Parallelizable Riemannian Alternating Direction Method of Multipliers for Non-convex Pose Graph Optimization

Xin Chen[1], Chunfeng Cui[1*], Deren Han[1], Liqun Qi[2],

[1]School of Mathematical Sciences, Beihang University, Beijing, China
[2]Department of Applied Mathematics, The Hong Kong Polytechnic University, HKSAR, China
{chenxin2020, chunfengcui, handr}@buaa.edu.cn, maqilq@polyu.edu.hk

Abstract

Pose graph optimization (PGO) is fundamental to robot perception and navigation systems, serving as the mathematical backbone for solving simultaneous localization and mapping (SLAM). Existing solvers suffer from polynomial growth in computational complexity with graph size, hindering real-time deployment in large-scale scenarios. In this paper, by duplicating variables and introducing equality constraints, we reformulate the problem and propose a Parallelizable Riemannian Alternating Direction Method of Multipliers (PRADMM) to solve it efficiently. Compared with the state-of-the-art methods that usually exhibit polynomial time complexity growth with graph size, PRADMM enables efficient parallel computation across vertices regardless of graph size. Crucially, all subproblems admit closed-form solutions, ensuring PRADMM maintains exceptionally stable performance. Furthermore, by carefully exploiting the structures of the coefficient matrices in the constraints, we establish the global convergence of PRADMM under mild conditions, enabling larger relaxation step sizes within the interval $(0, 2)$. Extensive empirical validation on two synthetic datasets and multiple real-world 3D SLAM benchmarks confirms the superior computational performance of PRADMM.

Code — https://github.com/HeartsHorizon/PRADMM

1 Introduction

Pose graph optimization (PGO) (Lu and Milios 1997) is a fundamental technique for trajectory estimation from noisy sensor data. This optimization framework corrects cumulative errors in motion systems, serving as the computational core for sensor networks (So and Ye 2007) and simultaneous localization and mapping (SLAM) (Smith, Self, and Cheeseman 1990). In robotics and autonomous driving, PGO enables SLAM by optimally reconciling pose estimates with environmental observations (Montemerlo et al. 2002), while also supporting structure-from-motion (Martinec and Pajdla 2007) and bundle adjustment (Bender et al. 2013).

However, scalability remains a critical bottleneck: existing solvers suffer from polynomial growth in computational complexity with graph size, hindering real-time deployment in large-scale scenarios. To address this limitation, we propose PRADMM—a Riemannian ADMM framework with convergence guarantees, enabling parallel computation across graph vertices.

*Corresponding author.

1.1 Related Work

Pose Graph Optimization Early first-order methods such as stochastic gradient descent (Olson, Leonard, and Teller 2006; Grisetti, Stachniss, and Burgard 2009) reduced computational complexity, while second-order techniques such as Gauss-Newton (Carlone and Dellaert 2015), Levenberg-Marquardt (Kümmerle et al. 2011) and trust-region methods (Rosen, Kaess, and Leonard 2012) achieved faster local convergence. Nevertheless, these approaches remain susceptible to convergence at local minima and have high computational complexity in Hessian construction. Efforts to construct sparse Hessians based on graph connectivity (Grisetti et al. 2010) ultimately fail to address this issue in dense graph models. To address non-convexity, initialization schemes such as chordal (Carlone et al. 2015b) and rotation synchronization (Nasiri, Moradi, and Hosseini 2018) have been proposed. Unlike prior methods, convex relaxation techniques guarantee convergence to certifiably global optima regardless of initialization (Carlone et al. 2015a; Rosen et al. 2019). However, solving the resultant semidefinite programming or its low-rank approximations remains computationally prohibitive. Recent work further accelerated convergence via problem-aware first-order methods (Fan and Murphey 2023) and trigonometric parameterizations (Nasiri, Hosseini, and Moradi 2020), balancing speed and accuracy. However, these methods scale poorly for large-scale or densely connected graphs, demanding the development of algorithms with near-constant complexity relative to graph scale.

Riemannian ADMM ADMM is an efficient algorithm widely applied in large-scale optimization across diverse domains (Gabay and Mercier 1976; Boyd et al. 2011). The emergence of optimization problems with manifold constraints has prompted increasing research interest in extending vanilla ADMM (see Table 1). Early approaches such as SOC (Lai and Osher 2014) and MADMM (Kovnatsky, Glashoff, and Bronstein 2016) demonstrated empirical effectiveness but lacked theoretical convergence guarantees. Subsequent advancements, including ADMM-NSSC (Lu

Algorithm	Manifold	A_N	Stepsize	Conv.	G.C.
ADMM (Wang, Yin, and Zeng 2019)	$\mathbb{R}^n$	$\mathrm{Im}(A_{-N}) \subseteq \mathrm{Im}(A_N)$	1	S.C.	✓
Prox-ADMM (Boţ and Nguyen 2020)	$\mathbb{R}^n$	surjective	$(0, 2)$	S.C.	✓
iADMM (Hien, Phan, and Gillis 2022)	$\mathbb{R}^n$	surjective	$(0, 2)$	S.C.	$\checkmark_1$
SOC (Lai and Osher 2014)	Stiefel	I_n	1	-	-
MADMM (Kovnatsky, Glashoff, and Bronstein 2016)	Riemannian	I_n	1	-	-
ADMM-NSSC (Lu et al. 2018)	Orthogonal	I_n	1	S.C.	-
Prox-ADMM (Zhang, Ma, and Zhang 2020)	Riemannian	I_n	1	E.C.	-
RADMM (Li, Ma, and Srivastava 2024)	Riemannian	I_n	1	E.C.	-
PieADMM (Chen et al. 2025)	Spherical	I_n	1	E.C.	-
PRADMM (Algorithm 2)	Riemannian	injective	$(0, 2)$	S.C.	✓

"S.C.", "E.C." and " G.C." represent "subsequence convergence", " ergodic complexity" and "global convergence".
$A_{-N} = [A_1, \ldots, A_{N-1}]$, and relaxed stepsize τ comes from the update of dual variables (Algorithm 2).
The symbol $\checkmark_1$ denotes validity solely for $\tau = 1$.

Table 1: Comparison of recent developments of Riemannian ADMM and part of Euclidean ADMM for non-convex problems.

et al. 2018), Prox-ADMM (Zhang, Ma, and Zhang 2020), RADMM (Li, Ma, and Srivastava 2024), and PieADMM (Chen et al. 2025), established rigorous convergence foundations for Riemannian ADMM variants. Notably, Li, Ma, and Srivastava (2024) enhanced the applicability of PRADMM to nonsmooth objectives via the Moreau smoothing technique. However, two fundamental limitations persist: (i) Existing theories (Lu et al. 2018; Zhang, Ma, and Zhang 2020; Li, Ma, and Srivastava 2024; Chen et al. 2025) uniformly require the last block coefficient matrix A_N in the linear equality constraints $\sum_{i=1}^{N} A_i x_i = b$ to be the identity matrix. This restriction persists even in Euclidean nonconvex ADMM theory (Wang, Yin, and Zeng 2019; Boţ and Nguyen 2020; Hien, Phan, and Gillis 2022), where A_N must typically be bijective or surjective, conditions that fail to hold in specific application scenarios. (ii) All these manifold-constrained ADMM analyses establish only subsequence convergence or ergodic complexity, and exclusively for the relaxed dual stepsize $\tau = 1$. Proving convergence for a wider range of step sizes and establishing global convergence remain significant theoretical challenges.

1.2 Main Contributions

Our specific contributions are summarized as follows.

- We decouple vertex correlations through variable splitting and constraint propagation and develop PRADMM with closed-form subproblem solutions. Given the sparsity measurement s, each subproblem complexity is $\mathcal{O}(s)$, which is near-constant relative to graph scale.
- We establish global convergence under row rank-deficient coefficient matrices, significantly weaker conditions than existing requirements. Concurrently, we permit extended relaxed dual steps $\tau \in (0, 2)$, surpassing conventional limits ($\tau \in (0, (1+\sqrt{5})/2)$ for convex and $\tau = 1$ for non-convex cases).
- Our algorithm is evaluated on two synthetic datasets at varying scales and five widely-used 3D SLAM benchmark datasets. Experimental results demonstrate that PRADMM consistently outperforms state-of-the-art graph-SLAM methods. As the problem scale increases, iPRADMM exhibits superior computational scalability with significantly less performance degradation.

2 Model

PGO is mathematically represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (Figure 1). The vertex set $\mathcal{V}$ contains $n = |\mathcal{V}|$ nodes, each corresponding to an unknown robot pose $(\tilde{q}_i, \boldsymbol{t}_i)$. The edge set $\mathcal{E}$ comprises $m = |\mathcal{E}|$ directed arcs, where each edge (i, j) represents a relative measurement $(\tilde{q}_{ij}, \boldsymbol{t}_{ij})$ between poses. The objective is to recover the n unknown poses from these m noisy relative observations. Here, $\tilde{q}_i, \tilde{q}_{ij} \in \mathbb{U}$ are unit quaternions, and $\boldsymbol{t}_i, \boldsymbol{t}_{ij} \in \mathbb{R}^n$ are 3-dimensional vectors for any $i, j = 1, \ldots, n$.

Given that the prior noise distribution is typically unknown in real-world scenarios, we leverage an augmented unit quaternion formulation with the following generative model (Chen et al. 2025):

$$\begin{aligned} \boldsymbol{t}_{ij} &= R_i^\top (\boldsymbol{t}_j - \boldsymbol{t}_i) + \boldsymbol{t}_\epsilon, \quad && \text{where } \boldsymbol{t}_\epsilon \sim \mathcal{N}(0, \Omega_1), \\ \tilde{q}_{ij} &= \tilde{q}_i^* \tilde{q}_j \tilde{q}_\epsilon, && \text{where } \tilde{q}_\epsilon \sim \text{vMF}([1, 0, 0, 0], \kappa), \end{aligned}$$

where R_i and $\tilde{q}_i$ are the rotation representation of vertex i in $SO(3)$ and unit quaternion, respectively. "vMF$(\boldsymbol{\mu}, \kappa)$" denotes a d-dimensional von Mises-Fisher distribution where $\boldsymbol{\mu} \in \mathbb{S}^{d-1}$ and $\kappa \geq 0$ are mean direction and concentration parameters, respectively. It is one of the most commonly used distributions to model data distributed on the surface of the unit hypersphere (Fisher 1953; Sra 2012; Hornik and Grün 2014). In fact, as κ increases, the vMF distribution becomes increasingly concentrated at the mean direction $\boldsymbol{\mu}$. When $\kappa = 0$, it corresponds to the uniform distribution on $\mathbb{S}^{d-1}$. When $\kappa \to +\infty$, the distribution approximates a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $1/\kappa$. The MLE of the graph vertices yields the optimization problem

$$\begin{aligned} &\min_{\{\tilde{q}_i, \boldsymbol{t}_i\}} \sum_{(i,j)\in\mathcal{E}} \|\tilde{t}_j - \tilde{t}_i - \tilde{q}_i \tilde{t}_{ij} \tilde{q}_i^*\|_{\Sigma_1}^2 + \|\tilde{q}_j^* \tilde{q}_i \tilde{q}_{ij} - 1\|_{\Sigma_2}^2 \\ &\text{s.t. } \tilde{q}_i \in \mathbb{U},\ \boldsymbol{t}_i \in \mathbb{R}^3,\ i = 1, 2, \ldots, n, \end{aligned} \tag{1}$$

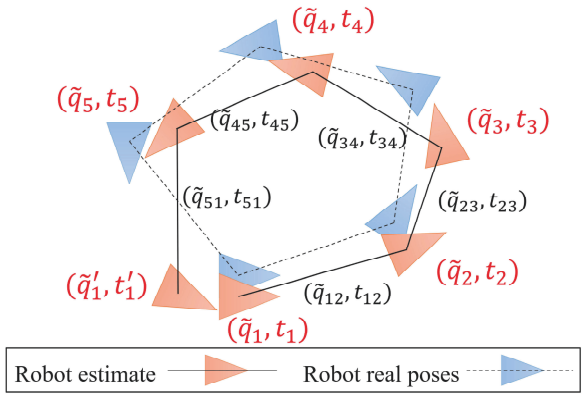

Figure 1: A pose-graph representation of the SLAM process.

where $\Sigma_1 = \begin{pmatrix} c & 0 \\ 0 & \Omega_1^{-1} \end{pmatrix}$ and $\Sigma_2 = \kappa I$ are positive semidefinite matrices, and $c > 0$ is an arbitrary scalar. $\tilde{t}_i = [0, \boldsymbol{t}_i]$ is a pure quaternion.

3 Algorithm

3.1 Reformulation via Structured Splitting

Scalability remains a critical bottleneck in PGO: existing solvers such as the proximal Riemannian gradient method (Gabay 1982), Riemannian conjugate gradient (Smith 1994), and Riemannian Newton method (Hu et al. 2018) suffer from polynomial growth in computational complexity with graph size. This bottleneck arises because computing gradients requires global pose information, resulting in high complexity $\mathcal{O}(m)$ and $\mathcal{O}(n^2)$ for rotations and translations, respectively.

To overcome this limitation, we employ a splitting strategy that decouples these vertices. The existing splitting strategy, SOC (Lai and Osher 2014), in Riemannian optimization separates manifold constraints from variables. However, the absence of closed-form solutions in subproblems fundamentally prevents computational complexity reduction via this approach. The cost of SOC is $\mathcal{O}(km + n^2)$ per iteration, where k is the number of internal cycles. Another splitting strategy, PieADMM (Chen et al. 2025), decouples rotation variables across vertices, achieving the computational complexity $\mathcal{O}(m/n)$. However, by overlooking the iteration cost for translation subproblems, the cost of $\boldsymbol{t}$ is still $\mathcal{O}(n^2)$. By introducing auxiliary variables $\tilde{p}_i$ and $\boldsymbol{s}_i$, we reformulate (1) into an equivalent model:

$$\begin{aligned} \min_{\{\tilde{p}_i, \tilde{q}_i, \boldsymbol{t}_i, \boldsymbol{s}_i\}} & \sum_{(i,j)\in\mathcal{E}} \underbrace{\|\tilde{t}_j - \tilde{s}_i - \tilde{q}_i \tilde{t}_{ij} \tilde{p}_i^*\|_{\Sigma_1}^2}_{f(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}}, \boldsymbol{t}, \boldsymbol{s})} + \underbrace{\|\tilde{p}_j^* \tilde{q}_i \tilde{q}_{ij} - 1\|_{\Sigma_2}^2}_{g(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}})} \\ \text{s.t.} \quad & \tilde{p}_i = \tilde{q}_i,\ \boldsymbol{t}_i = \boldsymbol{s}_i,\ \tilde{p}_i \in \mathbb{U},\ \tilde{q}_i \in \mathbb{R}^4, \\ & \boldsymbol{t}_i \in \mathbb{R}^3,\ \boldsymbol{s}_i \in \mathbb{R}^3,\ i = 1, \ldots, n. \end{aligned} \tag{2}$$

This reformulation yields two computational advantages, as demonstrated in the following:

(i) The quartic polynomial in (1) transforms into a multilinear model, enabling closed-form solutions for all subproblems.

(ii) Vertex interdependencies are decoupled, facilitating massively parallel computation. Specifically, our PRADMM algorithm achieves $\mathcal{O}(m/n)$ computational complexity for both rotations and translations.

Note that m depends on the density of the graph, i.e., $n \le m \le n^2$. Therefore, with fixed graph sparsity (i.e., constant ratio m/n), the theoretical iteration complexity becomes scale-invariant.

3.2 PRADMM

Let $\tilde{\boldsymbol{p}} = (\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_n)$, $\tilde{\boldsymbol{q}} = (\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_n)$, $\boldsymbol{t} = (\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n)$, and $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n)$. The augmented Lagrangian function of (2) is

$$\begin{aligned} \mathcal{L}_\beta(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}}, \boldsymbol{t}, \boldsymbol{s}, \boldsymbol{\lambda}, \boldsymbol{z}) = & f(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}}, \boldsymbol{t}, \boldsymbol{s}) + g(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}}) \\ & + \sum_{i=1}^n \left\{ -\langle \boldsymbol{\lambda}_i, \tilde{p}_i - \tilde{q}_i \rangle + \frac{\beta_1}{2} \|\tilde{p}_i - \tilde{q}_i\|^2 \right\} \\ & + \sum_{i=1}^n \left\{ -\langle \boldsymbol{z}_i, \boldsymbol{t}_i - \boldsymbol{s}_i \rangle + \frac{\beta_2}{2} \|\boldsymbol{t}_i - \boldsymbol{s}_i\|^2 \right\}, \end{aligned}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{4n}$ and $\boldsymbol{z} \in \mathbb{R}^{3n}$ are the Lagrange multipliers and $\beta_1, \beta_2 > 0$ are penalty parameters. In terms of algorithm design, we use the over-relaxation technique, which allows a larger step size for dual variables. The accelerated algorithm may reduce the number of iterations and achieve faster convergence. The iterative scheme of Parallelizable Riemannian ADMM is given by

$$\begin{cases} \tilde{\boldsymbol{p}}^{k+1} \in \arg\min_{\tilde{\boldsymbol{p}} \in \mathbb{U}^n} \mathcal{L}_\beta(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}}^k, \boldsymbol{t}^k, \boldsymbol{s}^k, \boldsymbol{\lambda}^k, \boldsymbol{z}^k) + \mathcal{Q}_1(\tilde{\boldsymbol{p}}) & \text{(3a)} \\ \tilde{\boldsymbol{q}}^{k+1} \in \arg\min_{\tilde{\boldsymbol{q}}} \mathcal{L}_\beta(\tilde{\boldsymbol{p}}^{k+1}, \tilde{\boldsymbol{q}}, \boldsymbol{t}^k, \boldsymbol{s}^k, \boldsymbol{\lambda}^k, \boldsymbol{z}^k) + \mathcal{Q}_2(\tilde{\boldsymbol{q}}) & \text{(3b)} \\ \boldsymbol{t}^{k+1} \in \arg\min_{\boldsymbol{t}} \mathcal{L}_\beta(\tilde{\boldsymbol{p}}^{k+1}, \tilde{\boldsymbol{q}}^{k+1}, \boldsymbol{t}, \boldsymbol{s}^k, \boldsymbol{\lambda}^k, \boldsymbol{z}^k) + \mathcal{Q}_3(\boldsymbol{t}) & \text{(3c)} \\ \boldsymbol{s}^{k+1} \in \arg\min_{\boldsymbol{s}} \mathcal{L}_\beta(\tilde{\boldsymbol{p}}^{k+1}, \tilde{\boldsymbol{q}}^{k+1}, \boldsymbol{t}^{k+1}, \boldsymbol{s}, \boldsymbol{\lambda}^k, \boldsymbol{z}^k) + \mathcal{Q}_4(\boldsymbol{s}) & \text{(3d)} \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \tau\beta_1(\tilde{\boldsymbol{p}}^{k+1} - \tilde{\boldsymbol{q}}^{k+1}) & \text{(3e)} \\ \boldsymbol{z}^{k+1} = \boldsymbol{z}^k - \tau\beta_2(\boldsymbol{t}^{k+1} - \boldsymbol{s}^{k+1}) & \text{(3f)} \end{cases}$$

where $\mathcal{Q}_i(x) = \frac{1}{2}\|x - x^k\|_{H_i}^2$; $H_i \succ 0$, $i = 1, \ldots, 4$, are positive definite matrices with diagonal structures. Here, $\frac{1}{2}\|\tilde{\boldsymbol{p}} - \tilde{\boldsymbol{p}}^k\|_{H_1}^2$ can be split as $\frac{1}{2}\sum_{i=1}^n \|\tilde{p}_i - \tilde{p}_i^k\|_{H_{1,i}}^2$. In other words, each block of H_1 is still a diagonal matrix with $H_{1,i} = \gamma_{1,i} I_4$. Assume H_2, H_3, H_4 have similar structures.

3.3 Subproblems

We partition the given directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ according to the vertices. We define $\mathcal{E}_i^{in} = \{(l, i) \in \mathcal{E}\}$ for all $l \in \mathcal{V}$, and $\mathcal{E}_i^{out} = \{(i, j) \in \mathcal{E}\}$ for all $j \in \mathcal{V}$. In other words, $\mathcal{E}_i^{in}$ represents all directed edges that pointing to vertex i, while $\mathcal{E}_i^{out}$ is the opposite. Then we have the properties that

$$\begin{aligned} & \mathcal{E} = \bigcup_{i \in \mathcal{V}} \left(\mathcal{E}_i^{in} \cup \mathcal{E}_i^{out} \right), \\ & \mathcal{E}_i^{in} \cap \mathcal{E}_i^{out} = \varnothing \text{ for all } i \in \mathcal{V}, \\ & \mathcal{E}_i^{in} \cap \mathcal{E}_j^{in} = \varnothing, \text{ for all } i \neq j. \end{aligned}$$

Next, we analyze the subproblems (3a)–(3d). All detailed derivations can be found in the appendix.

$\tilde{\boldsymbol{p}}$–subproblem: The problem (3a) can be reformulated as

$$\begin{aligned}\arg\min_{\tilde{\boldsymbol{p}}\in\mathbb{U}^n}\sum_{i=1}^{n}\Bigg\{&\sum_{(i,j)\in\mathcal{E}_i^{out}}\|M(\tilde{q}_i^k)M(\tilde{t}_{ij})D\tilde{p}_i-(\tilde{t}_j^k-\tilde{s}_i^k)\|_{\Sigma_1}^2\\&+\sum_{(l,i)\in\mathcal{E}_i^{in}}\|W(\tilde{q}_{li})W(\tilde{q}_l^k)\tilde{p}_i-1\|_{\Sigma_2}^2\\&+\frac{\beta_1}{2}\|\tilde{p}_i-(\tilde{q}_i^k+\frac{1}{\beta_1}\boldsymbol{\lambda}_i^k)\|^2+\frac{1}{2}\|\tilde{p}_i-\tilde{p}_i^k\|_{H_{1,i}}^2\Bigg\}.\end{aligned}\tag{4}$$

where $M(\cdot), W(\cdot)\in\mathbb{R}^{4\times4}$ are quaternion-generated matrices designed to simplify quaternion multiplication. The matrix $D=\mathrm{diag}(1,-1,-1,-1)$ is a diagonal matrix. Since $\tilde{p}_i$ are fully separable, we can update them in parallel, i.e.,

$$\tilde{p}_i^{k+1}=\arg\min_{\tilde{p}_i\in\mathbb{U}}\ \frac{1}{2}\tilde{p}_i^\top A_{1,i}^k\tilde{p}_i+(b_{1,i}^k)^\top\tilde{p}_i\tag{5}$$

where $A_{1,i}^k\in\mathbb{R}^{4\times4}$ and $b_{1,i}^k\in\mathbb{R}^4$ are obtained by rearranging (4). When Σ_1 and Σ_2 are both scalar matrices, $A_{1,i}^k$ is also a scalar matrix. The solution is $\tilde{p}_i^{k+1}=-b_{1,i}^k/\|b_{1,i}^k\|$ if $b_{1,i}^k$ is non-zero. When Σ_1 and Σ_2 are general matrices, the $\tilde{\boldsymbol{p}}$-subproblem (5) is a special quadratic constraint quadratic programming with spherical constraint. We show the results in Lemma 1 that (5) admits an eigenvalue problem.

Lemma 1. *(Adachi et al. 2017) Consider the spherical constrained problem*

$$\min_{x\in\mathbb{R}^n}m(x)=g^\top x+\frac{1}{2}x^\top Ax,\qquad\text{s.t. }\|x\|=\Delta,\tag{6}$$

where $A\in\mathbb{S}^{n\times n}$. For a solution (x^*,λ^*) of the problem (6)*, the multiplier λ^* is equal to the largest real eigenvalue of $\tilde{Q}(\lambda)$, where*

$$\tilde{Q}(\lambda)=\begin{pmatrix}-I & A+\lambda I\\ A+\lambda I & \frac{-gg^\top}{\Delta^2}.\end{pmatrix}.$$

From the above lemma, problem (5) aims at finding the largest real eigenvalue λ^* such that $\det(\tilde{Q}(\lambda^*))=0$. This can be reformulated as a generalized eigenvalue problem, in which we need to compute the rightmost eigenvalue λ_* such that $Q_1y=-\lambda_*Q_2y$, where

$$Q_1=\begin{pmatrix}-I_4 & A_{1,i}^k\\ A_{1,i}^k & -b_{1,i}^k(b_{1,i}^k)^\top\end{pmatrix},\qquad Q_2=\begin{pmatrix}0 & I_4\\ I_4 & 0\end{pmatrix}.$$

Then we can derive an eigenvalue problem (Algorithm 1) for solving problem (5).

$\tilde{q}_i$, $\boldsymbol{t}_i$, and $\boldsymbol{s}_i$ subproblems Similar as (5), we can update $\tilde{q}_i$, $\boldsymbol{t}_i$, and $\boldsymbol{s}_i$ in parallel. Since these subproblems can be formulated as least squares problems without manifold constraints, they admit directly closed-form solutions.

The pseudocode for solving (2) is summarized in Algorithm 2. All analytical expressions about $A_{s,i}^k$ and $b_{s,i}^k$ can be found in the appendix.

Algorithm 1: Eigenvalue problem for solving $\tilde{p}$-subproblem (3a) .

Input: $A_{1,i}^k$ and $b_{1,i}^k$, which are defined in (5).

1: Compute the rightmost eigenvalue λ_* such that $Qy=-\lambda_*y$, where

$$Q=\begin{pmatrix}A_{1,i}^k & -b_{1,i}^k(b_{1,i}^k)^\top\\ -I_4 & A_{1,i}^k\end{pmatrix}.$$

2: $\tilde{p}_i^{k+1}=-(A_{1,i}^k+\lambda_*I_4)^{-1}b_{1,i}^k$.

Output: $\tilde{p}_i^{k+1}$.

4 Convergence Analysis

For the ease of analysis, let us simplify the notations in model (2) as

$$\begin{aligned}\min_{x_i}\quad&f(x_1,x_2,x_3,x_4)+g(x_1,x_2),\\ \text{s.t.}\quad&\sum_{i=1}^{4}A_ix_i=0,\ x_1\in\mathcal{M},\end{aligned}\tag{7}$$

where $A_1=[I_{4n\times4n},O_{3n\times4n}]^\top$, $A_2=[-I_{4n\times4n},O_{3n\times4n}]^\top$, $A_3=[O_{4n\times3n},I_{3n\times3n}]^\top$, $A_4=[O_{4n\times3n},-I_{3n\times3n}]^\top$. O is a zero matrix. $\mathcal{M}\subseteq\mathbb{R}^{n_1}$ is a smooth Riemannian submanifold embedded in n_1-dimensional Euclidean space. For PGO model (2), $\mathcal{M}$ is the Cartesian product of spheres. Denote $\boldsymbol{x}=(x_1,x_2,x_3,x_4)$, $\boldsymbol{x}_{-i}=(x_1,\dots,x_{i-1},x_{i+1},x_4)$, and $\boldsymbol{x}^{k,i}=(\dots,x_{i-1}^{k+1},x_i^{k+1},x_{i+1}^k,\dots)$. Let $\boldsymbol{x}^{k,0}=\boldsymbol{x}^k$ and $\boldsymbol{x}^{k,4}=\boldsymbol{x}^{k+1}$. When we choose the same β_1 and β_2, Algorithm 2 can be rewritten as

$$\begin{cases}\text{for } i=1,2,3,4,\\ \quad x_i^{k+1}=\arg\min_{x_i\in\mathcal{X}_i}\mathcal{L}_\beta^k(x_i)+\frac{1}{2}\|x_i-x_i^k\|_{H_i}^2,\\ \lambda^{k+1}=\lambda^k-\tau\beta\sum_{i=1}^{4}A_ix_i^{k+1},\end{cases}\tag{8}$$

where $\mathcal{L}_\beta(\boldsymbol{x},\lambda)$ is the augmented Lagrangian function of (7) and $\mathcal{L}_\beta^k(x_i)=\mathcal{L}_\beta(\dots,x_{i-1}^{k+1},x_i,x_{i+1}^k,\dots,\lambda^k)$. $\mathcal{X}_i=\mathcal{M}$ for $i=1$, and $\mathcal{X}_i=\mathbb{R}^{n_i}$ for others.

First, we characterize the geometries of objective functions (2) in the following assumption.

Assumption 1. *We assume that*

(i) f and g are both proper and lower semicontinuous functions and bounded from below in the feasible region, i.e., $f^*=\inf_{x_i\in\mathcal{X}_i}f>-\infty$ and $g^*=\inf_{x_i\in\mathcal{X}_i}g>-\infty$.

(ii) The gradient of $f(\boldsymbol{x})$ is Lipschitz continuous on bounded subset of $\mathbb{R}^{n_1}\times\mathbb{R}^{n_2}\times\mathbb{R}^{n_3}\times\mathbb{R}^{n_4}$ with Lipschitz constant $L_f>0$, i.e., for any $\boldsymbol{x}$ and $\boldsymbol{y}\in\mathbb{R}^{n_1}\times\mathbb{R}^{n_2}\times\mathbb{R}^{n_3}\times\mathbb{R}^{n_4}$, it holds that

$$\|\nabla f(\boldsymbol{x})-\nabla f(\boldsymbol{y})\|\le L_f\|\boldsymbol{x}-\boldsymbol{y}\|^2.$$

Similarly, the gradient of g is Lipschitz continuous on bounded subset of $\mathbb{R}^{n_3}\times\mathbb{R}^{n_4}$ with Lipschitz constant $L_g>0$.

Algorithm 2: PRADMM for solving PGO Model (2).

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\tilde{q}_{ij} \in \mathbb{U}$, $\boldsymbol{t}_{ij} \in \mathbb{R}^3$ for all $(i,j) \in \mathcal{E}$.
Initial points $\tilde{\boldsymbol{p}}^0 \in \mathbb{U}^n$, $\tilde{\boldsymbol{q}}^0 \in \mathbb{U}^n$, $\boldsymbol{t}^0 \in \mathbb{R}^{3n}$, $\boldsymbol{s}^0 \in \mathbb{R}^{3n}$, $\boldsymbol{\lambda}^0 \in \mathbb{R}^{4n}$, $\boldsymbol{z}^0 \in \mathbb{R}^{4n}$.
Parameters $\beta_1, \beta_2 > 0$, $\tau \in (0,2)$, $H_i \succ 0$, $i = 1, \ldots, 4$. (For the choice of these parameters, see Theorem 1).

1: **for** $k = 1, 2, \ldots$ **do**
2: Update $\tilde{p}_i^{k+1}$, $i \in [n]$ in parallel as follows

$$\tilde{p}_i^{k+1} = \begin{cases} -\dfrac{b_{1,i}^k}{\|b_{1,i}^k\|}, & \text{if } \Sigma_i \text{ are scalar matrices,} \\ \text{run } \textbf{Algorithm 1}, & \text{otherwise.} \end{cases}$$

3: Update $\tilde{q}_i^{k+1}$, $i \in [n]$ in parallel by

$$\tilde{q}_i^{k+1} = (A_{2,i}^k)^{-1} b_{2,i}^k.$$

4: Update $\boldsymbol{t}_i^{k+1}$, $i \in [n]$ in parallel by

$$\boldsymbol{t}_i^{k+1} = (A_{3,i}^k)^{-1} b_{3,i}^k.$$

5: Update $\boldsymbol{s}_i^{k+1}$, $i \in [n]$ in parallel by

$$\boldsymbol{s}_i^{k+1} = (A_{4,i}^k)^{-1} b_{4,i}^k.$$

6: Update $\boldsymbol{\lambda}, \boldsymbol{z}$ as follows

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \tau\beta_1(\tilde{\boldsymbol{p}}^{k+1} - \tilde{\boldsymbol{q}}^{k+1}),$$
$$\boldsymbol{z}^{k+1} = \boldsymbol{z}^k - \tau\beta_2(\boldsymbol{t}^{k+1} - \boldsymbol{s}^{k+1}).$$

7: **end for**
Output: $\tilde{\boldsymbol{p}}^{k+1} \in \mathbb{U}^n$ **and** $\boldsymbol{t}^{k+1} \in \mathbb{R}^{3n}$**.**

(iii) For any fixed $\boldsymbol{x}_{-i}$, $i = 1, \ldots, 4$, the partial gradient $\nabla_i f(\boldsymbol{x})$ is globally Lipschitz with constant $L_{f,i}(\boldsymbol{x}_{-i}) > 0$, that is, for any $x_i, y_i \in \mathbb{R}^{n_i}$,

$$\|\nabla_i f(x_i, \boldsymbol{x}_{-i}) - \nabla_i f(y_i, \boldsymbol{x}_{-i})\| \le L_{f,i}(\boldsymbol{x}_{-i})\|x_i - y_i\|.$$

Similarly, $\nabla_i g(x_1, x_2)$, $i = 1, 2$, are also globally Lipschitz with constants $L_{g,1}(x_2) > 0$ and $L_{g,2}(x_1) > 0$.

(iv) If $\boldsymbol{x}_{-i}$ lies in a bounded subset, then the Lipschitz constants of the partial gradient $\nabla_i f(\boldsymbol{x})$ and $\nabla_i g(\boldsymbol{x})$ have uniform upper bounds, respectively, i.e.,

$$\sup_{x_j, j \ne i} L_{f,i}(\boldsymbol{x}_{-i}) \le L_{f,i}, \text{and } \sup_{x_j, j \ne i} L_{g,i}(x_{-i}) \le L_{g,i}.$$

(v) f and g are block multi-convex functions, i.e., for each i with fixed $n-1$ blocks $\boldsymbol{x}_{-i}$, and for any $x_i, y_i \in \mathbb{R}^{n_i}$,

$$f(y_i, \boldsymbol{x}_{-i}) \ge f(x_i, \boldsymbol{x}_{-i}) + \langle \nabla_i f(x_i, \boldsymbol{x}_{-i}), y_i - x_i \rangle.$$

Remark 1. *Assumption 1(ii) implies (iii), but (iii) may admit a smaller constant. Since f and g in PGO model* (2) *are polynomial functions, it is straightforward to verify that they satisfy this assumption (Chen et al. 2025).*

4.1 Subsequential Convergence

Our analysis circumvents a fundamental limitation in classical nonconvex ADMM convergence analysis. Existing works typically require that A_4 is surjective or $Im([A_1, A_2, A_3]) \subset Im(A_4)$ to bound the dual variable residuals (see (Boţ and Nguyen 2020; Hien, Phan, and Gillis 2022; Wang, Yin, and Zeng 2019)). In contrast, we define A_4 in (7) as injective which violates this standard assumption, necessitating a novel convergence analysis. Our key innovation lies in exploiting the invertibility of the combined matrix $[A_2, A_4]$—designed to facilitate alternating updates of x_2 and x_4 rather than conventional simultaneous updates. This novel framework enables us to bound the iterative residual $\|\lambda^{k+1} - \lambda^k\|$. We denote $\Delta x_i^k = x_i^k - x_i^{k-1}$, $\Delta \boldsymbol{x}^k = \boldsymbol{x}^k - \boldsymbol{x}^{k-1}$, and $\Delta\lambda^k = \lambda^k - \lambda^{k-1}$. This analysis framework follows the "bounded dual by primal" technique.

Lemma 2. *Suppose that Assumption 1 holds. Let $\{(\boldsymbol{x}^k, \lambda^k)\}$ be the sequence generated by* (8) *which is assumed to be bounded, then*

$$\begin{aligned} \|\Delta\lambda^{k+1}\|^2 \le\ & \alpha_1 \left(\|\Delta\lambda^k\|^2 - \|\Delta\lambda^{k+1}\|^2\right) \\ & + \tau\alpha_2 \Big(\|\Delta x_1^{k+1}\|^2_{7L_f^2+4L_g^2} + \|\Delta x_3^{k+1}\|^2_{3L_f^2} + \|\Delta x_3^k\|^2_{4L_f^2} \\ & + \|\Delta x_2^{k+1}\|^2_{7L_f^2 I + 4L_g^2 I + 4H_2^\top H_2} + \|\Delta x_2^k\|^2_{4H_2^\top H_2} \\ & + \|\Delta x_4^{k+1}\|^2_{3L_f^2 I + 3H_4^\top H_4} + \|\Delta x_4^k\|^2_{4L_f^2 I + 3H_4^\top H_4} \Big). \end{aligned} \tag{9}$$

where

$$\alpha_1 = \frac{|1-\tau|}{1-|1-\tau|}, \quad \alpha_2 = \frac{\tau}{(1-|1-\tau|)^2}.$$

Next, we construct a Lyapunov function

$$\begin{aligned} \Psi^k :&= \Psi(\boldsymbol{x}^k, \boldsymbol{x}^{k-1}, \lambda^k, \lambda^{k-1}) \\ &= \mathcal{L}_\beta(\boldsymbol{x}^k, \lambda^k) + \frac{\alpha_1}{\tau\beta}\|\Delta\lambda^k\|^2 + \sum_{i=1}^4 \|\Delta x_i^k\|^2_{M_i}, \end{aligned} \tag{10}$$

to prove the subsequential convergence.

Theorem 1. *Suppose that Assumption 1 holds. Let $\{(\boldsymbol{x}^k, \lambda^k)\}$ be the sequence generated by* (8) *which is assumed to be bounded. There exists a constant $\underline{\beta} > 0$, such that for all $\beta > \underline{\beta}$, we have*

(i) $\{\Psi^k\}$ is nonincreasing, i.e.,

$$\Psi^k - \Psi^{k+1} \ge \sum_{i=1}^4 (1-\delta_i)\|\Delta x_i^k\|^2_{M_i}, \tag{11}$$

with $0 < \delta_i < 1$, and the right-hand side is nonnegative.

(ii) the sequences $\{\Delta x_i^k\}$ and $\{\Delta\lambda^k\}$ converge to 0.

(iii) every limit point of the generated sequence $\{(\boldsymbol{x}^k, \lambda^k)\}$ is a stationary point of $\mathcal{L}_\beta$.

4.2 Global Convergence

Since the multiplication of two quaternion variables is a polynomial, the proposed model (2) satisfies the Riemannian Kurdyka–Łojasiewicz property (Theorem 3 in (Huang and Wei 2022)), which helps us establish the global convergence of (8). First, we prove the following bound for $\operatorname{grad}\Psi^{k+1}$.

Algorithm	$\sigma_r = 0.01, \sigma_t = 0.01$			$\sigma_r = 0.03, \sigma_t = 0.05$			$\sigma_r = 0.05, \sigma_t = 0.1$		
	Rel. Err.	NRMSE	Time (s)	Rel. Err.	NRMSE	Time (s)	Rel. Err.	NRMSE	Time (s)
mG-N	0.0711	0.0354	0.407	0.3683	0.1834	0.401	0.5024	0.2502	0.407
SE-sync	0.0711	0.0354	0.179	0.3683	0.1834	0.166	0.5025	0.2502	0.161
RS+PS	0.0699	0.0348	0.069	0.3457	0.1721	0.065	0.4861	0.2420	0.072
RGD	0.0692	0.0344	0.322	0.3087	0.1537	0.355	0.4729	0.2355	0.354
SOC	0.0691	0.0344	0.647	0.3085	0.1536	0.425	0.4729	0.2354	0.436
PieADMM	0.0689	0.0343	0.123	0.3085	0.1536	0.131	0.4729	0.2354	0.215
PRADMM (ours)	**0.0689**	**0.0343**	**0.065**	**0.3085**	**0.1536**	**0.034**	**0.4724**	**0.2352**	**0.042**

Table 2: Numerical results of different noise levels of circular ring datasets with $m = n = 100$.

Lemma 3. *Suppose that Assumption 1 holds. Let* $\{(\boldsymbol{x}^k, \lambda^k)\}$ *be the sequence generated by* (8) *which is assumed to be bounded. If* $\beta > \max\{\beta', \beta''\}$*, then there exist some constants* $\varrho_1, \varrho_2, \varrho_3 > 0$ *such that*

$$\|\operatorname{grad} \Psi^{k+1}\| \leq \varrho_1 \|\Delta \boldsymbol{x}^{k+1}\| + \varrho_2 \|\Delta \boldsymbol{x}^k\| + \varrho_3 \|\Delta \lambda^{k+1}\|.$$

Next, we show the whole sequence convergence. Let $\hat{z} := (\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}, \hat{\lambda}, \hat{\lambda})$ be a limit point of the sequence $\{z^k := (\boldsymbol{x}^k, \boldsymbol{x}^{k-1}, \lambda^k, \lambda^{k-1})\}$. The set of all limit points of $\{z^k\}$ is denoted by

$$\Omega^* := \left\{\hat{z} \mid \exists \{z^{k_j}\} \text{ such that } z^{k_j} \to \hat{z} \text{ as } j \to +\infty\right\}.$$

Theorem 2. *(A finite length property) Suppose that Assumption 1 holds. Let* $\{(\boldsymbol{x}^k, \lambda^k)\}$ *be the sequence generated by* (8) *which is assumed to be bounded. If* Ψ *satisfies the Riemannian KŁ property at every point in* Ω^**, and* $\beta > \max\{\beta', \beta''\}$*, then*

(i) The sequence $\{(\boldsymbol{x}^k, \lambda^k)\}$ *has finite length, that is*

$$\sum_{k=1}^{+\infty} \|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\| + \|\lambda^{k+1} - \lambda^k\| < +\infty. \tag{12}$$

(ii) The sequence $\{(\boldsymbol{x}^k, \lambda^k)\}$ *converges to a stationary point of* $\mathcal{L}_\beta$*.*

Remark 2. *While the global convergence of nonconvex optimization via KŁ functions has been extensively studied in (Bolte, Sabach, and Teboulle 2014; Pock and Sabach 2016), these results cannot be directly extended to our algorithm due to their exclusion of dual variables* λ*. Prior work (Hien, Phan, and Gillis 2022; Hien and Papadimitriou 2024) established global convergence for ADMM with over-relaxation at* $\tau = 1$*, but convergence guarantees for* $0 < \tau < 2$ *remained an open challenge. The core difficulty lies in bounding* $\|\Delta\lambda^{k+1}\|$ *rather than its squared counterpart* $\|\Delta\lambda^{k+1}\|^2$*.*

Although our analysis in Lemma 2 demonstrates that bounding $\|\Delta\lambda^{k+1}\|^2$ *inherently includes the term* $\|\Delta\lambda^k\|^2 - \|\Delta\lambda^{k+1}\|^2$*. This term vanishes when* $\tau = 1$ *but becomes intractable in our proof of global convergence with* $0 < \tau < 2$*. To overcome this persistent challenge, we develop a novel upper bound for* $\|\Delta\lambda^{k+1}\|$ *which enables term-by-term cancellation throughout the summation analysis. This theorem extends prior ADMM frameworks and resolves a key limitation in relaxation parameter selection, while preserving validity on Riemannian manifolds.*

5 Numerical Experiments

We test the effectiveness of Algorithm 2 on different 3D pose graph datasets. As a basis for comparison, we also evaluate the performance of several state-of-the-art pose-graph SLAM approaches, including the manifold-based Gauss-Newton (mG-N) method (Wagner, Birbach, and Frese 2011), SE-sync method (Rosen et al. 2019), pose synchronization (RS+PS) algorithm (Nasiri, Hosseini, and Moradi 2020), and PieADMM (Chen et al. 2025). The CPU time of SE-Sync is computed without the time spent on the optimality check. In addition, we complement our test experimental evaluation encompassing both the standard Riemannian gradient descent (Gabay 1982) with line search and specialized manifold splitting methods, such as SOC (Lai and Osher 2014). All experiments are performed using MATLAB 2020a on an Intel i7-10700F CPU desktop computer with 16GB of memory. More details of all experiments can be found in the appendix.

5.1 Synthetic Datasets

Data Settings. We evaluate the performance on two synthetic datasets:

(a). Circular ring: this is a single loop with a radius of 2 and odometric edges. The closed-loop constraint requires the first point to coincide with the last point. Recovering the trajectory poses is particularly challenging because of the sparse observations, i.e., $m = n$.

(b). Cube dataset: assume that a robot travels on a $2 \times 2 \times 2$ grid world and random loop closures are added between nearby nodes with probability p_{cube}. The total number of vertices is $n = \hat{n}^3$, where $\hat{n}$ is the number of nodes on each side of the cube, and the expectation of the number of edges is $\mathbb{E}(m) = 2(2\hat{n}^3 - 3\hat{n}^2 + 1)p_{cube} + \hat{n}^3 - 1$.

The noisy relative pose measurements are generated by

$$\boldsymbol{t}_{ij} = R_i^\top(\boldsymbol{t}_j - \boldsymbol{t}_i) + \boldsymbol{t}_\epsilon, \text{ where } \boldsymbol{t}_\epsilon \sim \mathcal{N}(0, \sigma_t^2 I_3),$$

$$\tilde{q}_{ij} = \tilde{q}_i^* \tilde{q}_j \tilde{q}_\epsilon, \qquad \text{where } \tilde{q}_\epsilon \sim \text{vMF}([1,0,0,0], \frac{1}{2}\sigma_r^2).$$

Experiment Settings. In our experiments, we set the noise level of the translation part $\sigma_t \in (0.01, 0.1)$, and the magnitude of rotation noise with $\sigma_r \in (0.01, 0.1)$. The relaxation stepsize $\tau = 1.4$. We estimate the upper bounds of L_f and L_g via the principal block in the Hessian matrices of f and g.

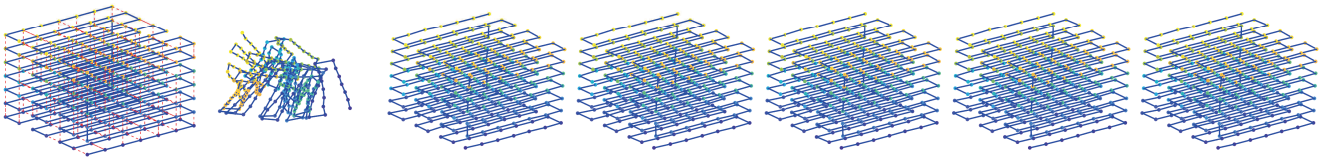

Figure 2: The comparison of cube trajectory with $\hat{n} = 7$. From left to right are the real trajectory, the corrupted trajectory, and the recovered results by mG-N, SE-sync, RS+PS, PieADMM, and PRADMM, respectively.

Algorithm	Rel. Err.	NRMSE	Time (s)
mG-N	0.1782	0.2057	2.157
SE-sync	0.1783	0.2059	0.575
RS+PS	0.1487	0.1717	0.139
PieADMM	0.1502	0.1735	0.920
PRADMM (ours)	0.1478	0.1707	0.213

Table 3: Numerical results of $\hat{n} = 7$ and $p_{cube} = 0.3$ for cube datasets with $m = 697$.

In addition, we find an upper bound of $L_{g,\tilde{\boldsymbol{q}}}$ using $\|\tilde{p}_i\| = 1$. Empirically, we set $H_1 \in \{0.1, 1, 10\}$ and $H_i = 0.001$ for $i = 2, 3, 4$. Then we adjust β around the range and check the descent property of the merit function defined in Theorem 1. We also set $tol = 10^{-4}$ and $MaxIter = 300$ for PRADMM. All algorithms use chordal to find an initial point. Results are averaged over 5 runs.

Experiment Results. The numerical results of circular ring datasets with different noise levels are listed in Table 2, which demonstrate that PRADMM achieves both faster computation and higher accuracy on small-scale circular ring datasets. This validates the significance of developing PRADMM for model (2). Due to non-parallelization or ill-fitting decomposition issues, we exclude RGD and SOC from our subsequent large-scale experiments.

For cube datasets, we evaluate $\hat{n}$ values ranging from 2 to 10 under relative noise conditions $\sigma_t = 0.1$ and $\sigma_r = 0.1$. Figure 2 visually compares reconstructed trajectories for $\hat{n} = 7$ with $p_{cube} = 0.3$, demonstrating comparable reconstruction quality across the tested algorithms. The results under the parameters $\hat{n} = 7$ and $p_{cube} = 0.3$ are listed in Table 3. PRADMM achieves superior reconstruction accuracy compared to alternative algorithms. For large-scale PGO problems, PRADMM maintains competitive efficiency – operating marginally slower than RS+PS yet substantially outperforming all other methods in computation speed. Figure 3 demonstrates PRADMM's stable efficiency with increasing graph size, indicating stronger scalability for large-scale applications.

5.2 SLAM Benchmark Datasets

The evaluations of this part are performed on the widely-used benchmark datasets (Rosen et al. 2019). We use chordal initialization technique for all methods. A portion of existing works (Liu et al. 2012; Fan and Murphey 2019; Nasiri, Hosseini, and Moradi 2020) assesses the precision of solutions by directly comparing rotational angles and translational distances between estimated poses, which quantify error magnitudes across different metrics. The results of the sphere datasets are shown in Table 4, which indicates that PRADMM is efficient in terms of CPU time and translational errors. For more experimental results from other benchmark datasets, please refer to the appendix.

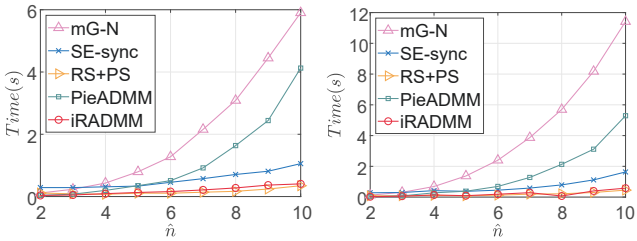

Figure 3: The trend of CPU time along with different $\hat{n}$ for cube datasets. Left: $p_{cube} = 0.3$. Right: $p_{cube} = 0.9$.

Algorithm	Loss(θ)	Loss($\boldsymbol{t}$)	Time (s)
mGN	1.51e+06	8.94e+03	23.739
SE-sync	1.49e+06	9.23e+03	1.142
RS+PS	1.49e+06	9.23e+03	0.987
PieADMM	1.63e+06	7.63e+03	5.697
PRADMM (ours)	1.63e+06	8.13e+03	0.387

Table 4: The numerical results of errors in rotation angle and translation, along with the CPU time consumed for sphere datasets with $n = 2200$, $m = 8647$.

6 Conclusion

This paper presents PRADMM, a novel algorithm for large-scale PGO. We decouple vertex correlations via mathematical reformulation, enabling fully parallelizable vertex updates and remarkably stable computation times regardless of graph size. Crucially, we establish global convergence under significantly weaker conditions (row rank-deficient coefficients) than existing requirements, while permitting extended dual steps ($\tau \in (0, 2)$), surpassing conventional limits on the manifolds. Extensive experiments show PRADMM consistently outperforms state-of-the-art graph-SLAM methods.

Acknowledgments

The authors are grateful to the anonymous referees for their careful reading of the manuscript and their valuable comments. This work was supported by the National Natural Science Foundation of China (Nos. 12471282 and 12131004), Research Center for Intelligent Operations Research, The Hong Kong Polytechnic University (4-ZZT8), and the Fundamental Research Funds for the Central Universities (No. YWF-22-T-204).

References

Adachi, S.; Iwata, S.; Nakatsukasa, Y.; and Takeda, A. 2017. Solving the trust-region subproblem by a generalized eigenvalue problem. *SIAM Journal on Optimization*, 27(1): 269–291.

Bender, D.; Schikora, M.; Sturm, J.; and Cremers, D. 2013. A graph based bundle adjustment for ins-camera calibration. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40: 39–44.

Bolte, J.; Sabach, S.; and Teboulle, M. 2014. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1): 459–494.

Boţ, R. I.; and Nguyen, D.-K. 2020. The proximal alternating direction method of multipliers in the nonconvex setting: convergence analysis and rates. *Mathematics of Operations Research*, 45(2): 682–712.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein., J. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1): 1–122.

Carlone, L.; and Dellaert, F. 2015. Duality-based verification techniques for 2D SLAM. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 4589–4596. IEEE.

Carlone, L.; Rosen, D. M.; Calafiore, G.; Leonard, J. J.; and Dellaert, F. 2015a. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 125–132. IEEE.

Carlone, L.; Tron, R.; Daniilidis, K.; and Dellaert, F. 2015b. Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE International Conference on Robotics and Automation*, 4597–4604. IEEE.

Chen, X.; Cui, C.; Han, D.; and Qi, L. 2025. Non-convex pose graph optimization in SLAM via proximal linearized Riemannian ADMM. *Journal of Optimization Theory and Applications*, 206(3): 1–43.

Fan, T.; and Murphey, T. 2019. Generalized proximal methods for pose graph optimization. In *The International Symposium of Robotics Research*, 393–409. Springer.

Fan, T.; and Murphey, T. D. 2023. Majorization minimization methods for distributed pose graph optimization. *IEEE Transactions on Robotics*, 40: 22–42.

Fisher, R. A. 1953. Dispersion on a sphere. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 217(1130): 295–305.

Gabay, D. 1982. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37: 177–219.

Gabay, D.; and Mercier, B. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1): 17–40.

Grisetti, G.; Kümmerle, R.; Stachniss, C.; and Burgard, W. 2010. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4): 31–43.

Grisetti, G.; Stachniss, C.; and Burgard, W. 2009. Nonlinear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 10(3): 428–439.

Hien, L. T. K.; and Papadimitriou, D. 2024. An inertial ADMM for a class of nonconvex composite optimization with nonlinear coupling constraints. *Journal of Global Optimization*, 1–22.

Hien, L. T. K.; Phan, D. N.; and Gillis, N. 2022. Inertial alternating direction method of multipliers for non-convex non-smooth optimization. *Computational Optimization and Applications*, 83(1): 247–285.

Hornik, K.; and Grün, B. 2014. On maximum likelihood estimation of the concentration parameter of von Mises-Fisher distributions. *Computational Statistics*, 29: 945–957.

Hu, J.; Milzarek, A.; Wen, Z.; and Yuan, Y. 2018. Adaptive quadratically regularized Newton method for Riemannian optimization. *SIAM Journal on Matrix Analysis and Applications*, 39(3): 1181–1207.

Huang, W.; and Wei, K. 2022. Riemannian proximal gradient methods. *Mathematical Programming*, 194(1): 371–413.

Kovnatsky, A.; Glashoff, K.; and Bronstein, M. M. 2016. MADMM: a generic algorithm for non-smooth optimization on manifolds. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, 680–696. Springer.

Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; and Burgard, W. 2011. G2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, 3607–3613.

Lai, R.; and Osher, S. 2014. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58: 431–449.

Li, J.; Ma, S.; and Srivastava, T. 2024. A Riemannian Alternating Direction Method of Multipliers. *Mathematics of Operations Research*.

Liu, M.; Huang, S.; Dissanayake, G.; and Wang, H. 2012. A convex optimization based approach for pose SLAM problems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1898–1903.

Lu, C.; Feng, J.; Lin, Z.; and Yan, S. 2018. Nonconvex sparse spectral clustering by alternating direction method of

multipliers and its convergence analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lu, F.; and Milios, E. 1997. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4: 333–349.

Martinec, D.; and Pajdla, T. 2007. Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE.

Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B.; et al. 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *AAAI/IAAI*, 593598.

Nasiri, S.-M.; Hosseini, R.; and Moradi, H. 2020. Novel parameterization for Gauss–Newton methods in 3-D pose graph optimization. *IEEE Transactions on Robotics*, 37(3): 780–797.

Nasiri, S. M.; Moradi, H.; and Hosseini, R. 2018. A linear least square initialization method for 3D pose graph optimization problem. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2474–2479. IEEE.

Olson, E.; Leonard, J.; and Teller, S. 2006. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2262–2269. IEEE.

Pock, T.; and Sabach, S. 2016. Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM journal on imaging sciences*, 9(4): 1756–1787.

Rosen, D. M.; Carlone, L.; Bandeira, A. S.; and Leonard, J. J. 2019. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *The International Journal of Robotics Research*, 38(2-3): 95–125.

Rosen, D. M.; Kaess, M.; and Leonard, J. J. 2012. An incremental trust-region method for robust online sparse least-squares estimation. In *2012 IEEE International Conference on Robotics and Automation*, 1262–1269. IEEE.

Smith, R.; Self, M.; and Cheeseman, P. 1990. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 167–193.

Smith, S. T. 1994. Optimization Techniques on Riemannian Manifolds. *Fields Institute Communications*, 3.

So, A. M.-C.; and Ye, Y. 2007. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2): 367–384.

Sra, S. 2012. A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of $I_s(x)$. *Computational Statistics*, 27: 177–190.

Wagner, R.; Birbach, O.; and Frese, U. 2011. Rapid development of manifold-based graph optimization systems for multi-sensor calibration and SLAM. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3305–3312. IEEE.

Wang, Y.; Yin, W.; and Zeng, J. 2019. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78: 29–63.

Zhang, J.; Ma, S.; and Zhang, S. 2020. Primal-dual optimization algorithms over Riemannian manifolds: an iteration complexity analysis. *Mathematical Programming*, 184(1-2): 445–490.