

# Chordal Averaging on Flag Manifolds and Its Applications

## Supplemental Material

### Abstract

This document supplements our submission entitled Chordal Averaging on Flag Manifolds and Its Applications. In particular, we present extended proofs of our propositions, provide ablation studies and broaden the experimental evaluations in our main paper. We further include a discussion on the connection of multiple rotation averaging to our motion averaging.

### 1. Flag Representations

A flag is a nested collection of subspaces of increasing dimension. An illustration of a  $\mathcal{FL}(1, 2; 3)$  is in Fig. 1).

Flags are a natural representation for time series data as nested “time subspaces.” Suppose we data at three times:  $\mathbf{x}_{t=1}, \mathbf{x}_{t=2}, \mathbf{x}_{t=3} \in \mathbb{R}^d$ . We can group these data based on their “effect over time” in the sense that time  $t = 1$  stands alone,  $t = 1$  affects  $t = 2$ , and  $t = 1$  and  $t = 2$  affect  $t = 3$ . This grouping gives us the flag of type  $\mathcal{FL}(1, 2, 3 : d)$ :

$$\text{span}\{\mathbf{x}_1\} \subset \text{span}\{\mathbf{x}_1, \mathbf{x}_2\} \subset \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\} \subset \mathbb{R}^d. \quad (1)$$

Flags can also model some hierarchical data using hierarchically nested subspaces. For a nice list of flags in mathematics, see [25].

Recall  $\mathcal{FL}(d+1) = \mathcal{FL}(d_1, d_2, \dots, d_k; d_{k+1} = d)$ , we take  $m_1 = 1$  and  $m_j = d_j - d_{j-1}$ . There are number of representations for flag manifolds involving quotients [25]. We mention the most popular representation in the manuscript. A number works [20, 21, 25] use

$$\frac{SO(d)}{S(O(m_1) \times O(m_2) \times \dots \times O(m_{k+1}))} \quad (2)$$

where  $S(O(m_1) \times \dots \times O(m_k))$  is

$$\{(\mathbf{M}_1, \dots, \mathbf{M}_k) : \prod_{i=1}^k \det(\mathbf{M}_i) = 1\}. \quad (3)$$

Other works [24, 23] represent flag manifolds using the quotient

$$\frac{St(d_k, d)}{O(m_1) \times O(m_2) \times \dots \times O(m_k)}. \quad (4)$$

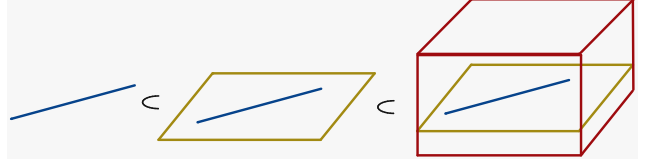


Figure 1: Illustration of a nested sequence of subspaces corresponding to a point on the flag manifold.

In this representation,  $\mathbf{X} \in St(d_k, d)$  is used to represent the equivalence class

$$[\mathbf{X}] = \{\mathbf{X}\mathbf{O} : \mathbf{O}_i \in O(m_i)\} \in \mathcal{FL}(d+1)$$

where  $\mathbf{O} = \text{diag}(\mathbf{O}_{m_1}, \dots, \mathbf{O}_{m_k})$ . We use this Stiefel quotient representation in this manuscript.

Ye *et al.* prove that  $\mathcal{FL}(d+1)$  is diffeomorphic to Eqs. 2 and 4 (see Prop. 4 and 12 in [25]). Additionally, Ye *et al.* prove that flags are a closed submanifold of

$$\text{Gr}(m_1, d) \times \text{Gr}(m_2, d) \times \dots \times \text{Gr}(m_k, d). \quad (5)$$

Our chordal distance on flag manifolds leverages this product-of-Grassmannians since it is the 2-norm of the chordal distances between each  $\text{Gr}(m_1, d)$ .

### 2. Proof of Proposition I

Before providing the full proof, let us recall Prop. I:

**Proposition 1.** The chordal flag-mean of  $\{[\mathbf{X}_i]\}_{i=1}^p \subset \mathcal{FL}(d+1)$  is

$$[\boldsymbol{\mu}] := \arg \min_{[\mathbf{Y}] \in \mathcal{FL}(d+1)} \sum_{i=1}^p \alpha_i d_c([\mathbf{X}^{(i)}], [\mathbf{Y}])^2 \quad (6)$$

and can be phrased into a Stiefel manifold optimization problem as

$$[\boldsymbol{\mu}] = \arg \min_{\mathbf{Y} \in St(d_k, d)} \sum_{j=1}^k m_j - \text{tr}(\mathbf{I}_j \mathbf{Y}^\top \mathbf{P}_j \mathbf{Y}) \quad (7)$$

where the matrices  $\mathbf{I}_j$  and  $\mathbf{P}_j$  are given in Eq. 8 and Eq. 9

respectively.

$$(\mathbf{I}_j)_{i,l} = \begin{cases} 1, & i = l \in \{d_{j-1} + 1, d_{j-1} + 2, \dots, d_j\} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

and define:

$$\mathbf{P}_j = \sum_{i=1}^p \alpha_j \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \quad (9)$$

For example, if we are averaging on  $\mathcal{FL}(1, 3; 4)$  we have

$$\mathbf{I}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{I}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

*Proof.* We begin by realizing Eq. 6 as an optimization problem using the definition of chordal distance:

$$\arg \min_{\mathbf{Y} \in \mathcal{FL}(d+1)} \sum_{i=1}^p \alpha_i \left( \sum_{j=1}^k m_j - \text{tr} \left( \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \right) \right).$$

Then we move our summations around to simplify our objective function:

$$\begin{aligned} & \sum_{i=1}^p \alpha_i \left( \sum_{j=1}^k m_j - \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right) \right) \\ &= \sum_{j=1}^k \left( \sum_{i=1}^p \alpha_i \right) m_j - \sum_{i=1}^p \alpha_i \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right), \\ &= \sum_{j=1}^k \left( \sum_{i=1}^p \alpha_i \right) m_j - \sum_{j=1}^k \sum_{i=1}^p \alpha_i \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right). \end{aligned}$$

Since  $\sum_{i=1}^p \alpha_i$  is constant with respect to  $\mathbf{Y}$ , Eq. 6 is equivalent to

$$\arg \min_{\mathbf{Y} \in \mathcal{FL}(d+1)} \sum_{j=1}^k m_j - \sum_{j=1}^k \sum_{i=1}^p \alpha_i \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right).$$

Using our definitions for  $\mathbf{I}_j$  and  $\mathbf{P}_j$ , we can write the objective function in terms of  $\mathbf{Y}$ :

$$\begin{aligned} & \sum_{j=1}^k m_j - \text{tr} \left( \mathbf{Y}_j^\top \left( \sum_{i=1}^p \alpha_i \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \right) \mathbf{Y}_j \right), \\ &= \sum_{j=1}^k m_j - \text{tr} \left( \mathbf{Y}_j^\top \mathbf{P}_j \mathbf{Y}_j \right), \\ &= \sum_{j=1}^k m_j - \text{tr} \left( \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{P}_j \right), \\ &= \sum_{j=1}^k m_j - \text{tr} \left( \mathbf{Y} \mathbf{I}_j \mathbf{Y}^\top \mathbf{P}_j \right). \end{aligned}$$

The third equality is true because  $\mathbf{Y}_j \mathbf{Y}_j^\top = \mathbf{Y} \mathbf{I}_j \mathbf{Y}^\top$ .

There are two constraints for  $\mathbf{Y} \in \mathcal{FL}(d+1)$  according to our representation for points on the flag manifold. The first constraint is  $\mathbf{Y}_j^\top \mathbf{Y}_j = \mathbf{I}$  for  $j = 1, 2, \dots, p$ . The second constraint is  $[\mathbf{Y}_j] \cap [\mathbf{Y}_i] = \emptyset$  for all  $i \neq j$ . These constraints are satisfied when  $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}$ , e.g.  $\mathbf{Y} \in St(d_k, d)$ .

Using trace invariance to cyclic permutations, the chordal flag mean optimization problem Eq. 6 is equivalent to the Stiefel optimization problem Eq. 7.  $\square$

### 3. Proof of Proposition III

**Proposition 2.** *The chordal flag-median of  $\{[\mathbf{X}_i]\}_{i=1}^p \subset \mathcal{FL}(d+1)$ ,*

$$[\boldsymbol{\eta}] = \arg \min_{\mathbf{Y} \in \mathcal{FL}(d+1)} \sum_{i=1}^p \alpha_i d_c([\mathbf{X}^{(i)}], [\mathbf{Y}]), \quad (10)$$

can be phrased with weights

$$w_i([\mathbf{Y}]) = \sum_{j=1}^k \frac{\alpha_i}{\max\{d_c([\mathbf{X}^{(i)}], [\mathbf{Y}]), \epsilon\}}$$

as the optimization problem

$$\arg \min_{\mathbf{Y} \in \mathcal{FL}(d+1)} \sum_{i=1}^p \sum_{j=1}^k m_j - w_i([\mathbf{Y}]) \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right)$$

with  $\epsilon = 0$  as long as  $d_c([\mathbf{X}^{(i)}], [\mathbf{Y}]) \neq 0$  for all  $i$ .

*Proof.* We can write Eq. 10 using the definition of chordal distance as

$$\arg \min_{\mathbf{Y} \in \mathcal{FL}(d+1)} \sum_{i=1}^p \alpha_i \sqrt{\sum_{j=1}^k m_j - \text{tr} \left( \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \right)}.$$

The orthogonality constraints for  $\mathbf{Y} \in \mathbb{R}^{d \times dk}$  to represent a point on  $\mathcal{FL}(d+1)$  are: (i)  $[\mathbf{Y}_j] \cap [\mathbf{Y}_i] = \emptyset$  for all  $i \neq j$  and  $\mathbf{Y}_j^\top \mathbf{Y}_j = \mathbf{I}$  for all  $j$ . Let  $\theta([\mathbf{Y}_i], [\mathbf{Y}_j])$  denote the vector of principal angles between  $[\mathbf{Y}_i]$  and  $[\mathbf{Y}_j]$ . Using  $\text{tr}(\mathbf{Y}_i^\top \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_i) = \|\cos \theta([\mathbf{Y}_i], [\mathbf{Y}_j])\|_2^2$ , we encode our orthogonality constraints as

$$\text{tr}(\mathbf{Y}_i^\top \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_i) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}.$$

We will now use

$$\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases}$$

to put these constraints into the Lagrangian.

Let  $\Lambda$  be a symmetric matrix of Lagrange multipliers corresponding to the orthogonality constraints. Denote the entry in the  $i$ th row and  $j$ th column of  $\Lambda$  as  $\lambda_{i,j}$ . With the constraints added to the objective, we define the Lagrangian in Eq. 11.

$$\begin{aligned} \mathcal{L}(\mathbf{Y}, \Lambda) &= \sum_{i=1}^p \alpha_i \sqrt{\sum_{j=1}^k m_j - \text{tr} \left( \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \right)} \\ &\quad - \sum_{i=j}^k \sum_{j=1}^k \lambda_{i,j} (m_j \delta_{i,j} - \text{tr}(\mathbf{Y}_i^\top \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_i)). \end{aligned} \quad (11)$$

The gradient of Eq. 11 w.r.t.  $\mathbf{Y}_j$  and  $\lambda_{i,j}$  is

$$\begin{aligned} \nabla_{\mathbf{Y}_j} \mathcal{L} &= - \sum_{i=1}^p \frac{\alpha_i \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j}{\sqrt{\sum_{j=1}^k m_j - \text{tr} \left( \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \right)}} \\ &\quad + 2 \sum_{\substack{i=1 \\ i \neq j}}^k \lambda_{i,j} \mathbf{Y}_i \mathbf{Y}_i^\top \mathbf{Y}_j + 4 \lambda_{j,j} \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_j, \\ \nabla_{\lambda_{i,j}} \mathcal{L} &= m_j \delta_{i,j} - \text{tr} \left( \mathbf{Y}_i^\top \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_i \right). \end{aligned}$$

Notice we are not dividing by zero because  $d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Y} \rrbracket) \neq 0$  for all  $i$ . Now we use  $\nabla_{\mathbf{Y}_j} \mathcal{L} = \mathbf{0}$  and  $\nabla_{\lambda_{i,j}} \mathcal{L} = 0$  to solve for  $\lambda_{j,j}$ .

First we will work with  $\nabla_{\mathbf{Y}_j} \mathcal{L} = \mathbf{0}$ .

$$\begin{aligned} \mathbf{0} &= - \sum_{i=1}^p \frac{\alpha_i \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j}{d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Y} \rrbracket)} \\ &\quad + 2 \sum_{\substack{i=1 \\ i \neq j}}^k \lambda_{i,j} \mathbf{Y}_i \mathbf{Y}_i^\top \mathbf{Y}_j + 4 \lambda_{j,j} \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_j, \\ &= - \sum_{i=1}^p \frac{\alpha_i \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j}{d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Y} \rrbracket)} \\ &\quad + 2 \sum_{\substack{i=1 \\ i \neq j}}^k \lambda_{i,j} \mathbf{Y}_j^\top \mathbf{Y}_i \mathbf{Y}_i^\top \mathbf{Y}_j + 4 \lambda_{j,j} \mathbf{Y}_j^\top \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_j, \\ &= - \sum_{i=1}^p \frac{\alpha_i \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right)}{d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Y} \rrbracket)} \\ &\quad + 2 \sum_{\substack{i=1 \\ i \neq j}}^k \lambda_{i,j} \text{tr} \left( \mathbf{Y}_j^\top \mathbf{Y}_i \mathbf{Y}_i^\top \mathbf{Y}_j \right) + 4 \lambda_{j,j} \text{tr} \left( \mathbf{Y}_j^\top \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_j \right). \end{aligned}$$

Using  $\nabla_{\lambda_{i,j}} \mathcal{L} = 0$  simplifies our equation to

$$\begin{aligned} 4 \lambda_{j,j} \text{tr} \left( \mathbf{Y}_j^\top \mathbf{Y}_j \mathbf{Y}_j^\top \mathbf{Y}_j \right) &= \sum_{i=1}^p \frac{\alpha_i \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right)}{d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Y} \rrbracket)}, \\ 4 m_j \lambda_{j,j} &= \sum_{i=1}^p \frac{\alpha_i \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right)}{d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Y} \rrbracket)}. \end{aligned}$$

For  $\llbracket \mathbf{Y} \rrbracket$  to minimize Eq. 10, we would want to maximize  $m_j \lambda_{j,j}$  for each  $j$ . That is to say, we wish to maximize  $\sum_{j=1}^k m_j \lambda_{j,j}$ :

$$\sum_{i=1}^p \sum_{j=1}^k \frac{\alpha_i}{d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Y} \rrbracket)} \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right). \quad (12)$$

Maximizing Eq. 12 is the same as minimizing

$$\sum_{i=1}^p \sum_{j=1}^k m_j - \frac{\alpha_i}{d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Y} \rrbracket)} \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right).$$

Using the definition of  $w_i(\llbracket \mathbf{Y} \rrbracket)$ , this minimization is

$$\arg \min_{\llbracket \mathbf{Y} \rrbracket \in \mathcal{FL}(d+1)} \sum_{i=1}^p \sum_{j=1}^k m_j - w_i(\llbracket \mathbf{Y} \rrbracket) \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right)$$

□

**Proposition 3.** Fix  $\llbracket \mathbf{Z} \rrbracket \in \mathcal{FL}(d+1)$ . Then the minimizer of

$$\sum_{i=1}^p \sum_{j=1}^k \left( m_j - w_i(\mathbf{Z}) \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right) \right) \quad (13)$$

over  $\llbracket \mathbf{Y} \rrbracket \in \mathcal{FL}(d+1)$  is the weighted chordal flag mean of  $\{\llbracket \mathbf{X}^{(i)} \rrbracket\}_{i=1}^p \in \mathcal{FL}(d+1)$  with weights  $w_i(\mathbf{Z})$ . Note:  $\epsilon = 0$  as long as  $d_c(\llbracket \mathbf{X}^{(i)} \rrbracket, \llbracket \mathbf{Z} \rrbracket) \neq 0$  for all  $i$ .

*Proof.* By re-arranging the summations in Eq. 13, we see its minimizer is also

$$\arg \min_{\llbracket \mathbf{Y} \rrbracket \in \mathcal{FL}(d+1)} \sum_{j=1}^k m_j - \sum_{j=1}^k \sum_{i=1}^p w_i(\mathbf{Z}) \text{tr} \left( \mathbf{Y}_j^\top \mathbf{X}_j^{(i)} \mathbf{X}_j^{(i)\top} \mathbf{Y}_j \right).$$

We showed that this is the same as the chordal flag-mean optimization problem with weights  $w_i(\mathbf{Z})$  in the proof of Prop. 1. □

## 4. Proof of Proposition VI

**Proposition 4.** Let  $[\mathbf{Y}] \in \mathcal{FL}(d+1)$  and  $\epsilon > 0$ . Assume that  $d([\mathbf{Y}], [\mathbf{X}^{(i)}]) > \epsilon$  for  $i = 1, 2, \dots, p$ . Denote the flag median objective function value as  $f : \mathcal{FL}(d+1) \rightarrow \mathbb{R}$  and an iteration of our chordal flag-median IRLS algorithm as  $T : \mathcal{FL}(d+1) \rightarrow \mathcal{FL}(d+1)$ . Then

$$f(T([\mathbf{Y}])) \leq f([\mathbf{Y}]).$$

*Proof.* Assuming that  $d([\mathbf{Y}], [\mathbf{X}^{(i)}]) > \epsilon$  for  $i = 1, 2, \dots, p$ , we define the function  $h : \mathcal{FL}(d+1) \times \mathcal{FL}(d+1) \rightarrow \mathbb{R}$  as

$$\begin{aligned} h([\mathbf{Z}], [\mathbf{Y}]) &= \sum_{i=1}^p w_i([\mathbf{Y}]) d_c([\mathbf{Z}], [\mathbf{X}^{(i)}])^2, \\ w_i([\mathbf{Y}]) &= \frac{1}{\max\{d_c([\mathbf{Y}], [\mathbf{X}^{(i)}]), \epsilon\}} \\ &= \frac{1}{d_c([\mathbf{Y}], [\mathbf{X}^{(i)}])}. \end{aligned}$$

Some algebra reduces  $h([\mathbf{Z}], [\mathbf{Y}])$  to

$$\begin{aligned} h([\mathbf{Z}], [\mathbf{Y}]) &= \sum_{i=1}^p w_i([\mathbf{Y}]) d_c([\mathbf{Z}], [\mathbf{X}^{(i)}])^2, \\ &= \sum_{i=1}^p \frac{d_c([\mathbf{Z}], [\mathbf{X}^{(i)}])^2}{d_c([\mathbf{Y}], [\mathbf{X}^{(i)}])}. \end{aligned}$$

$h([\mathbf{Z}], [\mathbf{Y}])$  is the weighted flag-mean objective function (of  $\{[\mathbf{X}^{(i)}]\}_i$  with weights  $w_i([\mathbf{Y}])$ ). So minimizing  $h([\mathbf{Z}], [\mathbf{Y}])$  over  $[\mathbf{Z}]$  is an iteration of our IRLS algorithm to compute the flag-median. In other words,

$$T([\mathbf{Y}]) = \arg \min_{[\mathbf{Z}] \in \mathcal{FL}(d+1)} h([\mathbf{Z}], [\mathbf{Y}]). \quad (14)$$

Using Eq. 14, we have

$$h(T([\mathbf{Y}]), [\mathbf{Y}]) \leq h([\mathbf{Y}], [\mathbf{Y}]).$$

By the definition of  $h$

$$\begin{aligned} h([\mathbf{Y}], [\mathbf{Y}]) &= \sum_{i=1}^p \frac{d_c([\mathbf{Y}], [\mathbf{X}^{(i)}])^2}{d_c([\mathbf{Y}], [\mathbf{X}^{(i)}])}, \\ &= \sum_{i=1}^p d_c([\mathbf{Y}], [\mathbf{X}^{(i)}]), \\ &= f([\mathbf{Y}]). \end{aligned}$$

This means, we have

$$h(T([\mathbf{Y}]), [\mathbf{Y}]) \leq f([\mathbf{Y}]). \quad (15)$$



Figure 2: Averaging a collection of faces belonging to three different identities, captured under varying illumination: center, left and right. Notice that the first dimension of the flag representations is center illuminated, better representing the mean compared to Grassmannian.

Now we use the identity from algebra:  $\frac{a^2}{b} \geq 2a - b$  for any  $a, b \in \mathbb{R}$  and  $b > 0$ . Let

$$a = d_c([\mathbf{Z}], [\mathbf{X}^{(i)}]) \text{ and } b = d_c([\mathbf{Y}], [\mathbf{X}^{(i)}]).$$

Then

$$\begin{aligned} h([\mathbf{Z}], [\mathbf{Y}]) &\geq 2 \sum_{i=1}^p d_c([\mathbf{Z}], [\mathbf{X}^{(i)}]) \\ &\quad - \sum_{i=1}^p d_c([\mathbf{Y}], [\mathbf{X}^{(i)}]), \\ &= 2f([\mathbf{Z}]) - f([\mathbf{Y}]). \end{aligned}$$

Now, take  $[\mathbf{Z}] = T([\mathbf{Y}])$ . This gives us

$$h(T([\mathbf{Y}]), [\mathbf{Y}]) \geq 2f(T([\mathbf{Y}])) - f([\mathbf{Y}]). \quad (16)$$

Then, combining Eq. 16 with Eq. 15, we have

$$\begin{aligned} 2f(T([\mathbf{Y}])) - f([\mathbf{Y}]) &\leq f([\mathbf{Y}]), \\ f(T([\mathbf{Y}])) &\leq f([\mathbf{Y}]). \end{aligned}$$

□

## 5. Further Experimental Evaluation

### 5.1. Further Qualitative Results on Faces Dataset

We now show in Fig. 2 further visualizations of Flag and Grassmann averages of faces.

### 5.2. Further Qualitative Results on MNIST

We use 20 examples (e.g., points on  $\mathcal{FL}(1, 2; 748)$ ) of 6s and add 10 examples of 7s. We use the same workflow from the manuscript to represent the MNIST digits on  $\text{Gr}(2, 748)$  and  $\mathcal{FL}(1, 2; 748)$ . We compute the averages on the Grassmannian [10, 22] and flag (ours). The reshaped first dimension of each of these averages is in Fig. 3. The brightness

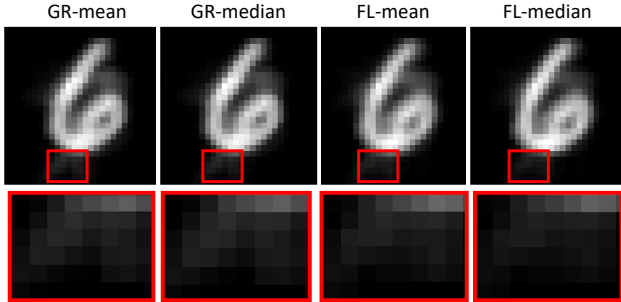


Figure 3: The first dimension of Grassmannian (“GR-”) and flag (“FL-”) averages of a data set with 20 representations of 6s and 10 representations of 7s. The bottom red boxes are the enlarged version of the upper image. Our flag-median is the least affected by the outlier examples of 7s.

of the bottom left corner of each image is brighter the more present the 7s digit (outlier class) is in the image. Notice the bottom left corner of each image, boxed in red, becomes darker as we move from left to right. So, our averaging on the flag is more robust to outliers than Grassmannian averaging. In fact, the bottom left corner of the flag-median is the darkest. Therefore, our flag-median is the least affected by the outlier examples of 7s.

### 5.3. LBG Clustering on UFC YouTube

We use a subset of the UCF YouTube Action dataset [19] to run a similar experiment to what was done by Mankovich *et al.* [22]. The dataset contains labeled RGB video clips of people performing actions. Within each labeled action, the videos are grouped into subsets of clips with common features. We take approximately one example from each subset from each class. This results in 23 examples of basketball shooting, 23 of biking/cycling, 25 of diving, 25 of golf swinging, 24 of horse back riding, 25 of soccer juggling, 23 of swinging, 24 of tennis swinging, 24 of trampoline jumping, 24 of volleyball spiking, and 24 of walking with a dog. We convert these frames to gray scale, then we use INTER\_AREA interpolation from the OpenCV package [6] to resize the frames to have only 450 pixels. This is, on average, only 1% of the number of pixels in the original frame. We vectorize and horizontally stack each video, then use the first 10 columns of  $\mathbf{Q}$  from the QR decomposition to realize each video as a point on  $\text{Gr}(10, 450)$  and  $\mathcal{FL}(1, 2, \dots, 10; 450)$ .

We run Linde-Buzo-Gray (LBG) clustering on these videos and report the resulting cluster purities in Fig. 4. Clustering on the flag manifold with our flag averages (blue boxes) improves cluster purities over Grassmannian methods. We also see higher variance in cluster purities for flag methods. Even though we are only working with approxi-

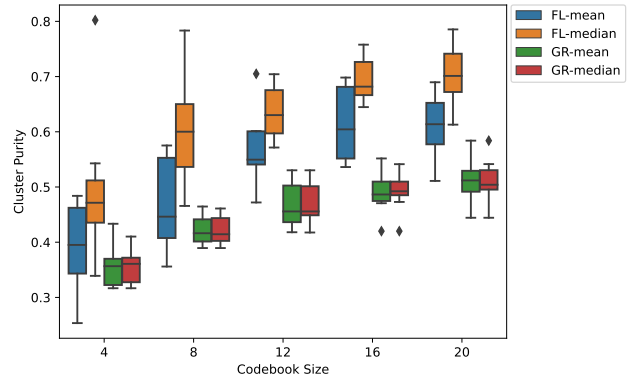


Figure 4: LBG cluster purities of YouTube videos with 10 experiments with different numbers of centers, codebook sizes. The Grassmannian, *e.g.* “GR-”, boxes are results from LBG clustering using chordal distance and Grassmannian averages from [10, 22]. The “FL-” boxes are results from using the flag chordal distance and our flag-mean and -median.

mately 1% of the total number of pixels in each frame, we are able to produce cluster purities that are competitive with those reported in [22] using a similar set of videos. Specifically, our flag-LBG clustering is well within 0.1 of the highest cluster purities reported in [22]. Overall, our flag methods improve cluster purities in a head-to-head experiment while remaining competitive with Grassmannian LBG with only using approximately 1% of pixels per frame.

### 5.4. Ablation Studies

**Robustness to initialization.** For Fig. 5, we fix a single-cluster dataset of 100 points on  $\mathcal{FL}(1, 2, 3; 10)$  then run our IRLS algorithm for the flag-median and Stiefel RTR [1, 5] for the flag-mean with initial points that are further and further away from the center of the dataset. Our dataset is computed the same way we compute synthetic datasets for the manuscript: compute a center,  $\llbracket \mathbf{C} \rrbracket \in \mathcal{FL}(1, 2, 3; 10)$ , and then add noise to the center using the parameter  $\delta$ . For this experiment we use  $\delta = .2$ . Our initial point for our IRLS algorithm and RTR is computed as the first 3 columns of the QR decomposition of  $\mathbf{C} + \mathbf{Z}\delta_{init}$  where  $\mathbf{Z} \in \mathbb{R}^{10 \times 3}$  has entries sampled from  $\mathcal{U}[-.5, .5)$ . We call  $\delta_{init}$  the noise added to the initial point and plot it on the  $x$ -axis of Fig. 5. The “Error” is the chordal distance on  $\mathcal{FL}(1, 2, 3; 10)$  between the center and the algorithm output. “Iterations” is the number of iterations of RTR for the flag-mean and IRLS for the flag-median until convergence. “Cost” is the objective function values of the algorithm output. Our IRLS algorithm estimates the flag-median is further away from the center,  $\llbracket \mathbf{C} \rrbracket$  than the flag-mean estimate. Also, the number of iterations of Stiefel RTR increases as we move the center further away from our dataset whereas our IRLS algorithm num-

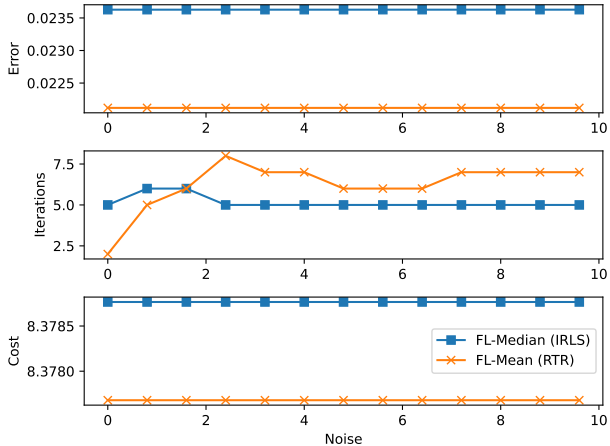


Figure 5: A plot of the robustness of our IRLS algorithm for the flag-median and Stiefel RTR for the flag-mean to initialization. For the median, we report the IRLS-iterations whereas for the mean, we report the RTR-iterations. Note that, even in large noise variances, both of the algorithms converge to a reasonable point regardless of initialization.

ber of iterations remains constant. Finally, the cost value for the flag-median estimate is higher than the flag-mean estimate because the flag-mean estimate objective function likely contains squares of values less than 1.

### Computation time.

We conducted the further experiments with ambient dimension and “dimension gap” and plot the runtime of Alg. 1 (FL-Mean) and Alg. 2 (FL-Median) in Fig. 6. As shown, the runtime increases linearly with dimension and decreases linearly with dimension gap. The FL-Mean is less affected than the FL-Median when changing  $d$  or  $d - k$ . For high  $d$ , the runtime for the FL-Median is unstable with high standard deviation and high changes in the mean runtime across small changes in  $d$ . In contrast, the FL-Mean is relatively stable in run-time to increasing  $d$ . When we vary dimension gap ( $d - k$ ), there is a negligible standard deviation in runtime for the FL-Mean and FL-Median. The FL-Median algorithm is very slow for low  $d - k$  and as fast as the FL-Mean for high  $d - k$ . The FL-Mean algorithm runtime is more stable to changes in dimension gap than the FL-Median.

### 5.5. Motion Averaging

**On error metrics.** We score the quality of our averages using the geodesic distance on the pose manifold  $SE(3)$  (or equivalently the geodesic distance on dual quaternions):

$$\epsilon(\mathbf{T}_1, \mathbf{T}_2) = \frac{1}{\pi} \|\log(\mathbf{R}_1^\top \mathbf{R}_2)\|_2 + \lambda_T \|\mathbf{t}_1 - \mathbf{t}_2\|_2 \quad (17)$$

where  $(\mathbf{R}_i, \mathbf{t}_i)$  are extracted from  $\mathbf{T}_i$  as rotational and trans-

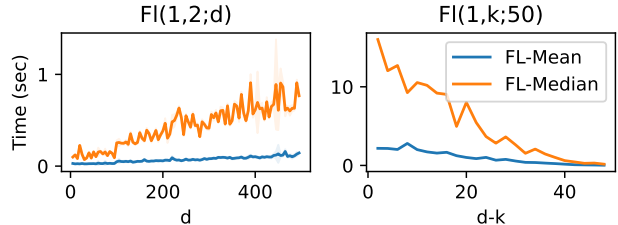


Figure 6: Time to compute the chordal flag-mean and -median of 10 points over 20 random trials. The shaded region is the standard deviation. We vary  $d$  in  $\mathcal{FL}(1, 2; d)$  (left) and vary  $d - k$  in  $\mathcal{FL}(1, k; d = 50)$  (right).

lational components, respectively.  $\lambda_T$  is a scene dependent strictly positive scaling factor. Note that, as discussed in the paper, this is very much related to the  $\lambda$  used in motion contraction.  $\log(\cdot) : SO(3) \rightarrow \mathfrak{so}(3)$  denotes the *logarithmic map* of the  $SO(3)$ -manifold. As such, this residual defined in Fig. 17 is equivalent to:

$$\epsilon(\mathbf{T}_1, \mathbf{T}_2) = \frac{1}{\pi} \arccos \left( \text{tr} \frac{\mathbf{R}_1^\top \mathbf{R}_2 - 1}{2} \right) + \lambda_s \|\mathbf{t}_1 - \mathbf{t}_2\|_2.$$

### Single rotation averaging.

Single rotation averaging where a set of rotation matrices are averaged, is a special case of motion averaging where the translational components are set to zero. Due to the compactness of the manifold, additional  $SO(3)$ -specific averaging algorithms can be employed for the case of pure rotations. To compare our method against a larger class of well established, rotation-specific averaging algorithms we opt for zeroing the translational components, performing averages and reporting only the angular errors. Figs. 7,8 present our results with increasing noise and increasing outliers respectively. For the case of outliers, we further include the recent robust methods of Rie & Civera [17]. *Naive* refers to the Euclidean averages of rotation matrices (with a subsequent projection).

**Impact of  $\lambda$ .** As we have discussed in the paper, the scene-dependent scaling  $\lambda$  is a hyper-parameter in our  $SE(3)$ -averaging. Note that, other distance metrics such as the ones dependent on 3D point distances exist [7]. These metrics exploit the action of 3D transformations on an auxiliary point set to measure distances in the 3D configuration of points. However, even those are somehow dependent upon a hyper-parameter such as the diameter of the point set or the point configurations. This is why we evaluate the impact of  $\lambda$  in our averaging. In particular, we design multiple experiments to average 250 random points on  $SE(3)$  generated with an angular noise level of 0.075. The radius of this scene is set to 1, up to a translational noise level of 0.15. This also means that the optimal  $\lambda^*$  (unknown during test) is 1. We then vary  $\lambda \in [0.002, 0.025, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.8, 2, 2.25, 2.5]$  and average 250 random points, over 50 runs. In each run, the point sets differ randomly. We compute the errors



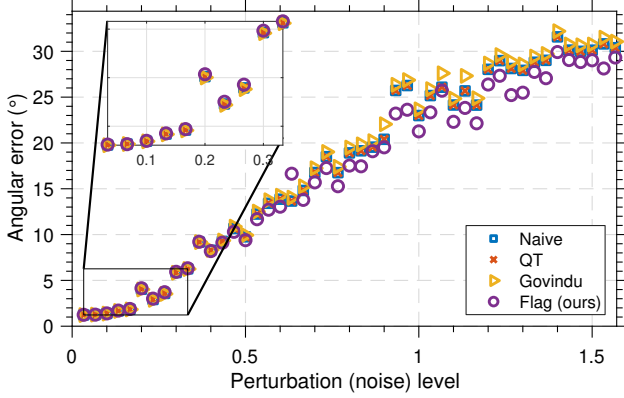


Figure 7: Single rotation averaging results for increasing levels of axial noise on synthetic, outlier-free data.

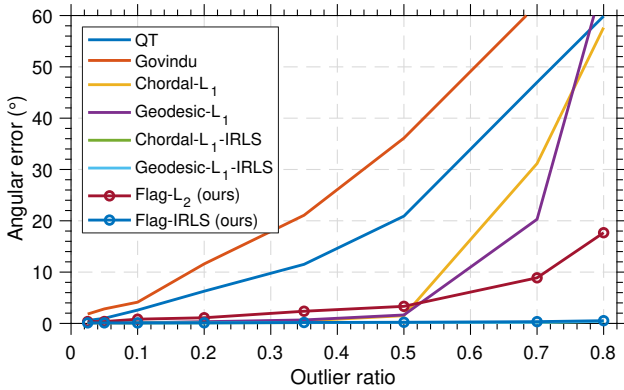


Figure 8: Single rotation averaging results for increasing levels of outliers on synthetic data with  $< 5^\circ$  of noise.

using Eq. 18 with  $\lambda = 1$  and accumulate them over all runs. Fig. 9 plots the average errors per each  $\lambda$ . In this outlier-free regime, our flag-mean and flag-median are almost aligned when  $\lambda = 1$ . Flag-median shows slight advantage over the mean for smaller values of  $\lambda$ .

## 6. On Motion & Rotation Averaging

Motion averaging lies at the heart of structure from motion and 3D reconstruction in multi-view settings. Typically, the problem of recovering individual motions for a set of cameras when we are given a number of relative motion estimates between camera pairs is known as *multiple motion averaging* or *transformation / motion synchronization* [16, 2, 4, 9, 11, 8, 12, 13]. This problem is foundational for 3D structure recovery. Synchronization algorithms usually solve multiple *single averaging* sub-problems robustly [3, 18, 14], hence the name multiple motion averaging. These sub-problems involving the computation of a robust-barycenter of a set of points on  $SE(3)$ , are commonly known as *robust single motion averaging* and

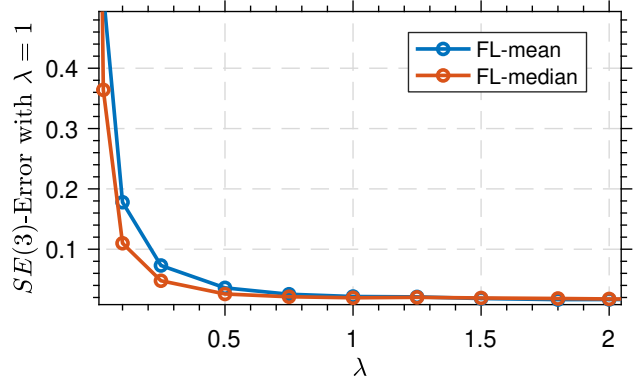


Figure 9: Impact of  $\lambda$  on our flag-mean and flag-median. During data generation we use  $\lambda^* = 1$ , whereas our algorithms use varying  $\lambda$  as plotted in the  $x$ -axis. We then compare the resulting averages to the ground truth average and report the deviation. This is an outlier-free regime and as expected, median & mean prototypes overlap when we are at the optimal value,  $\lambda = 1$ . Though, we also see that our algorithms are not too sensitive to the exact choice of this parameter.

is the focus of our paper. Although our method directly operates on the product manifold, it is a de-facto standard to decompose the problem into *single translation averaging* and *single rotation averaging*. The latter is particularly well studied due to the interesting mathematical structure of the problem [15, 14, 17, 13]. Nevertheless, our method is general enough to solve all of these variants, as we have experimented with in Figs. 7,8.

## References

- [1] P-A Absil, Christopher G Baker, and Kyle A Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007. 5
- [2] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral synchronization of multiple views in SE(3). *SIAM Journal on Imaging Sciences*, 9(4):1963–1990, 2016. 7
- [3] Tolga Birdal, Michael Arbel, Umut Simsekli, and Leonidas J Guibas. Synchronizing probability measures on rotations via optimal transport. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 7
- [4] Tolga Birdal, Umut Simsekli, Mustafa Onur Eken, and Slobodan Ilic. Bayesian pose graph optimization via Bingham distributions and tempered geodesic MCMC. *Advances in Neural Information Processing Systems*, 31, 2018. 7
- [5] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014. 5
- [6] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000. 5
- [7] Romain Brégier, Frédéric Devernay, Laetitia Leyrit, and James L Crowley. Defining the pose of any 3d rigid object and an associated distance. *International Journal of Computer Vision*, 126(6):571–596, 2018. 6
- [8] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2017. 7
- [9] Frank Dellaert, David M Rosen, Jing Wu, Robert Mahony, and Luca Carlone. Shonan rotation averaging: global optimality by surfing  $SO(p)^n$ . In *European Conference on Computer Vision*, pages 292–308. Springer, 2020. 7
- [10] Bruce Draper, Michael Kirby, Justin Marks, Tim Marrinan, and Chris Peterson. A flag representation for finite collections of subspaces of mixed dimensions. *Linear Algebra and its Applications*, 451:15–32, 2014. 4, 5
- [11] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging with the chordal distance: Global minimizers and strong duality. *IEEE Transactions of Pattern Analysis & Machine Intelligence (T-PAMI)*, 2019. 7
- [12] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–II. IEEE, 2001. 7
- [13] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004., volume 1, pages I–I. IEEE, 2004. 7
- [14] Richard Hartley, Khurram Aftab, and Jochen Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011. 7
- [15] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013. 7
- [16] Xiangru Huang, Zhenxiao Liang, Xiaowei Zhou, Yao Xie, Leonidas J Guibas, and Qixing Huang. Learning transformation synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8082–8091, 2019. 7
- [17] Seong Hun Lee and Javier Civera. Robust single rotation averaging. *arXiv preprint arXiv:2004.00732*, 2020. 6, 7
- [18] Xinyi Li and Haibin Ling. Hybrid camera pose estimation with online partitioning for slam. *IEEE Robotics and Automation Letters*, 5(2):1453–1460, 2020. 7
- [19] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos “in the wild”. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003. IEEE, 2009. 5
- [20] Xiaofeng Ma, Michael Kirby, and Chris Peterson. The flag manifold as a tool for analyzing and comparing sets of data sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4185–4194, 2021. 1
- [21] Xiaofeng Ma, Michael Kirby, and Chris Peterson. Self-organizing mappings on the flag manifold with applications to hyper-spectral image data analysis. *Neural Computing and Applications*, 34(1):39–49, 2022. 1
- [22] Nathan Mankovich, Emily J King, Chris Peterson, and Michael Kirby. The flag median and FlagIRLS. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10339–10347, 2022. 4, 5
- [23] Du Nguyen. Closed-form geodesics and optimization for Riemannian logarithms of Stiefel and flag manifolds. *Journal of Optimization Theory and Applications*, pages 1–25, 2022. 1
- [24] Renaud-Alexandre Pitaval and Olav Tirkkonen. Flag orbit codes and their expansion to Stiefel codes. In *2013 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2013. 1
- [25] Ke Ye, Ken Sze-Wai Wong, and Lek-Heng Lim. Optimization on flag manifolds. *Mathematical Programming*, 194(1):621–660, 2022. 1