Nonlinear Factor Recovery for Long-Term SLAM

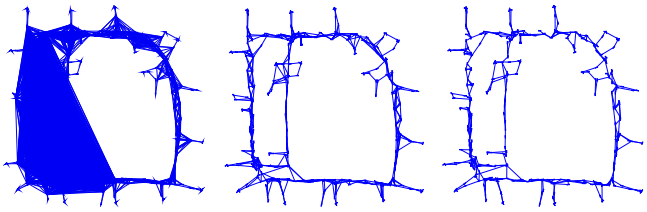
Mladen Mazuran Wolfram Burgard Gian Diego Tipaldi
Department of Computer Science, University of Freiburg, Germany
{mazuran, burgard, tipaldi}@informatik.uni-freiburg.de

***Abstract*—For long-term operations, graph-based SLAM approaches require to marginalize nodes in order to control the computational cost. In this paper, we present a method to recover a set of nonlinear factors that best represents the marginal distribution in terms of Kullback-Leibler divergence. The proposed method, which we call *nonlinear factor recovery (NFR)*, estimates both the mean and the information matrix of the set of nonlinear factors, where the recovery of the latter is equivalent to solving a convex optimization problem. NFR is able to provide either the dense distribution or a sparse approximation of it. In contrast to previous algorithms, our method does not necessarily require a global linearization point and can be used with any nonlinear measurement function. Moreover, we are not restricted to only use tree-based sparse approximations and binary factors, but we can include any topology and correlations between measurements. Experiments performed on several publicly available datasets demonstrate that our method outperforms the state of the art with respect to the Kullback-Leibler divergence and the sparsity of the solution.**

I. Introduction

Graph-based optimization techniques are an effective solution to the simultaneous localization and mapping (SLAM) problem. In graph-based optimization, the estimation problem is commonly associated with a factor graph, whose nodes represent the variables to be estimated and whose factors represent the measurements between the nodes. In most cases, the observations have nonlinear measurement functions and are affected by Gaussian noise. For such cases, it can be shown that performing inference on the factor graph representation is equivalent to nonlinear least squares minimization [Dellaert and Kaess, 2006]. By exploiting the sparse nature of the problem, researchers have developed effective optimization algorithms to solve even large-scale and challenging SLAM problems [Grisetti et al., 2010, Kaess et al., 2007, Kümmerle et al., 2011, Olson et al., 2006, Grisetti et al., 2007].

Unfortunately, when the number of variables is very large, the computational complexity of the estimation problem is high. In such cases, it is possible to reduce the problem size by eliminating a set of variables and minimizing the approximation loss in a statistical sense. To reduce the approximation error, the information related to the eliminated variables is preserved by marginalization. However, after successive marginalizations, the information matrix of the estimation problem becomes dense and sparsity enforcing methods need to be used [Kretzschmar et al., 2011, Kretzschmar and Stachniss, 2012, Carlevaris-Bianco and Eustice, 2013a, Vial et al., 2011, Huang et al., 2013]. Unfortunately, the marginalization process relies on the linear Gaussian assumption and can potentially introduce errors due to a suboptimal linearization point.



(a) Dense (b) Subgraph (c) Tree

Fig. 1. Conditional dependence graphs for the Intel dataset with 66.6% node reduction using the three marginalization methods proposed in this paper.

In this article we present a method to recover a set of nonlinear factors that best represents the marginal distribution in term of Kullback-Leibler divergence. The proposed method, *nonlinear factor recovery (NFR)*, aims at estimating the mean and the covariance of this set of factors. The current manuscript incorporates the initial findings reported in our previous publication [Mazuran et al., 2014], extends them in a more general framework, and provides an extended experimental and theoretical analysis. Specifically, we included:

- An extensive theoretical analysis of NRF;
- The capability of considering correlations between observations;
- The closed form solution for the correlated case;
- The proof that the work of Carlevaris-Bianco et al. [2014] is equivalent to a special case of NFR when using only relative measurements;
- The theoretical analysis of error propagation and NFR, when applied to odometry chains;
- The experimental analysis of NFR in 3D environments;
- The analysis of different linearization points.

Our method has several theoretical and practical advantages:

- The problem of determining factor covariances is convex;
- The approximation is performed either on a local or a global linearization point;
- The approach is general, i.e., any nonlinear measurement function can be used;
- The framework can consider any topology and any correlation between the measurements;
- The solution preserves the block structure of the matrix;
- A closed form solution exists in some particular cases.

We experimentally evaluate NFR using diverse sparsification and node reduction strategies and compare it with respect to the state of the art for both 2D and 3D SLAM settings. The results demonstrate that our method significantly outperforms the state of the art in terms of approximation accuracy and

sparseness of the solution, especially in online scenarios.

II. Related Work

Over the last decade, numerous efforts have been made towards minimizing the computational requirements of SLAM by reducing the amount of variables in the state space, while keeping the sparse structure of the problem. In the context of filtering, Thrun et al. [2004] introduced the sparse extended information filter (SEIF). The authors enforced sparsity whenever a node is marginalized by keeping only the edges with the largest entries (in terms of absolute value) in the information matrix. Eustice et al. [2005] provided a modification to SEIF minimizing the differences between the SEIF estimate and that of a non-sparsified filter. Vial et al. [2011] further extended SEIF by providing a method that ensures that the approximated information is strictly conservative. They also noted that the optimization need only be carried out on the Markov blanket of the node to marginalize. Our approach is similar in spirit to the one of Vial et al. [2011] but we are not restricted to the filtering setting and we are able to recover nonlinear measurement functions.

More recently, given the popularity of graph-based optimization solutions for SLAM, researchers investigated how to reduce the number of nodes in the SLAM graph, while keeping low approximation errors and the sparsity of the information matrix. Ila et al. [2010] proposed an information-theoretic approach to add only non-redundant nodes and highly informative edges to the graph. Despite this, the graph will eventually grow unbounded also with their approach, albeit at a slower rate. Johannsson et al. [2013] followed a similar idea. Instead of introducing spatially redundant nodes in the graph, they propagate their measurements through already existing nodes and add an additional measurement between them. In this way, the graph will only grow according to the size of the environment but not according to the operational time. We differ from them in two main aspects. First, we can deal with the elimination of nodes both spatially and temporally. Second, we can remove nodes at any point in time and not only at the time of insertion, thus keeping their information for as long as possible.

Another family of approaches focused on which node to remove from the graph via marginalization and how to treat the resulting Schur complement. Konolige and Bowman [2009] clustered nodes in the graph according to their spatial distance. Among each cluster, they removed the least recently used views, in order to keep a limited number of views and still capture the dynamic nature of the environment. The information of the removed nodes is kept via dense marginalization, causing an increased fill-in in the information matrix. A similar idea has also been introduced by Eade et al. [2010]. The authors proposed to remove nodes without image data or with views similar to existing nodes in the vicinity. To reduce complexity, they also proposed to prune some edges according to a heuristic based on node degree.

Some authors replaced the Markov blanket of the marginalized node with either a linearized measurement or a set of linearized measurements. Folkesson and Christensen [2004] considered a particular case of linearized measurement, called *star nodes*. In there, the position of each node is a linear function of a root node, which, in turn, is connected with the rest of the graph. Similarly, Frese [2007] proposed to use linearized measurements to remove nodes in cliques when performing relaxation, and applied this technique within the Treemap algorithm [Frese, 2006]. In order to keep the linearized measurements sparse, the author also removed low informative edges in the clique graph, in the same spirit of the thin junction tree algorithm [Paskin, 2003].

Kretzschmar et al. [2011] proposed a information-based criterion for determining which nodes to marginalize. They further employed the Chow-Liu tree approximation [Chow and Liu, 1968] to sparsify the Markov blanket of the marginalized nodes and keep the complexity low. Carlevaris-Bianco et al. extended the previous work by introducing *Generic linear constraint* (GLC) factors [Carlevaris-Bianco et al., 2014, Carlevaris-Bianco and Eustice, 2013a,b]. GLCs are n-ary edges of a factor graph, either dense or based on the Chow-Liu tree, which approximate the information matrix of the Markov blanket. With respect to those approaches, we explicitly consider nonlinear measurement functions and provide a sound mathematical framework based on convex minimization. The same authors further extended their approach to enforce a conservative approximation of the true marginalized potential [Carlevaris-Bianco and Eustice, 2014]. The authors proposed three different algorithms with increasing complexity and approximation accuracy, based on the covariance intersection framework. Our work is orthogonal to that and both can be jointly employed to obtain a sparse, conservative and nonlinear approximation of the true marginalized distribution.

Huang et al. [2013] approximated the dense information matrix solving an ℓ_1-regularized minimization problem. They used the alternating direction of multipliers method (ADMM) [Boyd et al., 2011] to solve the problem and determine a conservative and sparse approximation. The approach, however, requires the information matrix of the full graph. On the contrary, our approach is local, in the sense that we directly operate on the Markov blanket of the marginalized node. Moreover, we explicitly consider nonlinear measurements and the block structure of the state space, while they commit on a linearization point and do not preserve the block structure.

Sparsification problems similar to the one presented in this work have been considered in the machine learning community, under the name of *Sparse Inverse Covariance Selection* (SICS). Banerjee et al. [2006] first introduced the problem of estimating the sparsity pattern of an information matrix from a dense covariance by regularizing it with an ℓ_1 penalizer. They showed that the dual problem has a simpler solution and employed a block coordinate descent algorithm. The work has been extended by Friedman et al. [2008] with the introduction of the *Graphical Lasso*. They modified the Lasso algorithm to work directly on the primal problem and showed an improved convergence speed.

Duchi et al. [2008] extended the approach to deal with block

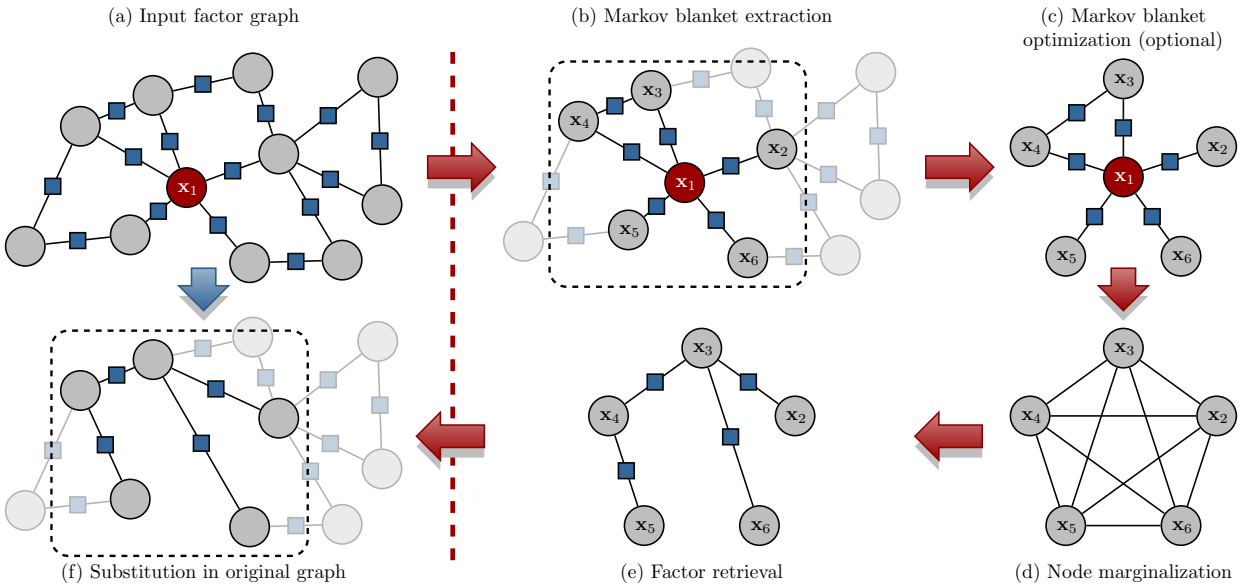


Fig. 2. The main steps of the node removal algorithm.

sparsity by introducing an $\ell_{1,\infty}$ regularization term. They solve the resulting minimization problem using a projected gradient descent algorithm. Schmidt et al. [2009] introduced the projected quasi newton algorithm (PQN) and showed its application to block inverse covariance selection problems.

Our method shares some grounds with the block inverse covariance selection problem with the difference being that we aim to obtain a set of nonlinear measurements that approximate the target information, instead of a specific information matrix.

III. Nonlinear Factor Recovery

In this section we describe our general framework for nonlinear factor recovery (NFR), which we first introduced in our previous work [Mazuran et al., 2014]. The proposed algorithm consists of different steps, depicted in Fig. 2. Without loss of generality, we assume that there exists a node selection method that specifies which node will be eliminated. This includes strategies such as the work of Kretzschmar et al. [2011], Kretzschmar and Stachniss [2012], or strategies based on the Euclidean distance [Johannsson et al., 2013]. Fig. 2(a) depicts both an example factor graph as input, together with the node $\mathbf{x}_1$ to be removed (in red).

In the first step of our method (Fig. 2(b)), we extract the Markov blanket of the node $\mathbf{x}_1$ from the original graph. In typical SLAM applications, the Markov blanket is in general sparse and composed of few nodes. Then, we decide on a linearization strategy to use. In this work we consider both global and local linearization points, and evaluate both approaches for the proposed graph topologies. Section III-A describes in more details the differences between the two linearization strategies.

We proceed to compute the linearized potential induced by the factors in the Markov blanket. In the case of local linearization, we first compute the configuration of the nodes in the Markov blanket via maximum likelihood optimization, by considering only measurements within the blanket. This provides us with an optimal local linearization point (Fig. 2(c)). In the case of global linearization, we keep the estimates of the nodes on the Markov blanket the same as the current best estimate of the full graph, as done by Carlevaris-Bianco et al. [2014].

Given the linearization point, we then compute the Schur complement of the node to be removed, obtaining the marginalized d-dimensional multivariate normal distribution $p(\mathbf{x})$. This distribution has mean $\boldsymbol{\mu}$ and strictly positive definite information matrix $\boldsymbol{\Omega}$, with inverse $\boldsymbol{\Sigma}$. We will explore the problem of a singular $\boldsymbol{\Omega}$ in Section IV-C. At this point, we can choose to either recover the exact potential using a dense nonlinear factor or to approximate the distribution by a sparse set of (potentially correlated) nonlinear factors. Section V introduces the topologies considered in this work and describes them in more details.

Suppose now that we are also given a set of m independent nonlinear measurements $\mathbf{z}_i$ with measurement functions $\mathbf{f}_i(\mathbf{x})$. These functions, for instance, can be derived from a sensor model or can be defined by an expert user. Using the chosen linearization point, NFR recovers the mean $\boldsymbol{\zeta}_i$ and information matrix $\mathbf{X}_i$ of the set of nonlinear factors between the nodes defined by the topology (Fig. 2(e)). This set of nonlinear factors induces a distribution $q(\mathbf{x})$, whose first two standardized moments, the mean $\boldsymbol{\nu}$ and the information matrix $\boldsymbol{\Upsilon}$, can be estimated via maximum likelihood inference. We formulate the nonlinear recovery such that the resulting linearized distribution $q(\mathbf{x})$ minimizes the Kullback-Leibler divergence (KLD) with respect to $p(\mathbf{x})$, which for multivariate

normal distributions is equivalent to:

$$D_{KL}(p(\mathbf{x})\|q(\mathbf{x})) = \int_{\mathbb{R}^d} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \, \mathrm{d}\mathbf{x} = \tag{1}$$

$$= \frac{1}{2} \left(\langle \boldsymbol{\Upsilon}, \boldsymbol{\Sigma} \rangle - \log\det(\boldsymbol{\Upsilon}\boldsymbol{\Sigma}) + \left\| \boldsymbol{\Upsilon}^{\frac{1}{2}} (\boldsymbol{\nu} - \boldsymbol{\mu}) \right\|_2^2 - d \right). \tag{2}$$

Here, $\langle \cdot, \cdot \rangle$ denotes the matrix inner product, while $\boldsymbol{\Upsilon}^{\frac{1}{2}}$ denotes any square root matrix of $\boldsymbol{\Upsilon}$.

The computation of mean and information matrix can be carried out independently. In fact, $\boldsymbol{\zeta}_i$ acts only on the squared Mahalanobis distance $\|\boldsymbol{\Upsilon}^{\frac{1}{2}}(\boldsymbol{\nu} - \boldsymbol{\mu})\|_2^2$, which can be brought down to zero by noting that $\boldsymbol{\nu} = \boldsymbol{\mu}$ when $\boldsymbol{\zeta}_i = \mathbf{f}_i(\boldsymbol{\mu})$ for all i. On the other hand, recovering the optimal information matrix is not trivial and we describe it in Section IV.

Finally, we replace the original factors in the Markov blanket with the newly recovered ones and substitute them in the original graph (Fig. 2(f)).

A. Global versus local linearization

The approximation of the Markov blanket of a node can be carried out on different linearization points. As a matter of fact, the choice of linearization point will impact the value of the measurement functions and the Jacobians that are introduced in Section IV. This, in turn, not only affects which mean and covariance will be chosen for the virtual measurements, but is also crucial to achieve low KLD values for the particular sparsification scenario at hand.

As mentioned before, we distinguish two major choices:

- *Global linearization*: we keep the estimates of the nodes on the Markov blanket the same as the current best estimate of the full graph. This is the approach taken by Carlevaris-Bianco et al. [2014] with GLC.
- *Local linearization*: we optimize the nodes on the Markov blanket as if they were not connected to the original graph. The estimates of the nodes are thus given only by considering the factors in the Markov blanket.

Both approaches have drawbacks and benefits. The global linearization point is most effective when the Markov blanket is strongly constrained by the factors that connect it to the remainder of the graph. In such a scenario, the nonlinearities of the factors *in* the Markov blanket have a negligible effect on the linearization point of the full graph. A possible example is the removal of nodes from a region of a factor graph with multiple loop-clusures, such as a batch sparsification scenario.

Using a local linearization point, on the other hand, guarantees the node removal to be independent of the linearization point of the full factor graph, and can be thus used without the need of optimizing the whole graph beforehand. Furthermore, as shown in Section VII, using a local linearization point produces superior results when the non-linearities in the Markov blanket are non negligible, such as in an incremental mapping scenario. This, however, comes at the expense of degraded accuracy in batch sparsification scenarios.

IV. Information Matrix Computation

The computation of the information matrices of the measurements cannot always be done by maintaining the exact relationship $\boldsymbol{\Upsilon}\boldsymbol{\Sigma} = \mathbf{I}$. Let $\mathbf{f}(\mathbf{x})$ and $\mathbf{z}$ be the vectors obtained by stacking all $\mathbf{f}_i(\mathbf{x})$ and all of the measurement random vectors, respectively. We define the following matrices:

$$\mathbf{A} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\boldsymbol{\mu}} \qquad \mathbf{X} = \mathrm{cov}(\mathbf{z})^{-1} = \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{X}_m. \end{bmatrix} \tag{3}$$

Here, $\mathbf{A}$ is a constant matrix that depends on the linearization point, while $\mathbf{X}$ is the combined information matrix that we wish to compute for the relative measurements. Thus, if we denote by n_i the number of rows or columns of $\mathbf{X}_i$ and taking into account the symmetry of information matrices, $\mathbf{X}$ is an optimization variable with dimension

$$\sum_{i=1}^{m} \frac{n_i(n_i+1)}{2}. \tag{4}$$

Notice that $\mathbf{X}$ is block diagonal due to the assumption of independence between nonlinear measurements. This comes without loss of generality, since correlated measurements can be expressed as a larger diagonal block in $\mathbf{X}$. In the limiting case, $\mathbf{X}$ can be fully dense. For clarity we will henceforth refer to $\mathcal{X}$ as the set of block diagonal matrices consistent with the measurement functions and their correlations, hence $\mathbf{X} \in \mathcal{X}$.

Then, the information matrix $\boldsymbol{\Upsilon}$ is given by:

$$\boldsymbol{\Upsilon} = \mathbf{A}^\top \mathbf{X} \mathbf{A}. \tag{5}$$

Note that we will assume $\mathbf{A}$ to be of full column rank, as otherwise the $\boldsymbol{\Upsilon}$ is singular and the KLD not well defined. Under this assumption we can then express the problem of computing the minimum of (2) as the following constrained optimization problem:

$$\begin{aligned} \text{minimize} \quad & \left\langle \mathbf{A}^\top \mathbf{X} \mathbf{A}, \boldsymbol{\Sigma} \right\rangle - \log\det\left(\mathbf{A}^\top \mathbf{X} \mathbf{A}\right) && (6) \\ \text{subject to} \quad & \mathbf{X} \in \mathcal{X} && (7) \\ & \mathbf{X} \succeq \mathbf{0} && (8) \end{aligned}$$

Note that (6)-(8) is indeed a *convex* optimization problem. In fact, it is known that $-\log\det(\cdot)$ is convex in its argument [Boyd and Vandenberghe, 2009], which, together with the fact that convex functions are closed with respect to positively weighted addition and composition with affine functions, gives the convexity of (6). More specifically, (6)-(8) is an instance of the MAXDET problem [Vandenberghe et al., 1998].

In general, a closed form solution to (6)-(8) does not exist, however, its convexity at least guarantees that we can always compute its global optimum.

A. Closed form solution

For special instances of the matrices $\mathbf{A}$ and $\mathbf{X}$ it is in fact possible to compute a closed form solution to problem (6)-(8). In the following we consider two particular instances where this is the case. The first is of practical interest as we

will show in Sections V-A and V-C; the second, on the other hand, is of theoretical importance for what will be presented in Section VI-C.

Here we denote with $\{\cdot\}_i$ the i-th diagonal block of the enclosed matrix, and, for aesthetic reasons, we denote by $\mathbf{A}^{\mp}$ the pseudoinverse of the transpose of $\mathbf{A}$. In an effort to not curb the readability of this article, any proposition reported in the main body will be stated without proof. We refer the reader to the Appendix for the mathematical proofs.

Proposition IV.1. *When* $\mathbf{A}$ *is invertible, the unique solution to problem* (6)-(8) *is given by:*

$$\mathbf{X}_i = \left(\left\{\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top\right\}_i\right)^{-1}. \tag{9}$$

Proposition IV.2. *When* $\mathbf{A}$ *is of full column rank and* $\mathcal{X}$ *is the set of fully dense matrices, one of the solutions to problem* (6)-(8) *is given by:*

$$\mathbf{X} = \mathbf{A}^{\mp}\boldsymbol{\Omega}\mathbf{A}^{+}. \tag{10}$$

Furthermore, (10) *yields equality between* $\mathbf{A}^\top\mathbf{X}\mathbf{A}$ *and* $\boldsymbol{\Omega}$.

B. Iterative solution

For the instances not covered by Propositions IV.1 and IV.2, the solution to problem (6)-(8) needs to be computed numerically. In order to do so one can either rely on the *Limited-memory Projected Quasi-Newton* algorithm (PQN) [Schmidt et al., 2009] as was the case for our previous work [Mazuran et al., 2014], or, rather, adopt a more traditional interior point approach.

Both approaches carry advantages and drawbacks:

- An interior point method requires the explicit computation (and inversion) of a Hessian matrix which, for large Markov blanket sizes, may be exceedingly large. At the same time, however, it requires only a small number of iterations to converge, and in fact has quadratic convergence due to the use of the Hessian.
- PQN, being a modification of the L-BFGS algorithm [Nocedal, 1980], has much more modest memory requirements, but is only super-linear in convergence, and thus may require many more iterations than interior point, particularly if we are interested in computing the solution up to high accuracy.

When optimizing via interior point we require a log barrier on the constraints and the ability to compute both gradient and Hessian of the resulting cost function. Note that we assume constraint (7) to be enforced implicitly by optimizing only with respect to the relevant variables, a choice that is also shared by the PQN approach.

Given a strictly feasible initial guess, say $\mathbf{X} = \mathbf{I}$, we thus transform the constrained problem (6)-(8), into a sequence of unconstrained problems where the log barrier parameter, say ρ, is iteratively decreased towards 0. Each unconstrained problem aims to minimize the cost $u(\mathbf{X})$, given by:

$$\left\langle \mathbf{A}^\top\mathbf{X}\mathbf{A}, \boldsymbol{\Sigma}\right\rangle - \log\det\left(\mathbf{A}^\top\mathbf{X}\mathbf{A}\right) - \rho\log\det\mathbf{X}. \tag{11}$$

We can therefore find the minimum of (11) by using Newton's method. If we denote by x_{jk} the entries of $\mathbf{X}$, by $\mathbf{J}^{jk}$ the single entry matrix with zeros everywhere except a one at (j,k), and also let

$$\boldsymbol{\Phi} = \mathbf{A}\left(\mathbf{A}^\top\mathbf{X}\mathbf{A}\right)^{-1}\mathbf{A}^\top, \tag{12}$$

then the gradient and the entries of the Hessian are given by:

$$\frac{\partial u}{\partial \mathbf{X}_i} = \left\{\mathbf{A}\left[\boldsymbol{\Sigma} - \left(\mathbf{A}^\top\mathbf{X}\mathbf{A}\right)^{-1}\right]\mathbf{A}^\top - \rho\mathbf{X}^{-1}\right\}_i, \tag{13}$$

$$\frac{\partial^2 u}{\partial x_{jk}\partial \mathbf{X}_i} = \left\{\boldsymbol{\Phi}\,\mathbf{J}^{jk}\,\boldsymbol{\Phi} + \rho\mathbf{X}^{-1}\mathbf{J}^{jk}\mathbf{X}^{-1}\right\}_i. \tag{14}$$

As for PQN, the method requires the ability to compute the gradient of the cost function, namely

$$\frac{\partial D_{KL}}{\partial \mathbf{X}_i} = \mathbf{A}\left[\boldsymbol{\Sigma} - \left(\mathbf{A}^\top\mathbf{X}\mathbf{A}\right)^{-1}\right]\mathbf{A}^\top, \tag{15}$$

and an Euclidean projection $\mathcal{P}(\mathbf{X})$ onto the constraint set, which in this case is the set of positive semidefinite matrices. In particular, if we denote by $\mathbf{V}\,\mathrm{diag}(\lambda_i)\mathbf{V}^\top$ the eigen decomposition of an arbitrary symmetric matrix $\mathbf{X}$, $\mathcal{P}(\mathbf{X})$ is known to be expressible in closed form [Higham, 1988] as:

$$\mathcal{P}(\mathbf{X}) = \underset{\mathbf{Y}\succeq\mathbf{0}}{\arg\min}\|\mathbf{X}-\mathbf{Y}\|_F^2 = \mathbf{V}\,\mathrm{diag}(\max\{0,\lambda_i\})\mathbf{V}^\top, \tag{16}$$

where $\|\cdot\|_F$ represents the Frobenius norm. Furthermore, since in our case $\mathbf{X}$ is block diagonal, this process can be carried out independently for each block, resulting in a very efficient linear-time projection.

The efficiency of PQN strongly depends on the initial guess and the speed of computation of the gradient (15). For the former we can use the educated initial guess reported in our previous work [Mazuran et al., 2014]. Note that this initialization cannot be used for interior point, as it often yields a point on the feasible set boundary. As for the gradient, since we are only interested in computing the blocks on the main diagonal of a, possibly large, matrix, the computation can be optimized by noting that for any $\mathbf{Y}$:

$$\left\{\mathbf{A}\,\mathbf{Y}\,\mathbf{A}^T\right\}_i = \sum_j\sum_k \mathbf{A}_{ij}\mathbf{Y}_{jk}\mathbf{A}_{ik}^T, \tag{17}$$

which can be computed very efficiently if $\mathbf{A}$ is sparse. For the gradient we would thus set $\mathbf{Y} = \boldsymbol{\Sigma} - \left(\mathbf{A}^\top\mathbf{X}\mathbf{A}\right)^{-1}$, but we can also use this same approach to recover the closed form solution (9), in which case $\mathbf{Y} = \boldsymbol{\Sigma}$.

C. Handling Rank Deficient Information Matrices

In Section IV, we assumed the information matrix $\boldsymbol{\Omega}$ to be invertible. Unfortunately, when dealing with node removal, this is not always the case. From a SLAM perspective, for example, if we are dealing only with relative $\mathrm{SE}(n)$ measurements, $\boldsymbol{\Omega}$ will be rank deficient, with $\mathrm{nullity}(\boldsymbol{\Omega}) = \gamma$, where γ is the dimension of an $\mathrm{SE}(n)$ pose.

If $\boldsymbol{\Omega}$ has n rows, then the distribution of $p(\mathbf{x})$ is actually an $(n-\gamma)$-dimensional multivariate normal embedded in an n-dimensional space. Therefore, we propose to project $p(\mathbf{x})$ and

$q(\mathbf{x})$ onto the $(n-\gamma)$-dimensional informative subspace and to compare the resulting $(n-\gamma)$-dimensional distributions. In order to do so, we require an $(n-\gamma)\times n$ projection matrix $\mathbf{\Pi}$, acting as an operator that projects any arbitrary information matrix $\mathbf{\Psi}$ onto the lower dimensional space, by computing $\mathbf{\Pi}\mathbf{\Psi}\mathbf{\Pi}^\top$.

While the use of a projection does guarantee that the KLD is well defined, if the measurement functions acting on $q(\mathbf{x})$ are not chosen wisely, $q(\mathbf{x})$ may very well have an information matrix with greater rank than the one we are approximating. An intuitive example is that of approximating a Markov blanket composed of only relative $\mathrm{SE}(n)$ measurements with a set of absolute measurements (e.g. GPS) on all poses. In such cases the rank should be limited structurally, by providing appropriate measurement functions.

Since $\mathbf{\Omega}$ is a symmetric real-valued matrix, its eigen decomposition is real-valued and always exists. Further, let us denote with the expression *rank revealing eigen decomposition* the factorization $\mathbf{\Omega} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, such that $\mathbf{\Lambda}$ is a strictly positive definite diagonal matrix, and $\mathbf{U}$ is a, possibly rectangular, full column rank matrix with orthonormal columns. Note that such a decomposition can be trivially computed by setting $\mathbf{\Lambda}$ to be the diagonal matrix of the non-null eigenvalues of $\mathbf{\Omega}$, and by setting $\mathbf{U}$ to be the matrix of eigenvectors associated to the non-null eigenvalues.

To account for singular information matrices, we can thus use as projection matrix $\mathbf{\Pi} = \mathbf{U}^\top$. We will thus convert the specification (6)-(8) into a well defined problem by applying the following substitution rules:

$$\mathbf{A} \mapsto \mathbf{A}\mathbf{U}, \tag{18}$$

$$\mathbf{\Sigma} \mapsto \mathbf{\Lambda}. \tag{19}$$

Note that this substitution should be applied for any instance of $\mathbf{A}$ and $\mathbf{\Sigma}$, *except* for Proposition IV.2. In fact, Proposition IV.2 holds regardless of the projection operation. Consequently, $\mathbf{A}$ and $\mathbf{\Omega}$ should respectively be kept as the actual Jacobian and the original, singular, information matrix.

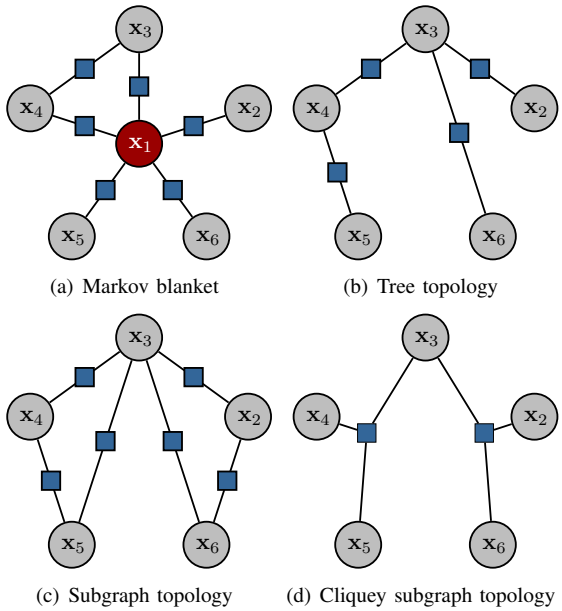
Even with this substitution, we can preserve efficiency when computing gradient (15) for PQN, by substituting into (17):

$$\mathbf{Y} = \mathbf{U}\left[\mathbf{\Sigma} - \left(\mathbf{U}^\top\mathbf{A}^\top\mathbf{X}\,\mathbf{A}\,\mathbf{U}\right)^{-1}\right]\mathbf{U}^\top. \tag{20}$$

V. Markov blanket topologies

In order to approximate the Markov blanket of a node, we first require a particular topology to select which nodes are affected by the nonlinear measurement functions. When dealing with relative $\mathrm{SE}(n)$ rigid body transformations, this is equivalent to determining the number of measurement functions and which nodes to use as their input. This, in turn, defines the Jacobian matrix $\mathbf{A}$ and the block structure of the information matrix $\mathbf{X}$.

In this section we propose multiple approaches towards computing a topology for the measurements, with varying levels of sparsity and, as a consequence, also accuracy and computational complexity. Fig. 3 gives an overview of the topologies we consider.



(a) Markov blanket (b) Tree topology

(c) Subgraph topology (d) Cliquey subgraph topology

Fig. 3. Different topologies for the Markov blanket approximation. In (a) we wish to remove node $\mathbf{x}_1$; we can either approximate the exact marginalization with (b) a tree, (c) a subgraph, or (d) a cliquey subgraph. Note that (c) and (d) introduce the same fill-in, but only (d) has a closed form solution.

A. Tree topology

The sparsest topology we propose is in the form of a *tree* of virtual measurements connecting the nodes in the Markov blanket. For this we rely on the Chow-Liu tree [Chow and Liu, 1968] approximation of an arbitrary distribution, which was first adopted for factor graph sparsification by Kretzschmar et al. [2011].

The goal of the Chow-Liu tree is that of computing the best approximation possible (in terms of KLD) of an n-variate arbitrary distribution $p(\mathbf{x}_1,\ldots,\mathbf{x}_n)$ by relying only on second order conditional distributions. In other words, this entails computing an index r and set of index pairs $\mathcal{I} \subseteq \{1,\ldots,n\}\times\{1,\ldots,n\}$ such that:

$$p(\mathbf{x}_1,\ldots,\mathbf{x}_n) \approx p(\mathbf{x}_r)\prod_{(i,j)\in\mathcal{I}} p(\mathbf{x}_i \mid \mathbf{x}_j). \tag{21}$$

Chow and Liu [1968] proved that the optimal set $\mathcal{I}$ is given by the edges of any directed maximum spanning tree (MST) of an undirected graph weighted on the mutual information between the nodes it connects. The index r, on the other hand, indicates the root of the directed MST. The Chow-Liu tree is therefore of practical importance since the MST can be computed efficiently by means of either Kruskal's or Prim's algorithm.

Unfortunately, as underlined by Carlevaris-Bianco et al. [2014], when dealing with under-constrained problems such as those with only $\mathrm{SE}(n)$ relative measurements, the mutual information

$$I(\mathbf{x}_i,\mathbf{x}_j) = -\frac{1}{2}\log\frac{\det\left(\mathbf{\Omega}_{ii} - \mathbf{\Omega}_{ij}\mathbf{\Omega}_{jj}^{-1}\mathbf{\Omega}_{ji}\right)}{\det\mathbf{\Omega}_{ii}} \tag{22}$$

is undefined. This follows from noting that $I(\mathbf{x}_i, \mathbf{x}_j)$ requires $\mathbf{\Omega}_{ii}$, $\mathbf{\Omega}_{jj}$, and $\mathbf{\Omega}_{ij}$ to be obtained by first marginalizing all $\mathbf{x}_k$ with $k \notin \{i, j\}$. Since the nullity of the Schur complement of a matrix cannot be smaller then the nullity of the matrix itself, it follows that $\mathbf{\Omega}_{ii} - \mathbf{\Omega}_{ij}\mathbf{\Omega}_{jj}^{-1}\mathbf{\Omega}_{ji}$ is singular. Furthermore, when dealing with only relative $\mathrm{SE}(n)$ measurements, $\mathbf{\Omega}$ has as nullity the dimension of a pose, therefore in such instances $\mathbf{\Omega}_{ii} - \mathbf{\Omega}_{ij}\mathbf{\Omega}_{jj}^{-1}\mathbf{\Omega}_{ji}$ is actually the null matrix.

To circumvent this problem, Carlevaris-Bianco and Eustice [2013a] adopt the Tikhonov regularization $\mathbf{X} + \varepsilon\,\mathbf{I}$, with $\varepsilon = 1$, whenever they require to compute the determinant of a singular matrix $\mathbf{X}$.

The use of (22), however, has as drawback the fact that it requires a quadratic number of Schur complements, one for each pair of nodes. Since each matrix inversion has cubic complexity in the number of rows, this approach effectively results in an algorithm with quintic complexity in the number of nodes of the Markov blanket.

To reduce this complexity, we propose to compute the mutual information directly from $\hat{\mathbf{\Sigma}} = (\mathbf{\Omega} + \varepsilon\,\mathbf{I})^{-1}$, the covariance associated to the Tikhonov regularization of $\mathbf{\Omega}$:

$$I(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2} \log \frac{\det \hat{\mathbf{\Sigma}}_{ii} \det \hat{\mathbf{\Sigma}}_{jj}}{\det \begin{bmatrix} \hat{\mathbf{\Sigma}}_{ii} & \hat{\mathbf{\Sigma}}_{ij} \\ \hat{\mathbf{\Sigma}}_{ji} & \hat{\mathbf{\Sigma}}_{jj} \end{bmatrix}}. \tag{23}$$

When the information matrix is invertible, (23) is exactly equivalent to (22). This can be trivially proven by substituting the entropy formula for a normal random variable into the mutual information identity:

$$I(\mathbf{x}, \mathbf{y}) = H(\mathbf{x}) + H(\mathbf{y}) - H(\mathbf{x}, \mathbf{y}). \tag{24}$$

Since the Tikhonov regularization is carried out on the full information matrix, rather than the argument of the determinant alone, this approach is not strictly equivalent to the one of Carlevaris-Bianco and Eustice [2013a], nevertheless, the differences are negligible and it is arguable which one would provide better results. Contrary to the multiple Schur complements required by (22), this approach requires only one inversion, where each computation of (23) has constant complexity in the number of nodes of the Markov blanket. Thus, this results in an overall cubic complexity. For an efficient computation of the inverse, we can then employ the Cholesky decomposition, since $\hat{\mathbf{\Sigma}} \succ \mathbf{0}$.

By using the Chow-Liu tree as a base, we can thus define a tree topology of factors, for example relative $\mathrm{SE}(n)$ measurements, with $\mathcal{I}$ as connectivity specification. A sample representation of such a topology is given in Fig. 3(b). Most importantly, however, if both the Markov blanket and the tree topology consist of relative $\mathrm{SE}(n)$ measurements, it is possible to solve (6)-(8) in closed form. This is a direct consequence of Lemmas A.6 and A.7, which ensure that under projection $\mathbf{U}^\top$ the product $\mathbf{AU}$ is invertible.

B. Subgraph topology

Due to the generality of NFR, we are not restricted to a mere tree topology when determining virtual measurements. Let n be the number of nodes in the Markov blanket; we can generalize the Chow-Liu tree by computing the maximum spanning subgraph with at most $\gamma\,(n-1)$ edges, where $\gamma \geq 1$ is a proportionality factor. If we choose $\gamma = 1$ then the method is equivalent to the Chow-Liu tree.

We employ a greedy heuristic and leverage on the *"information never hurts"* principle to compute such a subgraph. The algorithm starts by taking the Chow-Liu tree as a base and continues by augmenting it with the $\lfloor(\gamma - 1)\,(n-1)\rfloor$ edges with the highest mutual information that are not part of the spanning tree.

We can then use the resulting subgraph as connectivity specification for a *subgraph* topology such as the one depicted in Fig. 3(c). Note that choosing a bound on the number of edges as a proportion of the number of nodes allows to improve the accuracy in approximating the target information matrix $\mathbf{\Omega}$, while avoiding the quadratic fill-in of a dense marginalization. We refer to the special case of unbounded γ as a *dense* topology.

Contrary to the tree topology, the solution of (6)-(8) cannot be computed in closed form for the subgraph and dense topologies. This is because the Jacobian $\mathbf{A}$ remains rectangular even when projected and hence is not invertible. The information matrices of the factors thus need to be computed either with the PQN-based iterative solver or with an interior point procedure.

C. Cliquey subgraph topology

If we inspect the closed form solution in Proposition IV.1 we see that there is no strict need to consider the factors that define the Jacobian matrix $\mathbf{A}$ to be uncorrelated. In fact, by introducing an abstract "joint" factor, we may consider any number of virtual measurements to be correlated. The joint factor will then have as measurement function the stacked measurements of the original factors, and as information matrix the joint information matrix over all the involved measurements.

With this in mind, by using correlated measurements we can provide a better, albeit more dense, approximation of a target information $\mathbf{\Omega}$ than the one obtained by a tree. Contrary to the subgraph topology, such an approach will still keep a closed form solution, thus maintaining computational efficiency as an interior point procedure is not required.

When computing such a topology we introduce correlations between the factors yielded by the tree topology, and as with the subgraph approach, limit the final fill-in by a proportionality factor γ of $n - 1$, where n is the number of nodes in the Markov blanket. It is important to realize that correlating two measurements will also fully correlate the clique of nodes on which the measurements act, we thus refer to this topology as *cliquey subgraph*. Further, we refer to the special case of unbounded γ as a *cliquey dense* topology. We report in Fig. 3(d) a toy example, where the two ternary factors were obtained by taking the tree in Fig. 3(b) as an input and correlating two pairs of factors.

We propose in Algorithm 1 a greedy approach towards computing a set of cliques for the cliquey subgraph. In the algorithm, we further constrain the cliques to be over a tree

Algorithm 1 Clique computation for the cliquey subgraph.

Require: Vertices $\mathcal{V}$ and edges $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ of the tree topology. A proportionality factor γ for the maximum fill-in.

function CLIQUEYSUBGRAPH($\mathcal{V}, \mathcal{E}, \gamma$)
  // $\mathcal{C} \subseteq \mathcal{P}(\mathcal{V})$ specifies the set of cliques to return
  $\mathcal{C} \leftarrow \mathcal{E}$
  // Fill-in induced by $\mathcal{C}$ on the strictly upper triangular
  // part of the information matrix
  $\phi \leftarrow |\mathcal{V}| - 1$
  // Try to join pairs of cliques $\mathcal{Q}_1$ and $\mathcal{Q}_2$ by starting
  // from joins of smaller size to progressively bigger sizes
  for $s \in \{3, \dots, |\mathcal{V}|\}$ **do**
    for $\mathcal{Q}_1 \in \mathcal{C}$ **do**
      for $\mathcal{Q}_2 \in \mathcal{C} \setminus \{\mathcal{Q}_1\}$ **do**
        // Check if the cliques share vertices and their union
        // is not too large
        if $\mathcal{Q}_1 \cap \mathcal{Q}_2 \neq \varnothing \wedge |\mathcal{Q}_1 \cup \mathcal{Q}_2| \leq s$ **then**
          // Extra fill-in introduced by joining the cliques
          $\delta \leftarrow (|\mathcal{Q}_1| - 1)(|\mathcal{Q}_2| - 1)$
          // If the fill-in is acceptable join the cliques
          if $\phi + \delta \leq \gamma(|\mathcal{V}| - 1)$ **then**
            $\mathcal{C} \leftarrow (\mathcal{C} \setminus \{\mathcal{Q}_1, \mathcal{Q}_2\}) \cup \{\mathcal{Q}_1 \cup \mathcal{Q}_2\}$
            $\phi \leftarrow \phi + \delta$
          end if
        end if
      end for
    end for
  end for
  return $\mathcal{C}$
end function

of connected measurements, in order to limit the fill-in that a single correlation introduces.

VI. Theoretical Analysis

In this section, we provide a theoretical analysis of the optimality of our method and its relationship with GLC and error propagation.

A. Equality of NRF with respect to a fixed linearization point

The first important result of NRF is that we are able to exactly represent the true marginal distribution if the linearization point is kept fixed and if the Jacobian $\mathbf{A}$ and the information matrix $\mathbf{\Omega}$ satisfy particular properties. This statement can be coalesced into the following Proposition:

Proposition VI.1. *If $\mathbf{A}$ is of full row rank, $\mathbf{\Omega} \succeq \mathbf{0}$, and $\ker \mathbf{A} = \ker \mathbf{\Omega}$, then:*

$$\mathbf{A}^\top \left(\mathbf{A}\mathbf{\Omega}^+\mathbf{A}^\top\right)^{-1}\mathbf{A} = \mathbf{\Omega}. \tag{25}$$

Proposition VI.1 is important as it is a keystone in proving equality for any nonlinear function and rank-deficient information matrices, provided that the resulting Jacobian has full row rank.

In this section we concentrate ourselves to explicitly stating equality results for the $\mathrm{SE}(n)$ case. Specifically, it is possible to achieve information matrix equality when using the cliquey dense topology, by considering a set of fully correlated $\mathrm{SE}(n)$ measurements over any spanning tree of the Markov blanket. Thus, the following Proposition holds:

Proposition VI.2. *Let $\mathcal{M}$ be a connected factor graph composed of only $\mathrm{SE}(n)$ nodes and (possibly correlated) $\mathrm{SE}(n)$ relative factors. Then, if the linearization point is kept fixed, the removal of any number of nodes from $\mathcal{M}$ can be carried out with no approximation in terms of information matrix by using fully correlated relative $\mathrm{SE}(n)$ measurements alone.*

B. Non optimality of error propagation with respect to KLD

To limit the the introduction of too many nodes in the graph, many approaches rely on a pre-processing step, where short sequences of relative rigid body transformations are composed into a single one. The general approach is to create new factors, whose mean is the result of the composition and whose covariance is obtained via linear error propagation.

If we denote by $\boldsymbol{\delta}_i$ and $\mathbf{\Sigma}_i$ respectively the mean and covariance of the i-th relative rigid body transformation, then the approach computes an $\mathrm{SE}(n)$ measurement with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ as follows:

$$\boldsymbol{\mu} = \bigoplus_i \boldsymbol{\delta}_i, \tag{26}$$

$$\mathbf{\Sigma} = \sum_i \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\delta}_i} \mathbf{\Sigma}_i \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\delta}_i}^\top. \tag{27}$$

Since the approach is equivalent to removing a chain of nodes, we analyze its optimality with respect to the Kullback-Leibler divergence and compare it with NFR. The first important result is that error propagation is in general not optimal. As a counter example, let us consider the following relative $\mathrm{SE}(2)$ rigid body transformations:

$$\boldsymbol{\delta}_1 = \begin{bmatrix} 0 & 0 & \frac{\pi}{2} \end{bmatrix}^\top, \qquad \boldsymbol{\delta}_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top, \tag{28}$$

$$\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}. \tag{29}$$

Applying error propagation, it is straightforward to numerically verify that it will yield the following covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 8 & 3 \\ 1 & 3 & 4 \end{bmatrix}. \tag{30}$$

On the other hand, the optimal covariance matrix, computed according to Proposition IV.2, is:

$$\mathbf{X}^{-1} = \begin{bmatrix} 4 & 0 & -1 \\ 0 & 8 & 3 \\ -1 & 3 & 4 \end{bmatrix}. \tag{31}$$

Despite often being similar, the two covariances will in general be different, hence linear error propagation is not optimal in terms of Kullback-Leibler divergence. On the contrary, NFR is optimal, given Proposition VI.2 and the fact

that the single edge is equivalent to the dense topology in this case. We thus recommend the use of NFR even with tasks as trivial as composing multiple odometry measurements.

C. Equivalence between NFR and GLC

In this section we present two equivalence results between NFR and GLC when using only relative $\mathrm{SE}(n)$ measurements. The first result proves the equivalence between the dense versions of the two methods, while the second proves the equivalence between their tree-based counterparts. The results are important, since they show that in this case, GLC with reparametrization is indeed a special instance of NFR.

To prove the results, we first need to characterize the structure of a GLC dense factor and prove it can be expressed in the framework of NFR. This is accomplished by the following Proposition:

Proposition VI.3. *A dense GLC factor with* $\mathrm{SE}(n)$ *reparametrization is strictly equivalent to a factor with measurement function:*

$$\mathbf{r}(\mathbf{x}) = \mathbf{r}\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \vdots\\ \mathbf{x}_m\end{bmatrix}\right) = \begin{bmatrix}\ominus\mathbf{x}_1\\ \ominus\mathbf{x}_1 \oplus \mathbf{x}_2\\ \vdots\\ \ominus\mathbf{x}_1 \oplus \mathbf{x}_m\end{bmatrix} \tag{32}$$

Proposition VI.3 shows that we can express GLC with reparametrization as a nonlinear factor, whose nonlinear function is expressed by $\mathbf{r}(\mathbf{x})$. The nonlinear function $\mathbf{r}(\mathbf{x})$, however, is different from the one we consider in this article, since we only use relative measurements. Despite this, we can still prove equivalence, according to the following Proposition:

Proposition VI.4. *Let* $\mathcal{M}$ *be a connected factor graph composed of only* $\mathrm{SE}(n)$ *nodes and (possibly correlated)* $\mathrm{SE}(n)$ *relative factors. Then, if we remove a node from* $\mathcal{M}$ *and approximate the result via dense GLC, the resulting factor is strictly equivalent to a set of fully correlated relative* $\mathrm{SE}(n)$ *measurements with star topology.*

Similarly, we can prove equivalence between the tree-based approximations of GLC and NFR, owing to the following Proposition:

Proposition VI.5. *Let* $\mathcal{M}$ *be a connected factor graph composed of only* $\mathrm{SE}(n)$ *nodes and (possibly correlated)* $\mathrm{SE}(n)$ *relative factors. Then, if we remove a node from* $\mathcal{M}$ *and approximate the result via sparse GLC, each resulting factor is strictly equivalent to an* $\mathrm{SE}(n)$ *measurement.*

VII. Experiments

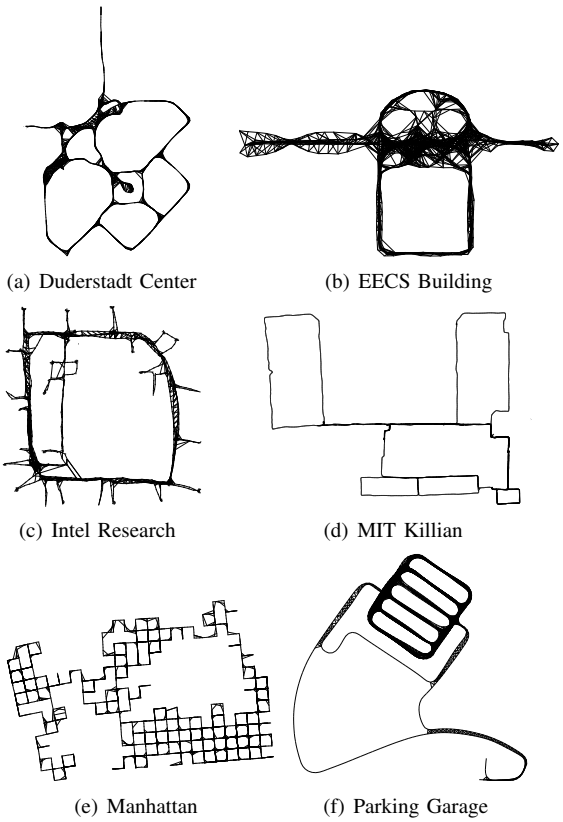
We implemented NFR by using as a factor graph optimization back-end g^{2}o [Kümmerle et al., 2011], and evaluated its accuracy when compared to the version of GLC using $\mathrm{SE}(n)$ reparametrization [Carlevaris-Bianco et al., 2014] on both two- and three-dimensional public datasets.

While in our previous work [Mazuran et al., 2014] we relied on the implementation of GLC provided with iSAM [Kaess et al., 2007], for this article we reimplemented the algorithm on g^{2}o. This choice was made in order to provide a fair comparison between the two methods for 3D datasets, as g^{2}o and iSAM parametrize and optimize differently on the $\mathrm{SE}(3)$ manifold. As a matter of fact, g^{2}o computes the errors in terms of condensed quaternions (quaternions without the real part), while iSAM in terms of Euler angles, further, g^{2}o uses symbolic gradients, while iSAM numerical ones.

While, for the most part, the updated $\mathrm{SE}(2)$ results agree with our previous values [Mazuran et al., 2014], in some instances the new implementation does provide slightly different KLD values, although the changes are not significant. The differences might be due to the fact that, in the iSAM implementation of GLC, the gradient of the reparametrization is computed numerically, even for $\mathrm{SE}(2)$ nodes.

TABLE I
Experimental datasets

Dataset	Type	# Nodes	# Edges	Fill-in
Duderstadt Center	$\mathrm{SE}(2)/\mathrm{SE}(3)$	545	1800	1.32%
EECS Building	$\mathrm{SE}(2)/\mathrm{SE}(3)$	615	2134	1.25%
Intel Research	$\mathrm{SE}(2)$	943	1833	0.52%
MIT Killian	$\mathrm{SE}(2)$	5489	7626	0.069%
Manhattan	$\mathrm{SE}(2)$	3500	5596	0.12%
Parking Garage	$\mathrm{SE}(3)$	1661	6275	0.52%



(a) Duderstadt Center (b) EECS Building (c) Intel Research (d) MIT Killian (e) Manhattan (f) Parking Garage

Fig. 4. Datasets considered in the experimental evaluation of this article.

A. Experimental setup

In order to evaluate different aspects of both GLC and our method we devised three different test applications: a full

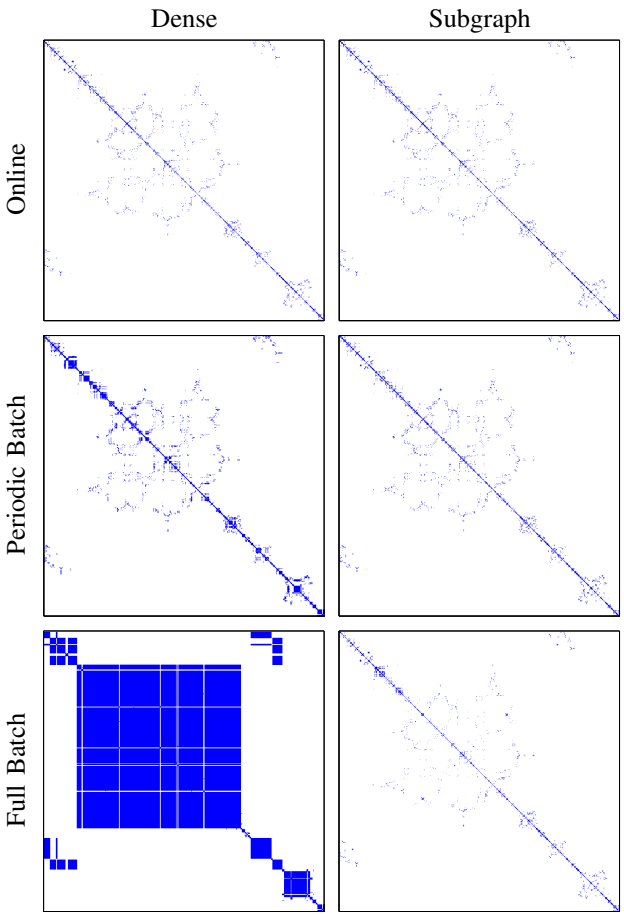


Fig. 5. Sparsity pattern of the Manhattan dataset with 80% reduction. The figure shows the fill-in of the information matrix for the dense and subgraph-based marginalization using the three strategies.

online node removal scenario, a periodic batch scenario and a full batch scenario.

In the full online scenario we build a factor graph incrementally: at each iteration we append a node to the graph, connecting it to the other ones via factors. Following the removal strategy described in [Carlevaris-Bianco and Eustice, 2013b, Alg. 3], we then remove the node if it is deemed spatially redundant. In the periodic batch scenario we, again, build the factor graph incrementally, however, instead of choosing whether to remove a node at each iteration, we do so once every 100 node insertions, and remove all the spatially redundant nodes that were added since the last removal. In the full batch scenario we take the complete graph and remove all spatially redundant nodes at the same time. As determining which nodes to remove is out of the scope of this article, we simulate the test on spatial redundancy by trivially keeping one node every t time steps.

The first two scenarios serve as representative test for typical online sparsification choices. The last, on the other hand, is meant as a stress test in which the linearization point is closest to the optimal solution, favoring node removal methods that rely on the global linearization point, such as GLC.

For each scenario we keep both a sparsified and baseline graph. We update the latter whenever we add a new node to the sparsified graph, without the further step of removing nodes. We then use the baseline graph as a target distribution (in terms of mean and covariance of the full estimate) against which we evaluate the KLD of our sparse approximation.

It should be noted that in the online and periodic batch scenario, it may not be possible to add a factor to the incremental graph if it connects a node which has already been removed. In such instances we instead add a factor to the closest existing node, for both the sparsified and baseline graph, and compute a new measurement accordingly. It is thus important to note that since the baseline comparison varies across the node removal scenarios, the Kullback-Leibler divergence is not comparable across different tests.

We evaluated the node removal approaches on publicly available datasets, five stemming from real data and a synthetic one, respectively, Duderstadt Center, EECS Building, Intel Research, MIT Killian, Parking Garage, and Manhattan. Of those, Duderstadt Center, EECS Building, and Parking Garage involve $\mathrm{SE}(3)$ poses. Since in our previous work [Mazuran et al., 2014] we projected Duderstadt Center and EECS Building onto the $\mathrm{SE}(2)$ manifold, in this article we consider both two- and three-dimensional realizations of the datasets. In Table I we provide a short overview of the datasets we considered, while Fig. 4 visualizes the full factor graphs of each dataset.

We considered four degrees of node reduction, respectively keeping one in two, three, four, or five nodes. Further, we evaluated the following approaches, where for NFR we use in all instances either relative $\mathrm{SE}(2)$ or $\mathrm{SE}(3)$ measurements:

- *GLC-Tree/NFR-Tree-Global*: GLC with Chow-Liu tree approximation. Note that, due to Proposition VI.5 and the optimality of GLC, in our test scenarios, this is *exactly* equivalent to NFR with Chow-Liu tree topology on the global linearization point. This prediction agrees with the numerical data, we thus report the two as the same method.
- *GLC-Dense/NFR-Cliquey-Dense*: GLC with dense factors. Note that, due to Proposition VI.4, this is equivalent to NFR with cliquey dense topology on the global linearization point, if the spanning tree is in fact a star. While the latter is in general not true, the results differ by such a negligible amount that we decided to group them together.
- *NFR-Tree-Local*: Chow-Liu tree topology on the local linearization point.
- *NFR-Subgraph-Global*: Subgraph topology with twice as many edges as the spanning tree, on the global linearization point.
- *NFR-Subgraph-Local*: The same as NFR-Subgraph-Global but on the local linearization point.
- *NFR-Cliquey-Subgraph*: Cliquey subgraph topology with the same fill-in limit as NFR-Subgraph-Local/Global, on the global linearization point.

We did not consider the cliquey subgraph and cliquey dense topologies on the local linearization point because they

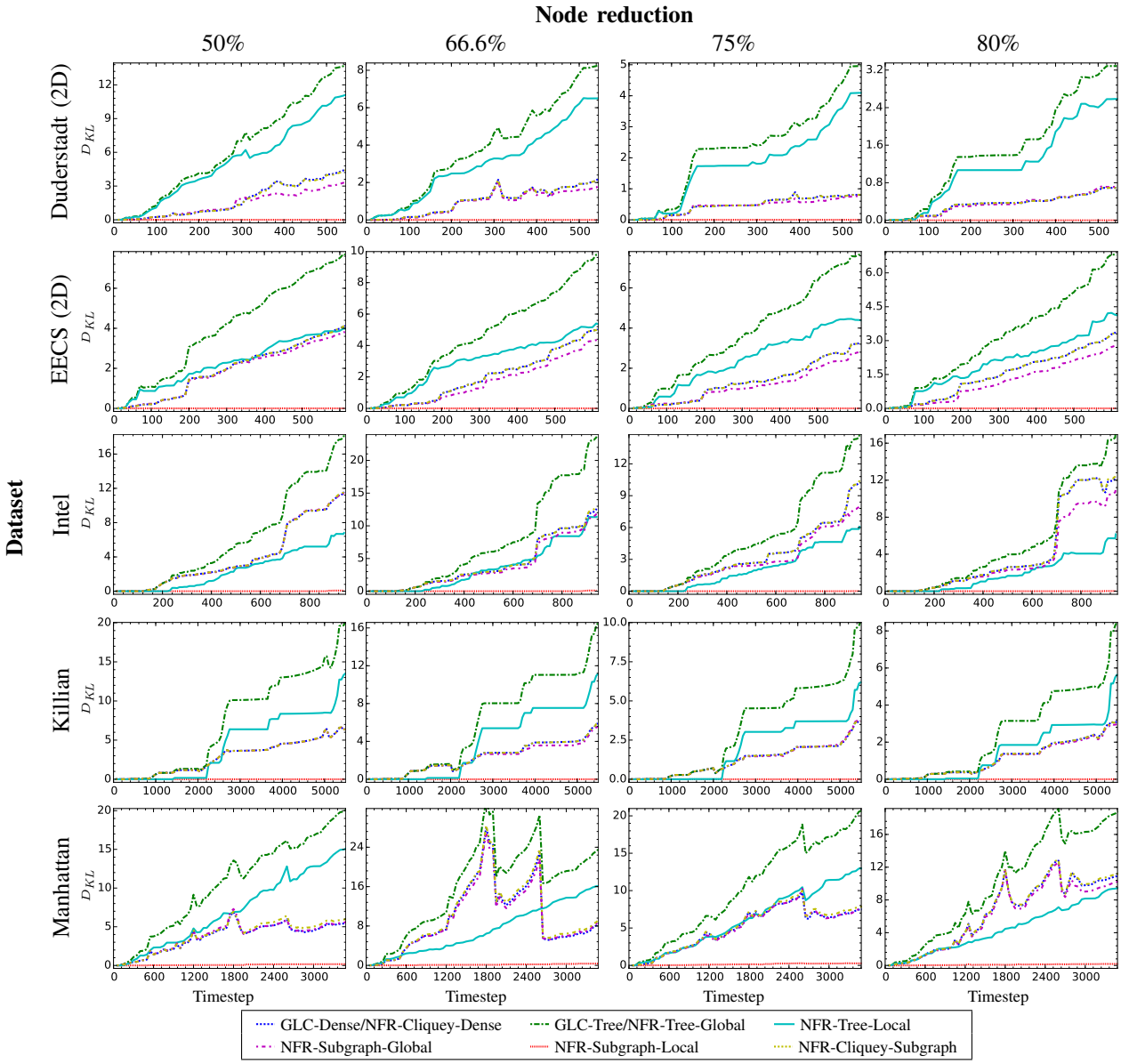


Fig. 6. Online sparsification results for 2D datasets. Each graph represents the KLD as a function of the time step for the six methods considered.

often cause the optimizer to diverge. This is possibly due to the fact that the linear correlations between measurements are not adequately preserved when the linearization point changes considerably. Furthermore, contrary to our previous work [Mazuran et al., 2014], we do not report the results for the dense relative measurement topology since it is practical only for online sparsification scenarios, and in such instances it is virtually equivalent to NFR-Subgraph-Local/Global.

In order to convey the difference in sparsity between using a subgraph and dense node removal, we report in Fig. 5 the sparsity pattern of the information matrix of the Manhattan dataset in the three different scenarios. While both dense and subgraph approaches maintain a adequately sparse information matrix for the full online scenario, the dense approach quickly degenerates for the other ones. The subgraph approaches, on the other hand, maintain sparsity even in the full batch scenario, producing accuracy results only bested by the dense approach.

B. *Evaluation criteria*

We focus the evaluation of the aforementioned approaches on the Kullback-Leibler divergence between the overall sparsified and baseline graphs, as it jointly captures differences in both estimate and information matrix. Specifically, we adopt the same formula as (2), but instead of considering only the Markov blanket of one variable we set $\boldsymbol{\nu}$ and $\boldsymbol{\Upsilon}$ to be the estimate and information matrix of the complete sparsified graph, while we set $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to be the estimate and covariance matrix of the nodes of the baseline graph that have not been removed in the sparsified graph.

Further, we note that the SLAM optimization problem is

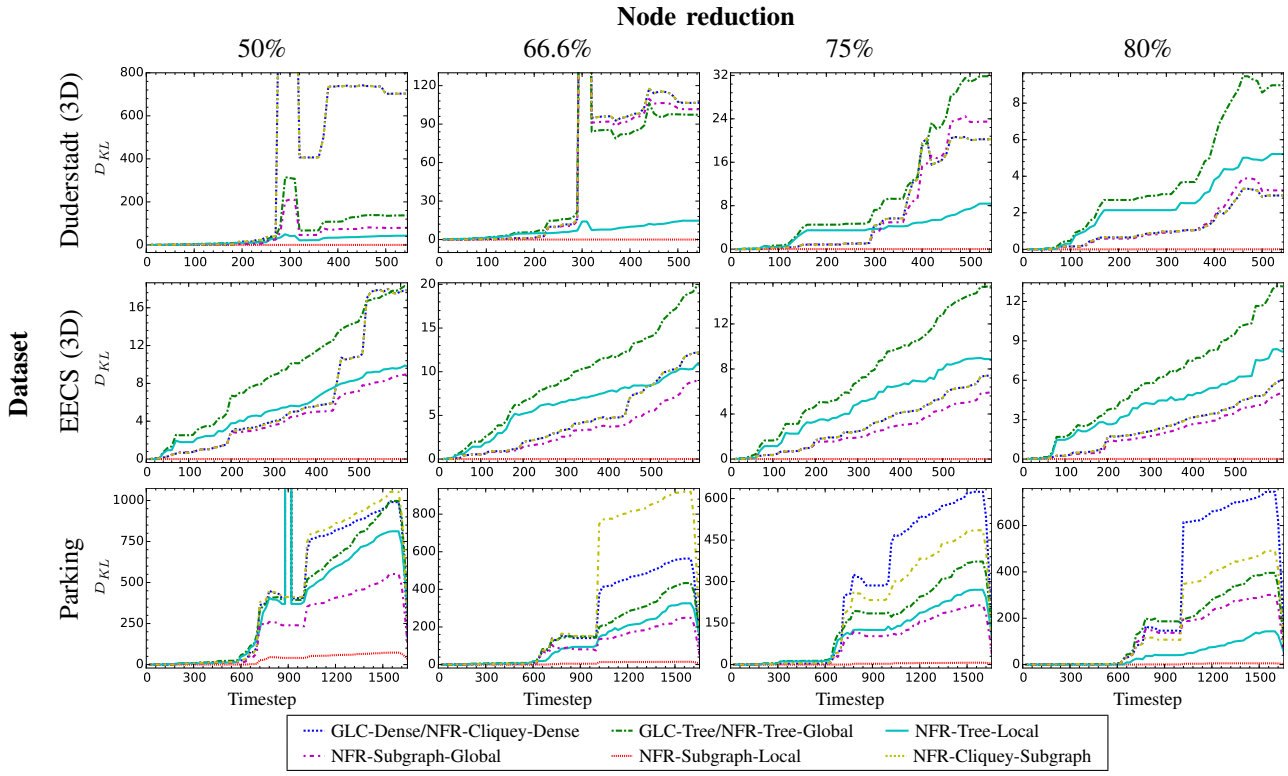


Fig. 7. Online sparsification results for 3D datasets. Each graph represents the KLD as a function of the time step for the six methods considered.

not formulated on a vector space, but rather on a differentiable manifold, where the matrices $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Sigma}$ express uncertainties with respect to a particular parameterization of the manifold that is implementation dependent. We thus replace the vector difference $\boldsymbol{\nu} - \boldsymbol{\mu}$ with the group difference associated to the manifold and map it according to the particular parameterization used by the optimizer. For instance, when dealing with $\mathrm{SE}(n)$ estimates, we consider the substitution

$$\boldsymbol{\nu} - \boldsymbol{\mu} \mapsto \begin{bmatrix} \boldsymbol{\psi}(\ominus \boldsymbol{\mu}_1 \oplus \boldsymbol{\nu}_1) \\ \vdots \\ \boldsymbol{\psi}(\ominus \boldsymbol{\mu}_o \oplus \boldsymbol{\nu}_o) \end{bmatrix}. \tag{33}$$

Here, an i subscript references the estimate of the i-th node of either the sparsified or baseline graph, o is the number of poses in the sparsified graph, and $\boldsymbol{\psi}$ is a homeomorphism that maps an $\mathrm{SE}(n)$ group element to the corresponding vector space defined by the chart. For instance, in the $\mathrm{SE}(3)$ implementation of g^{2}o [Kümmerle et al., 2011] this would be a 6-dimensional vector containing x, y, z coordinates and the condensed quaternion representation of the rotation.

In addition to the KLD we also consider to a limited extent an evaluation in terms of χ^2 value. This is in order to quantify the accuracy loss in the estimate alone, instead of considering it together with the discrepancy in information matrices.

We use an approach similar to the one proposed by Huang et al. [2009]: we substitute the estimated poses of the sparsified graph into the baseline one and then optimize the baseline graph with respect to the nodes not present in the sparsified graph. We then compute the resulting χ^2 value, which we refer to as χ^2_s and compare it with the χ^2 of the baseline graph, χ^2_b.

Contrary to Huang et al. [2009], we choose to evaluate the percentage increase in terms of χ value, i.e.

$$\Delta\chi_{\%} = 100 \cdot \frac{\chi_s - \chi_b}{\chi_b}. \tag{34}$$

This is for two reasons: first, a change in the estimate results in a roughly proportional, rather than quadratic, variation in terms of $\Delta\chi_{\%}$, which makes it a more intuitive value. Second, the χ^2 values vary wildly across the datasets, even when normalized by the number of degrees of freedom. This effectively results in different scales for different datasets, which does not allow to compute overall statistics.

C. Accuracy results

We report in Fig. 6 and in Fig. 7 the KLD results for the full online scenario, respectively for the two- and three-dimensional datasets. In both instances, NFR-Subgraph-Local achieves almost a numerical zero in terms of KLD, thus proving to be a near-optimal choice for online node removal. Further, in all instances, NFR-Tree-Local achieves superior KLD results than its global counterpart, suggesting that it is indeed a better choice to use a local linearization point in the absence of accurate information on the final linearization point.

Interestingly, less sparse approximations on the global linearization point do not improve significantly the KLD results, in fact, in some instances they may be even counterproductive.

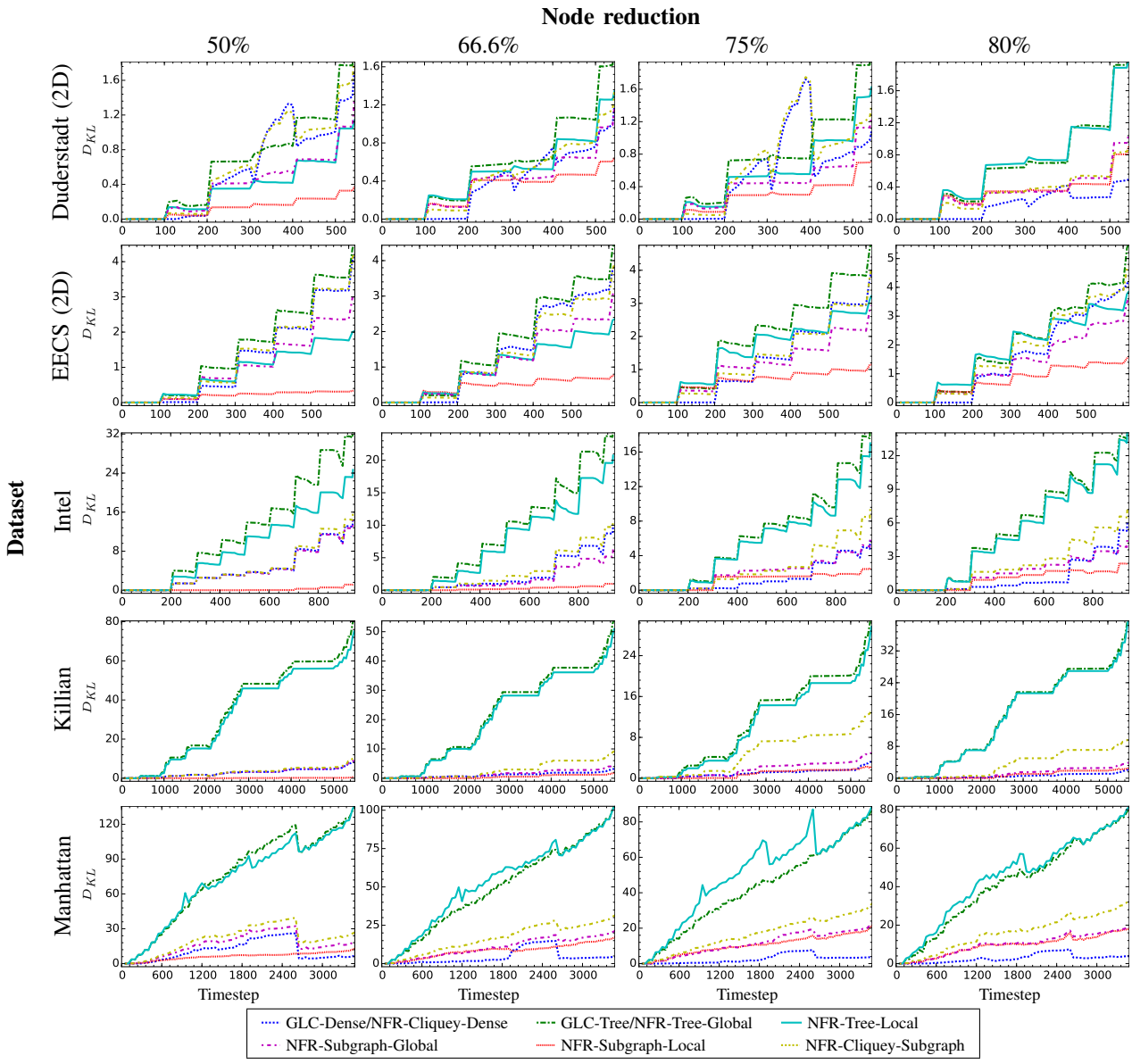


Fig. 8. Periodic batch sparsification results for 2D datasets. Each graph represents the KLD as a function of the time step for the six methods considered.

This is particularly the case for the Parking Garage dataset, where NFR-Cliquey-Subgraph and GLC-Dense/NFR-Cliquey-Dense consistently achieve the worst results. This may be the effect of considering only linear correlation between the measurements in the Markov blanket, as NFR-Subgraph-Global tends to obtain more competitive results. Having said this, for a full online approach, dense approximations still prove to be efficient from a sparsity perspective, and in all instances they achieve less than a 2% increase in fill-in when compared to the tree approach. In fact, removing nodes at each iteration ensures that there is never spatial redundancy, therefore, due to the limited range of sensors, the connectivity of a node cannot increase significantly, which translates to a less pronounced fill-in, even for dense node removal.

The $\mathrm{SE}(3)$ datasets, in general, prove to be a challenge for all methods. In fact, the strong nonlinearities of relative $\mathrm{SE}(3)$ transformations cause, at times, the optimization to get stuck in a local minimum. This is most evident in the Duderstadt Center dataset at 50% and 66.6% node reduction and the Parking Garage dataset at 50%, where the large discontinuities in KLD happen when the optimizer converges to a local, suboptimal, attractor. This shouldn't be viewed as failure on the side of the sparsification method, but rather as an inevitable byproduct of the inherent complexity of optimization on the $\mathrm{SE}(3)$ manifold. Indeed, this problem is shared by all the tested methods, and while for NFR-Subgraph-Local it has not surfaced in the experiments, it is not unreasonable to conjecture that it too is not immune.

We report in Fig. 8 and Fig. 9 the KLD results for the periodic batch sparsification scenario, respectively for two- and three-dimensional datasets. The graphs exhibit a *saw-tooth* behavior due to the periodic node removal that happens in

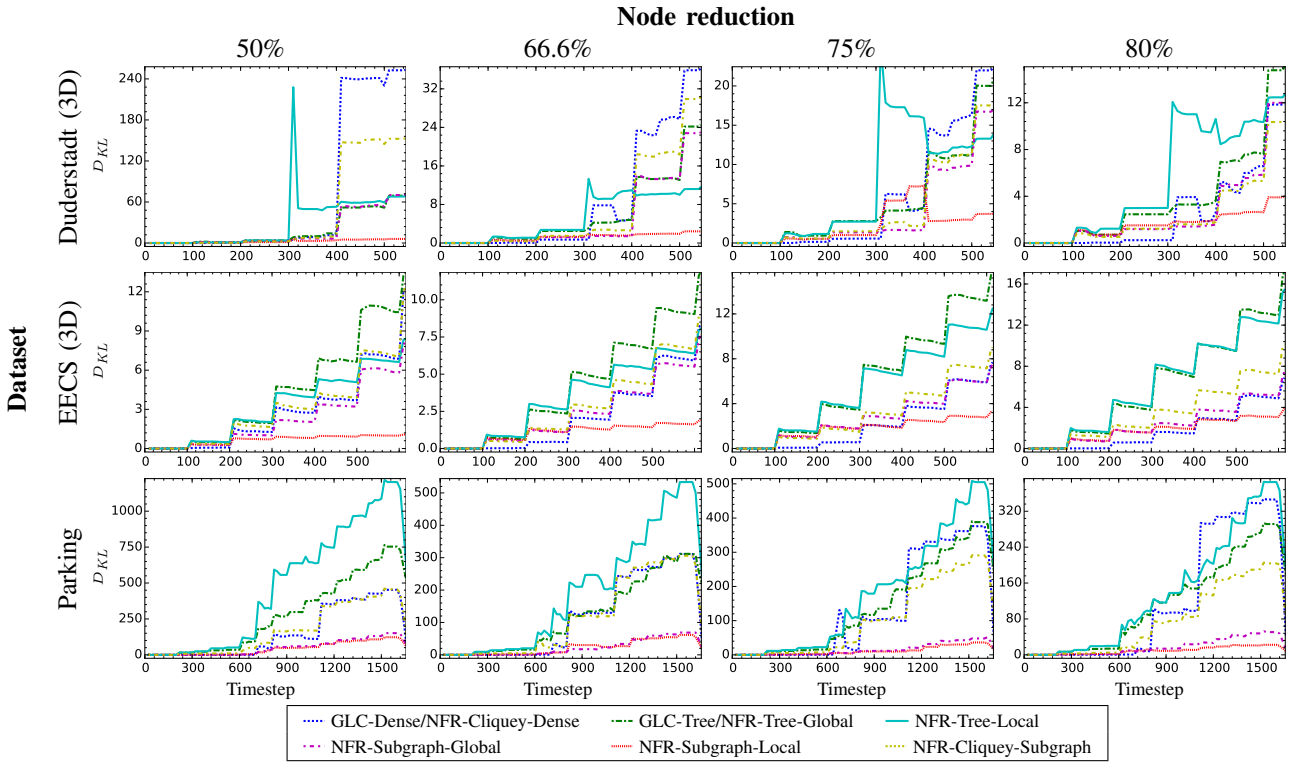


Fig. 9. Periodic batch sparsification results for 3D datasets. Each graph represents the KLD as a function of the time step for the six methods considered.

batches, thus resulting in localized sharp increases in KLD.

In this scenario, save for the Manhattan dataset, NFR-Subgraph-Local again achieves the best results. As for NFR-Tree-Local this time the results are mixed. In two-dimensional datasets, with the exception of Manhattan, the local linearization point generally proves to be better than its global counterpart. For three-dimensional datasets, only EECS Building shows consistent improvements.

For Manhattan and Parking Garage it is possible to motivate this behavior due to the datasets' particular structure. In the Manhattan dataset, in fact, small loop closures happen with a much higer frequency than that of the sparsification (every 100 iterations). This results in very rigidly constrained Markov blankets, which are better approximated on the global linearization point. This same effect is present in the Parking Garage dataset, due to the fact that it is locally highly connected. For the three-dimensional Duderstadt dataset the behavior is much more erratic. NFR-Tree-Local provides worse results than GLC-Tree halfway through the dataset, but leads to improvements near the end. This may, again, be caused by the optimization algorithm converging to a local attractor, as GLC-Dense and NFR-Cliquey-Subgraph result in steep and unexpectedly large KLD changes as well.

Although in many instances GLC-Dense performs well, it is not uncommon for it to provide worse results than a tree on the global linearization point, similarly to the online sparsification scenario. This is most evident in the three-dimensional Duderstadt dataset and in the Parking dataset. NFR-Cliquey-Subgraph provides results between those of GLC-Dense and GLC-Tree, for better or worse, while NFR-Subgraph-Global tends to provide results in the range of GLC-Dense without suffering from the large increases in KLD that at times plague it.

Contrary to the online scenario, this time the added fill-in of a dense node removal is significant and, in practice, unacceptable for real-time execution, with a maximum increase of more than 20% when compared to a tree. The subgraph approaches, on the other hand produce a much sparser graph, with a maximum increase of about 4%.

Table II presents the numeric results in terms of KLD and fill-in for the batch sparsification scenario. Note that we replaced any KLD value below 10^{-8} with a numerical zero, the reasoning being the finite accuracy of IEEE 754 double precision numbers.

The batch scenario is clearly more favorable towards the use of the global linearization point, where, save for a few exceptions, the local linearization point methods are consistently outperformed by their global counterpart. Further, in this case, GLC-Dense achieves the best results in terms of KLD, often with a practically nil value. However, with the exception of Killian, which is sparsely connected, this comes at a gross and unacceptable cost in sparsity, with EECS even resulting in a full fill-in.

In the tested datasets all of the subgraph approaches provide an often significant decrease in KLD when compared to their tree counterparts, at an acceptable cost in sparsity. There is no clear preferred choice as to which subgraph approach should be used, as any of the three achieves the minimum value

Dataset	Approach	Node reduction level							
		50%		66.6%		75%		80%	
		KLD	Fill-in	KLD	Fill-in	KLD	Fill-in	KLD	Fill-in
Duderstadt (2D)	GLC-Tree/NFR-Tree-Global	1.186	**1.42%**	1.425	**2.14%**	1.938	**2.77%**	2.463	**3.38%**
	NFR-Tree-Local	1.307	**1.42%**	1.695	**2.14%**	2.233	**2.77%**	2.870	**3.38%**
	NFR-Subgraph-Global	0.915	2.24%	1.044	3.11%	1.481	3.92%	1.541	4.85%
	NFR-Subgraph-Local	0.975	2.24%	1.179	3.11%	1.628	3.92%	1.809	4.85%
	NFR-Cliquey-Subgraph	**0.800**	2.05%	**0.723**	3.02%	**0.999**	3.90%	**1.327**	4.75%
	GLC-Dense/NFR-Cliquey-Dense	3.11e-4	21.2%	0	77.1%	2.02e-4	80.2%	0	84.9%
EECS (2D)	GLC-Tree/NFR-Tree-Global	1.975	**1.87%**	2.448	**2.99%**	3.978	**3.94%**	4.761	5.01%
	NFR-Tree-Local	2.519	**1.87%**	3.770	**2.99%**	6.180	**3.94%**	7.322	**5.00%**
	NFR-Subgraph-Global	1.323	2.91%	1.919	4.59%	2.666	6.28%	3.141	8.15%
	NFR-Subgraph-Local	1.672	2.91%	3.006	4.58%	4.200	6.28%	4.677	8.16%
	NFR-Cliquey-Subgraph	**1.267**	2.82%	**1.822**	4.53%	**2.540**	6.37%	**3.092**	8.39%
	GLC-Dense/NFR-Cliquey-Dense	2.22e-3	80.1%	0	100%	0	100%	0	100%
Intel	GLC-Tree/NFR-Tree-Global	46.49	**0.89%**	43.52	**1.27%**	39.71	1.64%	41.69	**1.91%**
	NFR-Tree-Local	55.79	**0.89%**	52.95	**1.27%**	48.78	**1.63%**	50.02	1.92%
	NFR-Subgraph-Global	**14.96**	1.22%	**20.26**	1.77%	**17.41**	2.25%	**16.89**	2.65%
	NFR-Subgraph-Local	17.07	1.22%	22.73	1.77%	19.30	2.25%	18.37	2.66%
	NFR-Cliquey-Subgraph	20.87	1.24%	22.31	1.76%	24.43	2.28%	20.19	2.74%
	GLC-Dense/NFR-Cliquey-Dense	0.881	3.16%	0.199	14.4%	0	66.1%	0	71.5%
Killian	GLC-Tree/NFR-Tree-Global	75.13	**0.13%**	151.1	**0.18%**	73.42	**0.24%**	129.1	**0.29%**
	NFR-Tree-Local	75.06	**0.13%**	154.0	**0.18%**	76.52	**0.24%**	132.9	**0.29%**
	NFR-Subgraph-Global	2.062	0.17%	45.58	0.25%	**17.01**	0.34%	**39.84**	0.40%
	NFR-Subgraph-Local	**0.530**	0.17%	46.16	0.25%	17.62	0.34%	40.55	0.40%
	NFR-Cliquey-Subgraph	3.416	0.16%	**44.59**	0.25%	24.33	0.34%	50.17	0.39%
	GLC-Dense/NFR-Cliquey-Dense	1.548	0.17%	0	0.40%	5.72e-3	0.52%	0	1.47%
Manhattan	GLC-Tree/NFR-Tree-Global	204.8	**0.26%**	167.0	**0.39%**	150.3	**0.52%**	144.2	0.65%
	NFR-Tree-Local	213.4	**0.26%**	172.5	**0.39%**	159.3	**0.52%**	154.1	**0.64%**
	NFR-Subgraph-Global	33.22	0.38%	46.30	0.62%	**58.33**	0.79%	**58.23**	0.95%
	NFR-Subgraph-Local	**32.43**	0.38%	**46.29**	0.62%	61.73	0.78%	60.51	0.95%
	NFR-Cliquey-Subgraph	47.13	0.36%	60.80	0.60%	71.28	0.80%	68.98	1.00%
	GLC-Dense/NFR-Cliquey-Dense	2.314	0.55%	0.775	2.54%	0	11.7%	0	35.3%
Duderstadt (3D)	GLC-Tree/NFR-Tree-Global	8.516	**1.42%**	6.477	**2.09%**	7.388	**2.79%**	9.743	**3.37%**
	NFR-Tree-Local	72.04	**1.42%**	19.49	**2.09%**	18.19	**2.79%**	19.86	**3.37%**
	NFR-Subgraph-Global	6.029	2.24%	3.923	3.07%	**5.268**	3.89%	5.900	4.49%
	NFR-Subgraph-Local	24.65	2.24%	7.568	3.08%	9.113	3.89%	8.955	4.49%
	NFR-Cliquey-Subgraph	**4.841**	2.04%	**3.576**	3.04%	5.637	3.70%	**4.693**	4.68%
	GLC-Dense/NFR-Cliquey-Dense	1.687	21.2%	1.27e-5	77.1%	4.65e-2	80.2%	4.72e-6	84.9%
EECS (3D)	GLC-Tree/NFR-Tree-Global	7.927	**1.85%**	10.26	2.96%	15.98	**3.86%**	19.23	**4.81%**
	NFR-Tree-Local	10.32	**1.85%**	13.11	**2.95%**	21.34	**3.86%**	24.38	**4.81%**
	NFR-Subgraph-Global	**4.667**	2.92%	7.795	4.71%	**8.623**	6.37%	**12.09**	7.88%
	NFR-Subgraph-Local	5.639	2.92%	9.339	4.73%	11.32	6.34%	15.66	7.88%
	NFR-Cliquey-Subgraph	5.084	2.84%	**6.857**	4.50%	9.960	6.37%	12.63	8.02%
	GLC-Dense/NFR-Cliquey-Dense	2.95e-3	80.1%	1.29e-5	100%	8.56e-6	100%	4.54e-6	100%
Parking	GLC-Tree/NFR-Tree-Global	730.7	**0.40%**	461.2	**0.60%**	373.0	**0.78%**	311.0	**0.97%**
	NFR-Tree-Local	859.4	0.41%	578.8	**0.60%**	462.7	**0.78%**	395.7	**0.97%**
	NFR-Subgraph-Global	**169.4**	0.69%	**138.9**	0.99%	**113.3**	1.29%	**104.3**	1.58%
	NFR-Subgraph-Local	236.8	0.69%	190.3	0.99%	151.8	1.29%	150.2	1.58%
	NFR-Cliquey-Subgraph	315.7	0.60%	248.8	0.83%	182.5	1.09%	148.0	1.34%
	GLC-Dense/NFR-Cliquey-Dense	1.52e-4	12.4%	7.94e-5	17.3%	4.87e-5	28.3%	6.61e-5	46.2%

TABLE II
FULL BATCH SPARSIFICATION RESULTS

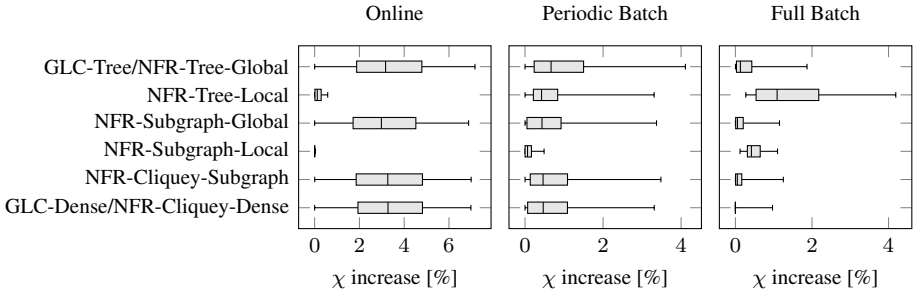


Fig. 10. Box plots of the percentage increase in χ with respect to the original factor graph, for the considered removal strategies and sparsification scenarios.

in some particular test. It is to be said, however, that since NFR-Cliquey-Subgraph has a closed form expression, from a computational standpoint it is significantly more efficient than NFR-Subgraph-Local/Global, especially if the size of the Markov blanket is significant.

Finally, we report in Fig. 10 the overall results in terms of $\Delta\chi_{\%}$ for all sparsification scenarios over all of the datasets considered. We report the results as box plots, displaying respectively the 5%, 25%, 50%, 75%, and 95% percentiles for the χ increase. For the online and periodic batch scenarios, the values are computed over the whole runs, while for the full batch result they represent statistics only on the overall $\Delta\chi_{\%}$.

The results agree to some extent with the KLD ones, although the differences between the methods are either magnified or shrunk. For instance, there is even more of a clear advantage to using the local linearization point for online scenarios, but the differences between other methods are not as clear-cut. This suggests that in such cases the KLD is strongly affected by the discrepancy in information matrices. For the periodic batch scenario, the local linearization point approaches overall provide either the best or competitive results, while the opposite is true for the full batch scenario.

VIII. Conclusion

In this paper, we presented a novel approach to represent the marginal distribution induced by the removal of nodes in graph-based SLAM. The goal of our approach is to estimate both the mean and the information matrix of the set of nonlinear factors that best represent the marginal distribution. We showed that estimating the former is equivalent to evaluating the nonlinear functions at the linearization point, while estimating the latter is equivalent to solving a convex optimization problem. Our approach can be used in variety of settings, ranging from representing the exact marginalization with a dense factor to sparsely approximating it over a tree- or a subgraph-based distribution. The proposed approach has several properties. It does not necessarily require a global linearization point, it can be used with any nonlinear measurement function, and it can consider any topology of possibily correlated measurements. We presented an extensive theoretical analysis of our method and characterized its properties with respect to GLC and linear error propagation. Finally, we performed an extensive experimental analysis on publicly available datasets and demonstrated the effectiveness of our approach. We quantified the algorithm performance in the SLAM context by sparsifing maps in an online and in a periodic batch fashion. In both cases, our technique outperforms state-of-the-art methods by closely recovering the original distribution and producing highly sparse graphs.

IX. Acknowledgements

We would like to thank Nicholas Carlevaris-Bianco, for providing us with the EECS and Duderstadt datasets and for his invaluable help in the use of GLC, and Luciano Spinello, for the fruitful discussions and insight we had while developing the initial ideas behind this work. This work has partly been supported by the European Commission under ERC-AG-PE7-267686-LIFENAV and FP7-610603-EUROPA2.

Appendix

Definition A.1. *Let* $\mathbf{x} = [\, x_1 \; x_2 \; \dots \; x_n \,]^\top$ *denote an* n*-dimensional vector, then we define the selection operator* $\pi_m(\cdot)$ *with* $m \leq n$ *as:*

$$\pi_m(\mathbf{x}) = [\, x_1 \; x_2 \; \dots \; x_m \,]^\top . \tag{35}$$

With a slight abuse of notation, we further denote $\pi_m(\mathcal{S})$ *to be the image of a set* $\mathcal{S}$ *under the selection operator* $\pi_m(\cdot)$*, i.e.:*

$$\pi_m(\mathcal{S}) = \{\pi_m(\mathbf{x}) \mid \forall \mathbf{x} \in \mathcal{S}\} . \tag{36}$$

Lemma A.2. *If* $\mathbf{A}$ *is square with* $\det \mathbf{A} \neq 0$ *and can be left-multiplied to* $\mathbf{B}$*, then* $\ker(\mathbf{AB}) = \ker \mathbf{B}$.

Proof:

$$\det \mathbf{A} \neq 0 \implies (\mathbf{Ax} = \mathbf{0} \iff \mathbf{x} = \mathbf{0}) \tag{37}$$

$$\mathbf{x} \in \ker(\mathbf{AB}) \iff \mathbf{ABx} = \mathbf{0} \tag{38}$$

$$\underset{(37)}{\iff} \mathbf{Bx} = \mathbf{x} \iff \mathbf{x} \in \ker \mathbf{B}. \tag{39}$$

■

Lemma A.3. $\ker\left(\mathbf{A}^\top \mathbf{A}\right) = \ker \mathbf{A}$

Proof: Let $(\cdot)^\perp$ denote the orthogonal complement operator. Then by Fredholm's theorem we have:

$$\operatorname{im} \mathbf{A} = \left(\ker\left(\mathbf{A}^\top\right)\right)^\perp \tag{40}$$

$$\implies \left(\mathbf{A}^\top \mathbf{Ax} = \mathbf{0} \iff \mathbf{Ax} = \mathbf{0}\right). \tag{41}$$

■

Lemma A.4. *Let* $\mathbf{A} \succeq \mathbf{0}$*, and let* $\mathbf{U}\mathbf{D}\mathbf{U}^\top$ *be the rank-revealing eigen decomposition of* $\mathbf{A}$*, then* $\ker \mathbf{A} = \ker\left(\mathbf{U}^\top\right)$.

Proof: By Lemma A.2, $\ker\left(\mathbf{D}^{\frac{1}{2}}\mathbf{U}^\top\right) = \ker\left(\mathbf{U}^\top\right)$, while by Lemma A.3 $\ker\left(\mathbf{D}^{\frac{1}{2}}\mathbf{U}^\top\right) = \ker\left(\mathbf{U}\mathbf{D}\mathbf{U}^\top\right) = \ker \mathbf{A}$. Therefore $\ker \mathbf{A} = \ker\left(\mathbf{U}^\top\right)$. ∎

Lemma A.5. *Let:*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \tag{42}$$

With $\mathbf{A}_{11}$ *an* $n \times n$ *matrix,* $\mathbf{A}_{22}$ *an* $m \times m$ *matrix, and* $\det(\mathbf{A}_{22}) \neq 0$*. Then the following holds:*

$$\ker\left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right) = \pi_n(\ker \mathbf{A}) \tag{43}$$

Proof: We can prove the equality as a direct consequence of the application of the Schur complement to a system of linear equations:

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \in \ker \mathbf{A} \iff \begin{cases} \mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2 = \mathbf{0} \\ \mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 = \mathbf{0} \end{cases} \tag{44}$$

$$\implies \left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right)\mathbf{x}_1 = \mathbf{0} \tag{45}$$

$$\implies \mathbf{x}_1 \in \ker\left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right). \tag{46}$$

Let now:

$$\mathbf{y}_1 \in \ker\left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right), \tag{47}$$

$$\mathbf{y}_2 = -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{y}_1. \tag{48}$$

Then it is easy to see that $\mathbf{y}_1$ and $\mathbf{y}_2$ satisfy (44). Therefore, we conclude:

$$\ker\left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right) = \pi_n(\ker \mathbf{A}). \tag{49}$$

∎

Lemma A.6. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be full row rank matrices of the same size. Then, if* $\ker \mathbf{A} = \ker \mathbf{B}$*,* $\mathbf{A}\mathbf{B}^\top$ *is invertible and the following holds:*

$$\left(\mathbf{A}\mathbf{B}^\top\right)^{-1} = \mathbf{B}^{\mp}\mathbf{A}^{+} \tag{50}$$

Proof: We prove the assertion by directly checking the definition of inverse, i.e. the following conditions need to hold:

$$\left(\mathbf{A}\mathbf{B}^\top\right)\left(\mathbf{B}^{\mp}\mathbf{A}^{+}\right) = \mathbf{I}, \qquad \left(\mathbf{B}^{\mp}\mathbf{A}^{+}\right)\left(\mathbf{A}\mathbf{B}^\top\right) = \mathbf{I}. \tag{51}$$

By the properties of the pseudoinverse, by Fredholm's theorem, and the $\ker \mathbf{A} = \ker \mathbf{B}$ hypothesis, we have that:

$$\operatorname{im}\left(\mathbf{A}^{+}\right) = \operatorname{im}\left(\mathbf{A}^\top\right) = (\ker \mathbf{A})^\perp = (\ker \mathbf{B})^\perp. \tag{52}$$

Furthermore, by the definition of Moore-Penrose pseudoinverse we have, that:

$$\mathbf{B}^{+}\mathbf{B} = \left(\mathbf{B}^{+}\mathbf{B}\right)^\top = \mathbf{B}^\top\mathbf{B}^{\mp}. \tag{53}$$

However, we note that $\mathbf{B}^{+}\mathbf{B}$ is the orthogonal projector onto $(\ker \mathbf{B})^\perp$ [Golub and Van Loan, 1996, pp. 257-258], thus:

$$\mathbf{B}^\top\mathbf{B}^{\mp}\mathbf{A}^{+} = \mathbf{A}^{+}. \tag{54}$$

Taking into account that for any full row rank matrix $\mathbf{A}$, $\mathbf{A}^{+} = \mathbf{A}^\top\left(\mathbf{A}\mathbf{A}^\top\right)^{-1}$, we have:

$$\mathbf{A}\mathbf{B}^\top\mathbf{B}^{\mp}\mathbf{A}^{+} = \mathbf{A}\mathbf{A}^{+} = \mathbf{A}\mathbf{A}^\top\left(\mathbf{A}\mathbf{A}^\top\right)^{-1} = \mathbf{I}. \tag{55}$$

Similarly, for the second condition, we have:

$$\operatorname{im}\left(\mathbf{B}^\top\right) = (\ker \mathbf{B})^\perp = (\ker \mathbf{A})^\perp. \tag{56}$$

Since $\mathbf{A}^{+}\mathbf{A}$ is the orthogonal projector onto $(\ker \mathbf{A})^\perp$, we have:

$$\mathbf{A}^{+}\mathbf{A}\mathbf{B}^\top = \mathbf{B}^\top. \tag{57}$$

Thus, finally:

$$\mathbf{B}^{\mp}\mathbf{A}^{+}\mathbf{A}\mathbf{B}^\top = \mathbf{B}^{\mp}\mathbf{B}^\top = \left(\mathbf{B}\mathbf{B}^\top\right)^{-\top}\mathbf{B}\mathbf{B}^\top = \mathbf{I}. \tag{58}$$

∎

Lemma A.7. *Let* $\mathbf{x}$ *denote any minimal vector representation of* m $\mathrm{SE}(n)$ *poses stacked together, and let* $\mathbf{y} = \pi_k(\mathbf{x})$*, where* k *is divisible by the dimension of a pose and the quotient is strictly smaller than* m*. Let also* $\mathbf{f}(\mathbf{x})$ *and* $\mathbf{g}(\mathbf{y})$ *be vector functions obtained by stacking any number of* $\mathrm{SE}(n)$ *relative measurements, respectively between the poses in* $\mathbf{x}$ *and* $\mathbf{y}$*. Furthermore, let* $\mathbf{f}(\mathbf{x})$ *(resp.* $\mathbf{g}(\mathbf{y})$*) be such that there is a chain of relative* $\mathrm{SE}(n)$ *measurements that connects any two poses in* $\mathbf{x}$ *(resp.* $\mathbf{y}$*). Then:*

$$\ker\left(\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\right) = \pi_k\left(\ker\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\right). \tag{59}$$

Proof: Let $\mathbf{z}$ be a vector of poses in the same format of $\mathbf{x}$. Let us also define the binary operator $\oplus$ between the vector representation $\boldsymbol{\delta}$ of an $\mathrm{SE}(n)$ pose and $\mathbf{z}$ as follows:

$$\boldsymbol{\delta} \oplus \mathbf{z} = \begin{bmatrix} (\boldsymbol{\delta} \oplus \mathbf{z}_1)^\top & (\boldsymbol{\delta} \oplus \mathbf{z}_2)^\top & \cdots & (\boldsymbol{\delta} \oplus \mathbf{z}_m)^\top \end{bmatrix}^\top. \tag{60}$$

Then, the following holds:

$$\mathbf{f}(\mathbf{z}) = \mathbf{f}(\boldsymbol{\delta} \oplus \mathbf{z}) \quad \forall \boldsymbol{\delta}. \tag{61}$$

This comes from noting that, by assumption, $\mathbf{f}(\mathbf{z})$ is given by stacking vector functions of the form $\ominus \mathbf{z}_i \oplus \mathbf{z}_j$, hence:

$$\ominus(\boldsymbol{\delta} \oplus \mathbf{z}_i) \oplus (\boldsymbol{\delta} \oplus \mathbf{z}_j) = \ominus \mathbf{z}_i \oplus (\ominus \boldsymbol{\delta} \oplus \boldsymbol{\delta}) \oplus \mathbf{z}_j = \ominus \mathbf{z}_i \oplus \mathbf{z}_j. \tag{62}$$

Let, now, $\mathbf{x} = \boldsymbol{\delta} \oplus \boldsymbol{z}$ and let us consider the gradient of $\mathbf{f}(\mathbf{x})$ with respect to $\boldsymbol{\delta}$. Clearly it is nil, since the function is constant in $\boldsymbol{\delta}$, however by the chain rule we also have that:

$$\frac{\partial \mathbf{f}}{\partial \boldsymbol{\delta}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \boldsymbol{\delta}} = \mathbf{0}. \tag{63}$$

In other words:

$$\ker\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right) \supseteq \operatorname{im}\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\delta}}\right). \tag{64}$$

Due to the minimality and connectedness assumptions we have $\operatorname{nullity}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right) = \gamma$, where γ is the dimension of a pose. Further, if there exists a $\boldsymbol{\delta}$ such that $\mathbf{x} = \boldsymbol{\delta} \oplus \boldsymbol{z}$, then it is also unique, therefore $\operatorname{rank}\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\delta}}\right) = \gamma$. We thus conclude that the columns of $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\delta}}$ form a basis that spans $\ker\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)$, therefore:

$$\ker\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right) = \operatorname{im}\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\delta}}\right). \tag{65}$$

Following the same approach for $\mathbf{g}(\mathbf{y})$ we find:

$$\ker\left(\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\right) = \operatorname{im}\left(\frac{\partial \mathbf{y}}{\partial \boldsymbol{\delta}}\right). \tag{66}$$

However, we note that $\frac{\partial \mathbf{y}}{\partial \boldsymbol{\delta}}$ is given by nothing more than the first k rows of $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\delta}}$, therefore we conclude:

$$\ker\left(\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\right) = \operatorname{im}\left(\frac{\partial \mathbf{y}}{\partial \boldsymbol{\delta}}\right) = \pi_k\left(\operatorname{im}\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\delta}}\right)\right) = \tag{67}$$

$$= \pi_k\left(\ker\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\right). \tag{68}$$

■

Proposition IV.1. *When* $\mathbf{A}$ *is invertible, the unique solution to problem* (6)-(8) *is given by:*

$$\mathbf{X}_i = \left(\left\{\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top\right\}_i\right)^{-1}. \tag{9}$$

Proof: Let us consider the gradient of the objective function (6) with respect to each block $\mathbf{X}_i$ on the diagonal of $\mathbf{X}$:

$$\frac{\partial D_{KL}}{\partial \mathbf{X}_i} = \left\{\mathbf{A}\left[\boldsymbol{\Sigma} - \left(\mathbf{A}^\top\mathbf{X}\mathbf{A}\right)^{-1}\right]\mathbf{A}^\top\right\}_i = \tag{69}$$

$$= \left\{\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top - \mathbf{A}\mathbf{A}^{-1}\mathbf{X}^{-1}\mathbf{A}^{-\top}\mathbf{A}^\top\right\}_i = \tag{70}$$

$$= \left\{\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top - \mathbf{X}^{-1}\right\}_i \tag{71}$$

Let us now forgo constraint (8); (6) is convex, therefore a necessary and sufficient condition for optimality is that the gradient be equal to $\mathbf{0}$. We thus find:

$$\mathbf{X}_i = \left(\left\{\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top\right\}_i\right)^{-1}. \tag{72}$$

However, since $\boldsymbol{\Sigma}$ is positive definite, so is $\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top$. Furthermore, since all principal minors of positive semisemidefinite matrices are positive semidefinite, we have $\mathbf{X}_i \succeq \mathbf{0}$ and therefore constraint (8) is satisfied. ■

Proposition IV.2. *When* $\mathbf{A}$ *is of full column rank and* $\mathcal{X}$ *is the set of fully dense matrices, one of the solutions to problem* (6)-(8) *is given by:*

$$\mathbf{X} = \mathbf{A}^{\mp}\boldsymbol{\Omega}\mathbf{A}^{+}. \tag{10}$$

Furthermore, (10) *yields equality between* $\mathbf{A}^\top\mathbf{X}\mathbf{A}$ *and* $\boldsymbol{\Omega}$.

Proof: Assume by ansatz that the solution to problem (6)-(8) is indeed (10). By the same reasoning of the proof of Proposition IV.1, we find that constraint (8) is satisfied. Furthermore, since $\mathbf{A}$ is of full column rank we can express its pseudoinverse as:

$$\mathbf{A}^{+} = \left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top. \tag{73}$$

We thus have:

$$\mathbf{A}^\top\mathbf{X}\mathbf{A} = \mathbf{A}^\top\mathbf{A}^{\mp}\boldsymbol{\Omega}\mathbf{A}^{+}\mathbf{A} = \tag{74}$$

$$= \mathbf{A}^\top\mathbf{A}\left(\mathbf{A}^\top\mathbf{A}\right)^{-\top}\boldsymbol{\Omega}\left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{A} = \boldsymbol{\Omega} \tag{75}$$

Since equality holds, the Kullback-Leibler distance must be 0, and therefore at a minimum. ■

Proposition VI.1. *If* $\mathbf{A}$ *is of full row rank,* $\boldsymbol{\Omega} \succeq \mathbf{0}$*, and* $\ker\mathbf{A} = \ker\boldsymbol{\Omega}$*, then:*

$$\mathbf{A}^\top\left(\mathbf{A}\boldsymbol{\Omega}^{+}\mathbf{A}^\top\right)^{-1}\mathbf{A} = \boldsymbol{\Omega}. \tag{25}$$

Proof: Let the rank-revealing eigen decomposition of $\boldsymbol{\Omega}$ be $\mathbf{U}\mathbf{D}\mathbf{U}^\top$, then $\boldsymbol{\Omega}^{+} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^\top$.

By Lemma A.4, $\ker\left(\mathbf{U}^\top\right) = \ker\boldsymbol{\Omega} = \ker\mathbf{A}$, therefore by Lemma A.6 we have that not only is $\mathbf{A}\mathbf{U}$ invertible, but also:

$$(\mathbf{A}\mathbf{U})^{-1} = \mathbf{U}^{+}\mathbf{A}^{+} = \mathbf{U}^\top\mathbf{A}^{+}. \tag{76}$$

Thus:

$$\mathbf{A}^\top\left(\mathbf{A}\boldsymbol{\Omega}^{+}\mathbf{A}^\top\right)^{-1}\mathbf{A} = \mathbf{A}^\top(\mathbf{A}\mathbf{U})^{-\top}\mathbf{D}(\mathbf{A}\mathbf{U})^{-1}\mathbf{A} = \tag{77}$$

$$= \mathbf{A}^\top\mathbf{A}^{\mp}\mathbf{U}\mathbf{D}\mathbf{U}^\top\mathbf{A}^{+}\mathbf{A}. \tag{78}$$

Since $\mathbf{A}^{+}\mathbf{A}$ is the orthogonal projector onto $(\ker\mathbf{A})^{\perp}$ and $\operatorname{im}\mathbf{U} = (\ker\mathbf{A})^{\perp}$, we have that $\mathbf{A}^{+}\mathbf{A}\mathbf{U} = \mathbf{U}$. By the definition of Moore-Penrose pseudoinverse, $\mathbf{A}^{+}\mathbf{A}$ is symmetric, therefore:

$$\mathbf{U}^\top\mathbf{A}^{+}\mathbf{A} = \mathbf{U}^\top\left(\mathbf{A}^{+}\mathbf{A}\right)^\top = \left(\mathbf{A}^{+}\mathbf{A}\mathbf{U}\right)^\top = \mathbf{U}^\top. \tag{79}$$

Substituting (79) into (78), we conclude that:

$$\mathbf{A}^\top\mathbf{A}^{\mp}\mathbf{U}\mathbf{D}\mathbf{U}^\top\mathbf{A}^{+}\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top = \boldsymbol{\Omega}. \tag{80}$$

■

Proposition VI.2. *Let* $\mathcal{M}$ *be a connected factor graph composed of only* $\mathrm{SE}(n)$ *nodes and (possibly correlated)* $\mathrm{SE}(n)$ *relative factors. Then, if the linearization point is kept fixed, the removal of any number of nodes from* $\mathcal{M}$ *can be carried out with no approximation in terms of information matrix by using fully correlated relative* $\mathrm{SE}(n)$ *measurements alone.*

Proof: Let $\mathbf{J}$ and $\boldsymbol{\Lambda}$ be respectively the stacked Jacobians and information matrices of the factors in $\mathcal{M}$. Removing a number of nodes on the current linearization point from $\mathcal{M}$ entails computing the Schur complement $\boldsymbol{\Omega}$ of the information matrix $\mathbf{J}^\top\boldsymbol{\Lambda}\mathbf{J}$. By Lemmas A.2, A.3, and A.5, we know that $\ker\boldsymbol{\Omega}$ is equal to $\ker\mathbf{J}$ with the dimensions associated to the marginalized variables removed.

We now wish to approximate the marginalized graph with a spanning tree of $\mathrm{SE}(n)$ relative measurements. Let $\mathbf{A}$ be the stacked Jacobian matrix of said tree, then under the assumption that the poses are minimally represented and the connectedness of the tree we have that $\mathbf{A}$ is of full row rank. Furthermore, by Lemma A.7, we have that $\ker\mathbf{A} = \ker\boldsymbol{\Omega}$.

Let us denote by $\mathbf{U}\mathbf{D}\mathbf{U}^\top$ the eigen decomposition of $\boldsymbol{\Omega}$, then by Lemmas A.4 and A.6, we know that the product of $\mathbf{A}\mathbf{U}$ is invertible. If we consider all the relative measurements in the spanning tree to be fully correlated, then by Proposition IV.1 we can compute the optimal information matrix $\mathbf{X}$ to assign to the measurements as:

$$\mathbf{X} = \left(\mathbf{A}\mathbf{U}\mathbf{D}^{-1}\mathbf{U}^\top\mathbf{A}^\top\right)^{-1} = \left(\mathbf{A}\boldsymbol{\Omega}^{+}\mathbf{A}^\top\right)^{-1}. \tag{81}$$

The information that the new edges will contribute is thus $\mathbf{A}^\top\mathbf{X}\mathbf{A}$, which, by Proposition VI.1 is equal to $\boldsymbol{\Omega}$. ■

Proposition VI.3. *A dense GLC factor with* $\mathrm{SE}(n)$ *reparametrization is strictly equivalent to a factor with measurement function:*

$$\mathbf{r}(\mathbf{x}) = \mathbf{r}\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \vdots\\ \mathbf{x}_m\end{bmatrix}\right) = \begin{bmatrix}\ominus\mathbf{x}_1\\ \ominus\mathbf{x}_1 \oplus \mathbf{x}_2\\ \vdots\\ \ominus\mathbf{x}_1 \oplus \mathbf{x}_m\end{bmatrix} \tag{32}$$

Proof: Let $\mathbf{\Omega}$ be the target information matrix and let $\mathbf{R}$ be the Jacobian of $\mathbf{r}(\mathbf{x})$ with respect to $\mathbf{x}$. We recall that a dense GLC factor has by definition identity covariance, and $\mathbf{G}\mathbf{r}(\mathbf{x})$ as measurement function, where $\mathbf{G}$ is a square root of $\mathbf{R}^{-\top}\mathbf{\Omega}\mathbf{R}^{-1}$ computed via eigen decomposition and $\mathbf{r}(\mathbf{x})$ is the reparametrization function for $\mathrm{SE}(n)$.

If we now denote by $\breve{\mathbf{R}}$ the Jacobian of $\mathbf{r}(\mathbf{x})$ at a different linearization point $\breve{\mathbf{x}}$, we see that a dense GLC factor contributes as information the matrix quantity:

$$\breve{\mathbf{R}}^{\top}\mathbf{R}^{-\top}\mathbf{\Omega}\mathbf{R}^{-1}\breve{\mathbf{R}}. \tag{82}$$

On the other hand, since $\mathbf{R}$ is invertible, by Proposition IV.2 we have that it is possible to achieve equality to $\mathbf{\Omega}$ by using $\mathbf{r}(\mathbf{x})$ as measurement function, and $\mathbf{R}^{-\top}\mathbf{\Omega}\mathbf{R}^{-1}$ as information matrix of the factor. If we evaluate the information matrix that this factor contributes at a different linearization point $\breve{\mathbf{x}}$ we, again, obtain (82), therefore the two factors are strictly equivalent. ∎

Proposition VI.4. *Let* $\mathcal{M}$ *be a connected factor graph composed of only* $\mathrm{SE}(n)$ *nodes and (possibly correlated)* $\mathrm{SE}(n)$ *relative factors. Then, if we remove a node from* $\mathcal{M}$ *and approximate the result via dense GLC, the resulting factor is strictly equivalent to a set of fully correlated relative* $\mathrm{SE}(n)$ *measurements with star topology.*

Proof: By Proposition VI.3, a dense GLC factor is equivalent to a factor with measurement function $\mathbf{r}(\mathbf{x})$. Let us express its Jacobian as follows:

$$\mathbf{R} = \begin{bmatrix}\mathbf{R}_{11} & \mathbf{0}\\ \mathbf{R}_{21} & \mathbf{R}_{22}\end{bmatrix}. \tag{83}$$

Where $\mathbf{R}_{11} = \frac{\partial}{\partial \mathbf{x}_1}\{\ominus\mathbf{x}_1\}$, while $\mathbf{R}_{21}$ and $\mathbf{R}_{22}$ together specify the Jacobian $\mathbf{A} = \begin{bmatrix}\mathbf{R}_{21} & \mathbf{R}_{22}\end{bmatrix}$ of a star topology of relative $\mathrm{SE}(n)$ measurements with respect to the nodes $\mathbf{x}$.

Therefore, if we denote by $\mathbf{\Lambda}$ the information matrix associated to the factor, the resulting information matrix $\mathbf{R}^{\top}\mathbf{\Lambda}\mathbf{R}$ that this factor contributes is equal to:

$$\begin{bmatrix}\mathbf{R}_{11}^{\top} & \mathbf{R}_{21}^{\top}\\ \mathbf{0} & \mathbf{R}_{22}^{\top}\end{bmatrix}\begin{bmatrix}\mathbf{\Lambda}_{11} & \mathbf{\Lambda}_{12}\\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{R}_{11} & \mathbf{0}\\ \mathbf{R}_{21} & \mathbf{R}_{22}\end{bmatrix} = \mathbf{\Psi} + \mathbf{\Xi}, \tag{84}$$

where:

$$\mathbf{\Psi} = \begin{bmatrix}\mathbf{R}_{21}^{\top}\mathbf{\Lambda}_{22}\mathbf{R}_{21} & \mathbf{R}_{21}^{\top}\mathbf{\Lambda}_{22}\mathbf{R}_{22}\\ \mathbf{R}_{22}^{\top}\mathbf{\Lambda}_{22}\mathbf{R}_{21} & \mathbf{R}_{22}^{\top}\mathbf{\Lambda}_{22}\mathbf{R}_{22}\end{bmatrix} = \mathbf{A}^{\top}\mathbf{\Lambda}_{22}\mathbf{A}, \tag{85}$$

$$\mathbf{\Xi} = \begin{bmatrix}\mathbf{R}_{11}^{\top}\mathbf{\Lambda}_{11}\mathbf{R}_{11} + \mathbf{\Theta} + \mathbf{\Theta}^{\top} & \mathbf{R}_{11}^{\top}\mathbf{\Lambda}_{12}\mathbf{R}_{22}\\ \mathbf{R}_{22}^{\top}\mathbf{\Lambda}_{21}\mathbf{R}_{11} & \mathbf{0}\end{bmatrix}, \tag{86}$$

$$\mathbf{\Theta} = \mathbf{R}_{11}\mathbf{\Lambda}_{12}\mathbf{R}_{21}. \tag{87}$$

Here, $\mathbf{\Psi}$ denotes the information contributed by the relative $\mathrm{SE}(n)$ measurements alone.

By the optimality of GLC, $\mathbf{R}^{\top}\mathbf{\Lambda}\mathbf{R}$ must be equal to the target information, which we denote by $\mathbf{\Omega}$. Moreover, by Proposition VI.2 there also exists $\bar{\mathbf{\Lambda}}$ such that $\mathbf{A}^{\top}\bar{\mathbf{\Lambda}}\mathbf{A} = \mathbf{\Omega}$. Since $\mathbf{\Xi}$ does not contribute information in the lower right corner we have:

$$\mathbf{R}_{22}^{\top}\mathbf{\Lambda}_{22}\mathbf{R}_{22} = \mathbf{R}_{22}^{\top}\bar{\mathbf{\Lambda}}\mathbf{R}_{22}. \tag{88}$$

Given that $\mathbf{R}_{22}$ is invertible, we obtain $\mathbf{\Lambda}_{22} = \bar{\mathbf{\Lambda}}$ and, as a consequence, $\mathbf{\Xi} = \mathbf{0}$. In a similar way, we can prove that $\mathbf{\Lambda}_{12} = \mathbf{0}$ and $\mathbf{\Lambda}_{11} = \mathbf{0}$: $\mathbf{\Lambda}_{12} = \mathbf{0}$ is proven by noting that $\mathbf{R}_{11}^{\top}\mathbf{\Lambda}_{12}\mathbf{R}_{22} = \mathbf{0}$ and $\mathbf{R}_{11}$ is invertible, while $\mathbf{\Lambda}_{11} = \mathbf{0}$ is proven by additionally noting that $\mathbf{R}_{11}^{\top}\mathbf{\Lambda}_{11}\mathbf{R}_{11} = \mathbf{0}$ when $\mathbf{\Lambda}_{12} = \mathbf{0}$. We finally conclude that the only informative portion of GLC is represented by the relative measurements part, concluding the proof. ∎

Proposition VI.5. *Let* $\mathcal{M}$ *be a connected factor graph composed of only* $\mathrm{SE}(n)$ *nodes and (possibly correlated)* $\mathrm{SE}(n)$ *relative factors. Then, if we remove a node from* $\mathcal{M}$ *and approximate the result via sparse GLC, each resulting factor is strictly equivalent to an* $\mathrm{SE}(n)$ *measurement.*

Proof: When approximating $\mathcal{M}$, GLC computes a set of factors from the conditional dependencies subsumed by the Chow-Liu tree of the distribution to be approximated. The contribution to the final information matrix of each factor is the marginal information $\mathbf{\Omega}_{i|j}$ between the nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ involved in the factor.

Each $\mathbf{\Omega}_{i|j}$ is obtained by marginalizing all nodes $\mathbf{x}_k$ with $k \notin \{i, j\}$, therefore, by Proposition VI.2, there exists an information matrix $\mathbf{\Lambda}$ by which a relative $\mathrm{SE}(n)$ measurement between $\mathbf{x}_i$ and $\mathbf{x}_j$ yields $\mathbf{\Omega}_{i|j}$. By the same reasoning of the proof of Proposition VI.4 we find that each binary GLC factor is strictly equivalent to a relative $\mathrm{SE}(n)$ measurement. ∎

References

O. Banerjee, L. E. Ghaoui, A. d'Aspremont, and G. Natsoulis. Convex optimization techniques for fitting sparse gaussian graphical models. In *Proceedings of the International Conference on Machine Learning*, 2006.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

N. Carlevaris-Bianco and R. M. Eustice. Generic factor-based node marginalization and edge sparsification for pose-graph SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013a.

N. Carlevaris-Bianco and R. M. Eustice. Conservative edge sparsification for graph SLAM node removal. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014.

N. Carlevaris-Bianco and R.M. Eustice. Long-term simultaneous localization and mapping with generic linear constraint node removal. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013b.

N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice. Generic node removal for factor-graph SLAM. *IEEE Transactions on Robotics*, 2014.

C. I. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25 (12):1181–1204, 2006.

J. C. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse gaussians. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence*, 2008.

E. Eade, P. Fong, and M. E. Munich. Monocular graph SLAM with complexity reduction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

R. Eustice, M. Walter, and J. Leonard. Sparse extended information filters: insights into sparsification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

J. Folkesson and H. Christensen. Graphical SLAM-a self-correcting map. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, 2004.

U. Frese. Treemap: An O(log n) algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21 (2):103–122, 2006.

U. Frese. Efficient 6-dof SLAM with treemap as a generic backend. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems*, 2007.

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

N. J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103(0):103–118, 1988.

G. Huang, M. Kaess, and J. J. Leonard. Consistent sparsification for graph optimization. In *Proceedings of the European Conference on Mobile Robots*, 2013.

S. Huang, Z. Wang, G. Dissanayake, and U. Frese. Iterated D-SLAM map joining: evaluating its performance in terms of consistency, accuracy and efficiency. *Autonomous Robots*, 27(4):409–429, 2009.

V. Ila, J. M. Porta, and J. Andrade-Cetto. Information-based compact pose SLAM. *IEEE Transactions on Robotics*, 26 (1):78–93, 2010.

H. Johannsson, M. Kaess, M.F. Fallon, and J.J. Leonard. Temporally scalable visual SLAM using a reduced pose graph. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast incremental smoothing and mapping with efficient data association. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007.

K. Konolige and J. Bowman. Towards lifelong visual maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

H. Kretzschmar and C. Stachniss. Information-theoretic compression of pose graphs for laser-based SLAM. *International Journal of Robotics Research*, 31(11):1219–1230, 2012.

H. Kretzschmar, C. Stachniss, and G. Grisetti. Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011.

M. Mazuran, G. D. Tipaldi, L. Spinello, and W. Burgard. Nonlinear graph sparsification for SLAM. In *Proceedings of Robotics: Science and Systems*, 2014.

J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

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

M. A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2003.

M. Schmidt, E. van den Berg, M. P. Friedlander, and K. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009.

S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, 2004.

L. Vandenberghe, S. Boyd, and S. P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.

J. Vial, H. Durrant-Whyte, and T. Bailey. Conservative sparsification for efficient and consistent approximate estimation.

In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.