Simplifying Certifiable Estimation: A Factor Graph Optimization Approach

Zhexin Xu, Nikolas R. Sanderson, and David M. Rosen

***Abstract*— *Factor graphs* are the dominant paradigm for modeling state estimation tasks in mobile robotics, as they afford both a convenient modular modeling language and fast, scalable inference algorithms. However, most state-of-the-art factor graph inference approaches rely on *local* optimization, which makes them susceptible to converging to incorrect estimates. Recent work has led to the design of novel *certifiable* optimization algorithms capable of efficiently recovering *verifiably globally optimal* estimates in practice. However, Despite these advantages, the widespread adoption of certifiable estimation methods has been limited by the extensive manual effort required to custom-design appropriate relaxations and efficient optimization algorithms. To address these challenges, in this paper we present a method that leverages the same factor graph and local optimization framework widely used in robotics and computer vision to design and deploy a broad range of certifiable estimators. We describe how to implement *lifted* versions of the variable and factor types typically encountered in robotic mapping and localization problems. The result is a set of *certifiable factors* that enables practitioners to develop and deploy globally optimal estimators with the same ease as conventional local methods. Experimental results validate our approach, demonstrating global optimality comparable to that achieved by state-of-the-art certifiable solvers.**

I. Introduction

State estimation is a fundamental problem in the field of robotics [1]. This task aims to recover the true state (e.g., robot pose, environment structure) from noisy measurements obtained from various sensors. Typical applications include SLAM (Simultaneous Localization and Mapping) [2] and SfM (Structure from Motion) [3]. The major challenge lies in the non-convexity and high dimensionality of the problem.

Factor graphs [4] are a powerful framework for modeling robotic state estimation problems. They leverage the insight that these problems are typically composed of a limited set of recurring measurement types, such as visual, inertial, or LiDAR observations, each represented by a corresponding measurement model called a *factor*. Given these elementary factors, more complex and high-dimensional state estimation tasks can be easily and naturally expressed by composing these simple constituent parts.

In robotics and computer vision applications, inference in factor graphs is typically performed using maximum likelihood or maximum a posterior inference. These formulations reduce the problem of *statistical estimation* to *optimization*. This is advantageous because the use of sparsity-exploiting smooth nonlinear programming algorithms enables even large-scale factor graphs to be processed very efficiently. At the same time however, this efficiency comes at the cost of *reliability*: since essentially all real-world state estimation problems in robotics are nonconvex, and standard nonlinear programming methods perform only *local* optimization, solutions obtained through these methods are not guaranteed to be *globally* optimal.

Factor graph optimization is commonly performed using *local* search techniques, such as gradient-based or Newton methods [5]. However, due to the inherent non-convexity of the objective function, solutions obtained through these methods are not guaranteed to be globally optimal. Although accurate results can often be achieved with good initialization [6], these approaches lack formal guarantees—a significant limitation in safety-critical applications where reliability is paramount.

Certifiable estimation refers to a class of methods that provide formal guarantees of global optimality by relaxing non-convex problems into convex formulations. Under moderate noise conditions, these relaxations are often tight, yielding *exact* solutions. While *certifiable* estimators offer strong theoretical guarantees, a key limitation lies in the need to manually construct suitable convex relaxations and develop specialized solvers for efficient computation. Moreover, existing implementations typically lack modularity and code reusability. Consequently, at present designing and implementing certifiable estimators for particular estimation tasks requires significant manual effort. This workflow is substantially less user-friendly compared to well-established factor graph-based libraries such as GTSAM [4] and g2o [7].

In this paper, we show that a broad class of certifiable estimators can be easily designed and deployed within the standard factor graph and local optimization paradigm widely adopted in robotics and computer vision. Specifically, we show how to implement *lifted* (i.e., relaxed) versions of common variable and factor types for mapping and localization tasks. Our approach is implemented directly on top of GTSAM, a widely adopted library for robotics state estimation. Extensive experiments on both synthetic and real-world SLAM datasets showed that the proposed approach preserves the same global optimality guarantees as existing state-of-the-art certifiable solvers [8]. In summary, the key contributions of this work are:

- We introduce a modular framework that streamlines the design and deployment of certifiable estimators within the factor graph paradigm. The proposed approach leverages lifted variables and factor types specifically tailored to classical robotic state estimation problems,

The authors are with the Robust Autonomy Lab, Institute for Experiential Robotics, Northeastern University, 360 Huntington Ave, Boston, MA 02115, USA. {xu.zhex, sanderson.n, d.rosen}@northeastern.edu

enabling seamless integration with existing factor graph optimization workflows.

- We show experimentally that our modular approach recovers the same (globally optimal) solutions as existing hand-crafted certifiable solvers, and does so with a similar computational efficiency.
- To facilitate further research, we will release the full implementation as open-source software.[1]

II. Notation

Let $I_d \in \mathbb{R}^{d\times d}$ denote the $d \times d$ identity matrix. The operator $\mathrm{vec}(\cdot)$ denotes the column-wise vectorization of a matrix. The Frobenius norm is defined as $\|A\|_F = \sqrt{\langle A, A\rangle_F}$. We denote by $\mathrm{Sym}(n)$ the set of real $n \times n$ real symmetric matrices. The operator $\mathrm{BlockDiag}_{d\times d}(M)$ denotes the block-diagonal matrix obtained by extracting all $d \times d$ diagonal blocks of M and zeroing out all off-diagonal entries. The following smooth manifolds are frequently used in this paper:

- The *orthogonal group*:

$$\mathrm{O}(d) \triangleq \left\{R \in \mathbb{R}^{d\times d} \mid R^\top R = I_d\right\} \tag{1}$$

- The *special orthogonal group*:

$$\mathrm{SO}(d) \triangleq \left\{R \in \mathbb{R}^{d\times d} \mid R^\top R = I_d,\ \det(R) = 1\right\} \tag{2}$$

- The *special Euclidean group*:

$$\mathrm{SE}(d) \triangleq \left\{(R, t) \mid R \in \mathrm{SO}(d),\ t \in \mathbb{R}^d\right\} \tag{3}$$

- The *Stiefel manifold*:

$$\mathrm{St}(p, d) \triangleq \left\{Y \in \mathbb{R}^{p\times d} \mid Y^\top Y = I_d\right\} \tag{4}$$

- The *unit sphere*:

$$S^{p-1} \triangleq \left\{r \in \mathbb{R}^p \mid \|r\|_2^2 = 1\right\} \tag{5}$$

Note that the Stiefel manifold plays a central role in certifiable estimation, as both the orthogonal group and the unit sphere are *special cases* of the Stiefel manifold:

$$\mathrm{O}(d) = \mathrm{St}(d, d), \quad \text{and} \quad S^{p-1} = \mathrm{St}(p, 1).$$

III. Related Work

In this section, we review prior work on factor graph representations of state estimation tasks (Section III-A), the Maximum Likelihood Estimation (MLE) in factor graphs (Section III-B), and the certifiable estimation (Section III-C).

A. Factor Graph Representation of the State Estimation Problem

Factor graphs [4] offer a versatile and computationally efficient representation for a wide range of robotic state estimation tasks. Formally, a *factor graph* is a bipartite graph $\mathcal{G} = \{\mathcal{F}, X, \mathcal{E}\}$, where $\mathcal{F}$ is a set of *factors* (corresponding to likelihoods for a set of measurements), X is the set of *variables* (parameters to be estimated), and the edge set $\mathcal{E}$ models which parameters are arguments to which factors. Let $X = \{x_1, x_2, \ldots, x_n\}$ represent the set of latent variables (i.e., the poses, or points) to be estimated, and let $\tilde{Z} = \left\{\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_m\right\}$ denote the corresponding collection of observed sensor measurements. Each node in the variable set X corresponds to an unknown state variable x_i, while each factor $f_i \in \mathcal{F}$ encodes a measurement $\tilde{Z}_i$ that imposes a constraint on the connected variables. We usually assume that the measurement $\tilde{Z}_k$ is sampled from a probabilistic generative model of:

$$\tilde{Z}_k \sim p_k\left(\cdot \mid X_k\right) \quad \forall k \in [m] \tag{6}$$

where $X_k \subseteq \Theta$ represent a *subset* of the full state variable set Θ that contributes to the generation of a specific measurement.

The generative model above captures the phenomenon of *sparsity*: while the *complete* collection of variables X may be very large, *each individual measurement* $\tilde{Z}_k$ typically only depends upon a *very small subset* of these states. As a result, under the assumption of independent measurement noise, the joint measurement likelihood $p(\tilde{Z} \mid X)$ admits the following factorization as a product of "simple" measurement likelihoods:

$$p(\tilde{Z} \mid X) = \prod_{i=1}^{m} p_k\left(\tilde{Z}_k \mid X_k\right) \tag{7}$$

Moreover, even in large-scale and complex estimation tasks, the individual factors $p_k(\tilde{Z}|X)$ appearing in 7 often belong to a very small set of parametric families modeling specific sensing modalities (e.g. camera, IMU, or LiDAR measurements). Consequently, even complex, high-dimensional instances of 7 can be constructed using a small number of variable and factor types, highlighting the inherent modularity and scalability of the factor graph representation.

B. Maximum Likelihood Estimation in Factor Graphs

Maximum Likelihood Estimation (MLE) provides a principled way to convert *statistical estimation* problems [1] into *optimization* problems [4]. In brief, the goal of maximum likelihood estimation is to find a *point estimate* X_{MLE} in $\mathcal{X}$ of the latent state X that *maximizes the joint likelihood* of the available data $\tilde{Z}$:

Solving such problems is to find a point estimate $\hat{X}_{\mathrm{MLE}} \in \mathcal{X}$ of the latent state X. that maximize the joint likelihood of all independent measurement functions:

$$\hat{X}_{\mathrm{MLE}}(\tilde{Z}) \triangleq \underset{X\in\mathcal{X}}{\mathrm{argmax}} p\left(\tilde{Z} \mid X\right) \tag{8}$$

In practical applications, it is standard to reformulate the maximum likelihood estimation problem as a minimization task by considering the *negative log-likelihood*. This transformation is justified by the fact that the logarithm is a monotonically increasing function, making the *maximization* of the likelihood equivalent to the *minimization* of its negative logarithm. Substituting the factorization (7) into the right-hand side of (8) and applying the negative logarithm, we see that maximizing the joint likelihood $p(\tilde{Z}|X)$

[1] https://github.com/NEU-RAL/CertifiableFactors

is equivalent to minimizing the sum of the negative log-likelihoods $l_k(X_k; \tilde{Z}_k) \triangleq -\log p(\tilde{Z}_k \mid X_k)$ of the individual measurements in (6):

$$\hat{X}_{\mathrm{MLE}}(\tilde{Z}) \triangleq \underset{X \in \mathcal{X}}{\operatorname{argmin}} \sum_{k=1}^{m} l_k\left(X_k; \tilde{Z}_k\right) \tag{9}$$

Where each summand $l_k\left(X_k; \tilde{Z}_k\right)$ denotes the negative log-likelihood, which will also be referred to as a *factor* in the following discussion.

Maximum likelihood estimation is attractive because it affords a fast and scalable approach to statistical inference, especially in complex, high-dimensional problems [9]. This advantage is largely due to the fact that MLE typically leads to smooth nonlinear least-squares problems, which can be efficiently solved using sparsity-aware methods such as gradient descent or quasi-Newton techniques [5].

In addition to algorithmic efficiency, MLE also benefits from practical usability. Specifically, the ease of implementation enabled by modern optimization frameworks has made it accessible for large-scale applications. Modern libraries like GTSAM [4], g2o [7], and Ceres [10] simplify implementation by *automatically* generating and solving the optimization problem from a factor graph model, eliminating manual derivation and accelerating development. These features make MLE both theoretically appealing and practically effective for large-scale robotic estimation.

However, despite these strengths, it is important to recognize a key limitation: these local methods solve inherently non-convex problems and therefore cannot guarantee *certifiably globally optimal* solutions.

C. Certifiable Estimation

As an alternative to *local* optimization, recent work has led to the development of a novel class of *certifiable estimators* that are provably capable of efficiently recovering *globally optimal* solutions of state estimation tasks in many practical settings [11]–[14].

In brief, these methods are based upon approximating a challenging (nonconvex) maximum likelihood estimation problem with a *convex relaxation* that can be solved to global optimality in practice [15]. Under moderate noise levels, these relaxations are frequently *exact* [8], [11], enabling the recovery of a *globally* optimal solution to the original (non-convex) estimation problem. In this subsection, we briefly review current state-of-the-art approaches to implementing these methods.

1) Constructing SDP with Shor's relaxation: Usually, certifiable estimators typically require substantially more manual effort to design and implement compared to standard factor graph-based techniques. The standard approach to constructing these involves formulating the original estimation problem as a *quadratically constrained quadratic program* (QCQP), an optimization problem of the form:

$$\begin{aligned} \min_{X \in \mathbb{R}^{n \times k}} \; & \left\langle Q, XX^\top \right\rangle \\ \text{s.t. } & \left\langle A_i, XX^\top \right\rangle = b_i, \quad i = 1, \ldots, m \end{aligned} \tag{10}$$

where $Q \in \mathrm{Sym}(n)$ and $A_i \in \mathrm{Sym}(n)$ are symmetric matrices, and $b_i \in \mathbb{R}^m$.

Given a QCQP of the form (8), we can construct a corresponding convex relaxation using *Shor's relaxation* [16]. In brief, the main idea behind Shor's relaxation is to replace the (rank-k) symmetric outer product $XX^\top$ in (8) with a *generic* positive-semidefinite matrix Z of the same size; this produces the following *semidefinite program* (SDP):

$$\begin{aligned} \min_{Z \in \mathrm{Sym}(n)} \; & \langle Q, Z \rangle \\ \text{s.t. } & \langle A_i, Z \rangle = b_i, \quad i = 1, \ldots, m \\ & Z \succeq 0 \end{aligned} \tag{11}$$

2) Burer–Monteiro Factorization: However, large-scale SDPs are hard to solve using standard off-the-shelf tools. Therefore, current state-of-the-art certifiable estimators rely upon *Burer-Monteiro factorization* ((BM) [17], [18]. The BM method reduces SDP complexity by leveraging

by leveraging the fact that large-scale SDPs often have *low-rank* solutions, i.e., solutions (Z^*) for which $\mathrm{rank}(Z^*) \ll n$. The main idea behind this approach is to replace the original decision variable Z with an *assumed* low-rank factorization of the form $Z = (YY^\top)$, for $Y \in \mathbb{R}^{n \times p}$, naturally enforcing $Z \succeq 0$ and $\mathrm{rank}(Z) \leq p$. The resulting *rank-p* Burer–Monteiro factorization formulation is:

$$\begin{aligned} \min_{Y \in \mathbb{R}^{n \times p}} \; & \left\langle Q, YY^\top \right\rangle \\ \text{s.t. } & \left\langle A_i, YY^\top \right\rangle = b_i, \quad i = 1, \ldots, m \end{aligned} \tag{12}$$

Since $p \ll n$ in practice, the resulting problem operates in a significantly lower-dimensional state space compared to the original SDP, which greatly accelerates the solution of large-scale estimation problems.

3) Verification: The Burer–Monteiro factorization accelerates the solution of large-scale SDPs by reducing dimensionality. However, since the rank-p relaxation in (12) remains non-convex, global optimality of the solution cannot be guaranteed in general. Nevertheless, there exists an efficient procedure to check whether a given solution is globally optimal, which is referred to as *verification* [19]. The *verification* step attempts to certify global optimality of current local optimal solution $Y^\star$ by constructing the *certificate matrix* C based on the Karush-Kuhn-Tucker (KKT) conditions [8]:

$$C \triangleq Q - \frac{1}{2}\,\mathrm{BlockDiag}_{d \times d}\!\left(Q\,Y^\star Y^{\star\top} + Y^\star Y^{\star\top}\,Q\right) \tag{13}$$

where $Q \in \mathrm{Sym}(n)$ is the data matrix as showing in (10). Global optimality is certified if the smallest eigenvalue $\lambda_{\min}$ of C is nonnegative (i.e., C is positive definite).

4) Riemannian Staircase: Since the minimum rank $r^\star = \mathrm{rank}(Z^\star)$ at which we can recover an optimal solution $Z^\star = Y^\star Y^{\star\top}$ of (11) from (12) is typically unknown, the *Riemannian staircase* strategy [18] is commonly employed to solve a sequence of rank-p Burer–Monteiro relaxations. This repeated until global optimality is certified, as outlined in Algorithm 1. At each level p, rank-p Riemannian optimization is used to obtain a candidate solution. Verification

Algorithm 1: Riemannian Staircase

```
Input: Initial feasible point Y ∈ ℝ^{n×p} for rank-p
       Burer-Monteiro factorization (12)
Output: A feasible estimate X̂ for problem (10),
        and the lower bound f*_SDP on (12)'s optimal
        value.
function RiemannianStaircase(Y_p):
  while true do
    // Find critical point of (12)
    Y*_p ← LocalOptimization(Y_p)
    // Construct certificate matrix in (13)
    C ← CertificateMatrix(Y_p)
    // Compute minimum eigenpair of certificate
       matrix C
    (λ_min, v_min) ← MinimumEigenpair(C)
    if λ_min > 0 then
        // Found low-rank factor for optimal
        solution of (12)
        Ŷ ← Y_p;
        f*_SDP ← f_p;
        break;
    // Saddle escape
    else
        p ← p + 1;
        // Construct second-order descent
           direction
        Ẏ_{p+1} ← (0  v)
        // Construct initial point for next instance
           of (12) using backtracking line-search
        Y_{p+1} ← LineSearch(Y_p, Ẏ_{p+1})
  f*_SDP ← tr(Q Ŷ^⊤ Ŷ);
  // Project to feasible set of 10
  X̂ ← RoundSolution(Ŷ);
  return {X̂, f*_SDP};
```

is performed by computing the minimum eigenpair of a certificate matrix. If verification fails, a *saddle-escape* [8] step is performed to construct a rank-$(p{+}1)$ initialization using the second-order descent direction from minimum eigenvalue, allowing the method to ascend to the next level of the staircase.

However, implementing certifiable methods within the above framework presents two main challenges. First, designing appropriate convex relaxations is often problem-specific and lacks a unified formulation. Second, developing optimization algorithms that can efficiently solve these relaxations typically involves substantial manual effort, including the custom design of solvers. Moreover, existing implementations tend to lack modularity and code reusability, making it difficult to adapt and deploy them across different applications

IV. Certifiable Factors

In a nutshell, the central idea motivating this paper is the observation that a broad class of state estimation tasks involving ranging data, including mapping and localization, can be formulated using a small set of factor and variable types: *relative rotation*, *relative translation*, and *point to point* range measurements. At the core of our approach is the formulation of a certifiable estimation problem using these certifiable factors, which are then optimized within a standard factor graph solver. In the following subsections, we first review the measurement models for each of these factors, and then describe how to *relax* them into the lifted variables and factors used in the corresponding *Burer–Monteiro factorization* (12).

A. Relative Rotation Measurement

A *relative rotation* describes the rotation necessary to align the orientation of one coordinate frame with another; given $R_i, R_j \in SO(d)$, the relative rotation R_{ij} from frame i to frame j is then:

$$R_{ij} \triangleq R_i^{-1} R_j \tag{14}$$

In practice, we assume that (noisy) measurements $\tilde{R}_{ij} \in SO(d)$ of relative rotations R_{ij} are sampled from the following probabilistic generative model:

$$\tilde{R}_{ij} = R_{ij}\eta_{ij}, \quad \eta_{ij} \sim \text{Langevin}\left(I_d, \kappa_{ij}\right) \tag{15}$$

where $\kappa_{ij} \geq 0$ is the *precision* of this measurement. The negative log-likelihood function corresponding to (14) is then:

$$\begin{aligned} l_k\left(R_i, R_j; \tilde{R}_{ij}\right) &= \kappa_{ij} \| R_i^\top R_j - \tilde{R}_{ij} \|_F^2 \\ &= \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 \end{aligned} \tag{16}$$

The second equality follows from the invariance of the Frobenius norm under orthogonal transformations. When we apply Shor's relaxation followed by Burer-Monteiro factorization to a problem containing a relative rotation measurement, the factor (16) is transformed to the following *lifted* factor in the resulting Burer-Monteiro factorization (12):

$$l_k\left(Y_i, Y_j; \tilde{R}_{ij}\right) = \kappa_{ij} \| Y_j - Y_i \tilde{R}_{ij} \|_F^2 \tag{17}$$

Here $Y_i, Y_j \in \text{St}(p, d)$ are higher-dimensional *lifts* of the original rotation variables $R_i, R_j \in \text{SO}(d)$.

B. Relative Translation Measurement

A *relative translation* measures the location of a point from the perspective of an observer in a given reference frame. This measurement type can be used to construct both pose-graph and pose-and-landmark formulations of SLAM. Given a point t_j in $\mathbb{R}^d$ and a coordinate frame $x_i = (R_i, t_i) \in SE(d)$, the relative translation of t_j as measured in frame x_i is then:

$$t_{ij} \triangleq R_i^\top (t_j - t_i) \tag{18}$$

In practice, we assume the noisy measurements $\hat{t}_{ij} \in \mathbb{R}^d$ of relative translations t_{ij} are sampled from the generative model:

$$\tilde{t}_{ij} = t_{ij} + \epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}\left(0, \tau_{ij}^{-1} I_d\right) \tag{19}$$

where $\tau_{ij} \geq 0$ is the precision of relative translation measurement.

$$\begin{aligned} l_k\left(t_i, t_j; \tilde{t}_{ij}\right) &= \tau_{ij} \| R_i^{\top}(t_j - t_i) - \tilde{t}_{ij} \|_2^2 \\ &= \tau_{ij} \| t_j - t_i - R_i \tilde{t}_{ij} \|_2^2 \end{aligned} \tag{20}$$

The second equality follows from the invariance of the Euclidean norm under orthogonal transformations. When we apply Shor's relaxation followed by Burer-Monteiro factorization to a problem containing a relative rotation measurement, the factor (20) is transformed to the following lifted factor in the resulting Burer-Monteiro factorization (12):

$$l_k\left(u_i, u_j; \tilde{t}_{ij}\right) = \tau_{ij} \left\| u_j - u_i - Y_i \tilde{t}_{ij} \right\|_2^2 \tag{21}$$

Here, $u_i, u_j \in \mathbb{R}^p$ represent the lifted translation variables, and $Y_i \in \mathrm{St}(p, d)$ is a corresponding lift of the rotation variable $R_i \in SO(d)$.

C. Range Measurement

Range measurements are point-to-point distance between two positions. In practice, it can represent either the distance between two poses or between a pose and a point (e.g., a landmark). Given two points $t_i, t_j \in \mathbb{R}^d$, the *range* between them is:

$$r_{ij} \triangleq \|t_i - t_j\|_2 \tag{22}$$

In practice, we assume that this noisy measurement $\tilde{r}_{ij}$ is sampled from the following probabilistic generative model:

$$\tilde{r}_{ij} = r_{ij} + \nu_{ij}, \qquad \nu_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2) \tag{23}$$

where σ_{ij}^2 is the variance of range measurement. The corresponding factor of the measurement is given by:

$$l_k\left(t_i, t_j; \tilde{r}_{ij}\right) = \frac{1}{\sigma^2}\left(\|t_j - t_i\|_2 - \tilde{r}_{ij}\right)^2 \tag{24}$$

Note that (24) is *not* actually a quadratic function (due to the presence of the unsquared norm). Nevertheless, a technique proposed in [13], [20] enables us to rewrite (24) as an equivalent QCQP by introducing an auxiliary unit vector $b_{ij} \in S^{p-1}$ that effectively models the *bearing* from point t_i to t_j. Applying Shor's followed by Burer-Monteiro factorization to this reformulation, we obtain the following lifted factor:

$$l_k\left(u_i, u_j, b_{ij}; \tilde{r}_{ij}\right) = \frac{1}{\sigma_{ij}^2} \|u_j - u_i - \tilde{r}_{ij} b_{ij}\|_2^2 \tag{25}$$

where $u_i, u_j \in \mathbb{R}^p$.

V. Experiments

In this section, we evaluate the performance of our factor graph-based certifiable estimation approach on both synthetic and real-world SLAM datasets. Concretely, we will apply our approach to recover globally optimal solutions to rotation averaging (RA) and pose-graph optimization (PGO) problems.

As a baseline for comparison, we compare the performance of our approach with SE-Sync [8], a custom built, highly optimized certifiable estimation algorithm specifically designed for RA and PGO problems.

A. Implementation Details

Our method is implemented in GTSAM, with custom variables and factors developed in `C++`. Optimization is performed using the Levenberg–Marquardt algorithm with a relative error tolerance of 10^{-5} with our minimum eigenvalue non-negativity tolerance was set 10^{-3}. For initialization, each point $Y \in \mathrm{St}(p, d)$ and translation $t \in \mathbb{R}^p$ is randomly sampled. In all experiments, the initial rank p_0 is set to two above the ambient problem dimension. We compare our results against SE-Sync in the RA (as *SO-Sync*) and PGO (as *translation-explicit*) configurations. All experiments were conducted on a laptop equipped with an Intel Core i7-11800H processor and 32 GB of RAM, running Ubuntu 20.04.

B. Rotation Averaging

In *rotation averaging*, one aims to determine the values of rotations $R_1, ..., R_n \in \mathrm{SO}(d)$, given a set of noisy measured relative rotations $\tilde{R}_{ij}$ between them. This problem can be modeled as a directed rotation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V} = [n]$ corresponds to the n unknown rotations, and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the available pairwise relative measurements. Under the assumption that these measurements are sampled from the noise model (15), the corresponding maximum likelihood estimation problem is: **Problem 1**. *(MLE formulation of RA).*

$$\min_{R_i \in \mathrm{SO(d)}} \sum_{(i,j) \in \mathcal{E}} \kappa_{ij} \left\| R_j - R_i \tilde{R}_{ij} \right\|_F^2 \tag{26}$$

Applying the results of Section IV-A, the lifted version of Problem 1 solved at each instance of the Riemannian Staircase is then:

Problem 2. *(rank-p Burer-Monteiro factorization of RA)*

$$\min_{Y_i \in \mathrm{St(p,d)}} \sum_{(i,j) \in \mathcal{E}} \kappa_{ij} \left\| Y_j - Y_i \tilde{R}_{ij} \right\|_F^2 \tag{27}$$

Table I presents the results for the Rotation Averaging (RA) problem. Here, p represents the terminal rank for the Riemannian Staircase (Algorithm 1) and total time includes the optimization time as well as the initialization and verification timing results of the algorithm. While we compare the final objective values obtained by SE-Sync and our proposed certifiable factors, our solutions are all verified to be globally optimal, as detailed in Algorithm 1. Note that although the objective values for the Parking Garage dataset differ

TABLE I: Results for Rotation Averaging on SLAM benchmark datasets

Dataset	Vertices	Edges	Objective Value		Opt. Time (s)		Total Time (s)		p
			SE-Sync [8]	Ours	SE-Sync [8]	Ours	SE-Sync [8]	Ours	
MIT (2D)	808	827	3.881×10^1	3.881×10^1	**0.01**	0.11	**0.02**	0.13	4
CSAIL (2D)	1045	1171	2.231×10^1	2.231×10^1	**0.02**	0.10	**0.02**	0.13	4
Intel (2D)	1728	2512	3.639×10^0	3.639×10^0	**0.02**	0.15	**0.03**	0.23	4
Kitti 05 (2D)	2761	2827	2.338×10^2	2.338×10^2	**0.06**	0.31	**0.06**	0.62	4
Manhattan (2D)	3500	5451	6.195×10^3	6.195×10^3	**0.07**	0.35	**0.09**	0.73	4
Kitti 00 (2D)	4251	4679	4.646×10^1	4.646×10^1	**0.07**	0.61	**0.09**	1.08	4
Kitti 02 (2D)	4661	4704	6.375×10^1	6.375×10^1	**0.11**	0.81	**0.12**	1.29	4
City10000 (2D)	10000	20687	2.172×10^2	2.172×10^2	**0.30**	3.61	**0.36**	2.06	4
Ais2klinik (2D)	15115	16727	4.683×10^1	4.683×10^1	**0.44**	3.49	**0.48**	8.48	4
SmallGrid (3D)	125	297	4.850×10^2	4.850×10^2	**0.01**	0.10	**0.01**	0.10	5
Parking Garage (3D)	1661	6275	1.692×10^{-3}	1.733×10^{-3}	**0.04**	0.36	**0.05**	0.54	5
Sphere (3D)	2500	4949	8.854×10^2	8.854×10^2	**0.23**	2.42	**0.26**	2.82	5
Torus (3D)	5000	10000	1.219×10^4	1.219×10^4	**0.46**	4.35	**0.54**	5.83	5
Cubicle (3D)	5750	7696	1.083×10^2	1.084×10^2	**0.43**	2.83	**0.50**	4.76	5
Grid (3D)	8000	22236	4.195×10^4	4.195×10^4	**2.17**	157.09	**3.33**	162.02	5
Rim (3D)	10093	18637	1.527×10^3	1.527×10^3	**0.99**	9.92	**1.14**	16.16	5

TABLE II: Results for Pose Graph Optimization on SLAM benchmark datasets

Dataset	Vertices	Edges	Objective Value		Opt. Time (s)		Total Time (s)		p
			SE-Sync [8]	Ours	SE-Sync [8]	Ours	SE-Sync [8]	Ours	
MIT (2D)	808	827	6.115×10^1	6.115×10^1	**0.27**	0.28	**0.28**	0.33	4
CSAIL (2D)	1045	1171	3.170×10^1	3.170×10^1	**0.07**	0.16	**0.08**	0.22	4
Intel (2D)	1728	2512	5.235×10^1	5.235×10^1	**0.63**	1.00	**0.64**	1.17	4
Kitti 05 (2D)	2761	2827	2.765×10^3	2.765×10^3	1.01	**0.70**	**1.02**	1.12	4
Manhattan (2D)	3500	5451	6.431×10^3	6.431×10^3	**0.65**	1.15	**0.67**	1.87	4
Kitti 00 (2D)	4251	4679	1.257×10^2	1.257×10^2	4.97	**1.07**	4.99	**2.33**	4
Kitti 02 (2D)	4661	4704	1.084×10^2	1.084×10^2	10.16	**1.52**	10.19	**2.79**	4
City10000 (2D)	10000	20687	6.386×10^2	6.386×10^2	16.27	**11.41**	**16.36**	16.64	4
Ais2klinik (2D)	15115	16727	1.886×10^2	1.886×10^2	699.90	**25.13**	699.97	**37.74**	4
SmallGrid (3D)	125	297	1.025×10^3	1.025×10^3	**0.05**	0.11	**0.05**	0.12	5
Parking Garage (3D)	1661	6275	1.263×10^0	1.263×10^0	105.87	**2.57**	105.90	**2.89**	5
Sphere (3D)	2500	4949	1.687×10^3	1.687×10^3	**2.11**	4.04	**2.16**	4.74	5
Torus (3D)	5000	10000	2.423×10^4	2.423×10^4	**1.56**	17.09	**1.69**	19.74	5
Cubicle (3D)	5750	7696	7.171×10^2	7.171×10^2	15.02	**13.74**	**15.15**	17.25	5
Grid (3D)	8000	22236	8.432×10^4	8.432×10^4	**13.00**	919.53	**15.76**	927.42	5
Rim (3D)	10093	18637	5.461×10^3	5.461×10^3	126.84	**63.19**	127.138	**73.90**	5

slightly, all final solutions are successfully certified through the verification step. As shown, our approach consistently achieves the same globally optimal solutions as SE-Sync in all RA datasets. For runtime comparisons, we evaluate our inexact–Newton-based Levenberg–Marquardt method against SE-Sync, which employs Riemannian trust region solver. SE-Sync consistently achieves faster runtime across all RA datasets. This outcome is expected, as SE-Sync is specifically tailored for optimizing RA problems.

C. Pose Graph Optimization

Similarly, in *pose-graph optimization*, one aims to determine the values of the poses $x_1, x_2, \ldots, x_n \in \mathrm{SE}(d)$, when given a set of noisy relative motions $\tilde{x}_{ij} \in \mathrm{SE}(d)$ between them. Note each pose x_i consists of a rotation component $R_i \in \mathrm{SO}(d)$ and a translation component $t_i \in \mathbb{R}^d$, i.e., $x_i = (R_i, t_i)$ and that each measurment $\tilde{x}_{ij}$ contains the translation $\tilde{t}_{ij}$ and the rotation $\tilde{R}_{ij}$ measurements respectively. It is modeled as a directed pose graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V} = [n]$ corresponds to the unknown poses, and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the available relative measurements. Under the assumption that these measurements are sampled from the noise model (15) and (19), the corresponding maximum likelihood estimation problem is:

Problem 3 *(MLE formulation of PGO).*

$$\min_{\substack{R_i \in \mathrm{SO(d)} \\ t_i \in \mathbb{R}^d}} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} \left\| R_j - R_i \tilde{R}_{ij} \right\|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2 \tag{28}$$

where d is the dimension of the problem (e.g., $2D$ or $3D$) and n is the number of pose variables.

Applying the results of Section IV-B, the lifted version of Problem 3 solved at each instance of the Riemannian Staircase is then:

Problem 4 *(rank-p Burer-Monteiro factorization of PGO).*

$$\begin{aligned} \min_{t_i \in \mathbb{R}^p,\, Y_i \in \mathrm{St}(p,d)} \sum_{(i,j)\in\mathcal{E}} & \kappa_{ij} \left\| Y_j - Y_i \tilde{R}_{ij} \right\|_F^2 \\ & + \tau_{ij} \left\| t_j - t_i - Y_i \tilde{t}_{ij} \right\|_2^2 \end{aligned} \tag{29}$$

Table II summarizes the results for the Pose Graph Optimization (PGO) problem. Here again, p represents the final rank for the Riemannian Staircase (Algorithm 1) and total time includes the optimization time as well as the initialization and verification timing results of the algorithm. Similar to the Rotation Averaging (RA) results, our method achieves the same verifiably globally optimal solutions as SE-Sync across all evaluated datasets, with verification confirming the global optimality guarantees of the proposed certifiable factors. The runtime analysis for PGO reveals that neither method consistently outperforms the other across all datasets. Performance varies depending on the dataset's structure and origin. Notably, SE-Sync demonstrates slower performance on real-world datasets such as `Parking Garage`, `Rim`, and `Ais2klinik`, whereas our approach exhibits the highest runtime on the synthetic `Grid3D` dataset. We see faster optimization times on certain datasets however our method's longer initialization times mean it still trails SE-Sync in total runtime.

VI. Conclusions

In this work, we show how one can easily implement and deploy certifiable estimators using existing factor-graph modeling and optimization libraries (such as GTSAM). Our contribution lies in providing a simplified, modular, and practical approach to solving these Riemannian optimization problems directly within the factor graph framework. We propose lifted versions of the variable and factor types that arise in the rank-p Burer–Monteiro factorization, as commonly encountered in state-of-the-art certifiable robotic mapping and localization problems, including rotation averaging, pose graph optimization, landmark-based SLAM, and range-aided SLAM. The effectiveness and optimality of the proposed method are validated through extensive comparisons against state-of-the-art certifiable solvers on both synthetic and real-world SLAM datasets. Future work will explore extending the proposed certifiable factors to a broader class of robotic state estimation problems.

References

[1] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[2] 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.

[3] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[4] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[5] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[6] 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)*. IEEE, 2015, pp. 4597–4604.

[7] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3607–3613.

[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, 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, no. 1, pp. 215–242, 2021.

[10] S. Agarwal, K. Mierle *et al.*, "Ceres solver: Tutorial & reference," *Google Inc*, vol. 2, no. 72, p. 8, 2012.

[11] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2137–2156, 2021.

[12] C. Holmes and T. D. Barfoot, "An efficient global optimality certificate for landmark-based SLAM," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1539–1546, 2023.

[13] A. Papalia, A. Fishberg, B. W. O'Neill, J. P. How, D. M. Rosen, and J. J. Leonard, "Certifiably correct range-aided SLAM," *IEEE Transactions on Robotics*, 2024.

[14] A. Thoms, A. Papalia, J. Velasquez, D. M. Rosen, and S. Narasimhan, "Distributed certifiably correct range-aided SLAM," *arXiv preprint arXiv:2503.03192*, 2025.

[15] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[16] N. Z. Shor, "Quadratic optimization problems," *Soviet Journal of Computer and Systems Sciences*, vol. 25, pp. 1–11, 1987.

[17] S. Burer and R. D. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[18] N. Boumal, V. Voroninski, and A. Bandeira, "The non-convex Burer-Monteiro approach works on smooth semidefinite programs," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[19] D. M. Rosen, "Accelerating certifiable estimation with preconditioned eigensolvers," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 507–12 514, 2022.

[20] T. Halsted and M. Schwager, "The riemannian elevator for certifiable distance-based localization," *Preprint*, 2022.