HIERARCHY-GBP: ACCELERATING FACTOR GRAPH INFERENCE VIA ABSTRACTION AND RECOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Gaussian Belief Propagation (GBP) is a distributed inference algorithm that passes messages in graphical models, making it attractive for scalable spatial intelligence. However, we find GBP most effective locally: it rapidly smooths message errors that vary sharply between neighbor variables, but corrects global errors across distant graph regions incrementally through long-range message propagations. We propose *Hierarchy-GBP (H-GBP)*, an iterative, two-stage framework that accelerates GBP by first solving these global errors with a coarse graph approximation (abstraction) and projecting the results back to the original graph (recovery), then refining the remaining local errors with GBP. We prove H-GBP convergence to the optimum by deriving the combined matrix operator of our abstraction and recovery steps and analyzing its spectral radius. Experiments on linear sparse graphs show that H-GBP converges fundamentally faster than standard GBP. Moreover, we validate H-GBP on two important spatial problems: Pose Graph Optimization (PGO) and Bundle Adjustment (BA). H-GBP markedly accelerates large-scale PGO and achieves state-of-the-art runtime across all tested BA scales.

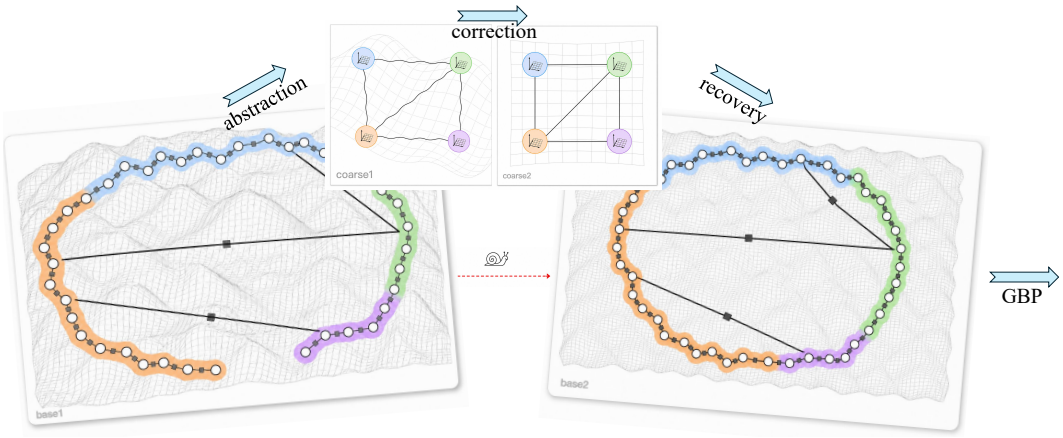


Figure 1: Illustration of Hierarchy-GBP: A fine pose graph is partitioned along odometry order into local groups, where each group extracts a low-rank subspace for coarse abstraction. The hierarchical cycle first removes coherent coarse error and then lets GBP damp the remaining local error.

1 INTRODUCTION

Gaussian Belief Propagation (GBP) is an efficient distributed graphical inference algorithm (Pearl, 2022; Yates et al., 2026) that has been successfully applied in error-correcting decoding (Frey & MacKay, 1997), multiuser detection (Bickson et al., 2008), and solving large-scale linear systems (Shental et al., 2008). Inference proceeds by passing Gaussian distributions between nodes in a probabilistic graphical model, iteratively updating each variable's marginal belief. This local, decoupled, and scalable computational model makes GBP a highly promising inference and reasoning framework for future spatial intelligence systems (Davison & Ortiz, 2019).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

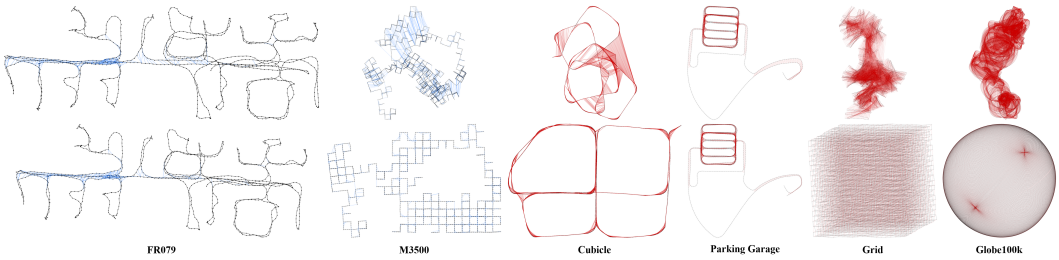


Figure 2: Pose graph's initial odometry trajectories versus trajectories after H-GBP optimization.



Figure 3: Bundle adjustment reconstructions with H-GBP optimization.

GBP has the desirable property that, if it converges on a Gaussian graph, it converges to the global optimal mean (Weiss & Freeman, 1999). However, its convergence rate is compromised by graph structures like local loops or long graph diameters (Malioutov et al., 2005; 2006; Du et al., 2018), and GBP can often require many iterations (El-Kurdi et al., 2012; Liu et al., 2012). Algebraically, GBP behaves as a smoother: it quickly damps local (high-frequency) errors that vary sharply between neighboring nodes, while global (low-frequency) errors across distant graph regions require many iterations to reduce. We observe the local-versus-global error separation in many spatial inference problems, including SLAM, 3D computer vision, and sensor networks.

This motivates a two-level view of GBP (see Fig.1). The *fine space* is the original variable space where GBP performs local message passing, while the *coarse space* is a low-dimensional graph approximation suitable for removing low-frequency errors. We solidify this idea with a localized *low-rank abstraction*. Variables are grouped locally and abstracted via their most prominent eigenvalues, forming a coarse problem that captures low-frequency errors (Briggs et al., 2000; Nataf et al., 2011; Spillane et al., 2014).

Our algorithm *Hierarchy-GBP (H-GBP)* alternates coarse residual correction with fine-level synchronous GBP. For linear Gaussian factor graphs, we derive an operator describing a 'V-cycle' of alternating coarse and fine steps, which lets us compute the spectral radius and predict a large improvement in convergence rate compared to standard GBP on the fine graph. Experiments on sparse spatial factor graphs (Dellaert & Kaess, 2017; Davison, 2018; Papadimitriou, 2025; Jiao et al., 2026) verify this predicted acceleration in linear problems. We also show that the abstraction-and-recovery principle extends to practically important, non-linear Pose Graph Optimization (PGO) (see Fig.2) and Bundle Adjustment (BA) (see Fig.3).

Specifically, our main contributions are:

- **H-GBP framework** We introduce a multilevel GBP framework for general sparse factor graphs. H-GBP builds a low-rank coarse space from low-eigenvalue modes, and alternates coarse corrections with GBP updates on the original fine graph.
- **Empirical validation** H-GBP converges faster than standard GBP on sparse linear factor graphs empirically. Extensive experiments on PGO and BA benchmarks show H-GBP's speed and accuracy benefits, and demonstrate the potential for broader spatial intelligence applications.
- **Convergence theory** For linear factor graphs, we establish a global precision convergence rate and derive a reduced cycle operator whose spectral radius is a geometric global convergence criterion. We prove the local convergence of H-GBP on manifolds.

2 Related Work

GBP on linear-Gaussian graphs has a strong theoretical foundation. Weiss & Freeman (1999) shows that when loopy GBP converges on a Gaussian graphical model, it recovers the correct posterior means, though the variances can be overconfident. Subsequent work connected GBP to walk-sum expansions and linear/quadratic solvers, clarifying that its behavior is governed by global coupling structure rather than by any single local loop in isolation (Malioutov et al., 2006; Shental et al., 2008; Moallemi & Van Roy, 2009). More recent analyses extended these perspectives to vector-valued Gaussian models, which arise in estimation problems (Du et al., 2018; Yates et al., 2026). Our H-GBP analysis takes a **matrix-operator view** of the underlying dynamics.

A second line of work improves GBP efficiency while remaining on a flat graph. Representative ideas include residual scheduling (Elidan et al., 2006), relaxation or damping (El-Kurdi et al., 2012), and structural modifications such as feedback message passing (Liu et al., 2012). These methods can reduce oscillation or avoid unnecessary updates, but they do so by reordering or partially restructuring message passing on the original graph. They do not build a coarse representation to divide and counter long-range, low-frequency error components.

H-GBP is inspired by domain-decomposition methods, where local smoothers remove high-frequency errors and coarse spaces correct slowly varying components (Briggs et al., 2000). The closest prior work combines GBP with multigrid for PDE/FEM systems (El-Kurdi et al., 2014; 2015). In contrast, H-GBP constructs the hierarchy directly on sparse factor graphs and derives the coupled coarse/fine message dynamics. To our knowledge, it is the first multilevel GBP framework formulated and analyzed for **general sparse factor graphs** beyond PDE/FEM discretizations.

Our work is also related to the factor-graph optimization literature in robotics (Tazaki, 2022). Factor graphs have become a standard abstraction for large-scale estimation and SLAM, while sparse nonlinear solvers such as g2o and Ceres remain the dominant practical baselines (Dellaert & Kaess, 2017; Kümmerle et al., 2011; Agarwal et al., 2023). Recently, there has been renewed interest in GBP as a distributed inference engine for spatial estimation, including multi-robot localization, continuous-time SLAM, and bundle adjustment on graph processors (Murai et al., 2024; Hug et al., 2024; Ortiz et al., 2020). H-GBP sits between these threads: it preserves the locality and parallelism of GBP with an abstract-then-recover acceleration mechanism.

3 Preliminaries

3.1 Factor Graphs

A factor graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ is a bipartite graph with variable nodes $\mathcal{V}$, factor nodes $\mathcal{F}$ and edges $\mathcal{E}$ that represent the factorisation of a joint probability distribution.

$$p(x) \propto \prod_{f \in \mathcal{F}} \psi_f(x_{\mathcal{N}(f)}) \tag{1}$$

where $x = (x_1, ..., x_n)^T$ is the graph's state vector and $x_{\mathcal{N}(f)}$ is the subset of variables connected to factor f.

Variables are associated with the marginal posterior distribution (so-called belief) of their state x_i, and each factor $f \in \mathcal{F}$ defines a joint Gaussian potential ψ_f over the states of its neighboring variables. Note that a factor's dimensionality depends on the number and dimension of its neighbours. In our work, all Gaussian potentials are defined in their canonical form, using the precision matrix Λ_f and information vector η_f:

$$\psi_f(x_{\mathcal{N}(f)}) \propto \exp\left\{-\frac{1}{2} x_{\mathcal{N}(f)}^\top \Lambda_f x_{\mathcal{N}(f)} + \eta_f^\top x_{\mathcal{N}(f)}\right\}$$

Collecting these parameters from all factors then yields the global linear system $Ax = b$, where A is the global precision matrix $\mathbf{\Lambda}$, b is the global information vector $\boldsymbol{\eta}$, and x is the global mean vector. The factor graph's unique Maximum A Posteriori (MAP) estimate is then given by $x_\star = A^{-1}b$.

3.2 Gaussian Belief Propagation

We provide here a brief overview of Gaussian Belief Propagation (GBP), and direct interested readers to Ortiz et al. (2021) for further detail.

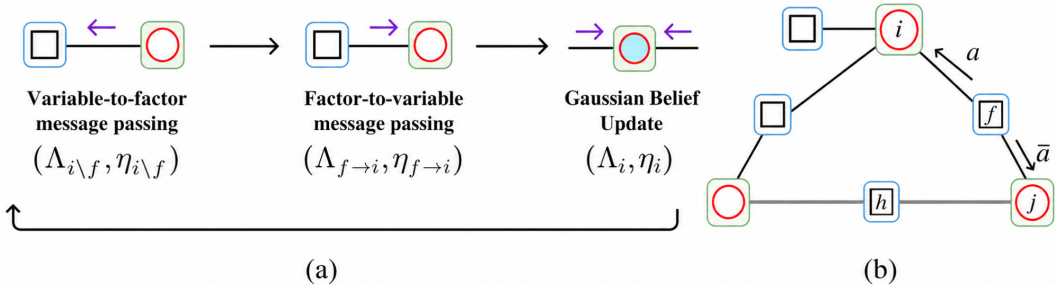


Figure 4: (a) Illustration of one synchronous iteration of Gaussian belief propagation. (b) Local factor-variable indexing illustration for the formulation of Eqs. (9) of message evolvement.

Inference for GBP takes place through iterative message passing between neighboring nodes, where each message represents a marginal probability distribution over the state space of a single variable node. Variable beliefs $(\widehat{\Lambda}_i, \widehat{\eta}_i)$ are then updated based on their most recently received messages (Fig. 4 (a)). We detail GBP's 3 update equations below, showing only factors with degree 2 or less, which are applied iteratively for inference. The Schur complement handles higher-degree factor nodes in the same way, so all steps extend exactly.

Variable-to-factor message update: Messages from variables to factors are the variable's belief with the most recent message from the receiving factor subtracted.

$$\Lambda_{i\setminus f} = \widehat{\Lambda}_i - \Lambda_{f\to i}, \qquad \eta_{i\setminus f} = \widehat{\eta}_i - \eta_{f\to i}. \tag{2}$$

Factor-to-variable message update: Messages from factors to variables sum the messages from all other neighboring variables and marginalise over them to calculate the factor's belief in terms of the receiving variable. To do this, we block partition the factor's joint precision matrix over its neighboring variables i and j: $\Lambda_f = \begin{bmatrix} \Lambda^f_{ii} & \Lambda^f_{ij} \\ \Lambda^f_{ji} & \Lambda^f_{jj} \end{bmatrix}$. From here, we can calculate updated parameters for our factor-to-variable message, Λ^+ & η^+:

$$\Lambda^+_{f\to i} = \Lambda^f_{ii} - \Lambda^f_{ij}(\Lambda^f_{jj} + \Lambda_{j\setminus f})^{-1}\Lambda^f_{ji}, \qquad \eta^+_{f\to i} = \eta^f_i - \Lambda^f_{ij}(\Lambda^f_{jj} + \Lambda_{j\setminus f})^{-1}\left(\eta^f_j + \eta_{j\setminus f}\right) \tag{3}$$

Belief update: Given an incoming message on each edge to variable i, where $i \in N(f)$, the belief in the variable is obtained by summing all incoming messages.

$$\widehat{\Lambda}_i = \sum_{f\in N(i)} \Lambda_{f\to i}, \qquad \widehat{\eta}_i = \sum_{f\in N(i)} \eta_{f\to i}, \tag{4}$$

In this work, we only consider synchronous GBP, in which one iteration consists of all variables in a graph updating their messages, all factors in the graph updating their messages, followed by all variables updating their belief.

4 Hierarchy-GBP: Algorithm and Matrix Operators

Algorithm 1 summarizes H-GBP as an iterative linear solver for the Gaussian system $Ax = b$. The prolongation matrix P is constructed once and kept fixed. Each cycle runs $\boldsymbol{k}$ synchronous GBP sweeps, solves the coarse residual problem $P^\top AP\,\delta z = P^\top(b - Ax)$, and injects $P\delta z$ using the message-preserving rule below (6). The resulting two-level iteration combines global coarse correction with local message passing.

To better explain and analyze linear H-GBP, we derive the matrix operator for each step and combine them into the operator of **one full cycle**. The steps include:

1. Abstraction & Recovery of a coarse graph approximation
2. The first GBP iteration after Recovery
3. The remaining GBP iterations

Note that separating steps 2 & 3 is necessary because we only project variable beliefs, not graph messages, to the coarse graph approximation. Once this approximation is solved and the residuals recovered, the variable beliefs are updated but inconsistent with the non-updated graph messages. We show that this inconsistency occurs only for one GBP iteration, so we compute the resulting adjusted operator that represents the first iteration after abstraction and recovery.

Let $x_\star = A^{-1}b$. We define the mean vector residual $v := x - x_\star$. Concatenating all factor-to-variable information vectors yields a big information vector η. Let $\eta^\star$ be a fixed point of the H-GBP system. We define the information residual $m := \eta - \eta^\star$. Note that $\eta^\star$ exists by construction (Appendix B.2).

4.1 Operator 1: Abstraction & Recovery of a Coarse Graph Approximation

We derive the matrix operator for the abstraction and recovery step of H-GBP. To start, we group the variable nodes together, forming G "supernodes". For each supernode group g, we combine the precision matrices from factors wholly contained in the group into a single matrix H_g, where

$$(H_g)_{i,j} = \sum_{N(f)\subset g} (\Lambda_f)_{i,j} \tag{5}$$

To reduce supernode dimensionality, for each group g we perform eigenvalue decomposition on $H_g = U\Sigma U^\top$. We then approximate H_g with its $\boldsymbol{r}$ smallest eigenvalues and corresponding eigenvectors U_r.

If we concatenate all groups' U_r matrices blockwise into a graph-wide matrix P, we get a transformation from the original graph's space to an approximated, coarse space. This global prolongation matrix is assembled blockwise as

$$P = \begin{bmatrix} U_{r,1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & U_{r,G} \end{bmatrix}$$

where $U_{r,g}$ represents U_r for group g and $\mathbf{0}$ is the zero matrix.

The coarse correction solves $(P^\top AP)\delta z = P^\top(b - Ax)$. To inject this correction into the variable beliefs, we first add increment $P\delta z$ to x. So, $x^+ = x + P\delta z$. And then, we update each variable's information vector by this new mean

$$\widehat{\eta}_i = \widehat{\Lambda}_i x_i^+. \tag{6}$$

Property 4.1. *The mean error v^+ after abstraction and recovery is*

$$v^+ = Cv, \quad C = I - P(P^\top AP)^{-1}P^\top A \tag{7}$$

4.2 Operator 2: The First GBP Iteration After Recovery

To derive the operator for Synchronous GBP, we stack all information vector messages from factor-to-variable, which allows Eq. (3) to be written in matrix form that represents the entire graph. After the graph's precision matrices have converged, we can substitute Section 3.2's update equations ((4) into (2) into (3)) and study the properties of the resulting matrix.

However, once the original graph's variable beliefs are updated from Eq. (6), they are inconsistent with the graph's messages, which did not change under abstraction and recovery.

As a result, the variable belief's information vector will not equal the sum of incoming information vectors. One synchronous GBP iteration will repair this, so we calculate the operator that represents the first iteration after abstraction and recovery in this section.

Property 4.2. *After precision matrices have converged, we consider the errors in the variables' mean vector $v = x - x_\star$ and information vector $m = \eta - \eta^\star$. The first GBP iteration with inconsistent*

beliefs & messages takes the form $m^+ = Jv + \mathcal{U}m$, where for factor f sending to variable i with other neighboring variable j (as in Fig 4b), these operators are given by:

$$(Jv)_{f\to i} = -\Lambda^f_{ij}(\Lambda^f_{jj} + \widehat{\Lambda}^\star_j - \Lambda^\star_{f\to j})^{-1}\widehat{\Lambda}^\star_j v_j, \qquad (\mathcal{U}m)_{f\to i} = \Lambda^f_{ij}(\Lambda^f_{jj} + \widehat{\Lambda}^\star_j - \Lambda^\star_{f\to j})^{-1} m_{f\to j}. \tag{8}$$

J and $\mathcal{U}$ are therefore our operators for the first iteration of GBP after abstraction and recovery.

4.3 Operator 3: The Remaining Synchronous GBP iterations

We next derive the matrix operator for synchronous GBP iterations once Section 4.2's inconsistency has been resolved. As messages and variable beliefs are now consistent, we can substitute all update equations in Section 3.2 without trouble.

Property 4.3. *Let $\eta = \begin{bmatrix}\eta^\top_{f\to i} & \eta^\top_{f\to j} & \cdots\end{bmatrix}^\top$ denote the stacked column vector of all factor-to-variable information vector messages in graph $\mathcal{G}$, letting us write Eq.* (3) *in matrix form. When precisions have converged, we substitute Section 3.2's equations (*(2) *into* (3) *into* (4)*) and show that the information vector's message update under one synchronous GBP iteration is of the affine form $\eta^+ = Q\eta + s$, where the block entries of Q are given by:*

$$Q_{(f\to i),(h\to j)} = -\Lambda^f_{ij}(\Lambda^f_{jj} + \widehat{\Lambda}^\star_j - \Lambda^\star_{f\to j})^{-1} \tag{9}$$

To demonstrate this property, a toy example is given in Appendix A.1. Here, j is the other neighboring node of f than i (see Fig. 4 (b)). $Q_{(f\to i),(h\to j)}$ indexes Q for the block corresponding to messages from factor f to variable i, and factor h to variable j, and the rest of the block entries of Q are $\mathbf{0}$.

The converged value $\eta^\star$ of GBP is unique and satisfies $\eta^\star = Q\eta^\star + s$. Recall that we define the error between our current eta-message and the fixed-point eta-message as $m = \eta - \eta^\star$. Changing variables from η to m removes the s term, and we arrive at our matrix operator, Q, for one iteration of GBP:

$$m^+ = Qm. \tag{10}$$

The message error after k-1 synchronous GBP iterations is then given by $m_{k+1} = Q^{k+1}m_0$, where m_0 is the initial mean error values.

Finally, H-GBP is iterative, and needs accurate variable means for the next abstraction and recovery. We therefore define a readout operator $\mathcal{R}$ that updates all errors in the variable means v using the variables' information vector error values m.

$$v_i = (\mathcal{R}m)_i = (\widehat{\Lambda}^\star_i)^{-1} \sum_{f\in N(i)} m_{f\to i} \tag{11}$$

Combining all operators yields a single matrix-operator description of one full H-GBP cycle.

Property 4.4. *From operators C, $Jv + \mathcal{U}m$, the remaining $k-1$ iterations of operator Q and $\mathcal{R}$, we get*

$$\begin{bmatrix} v^+ \\ m^+ \end{bmatrix} = T_{\mathrm{aug},k}\begin{bmatrix} v \\ m \end{bmatrix}, \qquad T_{\mathrm{aug},k} = \begin{bmatrix} C\mathcal{R}Q^{k-1}J & C\mathcal{R}Q^{k-1}\mathcal{U} \\ Q^{k-1}J & Q^{k-1}\mathcal{U} \end{bmatrix}. \tag{12}$$

4.4 Extension to manifolds

H-GBP naturally extends to nonlinear factor graph optimization by operating on linearizations in the tangent space. We use the standard right-perturbation convention from Lie theory (Solà et al., 2018) and factor-graph optimization (Dellaert & Kaess, 2017): at the current estimate $\bar{X}$, each residual admits the first-order approximation $r_f(\bar{X}_f \oplus \xi_f) \approx r_f(\bar{X}_f) + J_f\xi_f$. This yields the linear Gaussian system in the tangent space that we solve using H-GBP, before retracting $\bar{X}_i \leftarrow \bar{X}_i \operatorname{Exp}(\xi_i)$. Optimization on manifolds involves repeating the above process continuously until $\bar{X}_i$ converges.

5 Empirical Results

All experiments are on an Intel 13th Gen Core i9-13900HX CPU with 64 GB RAM, without GPU acceleration. Experimental details are provided in Appendix D.

5.1 LINEAR HIERARCHY-GBP

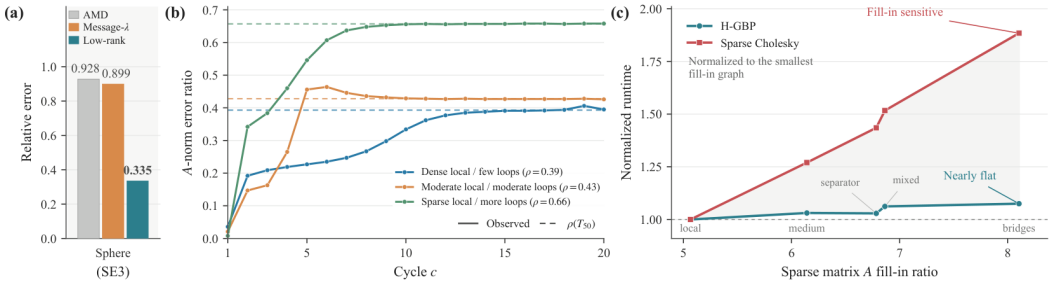


Figure 5: (a) Low-rank abstraction reduces error more effectively than alternative coarse spaces. (b) Successive-cycle error ratios approach the spectral-radius prediction as precision messages stabilize. (c) H-GBP runtime is less sensitive to sparse-Cholesky fill-in.

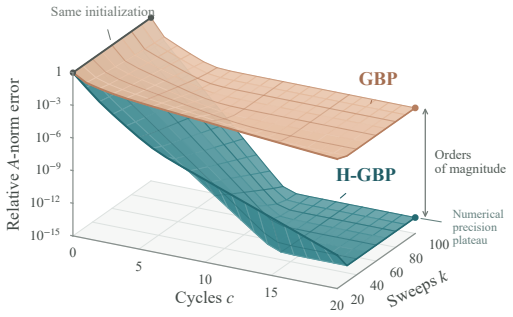


Figure 6: H-GBP versus GBP performance

We study linear H-GBP on fixed Gaussian pose-graph systems, using (x, y) variables for convergence comparisons and a fixed SE(3) linearization for the coarse-space ablation. Figure 6 shows that H-GBP removes the low-frequency error that local message passing alone struggles to eliminate. Varying both the number of cycles and the GBP iterations per cycle yields stable reduction, demonstrating H-GBP's consistent behavior across iteration budgets.

Figure 5(a) shows that local low-rank approximation yields substantially stronger error reduction than a basic multigrid-style abstraction under the same GBP budget, making our coarse-space choice important for sparse factor graphs. We find retaining four local modes per group gives a stable coarse correction with little sensitivity above this threshold. Figure 5(b) validates the correctness of the theory: as precision messages rapidly stabilize, the ratio of errors in successive cycles approaches $\rho(T_{\text{aug},50})$, matching the predicted asymptotic contraction factor.

A complementary experiment (Fig. 5(c)) evaluates solver performance across different levels of Cholesky fill-in. Sparse-Cholesky runtime grows $1.88\times$, whereas H-GBP runtime grows only $1.08\times$, revealing much weaker sensitivity to elimination fill-in.

5.2 HIERARCHY-GBP ON SE(2) AND SE(3) PGO

Table 1: Pose-graph benchmarks: runtime (s) ↓ / final cost ↓.

		1 thread		16 threads				
		g2o	**H-GBP**	g2o	Cycle	AMM	Schwarz	**H-GBP**
		Cholesky	**PGO**	PCG	PGO	PGO	PGO	**PGO**
Dataset	**Space**	**(ICRA 2011)**		**(ICRA 2011)**	**(T-RO 2021)**	**(T-RO 2024)**	**(FE Symposium 2026)**	
FR079	SE(2)	**0.02**/18.79	0.29/18.79	1.09/18.79	0.04/18.79	0.17/129.13	0.14/18.79	0.11/18.79
FRH	SE(2)	**0.04**/9.66e-5	0.35/9.66e-5	0.41/9.66e-5	0.09/9.66e-5	0.38/1.02e-4	0.13/9.66e-5	0.12/9.66e-5
M3500	SE(2)	**0.13**/68.96	0.90/68.96	1.87/68.96	0.23/68.96	0.91/70.45	1.47/68.96	0.27/68.96
ParkingGarage	SE(3)	**0.38**/0.63	7.56/0.63	19.72/0.63	0.85/0.63	1.37/0.65	49.78/0.63	1.77/0.63
Sphere	SE(3)	2.02/675.70	4.76/675.70	1.35/675.70	**1.08**/675.70	**1.08**/695.09	2.72/675.70	1.18/675.70
Cubicle	SE(3)	4.22/1,378.53	22.63/1,378.53	93.53/1,378.53	**4.17**/1,609.14	7.67/233,482.27	6.21/1,378.53	4.86/1,378.53
Grid	SE(3)	493.90/42,179.98	16.16/42,179.98	4.38/42,179.98	621.95/42,179.98	7.94/42,192.23	44.55/42,179.98	**3.90**/42,179.98
Globe10k	SE(3)	93.45/3,039.93	20.44/3,039.93	7.04/3,039.93	42.04/3,039.93	12.78/3,040.00	13.33/3,039.93	**5.34**/3,039.93
Globe100k	SE(3)	1,288.89/20,090.74	177.88/20,090.74	662.81/20,090.74	OOM/–	132.07/20,112.14	> 2000/–	**42.71**/20,090.74

1st 2nd 3rd: fastest runtimes per dataset. Red: runtime $> 20\times$ the fastest; cost $> 2\times$ the lowest; out of memory (OOM).

We evaluate H-GBP on SE(2) and SE(3) pose-graph benchmarks (Carlone et al., 2014; Olson et al., 2006; Carlone et al., 2015; Tazaki, 2022) (Fig. 2). We compare against g2o-Cholesky and g2o-

Table 2: Bundle adjustment benchmarks: runtime (s) ↓ / final sum-of-squares reprojection cost ↓.

	1 thread		16 threads				
	Ceres Sparse	**H-GBP BA**	Ceres Sparse	Ceres Iterative	RootBA QR	PowerBA	**H-GBP BA**
Dataset	**(Google Inc 2012)**		**(Google Inc 2012)**	**(Google Inc 2012)**	**(CVPR 2021)**	**(CVPR 2023)**	
Ladybug49	0.98/26,690	0.18/26,705	0.33/26,690	0.54/26,691	0.63/26,691	0.23/27,299	**0.07**/26,705
Venice89	19.22/567,243	2.68/576,781	5.91/567,243	5.03/567,055	6.22/572,780	1.91/580,416	**0.59**/576,781
Final93	7.39/291,090	1.59/291,092	1.69/291,090	4.29/291,090	6.84/291,090	1.79/291,159	**0.38**/291,092
Trafalgar257	7.21/193,577	1.31/197,378	2.38/193,577	14.41/193,671	22.72/193,398	1.11/205,427	**0.45**/197,378
Dubrovnik356	50.15/987,859	6.83/1,021,102	13.96/987,859	49.54/987,859	119.17/987,866	4.14/1,059,446	**1.63**/1,021,102
Final394	38.57/596,506	2.87/598,296	8.62/596,506	8.01/596,536	29.59/596,578	2.50/611,101	**0.83**/598,296
Ladybug1723	60.56/778,095	4.79/759,600	19.05/778,095	5.51/770,444	9.18/763,070	2.72/797,704	**1.22**/759,599
Venice1778	330.93/3,277,387	25.63/3,363,149	87.54/3,277,387	35.27/6,932,878	85.45/3,369,978	20.32/3,391,089	**5.89**/3,363,149
Final3068	586.17/3,348,655	12.40/3,240,217	142.78/3,348,655	13.17/3,452,711	102.23/3,254,156	6.15/3,520,615	**3.03**/3,240,217
Final4585	$>$ 2000/–	48.55/10,331,432	526.06/10,783,988	72.54/13,021,324	1,278.84/10,158,450	33.72/10,441,192	**11.20**/10,331,432

1st 2nd 3rd: fastest runtimes per dataset. Red: runtime $>$ 20$\times$ the fastest; cost $>$ 2$\times$ the lowest.

PCG (Kümmerle et al., 2011), AMM-PGO (Fan & Murphey, 2024), Cycle-PGO (Bai et al., 2021), and Schwarz-PGO (Tee et al., 2026). Table 1 reports runtime and final objective cost. H-GBP groups every 20 consecutive nodes in odometry order, retaining the 4 smallest eigenmodes for $\mathrm{SE}(2)$ and 12 for $\mathrm{SE}(3)$. Each relinearization is followed by 3 cycles of 100 GBP iterations for $\mathrm{SE}(2)$, or 5 cycles of 50 GBP iterations for $\mathrm{SE}(3)$.

H-GBP delivers consistent solution quality across all nine benchmarks: both its single-thread and 16-thread final costs match g2o-Cholesky at the reported precision. With 16 threads, it ranks among the three fastest methods on eight benchmarks and is the fastest on the three largest graphs, Grid, Globe10k, and Globe100k. On these graphs, it achieves approximately $\mathbf{126.6}\times$, $\mathbf{17.5}\times$, and $\mathbf{30.2}\times$ speedups over g2o-Cholesky, respectively. The corresponding speedups over the fastest competing baseline on each graph are $\mathbf{1.12}\times$, $\mathbf{1.32}\times$, and $\mathbf{3.09}\times$, showing that the large-scale advantage extends beyond comparison with a direct solver.

5.3 Hierarchy-GBP on Bundle Adjustment

We evaluate H-GBP BA on ten bundle-adjustment benchmarks against Ceres Solver (Agarwal et al., 2023) with either direct sparse linear solve or iterative PCG, RootBA-QR (Demmel et al., 2021), and PowerBA (Weber et al., 2023). Table 2 reports runtime and final sum-of-squares reprojection cost for the 16-thread comparison, with single-thread Ceres-Sparse and H-GBP BA included as references.

H-GBP operates on the camera system $S\Delta c = b$ obtained by Schur elimination of points at each linearization. For each pairwise potential, we simply fill in the corresponding off-diagonal blocks. The diagonal contributions define the unary potentials. (Note that this construction does not admit the assumptions. Therefore, the results of the BA are not intended to validate the theory, but rather to demonstrate the capabilities of H-GBP as an efficient solver.) Cameras are aggregated into connected groups of target size $\max(2, \lceil n_c/24 \rceil)$, and each group retains nine modes when abstracted.

H-GBP BA is the fastest on all ten datasets, achieving approximately $\mathbf{2.0}\times$ **to** $\mathbf{4.4}\times$ speedups over the fastest competing 16-thread baseline on each dataset, and $2.0\times$ to $4.7\times$ over PowerBA. This runtime advantage comes with consistently strong solution quality: H-GBP achieves lower final costs than PowerBA on every dataset, by up to 8.0%, and remains within 3.4% of the lowest reported cost throughout the benchmark suite. It also attains the lowest costs on Ladybug1723 and Final3068. On the largest instance, Final4585, H-GBP finishes in 11.20 s, approximately $3.0\times$ faster than PowerBA and $\mathbf{47.0}\times$ faster than 16-thread Ceres-Sparse, while remaining within 1.8% of the best reported cost. These results show that H-GBP sustains its **speed advantage** from small to large reconstructions **without** the large **cost outliers** observed for some baselines.

6 Main Theoretical Results

We give the main results here, while all proof details are included in the Appendix B. We make the same assumptions as in vector-valued standard GBP literature (Du et al., 2018).

Assumption 6.1. *The factor-to-variable precision messages* (3) *are initialized as positive semi-definite matrices:* $\Lambda_{f\to i}^{(0)} \succeq 0 \quad$ *for all* $(f \to i)$.

Assumption 6.2. *1. Each variable has a prior with positive definite precision. 2. Each binary factor precision admits* $\Lambda_f = [A_{f,i} \quad A_{f,j}]^\top W_f^{-1} [A_{f,i} \quad A_{f,j}]$, $W_f \succ 0$, *where* $A_{f,i}$ *and* $A_{f,j}$ *are full column rank.*

A useful property of H-GBP, shared with standard GBP, is that the precision-message typically converges rapidly in practice, often well before the eta-message. The following theorem proves global geometric convergence of the precision, providing a tool that links dynamics from initialization to the limiting fixed point.

Theorem 6.3. *There exist constants* $C_\Lambda < \infty$ *and* $\theta_\Lambda \in (0,1)$*, depending on the initialization, such that*

$$\|\Lambda^{(\ell)} - \Lambda^\star\| \le C_\Lambda \theta_\Lambda^\ell, \qquad \ell \ge 0,$$

for any fixed matrix norm after changing constants.

Let $\mathcal{M}$ be the stacked directed η-message error space. At the end of every cycle, the coarse correction is applied to the readout (11) of the current message state while the messages are preserved. Hence, the cycle-boundary states satisfy $v = C\mathcal{R}m$. Define the cycle-boundary subspace $\mathcal{Y}_{\mathrm{cyc}} := \{(C\mathcal{R}m, m) : m \in \mathcal{M}\}$. Define the embedding and projection $\iota_{\mathrm{cyc}}(m) := (C\mathcal{R}m, m)$, $\pi_{\mathrm{cyc}}(v, m) := m$. Then the exact reduced cycle operator on message space is

$$M_k := \pi_{\mathrm{cyc}} T_{\mathrm{aug},k} \iota_{\mathrm{cyc}} = Q^{k-1}(JC\mathcal{R} + \mathcal{U}). \tag{13}$$

Proposition 6.4 (Exact cycle-boundary reduction)**.** *The subspace* $\mathcal{Y}_{\mathrm{cyc}}$ *is* $T_{\mathrm{aug},k}$*-invariant. Moreover,*

$$T_{\mathrm{aug},k}\iota_{\mathrm{cyc}} = \iota_{\mathrm{cyc}} M_k.$$

Thus M_k is the cycle-boundary reduction of the operator $T_{\mathrm{aug},k}$. Appendix B.4 shows that the reduced dynamics satisfy $m_{t+1} = \widetilde{M}_{k,t} m_t = (M_k + E_t)m_t$, where $\sum_{t=0}^{\infty} \|E_t\| < \infty$. Thus, the full H-GBP trajectory is a summably perturbed version of the limiting linear cycle.

Theorem 6.5 (Necessary and sufficient condition)**.** *For the full H-GBP iteration, the following statements are equivalent:*

1. *for every admissible initialization,* $m_t \to 0$*;*
2. *for every compatible cycle-boundary state* $v_0 = C\mathcal{R}_0 m_0$*,* $(v_t, m_t) \to (0,0)$*;*
3. $\rho(M_k) < 1$*;* 4. $\rho(T_{\mathrm{aug},k}) < 1$.

Whenever these conditions hold, for every fixed precision initialization and every $q \in (\rho(M_k), 1)$*, there exists* $C_q < \infty$ *such that* $\|m_t\| \le C_q q^t \|m_0\|$, $\|v_t\|_A \le C_q q^t \|m_0\|$. *The corresponding fine means converge to* $x_\star = A^{-1}b$.

The same linear result also provides a direct local convergence guarantee for the nonlinear manifold extension in Section 4.4.

Theorem 6.6 (Local convergence on manifolds)**.** *Under the standard local conditions for inexact Gauss–Newton, suppose that at every linearization the tangent H-GBP system satisfies* $\rho(M_{k,t}) < 1$*. If each inner solve is run to a sufficiently small fixed relative-residual tolerance, then the nonlinear H-GBP iterates converge locally Q-linearly to a stationary point.*

7 Conclusion

We introduced Hierarchy-GBP, an abstraction-and-recovery framework to accelerate GBP on factor graphs by combining local message passing with a low-rank coarse correction. We showed H-GBP's convergence by analyzing the spectral radius of a derived matrix operator for one cycle of linear H-GBP. We found H-GBP is competitive in both runtime and solution quality on challenging pose-graph and bundle adjustment benchmarks. The results highlight H-GBP as a promising design principle for scalable spatial-intelligence systems. Future work will explore broader sparse factor-graph problems with H-GBP, fully message-passing upper levels, and hierarchy-free distributed abstractions.

AI USE STATEMENT

We used generative AI tools for language polishing. They were not used to develop research ideas, generate proofs, implement methods, or conduct experiments. The authors take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The assumptions and proofs are provided in Appendix B. Algorithmic details are given in Appendix C, and implementation details, experimental settings, and additional results are provided in Appendix D.

REFERENCES

Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres solver, 10 2023. URL `https://github.com/ceres-solver/ceres-solver`.

Fang Bai, Teresa Vidal-Calleja, and Giorgio Grisetti. Sparse pose graph optimization in cycle space. *IEEE Transactions on Robotics*, 2021. doi: 10.1109/TRO.2021.3050328.

Danny Bickson, Danny Dolev, Ori Shental, Paul H Siegel, and Jack K Wolf. Gaussian belief propagation based multiuser detection. In *2008 IEEE International Symposium on Information Theory*, pp. 1878–1882. IEEE, 2008.

William L Briggs, Van Emden Henson, and Steve F McCormick. *A multigrid tutorial*. SIAM, 2000.

Luca Carlone, Rosario Aragues, José A Castellanos, and Basilio Bona. A fast and accurate approximation for planar pose graph optimization. *The International Journal of Robotics Research*, 33(7): 965–987, 2014.

Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3d slam: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 4597–4604. IEEE, 2015.

Andrew J Davison. Futuremapping: The computational structure of spatial ai systems. *arXiv preprint arXiv:1803.11288*, 2018.

Andrew J Davison and Joseph Ortiz. Futuremapping 2: Gaussian belief propagation for spatial ai. *arXiv preprint arXiv:1910.14139*, 2019.

Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

Ron S. Dembo, Stanley C. Eisenstat, and Trond Steihaug. Inexact Newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982. doi: 10.1137/0719025.

Nikolaus Demmel, Christiane Sommer, Daniel Cremers, and Vladyslav Usenko. Square root bundle adjustment for large-scale reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Jian Du, Shaodan Ma, Yik-Chung Wu, Soummya Kar, and José MF Moura. Convergence analysis of distributed inference with vector-valued gaussian belief propagation. *Journal of Machine Learning Research*, 18(172):1–38, 2018.

Yousef El-Kurdi, Dennis Giannacopoulos, and Warren J Gross. Relaxed gaussian belief propagation. In *2012 IEEE International Symposium on Information Theory Proceedings*, pp. 2002–2006. IEEE, 2012.

Yousef El-Kurdi, Warren J Gross, and Dennis Giannacopoulos. Parallel multigrid acceleration for the finite-element gaussian belief propagation algorithm. *IEEE transactions on magnetics*, 50(2): 581–584, 2014.

Yousef El-Kurdi, Maryam Mehri Dehnavi, Warren J Gross, and Dennis Giannacopoulos. Parallel finite element technique using gaussian belief propagation. *Computer Physics Communications*, 193:38–48, 2015.

Gal Elidan, Ian McGraw, and Daphne Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 165–173, 2006.

Taosha Fan and Todd D. Murphey. Majorization minimization methods for distributed pose graph optimization. *IEEE Transactions on Robotics*, 40:22–42, 2024.

Brendan J Frey and David MacKay. A revolution: Belief propagation in graphs with cycles. *Advances in neural information processing systems*, 10, 1997.

S. Gratton, A. S. Lawless, and N. K. Nichols. Approximate Gauss–Newton methods for nonlinear least squares problems. *SIAM Journal on Optimization*, 18(1):106–132, 2007. doi: 10.1137/050624935.

Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

David Hug, Ignacio Alzugaray, and Margarita Chli. Hyperion–a fast, versatile symbolic gaussian belief propagation framework for continuous-time slam. In *European Conference on Computer Vision*, pp. 215–231. Springer, 2024.

Jianhao Jiao, Changkun Liu, Jingwen Yu, Boyi Liu, Qianyi Zhang, Yue Wang, and Dimitrios Kanoulas. Opennavmap: Structure-free topometric mapping via large-scale collaborative localization. *arXiv preprint arXiv:2601.12291*, 2026.

Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *2011 IEEE international conference on robotics and automation*, pp. 3607–3613. IEEE, 2011.

Ying Liu, Venkat Chandrasekaran, Animashree Anandkumar, and Alan S Willsky. Feedback message passing for inference in gaussian graphical models. *IEEE Transactions on Signal Processing*, 60 (8):4135–4150, 2012.

Dmitry Malioutov, Alan Willsky, and Jason Johnson. Walk-sum interpretation and analysis of gaussian belief propagation. *Advances in neural information processing systems*, 18, 2005.

Dmitry M Malioutov, Jason K Johnson, and Alan S Willsky. Walk-sums and belief propagation in gaussian graphical models. *The Journal of Machine Learning Research*, 7:2031–2064, 2006.

Ciamac C Moallemi and Benjamin Van Roy. Convergence of min-sum message passing for quadratic optimization. *IEEE Transactions on Information Theory*, 55(5):2413–2423, 2009.

Riku Murai, Ignacio Alzugaray, Paul HJ Kelly, and Andrew J Davison. Distributed simultaneous localisation and auto-calibration using gaussian belief propagation. *IEEE Robotics and Automation Letters*, 9(3):2136–2143, 2024.

Frédéric Nataf, Hua Xiang, Victorita Dolean, and Nicole Spillane. A coarse space construction based on local dirichlet-to-neumann maps. *SIAM Journal on Scientific Computing*, 33(4):1623–1642, 2011.

Edwin Olson, John Leonard, and Seth Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2262–2269. IEEE, 2006.

J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, 1970.

Joseph Ortiz, Mark Pupilli, Stefan Leutenegger, and Andrew J Davison. Bundle adjustment on a graph processor. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2413–2422. IEEE, 2020.

Joseph Ortiz, Talfan Evans, and Andrew J Davison. A visual introduction to gaussian belief propagation. *arXiv preprint arXiv:2107.02308*, 2021.

Fivos Papadimitriou. *Spatial Artificial Intelligence*. Springer, 2025.

Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Probabilistic and causal inference: the works of Judea Pearl*, pp. 129–138. 2022.

Ori Shental, Paul H Siegel, Jack K Wolf, Danny Bickson, and Danny Dolev. Gaussian belief propagation solver for systems of linear equations. In *2008 IEEE international symposium on information theory*, pp. 1863–1867. IEEE, 2008.

Joan Solà, Jérémie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

Nicole Spillane, Victorita Dolean, Patrice Hauret, Frédéric Nataf, Clemens Pechstein, and Robert Scheichl. Abstract robust coarse spaces for systems of pdes via generalized eigenproblems in the overlaps. *Numerische Mathematik*, 126(4):741–770, 2014.

Yuichi Tazaki. A spanning tree-based multi-resolution approach for pose-graph optimization. *IEEE Robotics and Automation Letters*, 7(4):10033–10040, 2022.

Yue Xiang Tee, Stephan Köhler, Oliver Rheinbach, and Sebastian Zug. Overlapping schwarz preconditioners for pose graph SLAM in robotics. Chemnitz FE Symposium, 2026. URL `https://www.chemnitz-am.de/cfem2026/talks/tee_xiang_yue.pdf`. Conference presentation abstract.

Simon Weber, Nikolaus Demmel, Tin Chon Chan, and Daniel Cremers. Power bundle adjustment for large-scale 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 281–289, 2023.

Yair Weiss and William Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Advances in neural information processing systems*, 12, 1999.

Jinchao Xu and Ludmil Zikatanov. Algebraic multigrid methods. *Acta Numerica*, 26:591–721, 2017.

Tom Yates, Yuzhou Cheng, Ignacio Alzugaray, Danyal Akarca, Pedro AM Mediano, and Andrew J Davison. Belief propagation converges to gaussian distributions in sparsely-connected factor graphs. *arXiv preprint arXiv:2601.21935*, 2026.

Supplementary Material

Contents

A Examples **14**

A.1 A three-node four-factor example 14

A.2 Standard GBP failure case 16

B Theory **17**

B.1 Convergence of precisions 18

B.2 MAP-centered message reference 20

B.3 Frozen linear H-GBP cycle 22

B.4 From the initial precision to the limiting cycle 23

B.5 Global convergence criterion and MAP target 24

B.6 H-GBP on manifolds 25

C Algorithm **27**

D Additional Details **28**

D.1 Implementation Details 28

D.2 Linear Experiment Details 28

D.3 SE2/SE3 Pose-Graph Experiment Details 30

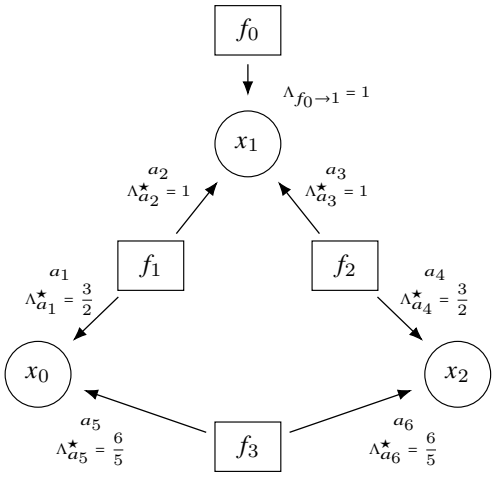
D.4 Bundle Adjustment Experiment Details 31

E Discussion **33**

A EXAMPLES

A.1 A THREE-NODE FOUR-FACTOR EXAMPLE

We give a small scalar example (see Fig. 7) illustrating how the limiting eta-transfer matrix Q is assembled from the local Schur-complement coefficients. Let $x_0, x_1, x_2 \in \mathbb{R}$. The graph has three binary factors f_{01}, f_{12}, f_{02}, and one unary factor f_0 attached to x_1.



Ordering
$a_1 = (f_1 \to x_0)$
$a_2 = (f_1 \to x_1)$
$a_3 = (f_2 \to x_1)$
$a_4 = (f_2 \to x_2)$
$a_5 = (f_3 \to x_0)$
$a_6 = (f_3 \to x_2)$

Definitions
$\eta_f = 0$ for all factors f.
$\Lambda_{f_0} = 1$.
$\Lambda_{f_1} = \Lambda_{f_2} = \Lambda_{f_3} = 6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$.
The unary message $f_0 \to x_1$ is fixed and is not stacked in η.

Figure 7: Scalar three-variable, four-factor example used to illustrate the construction of the limiting eta-message transfer matrix Q. Circles denote variables, boxes denote factors, and arrows denote directed factor-to-variable messages.

We set all information vectors to zero, since the affine term s is irrelevant for the eta recursion. The unary precision is set as

$$\Lambda_{f_0} = 1,$$

and each binary factor has precision

$$\Lambda_{f_1} = \Lambda_{f_2} = \Lambda_{f_3} = 6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Thus $\Lambda^f_{ii} = 6$, $\Lambda^f_{ij} = -6$, and $\Lambda^f_{jj} = 6$. The corresponding global information matrix is

$$A = \begin{bmatrix} 12 & -6 & -6 \\ -6 & 13 & -6 \\ -6 & -6 & 12 \end{bmatrix}.$$

Order the directed binary messages as

$$a_1 = (f_1 \to 0), \quad a_2 = (f_1 \to 1), \quad a_3 = (f_2 \to 1),$$
$$a_4 = (f_2 \to 2), \quad a_5 = (f_3 \to 0), \quad a_6 = (f_3 \to 2).$$

The unary factor sends a fixed message to x_1 and is not included in the stacked binary eta-message vector. Starting from zero binary factor-to-variable precision messages, i.e., $\Lambda^{(0)}_{a_\ell} = 0$ for all $\ell = 1, \ldots, 6$, the synchronous precision update converges to a fixed point. The following table lists

the iterates.

t	$\Lambda_{a_1}^{(t)}$	$\Lambda_{a_2}^{(t)}$	$\Lambda_{a_3}^{(t)}$	$\Lambda_{a_4}^{(t)}$	$\Lambda_{a_5}^{(t)}$	$\Lambda_{a_6}^{(t)}$
0	0.000	0.000	0.000	0.000	0.000	0.000
1	0.857	0.000	0.000	0.857	0.000	0.000
2	0.857	0.000	0.000	0.857	0.750	0.750
3	0.857	0.667	0.667	0.857	0.750	0.750
4	1.304	0.667	0.667	1.304	0.750	0.750
5	1.304	0.667	0.667	1.304	1.071	1.071
6	1.304	0.909	0.909	1.304	1.071	1.071
7	1.448	0.909	0.909	1.448	1.071	1.071
8	1.448	0.909	0.909	1.448	1.167	1.167
9	1.448	0.977	0.977	1.448	1.167	1.167
10	1.487	0.977	0.977	1.487	1.167	1.167
⋮	⋮	⋮	⋮	⋮	⋮	⋮
19	1.500	0.999	0.999	1.500	1.199	1.199
20	1.500	1.000	1.000	1.500	1.200	1.200
21	1.500	1.000	1.000	1.500	1.200	1.200

The exact fixed point for precision is therefore

$$\Lambda_{a_1}^\star = \frac{3}{2}, \quad \Lambda_{a_2}^\star = 1, \quad \Lambda_{a_3}^\star = 1, \quad \Lambda_{a_4}^\star = \frac{3}{2}, \quad \Lambda_{a_5}^\star = \frac{6}{5}, \quad \Lambda_{a_6}^\star = \frac{6}{5}.$$

Hence the limiting variable belief precisions are

$$\widehat{\Lambda}_0^\star = \frac{27}{10}, \qquad \widehat{\Lambda}_1^\star = 3, \qquad \widehat{\Lambda}_2^\star = \frac{27}{10}.$$

For a directed message $a = (f \to i)$, let $j(a)$ be the other endpoint of f, and let $\bar{a} = (f \to j(a))$. Then

$$D_a^\star = \Lambda_{j(a)j(a)}^f + \widehat{\Lambda}_{j(a)}^\star - \Lambda_{\bar{a}}^\star, \qquad B_a^\star = \Lambda_{ij(a)}^f (D_a^\star)^{-1}.$$

For example,

$$D_{a_1}^\star = 6 + \widehat{\Lambda}_1^\star - \Lambda_{a_2}^\star = 6 + 3 - 1 = 8, \qquad B_{a_1}^\star = (-6)8^{-1} = -\frac{3}{4}.$$

The six Schur coefficients are

a	$D_a^\star$	$B_a^\star$	$-B_a^\star$
a_1	8	$-\frac{3}{4}$	$\frac{3}{4}$
a_2	$\frac{36}{5}$	$-\frac{5}{6}$	$\frac{5}{6}$
a_3	$\frac{36}{5}$	$-\frac{5}{6}$	$\frac{5}{6}$
a_4	8	$-\frac{3}{4}$	$\frac{3}{4}$
a_5	$\frac{15}{2}$	$-\frac{4}{5}$	$\frac{4}{5}$
a_6	$\frac{15}{2}$	$-\frac{4}{5}$	$\frac{4}{5}$

.

Using the ordering $(a_1, \ldots, a_6)$, the limiting eta-transfer matrix is

$$Q = \begin{bmatrix} 0 & 0 & \frac{3}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{5}{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{5}{6} \\ 0 & \frac{3}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{4}{5} & 0 & 0 \\ \frac{4}{5} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

For instance, the first row encodes $\eta_{a_1}^+ = (3/4)\eta_{a_3}$: when sending $f_{01} \to 0$, the factor looks through node 1, and the only binary incoming message at node 1 excluding f_1 is $a_3 = (f_2 \to 1)$. The unary factor affects the denominator $D_{a_1}^\star$ and the affine offset, but not the linear message-to-message entry of Q.

Finally, Q consists of two directed three-cycles, each with product

$$\frac{3}{4} \cdot \frac{5}{6} \cdot \frac{4}{5} = \frac{1}{2}.$$

Therefore

$$\rho(Q) = \left(\frac{1}{2}\right)^{1/3} \approx 0.7937.$$

A.2 STANDARD GBP FAILURE CASE

In Fig. 8, we construct an example with four variables (each two-dimensional) and multiple factors. Numerically, the dynamical index $\rho(Q)$ of the standard GBP is greater than 1. Hierarchy-GBP uses the low-rank operator, group {0,1} and {2,3} as two groups. By choosing coarse rank $r = 3$ per group and performing $k = 8$ GBP iterations per cycle, the Hierarchy-GBP index $\rho(M_8)$ is less than 1.

$$\rho(Q) \approx 1.0019966, \qquad \rho(M_8) \approx 0.9205045.$$

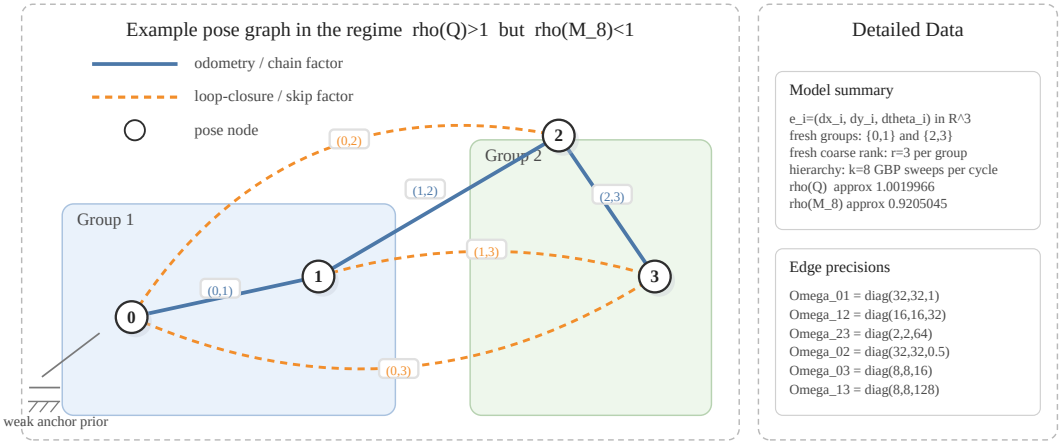


Figure 8: A four-node pose graph example, for which standard GBP does not converge, while H-GBP with $k = 8$ sweeps per cycle will converge.

B THEORY

The purpose of this appendix is to prove the two convergence claims in the main paper: first, for a fixed H-GBP cycle length k, the full hierarchy converges from every admissible initialization if and only if the reduced cycle operator satisfies $\rho(M_k) < 1$ (equivalently $\rho(T_{\mathrm{aug},k}) < 1$). Second, whenever this criterion holds, the fine-level mean converges to the exact MAP solution of the original linear Gaussian factor graph.

For readability, formulas are written for unary and binary factors. Higher-order factors use the same Schur-complement formulas (Ortiz et al., 2021) after replacing the single "other endpoint" by the block of all non-target variables.

We make the same assumptions (Assumption B.1, B.2) as in vector-valued standard GBP (Du et al., 2018) and state their results as facts here, before delving into our results of H-GBP in the following subsections.

Let $N_1(i)$ and $N_2(i)$ denote the unary and binary factors incident on variable i. For a binary factor $f = \{i, j\}$, write

$$\Lambda_f = \begin{bmatrix} \Lambda_{ii}^f & \Lambda_{ij}^f \\ \Lambda_{ji}^f & \Lambda_{jj}^f \end{bmatrix}.$$

For a directed binary message $a = (f \to i)$, with the other endpoint $j = j(a)$, define the cavity precision

$$\mathcal{C}_a(\Lambda) := \Lambda_j^{\mathrm{self}} + S_a(\Lambda), \qquad \Lambda_j^{\mathrm{self}} := \sum_{u \in N_1(j)} \Lambda_{u\to j}, \qquad S_a(\Lambda) := \sum_{h \in N_2(j)\setminus\{f\}} \Lambda_{h\to j}.$$

Assumption B.1. *The initial factor-to-variable precision messages satisfy $\Lambda_{f\to i}^{(0)} \succeq 0$ for every directed factor-to-variable edge.*

Assumption B.2. *The following regularity conditions hold.*

(i) Each variable has positive self-precision: $\Lambda_i^{\mathrm{self}} \succ 0$ for all $i \in \mathcal{V}$.

(ii) Each binary factor admits the measurement-form factorization

$$\Lambda_f = [A_{f,i} \quad A_{f,j}]^\top W_f^{-1} [A_{f,i} \quad A_{f,j}], \qquad W_f \succ 0,$$

where $A_{f,i}$ and $A_{f,j}$ are full column rank.

Remark. *Assumption B.2 is consistent with standard Gaussian-BP implementations. In practice, the Schur-complement block in each factor-to-variable update is always stabilized by adding a small positive diagonal term, for example*

$$D_{f\to i} \mapsto D_{f\to i} + \varepsilon I, \qquad \varepsilon \approx 10^{-10}.$$

For the precision recursion, this is equivalent to adding an arbitrarily weak unary regularizer with precision εI to each variable. After including this regularizer in $N_1(i)$, we have $\Lambda_i^{\mathrm{self}} \succeq \varepsilon I \succ 0$. Thus the condition $\Lambda_i^{\mathrm{self}} \succ 0$ should be understood as a standard technical regularity condition ensuring well-defined Schur complements, rather than as a restrictive modeling requirement.

Let S_+ and S_{++} denote the blockwise positive semidefinite and positive definite cones. The precision update map is $F : S_+ \to S_{++}$ such that

$$F_a(\Lambda) = \Lambda_{ii}^f - \Lambda_{ij}^f \big(\Lambda_{jj}^f + \mathcal{C}_a(\Lambda)\big)^{-1} \Lambda_{ji}^f. \tag{14}$$

Using the measurement factorization and the Woodbury identity, we get this fact from Du et al. (2018)

$$F_a(\Lambda) = A_{f,i}^\top \left[W_f + A_{f,j} \mathcal{C}_a(\Lambda)^{-1} A_{f,j}^\top\right]^{-1} A_{f,i}. \tag{15}$$

This form makes positivity, monotonicity, and differentiability of F transparent.

Theorem B.3 (standard GBP precision convergence (Du et al., 2018))**.** *Under Assumptions B.1–B.2, the recursion $\Lambda^{(\ell+1)} = F(\Lambda^{(\ell)})$ has a unique positive-definite fixed point $\Lambda^\star$. For every positive-semidefinite initialization, $\Lambda^{(\ell)} \to \Lambda^\star$. Moreover, for every fixed neighborhood of $\Lambda^\star$, the trajectory approaches that neighborhood at an exponential rate before the first entrance.*

B.1 CONVERGENCE OF PRECISIONS

Proposition B.4 (Precision dynamics are unchanged by the hierarchy)**.** *The H-GBP coarse correction changes only the variable belief information vectors and means. It does not alter factor-to-variable precision messages. Therefore H-GBP has exactly the same precision recursion, $\Lambda^{(\ell+1)} = F(\Lambda^{(\ell)})$, as standard synchronous GBP.*

Proof. The message-preserving injection rewrites $\widehat{\eta}_i$ so that the belief mean is shifted by the coarse correction, while leaving $\widehat{\Lambda}_i$, $\Lambda_{f\to i}$, and $\eta_{f\to i}$ unchanged. The next precision update therefore sees the same precision state as ordinary GBP. □

Now, we strengthen the convergence result to a global geometric convergence rate, which will be recursively used to control the early, nonstationary precision phase of H-GBP.

Lemma B.5 (Derivative of the precision map at the fixed point)**.** *The precision map F is Fréchet differentiable on a neighborhood of $\Lambda^\star$ inside $\mathbb{S}_{++}$. For a directed message $a = (f \to i)$, let $j = j(a)$, and define*

$$\mathcal{B}_a(\Lambda) := W_f + A_{f,j}\mathcal{C}_a(\Lambda)^{-1}A_{f,j}^\top.$$

Then for every symmetric blockwise increment H,

$$DF_a(\Lambda)[H] = A_{f,i}^\top \mathcal{B}_a(\Lambda)^{-1}A_{f,j}\mathcal{C}_a(\Lambda)^{-1}\left(\sum_{h\in N_2(j)\setminus\{f\}} H_{h\to j}\right)\mathcal{C}_a(\Lambda)^{-1}A_{f,j}^\top\mathcal{B}_a(\Lambda)^{-1}A_{f,i}.$$

In particular, $H \succeq 0$ implies $DF(\Lambda^\star)[H] \succeq 0$.

Proof. Equation (15) gives $F_a(\Lambda) = A_{f,i}^\top \mathcal{B}_a(\Lambda)^{-1}A_{f,i}$.

The map $\mathcal{C}_a(\Lambda)$ is linear in the relevant components of Λ. Hence, $D\mathcal{C}_a(\Lambda)[H] = \sum_{h\in N_2(j)\setminus\{f\}} H_{h\to j}$. Since $\mathcal{C}_a(\Lambda^\star) \succ 0$, by continuity $\mathcal{C}_a(\Lambda) \succ 0$ in a sufficiently small neighborhood of $\Lambda^\star$. Thus the inverse is differentiable there.

The Fréchet derivative formula for inversion (Higham, 2002) is $D(X^{-1})[H] = -X^{-1}HX^{-1}$, so

$$D(\mathcal{C}_a^{-1})(\Lambda)[H] = -\mathcal{C}_a(\Lambda)^{-1}D\mathcal{C}_a(\Lambda)[H]\mathcal{C}_a(\Lambda)^{-1}.$$

Next, $\mathcal{B}_a(\Lambda) = W_f + A_{f,j}\mathcal{C}_a(\Lambda)^{-1}A_{f,j}^\top$. Since W_f, $A_{f,j}$, and $A_{f,j}^\top$ are fixed with respect to Λ, differentiating gives $D\mathcal{B}_a(\Lambda)[H] = A_{f,j}D(\mathcal{C}_a^{-1})(\Lambda)[H]A_{f,j}^\top$.

Applying the same inversion derivative formula to $\mathcal{B}_a(\Lambda)^{-1}$, we have

$$D(\mathcal{B}_a^{-1})(\Lambda)[H] = -\mathcal{B}_a(\Lambda)^{-1}D\mathcal{B}_a(\Lambda)[H]\mathcal{B}_a(\Lambda)^{-1}.$$

Finally, since $A_{f,i}$ is fixed with respect to Λ, $DF_a(\Lambda)[H] = A_{f,i}^\top D(\mathcal{B}_a^{-1})(\Lambda)[H]A_{f,i}$. Hence, substituting all the equations above sequentially we get

$$DF_a(\Lambda)[H] = A_{f,i}^\top \mathcal{B}_a(\Lambda)^{-1}A_{f,j}\mathcal{C}_a(\Lambda)^{-1}\left(\sum_{h\in N_2(j)\setminus\{f\}} H_{h\to j}\right)\mathcal{C}_a(\Lambda)^{-1}A_{f,j}^\top\mathcal{B}_a(\Lambda)^{-1}A_{f,i}.$$

This is the claimed derivative formula. If $H \succeq 0$, the middle summed block is positive semidefinite, so every displayed block derivative is positive semidefinite. □

Proposition B.6 (Local strict contraction near the precision fixed point)**.** *There exist a monotone norm $\|\cdot\|_\star$ on the blockwise precision space, a radius $\varepsilon_0 > 0$, and a constant $q \in (0,1)$ such that*

$$\|\Lambda - \Lambda^\star\|_\star < \varepsilon_0 \quad\Longrightarrow\quad \|F(\Lambda) - \Lambda^\star\|_\star \le q\|\Lambda - \Lambda^\star\|_\star.$$

Proof. For a blockwise symmetric perturbation $H = \{H_a\}_{a\in\mathcal{D}}$, define

$$\|H\|_\star := \max_{a\in\mathcal{D}} \left\|(\Lambda_a^\star)^{-1/2}H_a(\Lambda_a^\star)^{-1/2}\right\|_2.$$

This norm is monotone on the positive semi-definite cone: if $0 \preceq H \preceq \widetilde{H}$, then each normalized block is ordered in the Loewner sense, and the spectral norm of positive semidefinite blocks is monotone. The definition also implies the blockwise order bound

$$-\|H\|_\star \Lambda^\star \preceq H \preceq \|H\|_\star \Lambda^\star. \tag{16}$$

We show first that $DF(\Lambda^\star)$ is strictly contractive in this norm. Fix $a = (f \to i)$, let $j = j(a)$, and write

$$S_a^\star := \sum_{h \in N_2(j) \setminus \{f\}} \Lambda_{h \to j}^\star, \quad \mathcal{C}_a^\star := \Lambda_j^{\mathrm{self}} + S_a^\star, \quad \mathcal{B}_a^\star := W_f + A_{f,j}(\mathcal{C}_a^\star)^{-1} A_{f,j}^\top.$$

Substituting $H = \Lambda^\star$ into Lemma B.5 gives

$$DF_a(\Lambda^\star)[\Lambda^\star] = A_{f,i}^\top (\mathcal{B}_a^\star)^{-1} A_{f,j} (\mathcal{C}_a^\star)^{-1} S_a^\star (\mathcal{C}_a^\star)^{-1} A_{f,j}^\top (\mathcal{B}_a^\star)^{-1} A_{f,i}.$$

Because $\Lambda_j^{\mathrm{self}} \succ 0$, $0 \preceq S_a^\star \prec \mathcal{C}_a^\star$, and hence $(\mathcal{C}_a^\star)^{-1} S_a^\star (\mathcal{C}_a^\star)^{-1} \prec (\mathcal{C}_a^\star)^{-1}$.

Also $W_f \succ 0$, so $A_{f,j}(\mathcal{C}_a^\star)^{-1} A_{f,j}^\top \prec W_f + A_{f,j}(\mathcal{C}_a^\star)^{-1} A_{f,j}^\top = \mathcal{B}_a^\star$.

Combining these inequalities gives

$$DF_a(\Lambda^\star)[\Lambda^\star] \prec A_{f,i}^\top (\mathcal{B}_a^\star)^{-1} A_{f,i} = F_a(\Lambda^\star) = \Lambda_a^\star.$$

Since there are finitely many directed messages, there exists $\alpha \in (0,1)$ such that

$$DF(\Lambda^\star)[\Lambda^\star] \preceq \alpha \Lambda^\star. \tag{17}$$

For arbitrary H, let $t = \|H\|_\star$. By (16), $-t\Lambda^\star \preceq H \preceq t\Lambda^\star$. Since $DF(\Lambda^\star)$ is positive and linear, it is order-preserving. Therefore $DF(\Lambda^\star)[-t\Lambda^\star] \preceq DF(\Lambda^\star)[H] \preceq DF(\Lambda^\star)[t\Lambda^\star]$. Linearity gives

$$-tDF(\Lambda^\star)[\Lambda^\star] \preceq DF(\Lambda^\star)[H] \preceq tDF(\Lambda^\star)[\Lambda^\star].$$

By (17), $DF(\Lambda^\star)[\Lambda^\star] \preceq \alpha \Lambda^\star$. Hence

$$-\alpha t \Lambda^\star \preceq DF(\Lambda^\star)[H] \preceq \alpha t \Lambda^\star.$$

Substituting $t = \|H\|_\star$ yields $-\alpha \|H\|_\star \Lambda^\star \preceq DF(\Lambda^\star)[H] \preceq \alpha \|H\|_\star \Lambda^\star$. By the definition of $\|\cdot\|_\star$, this implies

$$\|DF(\Lambda^\star)[H]\|_\star \le \alpha \|H\|_\star.$$

Thus $\|DF(\Lambda^\star)\|_{\star\to\star} := \sup_{H \neq 0} \frac{\|DF(\Lambda^\star)[H]\|_\star}{\|H\|_\star} \le \alpha < 1$.

Since F is C^1 near $\Lambda^\star$, choose $q \in (\alpha, 1)$ and then $\varepsilon_0 > 0$ so that the closed $\|\cdot\|_\star$-ball of radius ε_0 around $\Lambda^\star$ lies inside $\mathbb{S}_{++}$ and $\|DF(\Lambda)\|_{\star\to\star} \le q$ throughout that ball. For any Λ in the ball, the segment between $\Lambda^\star$ and Λ remains in the ball, and the finite-dimensional mean-value formula gives

$$F(\Lambda) - F(\Lambda^\star) = \int_0^1 DF\big(\Lambda^\star + s(\Lambda - \Lambda^\star)\big)[\Lambda - \Lambda^\star]\, ds.$$

Taking the $\|\cdot\|_\star$-norm proves the claimed local contraction. □

Theorem B.7 (Global geometric precision convergence)**.** *For every admissible precision initialization, there exist constants $C_\Lambda < \infty$ and $\theta_\Lambda \in (0,1)$ such that*

$$\|\Lambda^{(\ell)} - \Lambda^\star\|_\star \le C_\Lambda \theta_\Lambda^\ell, \qquad \ell \ge 0.$$

Proof. The imported exponential-entrance result B.3 implies a finite entrance into the ε_0-ball. After that entrance time, Proposition B.6 gives a geometric tail. Enlarging the prefactor to dominate the finite prefix yields the displayed all-time bound. □

B.2 MAP-CENTERED MESSAGE REFERENCE

Let $x_\star = A^{-1}b$. For any precision state Λ, define the canonical eta reference by

$$\eta^\star_{f\to i}(\Lambda) := \eta^f_i - (\Lambda_f x^\star_{N(f)})_i + \Lambda_{f\to i} x^\star_i. \tag{18}$$

Here $x^\star_{N(f)}$ is the stacked vector of MAP blocks adjacent to factor f, Λ_f is the precision matrix of factor f, and $(\Lambda_f x^\star_{N(f)})_i$ denotes the block corresponding to variable i. For unary messages, $\eta^\star_{u\to i} = \eta^u_i$. For a binary factor f with endpoints i and j,

$$\eta^\star_{f\to i}(\Lambda) = \eta^f_i - \Lambda^f_{ii} x^\star_i - \Lambda^f_{ij} x^\star_j + \Lambda_{f\to i} x^\star_i.$$

Let

$$\widehat{\Lambda}_i(\Lambda) := \sum_{f\in N(i)} \Lambda_{f\to i}, \qquad \widehat{\eta}^\star_i(\Lambda) := \sum_{f\in N(i)} \eta^\star_{f\to i}(\Lambda). \tag{19}$$

Lemma B.8 (Canonical reference has MAP readout and propagates with precision)**.** *For every precision state Λ,*

$$\widehat{\eta}^\star_i(\Lambda) = \widehat{\Lambda}_i(\Lambda) x^\star_i, \qquad \forall i.$$

Moreover, if one synchronous precision sweep maps Λ to $F(\Lambda)$, then the corresponding synchronous eta Schur update maps $\eta^\star(\Lambda)$ to $\eta^\star(F(\Lambda))$. In particular, $\eta^\star(\Lambda^\star)$ is a fixed point of the frozen eta-message equations at $\Lambda^\star$, and its belief mean is $x_\star$.

Proof. We first prove the readout identity. By the assembly of the global linear system,

$$b_i = \sum_{f\in N(i)} \eta^f_i, \qquad (Ax_\star)_i = \sum_{f\in N(i)} (\Lambda_f x^\star_{N(f)})_i, \qquad b_i - (A^\star_x)_i = 0.$$

By definition 19,

$$\widehat{\eta}^\star_i(\Lambda) - \widehat{\Lambda}_i(\Lambda) x^\star_i = \sum_{f\in N(i)} \left[\eta^\star_{f\to i}(\Lambda) - \Lambda_{f\to i} x^\star_i\right] = \sum_{f\in N(i)} \left[\eta^f_i - (\Lambda_f x^\star_{N(f)})_i\right].$$

Hence $\widehat{\eta}^\star_i(\Lambda) - \widehat{\Lambda}_i(\Lambda) x^\star_i = b_i - (Ax_\star)_i = 0$. Therefore $\widehat{\eta}^\star_i(\Lambda) = \widehat{\Lambda}_i(\Lambda) x^\star_i$.

We now prove the propagation identity. Consider a binary message $a = (f \to i)$, and let j be the other endpoint of f. Define

$$D_a(\Lambda) := \Lambda^f_{jj} + \sum_{h\in N(j)\setminus\{f\}} \Lambda_{h\to j}.$$

The corresponding precision update is

$$F_a(\Lambda) = \Lambda^f_{ii} - \Lambda^f_{ij} D_a(\Lambda)^{-1} \Lambda^f_{ji}. \tag{20}$$

The eta Schur update uses the cavity information at j,

$$r^\star_{a,j}(\Lambda) := \eta^f_j + \sum_{h\in N(j)\setminus\{f\}} \eta^\star_{h\to j}(\Lambda).$$

We claim that

$$r^\star_{a,j}(\Lambda) = D_a(\Lambda) x^\star_j + \Lambda^f_{ji} x^\star_i.$$

Indeed, by the readout identity just proved,

$$\sum_{h\in N(j)} \eta^\star_{h\to j}(\Lambda) = \left(\sum_{h\in N(j)} \Lambda_{h\to j}\right) x^\star_j.$$

Therefore

$$\begin{aligned} r^\star_{a,j}(\Lambda) &= \eta^f_j + \widehat{\eta}^\star_j(\Lambda) - \eta^\star_{f\to j}(\Lambda) \\ &= \eta^f_j + \widehat{\Lambda}_j(\Lambda) x^\star_j - \left[\eta^f_j - \Lambda^f_{jj} x^\star_j - \Lambda^f_{ji} x^\star_i + \Lambda_{f\to j} x^\star_j\right] \\ &= \left(\widehat{\Lambda}_j(\Lambda) - \Lambda_{f\to j} + \Lambda^f_{jj}\right) x^\star_j + \Lambda^f_{ji} x^\star_i \\ &= D_a(\Lambda) x^\star_j + \Lambda^f_{ji} x^\star_i. \end{aligned}$$

Substituting this into the eta Schur update gives

$$\begin{aligned}\eta^{+}_{f\to i} &= \eta^{f}_{i} - \Lambda^{f}_{ij} D_a(\Lambda)^{-1} r^{\star}_{a,j}(\Lambda) \\ &= \eta^{f}_{i} - \Lambda^{f}_{ij} D_a(\Lambda)^{-1} \left[D_a(\Lambda) x^{\star}_{j} + \Lambda^{f}_{ji} x^{\star}_{i} \right] \\ &= \eta^{f}_{i} - \Lambda^{f}_{ij} x^{\star}_{j} - \Lambda^{f}_{ij} D_a(\Lambda)^{-1} \Lambda^{f}_{ji} x^{\star}_{i}.\end{aligned}$$

Using 20, we obtain

$$\begin{aligned}\eta^{+}_{f\to i} &= \eta^{f}_{i} - \Lambda^{f}_{ij} x^{\star}_{j} - \big(\Lambda^{f}_{ii} - F_a(\Lambda)\big) x^{\star}_{i} \\ &= \eta^{f}_{i} - \Lambda^{f}_{ii} x^{\star}_{i} - \Lambda^{f}_{ij} x^{\star}_{j} + F_a(\Lambda) x^{\star}_{i}.\end{aligned}$$

This is exactly the canonical message

$$\eta^{\star}_{f\to i}(F(\Lambda)).$$

Unary eta messages are unchanged, and their canonical references are also unchanged. Hence the full synchronous eta update maps

$$\eta^{\star}(\Lambda) \quad \text{to} \quad \eta^{\star}(F(\Lambda)).$$

Finally, setting $\Lambda = \Lambda^{\star}$ and using $F(\Lambda^{\star}) = \Lambda^{\star}$, we get $\eta^{\star}(\Lambda^{\star}) \mapsto \eta^{\star}(F(\Lambda^{\star})) = \eta^{\star}(\Lambda^{\star})$. Thus $\eta^{\star}(\Lambda^{\star})$ is a fixed point of the frozen eta-message equations.

The readout identity at $\Lambda^{\star}$ gives $\widehat{\Lambda}_i(\Lambda^{\star})^{-1}\widehat{\eta}^{\star}_i(\Lambda^{\star}) = x^{\star}_i$ for every variable i, so the belief mean is $x_{\star}$. □

From now on, eta-message errors are always defined relative to the current canonical reference:

$$m := \eta - \eta^{\star}(\Lambda), \qquad v := x - x_{\star}.$$

At the limiting precision, this reduces to the usual frozen error $m = \eta - \eta^{\star}(\Lambda^{\star})$.

B.3 FROZEN LINEAR H-GBP CYCLE

Let P be the grouped prolongation matrix, and define

$$C := I - P(P^\top AP)^{-1}P^\top A. \tag{21}$$

The coarse residual correction $(P^\top AP)\delta z = P^\top(b - Ax),\ x^+ = x + P\delta z$ therefore maps the mean error by

$$v^+ = Cv, \qquad P^\top Av^+ = 0. \tag{22}$$

This is the Galerkin A-orthogonal projection onto $\mathrm{range}(P)^{\perp_A}$ (Briggs et al., 2000).

Now, let's fix the precision at $\Lambda^\star$ – analyze the system when all variances converge.

Recall in the main text, we have the following matrix operators: $(J,\mathcal{U})$ for repair sweep, Q for $k-1$ additional GBP sweeps, and $\mathcal{R}$ for the readout.

Together with the Galerkin correction C gives the limiting augmented cycle

$$\begin{bmatrix} v^+ \\ m^+ \end{bmatrix} = T_{\mathrm{aug},k} \begin{bmatrix} v \\ m \end{bmatrix}, \qquad T_{\mathrm{aug},k} = \begin{bmatrix} C\mathcal{R}Q^{k-1}J & C\mathcal{R}Q^{k-1}\mathcal{U} \\ Q^{k-1}J & Q^{k-1}\mathcal{U} \end{bmatrix}. \tag{23}$$

At a post-injection cycle boundary, messages are preserved and the mean has just been Galerkin-corrected. Hence boundary states satisfy

$$v = C\mathcal{R}m.$$

Define

$$\mathcal{Y}_{\mathrm{cyc}} := \{(C\mathcal{R}m, m) : m \in \mathcal{M}\}, \qquad \iota_{\mathrm{cyc}}(m) := (C\mathcal{R}m, m), \qquad \pi_{\mathrm{cyc}}(v, m) := m.$$

Proposition B.9 (Exact cycle-boundary reduction)**.** *The subspace $\mathcal{Y}_{\mathrm{cyc}}$ is invariant under $T_{\mathrm{aug},k}$, and the reduced cycle operator is*

$$M_k := \pi_{\mathrm{cyc}} T_{\mathrm{aug},k} \iota_{\mathrm{cyc}} = Q^{k-1}(JC\mathcal{R} + \mathcal{U}). \tag{24}$$

Moreover,

$$T_{\mathrm{aug},k}\iota_{\mathrm{cyc}} = \iota_{\mathrm{cyc}} M_k.$$

Proof. Substituting $v = C\mathcal{R}m$ into (23) gives

$$T_{\mathrm{aug},k}\iota_{\mathrm{cyc}}(m) = \big(C\mathcal{R}Q^{k-1}(JC\mathcal{R} + \mathcal{U})m, Q^{k-1}(JC\mathcal{R} + \mathcal{U})m\big) = \iota_{\mathrm{cyc}}(M_k m).$$

□

Lemma B.10 (Spectral equivalence)**.**

$$\rho(M_k) = \rho(T_{\mathrm{aug},k}).$$

Proof. The map ι_{cyc} is an isomorphism from $\mathcal{M}$ to $\mathcal{Y}_{\mathrm{cyc}}$, so M_k is similar to $T_{\mathrm{aug},k}|_{\mathcal{Y}_{\mathrm{cyc}}}$. Also, $T_{\mathrm{aug},k}$ maps the whole augmented space into $\mathcal{Y}_{\mathrm{cyc}}$; hence every nonzero eigenvector of $T_{\mathrm{aug},k}$ lies in $\mathcal{Y}_{\mathrm{cyc}}$. The two operators therefore have the same nonzero eigenvalues. □

B.4 From the initial precision to the limiting cycle

For cycle t, let $\Lambda_t^{[r]} := \Lambda^{(tk+r)}$, $r = 0, \dots, k$. The operators $J_t, \mathcal{U}_t, \mathcal{R}_t$ are evaluated at $\Lambda_t^{[0]}$, while $Q_{t,r}$ is evaluated at $\Lambda_t^{[r]}$. Because the canonical reference $\eta^\star(\Lambda)$ propagates with the precision update (Lemma B.8), the eta-error dynamics are exactly homogeneous at every precision state.

Definition B.11 (Exact cocycle and limiting operator)**.** *The exact cycle-boundary cocycle is*

$$\widetilde{M}_{k,t} := Q_{t,k-1} \cdots Q_{t,1} \big(J_t C \mathcal{R}_t + \mathcal{U}_t \big). \tag{25}$$

The limiting operator is

$$M_k = Q^{k-1}(JC\mathcal{R} + \mathcal{U}),$$

where $Q, J, \mathcal{U}, \mathcal{R}$ are evaluated at $\Lambda^\star$.

Proposition B.12 (Exact cycle-boundary dynamics)**.** *Let m_t be the directed eta-message error at the post-injection boundary of cycle t, defined relative to $\eta^\star(\Lambda_t^{[0]})$. Then*

$$v_t = C\mathcal{R}_t m_t, \qquad m_{t+1} = \widetilde{M}_{k,t} m_t.$$

Proof. Before injection, the readout error is $\mathcal{R}_t m_t$; the Galerkin step maps it to $C\mathcal{R}_t m_t$, proving the boundary relation. The first sweep gives $(J_t C\mathcal{R}_t + \mathcal{U}_t)m_t$, and the remaining $k-1$ message-consistent sweeps multiply successively by $Q_{t,1}, \dots, Q_{t,k-1}$. This is exactly (25). □

Lemma B.13 (Local Lipschitz dependence of frozen operators)**.** *There is a compact neighborhood $\mathcal{K}_0 \subset \mathbb{S}_{++}$ of $\Lambda^\star$ and a constant $L_{\mathrm{op}} < \infty$ such that, for all $\Lambda, \widetilde{\Lambda} \in \mathcal{K}_0$,*

$$\|J(\Lambda) - J(\widetilde{\Lambda})\| + \|\mathcal{U}(\Lambda) - \mathcal{U}(\widetilde{\Lambda})\| + \|\mathcal{R}(\Lambda) - \mathcal{R}(\widetilde{\Lambda})\| + \|Q(\Lambda) - Q(\widetilde{\Lambda})\| \le L_{\mathrm{op}} \|\Lambda - \widetilde{\Lambda}\|_\star.$$

The same operator families are uniformly bounded on $\mathcal{K}_0$.

Proof. Every block of these operators is a rational expression in the precision blocks, formed from sums, products, Schur denominators, and inverses. On a compact positive-definite neighborhood of $\Lambda^\star$, all denominators are uniformly nonsingular.

The maps are therefore C^1, Lipschitz, and uniformly bounded on $\mathcal{K}_0$. □

Lemma B.14 (Exponential convergence of the exact cocycle)**.** *For every admissible precision initialization and fixed k, there exist constants $C_{M,k} < \infty$ and $\theta_{M,k} \in (0,1)$ such that*

$$\|\widetilde{M}_{k,t} - M_k\| \le C_{M,k} \theta_{M,k}^t, \qquad t \ge 0.$$

Consequently, with $E_t := \widetilde{M}_{k,t} - M_k$,

$$\sum_{t=0}^{\infty} \|E_t\| < \infty.$$

Proof. By Corollary B.7, $\|\Lambda^{(\ell)} - \Lambda^\star\|_\star \le C_\Lambda \theta_\Lambda^\ell$. After a finite prefix, all within-cycle precision states lie in $\mathcal{K}_0$. Lemma B.13 then gives

$$\|J_t - J\| + \|\mathcal{U}_t - \mathcal{U}\| + \|\mathcal{R}_t - \mathcal{R}\| + \sum_{r=1}^{k-1} \|Q_{t,r} - Q\| \le C_k (\theta_\Lambda^k)^t.$$

A telescoping expansion of the product $Q_{t,k-1} \cdots Q_{t,1}$ against Q^{k-1}, together with uniform boundedness of all factors, yields the displayed bound. The finite prefix is absorbed into $C_{M,k}$, and the geometric bound is summable. □

Proposition B.15 (Stability under a vanishing perturbation)**.** *Let $z_{t+1} = (A + E_t) z_t$ in finite dimension and $E_t \to 0$. If $\rho(A) < 1$, then $z_t \to 0$ for every z_0. Moreover, for every $q \in (\rho(A), 1)$, there is $C_q < \infty$ such that*

$$\|z_t\| \le C_q q^t \|z_0\|.$$

In particular, the conclusion holds if $\sum_t \|E_t\| < \infty$.

Proof. Choose an induced norm with $\|A\| < q$, possible by the standard adapted-norm construction for finite-dimensional matrices. Since $E_t \to 0$, the tail satisfies $\|A + E_t\| \le q$. The finite prefix is absorbed into the constant. □

B.5 Global convergence criterion and MAP target

Definition B.16 (Admissible initialization)**.** *An admissible initialization is a post-injection cycle-boundary state with $\Lambda^{(0)} \succeq 0$ and finite $x^{(0)}$, $\eta^{(0)}$, for which all GBP Schur complements and H-GBP updates are well defined.*

Theorem B.17 (Global necessary and sufficient criterion for H-GBP)**.** *Fix the cycle length k. The following statements are equivalent:*

(a) For every admissible initialization, the full H-GBP trajectory converges to the MAP-centered fixed point.

(b) For every admissible precision initialization, the exact cocycle $m_{t+1} = \widetilde{M}_{k,t} m_t$ satisfies $m_t \to 0$ for every $m_0 \in \mathcal{M}$.

(c) $\rho(M_k) < 1$.

(d) $\rho(T_{\mathrm{aug},k}) < 1$.

When these conditions hold, for every $q \in (\rho(M_k), 1)$ there exists $C_q < \infty$ such that

$$\|m_t\| \le C_q q^t \|m_0\|, \qquad \|v_t\| \le C_q q^t \|m_0\|, \qquad t \ge 0,$$

after enlarging C_q to absorb the finite prefix.

Proof. By Proposition B.12, the cycle-boundary dynamics are

$$m_{t+1} = \widetilde{M}_{k,t} m_t, \qquad v_t = C\mathcal{R}_t m_t.$$

The readout operators are uniformly bounded along the compact precision trajectory, so $m_t \to 0$ is equivalent to $(v_t, m_t) \to (0, 0)$ at cycle boundaries. Within-cycle states are obtained by finitely many uniformly bounded maps, so cycle-boundary convergence is equivalent to full-trajectory convergence.

If $\rho(M_k) < 1$, Lemma B.14 gives $\widetilde{M}_{k,t} = M_k + E_t$ with $E_t \to 0$, in fact summably. Applying Proposition B.15 gives geometric convergence of m_t, and uniform boundedness of $C\mathcal{R}_t$ gives the same rate for v_t.

Conversely, suppose the full hierarchy converges for every admissible initialization. Choose $\Lambda^{(0)} = \Lambda^\star$. Then the precision is frozen, $\widetilde{M}_{k,t} = M_k$, and the message recursion is $m_{t+1} = M_k m_t$. Convergence for every m_0 in a finite-dimensional linear system is equivalent to $\rho(M_k) < 1$. Finally, Lemma B.10 gives $\rho(M_k) = \rho(T_{\mathrm{aug},k})$. □

Theorem B.18 (Exact MAP target)**.** *If the equivalent conditions of Theorem B.17 hold, then for every admissible initialization,*

$$x_t \to x_\star = A^{-1} b, \qquad \eta_t \to \eta^\star(\Lambda^\star), \qquad \Lambda_t \to \Lambda^\star.$$

Thus the fine mean produced by H-GBP converges to the exact MAP solution of the base linear Gaussian factor graph.

Proof. Precision convergence follows from Theorem B.3 and Proposition B.4. Theorem B.17 gives $m_t \to 0$ and $v_t \to 0$. Since $v_t = x_t - x_\star$, this proves $x_t \to x_\star$.

Since the canonical reference $\eta^\star(\Lambda)$ is continuous in Λ and $\Lambda_t \to \Lambda^\star$, we also have $\eta_t = \eta^\star(\Lambda_t) + m_t \to \eta^\star(\Lambda^\star)$. □

B.6 H-GBP ON MANIFOLDS

Using an iterative inner solver for each linearized least-squares problem is the truncated/inexact Gauss–Newton construction (Gratton et al., 2007; Dembo et al., 1982). H-GBP fits this construction: at outer iteration t, it approximately solves

$$A_t\xi = b_t, \qquad A_t = J_t^\top J_t, \qquad b_t = -J_t^\top r_t, \tag{26}$$

and retracts $X_{t+1} = X_t \oplus \widehat{\xi}_t$. Here $F(X) = \frac{1}{2}\|r(X)\|_2^2$, and derivatives and tangent norms are expressed in gauge-fixed local coordinates. We now give a direct H-GBP result for local nonlinear convergence and its outer rate. Because H-GBP approaches the tangent solution iteratively, the only added algorithmic condition is the standard relative-residual stopping rule

$$\eta_t := \frac{\|A_t\widehat{\xi}_t - b_t\|_{A_t^{-1}}}{\|b_t\|_{A_t^{-1}}} \le \bar{\eta}, \qquad 0 < \bar{\eta} < \frac{1-\gamma_\star}{1+\gamma_\star}. \tag{27}$$

If $b_t = 0$, the algorithm terminates. No additional structural assumption is introduced: the outer conditions are the standard local Gauss–Newton conditions, and the inner conditions are exactly those of our linear H-GBP theorem.

Lemma B.19 (Standard inexact Gauss–Newton)**.** *Let $X_\star$ be a stationary point. Assume that the whitened residual r is twice continuously differentiable, the gauge-fixed matrix $A_\star = J_\star^\top J_\star$ is positive definite, and the retraction is smooth. Define*

$$S_\star := \nabla^2 F(X_\star) - A_\star, \qquad \gamma_\star := \|A_\star^{-1/2} S_\star A_\star^{-1/2}\|_2 < 1.$$

Let $\xi_t^\star = A_t^{-1} b_t$ be the exact Gauss–Newton step. If

$$\frac{\|\widehat{\xi}_t - \xi_t^\star\|_{A_t}}{\|\xi_t^\star\|_{A_t}} \le \eta_t \le \bar{\eta},$$

with $\bar{\eta}$ as in (27)*, then, for X_0 sufficiently close to $X_\star$,*

$$\limsup_{t\to\infty} \frac{\|X_{t+1} \ominus X_\star\|_{A_\star}}{\|X_t \ominus X_\star\|_{A_\star}} \le \gamma_\star + (1+\gamma_\star)\limsup_{t\to\infty}\eta_t. \tag{28}$$

Hence $\eta_t \le \bar{\eta} < (1-\gamma_\star)/(1+\gamma_\star)$ gives local Q-linear convergence.

Proof. This follows from the standard perturbation theory of locally contractive fixed-point iterations (Ortega & Rheinboldt, 1970, Chapter 10): the exact Gauss–Newton map has local asymptotic contraction factor $\gamma_\star$, while a relative step perturbation η_t contributes at most $(1+\gamma_\star)\eta_t$ asymptotically. □

Lemma B.20 (Residual–error identity)**.** *For $A_t \succ 0$ and $b_t \neq 0$,*

$$\frac{\|A_t\widehat{\xi}_t - b_t\|_{A_t^{-1}}}{\|b_t\|_{A_t^{-1}}} = \frac{\|\widehat{\xi}_t - \xi_t^\star\|_{A_t}}{\|\xi_t^\star\|_{A_t}}. \tag{29}$$

Proof. Since $A_t\xi_t^\star = b_t$, we have $A_t\widehat{\xi}_t - b_t = A_t(\widehat{\xi}_t - \xi_t^\star)$. The identity follows directly from the definitions of the A_t- and A_t^{-1}-norms. □

Theorem B.21 (Local convergence of nonlinear H-GBP)**.** *Assume the conditions of Lemma B.19 hold. At every outer iteration before termination, suppose the tangent graph satisfies our existing linear H-GBP assumptions and*

$$\rho(M_{k,t}) < 1,$$

where $M_{k,t}$ is the H-GBP cycle operator for the current linearization. Run H-GBP until the stopping rule (27) *holds. Then $X_{t+1} = X_t \oplus \widehat{\xi}_t$ converges locally to $X_\star$.*

Proof. Theorems B.17 and B.18 imply that $\rho(M_{k,t}) < 1$ makes H-GBP converge geometrically to $\xi_t^\star = A_t^{-1} b_t$, so the stopping tolerance is reached in finitely many cycles. Lemma B.20 converts that tolerance exactly into the forcing condition of Lemma B.19, which proves the claim. □

Corollary B.22 (H-GBP-controlled outer rate)**.** *Under the assumptions of Theorem B.21, for any $q_t \in (\rho(M_{k,t}), 1)$, our linear theorem gives a finite K_t such that after T H-GBP cycles,*

$$\eta_t^{(T)} \leq K_t q_t^T.$$

Hence the stopping rule is reached after finitely many cycles. For example, it is sufficient to take

$$T_t \geq \left\lceil \frac{\log(K_t/\bar{\eta})}{-\log q_t} \right\rceil \qquad \textit{whenever } K_t > \bar{\eta}.$$

The resulting nonlinear outer rate satisfies

$$q_{\text{outer}} \leq \gamma_\star + (1 + \gamma_\star)\bar{\eta} < 1. \tag{30}$$

Thus H-GBP's geometric inner convergence guarantees finite attainment of the prescribed forcing tolerance, while that tolerance explicitly controls the nonlinear outer convergence rate.

Proof. The finite-cycle bound follows from the linear H-GBP theorem. Substituting $\eta_t \leq \bar{\eta}$ into Lemma B.19 gives the outer-rate bound. □

C ALGORITHM

Algorithm 1 Hierarchical Gaussian BP

Require: Fixed linear Gaussian factor graph $\mathcal{G}$ with factor parameters $\{(\eta_f, \Lambda_f)\}_{f\in\mathcal{F}}$; groups $\{G\}$; ranks $\{r_G\}$; sweeps k; cycles T.

Assemble A and b; initialize all factor-to-variable messages to zero.
for each group G **do**
    Build H_G from factors satisfying $N(f) \subseteq G$ using (5).
    Store the r_G lowest eigenvectors of H_G as the local basis P_G.
end for
Assemble $P = \mathrm{blkassemble}\{P_G\}_G$ and factorize $A_c = P^\top AP$.
for $t = 0, \ldots, T-1$ **do**
    for $s = 1, \ldots, k$ **do**
        Compute variable-to-factor messages by (2).
        Compute factor-to-variable messages $\Lambda^+_{f\to i}$, $\eta^+_{f\to i}$ by the Schur update (3).
        Update all variable beliefs by (4) (and means x_i optionally).
    end for
    Stack $x = (x_1^\top, \ldots, x_N^\top)^\top$ and compute $r_c = P^\top(b - Ax)$.
    Solve $A_c\delta z = r_c$ and prolong $\delta x = P\delta z$.
    Inject $x_i \leftarrow x_i + \delta x_i$ and $\widehat{\eta}_i \leftarrow \widehat{\Lambda}_i x_i$ for all variables. Keep all $(\eta_{f\to i}, \Lambda_{f\to i})$ unchanged.
end for
return $x = (x_1^\top, \ldots, x_N^\top)^\top$.

D Additional Details

D.1 Implementation Details

All timings are CPU-only, on an Intel Core i9-13900HX system (24 physical cores, 32 logical processors, 64 GB RAM). A 16-thread run specifies a worker budget, not 16 physical cores. The released configurations specify one policy for SE(2), one for SE(3), and one for BA; none selects parameters by dataset name.

Precision reuse and packed messages. Within a fixed Gaussian model, precision updates are independent of the canonical vectors. Once the precision stopping rule is met, we cache the Schur gain

$$B_{f\to i} = \Lambda^f_{ij}(\Lambda^f_{jj} + \Lambda_{j\setminus f})^{-1}, \qquad \eta^+_{f\to i} = \eta^f_i - B_{f\to i}(\eta^f_j + \eta_{j\setminus f}).$$

Subsequent eta-only sweeps reuse both this gain and the belief precisions, replacing repeated small factorizations by matrix–vector products. This changes the per-message arithmetic from $O(d^3)$ to $O(d^2)$ for block dimension d. Caches are refreshed when the linearized model changes. Numerical freezing uses the task-specific tolerances below, rather than assuming an exact fixed point after a prescribed number of sweeps.

Local basis construction. For PGO, a group G assembles $H_G \in \mathbb{R}^{d|G|\times d|G|}$ from its internal factors and priors, and retains the r_G lowest eigenmodes as P_G. Here $d = 3$ for SE(2) and $d = 6$ for SE(3), with ranks 4 and 12, respectively (capped by the group dimension). SE(2) uses block inverse iteration with Rayleigh–Ritz extraction and an eigen-residual acceptance test, warm-started from the preceding basis when available. SE(3) first exploits exact band structure with a selected-eigenpair solver when the local block is sufficiently narrow-banded, and otherwise attempts the partial solve. Unaccepted partial solves fall back to a full symmetric eigendecomposition. These backends target the same low-mode space; local bases are rebuilt at each relinearization.

Parallel and distributed structure. Fine-level factor and belief passes, as well as per-group basis construction, admit parallel execution with local data access. Contiguous message slots, fixed-size block kernels, preallocated workspaces, and persistent worker regions reduce allocation and synchronization overhead; restriction accumulates group contributions and prolongation applies local basis blocks. The one-thread PGO path avoids launching parallel worker teams. Our implementation is shared-memory: reductions and the reduced solve still provide global coupling. In preliminary experiments, GBP on the induced second-level factor graph was slower than a direct solve. A likely reason is the much smaller coarse-system dimension, which makes a factorization reused across corrections cheaper than repeated coarse message sweeps. We therefore use sparse Cholesky for the PGO coarse system and dense $LDL^\top$ for the small BA coarse system, while retaining loopy GBP on the fine message graph.

D.2 Linear Experiment Details

The convergence studies use fixed Gaussian systems, with a direct solution as the MAP reference. Figure 6 varies the smoothing budget on a synthetic translation graph; Fig. 5(a,b) examines coarse spaces and spectral contraction. The companion fill-in timing study in Fig. 5(c) is described separately below.

D.2.1 Two-lap translation graph

The graph has 200 two-dimensional position variables, arranged on two circular laps of radius 10, with 100 positions per lap and random seed 11. Odometry initialization integrates consecutive relative displacements with Gaussian noise $\mathcal{N}(0, 0.10^2 I_2)$. Within-group chain factors use those measurements; connectors, skips, and cross-lap closures use ground-truth displacements perturbed by $\mathcal{N}(0, 0.01^2 I_2)$. Twenty groups contain ten consecutive variables each. A soft prior of weight 10^{-6} anchors the first position. Each group retains four modes (two scalar modes in each coordinate), reducing dimension 400 to 80. Table 3 specifies the edge counts and information weights; the displayed realization selects 135 within-group skips from 160 eligible candidates.

Table 3: Edge construction for Fig. 6.

Edge type	Construction	Count	Weight
Local chain	adjacent poses within each group	180	1.0
Inter-group connector	last pose of a group to first pose of next group	20	1.0
Short-range skip	selected within-group skips, distance 2–6	135	0.8
Long-range loop closure	cross-lap edges with phase jitter $\delta \in [-3, 3]$ nodes	16	0.2

Table 4: Sphere final relative error $\|e - e_{\mathrm{MAP}}\|_2 / \|e_{\mathrm{MAP}}\|_2$, as displayed in Fig. 5(a).

Coarse abstraction	Relative error
AMD	0.928
Message-λ	0.899
Low-rank	0.335

We vary $k \in \{20, 30, \ldots, 100\}$ and $c \in \{0, 1, \ldots, 20\}$. One GBP cycle contains k synchronous sweeps; H-GBP adds a coarse correction to the same fine-sweep budget. Both start at the same odometry estimate. The surface plots

$$\log_{10}\left(\frac{\|e_c - e_\star\|_A}{\|e_0 - e_\star\|_A}\right).$$

Thus the comparison isolates the benefit of abstraction and recovery at equal numbers of fine sweeps, not equal wall time. The energy norm measures the quadratic-system error; because the anchor is weak, it should not be interpreted as an unweighted trajectory MSE.

D.2.2 Coarse-space ablation: Fig. 5(a)

This panel freezes the first SE(3) linearization of Sphere, $He = \eta$, and uses $e_{\mathrm{MAP}} = H^{-1}\eta$ as the reference. Ordered groups contain 20 poses. Low-rank abstraction uses the lowest modes of the block assembled from priors and factors fully contained in a group. Message-λ abstraction additionally includes incoming boundary-message precision blocks. The algebraic multigrid comparator (labelled AMD in the figure) instead uses block strength of connection to select coarse pose nodes and interpolate neighboring fine nodes (Xu & Zikatanov, 2017).

Starting from $e = 0$, all three methods apply a coarse correction, 50 synchronous GBP sweeps, and a second coarse correction. Each correction solves $(P^\top H P)\delta z = P^\top(\eta - He)$ and injects $P\delta z$. Table 4 reproduces the final relative Euclidean errors shown in the panel. The low-rank space yields a substantially smaller remaining error on this system.

D.2.3 Spectral contraction: Fig. 5(b)

The spectral experiment again uses 200 translation variables, group size 10, rank 4, and $k = 50$. Only the short- and long-range connectivity varies (Table 5). We plot successive-cycle energy-error ratios $\|e_c - e_\star\|_A / \|e_{c-1} - e_\star\|_A$, for $c = 1, \ldots, 20$, against the estimated fixed-precision spectral radius $\rho(T_{\mathrm{aug},50}) = \rho(M_{50})$. The ratios approach the predicted asymptotic rates as precision messages stabilize. Faster initial contraction is compatible with the theory: a spectral radius describes the asymptotic rate, not a universal one-step norm bound.

D.2.4 Fill-in sensitivity: Fig. 5(c)

The fill-in diagnostic concerns the linear system $A_0\delta = b_0$ obtained by the first linearization at the initial odometry estimate. The Globe1k variants all have matrix dimension 6144 and 98,544 lower-triangular nonzeros, but different elimination fill-in. The accompanying recorded runtimes cover ten nonlinear outer iterations, rather than only this first linear solve. H-GBP uses five cycles, fifty pre-sweeps, group size 20, rank 12, and one thread throughout. Each method's runtime is normalized by its own smallest-fill case (Table 6). Across these variants, Cholesky time grows by

Table 5: Graph variants used in the spectral-radius contraction test.

Case	p_{skip}	n_{loop}	w_{skip}	w_{loop}	Short-range	Long-loop	$\rho(T_{\text{aug},50})$
Dense local / few loops	1.0	8	1.2	0.05	160	8	0.393
Moderate local / moderate loops	0.8	16	0.8	0.20	135	16	0.428
Sparse local / more loops	0.3	24	0.4	1.00	44	24	0.657

Table 6: Runtime scaling under increasing sparse-matrix fill-in.

Mode	Fill-in ratio	H-GBP time (s)	Cholesky time (s)	H-GBP growth	Cholesky growth
local_diag	5.066	1.459	0.475	1.000	1.000
medium_chords	6.141	1.504	0.603	1.031	1.269
separator_weave	6.781	1.500	0.682	1.028	1.436
mixed_chords	6.864	1.549	0.720	1.061	1.516
hemisphere_bridges	8.102	1.568	0.894	1.075	1.883

$1.883\times$, versus $1.075\times$ for H-GBP. This is an empirical sensitivity comparison at fixed problem size and sparsity, not an asymptotic complexity claim.

D.3 SE2/SE3 Pose-Graph Experiment Details

Baselines and protocol. The Cholesky reference uses the official g2o `LinearSolverCholmod` kernel (Kümmerle et al., 2011) inside our matched analytic Lie-log front end, with block ordering and symbolic reuse. The archived CHOLMOD build is simplicial: its dominant numerical factorization is serial, so additional front-end threads do not accelerate that stage. g2o-PCG uses block-preconditioned conjugate gradients (relative tolerance 10^{-6}, iteration cap equal to the scalar system dimension). Both use twenty Gauss–Newton outer iterations.

AMM-PGO (Fan & Murphey, 2024) combines distributed majorization–minimization with Nesterov acceleration and adaptive restart; we use the official implementation with a parallel-node adapter and sixteen partitions. Cycle-PGO (Bai et al., 2021) optimizes relative poses using a minimum cycle basis and cycle-consistency constraints. Schwarz-PGO uses sixteen subdomains, one-hop overlap, a GDSW coarse space, and PCG tolerance 10^{-8} Tee et al. (2026). Cycle-PGO and Schwarz-PGO allow up to twenty nonlinear iterations. AMM-PGO retains its recorded native budgets: 100 iterations for FR079 and Globe100k, 20 for Grid, and 300 otherwise.

All methods receive the same released graph files, including the offline PSD-conditioned Cubicle graph and the generated Globe graphs. H-GBP, Cholesky, PCG, and Schwarz use Huber threshold 5. AMM-PGO retains its native scalar-weighted chordal robust objective and initialization; Cycle-PGO retains its native non-robust objective. For comparison, final poses are rescored using the same full-information, half-quadratic Lie-log cost, so displayed costs are not the Huber objectives used internally. Cycle estimates are reconstructed into vertex poses before scoring. Timings are measured wall times with native boundaries retained; in particular, Schwarz totals include parsing. Sixteen threads denotes the available worker budget, not guaranteed parallelism in every stage.

Unified H-GBP settings. SE(2) uses 20 outer iterations, $c = 3$, $k = 100$, group size 20, and $r = 4$; SE(3) uses 20, $c = 5$, $k = 50$, group size 20, and $r = 12$. Analytical Jacobians produce each tangent system. Loopy GBP pre-smoothing and a reduced sparse direct solve alternate within every outer iteration, followed by retraction. Precision freezing requires three consecutive checks below relative tolerance 10^{-8}. Checks initially occur every five sweeps (the SE(2) interval may back off to twenty-five); frozen precisions are rechecked every twenty-five sweeps and reactivated if necessary. These rules are shared across datasets within each space.

Objective trajectories. Figure 9 retains the archived objective-versus-outer comparison with the direct Gauss–Newton reference as a qualitative convergence illustration, separate from the current timing runs. H-GBP consistently reduces the objective across the nine graphs and avoids the pronounced early spikes seen in the direct reference on Cubicle and Grid.

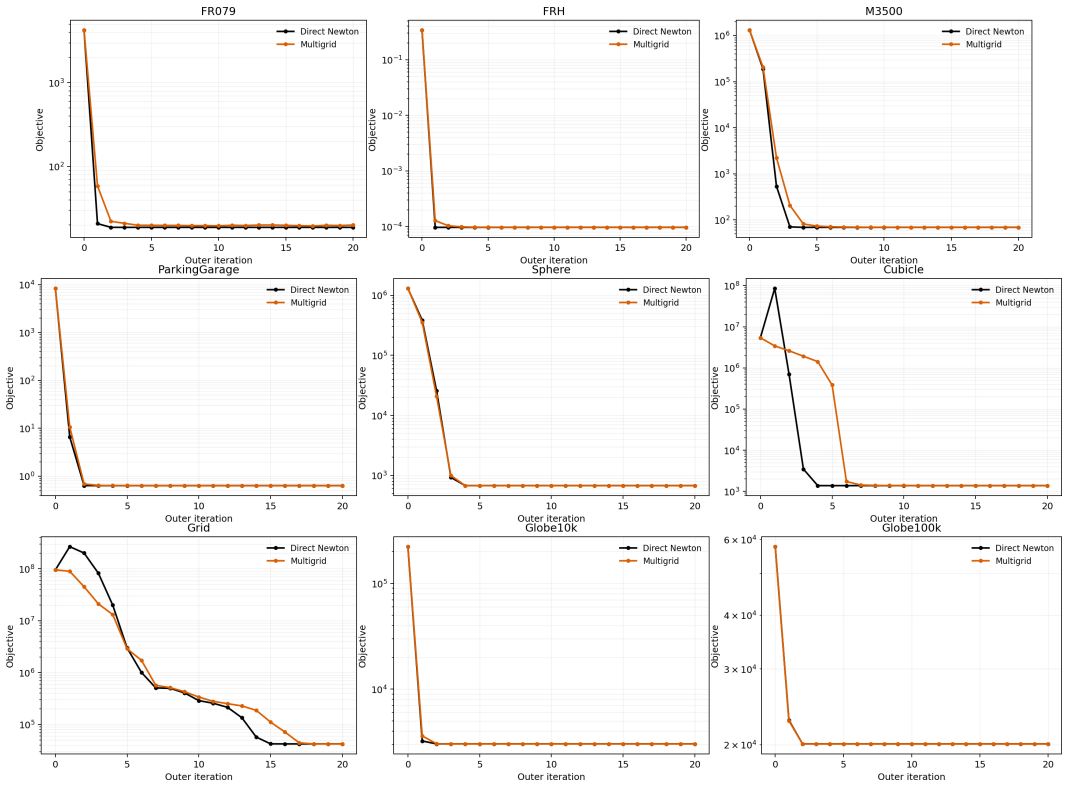


Figure 9: Archived objective trajectories of H-GBP and the direct Gauss–Newton reference over nonlinear outer iterations.

D.4 Bundle Adjustment Experiment Details

Baselines and common objective. Ceres Sparse uses sparse Cholesky on the reduced camera system (`SPARSE_SCHUR`, SuiteSparse/CHOLMOD); Ceres Iterative uses implicit Schur PCG with a Schur–Jacobi preconditioner (Agarwal et al., 2023). RootBA-QR eliminates landmarks by Householder QR in square-root form and iteratively solves the reduced camera problem (Demmel et al., 2021). PowerBA approximates the inverse Schur complement by a truncated power series, here of order ten (Weber et al., 2023). We use their released implementations through the common BAL/RootBA adapters, with double precision, identical input files, camera model, and normalization. All receive a budget of twenty nonlinear iterations and retain native stopping. Cost is the sum of squared two-dimensional reprojection residuals. Table timings exclude loading and common input normalization; Ceres reports `Solve` time, excluding wrapper problem construction, whereas H-GBP, RootBA, and PowerBA include their internal setup.

A single policy across scenes. Cameras are aggregated into connected groups of target size $\max(2, \lceil n_c/24 \rceil)$, and each group retains nine modes when abstracted. The damping coefficient for the fine graph is one. Precision messages are computed for each camera system until relative change is below 10^{-6}. We then reuse the cached eta maps across H-GBP applications. Grouping and sparsity follow the same graph rules for every scene.

Cost versus iteration time. Figure 10 compares the actual per-outer histories of H-GBP and PowerBA on all ten BA benchmarks. H-GBP reaches lower final cost on these runs while spending less time in the nonlinear iterations. To avoid mixing timing boundaries, the horizontal axis sums the recorded outer durations for both methods, excluding one-time setup, loading, and normalization; the main table separately reports solver totals. Each point is one outer attempt, and rejected PowerBA trials retain the previous accepted cost while their elapsed time is still counted. Each H-GBP curve is a

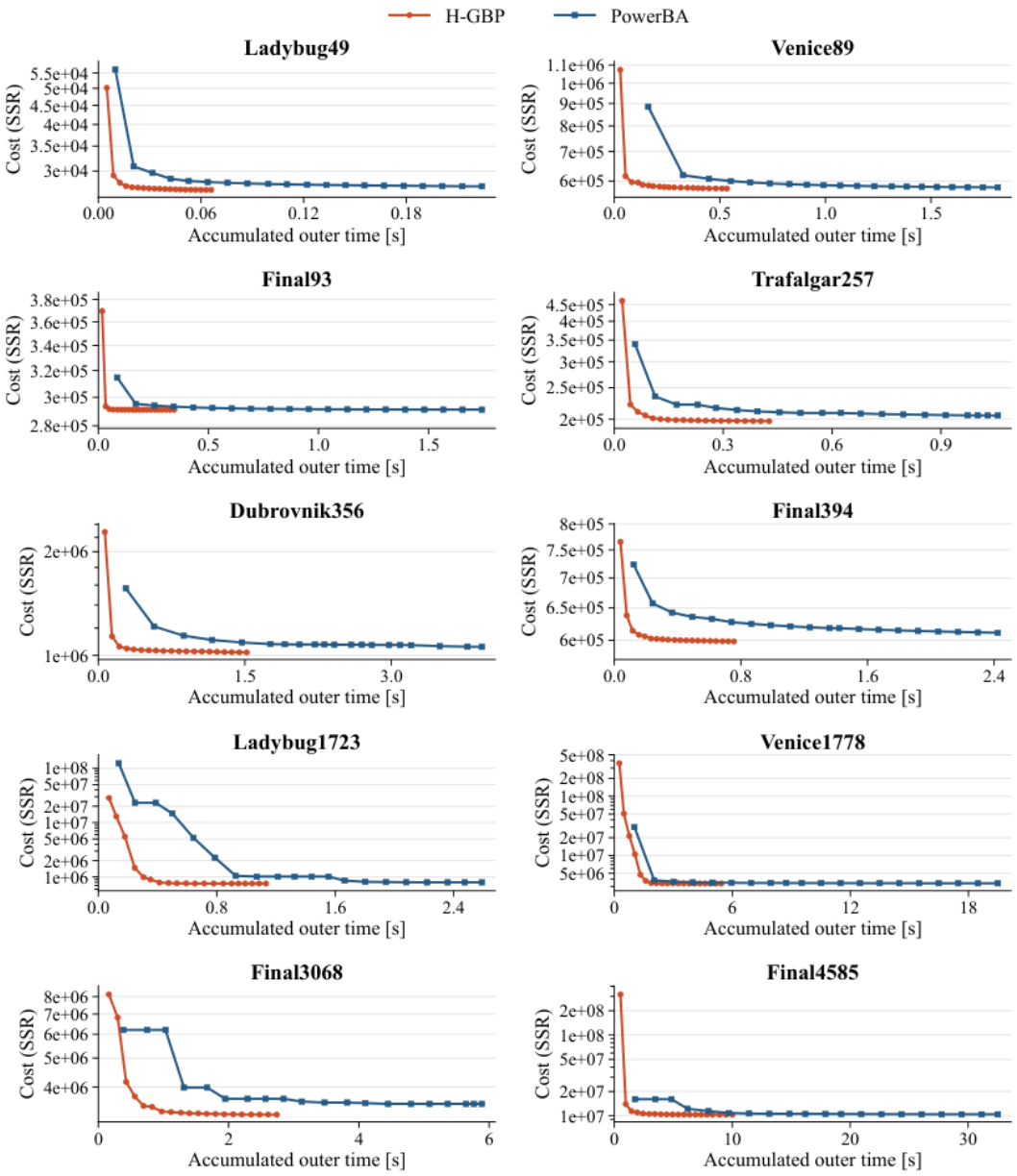


Figure 10: H-GBP versus PowerBA on BA, both with sixteen threads. Markers show outer attempts 1–20; vertical axes show reprojection cost (SSR) on logarithmic scales. Time is accumulated inside outer iterations, excluding one-time setup. Curves use recorded durations without rescaling.

complete run with median total runtime among three repeats, not a curve assembled from per-iteration minima.

E Discussion

Toward multiple levels. Algorithmically, extending the current two-level construction to a fully multilevel hierarchy is nontrivial. The upper-level system

$$A_c \delta z = r_c$$

is a Galerkin algebraic system, and different coarse factor-graph realizations may induce the same matrix equation. Thus a multilevel GBP construction must decide not only the dimension of each coarse variable, but also how unary and binary factors should be assigned among abstracted coarse nodes. A natural attempt is to assign one coarse variable to each fine group, with dimension equal to the retained local eigenbasis size, project within-group factors into unary coarse factors, and project cross-group factors into between-group coarse factors. In our preliminary experiments, however, GBP on this induced coarse factor graph typically required many iterations to solve the coarse system and was slower than a direct sparse solve at that level. This does not diminish the two-level result; rather, it highlights a compelling open direction: designing purely message-passing upper levels that retain the scalability and locality needed for fully distributed spatial-intelligence systems.

Broader applications. Finally, the abstraction-and-recovery principle behind H-GBP is not specific to pose graphs or bundle adjustment's cameras. Many sparse factor-graph problems have natural intermediate structures that could define useful coarse abstractions. In multi-robot SLAM and navigation, each robot's local subgraph could form a group, with higher-level messages coordinating inter-robot constraints. In visual odometry, pixels, patches, or object-level regions could be grouped into progressively coarser variables to accelerate GBP-based inference. Exploring such applications may reveal when hierarchical message passing can provide both distributed computation and strong global correction in broader sparse graphical models.