Simplifying Certifiable Estimation with Factor Graphs: Theory and System

Zhexin Xu, Nikolas R. Sanderson, Hanna Jiamei Zhang, and David M. Rosen, *Member, IEEE*

***Abstract*—Factor graphs provide a convenient modular modeling language for describing statistical estimation tasks, enabling the specification of complex problems by composing simple constituent parts. Current state-of-the-art software libraries typically perform inference in factor graphs using *local* optimization: this approach is operationally convenient (since fast local optimization algorithms can be automatically synthesized and run directly from the factor graph model), but may converge to suboptimal solutions. Alternatively, *certifiably correct* methods based upon convex (semidefinite) relaxation have recently emerged as a powerful approach to solving challenging inference tasks, enabling the efficient recovery of *verifiably globally optimal* solutions in many practical settings. However, at present certifiable methods are often prohibitively laborious to deploy; in particular, they frequently require the design of custom-built, structure-exploiting semidefinite optimization algorithms, necessitating both substantial subject matter expertise and tedious manual analysis, design, and implementation effort.**

In this paper, we bridge these two paradigms by describing a simple approach to designing, implementing, and deploying a wide variety of certifiable estimators using local optimization over factor graphs. Our approach is based upon studying the relation between a quadratically-constrained quadratic program and the Burer-Monteiro factorization of its associated semidefinite relaxation: specifically, we show how to construct a factor graph model for the latter by applying simple algebraic transformations to the *individual* variables and factors appearing in a factor graph model of the former. This insight enables us to easily design and deploy certifiable estimators using the same factor graph software libraries ubiquitously employed in robotics and computer vision. Experimental evaluation on a variety of geometric estimation tasks demonstrates that our *certifiable factor graph* methodology preserves the speed of *local* factor graph optimization approaches while enabling the recovery of *certifiably globally optimal* solutions. Moreover, our framework dramatically reduces the development effort necessary to implement these certifiable estimators, from the order of *months* to *hours*.

***Index Terms*—Factor graphs, certifiable estimation, semidefinite optimization, Burer-Monteiro factorization, Riemannian Staircase, SLAM**

I. Introduction

STATE estimation is the problem of inferring the values of some unknown parameters of interest from (typically noisy) sensor data. This is a foundational capability in mobile robotics, supporting such basic functions as perception, planning, navigation, and control [1], [2]. Canonical examples of state estimation problems in robotics include *simultaneous localization and mapping* (SLAM) [3], [4] and *structure from motion* (SfM) [5], among many others.

Factor graphs provide a powerful framework for modeling state estimation problems in robotics and computer vision [6]. These models are based upon the insight that the estimation tasks arising in these applications are typically composed of a limited set of recurring parameters (e.g. 3D points and robot poses) and measurement types (such as visual, inertial, or LiDAR observations). Given mathematical descriptions of these elementary parameters and sensing modalities, factor graphs provide a convenient, modular modeling language that enables even complex, high-dimensional state estimation problems to be easily and naturally expressed by composing these simple constituent parts.

Factor graphs have consequently become the dominant paradigm for modeling state estimation tasks, with several commonly-used state-of-the-art software libraries based upon this abstraction [7]–[9]. Given a factor graph model of a state estimation task, these libraries typically formalize the inference problem as *maximum likelihood estimation* (MLE) or *maximum a posteriori estimation* (MAP) [6], and then apply smooth local optimization methods to compute a point estimate. This approach is attractive from a computational standpoint, as it admits the use of fast first- or second-order smooth numerical optimization algorithms [10] to compute these point estimates efficiently; indeed, it enables processing estimation problems involving tens to hundreds of thousands of states on a single processor in real-time [8], [9], [11], [12]. Furthermore, it is straightforward to algorithmically synthesize and run a fast local optimization algorithm directly from the factor graph specification of an estimation task; from an operational standpoint, this enables practitioners to focus their attention exclusively on *modeling* the task to be solved (i.e. *what* to do), rather than on the minutiae of designing efficient inference algorithms for the resulting problem formulation (i.e. *how* to do it). However, this computational and operational expedience comes at the cost of *reliability*: because this approach is based upon *local* optimization, the recovered estimates lack guarantees of *correctness* (i.e. *global* optimality). Although accurate estimates can often be obtained with careful initialization [13], the absence of formal performance guarantees remains a critical limitation, especially in safety-critical applications.

Alternatively, *certifiably correct* estimation algorithms have recently emerged as a powerful class of techniques for robustly solving challenging state estimation tasks [4]. These techniques are based upon *convex relaxation* rather than local search: they proceed by constructing a convex (typically *semidefinite*) *approximation* of an MLE or MAP problem instance, and then (*globally*) solving this convex surrogate

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, zhang.hanna, d.rosen}@northeastern.edu.

to produce a point estimate. Remarkably, a growing body of work has shown that this approach is capable of efficiently recovering *exact*, *verifiably globally optimal* solutions to challenging estimation tasks in many practical settings. However, despite their tremendous potential, at present certifiably correct estimators can be prohibitively difficult to design and deploy in practice. In particular, these methods frequently require the solution of large-scale semidefinite relaxations that are far beyond the capabilities of off-the-shelf general-purpose optimization algorithms [14], necessitating the use of more scalable *custom built*, *structure exploiting* techniques. The design and implementation of these specialized semidefinite optimization methods requires substantial subject matter expertise in convex analysis and numerical optimization, along with significant manual analysis, design, and implementation effort. These practical implementation challenges present a serious barrier to the more widespread exploration and development of certifiable methods within the research community, as well as their deployment in real-world systems (particularly for practitioners without specialized expertise in applied mathematics).

In this paper, we describe a simple approach to designing, implementing, and deploying a wide variety of certifiable estimators using the same factor graph modeling and *local* optimization paradigm (and associated software libraries) already employed throughout robotics and computer vision. The main idea behind our approach is to exploit the relationship between a quadratically-constrained quadratic program (QCQP) and the Burer-Monteiro factorization [15] of its associated semidefinite relaxation. Specifically, we show that one can apply simple algebraic transformations (*lifts*) to the individual variables and factors appearing in a factor graph model of the QCQP to produce a corresponding factor graph model for the Burer-Monteiro-factored semidefinite relaxation. This result implies that the rank-restricted relaxation can be instantiated and locally optimized using existing factor graph libraries augmented with *certifiable* (lifted) variants of standard variable and factor types. When applied in combination with the Riemannian Staircase methodology [16], our approach thus enables practitioners to easily synthesize and deploy *certifiable* estimators using current state-of-the-art factor graph-based software libraries and workflows. Experimental evaluation on a variety of geometric estimation tasks demonstrates that certifiable estimators implemented using our factor graph-based approach preserve the computational speed of current state-of-the-art *local* factor graph optimization methods while enabling the recovery of *certifiably globally optimal solutions*. Moreover, our framework dramatically reduces the engineering effort necessary to design and implement these certifiable estimators, from the order of *months* to *hours*.

In summary, the key contributions of this work are:

1) We elucidate the connection between a factor graph model of a quadratically-constrained quadratic program and a factor graph model of the Burer-Monteiro factorization of its associated Shor relaxation. Specifically, we show that the latter can be constructed by applying simple algebraic transformations (*lifts*) to the individual variables and factors appearing in the former. This result implies that it is possible to instantiate and *locally* optimize these Burer-Monteiro factorizations using existing factor graph-based software libraries augmented with lifted variants of standard variable and factor types.
2) We show how to solve the Shor relaxation via the Riemannian Staircase methodology, using our certifiable factor graph framework to instantiate and *locally* optimize the (inner) Burer-Monteiro-factored semidefinite programs in each iteration. This framework enables practitioners to easily design and deploy certifiable estimators using existing factor graph-based software libraries and workflows.
3) We provide concrete descriptions of the lifts of several types of variables (points, poses, and unit vectors) and factors (relative rotations, rigid motions, pose-to-point measurements, and point-to-point ranges) commonly encountered in robotic mapping and navigation tasks.
4) We release an open-source C++ implementation of our approach integrated into GTSAM.[1]

The rest of this paper is organized as follows. We begin by surveying related work on certifiable estimation in Section III. Next, we briefly review the factor graph paradigm as commonly applied to solve state estimation problems in robotics and computer vision (in Section IV), and the construction of certifiable optimization methods for quadratically-constrained quadratic programs via the Riemannian Staircase methodology (in Section V), as these form the foundation of our approach. Section VI presents our main technical contributions: we show how one can easily design and deploy certifiable estimators using factor graphs to model and *locally* optimize the Burer-Monteiro-factored semidefinite programs appearing in each level of the Riemannian Staircase. We apply these results to obtain concrete descriptions of the lifted variants of several common variable and factor types (in Section VII), and then show how one can use these lifted types to implement certifiable estimators for several common estimation tasks in robotic navigation (Section VIII). Finally, Section IX concludes with a summary of this paper's contributions and a discussion of future research directions.

II. Notation

In this section we fix some notation that will be useful throughout the remainder of the paper. Given an integer $n > 0$, we write $[n] \triangleq \{1, \ldots, n\}$ to denote the set of integers from 1 to n (inclusive). Similarly, given an n-tuple $x = (x_1, \ldots, x_n)$ and a subset $S = \{i_1, \ldots, i_k\} \subseteq [n]$ of size k with $i_1 < \cdots < i_k$, we define x_S to be the k-tuple whose elements are indexed by the elements of S:

$$x_S \triangleq (x_{i_1}, \ldots, x_{i_k}). \tag{1}$$

We write $\mathbb{R}$ to denote the set of real numbers. Similarly, given an integer $n > 0$ we denote by $\mathbb{S}^n$ and $\mathbb{S}^n_+$ the sets of symmetric and symmetric positive-semidefinite matrices of size $n \times n$ (respectively). We write $\mathrm{O}(d)$, $\mathrm{SO}(d)$, and $\mathrm{SE}(d)$ for the *orthogonal*, *special orthogonal*, and *special Euclidean* groups acting on d-dimensional Euclidean space

[1] We will provide a link to our implementation following anonymous review.

(respectively). We will identify $\mathrm{O}(d)$ and $\mathrm{SO}(d)$ with their realizations as the following matrix Lie groups:

$$\mathrm{O}(d) \triangleq \{R \in \mathbb{R}^{d\times d} \mid R^{\mathsf{T}}R = I_d\}, \tag{2}$$

$$\mathrm{SO}(d) \triangleq \{R \in \mathbb{R}^{d\times d} \mid R^{\mathsf{T}}R = I_d,\ \det(R) = +1\}, \tag{3}$$

and $\mathrm{SE}(d)$ with its realization as the semidirect product $\mathrm{SE}(d) \cong \mathbb{R}^d \rtimes \mathrm{SO}(d)$, with group operations:

$$(t_1, R_1)\cdot(t_2, R_2) = (t_1 + R_1 t_2,\ R_1 R_2), \tag{4a}$$

$$(t, R)^{-1} = (-R^{-1}t, R^{-1}). \tag{4b}$$

Similarly, given integers k and n with $1 \le k \le n$, we let

$$\mathrm{St}(k, n) \triangleq \{X \in \mathbb{R}^{n\times k} \mid X^{\mathsf{T}}X = I_d\} \tag{5}$$

denote the set of ordered orthonormal k-frames in n-dimensional Euclidean space; this set forms a smooth compact matrix manifold called the *Stiefel manifold* [17]. Finally, we write S^n to denote the n-dimensional unit sphere centered on the origin in $(n+1)$-dimensional Euclidean space:

$$S^n \triangleq \{x \in \mathbb{R}^{n+1} \mid \|x\|_2 = 1\}. \tag{6}$$

Note that comparing (2) and (6) with (5) reveals that the orthogonal group and the unit sphere are themselves specific instances of Stiefel manifolds:

$$S^n = \mathrm{St}(1, n+1), \qquad \mathrm{O}(n) = \mathrm{St}(n, n). \tag{7}$$

III. Related work

In this section, we review prior work on certifiable estimation, with a particular emphasis on approaches leveraging BM factorization.

Recent work has introduced a novel class of *certifiable estimators* that are provably capable of efficiently recovering *globally optimal* solutions to state estimation problems in many practical scenarios. These methods typically rely on convex relaxation techniques, such as Shor's relaxation [18], which relax the original non-convex objective into a convex semidefinite program (SDP). A common approach to solving the resulting SDPs is the Primal-Dual Interior-Point Method [19], often implemented via solvers like Mosek [20], as used in [21]–[24]. However, to the best of our knowledge, these methods have been applied primarily to small- and medium-scale estimation problems. They often require relatively long computation times and have not been demonstrated to efficiently solve large-scale state estimation tasks such as Pose Graph Optimization (PGO) [25] or Bundle Adjustment (BA) [26].

Another widely adopted class of solvers leverages the BM factorization [15], [16], which reformulates semidefinite programs by expressing the large matrix variable as a product of low-rank factors, thereby reducing the dimensionality of the optimization problem. Specifically, BM factorization reduces the computational complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(np^2)$, where $p \ll n$ is the chosen rank of the factorization. This reduction in complexity enables scalable optimization for large-scale estimation problems. A variety of methods in robotics and state estimation have adopted this approach to enable large-scale certifiable estimation. Rosen *et al.* [27] introduced the first certifiable formulation for PGO, providing rigorous theoretical guarantees. Holmes *et al.* [21] and Fan *et al.* [28] extended this methodology to landmark-based SLAM with global optimality guarantees; the latter further established relaxation tightness by exploiting novel geometric constraints derived from complex-number representations. More recently, Papalia *et al.* [29] proposed CORA, a certifiable range-aided SLAM algorithm that incorporated range measurements into the certifiable PGO framework. These foundational works have since been extended to distributed systems [30]–[32]. Despite their theoretical and practical significance, these methods are often highly tailored to specific problem instances, requiring users to design custom relaxations and bespoke solvers. Such formulations typically depend on problem-specific structures, which tightly couple the optimization logic to a particular task and hinder reusability. In practice, several barriers remain for broader adoption: (i) constructing an appropriate relaxation often demands considerable domain knowledge and manual effort, as both the lifted variable structure and data matrices must be carefully engineered; (ii) solvers are commonly implemented from scratch, with limited modularity or code reuse, making adaptation to new applications challenging.

In contrast, our proposed factor graph based framework decouples the problem structure from the solver design. Practitioners only need to define lifted factor types by specifying cost and Jacobian evaluations, while general-purpose solvers handle the certifiable algorithm automatically. All steps previously requiring problem-specific engineering, such as relaxation, optimization, and certification, are integrated into a unified and reusable system. This modular and extensible abstraction not only accelerates the development of certifiable algorithms but also lowers the barrier to deploying globally optimal estimation methods across a wide range of applications. Ultimately, the factor graph paradigm enables users, even those familiar only with local optimization, to develop scalable and certifiable pipelines with minimal additional engineering effort.

A few attempts have been made to integrate certifiable estimation with factor-graph optimization. Shonan Rotation Averaging (Shonan RA) [33] pioneered this direction by lifting rotation variables on the rotation group $\mathrm{SO}(N)$ and employing Lie-group–based local search solvers. However, it is limited to rotation averaging and requires a cumbersome conversion between the Stiefel manifold and the rotation group. In contrast, we propose a general certifiable factor-graph framework that operates directly on manifold-valued variables and supports a wide range of state estimation tasks. We further present a comprehensive theoretical analysis establishing the connection between the original quadratically constrained quadratic programs (QCQPs) and their factor-graph representations in certifiable estimation via BM factorization.

IV. Factor graphs for robot perception

In this section we briefly review the use of *factor graphs* for modeling and solving statistical estimation problems arising in robotic perception and state estimation tasks. Readers are encouraged to consult the monograph [6] for additional detail.

A. Modeling perception problems with factor graphs

At its core, *perception* is the problem of constructing a model of the world from sensor data. As the measurements produced by real-world sensors are typically noisy, it is standard practice to formulate robotic perception problems as *probabilistic inference* tasks [1].

Robotic perception tasks are typically formalized as follows. Let $X \triangleq (X_1, \dots, X_N)$ denote a collection of N unknown parameters (i.e. the *model*) whose values we would like to infer, and $\tilde{z} \triangleq \{\tilde{z}_k\}_{k=1}^K$ be a set of K available noisy sensor measurements. We assume that each measurement $\tilde{z}_k$ is sampled (conditionally) independently from a probabilistic generative model according to:

$$\tilde{z}_k \sim p_k\left(\cdot | X_{S_k}\right) \qquad \forall k \in [K], \tag{8}$$

where parameter indices $S_k \subseteq [N]$ specify the *subset* X_{S_k} of the model parameters X on which the kth observation $\tilde{z}_k$ depends. Intuitively, each generative model (8) describes the operation of an individual sensor (i.e., it describes how a given state X_{S_k} of the world gives rise to a distribution $\tilde{z}_k \sim p_k(\cdot | X_{S_k})$ over possible noisy readings $\tilde{z}_k$ that sensor might return).

Let us now make a few simple but important observations about the measurement model (8). First, note that (8) makes explicit the fact that in many real-world applications, each individual measurement $\tilde{z}_k$ typically only depends upon a *small subset* X_{S_k} of the total set of model parameters X (i.e. $|S_k| \ll n$).[2] Second, the conditional independence of the measurements $\tilde{z}_k$ implies that the joint likelihood $p(\tilde{z}|X)$ of the complete model X given all of the data $\tilde{z}$ factors as the product of the individual measurement likelihoods in (8):

$$p(\tilde{z}|X) = \prod_{k=1}^{K} p_k(\tilde{z}_k | X_{S_k}). \tag{9}$$

Factor graphs provide a convenient graphical representation for modeling the factorization (9) of the joint likelihood $p(\tilde{z}|X)$. In brief, the factor graph $\mathcal{G}$ (see Figure 1) associated to the factorization (9) is the bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ in which:

- The *variable nodes* $\mathcal{V} \triangleq \{X_1, \dots, X_N\}$ consist of the model parameters to be estimated;
- The *factor nodes* $\mathcal{F} \triangleq \{p_1, \dots, p_K\}$ consist of the individual factors (i.e. measurement likelihoods, conditional distributions) p_k appearing on the right-hand side of (9);
- The *edge set* is defined as:

$$\mathcal{E} \triangleq \{(X_n, p_k) \in \mathcal{V} \times \mathcal{F} \mid X_n \in X_{S_k}\}; \tag{10}$$

  that is, variable X_n and factor p_k are joined by an edge in $\mathcal{G}$ if and only if X_n is an argument of the measurement likelihood p_k in (9).

From a practical standpoint, factor graphs are useful because they provide a simple, general, modular modeling language for constructing even complex, high-dimensional joint distributions $p(\tilde{z}|X)$ by composing simple constituent parts [i.e. the individual measurement likelihoods $p_k(\tilde{z}_k|X_{S_k})$ appearing in (8)]. Consequently, several current state-of-the-art software libraries for robotic perception [7]–[9] are based upon this abstraction: when using these libraries, a practitioner specifies a perception task by constructing its associated factor graph model. This simple, object-oriented programming paradigm enables practitioners to easily model a vast array of robotic perception and state estimation tasks by combining standard building blocks, each corresponding to familiar state representations and sensing modalities.

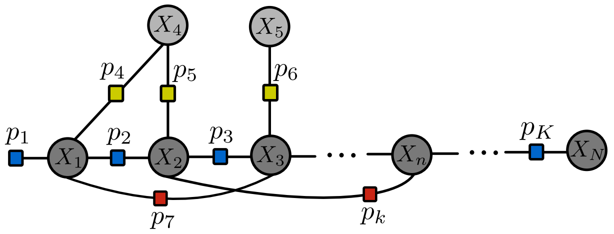


Fig. 1: A general factor graph representation of an estimation problem consisting of n variables X_n and K factors p_k. To highlight different possible variable and factor types in this representation consider a SLAM specific problem instance: p_1 is some *pose prior* and $X_1, X_2, X_3, \dots$ (dark grey) are *robot poses* connected by *odometry* measurements $p_2, p_3, \dots$ (blue). $X_4, X_5, \dots$ (light grey) are *landmarks* observed from specific robot poses through *measurements* $p_4, p_5, p_6, \dots$ (yellow). Finally, $p_7, \dots$ (red) are *loop closure* observations relating different robot poses.

B. Maximum likelihood estimation in factor graphs

Inference in factor graphs is typically performed via *maximum likelihood estimation* (MLE), or more generally *M-estimation* [34]. Recall that in this approach, one determines a point estimate $\hat{X}_{\text{MLE}}$ for the unknown parameter X by *maximizing* the value of the likelihood function $p(\tilde{z}|X)$; intuitively, this is the point estimate that "best explains" the available data $\tilde{z}$. Assuming that the likelihood function $p(\tilde{z}|X)$ factors according to (9), and that the nth parameter X_n takes values in the set $\mathcal{X}_n$, we may formalize the maximum likelihood estimation task as the following optimization problem:

$$\hat{X}_{\text{MLE}}(\tilde{z}) \triangleq \underset{X_n \in \mathcal{X}_n}{\operatorname{argmin}} \sum_{k=1}^{K} \ell_k(X_{S_k}; \tilde{z}_k), \tag{11}$$

where the kth summand $\ell_k(X_{S_k}; \tilde{z}_k)$ in (11), commonly referred to as a factor [3], is either the negative log-likelihood $-\log p_k(\tilde{z}_k|X_{S_k})$ of the kth factor p_k in (9), or a robust generalization thereof [34].

Formulating factor graph inference as the maximum likelihood estimation (11) provides several important advantages. From a theoretical standpoint, maximum likelihood estimation provides strong guarantees on the statistical properties of the resulting estimator $\hat{X}_{\text{MLE}}$, including asymptotic consistency and normality under mild conditions [35]. Computationally,

[2] For example, while a visual mapping task may involve the joint estimation of thousands of camera poses and millions of 3D point positions, *each individual observed image feature point* $\tilde{z}_k$ only depends upon *two* of these model parameters: the pose of the individual camera that captured that specific point observation, and the 3D position of the imaged point.

[3] while ℓ_k is the negative log-likelihood of a conditional distribution (or a robust generalization thereof), we will (in a slight abuse of terminology) refer to it as a "factor", since this is what is actually implemented to solve the corresponding MLE (or robust M-estimation) problem determined by the conditional distributions p_k.

in the typical case that the parameter spaces $\mathcal{X}_n$ are smooth manifolds and the summands ℓ_k are smooth functions, the maximum likelihood estimation (11) is a smooth optimization problem defined over a product of smooth manifolds, enabling us to apply fast sparsity-exploiting first-or second-order smooth optimization methods on manifolds [10], [17], [36] to efficiently recover critical points. This computational efficiency is essential for enabling real-time performance on resource-limited mobile robotic platforms (and indeed several state-of-the-art algorithms based upon this formulation are capable of processing inference problems involving tens to hundreds of thousands of states in real time [8], [11], [12]). Finally, from an operational standpoint, in the typical case in which each parameter set $\mathcal{X}_n$ is a "standard" smooth manifold whose global geometry and topology are known (e.g. common matrix Lie groups, etc.) [4], it is possible to automatically (algorithmically) synthesize and run a fast smooth local optimization algorithm to recover a critical point of the MLE (11) directly from the factor graph model $\mathcal{G}$ describing the inference task. This capability is a core feature of several notable state-of-the-art software libraries for robotic perception (including GTSAM [37], g2o [8], and Ceres [9]), and is a central element of their utility, as it enables practitioners to automatically synthesize and deploy highly-efficient inference algorithms directly from a specification of the task to be solved, without requiring any specialized knowledge (e.g. of geometry or numerical optimization) or manual algorithm design on the part of the user. The practical impact is that these libraries dramatically reduce the effort necessary to implement and deploy high-performance real-time perception systems, especially for users without extensive subject matter expertise in applied mathematics (e.g. differential geometry, numerical linear algebra, and numerical optimization).

However, despite its many favorable properties, practical implementations of the maximum likelihood estimation (11) often suffer from a lack of *reliability*, due to their reliance upon fast *local* optimization methods. Specifically, these local optimization methods can only guarantee convergence to *stationary points* of (11), rather than the *global minimizer* $\hat{X}_{\text{MLE}}$ requried in (11). The result is that the inference algorithms employed in common factor graph-based robotic perception libraries can be surprisingly brittle, often returning egregiously wrong estimates even when the problems to which they are applied are well-posed [4]. This lack of reliability motivates the use of the more robust *certifiable* estimation methods we consider in the next section.

V. Certifiable Estimation

One of the most exciting recent advances in robotic perception and state estimation has been the development of a novel class of inference methods that are capable of efficiently recovering *globally optimal* solutions of challenging nonconvex maximum likelihood estimation problems (11) in many practical settings [4]. These techniques, called *certifiably correct* methods, are based upon *convex relaxation* (rather than *local* optimization): they construct a *convex approximation* of the original MLE problem (11), and then (*globally*) solve it to recover a state estimate. While the idea of applying convex relaxations to solve challenging optimization problems is not new (see e.g. [38]), what distinguishes certifiably correct methods from prior work is that the estimates they return are often *exact, globally optimal* solutions of the original MLE (11), provided that the noise on the data $\tilde{z}$ is not too large [27], [33].

In this section we provide a brief overview of certifiably correct estimation methods. Section V-A introduces Shor's relaxation, which provides the convex SDP relaxation central to these methods. Section V-B presents Burer-Monteiro factorization, used to efficiently solve the resultant large-scale SDPs. Sections V-C and V-D describe how to verify global optimality of candidate solutions and escape saddle points using the Riemannian Staircase methodology, providing both optimality certificates and quantifiable suboptimality bounds when global optimality cannot be certified.

A. Shor's relaxation for quadratically-constrained quadratic programs

Many common robotic perception and state estimation tasks have MLEs (11) that naturally take the form of (or can be equivalently recast as) *quadratically-constrained quadratic programs* (QCQPs), i.e., optimization problems in which the objective and constraint functions are all *quadratic*. This class of problems admits a general procedure, called *Shor's relaxation* [18], for constructing strong convex relaxations. Consequently, Shor's relaxation for QCQPs plays a fundamental role in constructing the convex relaxations underpinning certifiable estimators for robotic perception tasks.

Let us assume we have an estimation problem in the form of a standard (generally nonconvex) QCQP:

$$\begin{aligned} f^\star_{\text{QCQP}} = \min_{X\in\mathbb{R}^{r\times d}} & \left\langle Q, XX^\mathsf{T}\right\rangle \\ \text{s.t. } & \left\langle A_m, XX^\mathsf{T}\right\rangle = b_m, \quad m=[M] \end{aligned} \tag{12}$$

with objective matrix $Q \in \mathbb{S}^r$, contraint matrices $A_m \in \mathbb{S}^r$ for all $m \in [M]$, constants $b_m \in \mathbb{R}^M$, and $d \leq r$. Note that the decision variable X only enters problem (12) through the outer product XX^T, which is a symmetric, positive semidefinite matrix satisfying $\operatorname{rank}(XX^\mathsf{T}) \leq d$ by construction.

To address NP-hardness of (12), Shor's relaxation replaces every instance of the low-rank symmetric outer product XX^T appearing in the QCQP (12) with a *generic* symmetric and positive semidefinite matrix $Z \in \mathbb{S}^r_+$; this substitution removes the implicit rank-d constraint and produces the following (convex) *semidefinite program* (SDP) relaxation:

$$\begin{aligned} f^\star_{\text{SDP}} = \min_{Z\in\mathbb{S}^r_+} & \left\langle Q, Z\right\rangle \\ \text{s.t. } & \left\langle A_m, Z\right\rangle = b_m, \quad m=[M]. \end{aligned} \tag{13}$$

While simple to construct, Shor's relaxation (13) turns out to possess many useful properties. First, in marked contrast

[4] More precisely: we assume that $\mathcal{X}_n \subseteq \mathbb{R}^{r_n}$ a smooth embedded submanifold of some Euclidean space $\mathbb{R}^{r_n}$, and that at each point $x \in \mathcal{X}_n$, we have access to computationally-convenient descriptions of (i) the tangent space $T_x(\mathcal{X}_n)$ of $\mathcal{X}_n$ at x, (ii) the orthogonal projection operator $\Pi_x \colon \mathbb{R}^{r_n} \to T_x(\mathcal{X}_n)$ from the ambient space $\mathbb{R}^{r_n}$ to the tangent space $T_x(\mathcal{X}_n) \subseteq \mathbb{R}^{r_n}$ at x, and (iii) a *retraction* operator $R_x \colon T_x(\mathcal{X}_n) \to \mathcal{X}_n$ at x [17], [36].

to the original QCQP (12), Shor's relaxation is a *convex program*, from which we can recover a *global* minimizer $Z^\star$ in polynomial time using standard interior-point methods. Second, every feasible point X in (12) *lifts* to a corresponding feasible point $Z = XX^\mathsf{T}$ in (13). However, since (13) drops the implicit rank-d constraint, its feasible set contains—and typically exceeds—all such lifted points from (12). Minimizing over this enlarged feasible set therefore yields an optimal value that *lower bounds* the optimal value of (12):

$$f^\star_{\text{SDP}} \leq f^\star_{\text{QCQP}}. \tag{14}$$

In turn, inequality (14) enables us to easily *upper bound* the suboptimality of *any* feasible point X as a solution of the QCQP (12) according to:

$$\langle Q, XX^\mathsf{T} \rangle - f^\star_{\text{QCQP}} \leq \langle Q, XX^\mathsf{T} \rangle - f^\star_{\text{SDP}}. \tag{15}$$

In particular, if the right-hand side of (15) is small, this enables us to conclude that X is a *near*-optimal solution of the QCQP (12) *without* having to compute its optimal value $f^\star_{\text{QCQP}}$. This observation forms the basis for the *solution verification* mechanism used in certifiable estimators. Finally, suppose we solve the Shor relaxation (13) and the recovered minimizer $Z^\star$ satisfies $\operatorname{rank}(Z^\star) \leq d$, so that it admits a symmetric rank-d factorization $Z^\star = X^\star X^{\star\mathsf{T}}$ for some $X^\star \in \mathbb{R}^{r \times d}$. Then the low-rank factor $X^\star$ is *globally optimal* for the original QCQP (12), and the right-hand side of inequality (15) is zero certifying $X^\star$'s optimality. In this case, the relaxation (13) is said to be *exact*.

The remarkable fact that justifies our interest in the Shor relaxed SDP (13) is that it very often turns out to be exact when the original QCQP (12) is an instance of the maximum likelihood estimation (11) and the noise on the available data $\tilde{z}$ is not too large. In this favorable case, we can efficiently recover an *exact, certifiably globally optimal* solution of the nonconvex QCQP (i.e., the *correct* maximum likelihood estimate $\hat{X}_{\text{MLE}}$) by solving the (*convex*) Shor relaxation (13). This algorithmic strategy is the basis of certifiable estimation methods.

B. Burer–Monteiro Factorization

BM factorization provides an efficient and scalable approach for solving large-scale SDPs arising from Shor's relaxation. Even with a convex-relaxed formulation, solving large SDPs remains challenging using standard off-the-shelf solvers—particularly in large-scale estimation problems common in robotics and computer vision. Motivated by the observed low-rank structure of optimal solutions [15], Burer *et al.* proposed a low-rank factorization to accelerate the solution of large SDPs. *Burer–Monteiro factorization* (BM) has since been adopted in many state-of-the-art certifiable estimators [15], [16], [39].

In brief, the BM method reduces SDP complexity by leveraging by leveraging the fact that large-scale SDPs (13) often have *low-rank* structure, i.e., solutions $Z^\star$ for which $\operatorname{rank}(Z^\star) \ll r$. With this in hand, one can replace the original decision variable Z with an *assumed* low-rank factorization of the form $Z = YY^\top$, for lifted variable $Y \in \mathbb{R}^{r \times p}$, naturally enforcing $Z \succeq 0$ and $\operatorname{rank}(Z) \leq p$. The resulting *rank-p* BM factorization of the problem is:

$$\begin{aligned} f^\star_{\text{BM}} = \min_{Y \in \mathbb{R}^{r \times p}} & \ \langle Q, YY^\top \rangle \\ \text{s.t. } & \ \langle A_m, YY^\top \rangle = b_m, \quad m = [M] \end{aligned} \tag{16}$$

Observe that setting $p = r$ recovers the original QCQP (12). Since $p \ll r$ in practice, the resulting problem operates in a significantly lower-dimensional state space compared to the original SDP, $r \times p$ rather than r^2 optimization variables, which reduces the number of computational operations required to solve the problem, greatly accelerating solves of these of large-scale estimation problems. This comes at the cost of reintroducing nonconvexity. In particular, Problem (16) is a nonconvex QCQP, just like the original formulation (12).

Yet, the BM formulation offers crucial computational advantages while maintaining theoretical guarantees. By varying the rank parameter p, BM factorization provides a *hierarchy of relaxations*: increasing p introduces additional degrees of freedom that enable new descent directions, allowing the optimizer to escape saddle points that appear as local minima at lower ranks. Critically, the BM problem preserves a fundamental relationship with the convex SDP relaxation:

$$f^\star_{\text{SDP}} \leq f^\star_{\text{BM}} \leq f^\star_{\text{QCQP}}. \tag{17}$$

When the inequality is tight (i.e., Shor's relaxation is exact), this certifies global optimality of the BM and QCQP solutions. The Riemannian Staircase [16] exploits this by solving a sequence of BM problems with increasing rank-p until obtaining a low-rank solution $Y^\star$ with $\operatorname{rank}(Y^\star) \leq d$ which provides a globally optimal solution $Z = YY^\top$ for both the SDP and original QCQP.

Efficient implementation of a BM approach requires two key components: (i) a computationally tractable test for certifying when a local BM solution corresponds to QCQP global optimum, and (ii) an efficient optimization scheme for solving this sequence nonconvex BM problems. In the following sections, we develop these verification and optimization tools and integrate them into a complete algorithmic framework for certifiable factor graph optimization.

C. Optimality Certification and Saddle-escape

We establish methods to efficiently extract global optima using the nonconvex rank-p BM factorization (16), despite the fact that local optimality does not imply global optimality. Specifically, we show that for any (locally optimal) first-order critical point Y of (16) that fails to yield a global minimizer $Z = YY^\top$ of (12), we can 1) detect this suboptimality (i.e. *verify* the candidate solution) as per [40] and 2) generate a second-order descent direction through embedding in a higher-dimensional BM problem instance.

Theorem 1 (Verify global optimality of candidate solutions obtained via BM factorization): Any first-order stationary point Y of the BM problem with Lagrange multipliers λ yields a candidate solution to the SDP via $Z = YY^\top$. Comparing the Karush–Kuhn–Tucker (KKT) conditions of the BM and SDP problems reveals that such a BM candidate solution satisfies all of the SDP's optimality conditions, *except* for dual feasibility, which requires the associated certificate matrix S to be positive semidefinite ($S \succeq 0$):

$$S \triangleq Q + \mathcal{A}^{\star}(\lambda) = Q + \sum_{m=1}^{M} \lambda_m A_m. \tag{18}$$

Here $Q \in \mathbb{S}^n$ is the data matrix from (12). For notational convenience we introduce the operator $\mathcal{A}^{\star} : \mathbb{R}^M \to \mathbb{S}^r$ which is the adjoint of the linear map $\mathcal{A} : \mathbb{S}^r \to \mathbb{R}^M$ describing the constraint set of (13), i.e. $\mathcal{A}(Z)_m = \langle A_m, Z \rangle, \forall m \in [M]$.

At a candidate solution $Y^{\star}$, the *stationarity condition* of the BM problem yields the Lagrange multipliers λ needed to form S:

$$SY = 0 = (Q + \mathcal{A}^{\star}(\lambda))Y. \tag{19}$$

This is simply a linear system where we can obtain λ by solving the following *least squares* problem.

$$\lambda^{\star} \triangleq \arg\min_{\lambda \in \mathbb{R}^M} \|QY + \mathcal{A}^{\star}(\lambda)Y\|_F^2 . \tag{20}$$

Note that this system only admits a unique solution $\lambda^{\star}$ if the *linear independence constraint qualification* (LICQ) holds. If the LICQ is satisfied, certification is executed efficiently by solving the linear system (20), forming S and evaluating positive semidefiniteness S. If the LICQ is not satisfed, then there exist multiple $\lambda^{\star}$s, but only *some* of which may correspond to a positive semidefinite matrix. See [39] for more detail on the role of the LICQ in for efficient certificate formulation.

Theorem 1 suggests that a rank-p BM solution can be certified as globally optimal for the original QCQP if the corresponding certificate matrix S is positive semidefinite (see Corollary 3, [41]). This amounts to solving a minimum eigenvalue problem which can be done efficiently using [40]: if $\lambda_{\text{min}} \geq 0$, global optimality is certified. However, when $\lambda_{\text{min}} < 0$, the corresponding eigenvector provides a second-order descent direction from the current solution Y, which can be embedded within a higher-rank instance of rank-$(p+1)$ BM factorization. This rank escalation strategy systematically explores increasingly expressive solution spaces until global optimality is achieved.

Theorem 2 (Saddle-escape Theory): Let $Y \in \mathbb{R}^{r \times p}$ be a KKT point of rank-p BM factorization (16) (with corresponding Lagrange multipliers $\lambda \in \mathbb{R}^M$) that satisfies the LICQ, and S be the certificate matrix defined in (18). Then exactly one of the following two cases holds: (a)

1) $S \succeq 0$ and $Z = YY^{\top}$ is a global minimizer of (13).
2) There exists $v \in \mathbb{R}^n$ such that $v^{\top}Sv < 0$, and in that case,

$$Y_{+} = \begin{pmatrix} Y & 0 \end{pmatrix}$$

is a KKT point of (4) attaining the same objective value as Y, and

$$\dot{Y}_{+} = \begin{pmatrix} 0 & v \end{pmatrix}$$

is a feasible second-order direction of descent from Y_{+}.

Further details can be found in Theorem 4 of [41]. In practice, we initialize the rank-$(p+1)$ problem using the second-order descent direction $v \in \mathbb{R}^r$ with backtracking line search [10] to ensure a favorable starting point.

Algorithm 1: Riemannian Staircase

```
Input: Initial feasible point $Y \in \mathbb{R}^{r \times p}$ for rank-p
       Burer-Monteiro factorization (16)
Output: A feasible estimate $\hat{X}$ for problem (12), and
        the lower bound $f^*_{\text{SDP}}$ on (16)'s optimal
        value.
function RIEMANNIANSTAIRCASE($Y_p$):
    while true do
        // Find critical point of (16)
        $Y_p^{\star} \leftarrow$ LOCALOPTIMIZATION($Y_p$)
        // Construct certificate matrix in (18)
        $S \leftarrow$ CERTIFICATEMATRIX($Y_p$)
        // Compute minimum eigenpair of certificate
           matrix S
        $(\lambda_{\text{min}}, v_{\text{min}}) \leftarrow$ MINIMUMEIGENPAIR($S$)
        if $\lambda_{\text{min}} > 0$ then
            // Found low-rank factor for optimal
               solution of (16)
            $\hat{Y} \leftarrow Y_p$;
            $f^*_{\text{SDP}} \leftarrow f_p$;
            break;
        else
            // Saddle escape
            // Construct second-order descent direction
            $p \leftarrow p + 1$
            // Construct initial point for next instance
               of (16) using backtracking line-search
            $\dot{Y}_{p+1} \leftarrow \begin{pmatrix} 0 & v_{\text{min}} \end{pmatrix}$
            $Y_{p+1} \leftarrow$ LINESEARCH($Y_p$, $\dot{Y}_{p+1}$)
    $f^*_{\text{SDP}} \leftarrow \langle Q, \hat{Y}\hat{Y}^{\top} \rangle$;
    // Project to feasible set of (12)
    $\hat{X} \leftarrow$ ROUNDSOLUTION($\hat{Y}$);
    return $\{\hat{X},\ f^*_{\text{SDP}}\}$;
```

D. Riemannian Staircase

Having introduced Shor's relaxation, BM factorization, and the associated verification and saddle-point escape techniques, we now present a unified algorithmic framework that integrates these components. Specifically, we introduce the *Riemannian Staircase* [16], [42], an efficient and scalable method for certifiable estimation. It exploits the fact that orthogonality constraints—common in estimation problems—naturally form smooth Riemannian manifolds, enabling efficient optimization using Riemannian methods.

Algorithm 1 solves the QCQP (12) through a sequence of progressively higher-rank BM factorizations. Starting from a feasible point $Y \in \mathbb{R}^{r \times p}$, it iteratively: (i) performs local optimization over the Stiefel manifold to find a second-order critical point $Y_p^{\star}$, (ii) constructs a certificate matrix S and computes its minimum eigenvalue λ_{min} to verify global optimality, and (iii) either terminates with a certified solution if $\lambda_{\text{min}} > 0$ or or escapes to a higher-rank formulation by constructing a descent direction $\dot{Y}_{p+1}$ from the negative eigenvector v_{min}. When escalating rank, backtracking line search [10] ensures the updated point Y_{p+1} maintains feasibility and achieves descent on the manifold. Upon certification of global optimality, the algorithm returns $\hat{Y} = Y_p^\star$ and the corresponding lower bound f_{SDP}^*, with the final estimate $\hat{X}$ obtained by projecting $\hat{Y}$ back to the feasible set of (12) via a rounding procedure. This approach guarantees global convergence under mild assumptions while maintaining computational efficiency through adaptive rank selection.

VI. Exploiting Factor Graph Structure for Scalable Certifiable Optimization

As established in earlier sections, many estimation problems in robotics and computer vision can be formulated as MLE (11) from factor graph models. Current approaches face a fundamental tradeoff: certifiable methods recast the MLE as a QCQP (12) and solve large-scale semidefinite relaxations without exploiting the convenience for modeling and solving offered by graph structure (Section V), while factor graph optimizers [7]–[9] leverage this structure for efficiency but cannot provide verified global optimal solution.

Our key insight is that factor graph structure induces exploitable algebraic properties in the corresponding QCQP formulation. For factor graph MLE problems (11) that can be written as a generic QCQP in (12), the data matrix $Q \in \mathbb{S}^r$ inherits the block sparsity pattern induced by each factor k acting only on a small subset of variables $X_{S_k} \subseteq X$, and constraint matrices $A_m \in \mathbb{S}^r$ are *block-separable*—each constraint involves only a single variable block—resulting in *block-diagonal* constraint matrices. This structure, preserved from the original factor graph, enables both efficient computation and certification.

In this section, we establish a rigorous theoretical foundation for our certifiable factor-graph framework by formalizing this connection between certifiable estimation via BM factorization and classical factor-graph representations. We begin by showing that factor graph structure in the corresponding MLE problem is preserved when formulated as a QCQP [5]. Building on this equivalence, we analyze both Shor's relaxation and the BM factorization within the factor-graph formalism, showing how these techniques naturally embed into the graph structure. We further demonstrate that the least-squares multipliers arising in the verification step (20) admit a block-diagonal decomposition aligned with the variable partition, enabling independent and parallel computation across factor blocks for efficient certificate formulation. Finally, we introduce our certifiable factor-graph optimization framework: an efficient, modular, and plug-and-play approach that leverages the Riemannian staircase for factor graphs to facilitate convenient optimization with provable optimality guarantees.

A. Exposing Factor Graph Structure in QCQPs

We reformulate the factor graph MLE problem (11) as a QCQP (12) to expose the *block separable* structure that factor graphs impose on the data matrices, i.e. objective and constraints. We begin by partitioning the full decision variable $X \in \mathbb{R}^{r \times d}$ into N *block-row variables* $X_n \in \mathbb{R}^{r_n \times d}$ corresponding to factor-graph variables in (11):

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix}, \quad \text{where} \quad \sum_{n=1}^{N} r_n = r. \tag{21}$$

Consider first the objective matrix Q. Recall from Section IV that each factor ℓ_k depends *only* on the subset of variables indexed by S_k. This localized dependence determines the sparsity pattern of the corresponding data matrix $Q_k \in \mathbb{S}^n$ for each factor k. Specifically, since factor k cannot couple variables outside X_{S_k} the following holds:

$$(Q_k)_{i,j} = 0 \text{ whenever } (i,j) \notin S_k \times S_k \tag{22}$$

This ensures that each Q_k has block structure reflecting the corresponding factor's variable dependencies, resulting in the full objective matrix decomposition

$$Q = \sum_{k=1}^{K} Q_k = \sum_{k=1}^{K} \begin{bmatrix} (Q_k)_{1,1} & \cdots & (Q_k)_{1,N} \\ \vdots & \ddots & \vdots \\ (Q_k)_{N,1} & \cdots & (Q_k)_{N,N} \end{bmatrix}. \tag{23}$$

Using the bilinearity property of the Frobenius inner product, the objective in (12) decomposes to

$$\langle Q, XX^\top \rangle = \sum_{k=1}^{K} \sum_{i,j=1}^{N} \langle (Q_k)_{i,j}, X_i, X_j^\top \rangle. \tag{24}$$

Let us now consider the constraint matrices A_m which define the feasible set. Recall that each variable can be varied independently of all others within its own parameter domain $X_n \in \mathcal{X}_n$ as per (11) and that $\mathcal{X}_n$ is a standard smooth manifold whose global geometry and topology are known. We will immediately leverage this *variable independence* to reveal the block-diagonal sparsity structure it imposes on each A_m, and exploit the *known manifold geometry* of each $\mathcal{X}_n$ later in Section VI-C.

When constraints act locally in this manner [6], the overall feasible set can be expressed as a *Cartesian product*

$$\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_N, \tag{25}$$

where X_n takes values in the subset $\mathcal{X}_n$.

This product structure implies that each quadratic constraint in (12) involves only a *single* variable X_n. Consequently, we may partition the index set $[M]$ of the constraints into subsets, where $L_n \subseteq [M]$ contains the indices of those constraints associated with only variable X_n. As a result, for $m \in L_n$, the

[5] Our method presupposes that the given factor graph problem *is* representable as a QCQP.

[6] Our method presupposes that the QCQPs constraints are induced exclusively by constraints on individual variables (Section VII-A)—hence are blockwise separable, with no cross-variable coupling.

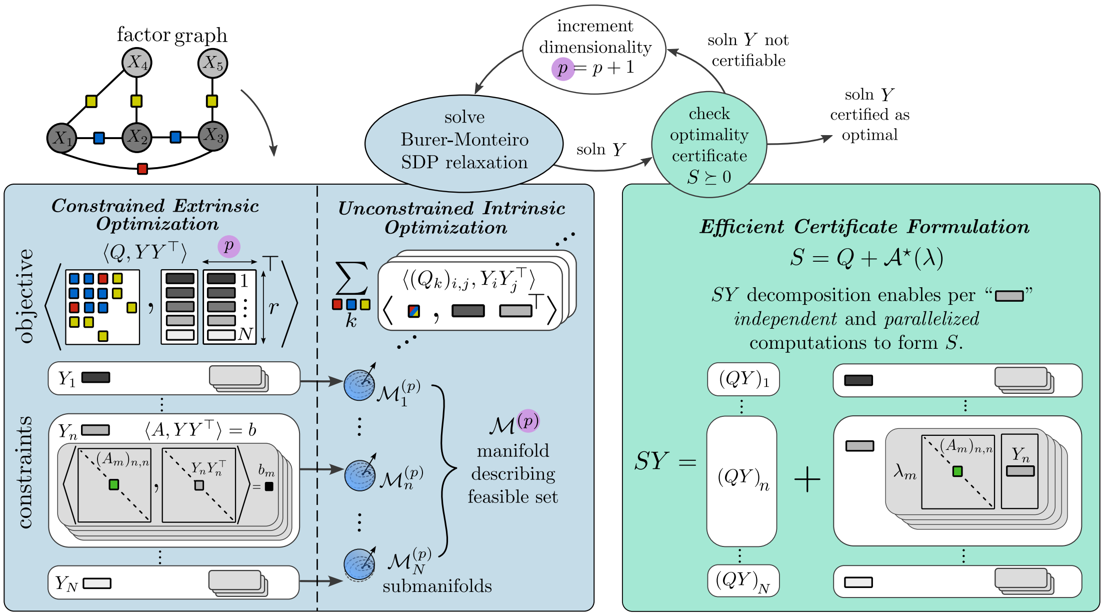

factor graph
X_4
X_5
X_1
X_2
X_3
increment dimensionality $p = p+1$
soln Y not certifiable
solve Burer-Monteiro SDP relaxation
soln Y
check optimality certificate $S \succeq 0$
soln Y certified as optimal
Constrained Extrinsic Optimization
objective
$\langle Q, YY^\top \rangle$
p
r
N
constraints
Y_1
Y_n
$\langle A, YY^\top \rangle = b$
$(A_m)_{n,n}$
$Y_n Y_n^\top$
b_m
Y_N
Unconstrained Intrinsic Optimization
$\sum_k$
$\langle (Q_k)_{i,j}, Y_i Y_j^\top \rangle$
$\mathcal{M}_1^{(p)}$
$\mathcal{M}_n^{(p)}$
$\mathcal{M}_N^{(p)}$
submanifolds
$\mathcal{M}^{(p)}$ manifold describing feasible set
Efficient Certificate Formulation
$S = Q + \mathcal{A}^\star(\lambda)$
SY decomposition enables per " " independent and parallelized computations to form S.
$SY =$
$(QY)_1$
$(QY)_n$
$(QY)_N$
λ_m
$(A_m)_{n,n}$
Y_n

Fig. 2: Overview of the Certi-FGO framework for certifiable factor graph optimization via. Riemannian staircase. Certi-FGO takes in a QCQP-representable factor graph problem and formulates its corresponding rank-p Burer–Monteiro factored SDP relaxation (left) which naturally retains the underlying factor graph structure: objective Q has sparse block structure in direct correspondence to per-factor block-variable dependencies and constraints A are block-separable and thus have a corresponding block-diagonal sparsity pattern. Consequently, each factor in the objective decomposes across block-variables, and constraints on each block-variable Y_n define a submanifold $\mathcal{M}_n^{(p)}$. The entire feasible set is captured by the product manifold $\mathcal{M}_n^{(p)}$ (34). This enables transformation of the constrained BM problem to an unconstrained intrinsic form with two key benefits: (1) automatic problem formulation directly from (lifted) factor graph structure, and (2) efficient solves via local Riemannian optimization. Following the Riemannian staircase procedure (top), solutions are realized as a sequence of local optimizations over lifted factor graphs, interleaved with certification steps. For computing the optimality certificate S (right), the block-separable structure enables decomposition into N independent problems that can be solved efficiently in parallel.

constraint matrix A_m is block diagonal with a single nonzero block $(A_m)_{n,n}$ the (n,n)-th position:

$$A_m = \begin{bmatrix} 0 \in \mathbb{R}^{d_1 \times d_1} & & & & \\ & \ddots & & & \\ & & (A_m)_{n,n} & & \\ & & & \ddots & \\ & & & & 0 \in \mathbb{R}^{d_N \times d_N} \end{bmatrix}. \tag{26}$$

Exposing the block structure, original constraints of QCQPs in (12):

$$\langle A_m, X\,X^\top \rangle = b_m,\ \forall m \in [M]. \tag{27}$$

can be rewritten as:

$$\langle (A_m)_{n,n}, X_n\,X_n^\top \rangle = b_m,\ \forall m \in L_n,\ \ n \in [N]. \tag{28}$$

In summary, the factor graph structure of the MLE (11) induces a block decomposition in the QCQP (12) with respect to block-row variables X_n written as:

$$\begin{aligned} \mathcal{G}(X) = \min_{X \in \mathbb{R}^{r \times d}} \quad & \sum_{k=1}^{K} \sum_{i,j=1}^{N} \langle (Q_k)_{i,j}, X_i\,X_j^\top \rangle \\ \text{s.t.} \quad & \langle (A_m)_{n,n}, X_n\,X_n^\top \rangle = b_m, \\ & \forall m \in L_n,\ \ n \in [N]. \end{aligned} \tag{29}$$

B. Shor's Relaxation over Factor Graphs

Factor graph structure in the QCQP (29) is preserved under Shor's relaxation (Section V-A). Specifically, lifting $X \in \mathbb{R}^{r \times d}$ to its Gram matrix $Z = X\,X^\top \in \mathbb{S}_+^r$, effectively dropping the rank-d constraint, yields a convex SDP relaxation (13), which preserves the sparsity structure of the original problem.

Under block-row partition (21), the lifted matrix Z takes the form:

$$Z \ = \ X\,X^\top = \begin{bmatrix} X_1\,X_1^\top & \cdots & X_1\,X_m^\top \\ \vdots & \ddots & \vdots \\ X_m\,X_1^\top & \cdots & X_m\,X_m^\top \end{bmatrix}, \tag{30}$$

where $X_i\,X_j^\top \ = \ Z_{i,j}$. Substituting Z blockwise into (29) yields the SDP relaxation:

$$\begin{aligned} \mathcal{G}(Z) = \min_{Z \in \mathbb{S}_+^r} \quad & \sum_{k=1}^{K} \sum_{i,j=1}^{N} \langle (Q_k)_{i,j}, Z_{i,j} \rangle \\ \text{s.t.} \quad & \langle (A_m)_{n,n}, Z_{n,n} \rangle = b_m, \\ & \forall m \in L_n,\ \ n \in [N]. \end{aligned} \tag{31}$$

Observe that the objective remains sparse and factor-separable and constraints remain block-diagonal, confirming that Shor's relaxation preserves the factor graph structure.

C. BM factorization over Factor Graphs

We show Burer-Monteiro factorization preserves the structure of the underlying factor graph—the objective remains sparse with respect to lifted variables and separable across lifted factors, while constraints remain block-separable despite dimensional lifting. Furthermore, we demonstrate how the resulting factor graph problem can be efficiently optimized using Riemannian optimization on the induced lifted manifolds.

For low-rank BM factorization, consider $Z = YY^\top$ with *lifted variable* $Y \in \mathbb{R}^{r \times p}$, p specifying the rank of the lifting $d < p \ll n$. As in (21), where we obtained N block-row variables, we partition Y to obtain N *lifted block row variables* $Y_n \in \mathbb{R}^{r_n \times p}$, which directly correspond to the original factor-graph variables X_n:

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_N \end{bmatrix}, \quad \text{where} \quad \sum_{n=1}^{N} r_n = r. \tag{32}$$

Similar to (30), the factorization $Z = YY^\top$ allows us to define blocks $Z_{i,j} = Y_i Y_j^\top$. Applying BM factorization through blockwise substitution of $Y_i Y_j^\top$ into (31) yields the nonconvex problem:

$$\begin{aligned} \mathcal{G}(Y) = \min_{Y \in \mathbb{R}^{r \times p}} \quad & \sum_{k=1}^{K} \sum_{i,j=1}^{N} \langle (Q_k)_{i,j}, Y_i Y_j^\top \rangle \\ \text{s.t.} \quad & \langle (A_m)_{n,n}, Y_n Y_n^\top \rangle = b_m, \\ & \forall m \in L_n, \ n \in [N]. \end{aligned} \tag{33}$$

The sparsity and block-separable structure of this BM factorization (33) *exactly matches* the one assumed in the factor graph decomposition (11) for the initial decomposed QCQP (29). We show next how this structure is leveraged to enable efficient optimization and certification of global optimum.

Rather than employing traditional nonlinear programming solvers for the constrained nonconvex problem (33), we leverage *intrinsic optimization*—recasting the problem as unconstrained optimization on the manifolds defined by the variable constraints—to obtain substantial efficiency gains.

After applying BM factorization (33), the constraint structure remains unchanged from the original QCQP (12): each constraint still acts on only a single (block) variable. Equivalently, each variable can be varied independently within its own domain $X_n \in \mathcal{X}_n$, which has known manifold geometry, and the overall feasible set can be expressed as a Cartesian product of the individual variable domains (25). It follows that for a rank-p BM problem, the overall *lifted feasible set*, manifold $\mathcal{M}^{(p)}$, for our intrinsic reformulation must also be a Cartesian product of individual lifted variable domains, manifolds $\mathcal{M}_n^{(p)}$:

$$\mathcal{M}^{(p)} := \mathcal{M}_1^{(p)} \times \cdots \times \mathcal{M}_N^{(p)}. \tag{34}$$

such that $Y_n \in \mathcal{M}_n^{(p)}$ and each constraint acting on lifted variable Y_n can be interpreted as defining the lifted smooth embedded sub-manifold:

$$\begin{aligned} \mathcal{M}_n^{(p)} := & \{ Y_n \in \mathbb{R}^{r_n \times p} : \langle (A_m)_{n,n}, Y_n Y_n^\top \rangle = b_m \}, \\ & \forall m \in L_n, n \in [N]. \end{aligned} \tag{35}$$

The lifted objective (33) is a sum over individual *lifted factors* $\sum_{i,j=1}^{N} \langle (Q_k)_{i,j}, Y_i Y_j^\top \rangle$, and inherits the same sparse dependency structure from the original per factor data matrices Q_k (24), ensuring the reformulated problem maintains a factor graph representation with variables and factors in *one-to-one* correspondence with the original factor graph MLE (11). This leads to the intrinsic formulation:

$$\mathcal{G}(Y) = \min_{Y \in \mathcal{M}^{(p)}} \sum_{k=1}^{K} \sum_{i,j=1}^{N} \langle (Q_k)_{i,j}, Y_i Y_j^\top \rangle = \langle Q, YY^\top \rangle. \tag{36}$$

where each $Y_n \in \mathcal{M}_n^{(p)}$. In this intrinsic viewpoint, the constraints are absorbed into the geometry of the search space: each Y_n evolves on its own manifold $\mathcal{M}_n^{(p)}$, and the overall feasible region is the Cartesian product $\mathcal{M}^{(p)}$. This reformulation yields an unconstrained optimization problem on a product of smooth manifolds, enabling the use of standard Riemannian optimization methods [17], [36].

In summmary, a rank-p BM factorization of QCQP with factor graph structure induces a set of *lifted factors* (also referred to as certifiable factors) acting on *lifted variables* $Y_n \in \mathcal{M}_n^{(p)}$. These *lifted factor graphs* can be efficiently solved using Riemannian optimization techniques [17], [36]. Section VII provides explicit detail on the construction of lifted factors for several commonly used measurement models in robotics.

D. Optimality Certification over Factor Graphs

In this subsection, we demonstrate that the Lagrange multipliers used in the certificate matrix S (18) possess a block-diagonal structure aligned with the variable partition, causing the linear system characterizing these multipliers to decompose into N *independent* linear systems. Consequently, the Lagrange multipliers λ for each variable block can be computed independently and in parallel, allowing us to solve them (and form the certificate matrix S) in a completely decoupled manner. This structural property reflects the sparsity of the factor graph, allowing seamless and efficient integration of optimality certification and the saddle-escape procedure described in Section V-C.

Theorem 3 (Exploiting block-diagonal structure of S to efficiently obtain least-squares Lagrange multipliers): Let $Y^\star$ be a candidate solution obtained from local-search optimization on the BM problem. Then the corresponding least-squares Lagrange multipliers possess a block-diagonal structure aligned with the variable partition, such that the multipliers for each variable block can be computed independently and in parallel across all variables.

Proof. Using the constraint reformulation based on block variables (28), we can rewrite the adjoint operator in the certificate matrix S (18) accordingly as

$$\mathcal{A}^\star(\lambda) = \sum_{m=1}^{M} \lambda_m A_m = \sum_{n=1}^{N} \sum_{m \in L_n} \lambda_m A_m. \tag{37}$$

Since the constraints are block-separable each A_m is block diagonal with a single nonzero block in the (n, n)-th position for all $m \in L_n$ (see Section VI-A), the matrix product $\mathcal{A}^\star(\lambda)Y$ in (20) separates across the N block-variables as follows:

$$\begin{aligned}\mathcal{A}^{\star}(\lambda)Y &= \Big(\sum_{k=1}^{K}\sum_{m\in L_n}\lambda_m A_m\Big)\begin{bmatrix}Y_1\\ \vdots\\ Y_N\end{bmatrix}\\ &= \sum_{k=1}^{K}\sum_{m\in L_n}\lambda_m\begin{bmatrix}\ddots & & \\ & (A_m)_{n,n} & \\ & & \ddots\end{bmatrix}\begin{bmatrix}\vdots\\ Y_n\\ \vdots\end{bmatrix} \qquad (38)\\ &= \begin{bmatrix}\Big(\sum_{m\in L_1}\lambda_m\,(A_m)_{1,1}\Big)Y_1\\ \vdots\\ \Big(\sum_{m\in L_N}\lambda_m\,(A_m)_{N,N}\Big)Y_N\end{bmatrix}.\end{aligned}$$

Given this and partitioning the matrix product $QY \in \mathbb{R}^{r\times p}$ into N block row variables $(QY)_n$ corresponding to block variables Y_n we rewrite the least squares problem (20) as:

$$\lambda^{\star} = \arg\min_{\lambda\in\mathbb{R}^M}\left\|\begin{bmatrix}(QY)_1\\ \vdots\\ (QY)_N\end{bmatrix} + \begin{bmatrix}\sum_{m\in L_1}\lambda_m(A_m)_{1,1}Y_1\\ \vdots\\ \sum_{m\in L_N}\lambda_m(A_m)_{N,N}Y_N\end{bmatrix}\right\|_F^2 \qquad (39)$$

which admits the block-wise decomposition for each block variable $n\in[N]$

$$\lambda^{\star}_{L_n} = \arg\min_{\lambda_{L_n}}\left\|(QY)_n + \sum_{m\in L_n}\lambda_m(A_m)_{n,n}Y_n\right\|_F^2, \qquad (40)$$

where λ_{L_n} is the subset of Lagrange multipliers corresponding to constraints indexed by $m\in L_n$, i.e. constraints that involve variable block n. Note that this decomposition is enabled by the block-separable structure of the constraints.

The global multiplier vector $\lambda\in\mathbb{R}^M$ collects all individual multipliers λ_m for $m=[M]$:

$$\lambda = \begin{bmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_M\end{bmatrix}. \qquad (41)$$

Thus, the least-squares problem (20) decomposes across block-variables, enabling the multipliers for each block to be computed *independently and in parallel* as per (40). □

Theorem 3 shows that the block-separable constraint structure inherited from the factor graph directly implies that computing the least-squares Lagrange multipliers decomposes into N independent, smaller least-squares problems—one for each variable block Y_n (40). This decomposition has significant practical implications: rather than solving a single large least-squares problem of dimension M, we can solve N independent least squares problem of smaller dimension $|L_n|$ (recall $L_n\subseteq[M]$) for each block-variable n in parallel, each involving only the constraints associated with given variable block. Our implementation exploits this decomposition directly, enabling efficient certificate computation that scales with the number of block-variables.

The solution of (40) depends on the specific constraint geometry. Fortunately, for the orthogonality constraints common in robotics and computer vision, we can derive *closed-form* expressions for each blockwise adjoint mapping $\mathcal{A}^{\star}(\lambda_{L_n})\in\mathbb{S}^{r_n}$ to avoid the need to explicitly solve for each set of blockwise multipliers λ_{L_i} as in (40) (see derivation in Appendix A). We make use of block-separable and orthogonal constraint structure in our implementation.

E. Riemannian Staircase over Factor Graphs

In this subsection, we present our complete framework for implementing the Riemannian staircase algorithm within the factor-graph setting. As described in Section V-D, Riemannian staircase provides a complete certifiable optimization pipeline integrating BM factorization, verification, and saddle-point escape. Since factor graph structure is preserved as described in Sections VI-A-VI-C, we can use lifted factors and variables to embed these modules directly into existing factor graph solvers [7] for a "plug-and-play" certifiable estimation framework over factor graphs.

Algorithm 2 summarizes this *Certifiable Factor-Graph Optimization* or `Certi-FGO` framework and is shown pictorially in Fig. 2. Starting from initial estimates, measurements, and an initial rank-p, the algorithm constructs a rank-p lifted factor graph via Burer-Monteiro factorization and performs local optimization over the corresponding lifted manifold $\mathcal{M}^{(p)}$. The VERIFICATION() routine obtains the smallest eigenpair (eigenvalue and corresponding eigenvector) of the certificate matrix (18) using [40]—if positive, global optimality is certified. Otherwise, the rank is increased and SADDLEESCAPE() initializes a new iterate using the negative-curvature direction. These steps repeat until certification succeeds, after which an SVD-based projection [43] maps the lifted solution back to the original ambient Euclidean space.

The key practical advantage of `Certi-FGO` is its seamless integration with existing factor-graph pipelines [37]. With intrinsic Riemannian optimization formulation, certifiable estimation reduces to standard local-search based factor-graph optimizations on the lifted variables and factors, with existing local-search frameworks remaining fully applicable.

VII. COMMON EXAMPLES OF LIFTED FACTORS

In this section, we build upon theoretical results from the previous section by deriving *lifted factors* for measurement models commonly encountered in robotics navigation and mapping in certifiable estimation via rank-p BM factorization. Specifically, we construct lifted versions of three widely used factors in estimation tasks: 1) *relative rotation*, 2) *relative translation*, and 3) *point-to-point range* measurements from *lifted* rotation, translation variables and unit-sphere variables. These lifted factors enable a broad class of estimation problems (see Section VIII) that can be implemented directly and solved with standard factor-graph solvers via local-search technique, while admitting verified optimal guarantees.

We begin by reviewing the standard variable and measurement (factor) models, then define their lifted counterparts. For the lifting procedure, see the initialization of higher-rank instances in the saddle-escape theory (Theorem 2). Specifically, within the Burer–Monteiro factorization, we construct lifted factors ℓ_k from lifted variables Y_n.

Algorithm 2: `Certi-FGO`: Certifiable Factor Graph Optimization

```
Input: Initial values $Y = \{Y_n\}_{n=1}^{N}$, initial rank-$p$,
       measurements $Z = \{Z_k\}_{k=1}^{K}$
Output: A feasible estimate $\hat{X}$ and the lower bound
        $f^*_{\mathrm{SDP}}$ on its optimal value
function CertiFGO($Y$, $Z$, $p$):
    // Lift variables to rank-p
    $Y_p \leftarrow$ Lift($Y$)
    while true do
        // Construct lifted factor graph at level-p
        $\mathcal{G}_p \leftarrow$ ConstructGraph($Z, p$)
        $Y_p^\star \leftarrow$ LocalOptimization($\mathcal{G}_p, Y_p$)
        // Construct certificate and compute minimum
           eigenvalue
        $(\lambda_{\min}, v_{\min}) \leftarrow$ Verification($\mathcal{G}_p, Y_p$)
        if $\lambda_{\min} > 0$ then
            return $\{\hat{Y},\ f^*_{\mathrm{SDP}}\}$;
        else
            $p \leftarrow p + 1$
            $Y_{p+1} \leftarrow$ SaddleEscape($Y_p^\star, v_{\min}$)
    return $\{\hat{Y},\ f^*_{\mathrm{SDP}}\}$;
$\hat{X} \leftarrow$ RoundSolution($\hat{Y}$);
return $\{\hat{X},\ f^*_{\mathrm{SDP}}\}$;
```

A. Lifted Variables

In intrinsic Riemannian optimization, the rank-p BM factorization replaces each original variable with a higher rank *lifted* instance that lives on a higher-dimensional manifold. This subsection describes the definition of lifted rotation and translation variables used in the rank-p formulation.

1) Rotation Variables: Rotation variables $R_i \in O(d)$ satisfy the orthogonality constraint $R_i R_i^\top = I_d$. Under the rank-p BM factorization with $p \geq d$, we represent each rotation by a lifted variable $Y_i \in \mathrm{St}(p, d)$. The manifold $\mathrm{St}(p, d)$ (7) generalizes the rotation group: when $p = d$, $\mathrm{St}(d, d) = O(d)$. The additional degrees of freedom for $p > d$ enable the Riemannian Staircase to escape saddle points during optimization.

Unit-sphere special case. The unit-sphere variable $k_{ij} \in S^{d-1}$, used in range measurements (see Section VII-B3), is a special case of a Stiefel variable with one column, $\mathrm{St}(d, 1) = \{k \in \mathbb{R}^{d\times 1} : k^\top k = 1\}$. A corresponding lifted variable is $b_{ij} \in S^{p-1}$ with $p \geq d$.

2) Translation Variables: Translation variables $t_i \in \mathbb{R}^d$ represent positions of robots or landmarks in Euclidean space. Under rank-p BM factorization, each translation is lifted to a higher-dimensional vector $u_i \in \mathbb{R}^p$ with $p \geq d$.

B. Standard Measurement Models and Factors

1) Relative Rotation Measurement: A *relative rotation* describes the orientation change required to align one coordinate frame with another. Indeed, such measurements arise in all state estimation tasks either from inertial measurement units (IMUs) that capture ego motion or from multi-view geometry [44], and they constitute the primary source of nonconvexity in the underlying optimization. As a prototypical instance of a problem defined entirely by relative rotations, rotation averaging (RA) is a foundational challenge in both computer vision and robotics [45], [46].

Given rotation variables $R_i, R_j \in SO(d)$, the relative rotation R_{ij} from frame i to frame j is then:

$$R_{ij} \triangleq R_i^{-1} R_j. \tag{42}$$

In practice, we assume that (noisy) measurements $\tilde{R}_{ij} \in SO(d)$ [7] of relative rotations R_{ij} are sampled from the following probabilistic generative model:

$$\tilde{R}_{ij} = R_{ij}\,\eta_{ij}, \quad \eta_{ij} \sim \mathrm{Langevin}\left(I_d, \kappa_{ij}\right). \tag{43}$$

where $\kappa_{ij} \geq 0$ is the *precision* of this measurement.

The resultant relative rotation factor is thus:

$$\begin{aligned} \ell_k\left(R_i, R_j; \tilde{R}_{ij}\right) &= \kappa_{ij}\|R_i^\top R_j - \tilde{R}_{ij}\|_F^2 \\ &= \kappa_{ij}\|R_j - R_i\tilde{R}_{ij}\|_F^2 \end{aligned} \tag{44}$$

The second equality follows from the invariance of the Frobenius norm under orthogonal transformations.

2) Relative Translation Measurement: A *relative translation* measures the displacement between two points within a given reference frame. Specifically, it can be obtained by decomposing full-pose measurements to isolate the translational component for simplified pose estimation [13], [47], or directly from translation measurements such as landmark observations [21]. These measurements are used in both pose-graph and landmark-based SLAM formulations.

Given a point $t_j \in \mathbb{R}^d$ and a coordinate frame $x_i = (R_i, t_i) \in \mathrm{SE}(d)$, the relative translation of t_j with respect to frame x_i is defined as:

$$t_{ij} \triangleq R_i^\top (t_j - t_i). \tag{45}$$

We assume noisy relative translations $\tilde{t}_{ij} \in \mathbb{R}^d$ modeled as:

$$\tilde{t}_{ij} = t_{ij} + \epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}(0, \tau_{ij}^{-1} I_d), \tag{46}$$

where $\tau_{ij} \geq 0$ denotes the measurement precision. The associated relative translation factor is:

$$\begin{aligned} \ell_k(t_i, t_j; \tilde{t}_{ij}) &= \tau_{ij}\|R_i^\top (t_j - t_i) - \tilde{t}_{ij}\|_2^2 \\ &= \tau_{ij}\|t_j - t_i - R_i\tilde{t}_{ij}\|_2^2, \end{aligned} \tag{47}$$

where the second equality uses the invariance of the Euclidean norm under orthogonal transformations.

[7] Throughout, we take the rotational variables in the QCQP to lie in the orthogonal group $O(d)$—i.e., we drop the determinant constraint $\det(R) = +1$ that defines $SO(d)$. Measurements and kinematic relations are still modeled on $SO(d)$; the relaxation only applys the (unlifted) rotation variables in the Burer–Monteiro parameterization. After optimization we project the estimate onto $SO(d)$. This projection preserves the globally optimal $SE(d)$ solution under our certificate; see App. C.1 of [27].

3) Range Measurement: Range measurements provide point-to-point distances, typically obtained from low-cost, low-power sensors such as Ultra-Wideband (UWB) radios [48] or Wi-Fi Positioning Systems (WPS) [49]. Specifically, in state estimation, range measurements can encode either the distance between two robot poses or between a pose and a landmark point, making them versatile factors in mapping and localization. These sensors offer sufficient accuracy for state-estimation tasks and are popular in applications such as pure-range localization [50], range-aided SLAM [29], [51], and multi-agent relative localization [52].

Given two points $t_i, t_j \in \mathbb{R}^d$, the *range* between them is:

$$r_{ij} \triangleq \|t_i - t_j\|_2. \tag{48}$$

We model the noisy measurement $\tilde{r}_{ij}$ as a sample from the following probabilistic generative process:

$$\tilde{r}_{ij} = r_{ij} + \nu_{ij}, \qquad \nu_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2), \tag{49}$$

where σ_{ij}^2 is the variance of range measurement.

The corresponding factor of a range measurement is given by:

$$\ell_k\left(t_i, t_j; \tilde{r}_{ij}\right) = \frac{1}{\sigma_{ij}^2}\left(\|t_j - t_i\|_2 - \tilde{r}_{ij}\right)^2. \tag{50}$$

Note that (50) is *not* actually a quadratic function (due to the presence of the unsquared norm). Nevertheless, a technique proposed in [29], [53] enables us to rewrite (50) as an equivalent quadratic term by introducing an auxiliary unit vector $k_{ij} \in S^{d-1}$ that effectively models the *bearing* from point t_i to t_j:

$$\ell_k\left(t_i, t_j, k_{ij}; \tilde{r}_{ij}\right) = \frac{1}{\sigma_{ij}^2}\|t_j - t_i - \tilde{r}_{ij}k_{ij}\|_2^2. \tag{51}$$

C. Lifted Factors

1) Lifted Relative Rotation Factor: When we apply Shor's relaxation followed by BM factorization, the factor (44) is transformed to the following *lifted* factor in the resulting rank-p BM factorization as:

$$\ell_k\left(Y_i, Y_j; \tilde{R}_{ij}\right) = \kappa_{ij}\|Y_j - Y_i\tilde{R}_{ij}\|_F^2, \tag{52}$$

where lifted variables $Y_i, Y_j \in \mathrm{St}(p,d)$ are higher-dimensional lifts of the original rotation $R_i, R_j \in \mathrm{O}(d)$.

2) Lifted Relative Translation Factor: To obtain a lifted factor representation, we apply Shor's relaxation followed by rank-p BM factorization, yielding the rank-p lifted factor:

$$\ell_k(u_i, u_j; \tilde{t}_{ij}) = \tau_{ij}\|u_j - u_i - Y_i\tilde{t}_{ij}\|_2^2, \tag{53}$$

where $u_i, u_j \in \mathbb{R}^p$ are lifted translations of the original vectors $t_i, t_j \in \mathbb{R}^d$.

3) Lifted Range Measurement Factors: Applying Shor's followed by Burer-Monteiro factorization to reformulation (51), we obtain the following lifted factor:

$$\ell_k\left(u_i, u_j, b_{ij}; \tilde{r}_{ij}\right) = \frac{1}{\sigma_{ij}^2}\|u_j - u_i - \tilde{r}_{ij}b_{ij}\|_2^2, \tag{54}$$

where $u_i, u_j \in \mathbb{R}^p$ are lifted versions of the original translation vectors $t_i, t_j \in \mathbb{R}^d$ and $b_{ij} \in \mathbb{S}^{p-1}$ is the lifted version of $k_{ij} \in \mathbb{S}^{d-1}$.

VIII. Experiments

In this section, we evaluate our certifiable factor graphs on both synthetic and real-world SLAM datasets. Using the common certifiable factors introduced in Section VII, we firstly construct three representative SLAM instances commonly used in practice and compare our approach against the corresponding state-of-the-art hand-crafted solvers on random initialization. Beyond these worst-case comparisons—which demonstrate the ability to recover *verifiably globally optimal* solutions—we further benchmark against a GTSAM-based local optimizer with a odometry-initialization to assess practical efficiency. The results indicate that our method achieves runtime comparable to fast factor-graph local methods while providing certificates of global optimality.

A. Experiments Details

Our method is implemented in C++ using GTSAM, and we solve the resulting problems with the Levenberg–Marquardt (LM) algorithm. Table III reports results under random initialization, and Table IV reports results under odometry-based initialization. We briefly summarize common settings below; instance-specific parameters are reported with each experiment.

Random initialization. We compare against state-of-the-art (SOTA) baselines SE-Sync [27], CPL-SLAM [28], and CORA [29]. All methods (ours and baselines) are initialized by random sampling, with the initial rank p_0 equal to the ambient dimension $d \in \{2, 3\}$. For SE-Sync and CORA, we adopt a translation-explicit formulation. For CPL-SLAM, we report results using the Landmark–SE-Sync variant provided by it, as the publicly released CPL-SLAM implementation did not execute successfully in our environment.

Odometry initialization. For local-search baselines, we run GTSAM's LM solver; where applicable, our method uses the same LM hyperparameters to ensure a fair comparison. Dataset-specific settings (e.g., noise levels, iteration caps) are specified with each example.

Certification criterion. Let S denote the certificate matrix and $f^\star$ the attained objective value. We set the relative eigenvalue margin

$$\eta := \min\{\eta_{\max}, \max\{10^{-6}f^\star, \eta_{\min}\}\}, \tag{55}$$

with $\eta_{\min} = 10^{-3}$ and $\eta_{\max} = 10^{-1}$, and consider the certificate valid when $S + \eta I \succ 0$.

All experiments were conducted on a laptop with an Intel Core i7-11800H CPU and 32 GB RAM, running Ubuntu 22.04.

B. Pose Graph Optimization

In *pose-graph optimization* (PGO), the goal is to estimate poses $x_1, \dots, x_n \in \mathrm{SE}(d)$ from noisy relative measurements $\tilde{x}_{ij} = (\tilde{R}_{ij}, \tilde{t}_{ij}) \in \mathrm{SE}(d)$. Each pose $x_i = (R_i, t_i)$ consists of a rotation $R_i \in \mathrm{SO}(d)$ and translation $t_i \in \mathbb{R}^d$. The problem is represented by a directed pose graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [n]$ are the unknown poses and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ encodes available pairwise measurements. Assuming the noise models in (43) and (46), the corresponding maximum likelihood estimation (MLE) problem is:

TABLE I

PGO DATASETS INFORMATION: SUMMARY OF SYNTHETIC AND REAL-WORLD DATASETS, REPORTING THE DIMENSION, NUMBER OF POSES, AND NUMBER OF RELATIVE-POSE MEASUREMENTS.

	Pose Graph Optimization														
	MIT	CSAIL	Intel	Kitti 05	Manhattan	Kitti 00	City	Ais2klinik	Small Grid	Garage	Sphere	Torus	Cubicle	Grid	Rim
Dim	2D	2D	2D	2D	2D	2D	2D	2D	3D	3D	3D	3D	3D	3D	3D
# Poses	808	1045	1728	2761	3500	4251	10000	15115	125	1661	2500	5000	5750	8000	10093
# Measurements	827	1171	2512	2827	5451	4679	20687	16727	297	6275	4949	10000	7696	22236	18637

TABLE II

LANDMARK AND RANGE-AIDED DATASETS INFORMATION: SUMMARY OF REAL-WORLD DATASETS REPORTING THE DIMENSION, NUMBERS OF POSES AND LANDMARKS, AND COUNTS OF RANGE AND RELATIVE-POSE MEASUREMENTS.

	Landmark SLAM		**Range-Aided SLAM**				
	Victoria [54]	Trees [11]	Goats 16 [55]	Goats 15 [55]	Plaza1 [56]	Plaza2 [56]	Single Drone [29]
Dim	2D	2D	2D	2D	2D	2D	3D
# Poses	6969	10000	201	472	9658	4091	1754
# Landmarks	151	100	4	4	4	4	1
# Measurements	10608	14442	772	1258	12983	5897	3507

Problem 1 *(MLE formulation of PGO).*

$$\begin{aligned} \min_{t_i\in\mathbb{R}^d,\, R_i\in\mathrm{SO(d)}} \sum_{(i,j)\in\mathcal{E}} & \kappa_{ij}\left\|R_j - R_i\tilde{R}_{ij}\right\|_F^2 \\ & + \tau_{ij}\left\|t_j - t_i - R_i\tilde{t}_{ij}\right\|_2^2 \end{aligned} \tag{56}$$

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

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

$$\begin{aligned} \min_{t_i\in\mathbb{R}^p,\, Y_i\in\mathrm{St}(p,d)} \sum_{(i,j)\in\mathcal{E}} & \kappa_{ij}\left\|Y_j - Y_i\tilde{R}_{ij}\right\|_F^2 \\ & + \tau_{ij}\left\| t_j - t_i - Y_i\tilde{t}_{ij}\right\|_2^2 \end{aligned} \tag{57}$$

We set `relative_error` to 10^{-5}, `absolute_error` to 10^{-4}, and `max_iteration` to 100. The LM optimizer terminates when either error tolerance is met.

a) Random initialization.: Table III summarizes PGO results obtained by our method (Algorithm 2). Across all datasets, our solver attains the same verifiably globally optimal objective values as SE-Sync, with successful certification confirming the guarantees of our certifiable factors. Consistent with [27], the SDP relaxation is tight for the evaluated datasets and noise regimes. Differences in runtime (see the *Opt. Time* column) primarily stem from algorithmic design: our implementation uses GTSAM's general-purpose LM with an inexact Hessian and requires relinearization at each iteration, whereas SE-Sync employs a specialized second-order Riemannian trust-region method with analytic derivatives and no relinearization. As a consequence, our approach is generally slower; however, SE-Sync is slower on some real-world datasets (e.g., `Garage`, `Rim`, `Ais2klinik`), while our method is notably slower on synthetic dataset (e.g., `Grid`). Random-initialization experiments thus approximate a practical worst case. The results show that our method provides the same strong guarantees while requiring a much simpler implementation (see VIII-F).

b) Odometry initialization.: Table IV evaluates efficiency under strong initializations. Most datasets are solved in comparable total time, and in many cases the algorithm terminates at the initial rank. Nevertheless, even with odometry-based initialization, local methods may converge to suboptimal solutions. By contrast, our approach certifies optimality; when verification fails, it automatically increases the BMrank via the Riemannian staircase until a certifiable solution is obtained.

C. Landmark SLAM

Because landmark measurements can be modeled as relative-translation factors (47), Landmark SLAM extends the PGO formulation (56) by applying the same translation factor on pose–landmark edges. We set `relative_error` to 10^{-5}, `absolute_error` to 10^{-4}, and `max_iteration` to 150. The LM optimizer terminates when either error tolerance is met.

a) Random initialization.: Table III reports results for Landmark SLAM under random initialization. On both datasets (*Trees*, *Victoria*), our method attains the same verifiably globally optimal objective values as handcrafted baselines, with successful certification confirming the guarantees of our certifiable factors. Runtime differences are primarily attributable to solver design; see Sections VIII-E–2 for discussion.

b) Odometry initialization.: Table IV summarizes results under strong initialization. We initialize poses from odometry and landmarks from ground-truth positions. On *Trees*, our method achieves runtimes comparable to local-search baselines while additionally providing optimality certificates. On *Victoria*, local methods converge to suboptimal solutions, whereas our approach recovers the globally optimal solution with certification.

D. Range-Aided SLAM

Range-aided SLAM extends PGO by incorporating range measurements $\tilde{r}_{ij} \in \mathbb{R}$, modeled by the noise distribution

TABLE III

Results of random initialization: Objective values and runtimes on PGO, landmark SLAM , and range-aided SLAM. We compare state-of-the-art baselines (SE-Sync [27], CPL-SLAM [28], CORA [29]) with our certifiable factor-graph methods.

PGO	**SE-Sync**				**Ours**			
	Obj. Value	Opt. Time	Tot. Time	Term. Rank	Obj. Value	Opt. Time	Tot. Time	Term. Rank
MIT	6.115×10^1	0.38	0.40	4	6.115×10^1	1.15	1.28	4
CSAIL	3.170×10^1	0.23	0.24	4	3.170×10^1	1.52	1.77	4
Intel	5.235×10^1	3.51	3.57	4	5.235×10^1	5.71	6.13	4
Kitti 05	2.765×10^3	1.31	1.36	5	2.765×10^3	15.58	17.35	4
Manhattan	6.431×10^3	2.38	2.43	3	6.431×10^3	17.12	18.28	4
Kitti 00	1.257×10^2	5.09	5.15	4	1.257×10^2	12.03	15.64	4
City	6.386×10^2	87.16	87.68	4	6.386×10^2	29.85	36.61	4
Ais2klinik	1.886×10^2	900.17	900.40	4	1.886×10^2	106.43	121.59	4
Small Grid	1.025×10^3	0.13	0.13	3	1.025×10^3	0.09	0.09	3
Garage	1.263×10^0	75.58	75.74	5	1.263×10^0	17.06	17.83	5
Sphere	1.687×10^3	14.90	15.03	5	1.687×10^3	8.41	9.67	5
Torus	2.423×10^4	14.53	14.78	5	2.423×10^4	106.71	110.26	5
Cubicle	7.171×10^2	52.09	52.23	4	7.171×10^2	49.61	54.56	5
Grid	8.432×10^4	65.56	68.03	5	8.432×10^4	311.27	318.41	3
Rim	5.461×10^3	357.19	357.67	4	5.461×10^3	130.09	145.62	5

Landmark SLAM	**CPL-SLAM**				**Ours**			
	Obj. Value	Opt. Time	Tot. Time	Term. Rank	Obj. Value	Opt. Time	Tot. Time	Term. Rank
Victoria	4.660×10^2	18.68	20.38	4	4.676×10^2	29.21	36.62	4
Trees	6.035×10^2	13.01	14.69	4	6.035×10^2	24.18	28.44	4

Range-Aided SLAM	**CORA**					**Ours**				
	SDP Value	Ref. Value	Opt. Time	Tot. Time	Term. Rank	SDP Value	Ref. Value	Opt. Time	Tot. Time	Term. Rank
Goats 16	3.689×10^3	3.894×10^3	0.77	0.80	4	3.709×10^3	3.894×10^3	2.41	2.54	4
Goats 15	1.607×10^4	1.820×10^4	2.37	2.45	5	1.609×10^4	1.820×10^4	3.62	3.71	4
Plaza1	1.291×10^3	1.332×10^3	13.32	14.93	6	1.286×10^3	1.332×10^3	192.18	206.89	5
Plaza2	7.244×10^2	7.343×10^2	4.67	5.06	3	7.259×10^2	7.343×10^2	48.85	51.04	4
Single Drone	7.223×10^0	7.698×10^0	6.63	7.36	4	7.246×10^0	7.698×10^0	21.06	22.30	5

Opt.Time denotes the sum of the optimization times at each level of the Riemannian staircase. **Tot.Time** is the total computation time, including initialization, optimization, verification, saddle-escape, and rounding. **Term.Rank** is the final rank at which the Riemannian staircase terminates. For PGO and Landmark-SLAM problems, **Obj.Value** coincides with both the QCQP value (12) and the SDP value (13) due to relaxation tightness under moderate conditions. For Range-Aided SLAM, since the relaxation is not always tight, we report both the **SDP Value** and the refined solution value **Ref. Value** (see VIII-D). The solutions of all the experiments above are verified with certification.

in (49). The objective is to estimate poses $x_1, x_2, \dots, x_n \in \mathrm{SE}(d)$ and landmarks $l_1, l_2, \dots, l_n \in \mathbb{R}^d$ that solve Problem 3.

Problem 3 *(MLE formulation of Range Aided SLAM).*

$$\min_{\substack{t_i\in\mathbb{R}^d,\\ R_i\in\mathrm{SO(d)}}} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij}\left\|R_j - R_i\tilde{R}_{ij}\right\|_F^2 + \tau_{ij}\left\|t_j - t_i - R_i\tilde{t}_{ij}\right\|_2^2 + \frac{1}{\sigma_{ij}^2}\left(\|t_j - t_i\|_2 - \tilde{r}_{ij}\right)^2 \tag{58}$$

Similarly, the lifted version can be expressed as:

Problem 4 *(rank-p Burer-Monteiro factorization of Range Aided SLAM).*

$$\min_{\substack{t_i\in\mathbb{R}^p,\\ Y_i\in\mathrm{St}(p,d),\\ b_{ij}\in S^{p-1}}} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij}\left\|Y_j - Y_i\tilde{R}_{ij}\right\|_F^2 + \tau_{ij}\left\|t_j - t_i - Y_i\tilde{t}_{ij}\right\|_2^2 + \frac{1}{\sigma_{ij}^2}\left\|t_j - t_i - \tilde{r}_{ij}r_{ij}\right\|_2^2 \tag{59}$$

Given the poor numerical conditioning of range-aided SLAM, we use stringent stopping criteria: `relative_error` and `absolute_error` are both set to 10^{-8}, and `max_iteration` to 300. The LM optimizer terminates when either error tolerance is met.

Remark 1: Range measurements are scalar and typically constrain the state only weakly, contributing limited information to the system. Consequently, the information matrix is often ill-conditioned and the objective may exhibit broad flat regions. In such regimes, Gauss–Newton or LM may produce only marginal decreases in the cost even far from optimality, so loose tolerances can trigger premature termination. To mitigate this, we adopt *tight* tolerances (relative and absolute errors on the order of 10^{-8}, with dataset-specific adjustments) and a higher iteration cap (300), permitting additional relinearizations and optimization steps that help escape flat regions.

a) Random initialization.: Table III reports results under random initialization. Because the relaxation is not always

TABLE IV

Results of odometry initialization: Objective values and runtimes on PGO, landmark SLAM, and range-aided SLAM. We compare a local-search–based solver (GTSAM [7]) with our certifiable factor-graph methods. All methods are initialized with odometry measurements for poses and ground-truth landmark positions.

PGO	**Ours**						**GTSAM**		
	Obj. Value	Opt. Time	Tot. Time	Init. Rank	Term. Rank	Verif.	Obj. Value	Tot. Time	Verif.
MIT	6.115×10^{1}	0.54	0.72	2	4	✓	1.298×10^{3}	0.17	✗
CSAIL	3.170×10^{1}	0.05	0.10	2	2	✓	3.170×10^{1}	0.06	✓
Intel	5.235×10^{1}	0.08	0.21	2	2	✓	5.235×10^{1}	0.09	✓
Kitti 05	2.765×10^{3}	0.33	1.093	2	2	✓	2.765×10^{3}	0.19	✓
Manhattan	6.431×10^{3}	0.46	0.93	2	2	✓	6.431×10^{3}	0.50	✓
Kitti 00	1.257×10^{2}	0.33	1.09	2	2	✓	1.257×10^{2}	0.37	✓
City	6.386×10^{2}	0.60	4.13	2	2	✓	6.386×10^{2}	0.73	✓
Ais2klinik	1.886×10^{2}	114.85	129.37	2	4	✓	1.325×10^{3}	56.05	✗
Small Grid	1.025×10^{3}	0.02	0.02	3	3	✓	1.025×10^{3}	0.02	✓
Garage	1.263×10^{0}	0.17	0.45	3	3	✓	1.263×10^{0}	0.25	✓
Sphere	1.687×10^{3}	0.29	0.86	3	3	✓	1.687×10^{3}	0.33	✓
Torus	2.423×10^{4}	29.40	33.15	3	5	✓	5.276×10^{4}	5.96	✗
Cubicle	7.171×10^{2}	0.89	3.46	3	3	✓	7.171×10^{2}	1.1	✓
Grid	8.432×10^{4}	103.183	110.276	3	3	✓	8.432×10^{4}	106.85	✓
Rim	5.461×10^{3}	2.52	10.86	3	3	✓	5.461×10^{3}	2.564	✓

Landmark SLAM	**Ours**						**GTSAM**		
	Obj. Value	Opt. Time	Tot. Time	Init. Rank	Term. Rank	Verif.	Obj. Value	Tot. Time	Verif.
Victoria	4.661×10^{2}	18.92	25.72	2	4	✓	1.691×10^{4}	7.34	✗
Trees	6.036×10^{2}	0.63	4.52	2	2	✓	6.036×10^{2}	0.70	✓

Range-Aided SLAM	**Ours**							**GTSAM**		
	SDP Value	Ref. Value	Opt. Time	Tot. Time	Init. Rank	Term. Rank	Verif.	Obj. Value	Tot. Time	Verif.
Goats 16	3.709×10^{3}	3.894×10^{3}	2.14	2.26	2	4	✓	4.087×10^{5}	0.57	✗
Goats 15	1.623×10^{4}	1.820×10^{4}	5.56	6.13	2	4	✓	1.665×10^{6}	0.51	✗
Plaza1	1.289×10^{3}	1.332×10^{3}	65.63	73.27	3	4	✓	1.337×10^{6}	157.99	✗
Plaza2	7.259×10^{2}	7.343×10^{2}	50.23	53.37	2	4	✓	3.138×10^{5}	69.97	✗
Single Drone	7.244×10^{0}	7.698×10^{0}	19.07	20.61	3	5	✓	1.867×10^{2}	2.74	✗

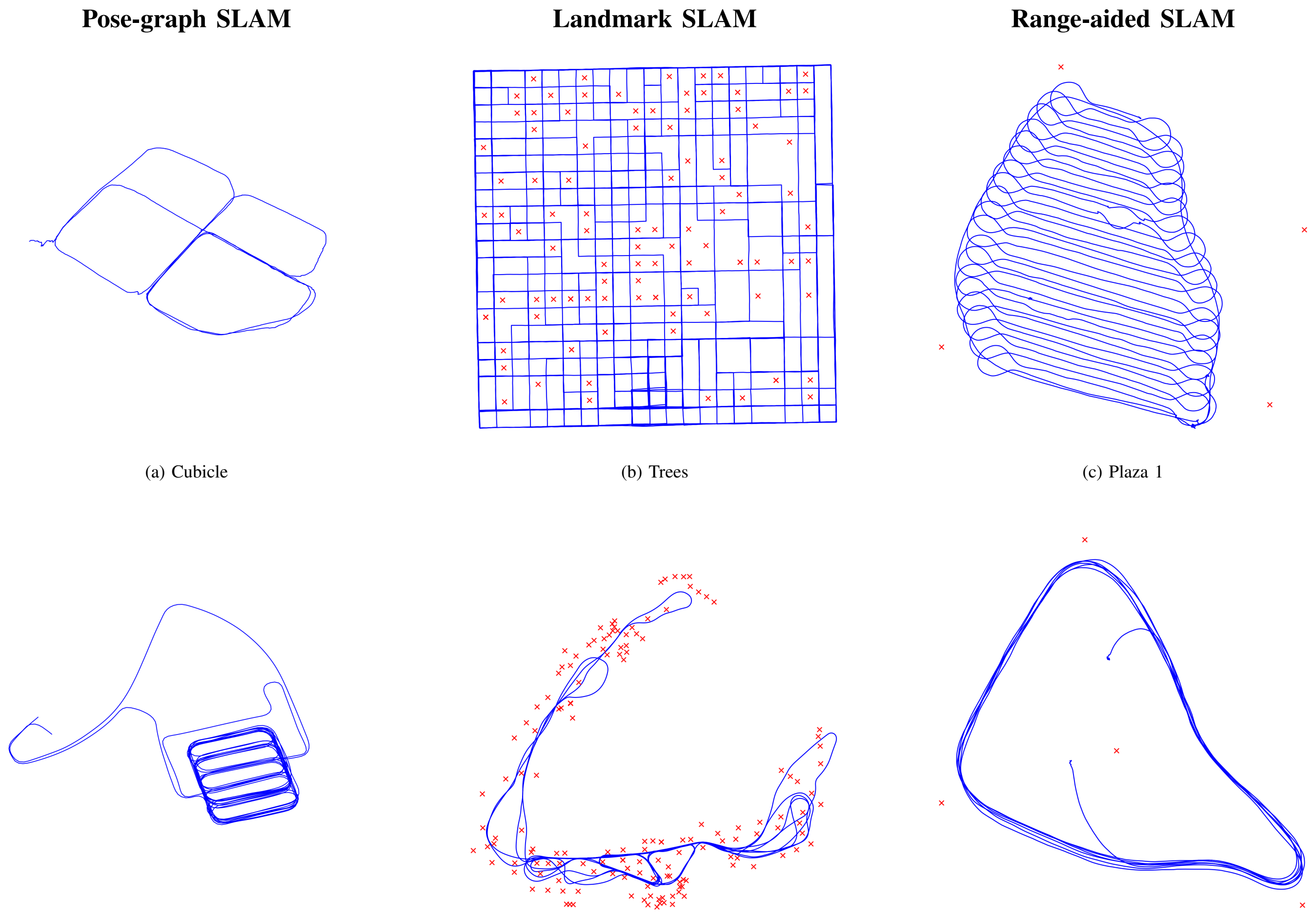
Our method provides certificates of global optimality, whereas the GTSAM local-search solver does not. Accordingly, we apply our certifier to the solutions returned by GTSAM. The verification result column (**Verif.**) reports the certification status for each solution: (✓) indicates a verified globally optimal solution, while (✗) indicates a failure to certify and thus a sub-optimal solution.

exact, an optimality gap can arise between the semidefinite program (SDP) and the original nonconvex objective. Both methods perform a *solution refinement* step that projects the SDP solution onto the feasible set of the original problem and re-optimizes: we report the refined objective values as *Refined Obj. Val.*, and the pre-refinement SDP values as *SDP Obj. Val.* Our SDP objectives are slightly higher than CORA's but remain globally certified. This discrepancy stems from the slower convergence of a general-purpose, inexact LM routine compared to CORA's truncated-Newton Riemannian trust-region (RTR) method, especially in the presence of ill-conditioned lifted range factors. As a consequence, our solver often climbs to higher BM-ranks before certification is achieved. Since both approaches share the same SDP-based initialization, the refined objective values coincide. In runtime, our method is markedly slower on certain datasets, primarily due to solver inefficiencies under ill-conditioned instances.

b) Odometry initialization.: Table IV summarizes results under strong initialization (poses from odometry, landmarks from ground truth). Notably, no local method produced a certifiably globally optimal solution on the range-aided SLAM tasks, indicating that local optimization can stall at suboptimal minima even with good initial guesses. By contrast, our approach returns certified solutions and, with good initialization, achieves shorter total runtime than in the random-initialization setting.

E. Discussion

Across all benchmarks, our certifiable factor-graph approach recovers solutions indistinguishable (within numerical termination tolerances) from those produced by custom-tailored certifiable methods, as expected since all approaches ultimately



(a) Cubicle

(b) Trees

(c) Plaza 1

(d) Garage

(e) Victoria

(f) Plaza 2

Fig. 3: Globally optimal results on select SLAM benchmark datasets encompassing pose-graph optimization, landmark, and range-aided problems. The blue denotes the trajectory that the robot traversed, while the red denotes the landmarks utilized in the case of landmark and range-aided SLAM.

solve the same underlying SDP relaxations. Nevertheless, our method is often—though not always—slower to converge in both wall-clock time and iteration count. This is unsurprising for several reasons:

1) **Implementation Specificity.** Generally speaking, the comparison baselines rely on problem-specific, highly optimized, hand-crafted implementations, whereas our pipeline deliberately uses a general-purpose backend.
2) **Different Optimization Model.** The baselines employ truncated-Newton Riemannian trust-region (RTR) algorithms with an *exact* Riemannian Hessian, whereas we rely on GTSAM's Levenberg–Marquardt method, which uses a regularized Gauss–Newton approximation $J^\top J + \lambda$ (where J is the Jacobian and λ is the damping factor) to approximate the Hessian, combined with direct (factorization-based) linear solves.
3) **Numerical Conditioning.** The performance gap is most pronounced in range-aided SLAM, where poor conditioning amplifies the difference between exact-Hessian RTR and LM. Similar behavior has also been observed in recent distributed CORA results [32].

Despite the slight difference in computational performance, the modest overhead is typically acceptable given the substantial reduction in implementation effort and the plug-and-play integration enabled by our framework. This trade-off could be further improved by designing dedicated solvers tailored to certifiable estimation, even within the general-purpose local-search optimization paradigm adopted here.

F. Implementation Effort and Practicality

To underscore the practicality of our approach, we quantify the implementation effort required for the examples discussed above. In each case, the proposed method reduces to assembling specific SLAM tasks using plug-and-play certifiable factors, while the underlying infrastructure automatically handles relaxation, sparsity-aware solvers, and certification—making the overall process as straightforward as standard local-search-based factor-graph optimization. In contrast, hand-crafted alternatives typically require designing problem-specific relaxations and custom solvers, involving substantial manual engineering effort. According to reports from the authors of prior systems (personal communication; details withheld to preserve anonymity), reproducing comparable experiments with such custom solvers often requires *several weeks to a month* of engineering per problem class. In comparison, assembling the GTSAM-based certifiable estimator can typically be completed in *a day or less*. This gap translates into significantly reduced development time, making the certifiable factor-graph approach practical for real-world deployment. *The central goal of this work is to lower the barrier to entry for non-expert users and to advance certifiable algorithms toward practical, everyday applications. It aims to democratize access to certifiably correct estimation, support seamless integration into robotic systems that already leverage factor-graph optimization, and provide researchers with a lightweight on-ramp by eliminating much of the implementation overhead.*

IX. Conclusions

In this work, we have demonstrated that certifiable estimators can be implemented and deployed within existing factor-graph modeling and optimization libraries (e.g., GTSAM) by leveraging a key theoretical insight: the correspondence between quadratically constrained quadratic programs (QCQPs) and factor-graph representations. This connection is instantiated through rank-p BM factorization, enabling seamless integration of the Riemannian staircase and certification mechanisms within standard factor-graph optimization back-ends. The resulting framework is simplified, modular, and practical: certifiable algorithms become as easy to run as standard local-search methods over factor graphs, making them suitable as plug-and-play state-estimation modules in real-world robotic systems.

Concretely, we introduce *lifted variables* and propose *lifted factors* for commonly used measurement types. These components can be assembled with minimal effort into three representative SLAM examples—pose-graph optimization, landmark-based SLAM, and range-aided SLAM—and benchmarked against state-of-the-art hand-crafted certifiable solvers, all within a practical, plug-and-play system compatible with off-the-shelf libraries (e.g., GTSAM). We validate the framework through extensive experiments on both synthetic and real-world SLAM datasets. Moreover, the proposed approach facilitates rapid prototyping of new certifiable estimation problems by allowing users to introduce new lifted factors for a wide variety of sensor measurements.

A key contribution of this work is the substantial reduction in implementation effort and the corresponding lowering of the barrier to entry: certifiable estimation becomes as accessible as standard factor-graph optimization. Whereas prior custom pipelines often require weeks to months of engineering per problem class, our plug-and-play factors enable users to assemble a certifiable estimation pipeline in less than a day. This reduction *democratizes access* to certifiably correct estimation for non-experts, simplifies deployment in robotic systems already using factor graphs, and provides researchers with a lightweight on-ramp for rapid experimentation. By enabling practitioners familiar with traditional SLAM back-ends to incorporate global optimality guarantees with minimal effort, our framework aims to stimulate further advances in certifiable state estimation. Future work will explore extending the proposed certifiable factors to a broader class of robotic estimation problems, including real-robot deployment and field robotics experiments.

References

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

[2] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[3] 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.

[4] 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.

[5] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[6] 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.

[7] F. Dellaert and G. Contributors, “borglab/gtsam,” May 2022. [Online]. Available: https://github.com/borglab/gtsam)

[8] 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.

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

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

[11] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[12] 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.

[13] 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.

[14] M. Todd, “Semidefinite optimization,” *Acta Numerica*, vol. 10, pp. 515–560, 2001.

[15] 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.

[16] 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.

[17] N. Boumal, *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

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

[19] S. J. Wright, *Primal-dual interior-point methods*. SIAM, 1997.

[20] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019. [Online]. Available: http://docs.mosek.com/9.0/toolbox/index.html

[21] 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.

[22] X. Yu and H. Yang, “Sim-sync: From certifiably optimal synchronization over the 3d similarity group to scene reconstruction with learned depth,” *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4471–4478, 2024.

[23] C. Holmes, F. Dümbgen, and T. D. Barfoot, “On semidefinite relaxations for matrix-weighted state-estimation problems in robotics,” *IEEE Transactions on Robotics*, 2024.

[24] F. Dümbgen, C. Holmes, B. Agro, and T. Barfoot, “Toward globally optimal state estimation using automatically tightened semidefinite relaxations,” *IEEE Transactions on Robotics*, vol. 40, pp. 4338–4358, 2024.

[25] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based slam,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2011.

[26] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, “Bundle adjustment in the large,” in *European Conference on Computer Vision ECCV*. Springer, 2010, pp. 29–42.

[27] 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.

[28] T. Fan, H. Wang, M. Rubenstein, and T. Murphey, "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.

[29] 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*, vol. 40, pp. 4265–4283, 2024.

[30] 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.

[31] 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," *IEEE Transactions on Robotics*, vol. 38, no. 4, 2022.

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

[33] F. Dellaert, D. M. Rosen, J. Wu, R. Mahony, and L. Carlone, "Shonan rotation averaging: Global optimality by surfing $SO(p)^n$," in *European Conference on Computer Vision (ECCV)*, Aug. 2020.

[34] P. Huber, *Robust Statistics*. Hoboken, N.J.: John Wiley & Sons, 2004.

[35] T. Ferguson, *A Course in Large Sample Theory*. Boca Raton, FL: Chapman & Hall/CRC, 1996.

[36] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[37] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Institute for Robotics & Intelligent Machines, Georgia Institute of Technology, Tech. Rep. GT-RIM-CP&R-2012-002, Sep. 2012.

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

[39] A. Papalia, Y. Tian, D. M. Rosen, J. P. How, and J. J. Leonard, "An overview of the Burer-Monteiro method for certifiable robot perception," *arXiv preprint arXiv:2410.00117*, 2024.

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

[41] ——, "Scalable low-rank semidefinite programming for certifiably correct machine perception," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 551–566.

[42] N. Boumal, "A riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints," *arXiv preprint arXiv:1506.00575*, 2015.

[43] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[44] R. Hartley, *Multiple view geometry in computer vision*. Cambridge university press, 2003, vol. 665.

[45] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin, "Rotation averaging and strong duality," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 127–135.

[46] O. Howell, H. Huang, and D. M. Rosen, "Multi-irreducible spectral synchronization for robust rotation averaging," *arXiv preprint arXiv:2311.16544*, 2023.

[47] K. Khosoussi, S. Huang, and G. Dissanayake, "A sparse separable SLAM back-end," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1536–1549, 2016.

[48] A. Fishberg and J. P. How, "Multi-agent relative pose estimation with uwb and constrained communications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 778–785.

[49] F. Dümbgen, C. Oeschger, M. Kolundžija, A. Scholefield, E. Girardin, J. Leuenberger, and S. Ayer, "Multi-modal probabilistic indoor localization on a smartphone," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2019, pp. 1–8.

[50] F. Dümbgen, C. Holmes, and T. D. Barfoot, "Safe and smooth: Certified continuous-time range-only localization," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1117–1124, 2022.

[51] F. Jiang, D. Caruso, A. Dhekne, Q. Qu, J. J. Engel, and J. Dong, "Robust indoor localization with ranging-imu fusion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 963–11 969.

[52] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-swarm: A decentralized omnidirectional visual–inertial–uwb state estimation system for aerial swarms," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3374–3394, 2022.

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

[54] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE transactions on robotics and automation*, vol. 17, no. 3, pp. 242–257, 2002.

[55] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, 2007.

[56] J. Djugash, B. Hamner, and S. Roth, "Navigating with ranging radios: Five data sets with ground truth," *Journal of Field Robotics*, vol. 26, no. 9, pp. 689–695, 2009.

Appendix A
Appendix

A. Expression of $\mathcal{A}^\star(\lambda)$ for Orthogonality Constraints

We present a *closed-form* expression for $\mathcal{A}^\star(\lambda)$ in (18), as required to form the certificate S in the BM problem. We focus on the case of orthogonality constraints arising from the Stiefel manifold (7), with the unit sphere (6) as a special instance. Such constraints naturally occur in many certifiable estimation problems and thus also their lifted-variable counterparts (see Section VII). Here we show that the adjoint map $\mathcal{A}^\star(\lambda)$ (under orthogonality constraints for problems with underlying factor graph structure) has a closed-form expression which enables direct construction of the certificate *without* explicitly solving for the multipliers.

While the Stiefel manifold in (7) is defined for matrices $Y \in \mathbb{R}^{r\times p}$, the block-variables in the rank-p BM factorization typically have the form $Y_n \in \mathbb{R}^{r_n \times p}$. Therefore, for notational convenience—so that each block-variable Y_n is a matrix submanifold of the Stiefel manifold—we adopt a *transposed formulation* of (12):

$$\begin{aligned} \min_{Y\in\mathbb{R}^{p\times r}} \quad & F(Y) = \langle Q, Y^\top Y\rangle, \\ \text{s.t.} \quad & \langle A_m, Y^\top Y\rangle = b_m, \quad \forall m \in [M], \end{aligned} \tag{60}$$

where the data matrices and constraints remain unchanged, but partition transposed variable $Y \in \mathbb{R}^{p\times r}$ into N *block-column* variables $Y_n \in \mathbb{R}^{p\times r_n}$:

$$Y = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_n \end{bmatrix}, \quad \text{such that} \quad r = \sum_{n=1}^{N} r_n. \tag{61}$$

1) Least-Squares Solution via Orthogonal Projection: Given $Y \in \mathbb{R}^{p\times r}$ and $Q \in \mathbb{S}^r$, consider the least-squares problem (20) transposed:

$$\min_{\lambda\in\mathbb{R}^M} \left\| YQ + Y\mathcal{A}^\star(\lambda)\right\|_F^2, \tag{62}$$

where the residual arises from the gradient of the Lagrangian of (60):

$$\nabla_Y \mathcal{L}(Y,\lambda) = 2\nabla_Y F(Y) + 2Y\mathcal{A}^\star(\lambda) = 2YS. \tag{63}$$

Although the Lagrangian and the least-squares multiplier system take a different form under the transposed variable formulation (compared with (20)), the certificate matrix S remains as defined in (18).

Rather than directly solving for the least squares λ, we first recover the product $Y\mathcal{A}^{\star}(\lambda)$, and reinterpret (62) as a matrix least-squares problem:

$$\min_{Y\mathcal{A}^{\star}(\lambda)\in\mathbb{R}^{p\times r}} \|YQ + Y\mathcal{A}^{\star}(\lambda)\|_F^2 . \tag{64}$$

We define the subspace:

$$\mathcal{S} := \left\{ Y\mathcal{A}^{\star}(\lambda) \;\middle|\; \lambda \in \mathbb{R}^M \right\} \subseteq \mathbb{R}^{p\times r}, \tag{65}$$

and let $\Pi_{\mathcal{S}} : \mathbb{R}^{p\times r} \to \mathcal{S}$ denote the orthogonal projector onto this subspace:

$$\Pi_{\mathcal{S}}(Z) := \arg\min_{W\in\mathcal{S}} \|Z - W\|_F^2. \tag{66}$$

Thus, the least-squares solution corresponds to projecting $-YQ$ onto $\mathcal{S}$, and satisfies

$$Y\mathcal{A}^{\star}(\hat{\lambda}) = -\Pi_{\mathcal{S}}(YQ). \tag{67}$$

In the objective (60), each block $(YQ)_n$ corresponds (up to scaling) to the Euclidean gradient of the objective. Therefore, the projection above recovers the projection of the Euclidean gradient onto the normal space of the Stiefel manifold product—used in computing the Riemannian gradient as [17], [36].

B. Orthogonal Projection for the Stiefel Manifold

Analogous with Theorem 3, given the block structure (61), the least-square system (64) decomposes as:

$$\min_{Y_n\mathcal{A}^{\star}(\lambda_{L_n})\in\mathbb{R}^{p\times r_n}} \|(YQ)_n + Y_n\mathcal{A}^{\star}(\lambda_{L_n})\|_F^2 , \tag{68}$$

where $(YQ)_n$ denotes the n-th column block partition of YQ. Each block $\mathcal{A}^{\star}(\lambda_{L_n})$ appears on the diagonal of the global matrix $\mathcal{A}^{\star}(\lambda)$, which has a block-diagonal structure aligned with the variable partition:

$$\mathcal{A}^{\star}(\lambda) = \begin{bmatrix} \mathcal{A}^{\star}(\lambda_{L_1}) & & & \\ & \mathcal{A}^{\star}(\lambda_{L_2}) & & \\ & & \ddots & \\ & & & \mathcal{A}^{\star}(\lambda_{L_N}) \end{bmatrix} \in \mathbb{S}^r, \tag{69}$$

where each $\mathcal{A}^{\star}(\lambda_{L_n}) \in \mathbb{S}^{r_n}$ corresponds to the column block-variable of Y defined in (61).

To derive the multipliers in closed form, we consider the constraint from $Y_n \in \mathrm{St}(p, r_n)$. The orthogonal projection of the Euclidean gradient $\nabla F(Y)_n$ onto the tangent space—i.e., the Riemannian gradient—is given by [42]:

$$\mathrm{Proj}_{Y_n}(\nabla F(Y)_n) = \nabla F(Y)_n - Y_n \mathrm{Sym}(Y_n^\top \nabla F(Y)_n), \tag{70}$$

where $\mathrm{Sym}(M) := \frac{1}{2}\left(M + M^\top\right)$ denotes the symmetrization of M. The corresponding normal space is [17, Section 7.2]:

$$N_{Y_n}(\mathrm{St}(p, r_n)) = \{Y_n A : A \in \mathbb{S}^{r_n}\}. \tag{71}$$

The orthogonal projector onto this space is the linear map:

$$\Pi_{Y_n} : \mathbb{R}^{p\times r_n} \to N_{Y_n}(\mathrm{St}(p, r_n)), \quad Z_n \mapsto Y_n\, \mathrm{Sym}(Y_n^\top Z_n). \tag{72}$$

Applying this projector to the gradient $(YQ)_n$, we obtain:

$$Y_n\mathcal{A}^{\star}(\lambda_n) = -Y_n\, \mathrm{Sym}(Y_n^\top (YQ)_n), \tag{73}$$

and therefore,

$$\mathcal{A}^{\star}(\lambda_n) = -\,\mathrm{Sym}(Y_n^\top (YQ)_n). \tag{74}$$

This result shows that each block $\mathcal{A}^{\star}(\lambda_n)$ can be computed independently and in parallel. Furthermore, rather than solving for λ explicitly, we directly recover $\mathcal{A}^{\star}(\lambda)$ in closed form to form the certificate S (18) directly.

The above derivation applies to blocks subject to orthogonality constraints. In the general case where Y_n is unconstrained (i.e., constraint matrices $A_m = 0, \forall m \in L_n$), the normal space is trivial and the projection becomes vacuous. In such cases, the corresponding certificate block is

$$\mathcal{A}^{\star}(\lambda_{L_n}) = 0. \tag{75}$$

Accordingly, the global adjoint map $\mathcal{A}^{\star}(\lambda)$ remains block-diagonal, with each diagonal block given by

$$\mathcal{A}^{\star}(\lambda_{L_n}) = \begin{cases} -\,\mathrm{Sym}(Y_n^\top (YQ)_n), & \text{if } Y_n \in \mathrm{St}(p, r_n), \\ 0, & \text{if } Y_n \text{ is unconstrained.} \end{cases} \tag{76}$$

This construction ensures that the certificate matrix respects the block-variable partition and includes only contributions from blocks with active constraints, thereby preserving both correctness and sparsity.