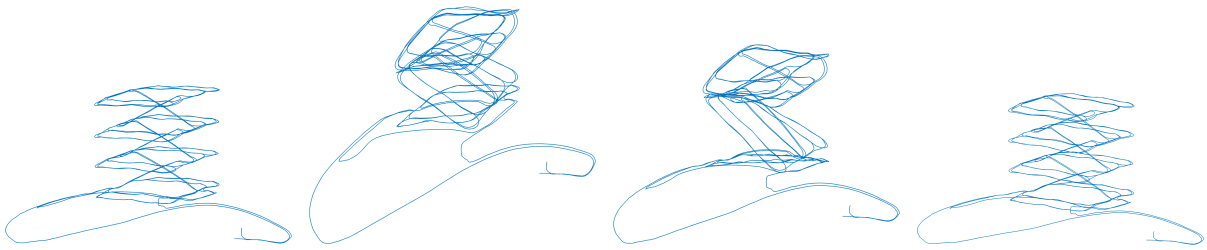


(a) intel (b) garage

Fig. 11: Absolute trajectory errors (ATE) of distributed PGO using AMM–PGO$^\#$ with the trivial, Huber and Welsch loss kernels on the 2D intel and 3D garage datasets. The outlier ratios of inter-node loop closures are $0 \sim 0.9$. The ATEs are computed against the outlier-free results of SE−Sync [8] and are averaged over 10 Monte Carlo runs. PCM [53] is used to initially reject spurious loop closures.

In Fig. 11(a), PCM [53] rejects most of the outlier inter-node loop closure for the intel dataset and AMM–PGO$^\#$ solves the distributed PGO problems regardless of the loss kernel types and outlier ratios. Note that AMM–PGO$^\#$ with the Welsch loss kernel has larger ATEs (avg. 0.057 m) against SE−Sync [8] than those with the trivial and Huber loss kernels (avg. 0.003 m), and we argue that this is related to the loss kernel types. The ATEs are evaluated based on SE−Sync using the trivial loss kernel, which is in fact identical/similar to distributed PGO with the trivial and Huber loss kernels but different from that with the Welsch loss kernel. Thus, the estimates from the trivial and Huber loss kernels are expected to be more close to those of SE−Sync, which result in smaller ATEs compared to the Welsch loss kernel if no outliers.

For the more challenging garage dataset, as is shown in Fig. 11(b), PCM fails for outlier ratios over 0.4, and further, distributed PGO with the trivial and Huber loss kernels results in ATEs as large as 65 m. In contrast, distributed PGO with the Welsch loss kernel still successfully estimates the poses with an average ATE of 2.5 m despite the existence of outliers—note that the garage dataset has a trajectory over 7 km. For the garage dataset, a qualitative comparison of distributed



(a) SE−Sync (b) The trivial loss kernel (c) The Huber loss kernel (d) The Welsch loss kernel

Fig. 12: Qualitative comparisons of distributed PGO with the trivial, Huber and Welsch loss kernels for the garage dataset with spurious inter-node loop closures. The outlier-free result of SE−Sync [8] is shown in Fig. 12(a) for reference. The outlier ratio of inter-node loop closures is 0.6 and PCM [53] is used for initial outlier rejection.

PGO with different loss kernels is also presented in Fig. 12, where the Welsch loss kernel still has the best performance. The results are not surprising since the Welsch loss kernel is known to be more robust against outliers than the other two loss kernels [46].

The results above indicate that our MM methods can be applied to distributed PGO in the presence of outlier inter-node loop closures when combined with robust loss kernels like Welsch and other outlier rejection techniques like PCM [53]. In addition, we emphasize again that our MM methods have provable convergence to first-order critical points for a broad class of robust loss kernels, whereas the convergence guarantees of existing distributed PGO methods [36]–[39] are restricted to the trivial loss kernel.

XI. Conclusion and Future Work

We presented majorization minimization (MM) methods for distributed PGO that has important applications in multi-robot SLAM. Our MM methods had provable convergence for a broad class of robust loss kernels in robotics and computer vision. Furthermore, we elaborated on the use of Nesterov's method and adaptive restart for acceleration and developed accelerated MM methods AMM–PGO* and AMM–PGO$^\#$ without sacrifice of convergence guarantees. In particular, we designed a novel adaptive restart scheme making AMM–PGO$^\#$ without a master node comparable to AMM–PGO* using a master node for information aggregation. The extensive experiments on numerous 2D and 3D SLAM datasets indicated that our MM methods outperformed existing state-of-the-art methods and robustly handled distributed PGO with outlier inter-node loop closures.

Our MM methods for distributed PGO can be improved as follows. A more tractable and robust initialization technique is definitely beneficial to the accuracy and efficiency of distributed PGO. Even though our MM methods have reliable performances against outliers, a more complete theoretical analysis for robust distributed PGO is still necessary. We might also extend our MM methods for differentiable distributed PGO [54]. In addition, our MM methods can be implemented as local solvers for distributed certifiably correct PGO [37] to handle poor or random initialization. Since all the nodes are now assumed to be synchronized, it is necessary and useful to extend our MM methods for asynchronous distributed PGO. Lastly, real multi-robot tests might make the results of our MM methods more convincing where not only the optimization time but also the communication overhead can be validated.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] D. M. Rosen, K. J. Doherty, A. Terán Espinoza, and J. J. Leonard, "Advances in inference and representation for simultaneous localization and mapping," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 215–242, 2021.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[5] A. Singer, "Angular synchronization by eigenvectors and semidefinite programming," *Applied and Computational Harmonic Analysis*, vol. 30, no. 1, pp. 20–36, 2011.

[6] A. Singer and Y. Shkolnisky, "Three-dimensional structure determination from common lines in cryo-em by eigenvectors and semidefinite programming," *SIAM Journal on Imaging Sciences*, vol. 4, no. 2, pp. 543–572, 2011.

[7] D. M. Rosen, M. Kaess, and J. J. Leonard, "Rise: An incremental trust-region method for robust online sparse least-squares estimation," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1091–1108, 2014.

[8] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the special euclidean group," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 95–125, 2019.

[9] D. M. Rosen, "Scalable low-rank semidefinite programming for certifiably correct machine perception," in *Intl. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, vol. 3, 2020.

[10] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

[11] T. Fan, H. Wang, M. Rubenstein, and T. Murphey, "Efficient and guaranteed planar pose graph optimization using the complex number representation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1904–1911.

[12] ——, "CPL-SLAM: Efficient and certifiably correct planar graph-based slam using the complex number representation," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1719–1737, 2020.

[13] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 545–565, 2016.

[14] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[15] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428–439, 2009.

[16] J. Briales and J. Gonzalez-Jimenez, "Cartan-sync: Fast and global se (d)-synchronization," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2127–2134, 2017.

[17] L. Li, A. Bayuelo, L. Bobadilla, T. Alam, and D. A. Shell, "Coordinated multi-robot planning while preserving individual privacy," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2188–2194.

[18] Y. Zhang and D. A. Shell, "Complete characterization of a class of privacy-preserving tracking problems," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 299–315, 2019.

[19] A. Cunningham, M. Paluri, and F. Dellaert, "DDF-SAM: Fully distributed SLAM using constrained factor graphs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[20] R. Aragues, L. Carlone, G. Calafiore, and C. Sagues, "Multi-agent localization from noisy relative pose measurements," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 364–369.

[21] A. Cunningham, V. Indelman, and F. Dellaert, "DDF-SAM 2.0: Consistent distributed smoothing and mapping," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 5220–5227.

[22] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[23] J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert, "Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[24] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "DOOR-SLAM: Distributed, online, and outlier resilient slam for robotic teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, 2020.

[25] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[26] V. Tchuiev and V. Indelman, "Distributed consistent multi-robot semantic localization and mapping," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4649–4656, 2020.

[27] Y. Chang, Y. Tian, J. P. How, and L. Carlone, "Kimera-Multi: A system for distributed multi-robot metric-semantic simultaneous localization and mapping," *arXiv:2011.04087*, 2020.

[28] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-Multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *arXiv:2106.14386*, 2021.

[29] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[30] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2016.

[31] T. Fan and T. Murphey, "Generalized proximal methods for pose graph optimization," in *International Symposium on Robotics Research (ISRR)*, 2019.

[32] ——, "Majorization minimization methods for distributed pose graph optimization with convergence guarantees," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[33] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence O (1/kˆ 2)," in *Doklady AN USSR*, vol. 269, 1983, pp. 543–547.

[34] ——, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[35] B. O'donoghue and E. Candes, "Adaptive restart for accelerated gradient schemes," *Foundations of Computational Mathematics*, vol. 15, no. 3, pp. 715–732, 2015.

[36] S. Choudhary, L. Carlone, Nieto *et al.*, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *The International Journal of Robotics Research*, vol. 36, no. 12, pp. 1286–1311, 2017.

[37] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *arXiv:1911.03721*, 2019.

[38] R. Tron and R. Vidal, "Distributed 3-d localization of camera sensor networks from 2-d image measurements," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3325–3340, 2014.

[39] E. Cristofalo, E. Montijano, and M. Schwager, "GeoD: Consensus-based geodesic distributed pose graph optimization," *arXiv:2010.00156*, 2020.

[40] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[41] Y. Tian, A. Koppel, A. S. Bedi, and J. P. How, "Asynchronous and parallel distributed pose graph optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5819–5826, 2020.

[42] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.

[43] T. Fan and T. Murphey, "Majorization minimization methods for distributed pose graph optimization," 2021. [Online]. Available: https://arxiv.org/abs/2108.00083

[44] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 62–69.

[45] L. Carlone and G. C. Calafiore, "Convex relaxations for pose graph optimization with outliers," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1160–1167, 2018.

[46] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[47] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.

[48] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," *Mathematical Programming*, vol. 156, no. 1-2, pp. 59–99, 2016.

[49] C. Jin, P. Netrapalli, and M. I. Jordan, "Accelerated gradient descent escapes saddle points faster than gradient descent," in *Conference On Learning Theory*, 2018.

[50] H. Li and Z. Lin, "Accelerated proximal gradient methods for nonconvex programming," in *Advances in Neural Information Processing Systems*, 2015, pp. 379–387.

[51] H. Zhang and W. W. Hager, "A nonmonotone line search technique and its application to unconstrained optimization," *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 1043–1056, 2004.

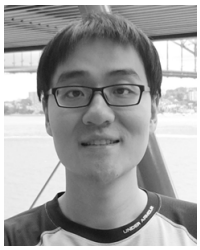
[52] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[53] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2916–2923.

[54] L. Pineda, T. Fan, M. Monge, S. Venkataraman, P. Sodhi, R. Chen, J. Ortiz, D. DeTone, A. Wang *et al.*, "Theseus: A library for differentiable nonlinear optimization," *arXiv:2207.09442*, 2022.

[55] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[56] A. McAdams, A. Selle, R. Tamstorf, J. Teran, and E. Sifakis, "Computing the singular value decomposition of 3x3 matrices with minimal branching and elementary floating point operations," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2011.



Taosha Fan received his B.E. degree in automotive engineering from Tongji University, Shanghai, China, in 2013, and M.S. degrees in mechanical engineering and mathematics from Johns Hopkins University, Baltimore, MD, USA, in 2015, and Ph.D. degree in mechanical engineering at Northwestern University, Evanston, IL, USA in 2022. His research interests lie at the intersection of robotic control, simulation and estimation. He is currently a research engineer in Meta AI, Pittsburgh, PA, USA.

Todd Murphey received his B.S. degree in mathematics from the University of Arizona and the Ph.D. degree in Control and Dynamical Systems from the California Institute of Technology.

He is currently a Professor of mechanical engineering with Northwestern University, Evanston, IL, USA. His laboratory is part of the Center for Robotics and Biosystems. His research interests include robotics, control, machine learning in physical systems, and computational neuroscience