# Supplementary material

## A Proof of Theorems 4 and 5

### A.1 Proof of Theorem 4

*Proof.* For part (i), let $\boldsymbol{Q}^\star \in \mathrm{SO}(p)^n$ be a minimizer of (2), with corresponding optimal value $\tilde{f}^\star$, and suppose for contradiction that $\boldsymbol{S}^\star = \Pi(\boldsymbol{Q}^\star)$ is *not* a minimizer of (19). Then there exists some $\boldsymbol{S}' \in \mathrm{St}(d,p)^n$ with $f(\boldsymbol{S}') < f(\boldsymbol{S}^\star) = \tilde{f}^\star$. However, since $\Pi$ is surjective for $p > d$, then there exists some $\boldsymbol{Q}' \in \Pi^{-1}(\boldsymbol{S}')$, and by definition of $\tilde{f}$ in (26) we have that $\tilde{f}(\boldsymbol{Q}') = f(\boldsymbol{S}') < \tilde{f}^\star$, contradicting the optimality of $\boldsymbol{Q}^\star$. We conclude that $\boldsymbol{S}^\star$ is indeed a minimizer of $f$ over $\mathrm{St}(d,p)^n$, and therefore the optimal values of (19) and (2) coincide.

Part (ii) follows immediately from part (i) since $\tilde{f}(\boldsymbol{Q}) = f(\boldsymbol{S})$ for any $\boldsymbol{Q} \in \mathrm{SO}(p)^n$ and $\boldsymbol{S} = \Pi(\boldsymbol{Q})$ by (26).

### A.2 Proof of Theorem 5

In this section we derive Theorem 5 as a consequence of Theorem 3 and the equivalence of the rank-restricted optimization (19) and the Shonan Averaging problem (2) (Theorem 4). To do so, we need to understand how the local geometry of the lifted objective $\tilde{f}$ relates to that of $f$ near a critical point $\boldsymbol{Q}^\star \in \mathrm{SO}(p)^n$. Recall that the tangent spaces of the rotational and Stiefel manifolds can be expressed as:

$$T_{\boldsymbol{Q}}(\mathrm{SO}(p)) = \left\{ \boldsymbol{Q}\dot{\boldsymbol{\Delta}} \mid \dot{\boldsymbol{\Delta}} \in \mathrm{Skew}(p) \right\} \tag{31}$$

and

$$T_{\boldsymbol{S}}(\mathrm{St}(d,p)) = \left\{ \boldsymbol{S}\dot{\boldsymbol{\Omega}} + \boldsymbol{V}\dot{\boldsymbol{K}} \mid \begin{array}{l} \boldsymbol{\Omega} \in \mathrm{Skew}(d), \\ \dot{\boldsymbol{K}} \in \mathbb{R}^{(p-d)\times d}, \\ \boldsymbol{V} \in \mathbb{R}^{p\times(p-d)}, \\ \boldsymbol{S}^\mathsf{T}\boldsymbol{V} = 0 \end{array} \right\} \tag{32}$$

respectively (cf. Example 3.5.2 of [1]). If we rewrite the elements of (31) in a block-partitioned form compatible with the action of the projection (22):

$$T_{\boldsymbol{Q}}(\mathrm{SO}(p)) = \left\{ \boldsymbol{Q}\begin{bmatrix} \dot{\boldsymbol{\Omega}} & -\dot{\boldsymbol{K}}^\mathsf{T} \\ \dot{\boldsymbol{K}} & \dot{\boldsymbol{\Gamma}} \end{bmatrix} \mid \begin{array}{l} \dot{\boldsymbol{\Omega}} \in \mathrm{Skew}(d), \\ \dot{\boldsymbol{K}} \in \mathbb{R}^{(p-d)\times d}, \\ \dot{\boldsymbol{\Gamma}} \in \mathrm{Skew}(p-d) \end{array} \right\} \tag{33}$$

then the derivative of the projection $\pi$ at $\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{S} & \boldsymbol{V} \end{bmatrix} \in \mathrm{SO}(p)$ is:

$$\begin{aligned} d\pi_{\boldsymbol{Q}} &\colon T_{\boldsymbol{Q}}(\mathrm{SO}(p)) \to T_{\boldsymbol{S}}(\mathrm{St}(d,p)) \\ d\pi_{\boldsymbol{Q}} &\left( \boldsymbol{Q}\begin{bmatrix} \dot{\boldsymbol{\Omega}} & -\dot{\boldsymbol{K}}^\mathsf{T} \\ \dot{\boldsymbol{K}} & \dot{\boldsymbol{\Gamma}} \end{bmatrix} \right) = \boldsymbol{S}\dot{\boldsymbol{\Omega}} + \boldsymbol{V}\dot{\boldsymbol{K}}. \end{aligned} \tag{34}$$

The next result is a direct consequence of (31)–(34):

**Lemma 1.** *The projection* $\pi\colon \mathrm{SO}(p) \to St(d,p)$ *is a submersion.*[6]

*Proof.* Let $\boldsymbol{Q} = \begin{bmatrix}\boldsymbol{S}\ \boldsymbol{V}\end{bmatrix} \in \mathrm{SO}(p)$, $\boldsymbol{S} = \pi(\boldsymbol{Q})$, and $\dot{\boldsymbol{Y}} \in T_{\boldsymbol{S}}(\mathrm{St}(d,p))$; we must show that there is some $\dot{\boldsymbol{X}} \in T_{\boldsymbol{Q}}(\mathrm{SO}(p))$ such that $\dot{\boldsymbol{Y}} = d\pi_{\boldsymbol{Q}}(\dot{\boldsymbol{X}})$. Since $\boldsymbol{S}^{\mathsf{T}}\boldsymbol{V} = 0$ (as the columns of $\boldsymbol{Q}$ are orthonormal), by (32) there exist some $\dot{\boldsymbol{\Omega}} \in \mathrm{Skew}(d)$ and $\dot{\boldsymbol{K}} \in \mathbb{R}^{(p-d)\times d}$ such that $\dot{\boldsymbol{Y}} = \boldsymbol{S}\dot{\boldsymbol{\Omega}} + \boldsymbol{V}\dot{\boldsymbol{K}}$. Letting $\dot{\boldsymbol{\Gamma}} \in \mathrm{Skew}(p-d)$, and defining

$$\dot{\boldsymbol{X}} = \boldsymbol{Q}\begin{bmatrix}\dot{\boldsymbol{\Omega}} & -\dot{\boldsymbol{K}}^{\mathsf{T}} \\ \dot{\boldsymbol{K}} & \dot{\boldsymbol{\Gamma}}\end{bmatrix} \in T_{\boldsymbol{Q}}(\mathrm{SO}(p)) \tag{35}$$

as in (33), it follows from (34) that $\dot{\boldsymbol{Y}} = d\pi_{\boldsymbol{Q}}(\dot{\boldsymbol{X}})$.

**Corollary 1.** *The projection* $\Pi\colon \mathrm{SO}(p)^n \to St(d,p)^n$ *is a submersion.*

**Proof of Theorem 5**:

Part(i): Applying the Chain Rule to (26) produces:

$$d\tilde{f}_{\boldsymbol{Q}^\star} = df_{\boldsymbol{S}^\star} \circ d\Pi_{\boldsymbol{Q}^\star}. \tag{36}$$

The left-hand side of (36) is 0 because $\boldsymbol{Q}^\star$ is a first-order critical point of $\tilde{f}$ by hypothesis, and Corollary 1 shows that $d\Pi_{\boldsymbol{Q}^\star}$ is full-rank (i.e., its image is all of $T_{\boldsymbol{S}^\star}(\mathrm{St}(d,p))^n$). It follows that $df_{\boldsymbol{S}^\star} = 0$, and therefore $\boldsymbol{S}^\star$ is a first-order critical point of (19).

Part (ii): Observe that $\Pi(\boldsymbol{Q}^+) = \boldsymbol{S}^+$ as defined in Theorem 3(ii); since $\boldsymbol{S}^+$ is a stationary point for (19), another application of the Chain Rule as in (36) shows that $d\tilde{f}_{\boldsymbol{Q}^+} = 0$, and therefore $\boldsymbol{Q}^+$ is also a first-order critical point of the lifted optimization (2) in dimension $p+1$. Theorem 3(ii) also provides the second-order descent direction $\dot{\boldsymbol{S}}^+$ from $\boldsymbol{S}^+$. The tangent vector $\dot{\boldsymbol{Q}}^+$ defined in (28) is constructed as a *lift* of $\dot{\boldsymbol{S}}^+$ to $T_{\boldsymbol{Q}^+}(\mathrm{SO}(p+1)^n)$; that is, so that $\dot{\boldsymbol{Q}}^+$ satisfies $d\Pi_{\boldsymbol{Q}^+}(\dot{\boldsymbol{Q}}^+) = \dot{\boldsymbol{S}}^+$. Indeed, using (27), (28), and (34) we can compute each block of $d\Pi_{\boldsymbol{Q}^+}(\dot{\boldsymbol{Q}}^+)$ as:

$$\begin{aligned} d\Pi_{\boldsymbol{Q}^+}(\dot{\boldsymbol{Q}}^+)_i &= d\pi_{\boldsymbol{Q}_i^+}\left(\begin{bmatrix}\boldsymbol{Q}_i^\star & 0 \\ 0 & 1\end{bmatrix}\begin{bmatrix}0 & -v_i \\ v_i^{\mathsf{T}} & 0\end{bmatrix}\right) \\ &= \begin{bmatrix}0 \\ v_i^{\mathsf{T}}\end{bmatrix} = \dot{\boldsymbol{S}}_i^+ \end{aligned} \tag{37}$$

for all $i \in [n]$, verifying that $\dot{\boldsymbol{Q}}^+$ is a lift of $\dot{\boldsymbol{S}}^+$.

Our goal now is to show that $\dot{\boldsymbol{Q}}^+$ is likewise a second-order descent direction from $\boldsymbol{Q}^+$. We do this using a proof technique similar to that of Proposition 5.5.6 in [1]. Let $R_{\boldsymbol{Q}^+}\colon T_{\boldsymbol{Q}^+}(\mathrm{SO}(p+1)^n) \to \mathrm{SO}(p+1)^n$ be any (first-order) retraction on the tangent space of $\mathrm{SO}(p+1)^n$ at $\boldsymbol{Q}^+$, and let $\epsilon > 0$ be sufficiently small that the curve:

$$\begin{aligned} \tilde{\gamma}&\colon (-\epsilon,\epsilon) \to \mathrm{SO}(p+1)^n \\ \tilde{\gamma}(t) &= R_{\boldsymbol{Q}^+}(t\dot{\boldsymbol{Q}}^+) \end{aligned} \tag{38}$$

[6] A smooth mapping $\varphi\colon \mathcal{X} \to \mathcal{Y}$ between manifolds is called a *submersion* at a point $x \in \mathcal{X}$ if its derivative $d\varphi_x\colon T_x(\mathcal{X}) \to T_{\varphi(x)}(\mathcal{Y})$ is surjective. It is called a *submersion* (unqualified) if it is a submersion at *every* $x \in \mathcal{X}$.

obtained by moving through the point $\boldsymbol{Q}^+ \in \mathrm{SO}(p+1)^n$ along the direction $\dot{\boldsymbol{Q}}^+$ is well-defined. Our approach simply involves examining the behavior of the lifted objective $\tilde{f}$ at points along the curve $\tilde{\gamma}$ in a neighborhood of $\tilde{\gamma}(0) = \boldsymbol{Q}^+$. To do so, we define:

$$\begin{aligned} \tilde{f}&\colon (-\epsilon, \epsilon) \to \mathbb{R} \\ \tilde{f}(t) &= \tilde{f} \circ \tilde{\gamma}(t) \end{aligned} \tag{39}$$

and then consider its first- and second-order derivatives. Once again using the Chain Rule (and unwinding the definition of $\tilde{f}$), we compute:

$$\begin{aligned} \tilde{f}'(t) &= d\tilde{f}_{\tilde{\gamma}(t)} \circ \tilde{\gamma}'(t) \\ &= df_{\Pi\circ\tilde{\gamma}(t)} \circ d\Pi_{\tilde{\gamma}(t)} \circ \tilde{\gamma}'(t) \\ &= \left\langle \operatorname{grad} f\left(\Pi \circ \tilde{\gamma}(t)\right),\, d\Pi_{\tilde{\gamma}(t)} \circ \gamma'(t) \right\rangle. \end{aligned} \tag{40}$$

Note that at $t = 0$, $\operatorname{grad} f\left(\Pi \circ \tilde{\gamma}(0)\right) = \operatorname{grad} f(\Pi(\boldsymbol{Q}^+)) = \operatorname{grad} f(\boldsymbol{S}^+) = 0$ since $\boldsymbol{S}^+$ is a stationary point. It follows from (40) that $f'(0) = 0$ (as expected, since we know that $\boldsymbol{Q}^+$ is a stationary point for the lifted optimization (2)). Continuing, we compute the second derivative of $\tilde{f}(t)$ by applying the Product and Chain Rules to differentiate the inner product on the final line of (40):

$$\begin{aligned} \tilde{f}''(t) &= \frac{d}{dt}\left[\left\langle \operatorname{grad} f\left(\Pi \circ \tilde{\gamma}(t)\right),\, d\Pi_{\tilde{\gamma}(t)} \circ \gamma'(t) \right\rangle\right] \\ &= \left\langle \frac{d}{dt}\left[\operatorname{grad} f\left(\Pi \circ \tilde{\gamma}(t)\right)\right],\, d\Pi_{\tilde{\gamma}(t)} \circ \gamma'(t) \right\rangle \\ &\quad + \left\langle \operatorname{grad} f\left(\Pi \circ \tilde{\gamma}(t)\right),\, \frac{d}{dt}\left[d\Pi_{\tilde{\gamma}(t)} \circ \gamma'(t)\right] \right\rangle. \end{aligned} \tag{41}$$

Now, we just saw that $\operatorname{grad} f\left(\Pi \circ \tilde{\gamma}(0)\right) = 0$, so the second term of the final line of (41) is zero for $t = 0$. Moreover, the derivative in the first term can be further developed as:

$$\frac{d}{dt}\left[\operatorname{grad} f\left(\Pi \circ \tilde{\gamma}(t)\right)\right] = \operatorname{Hess} f\left(\Pi \circ \tilde{\gamma}(t)\right)\left[d\Pi_{\tilde{\gamma}(t)} \circ \tilde{\gamma}'(t)\right]. \tag{42}$$

Therefore, at $t = 0$ (41) simplifies as:

$$\begin{aligned} \tilde{f}''(0) &= \left\langle \operatorname{Hess} f\left(\Pi \circ \tilde{\gamma}(0)\right)\left[d\Pi_{\tilde{\gamma}(0)} \circ \tilde{\gamma}'(0)\right],\, d\Pi_{\tilde{\gamma}(0)} \circ \gamma'(0) \right\rangle \\ &= \left\langle \operatorname{Hess} f\left(\Pi(\boldsymbol{Q}^+)\right)\left[d\Pi_{\boldsymbol{Q}^+}(\dot{\boldsymbol{Q}}^+)\right],\, d\Pi_{\boldsymbol{Q}^+}(\dot{\boldsymbol{Q}}^+) \right\rangle \\ &= \left\langle \operatorname{Hess} f\left(\boldsymbol{S}^+\right)\left[\dot{\boldsymbol{S}}^+\right],\, \dot{\boldsymbol{S}}^+ \right\rangle \\ &< 0 \end{aligned} \tag{43}$$

where the final line of (43) follows from the fact that $\dot{\boldsymbol{S}}^+$ is a second-order direction of descent from $\boldsymbol{S}^+$. We conclude from (43) that $\dot{\boldsymbol{Q}}^+$ is a second-order direction of descent from $\boldsymbol{Q}^+$, as desired. □

# B Gauss-Newton for Shonan Averaging

We can implement the local search for (first-order) critical points of (2) required in line 3 of the Shonan Averaging algorithm using the same Gauss-Newton approach described in Section 3.

## B.1 Linearization

As before, we first rewrite (2) more explicitly as the minimization of the sum of the individual measurement residuals, in a vectorized form:

$$\min_{\boldsymbol{Q}\in \mathrm{SO}(p)^n} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} \left\|\mathrm{vec}(\boldsymbol{Q}_j\boldsymbol{P} - \boldsymbol{Q}_i\boldsymbol{P}\bar{\boldsymbol{R}}_{ij})\right\|_2^2, \tag{44}$$

and reparameterize this minimization in terms in terms of the Lie algebra $\mathfrak{so}(p)$:

$$\min_{\boldsymbol{\delta}\in \mathfrak{so}(p)^n} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} \left\|\mathrm{vec}\left(\boldsymbol{Q}_j e^{[\boldsymbol{\delta}_j]}\boldsymbol{P} - \boldsymbol{Q}_i e^{[\boldsymbol{\delta}_i]}\boldsymbol{P}\boldsymbol{R}_{ij}\right)\right\|_2^2. \tag{45}$$

Once again, we approximate (45) to first order as the linear least-squares objective:

$$\min_{\boldsymbol{\delta}\in \mathfrak{so}(p)^n} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} \left\|\boldsymbol{F}_j\boldsymbol{\delta}_j - \boldsymbol{H}_i\boldsymbol{\delta}_i - \boldsymbol{b}_{ij}\right\|_2^2, \tag{46}$$

where now the Jacobians $\boldsymbol{F}_j$ and $\boldsymbol{F}_i$ and the right-hand side $\boldsymbol{b}_{ij}$ are calculated as

$$\boldsymbol{F}_j \doteq (\boldsymbol{P}^\mathsf{T} \otimes \boldsymbol{Q}_j)\bar{\boldsymbol{G}}_p \tag{47}$$

$$\boldsymbol{H}_i \doteq ((\boldsymbol{P}\bar{\boldsymbol{R}}_{ij})^\mathsf{T} \otimes \boldsymbol{Q}_i)\bar{\boldsymbol{G}}_p \tag{48}$$

$$\boldsymbol{b}_{ij} \doteq \mathrm{vec}(\boldsymbol{Q}_i\boldsymbol{P}\bar{\boldsymbol{R}}_{ij} - \boldsymbol{Q}_j\boldsymbol{P}) \tag{49}$$

with $\bar{\boldsymbol{G}}_p$ the matrix of vectorized $\mathfrak{so}(p)^n$ generators. Note that the right-hand side $\boldsymbol{b}_{ij}$ in (49) in fact involves only the Stiefel manifold elements $\boldsymbol{S}_i = \pi(\boldsymbol{Q}_i)$:

$$\boldsymbol{b}_{ij} = \mathrm{vec}(\boldsymbol{S}_i\bar{\boldsymbol{R}}_{ij} - \boldsymbol{S}_j). \tag{50}$$

## B.2 The structure of the Lie algebra

In this section we investigate the structure of the Lie algebra $\mathfrak{so}(p)$ as it pertains to the linearized objective in (46). Recall that $\mathfrak{so}(p)$ is identified with the tangent space $T_{\mathbf{I}_p}(\mathrm{SO}(p)) = \mathrm{Skew}(p)$, the space of $p \times p$ skew-symmetric matrices. Once again writing these matrices in a block-partitioned form compatible with the projection $\pi$ in (22) (as in (33)) produces:

$$\mathfrak{so}(p) = \left\{ \begin{bmatrix} [\boldsymbol{\omega}] & -\boldsymbol{K}^\mathsf{T} \\ \boldsymbol{K} & [\gamma] \end{bmatrix} \mid \begin{matrix} \boldsymbol{\omega} \in \mathfrak{so}(d), \\ \boldsymbol{K} \in \mathbb{R}^{(p-d)\times d}, \\ \boldsymbol{\gamma} \in \mathfrak{so}(p-d) \end{matrix} \right\}, \tag{51}$$

and it follows from (22) the derivative of $\pi$ at $\mathbf{I}_p$ is:

$$d\pi_{\mathbf{I}_p}\left(\begin{bmatrix}[\boldsymbol{\omega}] & -\boldsymbol{K}^\mathsf{T} \\ \boldsymbol{K} & [\gamma]\end{bmatrix}\right) = \begin{bmatrix}[\boldsymbol{\omega}] \\ \boldsymbol{K}\end{bmatrix}. \tag{52}$$

Note that $d\pi_{\mathbf{I}_p}$ does *not* depend upon the $(p-d)$-dimensional vector $\boldsymbol{\gamma}$. In particular, let us define:

$$V_{\mathbf{I}_p} \doteq \ker d\pi_{\mathbf{I}_p} = \left\{ \begin{pmatrix} 0 & 0 \\ 0 & [\gamma] \end{pmatrix} \mid \boldsymbol{\gamma} \in \mathfrak{so}(p-d) \right\} \subset \mathfrak{so}(p). \tag{53}$$

Geometrically, the set $V_{\mathbf{I}_p}$ defined in (53) consists of those directions of motion $\dot{\boldsymbol{\Omega}} \in \mathfrak{so}(p)$ at $\mathbf{I}_p$ along which the projection $\pi$ is *constant*. Equivalently:

$$V_{\mathbf{I}_p} = T_{\mathbf{I}_p}(\pi^{-1}(\boldsymbol{P})); \tag{54}$$

that is, $V_{\mathbf{I}_p}$ is the set of vectors that are *tangent* to the preimage $\pi^{-1}(\boldsymbol{P})$ of $\boldsymbol{P} = \pi(\mathbf{I}_p)$ at $\mathbf{I}_p$. If we think of the elements of the preimage $\pi^{-1}(\boldsymbol{P})$ as being vertically "stacked" above their common projection $\boldsymbol{P} \in \mathrm{St}(d,p)$, then the subspace $V_{\mathbf{I}_p}$ of the Lie algebra is precisely the set of tangent vectors at $\mathbf{I}_p$ that correspond to *vertical motions*. Consequently, $V_{\mathbf{I}_p}$ is referred to as the *vertical space*. We may define a corresponding *horizontal space* in the natural way, i.e., as the orthogonal complement of the vertical space:

$$H_{\mathbf{I}_p} \doteq V_{\mathbf{I}_p}^{\perp} = \left\{ \begin{pmatrix} [\boldsymbol{\omega}] & -\boldsymbol{K}^\mathsf{T} \\ \boldsymbol{K} & 0 \end{pmatrix} \mid \begin{array}{l} \boldsymbol{\omega} \in \mathfrak{so}(d), \\ \boldsymbol{K} \in \mathbb{R}^{(p-d)\times d} \end{array} \right\} \subset \mathfrak{so}(p). \tag{55}$$

The significance of (51)–(55) is that, to first order, the exponential map (or any retraction) can be written as $\mathbf{I} + [\boldsymbol{\delta}]$ for $\boldsymbol{\delta} \in \mathfrak{so}(p)$. In conjunction with the projection map $\pi$ (equivalently, with $\boldsymbol{P}$), this implies:

$$\boldsymbol{Q}_i e^{[\boldsymbol{\delta}]}\boldsymbol{P} \approx \boldsymbol{Q}_i(\mathbf{I} + [\delta])\boldsymbol{P} = \boldsymbol{S}_i + \boldsymbol{Q}_i \begin{bmatrix}[\boldsymbol{\omega}] \\ \boldsymbol{K}\end{bmatrix}. \tag{56}$$

From this we see that the derivative of the cost function (45) will not depend on the $(p-d)$-dimensional vector $\boldsymbol{\gamma}$, i.e., on the component of $[\boldsymbol{\delta}]$ lying in the vertical space $V_{\mathbf{I}_p}$. This makes intuitive sense, since the Shonan Averaging objective $\tilde{f}(\boldsymbol{Q})$ in (2) is defined in terms of the projection $\Pi(\boldsymbol{Q})$ (recall (26)), and moving along vertical directions leaves this projection unchanged.

This in turn enables us to characterize the Jacobians in more detail. If we split $\boldsymbol{Q}_i = \begin{bmatrix}\boldsymbol{S}_i & \boldsymbol{V}_i\end{bmatrix}$, the Jacobians $\boldsymbol{F}_j$ and $\boldsymbol{H}_i$ can be shown to be:

$$\boldsymbol{F}_j \doteq \left[(\mathbf{I}_d \otimes \boldsymbol{S}_j)\bar{\boldsymbol{G}}_d \; (\mathbf{I}_d \otimes \boldsymbol{V}_j) \; 0\right] \tag{57}$$

$$\boldsymbol{H}_i \doteq \left[(\bar{\boldsymbol{R}}_{ij}^\mathsf{T} \otimes \boldsymbol{S}_i)\bar{\boldsymbol{G}}_d \; (\bar{\boldsymbol{R}}_{ij}^\mathsf{T} \otimes \boldsymbol{V}_i) \; 0\right] \tag{58}$$

where $\bar{\boldsymbol{G}}_d$ is the matrix of vectorized generators for the Lie algebra $\mathfrak{so}(d)$ (as in Section 3). Again we see that the last columns, corresponding to the vertical directions, are zero.

The astute reader may now wonder whether the rank-deficiency of these Jacobians poses any numerical difficulties when solving the linear systems needed to compute the update step $\boldsymbol{\delta}$. In fact there are several straightforward ways to address this. One approach is simply to employ the Levenberg-Marquardt method directly in conjunction with the Jacobians (58); in this case, the Tikhonov regularization applied by the LM algorithm itself will ensure that all of the linear systems to be solved are nonsingular. Moreover, this regularization will additionally encourage update steps to lie in the horizontal subspace, since any update with a nonzero vertical component will have to "pay" for the length of that component (via regularization), while "gaining" nothing for it (in terms of reducing the local model of the objective).

Alternatively, one can remove the final $(p - d)$ all-0's columns from the Jacobians in (58), and solve the resulting reduced linear system in the variables $(\boldsymbol{\omega}, \boldsymbol{K})$. Geometrically, this corresponds to minimizing the local quadratic model of the objective (46) over the horizontal subspace; a horizontal full-space update $[\boldsymbol{\delta}]$ can then be obtained by simply taking $\boldsymbol{\gamma} = 0$. It is straightforward to see that this procedure corresponds to computing a pseudoinverse (minimum-norm) minimizer of the quadratic model (46).

Finally, a third approach is to regularize the original Shonan Averaging problem (2) by adding a prior term on the Karcher mean of the rotations $\boldsymbol{Q}_i$ in $\boldsymbol{Q}$; this has the effect of fixing the gauge for the underlying estimation problem, similarly to the use of "inner constraints" in photogrammetry [34].

# C More Experimental Results on the YFCC Datasets

In this section we present more extensive results on the datasets derived by Heinly et al. [21] from the large-scale YFCC-100M dataset [33,21]. As in the main paper, the relative measurements for these were derived from the SFM solution provided with the data, and corrupted with noise as before, using $\sigma = 0.2$. For all results below, minimum, average, and maximum running times (in seconds) are computed over 10 random initializations for each dataset. Also shown is the fraction of cases in which the method converges to a global minimizer. All Shonan Averaging variants examined below use the same Levenberg-Marquardt non-linear optimizer, with a Jacobi-preconditioned conjugate gradient method as the linear system solver.

## C.1 Small Datasets ($n < 50$)

Table 4 shows additional experimental results on the small YFCC datasets (with $n < 50$) with a more systematic exploration of the Shonan parameters. The intent is to provide more quantitative results for Shonan Averaging and its convergence properties at different levels of $p$, as well as explore parameter settings for optimal performance in practical settings.

In particular, we compared

- SA: Shonan Averaging with $p_{min} = 5$ and $p_{max} = 30$.
- BD: the block coordinate descent method from [13,14].
- SL: Same as SA, but starting (L)ow, from $SO(3)$: $p_{min} = 3$ and $p_{max} = 30$.
- S3: Only run with $p = 3$, which corresponds to LM in the main paper: we only optimize at the base $SO(3)$ level. Note that this approach has no guarantee of converging to a global minimizer.
- S4: Similar to S3, but with $P = 4$: assesses in what percentage of cases we converge to global minima for $p = 4$.
- S5: Similar to S3 and S4, but for $p = 5$.
- SK: Shonan Averaging with $p_{min} = 5$ and $p_{max} = 30$, i.e., the same as SA, but with a different prior to fix the gauge freedom.

The last version, SK, was inspired by [38], who stressed the importance of fixing the gauge symmetry to make the rotation averaging problem better-behaved. In particular, we fixed the Karcher mean of all rotations (for any level $p$) to remain at its initial value, similar to the "inner constraints" often used in photogrammetry [34].

In Table 4, we have indicated the best performing method out of SA, SL, and SK. The specialized solvers S3, S4, and S5 that only optimize at one level are not considered in the comparison, as they are not guaranteed to converge to a global minimum, and are only there to provide insight into the relative performance in those different $SO(p)$ spaces.

| dataset | method | error | min | avg | max | success |
|---|---|---|---|---|---|---|
| statue_of_liberty_1 | SA | 0.000% | 0.038 | 0.313 | 1.211 | 100% |
| (n=19, m=54) | SL | 0.000% | 0.010 | 0.219 | 0.901 | 100% |
| | S3 | nan% | nan | nan | nan | 0% |
| | S4 | 0.001% | 0.009 | 0.016 | 0.019 | 30% |
| | S5 | 0.001% | 0.010 | 0.018 | 0.022 | 40% |
| | SK | -0.000% | 0.008 | **0.108** | 0.459 | 100% |
| natural_history_museum_london | SA | 0.000% | 0.019 | 0.036 | 0.049 | 100% |
| (n=30, m=274) | SL | 0.000% | 0.011 | **0.021** | 0.068 | 100% |
| | S3 | 0.000% | 0.010 | 0.013 | 0.015 | 60% |
| | S4 | 0.000% | 0.011 | 0.016 | 0.021 | 100% |
| | S5 | 0.000% | 0.009 | 0.014 | 0.018 | 100% |
| | SK | -0.000% | 0.021 | **0.022** | 0.024 | 100% |
| statue_of_liberty_2 | SA | 0.000% | 0.030 | 0.063 | 0.094 | 100% |
| (n=39, m=156) | SL | 0.001% | 0.011 | **0.034** | 0.060 | 100% |
| | S3 | 0.000% | 0.010 | 0.028 | 0.046 | 40% |
| | S4 | 0.000% | 0.011 | 0.024 | 0.047 | 100% |
| | S5 | 0.000% | 0.010 | 0.030 | 0.057 | 100% |
| | SK | 0.000% | 0.019 | 0.050 | 0.113 | 100% |
| taj_mahal_entrance | SA | 0.000% | 0.071 | 0.117 | 0.165 | 100% |
| (n=42, m=1272) | SL | 0.000% | 0.032 | **0.043** | 0.062 | 100% |
| | S3 | 0.000% | 0.037 | 0.046 | 0.063 | 80% |
| | S4 | 0.000% | 0.033 | 0.042 | 0.051 | 100% |
| | S5 | 0.000% | 0.032 | 0.037 | 0.039 | 100% |
| | SK | -0.000% | 0.070 | 0.081 | 0.092 | 100% |
| sistine_chapel_ceiling_1 | SA | 0.000% | 0.102 | 0.173 | 0.246 | 100% |
| (n=49, m=1754) | SL | 0.000% | 0.064 | **0.085** | 0.108 | 100% |
| | S3 | 0.000% | 0.057 | 0.073 | 0.087 | 60% |
| | S4 | 0.000% | 0.072 | 0.121 | 0.293 | 100% |
| | S5 | 0.000% | 0.064 | 0.083 | 0.095 | 100% |
| | SK | -0.000% | 0.102 | 0.116 | 0.130 | 100% |

**Table 4.** More results on YFCC datasets with $n < 50$. In this table, we compare 6 methods (see text for details). For each, we show the relative error with respect to SA, we give the minimum, average, and maximum running times (in seconds), and the fraction of cases in which the method converges to a global minimizer.

*Conclusions* For these small datasets, block-coordinate descent [13,14] performs very well. Even so, Shonan averaging with Karcher mean is faster in several cases. One such example in this Table is the Statue of Liberty dataset with $n = 39$ and $m = 156$. Interestingly, SK appears to be the fastest Shonan Averaging variant for some datasets, despite the fact that it contains an additional *dense* term in the objective that involves *all* of the poses. We conjecture that for these relatively small datasets, the inclusion of the prior on the Karcher mean helps to promote faster convergence of the manifold optimization by penalizing components of the update step that lie in the subspace of (global) gauge symmetry directions for the rotation averaging problem. Intuitively, it discourages the step from having a component that does not "actually change" the solution. This can be important in the context of a trust-region method like ours, where the *total* length of the step is restricted at each iteration. It would also be interesting to investigate what (if any) effect the inclusion of the Karcher mean has on the presence of suboptimal critical points at each level of the Riemannian Staircase, although we leave these questions for future work.

Also clear is that starting Shonan Averaging with $p = 3$, shown as SL in the table, is always either on par or much faster than SA. There is a simple explanation for this: in many instances, it is possible to recover a global minimizer from the optimization at the lowest level $p = 3$. This can be appreciated by comparing with the results of S3: it rarely finds global minima *every* time, but when it does it is obviously the fastest of all methods. In SL, we only move to the next $SO(p)$ level if that does *not* happen, and **hence we get the best of both worlds: fast convergence if we happened to pick a lucky initial estimate, and upgrade to global optimality if not**.

The S3, S4, and S5 lines are shown to indicate at what level this occurs, and for these datasets it is almost always at $p = 4$. However, the results reveal that there are indeed no guarantees, so it is not a good idea to run Shonan Averaging at a single level: the Riemmannian Staircase provides the global guarantee but at minimal extra cost, as it is *only* triggered when we converge to a suboptimal critical point at a lower level $p$.

### C.2 Intermediate-size Datasets ($50 \leq n < 150$)

In Tables 5 and 6 we show additional results on increasingly larger YFCC datasets, with exactly the same parameters as in the previous section. However, here we omit the S3-S5 variants *unless* they do not converge to global minima in *all* tested cases.

*Conclusion* From these results **it is clear that SL starts to emerge as the best among the global optimization methods in terms of average running time**. Again, we observe that for these larger datasets it is rare *not* to converge to a global minimizer at $p = 3$, which is interesting in its own right. Of course, there are some exceptions, e.g., the Big Ben dataset in Table 6 with $n = 101$ and $m = 1880$, for which global minimizers were *not* found using only local search with $p \in \{3, 4, 5\}$. The BD method is still competitive in cases where the number of measurements is large, as BD's running time is dominated by the number of images $n$, given that it is optimizing for a $3n \times 3n$ PSD matrix.

Results on these larger datasets suggest that finding global minima is actually *harder* in general for small datasets than for larger, well connected datasets. We can gain some theoretical insight into this empirical finding in light of several recent works [38,27,5] that have studied the connection between graph-theoretic properties of the measurement network $G = (n, \mathcal{E})$ that underpins the rotation averaging problem, and the statistical and geometric/computational properties of the resulting maximum-likelihood estimation (1). In a nutshell, these investigations indicate that *both* the statistical properties of the maximum-likelihood

| dataset | method | error | min | avg | max | success |
|---|---|---|---|---|---|---|
| sistine_chapel_ceiling_2 | SA | 0.000% | 0.101 | 0.162 | 0.234 | 100% |
| (n=51, m=1670) | SL | 0.000% | 0.065 | **0.115** | 0.332 | 100% |
| | S3 | 0.000% | 0.076 | 0.081 | 0.084 | 50% |
| | SK | -0.000% | 0.097 | 0.127 | 0.211 | 100% |
| milan_cathedral | SA | 0.000% | 0.175 | 0.203 | 0.220 | 100% |
| (n=69, m=2782) | SL | 0.000% | 0.078 | **0.092** | 0.105 | 100% |
| | SK | -0.000% | 0.170 | 0.190 | 0.206 | 100% |
| reichstag | SA | 0.000% | 0.166 | 0.291 | 0.399 | 100% |
| (n=71, m=2554) | SL | 0.000% | 0.073 | **0.084** | 0.108 | 100% |
| | SK | -0.000% | 0.179 | 0.193 | 0.215 | 100% |
| piazza_dei_miracoli | SA | 0.000% | 0.261 | 0.353 | 0.593 | 100% |
| (n=74, m=3456) | SL | 0.000% | 0.092 | **0.119** | 0.147 | 100% |
| | S3 | 0.000% | 0.102 | 0.119 | 0.146 | 90% |
| | SK | -0.000% | 0.260 | 0.344 | 0.413 | 100% |
| ruins_of_st_pauls | SA | 0.000% | 0.337 | 0.426 | 0.735 | 100% |
| (n=82, m=4998) | SL | 0.000% | 0.149 | **0.173** | 0.259 | 100% |
| | SK | -0.000% | 0.372 | 0.450 | 0.544 | 100% |
| mount_rushmore | SA | 0.000% | 0.264 | 0.366 | 0.511 | 100% |
| (n=83, m=4012) | SL | 0.000% | 0.105 | **0.123** | 0.193 | 100% |
| | SK | -0.000% | 0.263 | 0.306 | 0.352 | 100% |
| london_bridge_3 | SA | 0.000% | 0.113 | 0.141 | 0.243 | 100% |
| (n=88, m=1500) | SL | 0.000% | 0.050 | **0.070** | 0.095 | 100% |
| | S3 | 0.000% | 0.052 | 0.066 | 0.084 | 90% |
| | SK | -0.000% | 0.115 | 0.128 | 0.143 | 100% |
| palace_of_versailles_chapel | SA | 0.000% | 0.296 | 0.319 | 0.379 | 100% |
| (n=91, m=3964) | SL | 0.000% | 0.102 | **0.118** | 0.138 | 100% |
| | SK | -0.000% | 0.291 | 0.313 | 0.348 | 100% |
| pieta_michelangelo | SA | 0.000% | 0.540 | 0.709 | 1.051 | 100% |
| (n=93, m=7728) | SL | 0.000% | 0.194 | **0.220** | 0.228 | 100% |
| | SK | -0.000% | 0.538 | 0.559 | 0.614 | 100% |
| blue_mosque_interior_2 | SA | 0.000% | 0.169 | 0.220 | 0.348 | 100% |
| (n=95, m=2288) | SL | 0.000% | 0.075 | 0.149 | 0.510 | 100% |
| | S3 | 0.000% | 0.087 | **0.106** | 0.140 | 90% |
| | SK | -0.000% | 0.172 | 0.200 | 0.222 | 100% |
| st_vitus_cathedral | SA | 0.000% | 0.592 | 0.909 | 1.306 | 100% |
| (n=97, m=8334) | SL | 0.000% | 0.250 | **0.291** | 0.326 | 100% |
| | S3 | 0.000% | 0.252 | 0.298 | 0.387 | 90% |
| | SK | -0.000% | 0.571 | 0.650 | 0.805 | 100% |

**Table 5.** YFCC results for $50 \leq n \leq 100$. Same format as Table 4 but only showing S3-S5 results if they do not converge to global minima in *all* tested cases.

estimation (1) *and* its computational hardness are controlled by the *algebraic connectivity* $\lambda_2(L(G))$, i.e., the *second*-smallest eigenvalue of the weighted graph Laplacian $L(G)$ associated with the graph $G$. Larger values imply *both* a better (lower-uncertainty) estimate *and* that the resulting relaxation (16) is stronger. For densely-connected measurement networks (of the kind that frequently appear

| dataset | method | error | min | avg | max | success |
|---|---|---|---|---|---|---|
| big_ben_1 | SA | 0.000% | 0.196 | 4.084 | 30.563 | 100% |
| (n=101, m=1880) | SL | 0.000% | 0.080 | **0.348** | 1.285 | 100% |
| | S3 | 0.000% | 0.105 | 0.127 | 0.199 | 70% |
| | S4 | 0.000% | 0.090 | 0.118 | 0.172 | 60% |
| | S5 | 0.000% | 0.077 | 0.140 | 0.197 | 80% |
| | SK | 0.000% | 0.185 | 1.001 | 5.572 | 100% |
| london_bridge_2 | SA | 0.000% | 0.220 | 0.351 | 1.218 | 100% |
| (n=106, m=2742) | SL | 0.000% | 0.091 | **0.114** | 0.136 | 100% |
| | SK | -0.000% | 0.222 | 0.243 | 0.264 | 100% |
| palazzo_pubblico | SA | 0.000% | 0.331 | 0.373 | 0.427 | 100% |
| (n=112, m=4420) | SL | 0.000% | 0.155 | **0.262** | 0.926 | 100% |
| | SK | -0.000% | 0.340 | 0.382 | 0.442 | 100% |
| london_bridge_1 | SA | 0.000% | 0.416 | 0.472 | 0.543 | 100% |
| (n=118, m=5690) | SL | 0.000% | 0.168 | **0.203** | 0.285 | 100% |
| | SK | -0.000% | 0.423 | 0.469 | 0.536 | 100% |
| national_gallery_london | SA | 0.000% | 0.186 | 0.233 | 0.351 | 100% |
| (n=124, m=2160) | SL | 0.000% | 0.089 | **0.107** | 0.131 | 100% |
| | SK | -0.000% | 0.179 | 0.210 | 0.246 | 100% |
| lincoln_memorial | SA | 0.000% | 0.324 | 0.374 | 0.548 | 100% |
| (n=127, m=3516) | SL | 0.000% | 0.119 | **0.135** | 0.185 | 100% |
| | SK | -0.000% | 0.259 | 0.301 | 0.329 | 100% |
| grand_central_terminal_new_york | SA | 0.000% | 0.414 | 0.584 | 1.043 | 100% |
| (n=132, m=5880) | SL | 0.000% | 0.187 | **0.220** | 0.291 | 100% |
| | SK | -0.000% | 0.528 | 0.586 | 0.661 | 100% |
| paris_opera_2 | SA | 0.000% | 0.809 | 0.888 | 0.982 | 100% |
| (n=133, m=10778) | SL | 0.000% | 0.341 | **0.395** | 0.478 | 100% |
| | SK | -0.000% | 0.837 | 0.906 | 1.058 | 100% |

**Table 6.** YFCC results for $100 \leq n \leq 150$. Same layout as Table 5.

in structure-from-motion applications), it is an elementary result from algebraic graph theory that this quantity can grow at a rate of up to $O(n)$. This provides insight into the observation that problems with larger measurement networks appear easier to solve, assuming a reasonably dense set of measurements.

### C.3 Larger Datasets ($n \geq 150$)

Finally, in Table 7 we show additional results on the largest YFCC datasets with $n \geq 150$. The block-coordinate descent method from [14] did not converge in reasonable time for many of the larger datasets, which is because we use the minimum eigenvalue optimality certificate threshold $\lambda_{min}$ to establish convergence. The threshold we used in all experiments was $10^{-5}$, and for these datasets it takes a long time for BD to reach that level, in contrast to Shonan Averaging.

*Conclusion* For these large datasets convergence to the global minimizer occurs almost always at $p = 3$ and hence the SL and S3 methods are basically identical in terms of operation and performance. However, SL comes with a global guarantee: in the rare case that S3 does not converge to the global minimizer at the $SO(3)$ level, SL will simply move up to $SO(4)$ and up, and thereby still recover the true maximum likelihood estimate.

| dataset | method | error | min | avg | max | success |
|---|---|---|---|---|---|---|
| st_peters_basilica_interior_2 | SA | 0.000% | 0.948 | 1.165 | 1.738 | 100% |
| (n=173, m=11688) | SL | 0.000% | 0.557 | **0.713** | 0.897 | 100% |
| | S3 | 0.000% | 0.531 | 0.587 | 0.715 | 80% |
| pantheon_interior | SA | 0.000% | 0.852 | 1.001 | 1.570 | 100% |
| (n=186, m=10000) | SL | 0.000% | 0.449 | **0.509** | 0.623 | 100% |
| | S3 | 0.000% | 0.455 | 0.530 | 0.642 | 100% |
| florence_cathedral_dome_interior_1 | SA | 0.000% | 2.435 | 2.848 | 3.226 | 100% |
| (n=213, m=31040) | SL | 0.000% | 1.869 | **2.170** | 2.499 | 100% |
| | S3 | 0.000% | 1.856 | 2.209 | 2.814 | 100% |
| paris_opera_1 | SA | 0.000% | 3.673 | 4.145 | 4.467 | 100% |
| (n=254, m=45754) | SL | 0.000% | 1.686 | **2.024** | 2.677 | 100% |
| | S3 | 0.000% | 1.683 | 2.104 | 2.510 | 100% |
| pike_place_market | SA | 0.000% | 4.437 | 5.255 | 7.285 | 100% |
| (n=265, m=53242) | SL | 0.000% | 2.019 | **2.322** | 3.400 | 100% |
| | S3 | 0.000% | 2.122 | 2.246 | 2.384 | 100% |
| blue_mosque_interior_1 | SA | 0.000% | 3.581 | 4.153 | 4.964 | 100% |
| (n=272, m=40292) | SL | 0.000% | 2.268 | **2.708** | 2.991 | 100% |
| | S3 | 0.000% | 2.511 | 2.997 | 3.645 | 100% |
| notre_dame_rosary_window | SA | 0.000% | 8.102 | 8.652 | 10.515 | 100% |
| (n=326, m=93104) | SL | 0.000% | 3.478 | **4.074** | 4.690 | 100% |
| | S3 | 0.000% | 3.885 | 4.326 | 5.012 | 100% |
| british_museum | SA | 0.000% | 4.297 | 4.851 | 7.846 | 100% |
| (n=344, m=45450) | SL | 0.000% | 1.927 | **2.558** | 3.400 | 100% |
| | S3 | 0.000% | 2.009 | 2.558 | 3.911 | 100% |
| palace_of_westminster | SA | 0.000% | 1.334 | 1.480 | 1.692 | 100% |
| (n=345, m=11522) | SL | 0.000% | 0.598 | **0.837** | 1.242 | 100% |
| | S3 | 0.000% | 0.731 | 1.038 | 2.397 | 100% |
| louvre | SA | 0.000% | 2.872 | 3.141 | 3.709 | 100% |
| (n=367, m=26656) | SL | 0.000% | 1.606 | **2.284** | 3.265 | 100% |
| | S3 | 0.000% | 1.648 | 2.058 | 2.495 | 100% |
| st_peters_basilica_interior_1 | SA | 0.000% | 5.283 | 5.654 | 6.295 | 100% |
| (n=365, m=55024) | SL | 0.000% | 3.255 | **4.030** | 4.995 | 100% |
| | S3 | 0.000% | 3.662 | 4.169 | 4.728 | 100% |
| st_pauls_cathedral | SA | 0.000% | 7.385 | 7.861 | 8.384 | 100% |
| (n=370, m=83060) | SL | 0.000% | 3.385 | **3.792** | 4.373 | 100% |
| | S3 | 0.000% | 3.420 | 4.176 | 6.689 | 100% |
| westminster_abbey_1 | SA | 0.000% | 4.238 | 4.582 | 4.957 | 100% |
| (n=501, m=38863) | SL | 0.000% | 2.059 | **2.396** | 2.710 | 100% |
| | S3 | 0.000% | 2.209 | 2.693 | 3.385 | 100% |
| pantheon_exterior | SA | 0.000% | 6.922 | 7.876 | 11.202 | 100% |
| (n=720, m=49256) | SL | 0.000% | 3.798 | **5.247** | 6.765 | 100% |
| | S3 | 0.000% | 4.091 | 4.945 | 6.156 | 100% |

**Table 7.** Comparing SA, SL and S3 (see text) on YFCC Datasets with $n \geq 150$. In many cases S3 has a lower average computation time than SL, since it performs only local optimization at the *lowest* level $p = 3$ of the Riemannian Staircase. However, in contrast to SL, S3 has *no* guarantees regarding convergence to global minima.