Parametric Regression on the Grassmannian

Yi Hong, Roland Kwitt, Nikhil Singh, Nuno Vasconcelos and Marc Niethammer

Abstract—We address the problem of fitting parametric curves on the Grassmann manifold for the purpose of intrinsic parametric regression. We start from the energy minimization formulation of linear least-squares in Euclidean space and generalize this concept to general nonflat Riemannian manifolds, following an *optimal-control* point of view. We then specialize this idea to the Grassmann manifold and demonstrate that it yields a simple, extensible and easy-to-implement solution to the parametric regression problem. In fact, it allows us to extend the basic geodesic model to (1) a "time-warped" variant and (2) cubic splines. We demonstrate the utility of the proposed solution on different vision problems, such as shape regression as a function of age, traffic-speed estimation and crowd-counting from surveillance video clips. Most notably, these problems can be conveniently solved within the same framework without any specifically-tailored steps along the processing pipeline.

Index Terms—Parametric regression, Grassmann manifold, geodesic shooting, time-warping, cubic splines

1 Introduction

Many data objects in computer vision problems admit a subspace representation. Examples include feature sets obtained after dimensionality reduction via principal component analysis (PCA), observability matrix representations of linear dynamical systems, or landmark-based representations of shapes. Assuming equal dimensionality (*e.g.*, the same number of landmarks), data objects can be interpreted as points on the Grassmannian $\mathcal{G}(p,n)$, *i.e.*, the manifold of p-dimensional linear subspaces of $\mathbb{R}^n$. The seminal work of [1] and the introduction of efficient processing algorithms to manipulate points on the Grassmannian [2] has led to a variety of principled approaches to solve different vision and learning problems. These include domain adaptation [3], [4], gesture recognition [5], face recognition under illumination changes [6], or the classification of visual dynamic processes [7]. Other works have explored subspace estimation via conjugate gradient descent [8], mean shift clustering [9], or the definition of suitable kernel functions [10], [11], [12] that can be used with a variety of kernel-based machine learning techniques.

Since, most of the time, the primary objective is to perform classification or recognition tasks on the Grassmannian, the problem of intrinsic regression in a parametric setting has gained little attention. However, modeling the relationship between manifold-valued data and associated descriptive variables has the potential to address many problems in a principled way. For instance, it enables prediction of the descriptive variable while respecting the geometry of the underlying space. Further, in scenarios such as shape regression, a common problem in computational anatomy, we are specifically interested in summarizing continuous trajectories that capture variations in the manifold-valued variable as a function of the scalar independent variable. Fig. 1 illustrates these two inference objectives. While predictions of the scalar-valued variable could, in principle, be formulated within existing frameworks such as Gaussian processes or support vector regression, *e.g.*, by using Grassmann kernels [10], [11], it is unclear how to or if it is possible to address the second inference objective in such a formulation.

In this work, we propose an approach to intrinsic regression which allows us to directly fit parametric curves to a collection of data points on the Grassmann manifold, indexed by a scalar-valued variable. This facilitates to address both aforementioned inference tasks within the same framework. Preliminary versions of this manuscript [13], [14] focused on (1) fitting geodesics and (2) how to re-parametrize the independent variable to increase flexibility. The contribution of this work has multiple aspects, as outlined below.

First, we revisit the optimal-control perspective of curve fitting in Euclidean space as an example (Section 3) and then discuss extensions of linear and cubic spline regression on Riemannian manifolds (Section 4) and the Grassmannian in particular (Section 5). We argue that this exposition offers greater insight into the proposed solution of shooting strategy via optimal-control.

Second, we introduce a variational spline formulation (Section 5.5) based on the concept of *acceleration* control. While this is similar to prior work in the literature, *e.g.*, Machado *et al.* [15], it bypasses the need to explicitly compute the Riemannian curvature tensor which is a considerable advantage from a computational point of view. We eventually arrive at a set of evolution equations for cubic splines on the Grassmannian that enable straightforward numerical optimization. All proposed models are *simple* and natural extensions of classic re-

- *Y. Hong, N. Singh and M. Niethammer are with the Department of Computer Science, University of North Carolina, Chapel Hill, NC, USA. E-mail: yihong@cs.unc.edu, nsingh@cs.unc.edu, mn@cs.unc.edu*
- *R. Kwitt is with the Department of Computer Science, University of Salzburg, A-5020 Salzburg, Austria. E-mail: rkwitt@gmx.at*
- *N. Vasconcelos is with the Department of Electrical and Computer Engineering, University of California San Diego, CA, USA. E-mail: nvasconcelos@ucsd.edu*

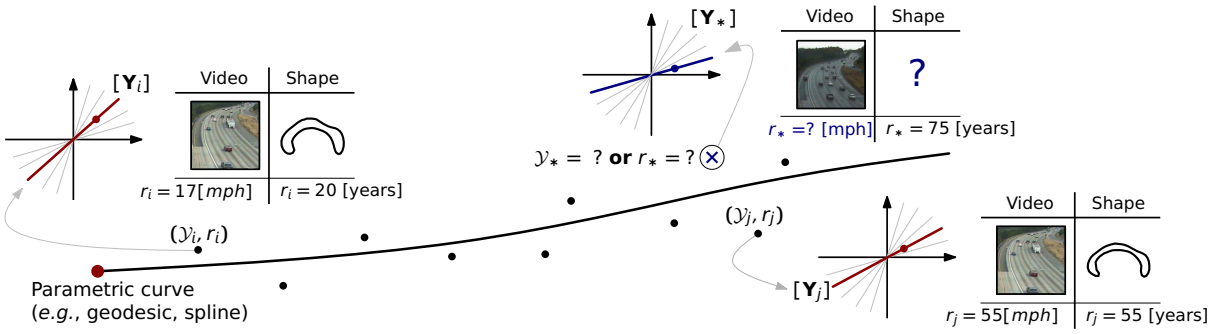

Fig. 1: Illustration of parametric regression and inference. At the point marked $\otimes$, the objective for (1) traffic videos is to predict the independent variable r_* (*e.g.*, speed), whereas for (2) corpus callosum shapes we seek the manifold-valued $\mathcal{Y}_*$ at specific values of the independent variable (*e.g.*, age). Elements on the Grassmannian are visualized as lines through the origin, *i.e.*, $\mathcal{Y}_i \in \mathcal{G}(1,2)$.

gression models in Euclidean space. They provide a *compact* representation of the complete curve, as opposed to discrete curve fitting approaches which typically return a sampling of the sought-for curves (*cf.* [16]). In addition, the parametric form of the curves, *i.e.*, given by initial conditions, allows to freely move along them and synthesize additional observations. We also note that parametric regression opens up the possibility of statistical analysis of curves on the manifold, which is essential, *e.g.*, in comparative studies in medical imaging.

Our *third* contribution is a more comprehensive (compared to [13], [14]) experimental evaluation on synthetic (Section 6) and real data (Section 7) which includes comparisons to alternative strategies from the literature. In particular, we demonstrate versatility of the proposed approach on two types of vision problems where data objects admit a representation on the Grassmannian. First, we model the aging trends in human brain structures and the rat calvarium (Section 7.2) under an affine-invariant representation of shape [17]. Second, we use our models to predict traffic speed and crowd counts (Section 7.3) from dynamical system representations of surveillance video clips *without* any specifically tailored preprocessing. All these problems are solved within the same framework with minor parameter adjustments.

In summary, our approach offers a simple solution that is (1) extensible, (2) easy to implement and (3) does not require specific knowledge of differential geometric concepts such as curvature or Jacobi fields.

2 Previous Work

At the coarsest level, we distinguish between two categories of regression approaches: *parametric* and *non-parametric* strategies, with all the known trade-offs on both sides [18]. In fact, non-parametric regression on *nonflat* manifolds has gained considerable attention over the last years. Strategies range from kernel regression [19] on the manifold of diffeomorphic transformations to gradient-descent [20] approaches on manifolds commonly encountered in computer vision, such as the group of rotations $\mathcal{SO}(3)$ or Kendall's shape space. In other works, discretizations of the curve fitting problem have been explored [21], [22], [16] which, in some cases, even allow to employ second-order optimization methods [23]. Because our work is a representative of the *parametric* category, we mostly focus on parametric approaches in the following review.

While differential geometric concepts, such as geodesics and intrinsic higher-order curves, have been well-studied [24], [25], [26], [15], [27], their use for parametric regression, *i.e.*, finding parametric relationships between the manifold-valued variable and an independent scalar-valued variable, has only recently gained interest. A variety of methods extending concepts of regression in Euclidean space to nonflat manifolds have been proposed. Rentmeesters [28], Fletcher [29] and Hinkle *et al.* [30] address the problem of geodesic fitting on Riemannian manifolds, primarily focusing on symmetric spaces, to which the Grassmannian belongs. Batzies *et al.* [27] study a theoretical characterization of fitting geodesics on the Grassmannian. Niethammer *et al.* [31] generalized linear regression to the manifold of diffeomorphisms to model image time-series data, followed by works extending this concept [32], [33] and enabling the use of higher-order models [34].

From a conceptual point of view, we can identify two groups of solution strategies to solve parametric regression problems on nonflat manifolds: first, *geodesic shooting* based strategies which address the problem using adjoint methods from an optimal-control point of view [31], [32], [33], [34], [30]; the second group comprises strategies which are based on optimization techniques that leverage *Jacobi fields* to compute the required gradients [28], [29]. Our approach is a representative of the first category. Unlike Jacobi field approaches, our method does not require computation of the curvature explicitly and easily extends to higher-order models, such as the proposed cubic splines extension.

In the context of computer-vision problems, Lui [5] recently adapted the known Euclidean least-squares solution to the Grassmannian. While this strategy works remarkably well for the presented gesture recognition tasks, the formulation does not guarantee the minimization of the sum-of-squared geodesic distances within the manifold, which would be the natural extension of least-squares to Riemannian manifolds according to the

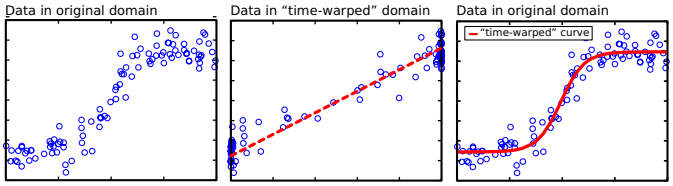

Fig. 2: Illustration of time-warped regression in $\mathbb{R}$. The dashed straight-line (middle) shows the fitting result in the warped time coordinates, and the solid curve (right) demonstrates the fitting result to the original data points (left).

literature. Hence, the geometric and variational interpretation of [5] remains unclear. In contrast, we address the problem from the aforementioned energy-minimization point of view which allows us to guarantee, by design, consistency with the geometry of the manifold.

To the best of our knowledge, the closest works to ours are [28], [29] and [27] in the context of fitting *geodesics*, as well as [15] (and to some extent [30]) in the context of fitting *higher-order curves*.

In [27], Batzies *et al.* present a theoretical study of fitting geodesics (*i.e.*, first-order curves) on the Grassmannian and derive a set of optimality criteria. However, the work is purely theoretical and, as mentioned in [27, Sect. 1], the objective is *not* to provide a numerical solution scheme. Rentmeesters [28] and Fletcher [29] propose optimization based on Jacobi fields to fit geodesics on general Riemannian manifolds. Contrary to our approach, it does not follow trivially how to generalize [28] (or [29]) to higher-order models.

In [15], Machado *et al.* specifically address the problem of fitting higher-order curves on Riemannian manifolds. Based on earlier works by Noakes *et al.* [24], Camarinha *et al.* [25] and Crouch & Leite [26], they introduce a different variational formulation of the problem and derive optimality criteria from a theoretical point of view. From a practical perspective, it remains unclear (as with [27] in case of geodesics) how these optimality criteria translate into a numerical optimization scheme. In other work, Hinkle *et al.* [30] address the problem of fitting polynomials, but mostly focus on manifolds with a Lie group structure[1]. In that case, adjoint optimization is greatly simplified. However, in general, curvature computations are required which can be tedious.

In comparison to prior work, we derive *alternative* optimality criteria for geodesics and cubic splines using principles from optimal-control. These conditions not only form the basis for our shooting approach, but also naturally lead to convenient iterative algorithms. By construction, the obtained solutions are guaranteed to be the sought-for curves (*i.e.*, geodesics, splines) on the manifold. In addition, our formulation for cubic splines does *not* require computation of the Riemannian curvature tensor.

1. $\mathcal{G}(p,n)$ does not possess such a group structure.

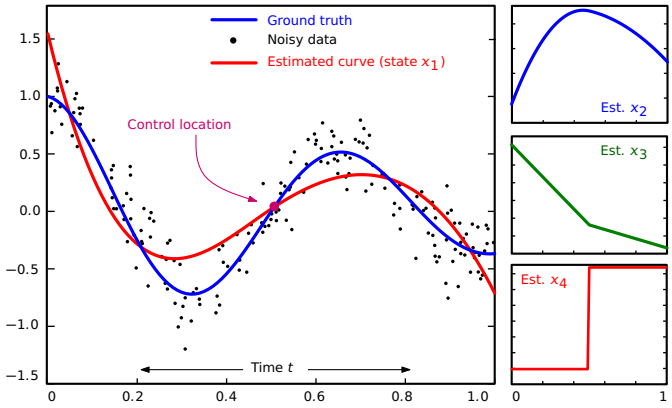

Fig. 3: Cubic spline regression in $\mathbb{R}$. The left side shows the regression result, and the remaining plots show the other states.

3 Regression in $\mathbb{R}^n$ via Optimal-Control

We start with a review of linear regression in $\mathbb{R}^n$ and discuss its solution via optimal-control. While regression is a well studied statistical technique and several solutions exist for univariate and multivariate models, we will see that the presented optimal-control perspective not only allows to easily generalize regression to manifolds but also to define more complex parametric models on these manifolds.

3.1 Linear regression

A straight line in $\mathbb{R}^n$ can be defined as an acceleration-free curve with parameter t, represented by states, $(x_1(t), x_2(t))$, such that $\dot{x}_1 = x_2$, and $\dot{x}_2 = 0$, where $x_1(t) \in \mathbb{R}^n$ is the *position* of a particle at time t and $x_2(t) \in \mathbb{R}^n$ represents its *velocity* at t. Let $\{y_i\}_{i=0}^{N-1} \in \mathbb{R}^n$ denote a collection of N measurements at time instances $\{t_i\}_{i=0}^{N-1}$ with $t_i \in [0,1]$. We define the linear regression problem as that of estimating a parametrized linear motion of the particle $x_1(t)$, such that the path of its trajectory best fits the measurements in the least-squares sense. The unconstrained optimization problem, from an optimal-control perspective, is

$$\min_{\boldsymbol{\Theta}} \quad E(\boldsymbol{\Theta}) = \sum_{i=0}^{N-1} \|x_1(t_i) - y_i\|^2 + \int_0^1 \lambda_1^\top(\dot{x}_1 - x_2) + \lambda_2^\top(\dot{x}_2)\, dt \ , \tag{1}$$

with $\boldsymbol{\Theta} = \{x_i(0)\}_{i=1}^2$, *i.e.*, the *initial conditions*, and $\lambda_1, \lambda_2 \in \mathbb{R}^n$ are time-dependent Lagrangian multipliers. For readability, we have omitted the argument t for $\lambda_1(t)$ and $\lambda_2(t)$. These variables are also referred to as *adjoint* variables, enforcing the dynamical "straight-line" constraints. Evaluating the gradients with respect to the state variables results in the *adjoint system* as $\dot{\lambda}_1 = 0$, and $-\dot{\lambda}_2 = \lambda_1$, with jumps in λ_1 as $\lambda_1(t_i^+) - \lambda_1(t_i^-) = 2(x_1(t_i) - y_i)$, at measurements t_i. The optimality conditions on the gradients also result in the boundary conditions $\lambda_1(1) = 0$ and $\lambda_2(1) = 0$. Finally,

the gradients with respect to the initial conditions are

$$\nabla_{x_1(0)} E = -\lambda_1(0), \text{ and } \nabla_{x_2(0)} E = -\lambda_2(0) \ . \quad (2)$$

These gradients are evaluated by integrating backward the adjoint system to $t = 0$ starting from $t = 1$.

This optimal-control perspective constitutes a general method for estimating first-order curves which allows to generalize the notion of straight lines to manifolds (geodesics), as long as the forward system (dynamics), the gradient computations, as well as the gradient steps all respect the geometry of the underlying space.

3.2 Time-warped regression

Fitting straight lines is too restrictive for some data. Hence, the idea of time-warped regression is to use a simple model to warp the time-points, or more generally the independent variable, when comparison to data is performed, *e.g.*, as in the data matching term of Eq. (1). The *time-warp* should maintain the order of the data, and hence needs to be diffeomorphic. This is conceptually similar to an *error-in-variables* model where uncertainties in the independent variables are modeled. However, in the concept of time-warping, we are not directly concerned with modeling such uncertainties, but instead in obtaining a somewhat richer model based on a known and easy-to-estimate linear regression model.

In principle, the mapping of the time points could be described by a general diffeomorphism. In fact, such an approach is followed in [35] for spatio-temporal atlas-building in the context of shape analysis. Our motivation for proposing an approach to linear regression with *parametric* time-warps is to keep the model simple while gaining more flexibility. Extensions to non-parametric approaches can easily be obtained. A representative of a simple parametric regression model is *logistic regression*[2] which is typically used to model saturation effects. Under this model, points that are close in time for the linear fit may be mapped to points far apart in time, thereby allowing to model saturations for instance (*cf.* Fig. 2). Other possibilities of parametric time-warps include those derived from families of quadratic, logarithmic and exponential functions.

Formally, let $f : \mathbb{R} \to \mathbb{R}$, $t \mapsto \bar{t} = f(t; \boldsymbol{\theta})$ denote a parametrized (by $\boldsymbol{\theta}$) time-warping function and let $x_1(\bar{t})$ denote the particle on the regression line in the warped time coordinates $\bar{t}$. Following this notation, the states are denoted as $(x_1(\bar{t}), x_2(\bar{t}))$ and represent position and slope in re-parametrized time $\bar{t}$. In *time-warped regression*, the *data matching* term in the sum of Eq. (1) then becomes $\|x_1(f(t_i; \boldsymbol{\theta})) - y_i\|^2$ and the objective (as before) is to optimize $x_1(\bar{t}_0)$ and $x_2(\bar{t}_0)$ as well as the parameter $\boldsymbol{\theta}$.

A convenient way to minimize the energy functional in Eq. (1) with the adjusted data matching term is to use an alternating optimization strategy. That is, we first fix $\boldsymbol{\theta}$ to update the initial conditions, and then fix the initial conditions to update $\boldsymbol{\theta}$. This requires the derivative of the energy with respect to $\boldsymbol{\theta}$ for fixed $x_1(\bar{t})$. Using the chain rule, we obtain the gradient $\nabla_{\boldsymbol{\theta}} E$ as

$$2 \sum_{i=0}^{N-1} (x_1(f(t_i; \boldsymbol{\theta})) - y_i)^\top \dot{x}_1(f(t_i; \boldsymbol{\theta})) \nabla_{\boldsymbol{\theta}} f(t_i; \boldsymbol{\theta}) \ . \quad (3)$$

Given a numerical solution to the regression problem of Section 3.1, the time-warped extension alternatingly updates (1) the initial conditions $(x_1(\bar{t}_0), x_2(\bar{t}_0))$ in the warped time domain using the gradients in Eq. (2) and (2) $\boldsymbol{\theta}$ using the gradient in Eq. (3). Fig. 2 visualizes the principle of time-warped linear regression on a collection of artificially generated data points. While the new model only slightly increases the overall complexity, it notably increases modeling flexibility by using a curve instead of a straight line.

3.3 Cubic spline regression

To further increase the flexibility of a regression model, cubic splines are another commonly used technique. In this section, we revisit cubic spline regression from the optimal-control perspective. This will facilitate the transition to general Riemannian manifolds.

3.3.1 Variational formulation

An acceleration-controlled curve with time-dependent states (x_1, x_2, x_3) such that $\dot{x}_1 = x_2$ and $\dot{x}_2 = x_3$, defines a cubic curve in $\mathbb{R}^n$. Such a curve is a solution to the energy minimization problem, *cf.* [36],

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & E(\boldsymbol{\Theta}) = \frac{1}{2} \int_0^1 \|x_3\|^2 \, dt, \\ \text{subject to} \quad & \dot{x}_1 = x_2(t) \text{ and } \dot{x}_2 = x_3(t) \ , \end{aligned} \quad (4)$$

with $\boldsymbol{\Theta} = \{x_i(t)\}_{i=1}^3$. Here, x_3 is referred to as the *control variable* that describes the acceleration of the dynamics in this system. Similar to the strategy for fitting straight lines, we can get a relaxation solution to Eq. (4) by adding adjoint variables which leads to the system of adjoint equations $\dot{\lambda}_1 = 0$ and $\dot{x}_3 = -\lambda_1$.

3.3.2 From relaxation to shooting

To obtain the shooting formulation, we explicitly add the evolution of x_3, *i.e.*, $\dot{x}_3 = -\lambda_1$, as another dynamical constraint; this increases the order of the dynamics. Setting $x_4 = -\lambda_1$ results in the classical system of equations for shooting cubic curves

$$\dot{x}_1 = x_2(t), \quad \dot{x}_2 = x_3(t), \quad \dot{x}_3 = x_4(t), \quad \dot{x}_4 = 0 \ . \quad (5)$$

The states (x_1, x_2, x_3, x_4), at all times, are entirely determined by their initial values $\{x_i(0)\}_{i=1}^4$ and, in particular we have $x_1(t) = x_1(0) + x_2(0)t + \frac{1}{2}x_3(0)t^2 + \frac{1}{6}x_4(0)t^3$.

3.3.3 Data-independent controls

Using the shooting equations of Eq. (5) for cubic splines, we can define a *smooth* curve that best fits the data in the least-squares sense. Since a cubic polynomial by itself is restricted to only fit "cubic-like" data, we add flexibility

2. Not to be confused with the statistical classification method.

by gluing piecewise cubic polynomials together. Typically, we define controls at pre-defined locations, and only allow the state x_4 to jump at those locations.

We let $\{t_c\}_{c=1}^{C}, t_c \in (0,1)$ denote C data-independent fixed control points, which implicitly define $C+1$ intervals in $[0,1]$, denoted as $\{\mathcal{I}_c\}_{c=1}^{C+1}$. The constrained energy minimization problem corresponding to the regression task, in this setting, can be written as

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & E(\boldsymbol{\Theta}) = \sum_{c=1}^{C+1} \sum_{t_i \in \mathcal{I}_c} \|x_1(t_i) - y_i\|^2, \\ \text{subject to} \quad & \left.\begin{aligned} \dot{x}_1 = x_2(t),\ \dot{x}_2 = x_3(t), \\ \dot{x}_3 = x_4(t),\ \dot{x}_4 = 0, \end{aligned}\right\} \text{within } \mathcal{I}_c \\ & \text{and } x_1, x_2, x_3 \text{ are continuous across } t_c \ , \end{aligned} \tag{6}$$

with parameters $\boldsymbol{\Theta} = \{\{x_i(0)\}_{i=1}^4, \{x_4(t_c)\}_{c=1}^C\}$. Using time-dependent adjoint states $\{\lambda_i\}_{i=1}^4$ for the dynamics constraints, and (time-independent) duals $\nu_{c,i}$ for the continuity constraints, we derive the adjoint system of equations from the unconstrained Lagrangian as

$$\dot{\lambda}_1 = 0,\ \dot{\lambda}_2 = -\lambda_1,\ \dot{\lambda}_3 = -\lambda_2,\ \dot{\lambda}_4 = -\lambda_3 \ . \tag{7}$$

The gradients *w.r.t.* the initial conditions $\{x_i(0)\}_{i=1}^4$ are

$$\begin{aligned} \nabla_{x_1(0)} E = -\lambda_1(0), \quad \nabla_{x_2(0)} E = -\lambda_2(0), \\ \nabla_{x_3(0)} E = -\lambda_3(0), \quad \nabla_{x_4(0)} E = -\lambda_4(0) \ . \end{aligned} \tag{8}$$

The *jerks* (*i.e.*, rate of acceleration change) at $x_4(t_c)$ are updated using $\nabla_{x_4(t_c)} E = -\lambda_4(t_c)$. The values of the adjoint variables at 0 are computed by integrating backward the adjoint system starting from $\forall i : \lambda_i(1) = 0$. Note that λ_1, λ_2 and λ_3 are continuous at joints, but λ_1 jumps at the data-point location as per $\lambda_1(t_i^+) - \lambda_1(t_i^-) = 2(x_1(t_i) - y_i)$. During backward integration, λ_4 starts with zero at each t_{c+1} and the accumulated value at t_c is used for the gradient update of $x_4(t_c)$.

It is critical to note that, along the time t, such a formulation guarantees that $x_4(t)$ is piecewise constant, $x_3(t)$ is piecewise linear, $x_2(t)$ is piecewise quadratic, and $x_1(t)$ is piecewise cubic; this results in a cubic spline curve. Fig. 3 demonstrates this shooting-based spline fitting method on data in $\mathbb{R}$. While it is difficult to explain this data with one simple cubic curve, it suffices to add one control point to recover the underlying trend. The state x_4 experiences a jump at the control location that integrates up three-times to give a C^2-continuous evolution for the state x_1.

4 Regression on Riemannian Manifolds

In this section, we adopt the optimal-control perspective of Section 3 and generalize the regression problems to nonflat, smooth Riemannian manifolds. In the literature this generalization is typically referred to as *geodesic regression*. For a thorough treatment of Riemannian manifolds, we refer the reader to [37]. We remark that the term geodesic regression here does not refer to the model that is fitted but rather to the fact that the Euclidean distance in the data matching term of the energies is replaced by the geodesic distance on the manifold. In particular, the measurements $\{y_i\}_{i=0}^{N-1}$ in Euclidean space now become elements $\{Y_i\}_{i=0}^{N-1}$ on some Riemannian manifold $\mathcal{M}$ with Riemannian metric $\langle \cdot, \cdot \rangle_p$ at $p \in \mathcal{M}$[3]. The geodesic distance, induced by this metric, will be denoted as d_g. We also replace t_i with r_i, indicating that the independent value does not have to be *time*, but can also represent other entities, such as counts or speed.

Our first objective is to estimate a geodesic $\gamma : \mathbb{R} \to \mathcal{M}$, represented by initial point $\gamma(r_0)$ and initial velocity $\dot{\gamma}(r_0)$ at the tangent space $\mathcal{T}_{\gamma(r_0)}\mathcal{M}$, *i.e.*,

$$\begin{aligned} \min_{\boldsymbol{\Theta}} E(\boldsymbol{\Theta}) = \alpha \underbrace{\int_0^1 \langle \dot{\gamma}, \dot{\gamma} \rangle_{\gamma(r)} \, dr}_{\text{Regularity}} + \frac{1}{\sigma^2} \underbrace{\sum_{i=0}^{N-1} d_g^2(\gamma(r_i), Y_i)}_{\text{Data-matching}} \\ \text{subject to} \quad \nabla_{\dot{\gamma}} \dot{\gamma} = 0 \text{ (geodesic equation)} \ , \end{aligned} \tag{9}$$

with $\boldsymbol{\Theta} = \{\gamma(0), \dot{\gamma}(0)\}$ and ∇ denoting the Levi-Civita connection on $\mathcal{M}$. The covariant derivative $\nabla_{\dot{\gamma}} \dot{\gamma}$ of value 0 ensures that the curve is a geodesic. The parameters $\alpha \geq 0$ and $\sigma > 0$ balance the regularity and the data-matching term. In the Euclidean case, there is typically no regularity term because we usually do not have prior knowledge about the slope. Similarly, on Riemannian manifolds we may penalize the initial velocity by choosing $\alpha > 0$; but typically, α is also set to 0. The regularity term on the velocity can be further reduced to a smoothness penalty at r_0, *i.e.*, $\int_0^1 \langle \dot{\gamma}, \dot{\gamma} \rangle dr = \langle \dot{\gamma}(r_0), \dot{\gamma}(r_0) \rangle$, because of the energy conservation along the geodesic. Also, since the geodesic is represented by the initial conditions $(\gamma(r_0), \dot{\gamma}(r_0))$, we can move along the geodesic and estimate the point $\gamma(r_i)$ that corresponds to Y_i.

4.1 Optimization via geodesic shooting

Taking the optimal-control point of view, the second-order problem of Eq. (9) can be written as a system of first-order, upon the introduction of auxiliary states

$$X_1(r) = \gamma(r), \quad \text{and} \quad X_2(r) = \dot{\gamma}(r) \ . \tag{10}$$

Here, X_1 corresponds to the *intercept* and X_2 corresponds to the *slope* in classic linear regression. Considering the simplified smoothness penalty of the previous section, the constrained minimization of Eq. (9) reduces to

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & E(\boldsymbol{\Theta}) = \alpha \langle X_2(r_0), X_2(r_0) \rangle + \\ & \frac{1}{\sigma^2} \sum_{i=0}^{N-1} d_g^2(X_1(r_i), Y_i) \\ \text{subject to} \quad & \nabla_{X_2} X_2 = 0 \ , \end{aligned} \tag{11}$$

with $\boldsymbol{\Theta} = \{X_i(r_0)\}_{i=1}^2$. Note that $X_1(r_i)$ is the estimated point on the geodesic at r_i, obtained by shooting forward with $X_1(r_0)$ and $X_2(r_0)$. Analogously to the elaborations of previous sections, we convert Eq. (11) to an unconstrained minimization problem via time-dependent

3. We omit the subscript p when it is clear from the context.

adjoint variables, then take variations with respect to its arguments and eventually get (1) dynamical systems of states and adjoint variables, (2) boundary conditions on the adjoint variables, and (3) gradients with respect to initial conditions. By shooting forward / backward and updating the initial states via the gradients, we can obtain a numerical solution to the problem.

4.2 Time-warped regression

The time-warping strategy of Section 3.2 can also be adapted to Riemannian manifolds, because it focuses on warping the axis of the independent scalar-valued variable, not the axis of the dependent manifold-valued variable. In other words, the time-warped model is independent of the underlying type of space. Formally, given a warping function f (*cf.* Section 3.2), all instances of the form $X_j(r_i)$ in Eq. (11) are replaced by $X_j(f(r_i;\boldsymbol{\theta}))$ for $j = 1, 2$. While the model retains its simplicity, *i.e.*, we still fit geodesic curves, the warping function allows for increased modeling flexibility.

Since we have an existing solution to the problem of fitting geodesic curves, the easiest way to minimize the resulting energy is by alternating optimization, similar to Section 3.2. This requires the derivative of the energy with respect to $\boldsymbol{\theta}$ for fixed $X_i(\overline{r})$. Application of the chain rule and [20, Appendix A] yields

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} E = {} & 2\alpha\langle \dot{X}_2(f(r_0;\boldsymbol{\theta})), X_2(f(r_0;\boldsymbol{\theta}))\rangle \nabla_{\boldsymbol{\theta}} f(r_0;\boldsymbol{\theta}) \\ & - \frac{2}{\sigma^2}\sum_{i=0}^{N-1} \langle \mathrm{Log}_{X_1(f(r_i;\boldsymbol{\theta}))} Y_i, \dot{X}_1(f(r_i;\boldsymbol{\theta}))\rangle \nabla_{\boldsymbol{\theta}} f(r_i;\boldsymbol{\theta})\end{aligned} \tag{12}$$

where $\mathrm{Log}_{X_1(f(r_i;\boldsymbol{\theta}))} Y_i$ denotes the Riemannian log-map, *i.e.*, the initial velocity of the geodesic connecting $X_1(f(r_i;\boldsymbol{\theta}))$ and Y_i in unit time and $\dot{X}_1(f(r_i;\boldsymbol{\theta}))$ is the velocity of the regression geodesic at the warped-time point. This leaves to choose a good parametric model for $f(r;\boldsymbol{\theta})$. As we require the time warp to be diffeomorphic, we choose a parametric model which is diffeomorphic by construction. One possible choice is the generalized logistic function [38], *e.g.*, with asymptotes 0 for $r \to -\infty$ and 1 for $r \to \infty$, given by

$$f(r;\boldsymbol{\theta}) = \frac{1}{(1+\beta e^{-k(r-M)})^{1/m}} \ , \tag{13}$$

with $\boldsymbol{\theta} = (k, M, \beta, m)$. The parameter k controls the growth rate, M is the time of maximum growth if $\beta = m$, β and m define the value of f at $t = M$, and $m > 0$ affects the asymptote of maximum growth. In our case, we fix $(\beta, m) = (1, 1)$ and only optimize for (k, M). By using this function, we map the original infinite time interval to a warped time-range from 0 to 1. In summary, the algorithm using alternating optimization is as follows:

0) Initialize $\boldsymbol{\theta}$ such that the warped time is evenly distributed within (0, 1).
1) Compute $\{\overline{r}_i = f(r_i;\boldsymbol{\theta})\}_{i=0}^{N-1}$ and perform standard geodesic regression using the new time-points.
2) Update $\boldsymbol{\theta}$ by numerical optimization using the gradient given in Eq. (12).
3) Check convergence. If not converged goto 1).

4.3 Cubic spline regression

Similar to Section 3.3, cubic curves on a Riemannian manifold $\mathcal{M}$ can be defined as solutions to the variational problem of minimizing an acceleration-based energy. The notion of acceleration is defined using the covariant derivatives on Riemannian manifolds [24], [25]. In particular, we define a *time-dependent control*, *i.e.*, a forcing variable $X_3(r)$, as

$$X_3(r) = \nabla_{X_2(r)} X_2(r) = \nabla_{\dot{X}_1(r)} \dot{X}_1(r) \ . \tag{14}$$

We can interpret $X_3(r)$ as a control that forces the curve $X_1(r)$ to deviate from being a geodesic [39] (which is the case if $X_3(r) = 0$). As an end-point problem, a Riemannian cubic curve is thus defined by the curve $X_1(r)$ such that it minimizes an energy of the form

$$E(X_1) = \frac{1}{2}\int_0^1 \|\nabla_{\dot{X}_1} \dot{X}_1\|^2 dt, \tag{15}$$

where the norm $\|\cdot\|$ is induced by the metric on $\mathcal{M}$ at X_1. In Section 5.5, this concept will be adapted to the Grassmannian to enable regression with cubic splines.

5 Regression on the Grassmannian

The Grassmannian is a type of Riemannian manifold where the geodesic distance, parallel transport, as well as the Riemannian log-/exp-map are relatively simple to compute (see [2] and suppl. material). Before specializing our three regression models to this manifold, we first discuss its Riemannian structure in Section 5.1 (see [40] for details) and review how different types of data can be represented on the Grassmannian in Section 5.2.

5.1 Riemannian structure of the Grassmannian

The *Grassmann* manifold $\mathcal{G}(p, n)$ is defined as the set of p-dimensional linear subspaces of $\mathbb{R}^n$, typically represented by an orthonormal matrix $\mathbf{Y} \in \mathbb{R}^{n\times p}$, such that $\mathcal{Y} = \mathrm{span}(\mathbf{Y})$ for $\mathcal{Y} \in \mathcal{G}(p, n)$. It can equivalently be defined as a quotient space within the special orthogonal group $\mathcal{SO}(n)$ as $\mathcal{G}(p, n) := \mathcal{SO}(n)/(\mathcal{SO}(n-p) \times \mathcal{SO}(p))$. The *canonical metric* $g_{\mathcal{Y}} : \mathcal{T}_{\mathcal{Y}}\mathcal{G}(p, n) \times \mathcal{T}_{\mathcal{Y}}\mathcal{G}(p, n) \to \mathbb{R}$ on $\mathcal{G}(p, n)$ is given by

$$g_{\mathcal{Y}}(\boldsymbol{\Delta}_{\mathcal{Y}}, \boldsymbol{\Delta}_{\mathcal{Y}}) = \mathrm{tr}\ \boldsymbol{\Delta}_{\mathcal{Y}}^\top \boldsymbol{\Delta}_{\mathcal{Y}} = \mathrm{tr}\ \mathbf{C}^\top(\mathbf{I}_n - \mathbf{Y}\mathbf{Y}^\top)\mathbf{C} \ , \tag{16}$$

where $\mathbf{I}_n$ denotes the $n \times n$ identity matrix, $\mathcal{T}_{\mathcal{Y}}\mathcal{G}(p, n)$ is the tangent space at $\mathcal{Y}$ and $\mathbf{C} \in \mathbb{R}^{n\times p}$ is arbitrary. Under this choice of metric, the arc-length of the geodesic connecting two subspaces $\mathcal{Y}, \mathcal{Z} \in \mathcal{G}(p, n)$ is related to the *canonical angles* $\phi_1, \dots \phi_p \in [0, \pi/2]$ between $\mathcal{Y}$ and $\mathcal{Z}$ as $d_g^2(\mathcal{Y}, \mathcal{Z}) = ||\boldsymbol{\phi}||_2^2$. In what follows, we slightly change notation and use $d_g^2(\mathbf{Y}, \mathbf{Z})$, with $\mathcal{Y} = \mathrm{span}(\mathbf{Y})$ and $\mathcal{Z} = \mathrm{span}(\mathbf{Z})$. In fact, the (squared) geodesic distance can be computed from the SVD decomposition $\mathbf{U}(\cos\boldsymbol{\Sigma})\mathbf{V}^\top = \mathbf{Y}^\top\mathbf{Z}$ as $d_g^2(\mathbf{Y}, \mathbf{Z}) = ||\cos^{-1}(\mathrm{diag}\,\boldsymbol{\Sigma})||^2$ (cf. [2]), where $\boldsymbol{\Sigma}$ is diagonal with principal angles ϕ_i.

Finally, consider a curve $\gamma : [0, 1] \to \mathcal{G}(p, n), r \mapsto \gamma(r)$ such that $\gamma(0) = \mathcal{Y}_0$ and $\gamma(1) = \mathcal{Y}_1$, with $\mathcal{Y}_0$ represented

Algorithm 1: Standard Grassmannian geodesic regression (Std-GGR)

Data: $\{(r_i, \mathbf{Y}_i)\}_{i=0}^{N-1}$, α and σ^2
Result: $\mathbf{X}_1(r_0)$, $\mathbf{X}_2(r_0)$
Set initial $\mathbf{X}_1(r_0)$ and $\mathbf{X}_2(r_0)$, *e.g.*, $\mathbf{X}_1(r_0) = \mathbf{Y}_0$, and $\mathbf{X}_2(r_0) = \mathbf{0}$.
while *not converged* **do**
  Solve Eqs. (19) with $\mathbf{X}_1(r_0)$ and $\mathbf{X}_2(r_0)$ forward for $r \in [r_0, r_{N-1}]$.
  Solve $\begin{cases} \dot{\lambda}_1 = \lambda_2 \mathbf{X}_2^\top \mathbf{X}_2, \ \lambda_1(r_{N-1}+) = 0, \\ \dot{\lambda}_2 = -\lambda_1 + \mathbf{X}_2(\lambda_2^\top \mathbf{X}_1 + \mathbf{X}_1^\top \lambda_2), \ \lambda_2(r_{N-1}) = 0 \end{cases}$ backward with jump conditions
  $\lambda_1(r_i-) = \lambda_1(r_i+) - \frac{1}{\sigma^2} \nabla_{\mathbf{X}_1(r_i)} d_g^2(\mathbf{X}_1(r_i), \mathbf{Y}_i)$, and $\nabla_{\mathbf{X}_1(r_i)} d_g^2(\mathbf{X}_1(r_i), \mathbf{Y}_i)$ computed as $-2 \operatorname{Log}_{\mathbf{X}_1(r_i)} \mathbf{Y}_i$. For multiple measurements at a given r_i, the jump conditions for each measurement are added up.
  Compute gradients with respect to initial conditions:

$$\begin{aligned} \nabla_{\mathbf{X}_1(r_0)} E &= -(\mathbf{I}_n - \mathbf{X}_1(r_0)\mathbf{X}_1(r_0)^\top)\lambda_1(r_0-) + \mathbf{X}_2(r_0)\lambda_2(r_0)^\top \mathbf{X}_1(r_0), \\ \nabla_{\mathbf{X}_2(r_0)} E &= 2\alpha \mathbf{X}_2(r_0) - (\mathbf{I}_n - \mathbf{X}_1(r_0)\mathbf{X}_1(r_0)^\top)\lambda_2(r_0). \end{aligned}$$

  Use a line search with these gradients to update $\mathbf{X}_1(r_0)$ and $\mathbf{X}_2(r_0)$ (see suppl. material).
end

by $\mathbf{Y}_0$ and $\mathcal{Y}_1$ represented by $\mathbf{Y}_1$. The *geodesic equation* for such a curve, given that $\dot{\mathbf{Y}} = {}^d\!/_{dr}\mathbf{Y}(r) \doteq (\mathbf{I}_n - \mathbf{Y}\mathbf{Y}^\top)\mathbf{C}$, on $\mathcal{G}(p, n)$ is given by

$$\ddot{\mathbf{Y}}(r) + \mathbf{Y}(r)[\dot{\mathbf{Y}}(r)^\top \dot{\mathbf{Y}}(r)] = \mathbf{0} \ , \tag{17}$$

which also defines the Riemannian exponential map on the Grassmannian as an ODE for convenient numerical computations. Integrating Eq. (17), starting with initial conditions, "shoots" the geodesic forward in time.

5.2 Representation on the Grassmannian

We describe two types of data that can be represented on $\mathcal{G}(p, n)$: linear dynamical systems (LDS) and shapes.

Linear dynamical systems. In the computer vision literature, *dynamic texture* models [41] are commonly applied to model videos as realizations of linear dynamical systems (LDS). For a video, represented by a collection of vectorized frames $\mathbf{y}_1, \ldots, \mathbf{y}_\tau$ with $\mathbf{y}_i \in \mathbb{R}^n$, the standard dynamic texture model with p states has the form

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{w}_k, & \mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{W}), \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, & \mathbf{v}_k &\sim \mathcal{N}(0, \mathbf{R}) \ , \end{aligned} \tag{18}$$

with $\mathbf{x}_k \in \mathbb{R}^p, \mathbf{A} \in \mathbb{R}^{p \times p}$, and $\mathbf{C} \in \mathbb{R}^{n \times p}$. When relying on the prevalent system identification of [41], the matrix $\mathbf{C}$ is, by design, of (full) rank p (*i.e.*, the number of states) and by construction we obtain an *observable* system, where a full rank *observability* matrix $\mathbf{O} \in \mathbb{R}^{np \times p}$ is defined as $\mathbf{O} = [\mathbf{C} \ (\mathbf{C}\mathbf{A}) \ (\mathbf{C}\mathbf{A}^2) \ \cdots \ (\mathbf{C}\mathbf{A}^{p-1})]^\top$. This system identification is not unique because systems $(\mathbf{A}, \mathbf{C})$ and $(\mathbf{T}\mathbf{A}\mathbf{T}^{-1}, \mathbf{C}\mathbf{T}^{-1})$ with $\mathbf{T} \in \mathcal{GL}(p)$ have the same transfer function. Hence, the realization subspace spanned by $\mathbf{O}$ is a point on the Grassmannian $\mathcal{G}(p, n)$ and the observability matrix is a representer of this subspace. We identify an LDS model for a video by its $np \times p$ orthonormalized observability matrix.

Shapes. Let a shape be represented by a collection of m landmarks. A *shape matrix* is constructed from its m landmarks as $\mathbf{L} = [(x_1, y_1, \ldots); (x_2, y_2, \ldots); \ldots; (x_m, y_m, \ldots)]$. Using SVD on this matrix, *i.e.*, $\mathbf{L} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, we obtain an affine-invariant shape representation from the left-singular vectors $\mathbf{U}$ [17], [42]. This establishes a mapping from the shape matrix to a point on the Grassmannian (with $\mathbf{U}$ as the representative). Such a representation has been used for facial aging regression for instance [43].

5.3 Standard geodesic regression

We start by adapting the inner-product and the squared geodesic distance in Eq. (9) to $\mathcal{G}(p, n)$. Given the auxiliary states of Eq. (10), now denoted as matrices $\mathbf{X}_1$ (initial point) and $\mathbf{X}_2$ (velocity), we can write the geodesic equation of Eq. (17) as a system of first-order dynamics:

$$\dot{\mathbf{X}}_1 = \mathbf{X}_2, \quad \text{and} \quad \dot{\mathbf{X}}_2 = -\mathbf{X}_1(\mathbf{X}_2^\top \mathbf{X}_2) \ . \tag{19}$$

For a point on $\mathcal{G}(p, n)$ it should further hold that (1) $\mathbf{X}_1(r)^\top \mathbf{X}_1(r) = \mathbf{I}_p$ and (2) the velocity at $\mathbf{X}_1(r)$ needs to be orthogonal to that point, *i.e.*, $\mathbf{X}_1(r)^\top \mathbf{X}_2(r) = \mathbf{0}$. If we enforce these two constraints at the starting point r_0, they will remain satisfied along the geodesic. This yields

$$\begin{aligned} \min_{\mathbf{\Theta}} \quad & E(\mathbf{\Theta}) = \alpha \ \text{tr} \ \mathbf{X}_2(r_0)^\top \mathbf{X}_2(r_0) + \\ & \frac{1}{\sigma^2} \sum_{i=0}^{N-1} d_g^2(\mathbf{X}_1(r_i), \mathbf{Y}_i) \\ \text{subject to} \quad & \mathbf{X}_1(r_0)^\top \mathbf{X}_1(r_0) = \mathbf{I}_p, \\ & \mathbf{X}_1(r_0)^\top \mathbf{X}_2(r_0) = \mathbf{0} \text{ and Eq. (19)} \ , \end{aligned} \tag{20}$$

with $\mathbf{\Theta} = \{\mathbf{X}_i(r_0)\}_{i=1}^2$. Based on the adjoint method, we obtain the shooting solution to Eq. (20), listed in Alg. 1. Note that the jump conditions on λ_1 involve the gradient of the residual term $d_g^2(\mathbf{X}_1(r_i), \mathbf{Y}_i)$ with respect to $\mathbf{X}_1(r_i)$, *i.e.*, the base point of the residual on the fitted geodesic; this gradient is $-2 \operatorname{Log}_{\mathbf{X}_1(r_i)} \mathbf{Y}_i$, *cf.* suppl. material. We refer to this problem of fitting a geodesic as *standard Grassmannian geodesic regression (Std-GGR)*.

5.4 Time-warped regression

Since the concept of time-warped geodesic regression is generic for Riemannian manifolds, specialization to the Grassmannian is straightforward. We only need to use

Algorithm 2: Cubic-spline Grassmannian geodesic regression (CS-GGR)

Data: $\{(r_i, \mathbf{Y}_i)\}_{i=0}^{N-1}$, $\{r_c\}_{c=1}^{C}$, α and σ^2
Result: $\mathbf{X}_1(r_0)$, $\mathbf{X}_2(r_0)$, $\mathbf{X}_3(r_0)$, $\mathbf{X}_4(r_0)$, $\{\mathbf{X}_4(r_c^+)\}_{c=1}^{C}$
Set initial $\mathbf{X}_1(r_0)$ as $\mathbf{Y}_0$ for example, and $\mathbf{X}_2(r_0)$, $\mathbf{X}_3(r_0)$, $\mathbf{X}_4(r_0)$, $\{\mathbf{X}_4(r_c^+)\}_{c=1}^{C}$ as zero matrices.
while *not converged* **do**
Solve Eq. (23) forward in each interval with $\mathbf{X}_1(r_0)$, $\mathbf{X}_2(r_0)$, $\mathbf{X}_3(r_0)$, $\mathbf{X}_4(r_0)$, $\{\mathbf{X}_4(r_c^+)\}_{c=1}^{C}$, and $\{\mathbf{X}_1(r_c^+) = \mathbf{X}_1(r_c^-)$, $\mathbf{X}_2(r_c^+) = \mathbf{X}_2(r_c^-)$, $\mathbf{X}_3(r_c^+) = \mathbf{X}_3(r_c^-)\}_{c=1}^{C}$.
Solve $\begin{cases} \dot{\lambda_1} &= \lambda_2 \mathbf{X}_2^\top \mathbf{X}_2 - \lambda_3(\mathbf{X}_4^\top \mathbf{X}_1 - \mathbf{X}_3^\top \mathbf{X}_2) - \mathbf{X}_4(\lambda_3^\top \mathbf{X}_1 + \mathbf{X}_2^\top \lambda_4), \\ \dot{\lambda_2} &= -\lambda_1 + \mathbf{X}_2(\lambda_2^\top \mathbf{X}_1 + \mathbf{X}_1^\top \lambda_2 - \lambda_4^\top \mathbf{X}_3 - \mathbf{X}_3^\top \lambda_4) + \mathbf{X}_3(\lambda_3^\top \mathbf{X}_1 + \mathbf{X}_2^\top \lambda_4) + \lambda_4(-\mathbf{X}_1^\top \mathbf{X}_4 + \mathbf{X}_2^\top \mathbf{X}_3), \\ \dot{\lambda_3} &= -\lambda_2 - \lambda_4 \mathbf{X}_2^\top \mathbf{X}_2 + \mathbf{X}_2(\mathbf{X}_1^\top \lambda_3 + \lambda_4^\top \mathbf{X}_2) + 2\alpha \mathbf{X}_3, \\ \dot{\lambda_4} &= \lambda_3 - \mathbf{X}_1(\mathbf{X}_1^\top \lambda_3 + \lambda_4^\top \mathbf{X}_2) \end{cases}$ backward
with $\lambda_1(r_{N-1}) = \lambda_2(r_{N-1}) = \lambda_3(r_{N-1}) = \lambda_4(r_{N-1}) = \lambda_4(r_c^-) = 0$, and $\{\lambda_1(r_c^-) = \lambda_1(r_c^+)$, $\lambda_2(r_c^-) = \lambda_2(r_c^+)$, $\lambda_3(r_c^-) = \lambda_3(r_c^+)\}_{c=1}^{C}$, as well as jump conditions $\lambda_1(r_i^-) = \lambda_1(r_i^+) - \frac{1}{\sigma^2}\nabla_{\mathbf{X}_1(r_i)} d_g^2(\mathbf{X}_1(r_i), \mathbf{Y}_i)$, and $\nabla_{\mathbf{X}_1(r_i)} d_g^2(\mathbf{X}_1(r_i), \mathbf{Y}_i)$ computed as $-2\,\mathrm{Log}_{\mathbf{X}_1(r_i)} \mathbf{Y}_i$. For multiple measurements at a given r_i, the jump conditions for each measurement are added up.
Compute gradients with respect to initial conditions and the fourth state at control points:

$$\begin{aligned} \nabla_{\mathbf{X}_1(r_0)} E &= -(\mathbf{I}_n - \mathbf{X}_1(r_0)\mathbf{X}_1(r_0)^\top)\lambda_1(r_0^-) + \mathbf{X}_2(r_0)\lambda_2(r_0)^\top \mathbf{X}_1(r_0) + \mathbf{X}_3(r_0)\lambda_3(r_0)^\top \mathbf{X}_1(r_0), \\ \nabla_{\mathbf{X}_2(r_0)} E &= -(\mathbf{I}_n - \mathbf{X}_1(r_0)\mathbf{X}_1(r_0)^\top)\lambda_2(r_0), \qquad \nabla_{\mathbf{X}_3(r_0)} E = -(\mathbf{I}_n - \mathbf{X}_1(r_0)\mathbf{X}_1(r_0)^\top)\lambda_3(r_0), \\ \nabla_{\mathbf{X}_4(r_0)} E &= -\lambda_4(r_0), \qquad \nabla_{\mathbf{X}_4(r_c^+)} E = -\lambda_4(r_c^+),\ c = 1...C\ . \end{aligned}$$

Use a line search with these gradients to update $\mathbf{X}_1(r_0)$, $\mathbf{X}_2(r_0)$, $\mathbf{X}_3(r_0)$, $\mathbf{X}_4(r_0)$, and $\{\mathbf{X}_4(r_c^+)\}_{c=1}^{C}$.
end

the Std-GGR solution during the alternating optimization steps. By choosing the generalized logistic function of Eq. (13) to account for saturations of scalar-valued outputs, the time-warped model on $\mathcal{G}(p,n)$ can be used to capture saturation effects for which standard geodesic regression is not sensible. We refer to this strategy as *time-warped Grassmannian geodesic regression (TW-GGR)*.

5.5 Cubic spline regression

To enable cubic spline regression on the Grassmannian, we follow Section 4.3 and add the external force $\mathbf{X}_3$. In other words, we represent an acceleration-controlled curve $\mathbf{X}_1(r)$ on $\mathcal{G}(p,n)$ using a dynamic system with states $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$ such that

$$\mathbf{X}_2 = \dot{\mathbf{X}}_1, \quad \text{and} \quad \mathbf{X}_3 = \dot{\mathbf{X}}_2 + \mathbf{X}_1(\mathbf{X}_2^\top \mathbf{X}_2)\ . \tag{21}$$

Note that if $\mathbf{X}_3 = \mathbf{0}$, the second equation is reduced to the geodesic equation of Eq. (17); this indicates that the curve is *acceleration-free*. To obtain an acceleration-controlled curve, we need to solve

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & E(\boldsymbol{\Theta}) = \frac{1}{2}\int_0^1 \mathrm{tr}\ \mathbf{X}_3^\top \mathbf{X}_3\ dr \\ \text{subject to} \quad & \mathbf{X}_1^\top \mathbf{X}_1 = \mathbf{I}_p,\ \mathbf{X}_1^\top \mathbf{X}_2 = \mathbf{0}, \text{ and Eq. (21)} \end{aligned} \tag{22}$$

with $\boldsymbol{\Theta} = \{\mathbf{X}_i(r_0)\}_{i=1}^{3}$. In particular, the relaxation solution to Eq. (21) gives us (see suppl. material) the system of equations for shooting cubic curves on $\mathcal{G}(p,n)$:

$$\begin{aligned} \dot{\mathbf{X}}_1 &= \mathbf{X}_2, \\ \dot{\mathbf{X}}_2 &= \mathbf{X}_3 - \mathbf{X}_1\mathbf{X}_2^\top \mathbf{X}_2, \\ \dot{\mathbf{X}}_3 &= -\mathbf{X}_4 + \mathbf{X}_1\mathbf{X}_1^\top \mathbf{X}_4 - \mathbf{X}_1\mathbf{X}_2^\top \mathbf{X}_3, \\ \dot{\mathbf{X}}_4 &= \mathbf{X}_3\mathbf{X}_2^\top \mathbf{X}_2 + \mathbf{X}_2\mathbf{X}_4^\top \mathbf{X}_1 - \mathbf{X}_2\mathbf{X}_3^\top \mathbf{X}_2\ . \end{aligned} \tag{23}$$

It is important to note that $\mathbf{X}_1$ does not follow a geodesic path under non-zero force $\mathbf{X}_3$. Hence, the constraints $\mathbf{X}_1(r)^\top \mathbf{X}_1(r) = \mathbf{I}_p$ and $\mathbf{X}_1(r)^\top \mathbf{X}_2(r) = \mathbf{0}$ should be enforced at *every* instance of r to keep the path on the manifold. However, we can show (see suppl. material) that enforcing $\mathbf{X}_1(r)^\top \mathbf{X}_2(r) = \mathbf{0}$ at all times already guarantees that $\mathbf{X}_1(r)^\top \mathbf{X}_1(r) = \mathbf{I}_p$ if this holds initially at $r = 0$. Also, $\mathbf{X}_1(r)^\top \mathbf{X}_2(r) = \mathbf{0}$ implies that $\mathbf{X}_1(r)^\top \mathbf{X}_3(r) = \mathbf{0}$. By using this fact during relaxation, the constraints are already implicitly captured in Eqs. (23). Subsequently, for shooting we only need to guarantee that all these constraints hold initially. To get a cubic spline curve, we follow Section 3.3.3 and introduce control points $\{r_c\}_{c=1}^{C}$, which divide the support of the independent variable into several intervals $\mathcal{I}_c$. The first three states should be continuous at the control points, but the state $\mathbf{X}_4$ is allowed to jump. Hence, the spline regression problem on $\mathcal{G}(p,n)$ becomes, *cf.* Eq. (6),

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & E(\boldsymbol{\Theta}) = \alpha \int_{r_0}^{r_{N-1}} \mathrm{tr}\ \mathbf{X}_3^\top \mathbf{X}_3\ dr\ + \\ & \frac{1}{\sigma^2}\sum_{i=0}^{N-1} d_g^2(\mathbf{X}_1(r_i), \mathbf{Y}_i) \\ \text{subject to} \quad & \mathbf{X}_1(r_0)^\top \mathbf{X}_1(r_0) = \mathbf{I}_p, \\ & \mathbf{X}_1(r_0)^\top \mathbf{X}_2(r_0) = \mathbf{0}, \\ & \mathbf{X}_1(r_0)^\top \mathbf{X}_3(r_0) = \mathbf{0}, \\ & \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \text{ are continuous at } \{r_c\}_{c=1}^{C}, \\ & \text{and Eqs. (23) holds in each } \mathcal{I}_c\ , \end{aligned} \tag{24}$$

with $\boldsymbol{\Theta} = \{\{\mathbf{X}_i(r_0)\}_{i=1}^{4}, \{\mathbf{X}_4(r_c^+)\}_{c=1}^{C}\}$. Alg. 2 lists the shooting solution to Eq. (24), referred to as *cubic-spline Grassmannian geodesic regression (CS-GGR)*.

Method	$\mathbf{X}_1(r_0)$	$\mathbf{X}_2(r_0)$	$\mathbf{X}_3(r_0)$	$\mathbf{X}_4(r_0)$	k	M	GT *vs.* Data	Data *vs.* Est.	**GT *vs.* Est.**
Std-GGR	0.02	0.16	–	–	–	–	0.7e-2	0.7e-2	0.3e-3
Rentmeesters [28]	0.02	0.16	–	–	–	–	0.7e-2	0.6e-2	0.3e-3
TW-GGR	0.02	0.16	–	–	0.05	0.6e-2	6.9e-3	6.6e-3	0.3e-3
CS-GGR	0.07	0.54	0.36	0.97	–	–	6.8e-3	5.8e-3	1.1e-3
Su *et al.* [16]	–	–	–	–	–	–	6.8e-3	8.2e-3	3.5e-3

TABLE 1: Comparison of the regression results on synthetic data. *First*, we report differences in the initial conditions $\mathbf{X}_i(r_0)$: for $\mathbf{X}_1$, we report the geodesic distance on the Grassmannian; for $\mathbf{X}_2$, $\mathbf{X}_3$ and $\mathbf{X}_4$, we report $\|\mathbf{X}_i^{Est.} - \mathbf{X}_i^{GT}\|_F / \|\mathbf{X}_i^{GT}\|_F$. For multiple $\mathbf{X}_4 s$, we take the average. For TW-GGR, we also report the difference in the parameters of the time-warp function (k, M). *Second*, we report the mean squared (geodesic) distance (MSD) between two curves. In particular, we compute (1) the MSD between the data points and the corresponding points on the ground truth (GT) curves (GT *vs.* Data); (2) the MSD between the data points and the points on the estimated regression curves (Data *vs.* Est.) and (3) the MSD between the points on the ground truth curves and the data points on the estimated regression curves (Data *vs.* Est.). The second row shows a comparison to [28] (conceptually similar to [29]). The last row lists the (best) MSDs for the approach of Su *et al.* [16] on the data used to test CS-GGR (for $\lambda_1/\lambda_2 = 10$).

6 Experiments on Synthetic Data

We first demonstrate Std-GGR, TW-GGR and CS-GGR on synthetic data and compare against two approaches from the literature [28], [16].

Each data point in the following experiment represents a 2D sine / cosine signal, sampled at 630 evenly-spaced locations in $[0, 10\pi]$. These signals $\mathbf{s} \in \mathbb{R}^{2\times 630}$ are then linearly projected into $\mathbb{R}^{24}$ via $\bar{\mathbf{s}} = \mathbf{U}\mathbf{s}$, where $\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{24})$ and $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. White Gaussian noise with $\sigma = 0.1$ is added to $\bar{\mathbf{s}}$. For each embedded signal $\bar{\mathbf{s}} \in \mathbb{R}^{24\times 630}$, we estimate a two-state (*i.e.*, $p = 2$) LDS as discussed in Section 5.2, and use the corresponding observability matrix to represent it as a point on $\mathcal{G}(2, 48)$. Besides, each data point has an associated scalar value; this independent variable is uniformly distributed within $(0, 10)$ and controls the *signal frequency* of the data point. For Std-GGR, we directly use this value as the signal frequency to generate 2D signals, while for TW-GGR and CS-GGR, a generalized logistic function or a sine function is adopted to convert the values to a signal frequency for data generation. It is important to note that the *largest* eigenvalue of the state-transition matrix $\mathbf{A}$ reflects the frequency of the sine / cosine signal.

To quantitatively assess the quality of the fitting results, we design a "denoising" experiment. The data to be used for denoising is generated as follows: First, we use each regression method to estimate a model from the (clean) data points we just generated. In the second step, we take the initial conditions of each model, shoot forward and record the points along the regression curve at fixed values of the independent variable (*i.e.*, the signal frequency). These points serve as our *ground truth (GT)*. In a final step, we take each point on the ground truth curve, generate a random tangent vector at each location and shoot forward along that vector for a small time (*e.g.*, 0.03). The newly generated points then serve as the "noisy" measurements of the original points.

To obtain fitting results on the noisy data, we initialize the first state $\mathbf{X}_1$ with the first data point, and all other initial conditions with $\mathbf{0}$. Table 1 lists the differences between our estimated regression curves (Est.) and the corresponding ground truth using two strategies: (1) comparison of the initial conditions as well as the parameters of the warping function in TW-GGR; (2) comparison of the full curves (sampled at the values of the independent variable) and the data points. The numbers indicate that all three models allow us to capture different types of relationships on $\mathcal{G}(2, 48)$. We compare to [28] which is a representative for Jacobi field based parametric regression (see also [29]). Since this approach fits a geodesic and returns an initial point and a velocity vector (as in Std-GGR), we report the same quantitative measures in Table 1. As expected, we essentially obtain the same solution, since the same energy is minimized.

In the context of fitting cubic splines, we compare CS-GGR against the discrete curve fitting approach of Su *et al.* [16], adapted to $\mathcal{G}(p, n)$. Since, [16] does not output the fitted curve in parametric form, but as a collection of points sampled along the sought-for curve, Table 1 only reports performance measures computed from sample points. Additionally, we assess performance by adopting a different evaluation protocol. In particular, we take the observability matrices of the linear dynamical systems estimated from each $\bar{\mathbf{s}}$ as our ground truth. We then perturb the signal frequency with Gaussian noise, estimate new dynamical systems and eventually run [16] and CS-GGR on the observability matrices of these systems. For evaluation, we report the *mean absolute error (MAE)* in the largest eigenvalue of the state-transition matrix $\mathbf{A}$ to the ground truth. Fig. 4 shows a visualization of the (real-part) of the largest eigenvalue for different levels of noise. The data matching / smoothing balance for [16] was set to $(\lambda_1, \lambda_2) = (1, 0.1)$[4]. As we see from Fig. 4, the numeric results are fairly similar between both strategies. However, CS-GGR is guaranteed to return a curve with a smooth change in momentum, whereas controlling data-matching *vs.* smoothness in [16] can lead to instantaneous momentum changes at the sampling locations. Further, storage complexity of our approach scales with the number of control-points, whereas storage complexity of [16] scales with the the number of sampled points, highlighting one advantage of fitting parametric models with respect to storage requirements. Finally, we remark that we can generate arbitrarily many points along our parametric curves *after* fitting. In contrast, discrete curve fitting strategies would require re-estimation of the curve once the number of desired samples increases.

4. Additional results can be found in the suppl. material.

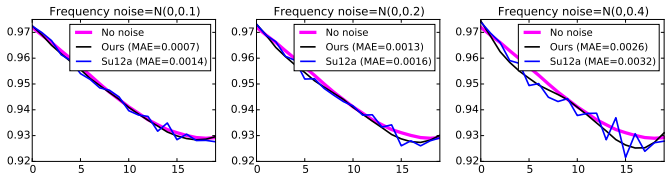

Fig. 4: CS-GGR (1 control point) *vs.* Su *et al.* [16] ($\lambda_1/\lambda_2 = 10$) in terms of the the largest eigenvalue of the state-transition matrix $\mathbf{A}$ of Eq. (18) (reconstructed from the observability matrices that we obtain along each path) to the ground truth.

Fig. 5: Corpora callosa (with the subject's age) [29].

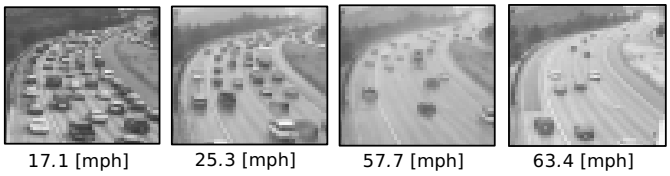

Fig. 6: Examples of the UCSD traffic dataset [44].

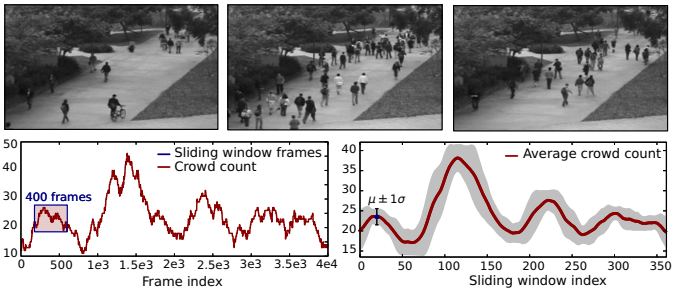

Fig. 7: *Top*: Example frames from the UCSD pedestrian dataset [45]. *Bottom*: Total crowd count over all frames (left), and average people count over a 400-frame sliding window (right).

7 Applications

To demonstrate Std-GGR, TW-GGR and CS-GGR on actual vision data, we present four applications: in the first two applications, we regress the manifold-valued variable, *i.e.*, landmark-based shapes; in the last two applications, we predict the independent variable based on the regression curve fitted to the manifold-valued data, *i.e.*, LDS representations of surveillance videos.

7.1 Datasets

Corpus callosum shapes [29]. We use a collection of 32 corpus callosum shapes with ages varying from 19 to 90 years, see Fig. 5. Each shape is represented by $m = 64$ 2D boundary landmarks, and is projected to a point on the Grassmannian using the representation of Section 5.2.

Rat calvarium landmarks [46]. We use 18 individuals with 8 time points from the Vilmann rat data, each in the age range of 7 to 150 days. Each shape is represented by a set of 8 landmarks. Fig. 9 (left) shows a selection of the landmarks projected onto the Grassmannian, using the same representation as the corpus callosum data.

UCSD traffic dataset [44]. This dataset was introduced in the context of clustering traffic flow patterns with LDS models. It contains a collection of short traffic video clips, acquired by a surveillance system monitoring highway traffic. There are 253 videos in total and each video is roughly matched to the speed measurements from a highway-mounted speed sensor. We use the pre-processed video clips introduced in [44] which were converted to grayscale and spatially normalized to 48×48 pixels with zero mean and unit variance. Our rationale for using an LDS representation for speed prediction is the fact that clustering and categorization experiments in [44] showed compelling evidence that dynamics are indicative of the traffic class. We argue that the notion of speed of an object (*e.g.*, a car) could be considered a property that humans infer from its visual dynamics.

UCSD pedestrian dataset [45]. We use the `Peds1` subset which contains 4000 frames with a ground-truth people count associated with each frame, see Fig. 7. Similar to [45] we ask the question whether we can infer the number of people in a scene (or clip) without actually detecting the people. While this problem has been addressed by resorting to crowd / motion segmentation and Gaussian process regression on low-level features extracted from the segmentation regions, we go one step further and try to avoid any preprocessing at all. In fact, our objective is to infer an *average* people count from an LDS representation of short video segments (*i.e.*, within a temporal sliding window). This is plausible because the visual dynamics of a scene change as people appear in it. In fact, it could be considered as another form of "traffic". Further, an LDS does not only model the dynamics, but also the appearance of videos; both aspects are represented in the observability matrix. However, such a strategy does not allow for fine-grain frame-by-frame predictions as in [45]. Yet, it has the advantages of not requiring any pre-selection of features or potentially unstable preprocessing steps such as the aforementioned crowd segmentation. In our setup, we split the 4000 frames into 37 video clips via a sliding window of size 400, shifted by 100 frames (see Fig. 7), and associate an *average* people count with each clip. The clips are spatially down-sampled to 60×40 pixel (original: 238×158) to keep the observability matrices at a reasonable size. Since the overlap between the clips potentially biases the experiments, we introduce a weighted variant of system identification (see suppl. material) with weights based on a Gaussian function centered at the middle of the sliding window and a standard deviation of 100. While this ensures stable system identification, by still using 400 frames, it reduces the impact of the overlapping frames on the parameter estimates. With this strategy, the average crowd count is localized to a smaller region.

General settings. In all experiments, α in the energy function is set to 0, σ to 1, the initial point is set to be the first data point, and all other initial conditions are set to zero. For the parameter(s) $\boldsymbol{\theta}$ of TW-GGR, we fix $(\beta, m) = (1, 1)$ so that M is the time of the maximal growth. Usually, one control point is used in CS-GGR.

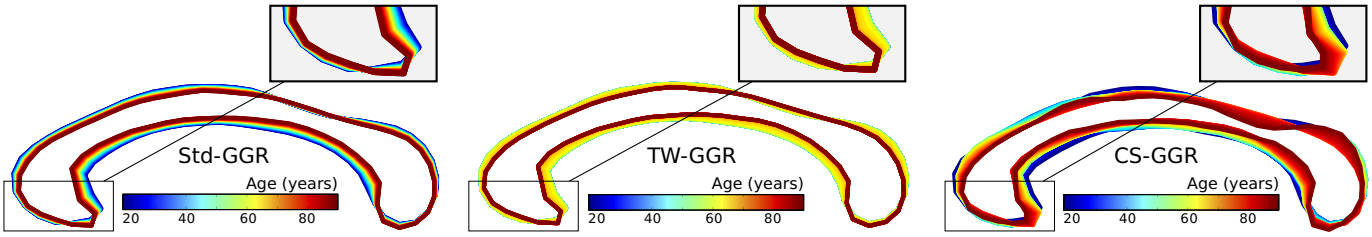

Fig. 8: Comparison between Std-GGR, TW-GGR and CS-GGR (with one control point) on the corpus callosum data [29]. The shapes are generated along the fitted curves and are colored by age (best viewed in color).

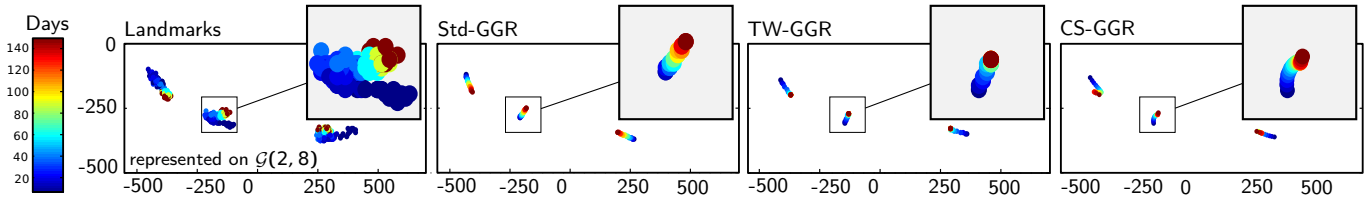

Fig. 9: Comparison between Std-GGR, TW-GGR and CS-GGR (with one control point) on the rat calvarium data (3/8 landmarks shown) [46]. The shapes are generated along the fitted curves and the landmarks are colored by age in days (best-viewed in color).

	Corpus callosum [47]					
	[28]	Std-GGR	TW-GGR	(1)CS-GGR	(2)CS-GGR	[16]
Energy	0.35	0.35	0.34	0.32	**0.31**	–
R^2	0.12	0.12	0.15	0.21	**0.23**	0.15
MSE (e-2)	1.25	1.25	**1.22**	1.36	1.43	1.25
	Rat calvarium [46]					
	[28]	Std-GGR	TW-GGR	(1)CS-GGR	(2)CS-GGR	[16]
Energy	0.32	0.32	0.18	**0.16**	**0.16**	–
R^2	0.61	0.61	0.79	0.81	0.81	**0.89**†
MSE (e-3)	2.3	2.3	1.3	**1.2**	**1.2**	4.1†

TABLE 2: Comparison of Std-GGR, TW-GGR and CS-GGR with one (1) and two (2) control points to the approaches of Rentmeesters [28] and Su *et al.* [16] (for $\lambda_1/\lambda_2 = 1/10$). For *Energy* and *MSE* smaller values are better, for R^2 larger values are better. In case of [16], we fit *one* curve to each individual in the rat calvarium data; MSE and R^2 are then averaged.

7.2 Regressing the manifold-valued variable

The first category of applications leverages the regressed relationship between the independent variable, *i.e.*, age, and the manifold-valued dependent variable, *i.e.*, shapes. *The objective is to estimate the shape for a given age*. We demonstrate Std-GGR, TW-GGR and CS-GGR on both corpus callosum and rat calvarium data. The control point for CS-GGR is set to the mean age of the subjects. Three measures are used to quantitatively compare the regression results: (1) the regression *energy*, *i.e.*, the data matching error over all observations; (2) the R^2 statistic on the Grassmannian, which is between 0 and 1, with 1 indicating a perfect fit and 0 indicating a fit no better than the Fréchet mean (see [47] for more details); and (3) the *mean squared error (MSE)* on the testing data, reported by means of (leave-one-subject-out) crossvalidation (CV). On both datasets, we compare against the approaches of Rentmeesters [28] and Su *et al.* [16]. In case of the latter approach, the data *vs*. smoothness weighting (*i.e.*, λ_1/λ_2) is chosen to achieve an MSE as close as possible (or better) to the best result of our approaches.

Corpus callosum aging. Fig. 8 shows the corpus callosum shapes along the fitted curves for the time points in the data. The shapes are recovered from the points along the curve through scaling by the mean singular values of the SVD results. Table 2 lists the quantitative measurements. With Std-GGR, the corpus callosum starts to shrink from age 19 ($= \min_i\{t_i\}_{i=1}^{32}$), which is consistent with the regression results in [47] and [30]. However, according to biological studies [48], [49], the corpus callosum size remains stable during the most active years of the lifespan, which is consistent with our TW-GGR result. As we can see from the optimized logistic function in Fig. 10 (left), TW-GGR estimates that thinning starts at ≈ 50 years, and at the age of 65, the shrinking rate reaches its peak. From the CS-GGR results, we first observe that the R^2 value increases notably to $0.21/0.23$, compared to 0.12 for Std-GGR. While this suggests a better fit to the data, it is not a fair comparison, since the number of parameters for CS-GGR increases as well and a higher R^2 value is expected. Secondly, the more interesting observation is that, qualitatively, we observe higher-order shape changes in the anterior and posterior regions of the corpus callosum, shown in the zoomed-in regions of Fig. 8; this is similar to what is reported in [30] for polynomial regression in 2D Kendall shape space. However, our shape representation, by design, easily extends to point configurations in $\mathbb{R}^3$. This is in contrast to 3D Kendall shape space which has a substantially more complex structure than its 2D variant [50]. Additionally, we notice that the result of [28] equals the result obtained via Std-GGR (as expected). For [16], the result is comparable to TW-GGR.

Rat calvarium growth. Fig. 9 (leftmost) shows the projection of the original data on $\mathcal{G}(2, 8)$, as well as (part of) the data samples generated along the fitted curves. Table

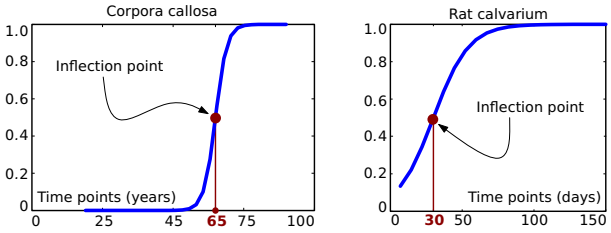

Fig. 10: Estimated time-warp functions for TW-GGR.

2 lists the performance measures. From the zoomed-in regions in Fig. 9, we observe that the rat calvarium grows at an approximately constant speed during the first 150 days if the relationship is modeled by Std-GGR. However, the estimated logistic curve for TW-GGR, shown in Fig. 10 (right), indicates that the rat calvarium only grows fast in the first few weeks, reaching its peak at 30 days; then, the rate of growth gradually levels off and becomes steady after around 14 weeks. In fact, similar growth curves for the rat skull were reported in [51]. Based on their study, the growth velocities of viscerocranium length and nurocranium width rose to the peak in the $26-32$ days period. Comparing the R^2 values for TW-GGR and CS-GGR, we see an interesting effect: although, we have more parameters in CS-GGR, the R^2 score only marginally improves. This indicates that TW-GGR already sufficiently captures the relationship between age and shape. It further confirms, to a large extent, a hypothesis from [30], where the authors noted that the cubic polynomial in 2D Kendall shape space degrades to a geodesic under polynomial time re-parametrization. Since TW-GGR re-parametrizes time (not via a cubic polynomial, but via a logistic function), it is not surprising that this relatively simple model exhibits similar performance to the more complex CS-GGR model. Similar to the corpus callosum data (and the synthetic data), [28] gives the same results as Std-GGR. For [16], we record an MSE of 4.1e-3, however, the corresponding R^2 score is higher. This can be explained, in part, by the fact that we fit *one* model per individual (as opposed to one model for all individuals) and then average the MSE and R^2 scores. This is done because [16] cannot handle multiple data instances per time point.

7.3 Predicting the independent variable

In the second category of applications the *objective is to predict the independent variable using its regressed relationship with the manifold-valued dependent variable.* Specifically, given a point on $\mathcal{G}(p,n)$, *e.g.*, an LDS representation of a video clip, we search along the regressed curve (with a step size of 0.05 in our experiments) to find its closest point, and then take the corresponding independent variable of this closest point as its predicted value. This could be considered a variant of nearest-neighbor regression where the search space is restricted to the sampled curve. The case when the search space is *not* restricted, but contains all data points, will be referred to as our *baseline*. Note that in our case, search complexity is controlled via the step-size, while the search complexity for the *baseline* scales linearly with the sample size.

Furthermore, we remark that in this category of applications, TW-GGR is not appropriate for predicting the independent variable for the following reasons: First, in case of the traffic speed measurement, the generalized logistic function tends to degenerate to almost a step-function, due to the limited number of measurement points in the central regions. In other words, two greatly different independent variables would correspond to two very close data points, even the same one, which would result in a large prediction error. Second, in case of crowd-counting, there is absolutely no prior knowledge about any saturation or growth effect which could be modeled via a logistic function. Consequently, we only demonstrate Std-GGR and CS-GGR on the two datasets. Note that prediction based on nearest neighbors could be problematic in case of CS-GGR, since the model does not guarantee a monotonic curve. We report the mean regression *energy* and the *mean absolute error (MAE)*, computed over all folds in a cross-validation setup with a dataset-dependent number of folds.

Speed prediction. For each video clip, we estimate LDS models with $p = 10$ states. The control point of CS-GGR and the breakpoint for piecewise Std-GGR is set at 50 [mph]. Results are reported for 5-fold CV, see Fig. 11. The quantitative comparison to the baseline in Table 3 shows that piecewise Std-GGR has the best performance.

Crowd counting. For each video clip, we estimate LDS models with $p = 10$ states using weighted system identification. For CS-GGR, the control point is set to a count of 23 people which separates the 37 videos into two groups of roughly equal size. Quantitative results for 4-fold CV are reported in Table 3. Fig. 12 shows the predictions *vs.* the ground truth; and both Std-GGR and CS-GGR output predictions "close" to the ground truth, mostly within 1σ (shaded region) of the average crowd count. However, a closer look at Table 3 reveals a typical overfitting effect for CS-GGR: while the training MAE is quite low, the testing MAE is higher than for the simpler Std-GGR approach. Even though both models exhibit comparable performance (considering the standard deviations), Std-GGR is preferable, due to fewer parameters and its guaranteed monotonic regression curve.

8 Discussion

In this paper, we developed a general theory for parametric regression on the Grassmann manifold from an optimal-control perspective. By introducing the basic principles for fitting models of increasing order for the special case of $\mathcal{M} = \mathbb{R}^n$, we established the framework that was then used for a generalization to Riemannian manifolds and, in particular, the Grassmann manifold.

From an application point of view, we have seen that quite different vision problems can be solved within the same framework under minimal data preprocessing. We compared our regression approaches to two alternative approaches in the literature. In comparison, we achieved similar or better performance, while providing a unified formulation and straightforward implementation.

	Traffic speed				Crowd counting			
	Baseline	Std-GGR	Std-GGR (PW)	CS-GGR	Baseline	Std-GGR	Std-GGR (PW)	CS-GGR
Mean energy	–	2554.88	**2461.95**	2670.84	–	273.81	**224.87**	244.02
Train-MAE	–	2.98 ± 0.33	$\mathbf{1.48 \pm 0.07}$	2.42 ± 0.35	–	0.97 ± 0.07	$\mathbf{0.59 \pm 0.13}$	0.63 ± 0.19
Test-MAE	4.14 ± 0.36	4.44 ± 0.16	$\mathbf{3.46 \pm 0.64}$	6.32 ± 1.62	2.40 ± 0.53	$\mathbf{1.88 \pm 0.75}$	2.14 ± 1.03	2.11 ± 0.76

TABLE 3: Mean energy and mean absolute errors (MAE) over all CV-folds $\pm 1\sigma$ on training and testing data. Comparisons to [28] and [16] were left-out, because [28] did not converge appropriately and [16] did not scale to the size of these problems.

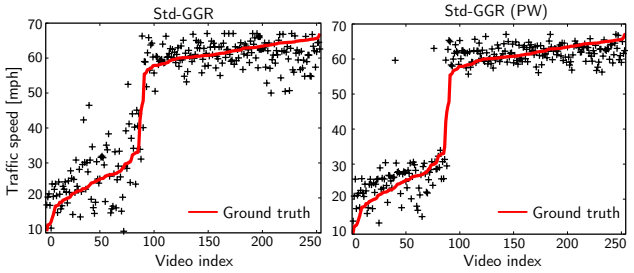

Fig. 11: Traffic speed predictions via 5-fold CV. The red solid curve shows the ground truth (best-viewed in color).

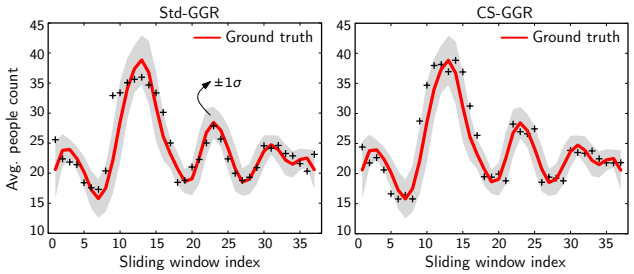

Fig. 12: Crowd counting results via 4-fold CV. Predictions are shown as a function of the sliding window index. The gray envelope indicates the weighted standard deviation ($\pm 1\sigma$) around the average crowd size in a sliding window (best-viewed in color).

Our approaches also scale better to larger problems, thereby allowing for experiments on the traffic and the pedestrian data sets. While the presented applications are limited to shape analysis and surveillance video processing, our method should be widely applicable to other problems on the Grassmann manifold, *e.g.*, domain adaptation, facial pose regression, or the recently proposed domain evolution problems.

Regarding the limitations of the proposed approach, we note that the issue of model selection is critical. In fact, whether we should use Std-GGR, TW-GGR or CS-GGR highly depends on our prior knowledge of the data. In shape regression, for instance such prior knowledge is frequently available, since the medical / biological literature already provides evidence for different growth and saturation effects as a function of age. For applications where prediction of the independent variable is of importance, *e.g.*, traffic or or crowd surveillance, we additionally have computational constraints in many cases. Interestingly, a simple geodesic curve as a model for regression can often provide sufficiently good performance, as we observed in the crowd counting experiment. We hypothesize that this can be explained, to some extent, by the fact that geodesic regression respects the geometry of the underlying space. In this space, the relationship between the dependent and the independent variable might be relatively simple to model. In contrast, approaches where video content is represented by feature vectors and conventional regression approaches with standard kernels are used, more flexible models might be needed. TW-GGR can serve as a hybrid solution when we have prior knowledge about the data; however, samples throughout the range of the independent variable are needed to avoid degenerate cases of the warping function, which could be avoided via regularization. Furthermore, the model criticism approach in [52] provides another alternative for model selection.

ACKNOWLEDGMENTS

This work was supported by NSF grants EECS-1148870 and IIS-1208522.

REFERENCES

[1] 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.
[2] 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.
[3] R. Gopalan, R. Li, and R. Chellappa, "Domain adaption for object recognition: An unsupervised approach," in *ICCV*, 2011.
[4] J. Zheng, M.-Y. Liu, R. Chellappa, and P. Phillips, "A Grassmann manifold-based domain adaption approach," in *ICML*, 2012.
[5] Y. Lui, "Human gesture recognition on product manifolds," *JMLR*, vol. 13, pp. 3297–3321, 2012.
[6] Y. Lui, J. Beveridge, and M. Kirby, "Canonical Stiefel quotient and its application to generic face recognition in illumination spaces," in *BTAS*, 2009.
[7] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, "Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2273–2285, 2011.
[8] S. Mittal and P. Meer, "Conjugate gradient descent on Grassmann manifolds for robust subspace estimation," *Image Vision Comput.*, vol. 30, pp. 417–427, 2012.
[9] H. Çetingül and R. Vidal, "Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds," in *CVPR*, 2009.
[10] J. Hamm and D. Lee, "Grassmann discriminant analysis: A unifying view on subspace learning," in *ICML*, 2008.
[11] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Optimizing over radial kernels on compact manifolds," in *CVPR*, 2014.
[12] M. Harandi, M. Salzmann, S. Jayasumana, R. Hartley, and H. Li, "Expanding the Family of Grassmannian Kernels: An Embedding Perspective," in *ECCV*, 2014.
[13] Y. Hong, R. Kwitt, N. Singh, B. Davis, N. Vasconcelos, and M. Niethammer, "Geodesic regression on the Grassmannian," in *ECCV*, 2014.
[14] Y. Hong, N. Singh, R. Kwitt, and M. Niethammer, "Time-warped geodesic regression," in *MICCAI*, 2014.
[15] L. Machado, F. Silva Leite, and K. Krakowski, "Higher-order smoothing splines versus least-squares problems on Riemannian manifolds," *J. Dyn. Control Syst.*, vol. 16, no. 1, pp. 121–148, 2010.
[16] 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.

[17] E. Begelfor and W. Werman, "Affine invariance revisited," in *CVPR*, 2006.
[18] M. Moussa and M. Cheema, "Non-parametric regression in curve fitting," *The Statistician*, vol. 41, pp. 209–225, 1998.
[19] B. Davis, P. T. Fletcher, E. Bullit, and S. Joshi, "Population shape regression from random design data," in *ICCV*, 2007.
[20] C. Samir, P. Absil, A. Srivastava, and E. Klassen, "A gradient-descent method for curve fitting on Riemannian manifolds," *Found. Comp. Math*, vol. 12, no. 1, pp. 49–73, 2012.
[21] N. Boumal and P.-A. Absil, "A discrete regression method on manifolds and its application to data on $\mathcal{SO}(n)$," in *IFAC*, 2011.
[22] ——, "Discrete regression methods on the cone of positive-definite matrices," in *ICASSP*, 2011.
[23] N. Boumal, "Interpolation and regression of rotation matrices," in *Geometric Science of Information*, ser. Lecture Notes in Computer Science, F. Nielsen and F. Barbaresco, Eds. Springer Berlin Heidelberg, 2013, vol. 8085, pp. 345–352.
[24] L. Noakes, G. Heinzinger, and B. Paden, "Cubic splines on curved spaces," *IMA J. Math. Control Info.*, vol. 6, no. 4, pp. 465–473, 1989.
[25] M. Camarinha, F. S. Leite, and P. Crouch, "Splines of class C^k on non-Euclidean spaces," *IMA J. Math. Control Info.*, vol. 12, no. 4, pp. 399–410, 1995.
[26] P. Crouch and F. S. Leite, "The dynamic interpolation problem: On Riemannian manifolds, Lie groups, and symmetric spaces," *J. Dyn. Control Syst.*, vol. 1, no. 2, pp. 177–202, 1995.
[27] E. Batzies, K. Hüper, L. Machado, and F. S. Leite, "Geometric mean and geodesic regression on Grassmannians," *Linear Algebra Appl.*, vol. 466, pp. 83–101, 2015.
[28] Q. Rentmeesters, "A gradient method for geodesic data fitting on some symmetric Riemannian manifolds," in *CDC-ECC*, 2011.
[29] P. T. Fletcher, "Geodesic regression and the theory of least squares on Riemannian manifolds," *Int. J. Comput. Vision*, vol. 105, no. 2, pp. 171–185, 2012.
[30] J. Hinkle, P. T. Fletcher, and S. Joshi, "Intrinsic polynomials for regression on Riemannian manifolds," *J. Math. Imaging Vis.*, vol. 50, pp. 32–52, 2014.
[31] M. Niethammer, Y. Huang, and F.-X. Vialard, "Geodesic regression for image time-series," in *MICCAI*, 2011.
[32] Y. Hong, S. Joshi, M. Sanchez, M. Styner, , and M. Niethammer, "Metamorphic geodesic regression," in *MICCAI*, 2012.
[33] N. Singh, J. Hinkle, S. Joshi, and P. T. Fletcher, "A vector momenta formulation of diffeomorphisms for improved geodesic regression and atlas construction," in *ISBI*, 2013.
[34] N. Singh and M. Niethammer, "Splines for diffeomorphic image regression," in *MICCAI*, 2014.
[35] S. Durrleman, X. Pennec, A. Trouvé, J. Braga, G. Gerig, and N. Ayache, "Toward a comprehensive framework for the spatiotemporal statistical analysis of longitudinal shape data," *Int. J. Comput. Vision*, vol. 103, no. 1, pp. 22–59, 2013.
[36] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The Theory of Splines and Their Applications*. Academic Press, 1967.
[37] W. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.
[38] D. Fekedulegn, M. Mac Siurtain, and J. Colbert, "Parameter estimation of nonlinear growth models in forestry," *Silva Fennica*, vol. 33, no. 4, pp. 327–336, 1999.
[39] A. Trouvé and F.-X. Vialard, "Shape splines and stochastic shape evolutions: A second order point of view," *Quart. Appl. Math*, vol. 70, no. 2, pp. 219–251, 2012.
[40] P.-A. Absil, R. Mahony, and R. Sepulchre, "Riemannian geometry of Grassmann manifolds with a view on algorithmic computation," *Acta Appl. Math.*, vol. 80, no. 2, pp. 199–220, 2004.
[41] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vision*, vol. 51, no. 2, pp. 91–109, 2003.
[42] D. Sepiashvili, J. Moura, and V. Ha, "Affine-permutation symmetry: Invariance and shape space," in *IEEE Workshop on Statistical Signal Processing*, 2003.
[43] P. Turaga, S. Biswas, and R. Chellappa, "The role of geometry for age estimation," in *ICASSP*, 2010.
[44] A. Chan and N. Vasconcelos, "Classification and retrieval of traffic video using auto-regressive stochastic processes," in *IV*, 2005.
[45] ——, "Counting people with low-level features and Bayesian regression," *IEEE Trans. Image Process.*, vol. 12, no. 4, pp. 2160–2177, 2012.
[46] F. Bookstein, "Morphometric tools for landmark data: geometry and biology," *Cambridge Univ. Press*, 1991.
[47] P. T. Fletcher, "Geodesic regression and the theory of least squares on Riemannian manifolds," *Int. J. Comput. Vision*, vol. 105, no. 2, pp. 171–185, 2013.
[48] K. Hopper, S. Patel, T. Cann, T. Wilcox, and J. Schaeffer, "The relationship of age, gender, handedness and sidedness to the size of the corpus callosum," *Acad. Radiol.*, vol. 1, pp. 243–248, 1994.
[49] S. Johnson, T. Farnworth, J. Pinkston, E. Bigler, and D. Blatter, "Corpus callosum surface area across the human adult life span: Effect of age and gender," *Brain Res. Bull.*, vol. 35, no. 4, pp. 373–377, 1994.
[50] I. Dryden and K. Mardia, *Statistical Shape Analysis*. Wiley, 1998.
[51] P. Hughes, J. Tanner, and J. Williams, "A longitudinal radiographic study of the growth of the rat skull," *J. Anat.*, vol. 127, no. 1, pp. 83–91, 1978.
[52] Y. Hong, R. Kwitt, and M. Niethammer, "Model criticism for regression on the Grassmannian," in *MICCAI*, 2015.

Yi Hong received the BS degree in 2007 from the Wuhan University, and the MS degree in 2011 from the Chinese Academy of Science, China. She is currently working towards her PhD degree in the Department of Computer Science, University of North Carolina at Chapel Hill, USA. Her main research interests include image and shape analysis, especially developing estimation models and statistical shape analysis methods in non-Euclidean spaces.

Roland Kwitt received his PhD degree in computer science from the University of Salzburg, Austria, in 2010. From 2011 to 2013 he was a scientist with Kitware's computer vision group in NC, USA. He is the recipient of a special appreciation award from the Austrian ministry of science (2005) and the MICCAI Young Scientist Award (2011). Since 2013, he holds an assistant professor position at the University of Salzburg, Austria. His research interests include computer vision, medical imaging and machine learning.

Nikhil Singh received his PhD from the Scientific Computing and Imaging Institute (SCI), University of Utah, USA, in 2013. He is currently a postdoc research associate at the Department of Computer Science, at the University of North Carolina, Chapel Hill, USA. His research focuses on statistical analysis of data with an emphasis on its geometry and topological characteristics. He develops manifold based methods of statistics to explain data that exhibit nonlinear variability and necessitate non-Euclidean analysis.

Nuno Vasconcelos received his PhD from the Massachusetts Institute of Technology in 2000. From 2000 to 2002, he was a member of the research staff at the Compaq Cambridge Research Laboratory. In 2003, he joined the Department of Electrical and Computer Engineering at the University of California, San Diego, where he is the head of the Statistical Visual Computing Laboratory. His work spans various areas, including computer vision, machine learning, signal processing, and multimedia systems.

Marc Niethammer received his PhD in Electrical and Computer Engineering from the Georgia Institute of Technology in 2004. After spending time as a research fellow at Brigham and Women's Hospital in Boston, he joined the computer science department of the University of North Carolina at Chapel Hill in 2008. He specializes in medical image analysis with a focus on image registration, shape analysis and spatiotemporal models.