Parametric Regression on the Grassmannian

Supplementary Material

This supplementary material contains technical details and additional results that were left-out in the original submission for brevity. References to sections or figures in the manuscript are marked as **[Paper, XXX]**. The source code for all approaches (including our implementations of Su *et al.* [1] and Rentmeesters [2]) is publicly available at:

https://bitbucket.org/yi_hong/ggr_all

1 Residuals to curves on $\mathcal{G}(p, n)$

In **[Paper, Algorithms 1 and 2]** we need the gradient of the residuals $d_g^2(\mathbf{Y}(r_i), \mathbf{Y}_i)$ (here, $\mathbf{Y}(r_i)$ is equal to $\mathbf{X}_1(r_i)$) with respect to the base point $\mathbf{Y}(r_i)$ in order to compute the jump conditions for the adjoint variable λ_1. The residual measures the squared geodesic distance between the point $\mathbf{Y}(r_i)$ on the fitted curve and the corresponding measurement $\mathbf{Y}_i$. To derive this gradient, we consider the constrained minimization problem of two points with exact matching:

$$\begin{aligned} E(\mathbf{Y}(r)) = & \int_{r_0}^{r_1} \operatorname{tr} \dot{\mathbf{Y}}(r)^\top \dot{\mathbf{Y}}(r)\, dr, \\ \text{subject to} \quad & \mathbf{Y}(r_0) = \mathbf{Y}_0, \mathbf{Y}(r_1) = \mathbf{Y}_1 \\ & \text{and } \mathbf{Y}(r_0)^\top \mathbf{Y}(r_0) = \mathbf{I}_p \ , \end{aligned} \tag{1}$$

with $\dot{\mathbf{Y}}(r) = (\mathbf{I}_n - \mathbf{Y}(r)\mathbf{Y}(r)^\top)\mathbf{C}$. We know that the squared distance can be formulated as $d_g^2(\mathbf{Y}_0, \mathbf{Y}_1) = \min_{\mathbf{Y}(r)} E(\mathbf{Y}(r))$ for $r_0 = 0$ and $r_1 = 1$. After adding the constraint on the form of $\dot{\mathbf{Y}}(r)$ via the time-dependent adjoint variable λ and the constraint $\mathbf{Y}(0)^\top \mathbf{Y}(0) = \mathbf{I}_p$ via λ_c, we obtain (by taking variations), among other terms, the optimality condition

$$(2\mathbf{C}^\top - \lambda^\top)(\mathbf{I}_n - \mathbf{Y}(r)\mathbf{Y}(r)^\top) = \mathbf{0} \ , \tag{2}$$

and another optimality condition for a free initial condition[1] $\mathbf{Y}(0)$

$$\nabla_{\mathbf{Y}(0)} E = -\lambda(0) + \mathbf{Y}(0)(\lambda_c^\top + \lambda_c) = \mathbf{0} \ . \tag{3}$$

[1] Note that technically we started with $\mathbf{Y}(r_0) = \mathbf{Y}_0$, i.e., this condition would not be free and we would not need to consider variations of $\mathbf{Y}(r_0)$. However, our goal is to compute the energy gradient with respect to the initial condition $\mathbf{Y}(r_0)$. Consequentially, the variation of the energy with respect to it allows us to compute $\nabla_{\mathbf{Y}(0)} E$.

Left-multiplication by $\mathbf{Y}(0)^\top$ yields $\lambda_c + \lambda_c^\top = \mathbf{Y}^\top(0)\lambda(0)$ which we can use to obtain, upon back-substitution into Eq. (3),

$$-(\mathbf{I}_n - \mathbf{Y}(0)\mathbf{Y}(0)^\top)\lambda(0) = \mathbf{0} \ . \tag{4}$$

Using Eq. (2) and the above expression for $\dot{\mathbf{Y}}(r)$, we can obtain

$$\nabla_{\mathbf{Y}(0)} E = -(\mathbf{I}_n - \mathbf{Y}(0)\mathbf{Y}(0)^\top)\lambda(0) = -2\dot{\mathbf{Y}}(0) \ , \tag{5}$$

with $\dot{\mathbf{Y}}(0)$ being the tangent vector at $\mathbf{Y}(0)$ such that $\mathrm{Exp}_{\mathbf{Y}(0)}(\dot{\mathbf{Y}}(0)) = \mathbf{Y}(1)$. This tangent vector can be computed via the log-map $\dot{\mathbf{Y}}(0) = \mathrm{Log}_{\mathbf{Y}(0)}(\mathbf{Y}(1))$ using the efficient algorithms in [3] for instance. Further details are also provided in Section 5 of this supplementary material.

2 Line search on the Grassmannian

Performing a line search on the Grassmann manifold in **[Paper, Algorithm 1]** is not as straightforward as in Euclidean space since we need to assure that the constraints for $\mathbf{Y}(r_0)$ and $\dot{\mathbf{Y}}(r_0)$ are fulfilled for any given step. In particular, changing $\mathbf{Y}(r_0)$ will change the associated tangent vector $\dot{\mathbf{Y}}(r_0)$. Once, we have updated $\mathbf{Y}(r_0)$ to $\mathbf{Y}^u(r_0)$ by moving along the geodesic defined by $\mathbf{Y}(r_0)$ and the gradient of the energy with respect to this initial point, *i.e.*, $\nabla_{\mathbf{Y}(r_0)} E$, we can transport the tangent $\dot{\mathbf{Y}}(r_0)$ to $\mathbf{Y}^u(r_0)$ using the closed form solution for *parallel transport* of [4] (see also Section 5). In particular,

$$\dot{\mathbf{Y}}^u(r_0) = \ [\mathbf{Y}(r_0)\mathbf{V} \ \ \mathbf{U}] \begin{pmatrix} -\sin t\boldsymbol{\Sigma} \\ \cos t\boldsymbol{\Sigma} \end{pmatrix} \mathbf{U}^\top + \ (\mathbf{I}_n - \mathbf{U}\mathbf{U}^\top)\dot{\mathbf{Y}}(r_0) \tag{6}$$

where $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$ is the compact SVD of the tangent vector at $\mathbf{Y}(r_0)$ along the geodesic connecting $\mathbf{Y}(r_0)$ and $\mathbf{Y}^u(r_0)$. Note that $t = 1$ in Eq. (6). Algorithm 1 lists the line search procedure in full technical detail.

Algorithm 1: Grassmannian equivalent of $x^{k+1} = x^k - \Delta t g$, where Δt is the timestep and g is the gradient.

Data: $\mathbf{Y}(r_0)$, $\dot{\mathbf{Y}}(r_0)$, $\nabla_{\mathbf{Y}(r_0)} E$, $\nabla_{\dot{\mathbf{Y}}(r_0)} E$, Δt
Result: Updated $\mathbf{Y}^u(r_0)$ and $\dot{\mathbf{Y}}^u(r_0)$
Compute $\overline{\dot{\mathbf{Y}}}^u(r_0) = \dot{\mathbf{Y}}(r_0) - \Delta t \nabla_{\mathbf{X}_2(r_0)} E$
Compute $\mathbf{Y}^u(r_0)$ by flowing for Δt along geodesic with initial condition $(\mathbf{Y}(r_0), -\nabla_{\mathbf{X}_1(r_0)} E)$
Transport $\overline{\dot{\mathbf{Y}}}^u(r_0)$ along the geodesic connecting $\mathbf{Y}(r_0)$ to $\mathbf{Y}^u(r_0)$, using (6), resulting in $\overline{\dot{\mathbf{Y}}}^u_T(r_0)$
Project updated initial velocity onto the tangent space (for consistency):
$\dot{\mathbf{Y}}^u(r_0) \leftarrow (\mathbf{I}_n - \mathbf{Y}^u(r_0)\mathbf{Y}^u(r_0)^\top)\overline{\dot{\mathbf{Y}}}^u_T(r_0)$.

Note: when implementing **[Paper, Algorithm 1]** and **[Paper, Algorithm 2]** it is important to pay attention to the ordering of the matrix multiplications, as performing them in an appropriate order will reduce time and memory complexity.

3 Cubic splines on $\mathcal{G}(p,n)$

We derive the evolution equations for $\mathbf{X}_3$ and $\mathbf{X}_4$ of **[Paper, Section 5.5]**. First, we show that by enforcing $\mathbf{X}_1^\top \mathbf{X}_2 = \mathbf{0}$ at all time points, we only need to enforce $\mathbf{X}_1^\top \mathbf{X}_1 = \mathbf{I}_p$ initially. This can be seen by taking the derivative of $\mathbf{X}_1^\top \mathbf{X}_1$ with respect to r, *i.e.*,

$$\frac{d}{dr}\mathbf{X}_1^\top \mathbf{X}_1 = \dot{\mathbf{X}}_1^\top \mathbf{X}_1 + \mathbf{X}_1^\top \dot{\mathbf{X}}_1 = \mathbf{X}_2^\top \mathbf{X}_1 + \mathbf{X}_1^\top \mathbf{X}_2 = \mathbf{0} \ . \tag{7}$$

In other words, if $\mathbf{X}_1(0)^\top \mathbf{X}_1(0) = \mathbf{I}_p$ holds, then $\mathbf{X}_1^\top \mathbf{X}_1 = \mathbf{I}_p$ holds for all r by enforcing $\mathbf{X}_1^\top \mathbf{X}_2 = \mathbf{0}$ at all time points. With this result in mind, the optimization problem of **[Paper, Eq. (22)]** can be rewritten as

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & E(\boldsymbol{\Theta}) = \frac{1}{2}\int_0^1 \operatorname{tr}\, \mathbf{X}_3^\top \mathbf{X}_3 \, dr \\ \text{subject to} \quad & \mathbf{X}_2 = \dot{\mathbf{X}}_1 \\ & \mathbf{X}_3 = \dot{\mathbf{X}}_2 + \mathbf{X}_1 \mathbf{X}_2^\top \mathbf{X}_2, \\ & \mathbf{X}_1^\top \mathbf{X}_2 = \mathbf{0}, \\ & \mathbf{X}_1(0)^\top \mathbf{X}_1(0) = \mathbf{I}_p \ . \end{aligned} \tag{8}$$

For the first three time-evolving constraints, we introduce three time-dependent adjoint variables, λ_1, λ_2, and λ_3, to convert the constrained problem to an unconstrained one. We then take the variations with respect to the state variables which results in the following system of adjoint equations:

$$\begin{aligned} -\dot{\lambda}_1^\top + (\mathbf{X}_2^\top \mathbf{X}_2)\lambda_2^\top + \lambda_3 \mathbf{X}_2^\top &= \mathbf{0}, \\ -\dot{\lambda}_2^\top - \lambda_1^\top + \mathbf{X}_1^\top \lambda_2 \mathbf{X}_2^\top + \lambda_2^\top \mathbf{X}_1 \mathbf{X}_2^\top + \lambda_3^\top \mathbf{X}_1^\top &= \mathbf{0}, \\ \mathbf{X}_3^\top - \lambda_2^\top &= \mathbf{0} \ . \end{aligned} \tag{9}$$

Using $\mathbf{X}_3 = \lambda_2$ and setting $\mathbf{X}_4 = \lambda_1$, we can rewrite the adjoint equations (9) as

$$\dot{\mathbf{X}}_4^\top = (\mathbf{X}_2^\top \mathbf{X}_2)\mathbf{X}_3^\top + \lambda_3 \mathbf{X}_2^\top, \tag{10}$$

$$\dot{\mathbf{X}}_3^\top = -\mathbf{X}_4^\top + \mathbf{X}_1^\top \mathbf{X}_3 \mathbf{X}_2^\top + \mathbf{X}_3^\top \mathbf{X}_1 \mathbf{X}_2^\top + \lambda_3^\top \mathbf{X}_1^\top \ . \tag{11}$$

Next, we derive the form of $\mathbf{X}_3$ using the dynamic constraints. By taking the time-derivative of $\mathbf{X}_1^\top \mathbf{X}_2$ we get

$$\begin{aligned} \frac{d}{dr}\mathbf{X}_1^\top \mathbf{X}_2 &= \dot{\mathbf{X}}_1^\top \mathbf{X}_2 + \mathbf{X}_1^\top \dot{\mathbf{X}}_2 \\ &= \dot{\mathbf{X}}_1^\top \mathbf{X}_2 + \mathbf{X}_1^\top (\mathbf{X}_3 - \mathbf{X}_1 \mathbf{X}_2^\top \mathbf{X}_2) \\ &= \mathbf{X}_2^\top \mathbf{X}_2 + \mathbf{X}_1^\top \mathbf{X}_3 - \mathbf{X}_2^\top \mathbf{X}_2 \\ &= \mathbf{X}_1^\top \mathbf{X}_3 \ . \end{aligned} \tag{12}$$

However, the constraint $\mathbf{X}_1^\top \mathbf{X}_2 = \mathbf{0}$ implies $\frac{d}{dr}\mathbf{X}_1^\top \mathbf{X}_2 = \mathbf{0}$ which yields

$$\mathbf{X}_1^\top \mathbf{X}_3 = \mathbf{0} \ . \tag{13}$$

Next $\mathbf{X}_1^\top \mathbf{X}_3 = \mathbf{0}$ implies $\frac{d}{dr}\mathbf{X}_1^\top \mathbf{X}_3 = \mathbf{0}$ and we get, as a side result,

$$\mathbf{X}_2^\top \mathbf{X}_3 + \mathbf{X}_1^\top \dot{\mathbf{X}}_3 = \mathbf{0} \ . \tag{14}$$

We can then use (13) to simplify Eq. (11) to

$$-\dot{\mathbf{X}}_3^\top - \mathbf{X}_4^\top + \lambda_3^\top \mathbf{X}_1^\top = \mathbf{0} \quad \Leftrightarrow \quad -\dot{\mathbf{X}}_3 - \mathbf{X}_4 + \mathbf{X}_1 \lambda_3 = \mathbf{0} \ . \tag{15}$$

Upon left-multiplication of Eq. (15) by $\mathbf{X}_1$ we obtain the expression for λ_3 as

$$\lambda_3 = \mathbf{X}_1^\top \dot{\mathbf{X}}_3 + \mathbf{X}_1^\top \mathbf{X}_4 \overset{(14)}{=} \mathbf{X}_1^\top \mathbf{X}_4 - \mathbf{X}_2^\top \mathbf{X}_3 \ . \tag{16}$$

Substituting λ_3 into Eq. (15) yields the evolution equation for $\dot{\mathbf{X}}_3$ as

$$\dot{\mathbf{X}}_3 = -\mathbf{X}_4 + \mathbf{X}_1 \mathbf{X}_1^\top \mathbf{X}_4 - \mathbf{X}_1 \mathbf{X}_2^\top \mathbf{X}_3 \tag{17}$$

and substituting λ_3 in Eq. (10) yields the evolution equation for $\dot{\mathbf{X}}_4$

$$\dot{\mathbf{X}}_4 = \mathbf{X}_3 \mathbf{X}_2^\top \mathbf{X}_2 + \mathbf{X}_2 \mathbf{X}_4^\top \mathbf{X}_1 - \mathbf{X}_2 \mathbf{X}_3^\top \mathbf{X}_2 \ . \tag{18}$$

4 System identification

To support a non-uniform weighting of samples during system identification in **[Paper, Section 7]**, we propose a *temporally* localized variant of [5]. This is beneficial in situations where we need a considerable number of frames for stable system identification, yet not all samples should contribute equally to the LDS parameter estimates. Specifically, given the measurement matrix $\mathbf{M} = [\mathbf{y}_1, \cdots, \mathbf{y}_\tau]$ and a set of weights $\mathbf{w} = [w_1, \cdots, w_\tau]$, such that $\sum_i w_i = \tau$, we perform a weighted SVD of $\mathbf{M}$, *i.e.*,

$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \mathbf{M}\mathrm{diag}(\sqrt{\mathbf{w}}) \ . \tag{19}$$

Then, as in [5], $\mathbf{C} = \mathbf{U}$ and $\mathbf{X} = \mathbf{\Sigma}\mathbf{V}^\top$. Once the state matrix $\mathbf{X}$ has been determined, $\mathbf{A}$ can be computed as $\mathbf{A} = \mathbf{X}_2^\tau \mathbf{W}^{\frac{1}{2}} (\mathbf{X}_1^{\tau-1} \mathbf{W}^{\frac{1}{2}})^\dagger$, where † denotes the pseudoinverse, $\mathbf{X}_2^\tau = [\mathbf{x}_2, \cdots, \mathbf{x}_\tau]$, $\mathbf{X}_1^{\tau-1} = [\mathbf{x}_1, \cdots, \mathbf{x}_{\tau-1}]$ and $\mathbf{W}^{\frac{1}{2}}$ is a diagonal matrix with $W_{ii}^{\frac{1}{2}} = [\frac{1}{2}(w_i + w_{i+1})]^{1/2}$.

5 Differential geometric tools on $\mathcal{G}(p, n)$

In this part of the supplementary material, we present the differential geometric "tools" required for our forward / backward shooting approach. In particular, we provide formulas / derivations for the Riemannian *exponential map (exp-map)*, the *inverse exponential map (aka log-map)* and *parallel transport* on $\mathcal{G}(p, n)$. Additionally, we list exemplary implementations in MATLAB (partly taken from the manopt [6] toolbox and our source code).

5.1 Exponential map

The exponential map (exp-map) maps a point $\mathbf{D}$ in the tangent space $\mathcal{T}_{\mathcal{Y}}\mathcal{G}(p,n)$ at $\mathcal{Y} = \text{span}(\mathbf{Y})$ to a point $\mathcal{Z} = \text{span}(\mathbf{Z})$ on the manifold $\mathcal{G}(p,n)$, *i.e.*,

$$\text{Exp}_{\mathbf{Y}}(t\mathbf{D}) = \mathbf{Z}, \quad t \in [0,1]$$

along the geodesic that connects $\mathcal{Y}$ and $\mathcal{Z}$. By letting $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ denote the compact SVD decomposition of $\mathbf{D}$, the exp-map on $\mathcal{G}(p,n)$, in terms of representers $\mathbf{Y}$ and $\mathbf{Z}$, can be written as

$$\mathbf{Z} = \mathbf{Y}\mathbf{V}\cos(t\mathbf{\Sigma})\mathbf{V}^\top + \mathbf{U}\sin(t\mathbf{\Sigma})\mathbf{V}^\top \ . \tag{20}$$

This computation is implemented in the manopt [6] toolbox. For convenience, MATLAB code (adjusted for readability) is listed below. Computation of $\mathbf{D}$ via the *inverse exponential map*, given $\mathbf{Y}$ and $\mathbf{Z}$ as input, is described in the next section.

```
function Z = exp_map(Y,D,t)
% EXP_MAP computes the exponential map on the Grassmannian G(p,n).
%
% Input:
%   Y - (n x p) matrix (representer for subspace \mathcal{Y})
%   D - (n x p) matrix (tangent vector at \mathcal{Y})
%   t - Time in [0,1]
%
% Output:
%
%   Z - (n x p) matrix; Result of walking along the geodesic, defined
%               by (Y,D), for time t;
%
% Reference - Eq. (2.65) from
%
%   Edelman et al.
%   The Geometry of algorithms with orthogonality constraints
%   Online: http://arxiv.org/pdf/physics/9806030.pdf
    tD = t*D;
    [U, S, V] = svd( tD, 0 );
    cosS = diag( cos( diag( S ) ) );
    sinS = diag( sin( diag( S ) ) );
    Z = Y*V*cosS*V' + U*sinS*V';
    [Q, unused] = qr( Z, 0 );
    Z = Q;
end
```

5.2 Inverse exponential map

The inverse exponential map (log-map) computes the mapping from a neighborhood of $U \subset \mathcal{G}(p,n)$ of $\mathcal{X}$ to $\mathcal{T}_{\mathcal{X}}\mathcal{G}(p,n)$. In terms of representers $\mathbf{X}, \mathbf{Y}$ for the subspaces $\mathcal{X} = \text{span}(\mathbf{X}), \mathcal{Y} = \text{span}(\mathbf{Y})$ we will write

$$\mathbf{H} = \text{Log}_{\mathbf{X}}(\mathbf{Y}) \ .$$

In other words, $\mathbf{H}$ is the direction matrix which allows you to start at $\mathcal{X}$ and walk along the geodesic in the direction of $\mathbf{H}$ to reach $\mathcal{Y}$ in unit time ($t = 1$), *i.e.*,

$$\mathbf{Y} = \text{Exp}_{\mathbf{X}}(\mathbf{H}) \ .$$

Let $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. By multiplying Eq. (20) with $\mathbf{X}^\top$ on the left-hand side (and $t = 1$), we get

$$\begin{aligned}\mathbf{X}^\top\mathbf{Y} &= \underbrace{\mathbf{X}^\top\mathbf{X}}_{=\mathbf{I}_p}\mathbf{V}\cos(\mathbf{\Sigma})\mathbf{V}^\top + \underbrace{\mathbf{X}^\top\mathbf{U}}_{=\mathbf{0}}\sin(\mathbf{\Sigma})\mathbf{V}^\top \\ &= \mathbf{V}\cos(\mathbf{\Sigma})\mathbf{V}^\top \ .\end{aligned}$$

Consequently, it follows that

$$\mathbf{U}\sin(\mathbf{\Sigma})\mathbf{V}^\top = \mathbf{Y} - \mathbf{X}\mathbf{X}^\top\mathbf{Y} \ .$$

We can now write

$$\mathbf{U}\sin(\mathbf{\Sigma})\mathbf{V}^\top(\mathbf{V}\cos(\mathbf{\Sigma})\mathbf{V}^\top)^{-1} = (\mathbf{Y} - \mathbf{X}\mathbf{X}^\top\mathbf{Y})(\mathbf{X}^\top\mathbf{Y}^\top)^{-1}$$

which – upon noting that (1) $(\mathbf{V}\cos(\mathbf{\Sigma})\mathbf{V}^\top)^{-1} = \mathbf{V}\cos(\mathbf{\Sigma})^{-1}\mathbf{V}^\top$ and (2) $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_p$ – reduces to

$$\mathbf{U}\tan(\mathbf{\Sigma})\mathbf{V}^\top = (\mathbf{Y} - \mathbf{X}\mathbf{X}^\top\mathbf{Y})(\mathbf{X}^\top\mathbf{Y}^\top)^{-1} \ .$$

This yields $\mathbf{H}$ via a SVD decomposition of $(\mathbf{Y} - \mathbf{X}\mathbf{X}^\top\mathbf{Y})(\mathbf{X}^\top\mathbf{Y}^\top)^{-1}$ as

$$\mathbf{H} = \mathrm{Log}_{\mathbf{X}}(\mathbf{Y}) = \mathbf{U}\arctan(\mathbf{\Sigma})\mathbf{V}^\top \ .$$

This is also the way the Grassmannian log-map is implemented in the manopt [6] toolbox. For convenience, MATLAB code (adjusted for readability) is listed below.

```
function H = log_map(X,Y)
% LOG_MAP computes the inverse exponential map on the Grassmannian.
%
% Input:
%   X – (n x p) matrix (representer for subspace \mathcal{X})
%   Y – (n x p) matrix (representer for subspace \mathcal{Y})
%
% Output:
%
%   H – (n x p) direction matrix in the tangent space at subspace \mathcal{X}
    [unused, p] = size(X);
    ytx = Y.'*X;
    At = Y.'-ytx*X.';
    Bt = ytx\At;
    [U, S, V] = svd(Bt.', 'econ');

    U = U(:, 1:p);
    S = diag(S);
    S = S(1:p);
    V = V(:, 1:p);
    H = U*diag(atan(S))*V.';
end
```

5.3 Parallel transport

Given two subspaces $\mathcal{X}, \mathcal{Y}$, represented via $\mathbf{X}, \mathbf{Y}$ and a direction matrix $\mathbf{H} \in \mathcal{T}_{\mathcal{X}}\mathcal{G}(p,n)$ such that $\mathbf{Y} = \mathrm{Exp}_{\mathbf{X}}(\mathbf{H})$, the objective is to transport an arbitrary tangent vector $\mathbf{\Delta}$ at $\mathcal{T}_{\mathcal{X}}\mathcal{G}(p,n)$ to $\mathcal{T}_{\mathcal{Y}}\mathcal{G}(p,n)$

along the geodesic connecting $\mathcal{X}$ and $\mathcal{Y}$. According to Edelman *et al.* [4], letting $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, parallel transport (denoted by τ) can be computed via

$$\tau\mathbf{\Delta}(t) = -\mathbf{Y}\mathbf{V}\sin(t\mathbf{\Sigma})\mathbf{U}^\top\mathbf{\Delta} + \mathbf{U}\cos(t\mathbf{\Sigma})\mathbf{U}^\top + (\mathbf{I}_n - \mathbf{U}\mathbf{U}^\top)\mathbf{\Delta}, \quad t \in [0,1] \ .$$

This is *not* implemented in the manopt toolbox. For convenience, MATLAB code is listed below.

```
function Delta = parallel_transport(O1,B,O2,t)
% PARALLEL_TRANSPORT parallel transports a tangent along geodesic.
%
% Input:
%   O1 - (n x p) matrix (representer for subspace \mathcal{X})
%   O2 - (n x p) matrix (representer for subspace \mathcal{Y})
%   B  - (n x p) matrix (tangent vector at \mathcal{X})
%   t  - time in [0,1]
%
% Output:
%
%   Delta - (n x p) matrix (tangent vector B, transported to the
%           tangent space at \mathcal{Y}
    H = log_map(O1,O2); % compute direction matrix
    [U,S,V] = svd(H, 'econ');
    Sigmat = diag(S)*t;
    M = [diag(-sin(Sigmat)); diag(cos(Sigmat))];
    part0 =[O1*V U]*(M*(U'*B));
    part1 = B - U*(U'*B);
    Delta = part0 + part1;
end
```

5.4 Miscellaneous

The paper only gives a very concise description of the Riemmannian structure of $\mathcal{G}(p,n)$. Here, we provide a more comprehensive description of the relationship between $\mathbf{\Delta}_\mathcal{Y}$ and $\mathbf{C}$ from **[Paper, Eq. (16)]**. Every tangent direction $\mathbf{\Delta}_\mathcal{Y} \in \mathcal{T}_\mathcal{Y}\mathcal{G}(p,n)$ at a point $\mathcal{Y} \in \mathcal{G}(p,n)$ (represented by $\mathbf{Y}$) can be written as

$$\begin{aligned}\mathbf{\Delta}_\mathcal{Y} &= (\mathbf{I}_n - \mathbf{Y}\mathbf{Y}^\top)\mathbf{C} \\ &= (\mathbf{Y}_\perp\mathbf{Y}_\perp^\top)\mathbf{C}\end{aligned}$$

where $\mathbf{C}$ is an arbitrary $(n \times p)$ matrix and $\mathbf{Y}_\perp \in \mathbb{R}^{n\times(n-p)}$ denotes the orthogonal complement of $\mathbf{Y}$. This can be interpreted as projecting an arbitrary $(n \times p)$ matrix into the subspace span$(\mathbf{Y}_\perp)$. Now, if we take the canonical Euclidean metric, we get

$$\begin{aligned}\text{tr } (\mathbf{\Delta}_\mathcal{Y}^\top\mathbf{\Delta}_\mathcal{Y}) &= \text{tr } [(\mathbf{I}_n - \mathbf{Y}\mathbf{Y}^\top)\mathbf{C}]^\top[(\mathbf{I}_n - \mathbf{Y}\mathbf{Y}^\top)\mathbf{C}] \\ &= \text{tr } [\mathbf{C}^\top(\mathbf{I}_n - \mathbf{Y}\mathbf{Y}^\top)\mathbf{C}]\end{aligned}$$

since, for orthogonal projectors $\mathbf{P} = (\mathbf{I}_n - \mathbf{Y}\mathbf{Y}^\top)$, it holds that $\mathbf{P} = \mathbf{P}^2$. This is the result listed in **[Paper, Eq. (16)]**.

6 Additional curve-fitting comparisons

6.1 Synthetic data

In **[Paper, Fig. 4]**, we present a comparison to the work of Su *et al.* [1] for only one specific choice of data-matching (controlled via λ_1) *vs.* smoothness (controlled via λ_2) balance (due to space restrictions), in particular, $(\lambda_1, \lambda_2) = (1, 0.1) \Rightarrow \lambda = \lambda_1/\lambda_2 = 10$. Fig. 1 shows additional comparisons for various choices of λ from $\lambda = 10$ in Fig. 1(a) to $\lambda = 0.1$ in Fig. 1(d).

6.2 Real (shape) data

In the paper, we compare our methods (Std-GGR, TW-GGR and CS-GGR) to Rentmeesters [2] and Su *et al.* [1] on the *shape data* of **[Paper, Section 7]**. For [2], the evaluation protocol is exactly the same as for our approaches, since we also obtain a parametric regression model.

For Su *et al.* [1], we note *two* subtle adjustments in the evaluation protocol. First, we *cannot* run full leave-one-subject-out cross-validation (CV), since we cannot leave-out the first or the last data point, simply because we obtain the sought-for curve as a collection of samples along that curve. Consequently, there would be no way to *extrapolate* to the left-out time points. For that reason, we always keep the data instances at the first and last time point and perform leave-one-subject-out CV on the remaining instances. Computation of the R^2 score and the MSE on the corpus callosum data is then done in the same way as for Std-GGR, TW-GGR and CS-GGR. Table 1 lists the detailed results of Su *et al.* [1] for various choices of λ_1/λ_2.

λ_1	λ_2	MSE (e-2)	R^2
1	0.1	1.36	0.29
1	1.0	1.27	0.20
1	5.0	1.25	0.16
1	10.0	1.25	0.15

Table 1: Results of Su *et al.* [1] on the *corpus callosum* data of **[Paper, Section 7]** for various choices of λ_1/λ_2. The best result of our methods, reported in **[Paper, Table 2]**, is an MSE of 1.22e-2 (with an R^2 score of 0.15). We report the result of Su *et al.* at $\lambda_1/\lambda_2 = 1/10$ (*i.e.*, data matching *vs.* smoothness) in the paper (marked green here), since the corresponding MSE of 1.25e-2 is closest to the MSE of our best result. Any further increase in λ_2 did not lead to better MSE scores.

The second difference is in the evaluation protocol for the rat calvarium data. This dataset has multiple data instances at each time point. In particular, we have 18 data instances (*i.e.*, the 18 individuals) at each of the 8 time points in the data. The method of Su *et al.* does not directly support such a setting. For that reason, we fit *one* model per individual, compute MSE and R^2 and then average the results. Table 2 reports the detailed results for various choices of λ_1/λ_2.

λ_1	λ_2	MSE (e-3)	R^2
1	0.1	4.4	0.97
1	1.0	4.4	0.93
1	5.0	4.3	0.92
1	10.0	4.1	0.89

Table 2: Results of Su *et al.* [1] on the *rat calvarium* data of **[Paper, Section 7]** for various choices of λ_1/λ_2. The best result of our methods, reported in **[Paper, Table 2]**, is an MSE of 1.2e-3 (with a R^2 score of 0.81). We report the result of Su *et al.* at $\lambda_1/\lambda_2 = 1/10$ (*i.e.*, data matching *vs.* smoothness) in the paper (marked green here), since the corresponding MSE of 4.1e-3 is closest to the MSE of our best result. Any further increase in λ_2 did not lead to better MSE scores.

References

[1] J. Su, I. Dryden, E. Klassen, H. Le, and A. Srivastava, "Fitting smoothing splines to time-indexed, noisy points on non-linear manifolds," *Image Vision Comput.*, vol. 30, pp. 428–442, 2012.

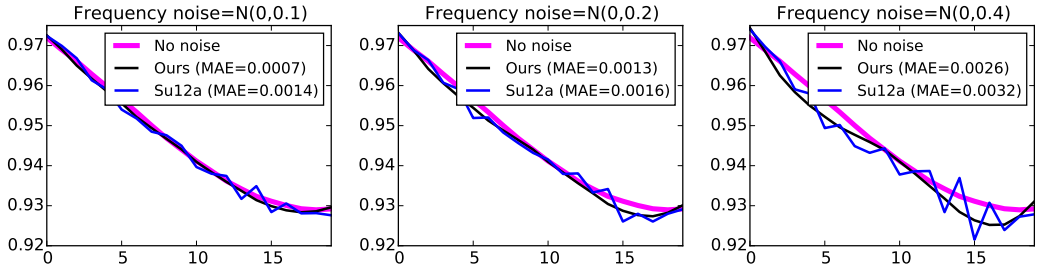
[2] Q. Rentmeesters, "A gradient method for geodesic data fitting on some symmetric Riemannian manifolds," in *CDC-ECC*, 2011.

[3] K. Gallivan, A. Srivastava, L. Xiuwen, and P. V. Dooren, "Efficient algorithms for inferences on Grassmann manifolds," in *Statistical Signal Processing Workshop*, 2003, pp. 315–318.

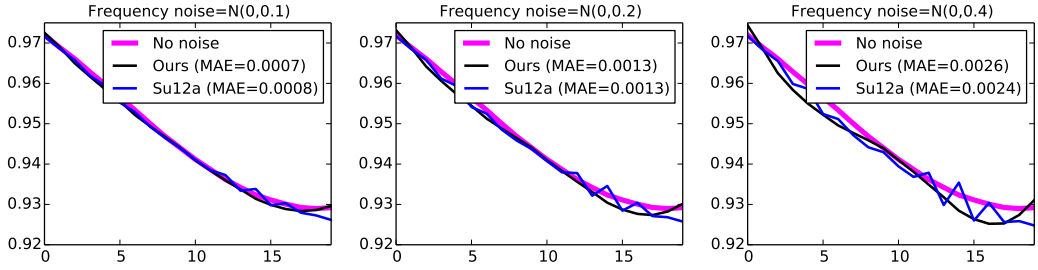
[4] A. Edelman, T. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353, 1998.

[5] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vision*, vol. 51, no. 2, pp. 91–109, 2003.

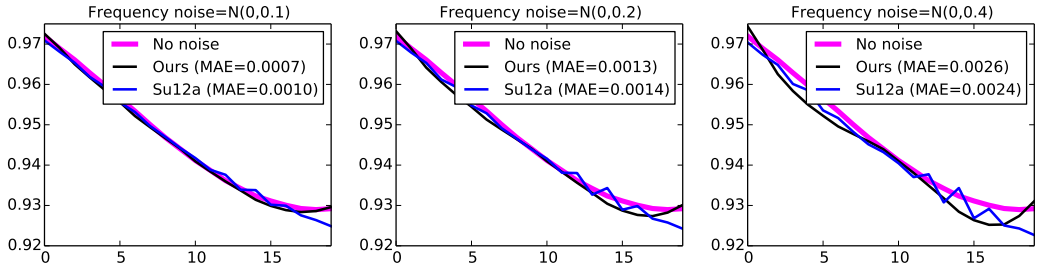
[6] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, pp. 1455–1459, 2014. [Online]. Available: http://www.manopt.org



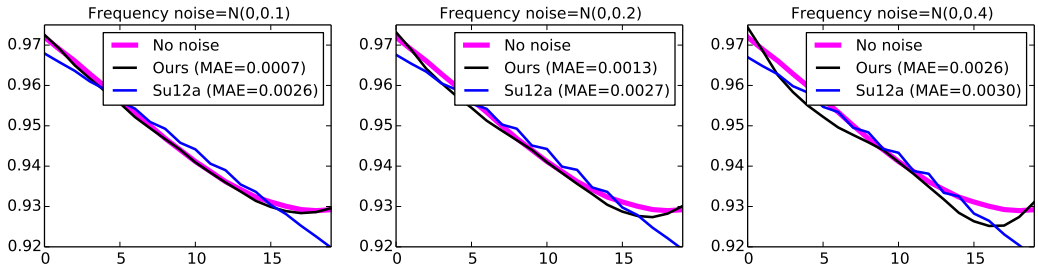
(a) $\lambda_1 = 1, \lambda_2 = 0.1$



(b) $\lambda_1 = 1, \lambda_2 = 0.5$



(c) $\lambda_1 = 1, \lambda_2 = 1$



(d) $\lambda_1 = 1, \lambda_2 = 10$

Figure 1: Additional comparisons to Su *et al.* [1] for various choices of λ_1, λ_2 and different levels of Gaussian noise added to the signal frequencies. The x-axis shows the index of 20 different signal frequencies (obtained by sampling a sine function), the y-axis shows the real-part of the largest eigenvalue of the state-transition matrix $\mathbf{A}$, reconstructed from the observability matrices of the associated linear dynamical systems (Fig. 1(a) is shown in **[Paper, Fig. 4]**).