

# Shape Analysis of Functional Data with Elastic Partial Matching

Darshan Bryner and Anuj Srivastava, *Fellow, IEEE*

**Abstract**—Elastic Riemannian metrics have been used successfully for statistical treatments of functional and curve shape data. However, this usage suffers from a significant restriction: the function boundaries are assumed to be fixed and matched. In practice, functional data often comes with unmatched boundaries. It happens, for example, in dynamical systems with variable evolution rates, such as COVID-19 infection rate curves associated with different geographical regions. Here, we develop a Riemannian framework that allows for partial matching, comparing, and clustering of functions with phase variability *and* uncertain boundaries. We extend past work by (1) Defining a new diffeomorphism group  $G$  over the positive reals that is the semidirect product of a time-warping group and a time-scaling group; (2) Introducing a metric that is invariant to the action of  $G$ ; (3) Imposing a Riemannian Lie group structure on  $G$  to allow for an efficient gradient-based optimization for elastic partial matching; and (4) Presenting a modification that, while losing the metric property, allows one to control the amount of boundary disparity in the registration. We illustrate this framework by registering and clustering shapes of COVID-19 rate curves, identifying basic patterns, minimizing mismatch errors, and reducing variability within clusters compared to previous methods.

**Index Terms**—Functional data analysis, elastic partial matching, phase variability, COVID-19 rates, elastic Riemannian metric.

## 1 INTRODUCTION

The field of functional data analysis (FDA) has seen a tremendous growth and activity over the last few years [10], [12], [16], [23], [27]. This phenomenal interest in FDA stems in part from our growing ability to record, store, and transmit data streams that are indexed over near-continuous times. A number of books and papers, including the ones mentioned above, have highlighted clear benefits of analyzing these data streams as continuous functions rather than as discrete time-series. In FDA, one treats observed functions as elements of a function space, endows a metric structure on this space, and uses the geometry of this metric space to perform statistical analyses. The main challenges in the analysis of functional data come from the infinite dimensionality of function spaces, the contamination of data due to noise, and the presence of phase variability within the data. We highlight the last, and arguably most important, of these issues next.

**Data with Phase Variability:** Functional data displays an interesting phenomenon that makes it unique from the perspective of statistical analysis. Real life functional data often comes with phase variability, *i.e.*, functions are often observed with perturbations or warplings of the time axis, resulting in the horizontal movements of peaks and valleys. In other words, noise in real data manifests itself as not only vertical (or additive) but also horizontal (or compositional), reflecting an inherent lack of temporal synchronization

across functions. For example, in the case of COVID-19 infection-rate curves, the high and low points for different regions occur at different times, implying different evolution rates of the virus cycle. Several papers [18], [19], [22], [29], [30] and a book [27] have documented and formally developed the concept of phase variability in functional data with the guidance that ignoring this phase variability leads to inflated variance in the data, loss of structures, and reduced power in hypothesis testing. Indeed these papers provide ways of separating phase and amplitude components (also termed *alignment or registration* of functions) and then performing either individualized or joint statistical analyses [15], [17], [22], [27], [31]. The proposed methods (for phase-amplitude separation) differ in their choices of metrics, tools for optimizations, and their definitions of phase; however, most of this work has a fundamental shortcoming. It assumes that the functions are fully observed over a common interval and moreover that the endpoints match perfectly across observations.

In more mathematical terms, let  $\{f_i : [0, T] \rightarrow \mathbb{R}, i = 1, 2, \dots, n\}$  be the set of observed functions on an interval  $[0, T]$ . Isolating phase variability implies finding time-warping functions  $\{\gamma_i\}$  such that the time-warped functions  $\{f_i \circ \gamma_i\}$  are *matched* (or *aligned*, or *registered*) where, in this context, a good matching generally refers to having functions with peaks and valleys co-located across observations. Here, the warping functions  $\{\gamma_i\}$  are constrained to be diffeomorphisms of  $[0, T]$  to itself such that  $\gamma_i(0) = 0$  and  $\gamma_i(T) = T$ . This last property implies that the endpoints of the data,  $\{f_i(0)\}$  and  $\{f_i(T)\}$ , are assumed to be already matched across all functions.

**Data with Sliding Right Boundary:** In many situations where the phase variability is present not just in the interior of  $[0, T]$  but also on the boundaries, the boundary-matching

- D. Bryner is with Naval Surface Warfare Center Panama City Division, Panama City, FL.  
E-mail: darshan.bryner@navy.mil
- A. Srivastava is with Department of Statistics, Florida State University, Tallahassee, FL.

Manuscript received February, 2021; revised XXX.

assumption breaks down and presents a major challenge. Sometimes one can assume that the left boundaries  $\{f_i(0)\}$  are either synchronized or can be synchronized perhaps through a simple shift. However, the right boundaries are still variable, i.e.,  $\gamma_i(T) \neq T$ . This situation arises, for example, in COVID-19 data where at the end of any observed interval, the infection rate curves for different regions are at different states of evolution. Another example of uncertain boundaries arises in censored data. Censoring is a process of randomly truncating the observation interval before the scheduled end is reached. In mathematical terms, such censored observations are denoted by  $\{f_i : [0, T_i] \rightarrow \mathbb{R}, i = 1, 2, \dots, n\}$ , with the end times  $T_i \in \mathbb{R}_+$  being arbitrary. There is a significant literature for analyzing right-censored functional data [5], [8], especially in survival analysis [7], [14], where the aim is to model censoring times as random variables and use the distribution of  $\min(T, T_i)$  explicitly in the likelihood functions. However, this literature does not account for phase variability and assumes that the functional data is fully registered. Thus, the analysis naturally deteriorates when phase variability is present in the actual data. A commonly used, albeit naive, solution is to simply time scale (linearly) each observation so that they all have the same domain  $[0, T]$ . However, this fails to address some deeper issues.

**Data with Both Phase Variability and Sliding Right Boundary:** In this paper, we are concerned with functional data with: (1) Random phase variability; and (2) Variable right endpoints  $T_i$ s. The challenge is to separate the phase from the amplitude when the right endpoints are no longer synchronized. In other words, we wish to align or register functions, essentially by matching their peaks and valleys inside the domain while simultaneously placing their floating right boundaries correctly. Some researchers refer to this problem as that of *elastic partial matching* [9], [21], [25]. An example of this situation arises in the analysis of Berkeley growth curves [24], where the natural start time of growth is birth. However, the end of the fixed 20-year observation period may not match across subjects due to different biological clocks. Another example mentioned earlier involves incidence rates of the COVID-19 virus for different geographical locations where different cities/states/countries exhibit different rates of evolution in the infections. The incidence's start time can be defined as the first positive case, but due to epidemiological and demographic factors, the pandemic evolution is undoubtedly not well synchronized across regions. Different regions experience different growths and decline rates, and one needs to perform registration to understand these patterns.

How can we analyze such data while preserving salient structures in the data? We illustrate this problem with a simple example. Figure 1(a) shows a pair of functions,  $f_1$  and  $f_2$ , where the left boundaries are matched well but the right boundaries are different, with the observed endpoints being  $T_1 = 2$  and  $T_2 = 1$ . Furthermore, the two functions have similar shapes except that  $f_1$  is missing a piece relative to  $f_2$ ; thus,  $f_1$  can only be well matched with a part of  $f_2$ , despite  $T_2 < T_1$ . If we linearly stretch  $f_2$  to match their boundaries by a simple time scaling ( $t \mapsto 2t$ ), we obtain the

result in (b). If, in addition to a linear scaling, we also time-warp  $f_2$  to match with  $f_1$ , we get (c). Finally, if we stretch  $f_2$  in such a way that  $f_1$  is matched to a *part* of  $f_2$  ( $t \mapsto 3t$ ) and then apply time warping, we get (d). Amongst these solutions, panel (d) provides the most satisfactory result.

In order to reach this solution, we need to infer two items: (1) which parts of the two functions match, i.e. how much time-scaling is needed, and (2) which time-warping aligns the two matched parts? If we have a set of functions, each with an uncertain right boundary, then the joint registration, modeling, and analysis of this data become even more challenging.

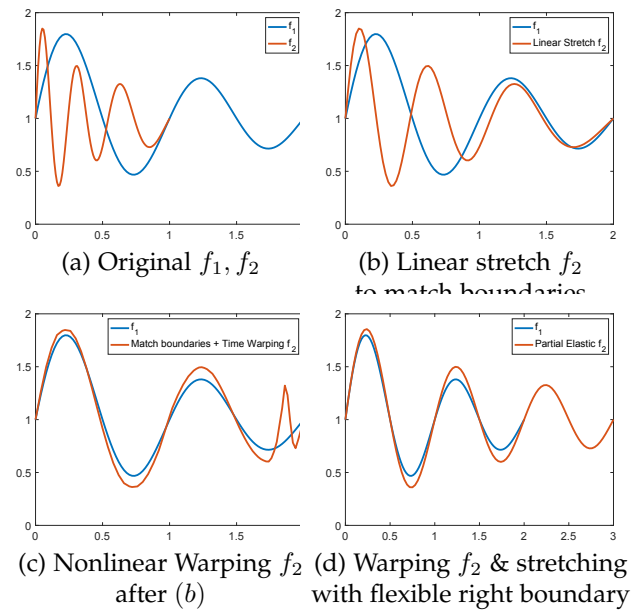


Fig. 1. Illustration of different potential solutions for matching curves with different shapes and flexible right boundaries.

The following items summarize the main challenges in developing mathematical solutions:

- 1) **Mathematical Representations & Metrics:** One needs mathematical representations that can account for uncertain boundaries and time-warplings of data. Specifically, we need metrics and objective functions such that partial matching of functions can be posed as binary optimization problems. Similar to past works on elastic shape analysis of functions [27], we need metrics that are invariant to actions of nuisance groups applicable here – time-warping and time-scaling.
- 2) **Interaction of Warping & Sliding Boundaries:** While the solutions for the individual problems – time-warping to match extrema and time-scaling to match the boundaries – are well known, the combination makes the problem more difficult. The combination requires diffeomorphic transformations of domains but without a fixed right boundary. This, in turn, demands searching over all combinations of linear stretches and nonlinear warplings to match any two functions. One needs to impose additional structure to help efficiently optimize over the joint space.

This formulation resembles the problem of partial matching of shapes, and there are several papers in the literature on this topic, see [1], [2], [6], [11], [20], [33] for example. However, our focus is on Riemannian approaches as they provide a comprehensive toolbox for the statistical analysis of shapes, including geodesics, proper metrics, statistical summaries, and probabilistic modeling. This paper develops a Riemannian framework for elastic analysis of functional data with right floating boundaries. Similar to the past works, we develop a framework where we represent both time-warping and time-scaling as group actions on a set of functions. A fully automated optimization procedure compares any two functions by optimally time-warping their domains and varying the right boundaries. The essential contribution here is a parametrization of the joint action of these groups, which facilitates the partial matching of curves. Additionally, we develop an extension that allows us to choose a scalar weight  $\lambda > 0$  for balancing the dual goals of matching the interiors and the right boundaries; however, some metric properties are lost when  $\lambda \neq 1$ .

The rest of this paper is organized as follows. We develop a mathematical framework for partially aligning functions using an elastic Riemannian metric and a square-root representation in Section 2. Section 3 develops the optimization of an objective function for pairwise matching of functions. Section 4 presents a set of experimental results involving both simulated and real data (COVID-19 infection rate curves) to demonstrate the success of the proposed framework. The paper ends with a summary and some conclusions in Section 5.

## 2 PROPOSED MATHEMATICAL FRAMEWORK

In this section, we develop an elastic Riemannian framework for representing, partially matching, and comparing functional data with phase variability and sliding right boundaries. Before we develop our approach, we summarize the past ideas for phase-amplitude separation, or registration of functions, with matched boundaries. We will follow the approach presented in [27], [28], [29].

### 2.1 Past Work in the Alignment of Functions with Fixed Boundaries

Let  $\mathcal{F} = \{f : [0, T] \rightarrow \mathbb{R} \mid f \text{ is absolutely continuous}\}$  be the set of functions of interest. Let  $\Gamma_T$  be the group of all positive diffeomorphisms from  $[0, T]$  to itself that preserve the boundaries. For any  $f \in \mathcal{F}$  and  $\gamma \in \Gamma_T$ , the composition  $f \circ \gamma$  denotes a time warping of  $f$  while keeping the boundaries fixed ( $\gamma(0) = 0, \gamma(T) = T$ ). In order to register functions, one represents them by their square-root velocity functions (SRVFs). For any  $f \in \mathcal{F}$ , its SRVF is given by  $q(t) = \text{sign}(\dot{f}(t))\sqrt{|\dot{f}(t)|}$ . In fact, this mapping  $f \mapsto q$  forms a bijection between  $\mathcal{F}$  and  $\mathbb{L}^2([0, T], \mathbb{R})$ , up to a constant. One can reconstruct a function  $f$  from its SRVF  $q$  and  $f(0)$  using  $f(t) = f(0) + \int_0^t q(s)|q(s)| ds, t \in [0, T]$ .

For any  $\gamma \in \Gamma_T$ , the SRVF of the time-warped function  $f \circ \gamma$  is given by  $(q * \gamma) \triangleq (q \circ \gamma)\sqrt{\dot{\gamma}}$ . A very important property of SRVFs is that for any two functions  $f_1, f_2 \in \mathcal{F}$ , and their SRVFs  $q_1, q_2$ , we have  $\|q_1 - q_2\| = \|(q_1 * \gamma) - (q_2 * \gamma)\|$  for all  $\gamma \in \Gamma_T$ . (The norm here is the standard

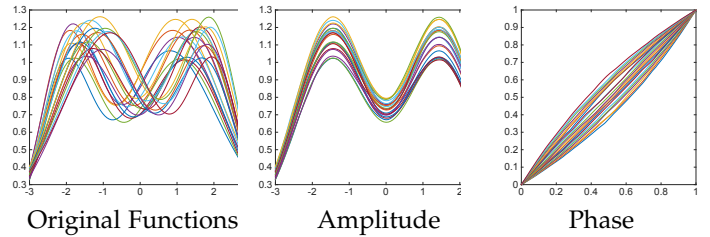


Fig. 2. Phase-amplitude separation of uncensored functional data using SRVF representation.

$\mathbb{L}^2$  norm:  $\|q\| = \sqrt{\int_0^T q(t)^2 dt}$ .) Due to this property, the  $\mathbb{L}^2$  norm between SRVFs is called the *elastic metric*. With the elastic metric, the problem of registration between functions can be solved as the following optimization:

$$\hat{\gamma} = \underset{\gamma \in \Gamma_T}{\operatorname{argmin}} \|q_1 - (q_2 * \gamma)\|. \quad (1)$$

This optimization is solved either using the dynamic programming algorithm or a gradient-based approach, as needed, and the resulting registration is called *dense elastic registration*. For any  $t \in [0, T]$ , the point  $f_1(t)$  is said to be matched to the point  $f_2(\hat{\gamma}(t))$ .

For alignment of multiple functions, one exploits the fact that the minimum in Eqn. 1 is actually a proper distance on the quotient space  $\mathbb{L}^2/\Gamma_T$ . Using this distance, one can define a mean function under this metric and then align the given functions to this mean using Eqn. 1. For a given set of functions  $\{f_1, f_2, \dots, f_n\}$ , this framework results in a set of time-warping functions (also called phases)  $\{\gamma_i\}$  and the corresponding aligned functions  $\{f_i \circ \gamma_i\}$  (also called amplitudes). The full procedure for this Phase-Amplitude separation has been outlined in Chapter 8 of the textbook [27]. Figure 2 shows an example of this approach – the left panel shows the original functions  $\{f_i\}$ , the middle panel shows the time-warped aligned functions  $\{f_i \circ \gamma_i\}$ , and the right panels shows the optimal warping functions  $\{\gamma_i\}$ .

This approach has been remarkably successful in the alignment of functional data in a variety of applications. However, as mentioned earlier, this framework assumes that both the endpoints  $t = 0$  and  $t = T$  are preregistered in all observations. In Fig. 2, the endpoints are kept fixed while the interior is time-warped. Next, we consider the problem of aligning functional data with a sliding right boundary. Note that one of the boundaries, say the left one, can be matched by translation, and that leaves only the other boundary to be matched.

### 2.2 Joint Time-Warping and Time Scaling

Re-define the set of functions to be  $\mathcal{F} = \{f : [0, \infty) \rightarrow \mathbb{R} \mid f \text{ is absolutely continuous}\}$ . Define the space  $\mathcal{F}_0 \subset \mathcal{F}$  as the space of all absolutely continuous, right-censored functions on  $[0, \infty)$  as

$$\mathcal{F}_0 = \{f \in \mathcal{F} \mid \exists c \geq 0 \text{ s.t. } f(t) = \text{const. for all } t > c\}.$$

Let  $\mathbb{L}_0^2 \subset \mathbb{L}^2([0, \infty), \mathbb{R})$  be the set of SRVFs of the elements of  $\mathcal{F}_0$ . That is, any  $q \in \mathbb{L}_0^2$  is a square-integrable function such that  $q(t) = 0$  for all  $t > c$  and some  $c > 0$ . Define

the preshape distance between censored functions as the  $\mathbb{L}^2$  distance on their corresponding SRVFs.

**Definition 1.** The preshape distance on  $\mathcal{F}_0$  is given by the  $\mathbb{L}^2$  distance

$$d_p(f_1, f_2)^2 = \int_0^\infty (q_1(t) - q_2(t))^2 dt,$$

where  $q_1, q_2$  are SRVFs of  $f_1, f_2$ , respectively.

Next, we re-define the diffeomorphism group  $\Gamma$  to be on the domain  $[0, \infty)$ :

$$\Gamma = \{\gamma : [0, \infty) \rightarrow [0, \infty) \mid \gamma(0) = 0, \gamma \text{ is a positive diffeomorphism}\}.$$

One can show that  $\Gamma$  has properties similar to  $\Gamma_T$  in the previous section. Namely,  $\Gamma$  is a group under composition as the binary operation, and  $\Gamma$  forms a right group action on the function space  $\mathcal{F}_0$  also by function composition. In particular, for a function  $f \in \mathcal{F}_0$  with censoring point  $c$ , the group action is  $f \mapsto f \circ \gamma$ , which sends the censoring point  $c$  to  $\gamma^{-1}(c)$ . Also similar to that of  $\Gamma_T$ , the corresponding right group action of  $\Gamma$  on  $\mathbb{L}_0^2$  is given by  $q * \gamma = (q \circ \gamma)\sqrt{\gamma'}$ . Furthermore, the group  $\Gamma$  also acts by isometries with respect to the preshape distance  $d_p$ . That is, for any  $\gamma \in \Gamma$  and  $f_1, f_2 \in \mathcal{F}_0$ , the distance preserving property  $d_p(f_1, f_2) = d_p(f_1 \circ \gamma, f_2 \circ \gamma)$  holds.

At this stage, one can use the past formulation to register functions according to  $\inf_{\gamma \in \Gamma} d_p(f_1, f_2 \circ \gamma)$ . However, since the functions  $q_1$  and  $q_2$  are zero on the right sides, i.e., they contain no information to help guide function registration, the use of the full  $\Gamma$  is excessive. Instead, we will seek a proper subgroup of  $\Gamma$  that makes the optimization computationally efficient and the result more interpretable. We construct this new group  $G \subset \Gamma$  from a combination of the time-scaling  $H \subset \Gamma$  and time-warping  $N \subset \Gamma$ , which have the following definitions.

**Definition 2.** The **time-scaling** set  $H$  is given by

$$H = \{h \in \Gamma \mid h(t) = at \text{ for any } a > 0\},$$

the set of all positive, linear functions on  $[0, \infty)$ .

**Definition 3.** The **time-warping** set  $N$  is given by

$$N = \cup_{b>0} \{n \in \Gamma \mid n(t) = t \text{ for } t \geq b\},$$

the set of fixed-interval diffeomorphisms on  $[0, b]$  completed with identity from  $b$  to infinity, unioned over all  $b > 0$ . The point  $b$  is termed the *pivot point* of the diffeomorphism  $n$ .

Now, any group element  $g \in G$  is constructed such that  $g = n \circ h$ , and since  $G \subset \Gamma$ , the group actions of  $G$  on  $\mathcal{F}_0$  and  $\mathbb{L}_0^2$  remain the same as that of  $\Gamma$ . That is, for any  $g = n \circ h \in G$  and  $f \in \mathcal{F}$ , the action of  $g$  on  $f$  is given by  $f \circ n \circ h$ . Also, the action of  $G$  on  $\mathcal{F}$  is by isometries under the metric  $d_p$  simply because the larger group  $\Gamma$  does the same.

Later on, we establish that  $G$  is a proper subgroup of  $\Gamma$ , thus forming the quotient space  $\mathbb{L}_0^2/G$  and defining a shape distance on the quotient space. However, first, we provide intuition and motivation for this specific construction. This formulation allows us to restrict the search from a prohibitively large space  $\Gamma$  to only the pair  $a \in \mathbb{R}_+$  and  $\gamma \in \Gamma_b$ ,

where  $b$  is the pivot mentioned above. Further, if the pivot point is known, we can simply perform the time-warping on the fixed space  $\Gamma_1$  instead of  $\Gamma_b$ . We can thus leverage existing computational tools to optimize over this joint space  $(a, \gamma) \in \mathbb{R}_+ \times \Gamma_1$ . The resulting optimal registration is also highly interpretable – it is a global stretching/compression of the function combined with a nonlinear warping.

However, the question remains: What should the value of the pivot point  $b$  be? Suppose we are registering the function  $f_2$  to  $f_1$  in  $\mathcal{F}_0$ , with the respective censoring points being  $c_2$  and  $c_1$ . That is, we solve for  $\inf_{g \in G} d_p(f_1, f_2 \circ g)$  with  $g = n \circ h$  and  $h = at$ . Since the SRVFs  $q_1$  and  $q_2$  are zero after their respective censoring points, there is no need to perform any time-warping beyond the minimum of the two censoring points. Moreover, since after time-scaling, the censoring point  $c_2$  becomes  $h^{-1}(c_2) = c_2/a$ , we can set the pivot point  $b$  to be  $b = \min(c_1, c_2/a)$ . Thus, the pivot  $b$  only depends on the free parameter  $a$ , simplifying the subsequent optimization.

Now, we can write the functional form of  $g \in G$  in a parameterized form as follows. First, note that for  $h \in H$  with slope  $a$  and for  $n \in N$  with pivot point  $b$ , the operation  $h \circ n \circ h^{-1}$  sends  $n$  to another element  $\tilde{n} \in N$  with pivot point  $b/a$ . Furthermore, this operation does not change the shape of  $n$  and simply scales the time-warping function along the diagonal. We can thus write any  $n \in N$  as  $n = h \circ n_0 \circ h^{-1}$ , where  $n_0 \in N$  has a pivot of  $b = 1$ . That is, if we let  $h(t) = bt$  with the desired pivot point  $b$ , then  $n(t)$  can be written as  $n(t) = bn_0(t/b)$ . By letting  $n_0(t) = \gamma(t)$  on  $0 \leq t \leq 1$  for  $\gamma \in \Gamma_1$ , we can identify any  $g \in G$  with the triplet  $(a, b, \gamma)$  with

$$g(t) = \begin{cases} ab\gamma(t/b) & t \leq b \\ at & t > b. \end{cases} \quad (2)$$

However, as explained in the previous paragraph, since we know the censoring points  $c_1$  and  $c_2$ , we simplify this parameter space by setting the pivot value to  $b = \min(c_1, c_2/a)$ . Now, any  $g \in G$  can be identified with the pair  $(a, \gamma) \in \mathbb{R}_+ \times \Gamma_1$ .

## 2.3 $G$ is a Proper Subgroup of $\Gamma$

In order to show that  $G$  is a proper subgroup of  $\Gamma$ , we first show that the time-scaling set  $H$  and the time-warping set  $N$  are both subgroups of  $\Gamma$ , with  $N$  being a *normal* subgroup. Then we show that  $G$  can be constructed as an outer semidirect product of  $H$  and  $N$ ; hence, it is a proper subgroup of  $\Gamma$ .

**Lemma 1.** The time-scaling set  $H$  is a subgroup of  $\Gamma$ .

**Proof:**  $H \subset \Gamma$  by definition. The closure property is achieved due to linearity:  $h_1 \circ h_2 = h_1(a_2t) = a_2h_1(t) = a_2a_1t \in H$ . Any  $h \in H$  has an inverse  $h^{-1}(t) = t/a$ , and thus  $h^{-1} \in H$ . The identity element  $e(t) = t$  is also in  $H$ .  $\square$

**Lemma 2.** The time-warping set  $N$  is a normal subgroup of  $\Gamma$ .

**Proof:**  $N \subset \Gamma$  by definition. To establish the closure property, note that for any  $n_1, n_2 \in N$ , the composition  $\tilde{n} = n_1 \circ n_2$  is also a member of  $N$ , with  $\tilde{b} = \max(b_1, b_2)$  such that  $\tilde{n}(t) = t$  for  $t \geq \tilde{b}$ .



To prove that  $N$  is a *normal* subgroup of  $\Gamma$ , we need to show that  $\gamma \circ n \circ \gamma^{-1} \in N$  for all  $\gamma \in \Gamma$  and  $n \in N$ . Evaluating the composition, we get

$$\begin{aligned} \gamma(n(\gamma^{-1}(t))) &= \begin{cases} \gamma(n(\gamma^{-1}(t))) & 0 \leq t \leq \gamma(a) \\ \gamma(\gamma^{-1}(t)) & t > \gamma(a) \end{cases} \\ &= \begin{cases} (\gamma \circ n \circ \gamma^{-1})(t) & 0 \leq t \leq \gamma(a) \\ t & t > \gamma(a), \end{cases} \end{aligned}$$

which is a member of  $N$ .  $\square$

We need to form a product of  $N$  and  $H$  to reach the desired subgroup  $G$ . However, since  $N$  and  $H$  do not commute, we cannot simply construct a direct product group  $N \times H$ . Instead, using the fact that  $N$  and  $H$  are both groups, and that  $N \cap H = \{e\}$ , the identity element, we can construct a new subgroup  $G \subset \Gamma$  as an *outer semidirect product* of both  $N$  and  $H$ . As a first step, we define the following homomorphism.

**Definition 4.** Let  $\phi : H \rightarrow \text{Aut}(N)$  be the group homomorphism defined by  $\phi(h)(n) = h \circ n \circ h^{-1}$  for all  $h \in H$  and  $n \in N$ . We write this homomorphism as  $\phi_h : N \rightarrow N$  for brevity.

Before we proceed, we establish that this mapping is indeed a homomorphism.

**Lemma 3.** The map  $h \mapsto \phi_h$  is a homomorphism  $\phi : H \rightarrow \text{Aut}(N)$ , i.e.  $\phi_{h_1} \circ \phi_{h_2} = \phi_{h_1 \circ h_2}$ .

**Proof:** For simplicity, we suppress the symbol  $\circ$  for function composition in this proof. Since  $N$  is a normal subgroup of  $\Gamma$ , and  $H \subset \Gamma$ , it holds that  $\phi_{h_1}\phi_{h_2}$  and  $\phi_{h_1 h_2}$  are both functions from  $N \rightarrow N$ . For any  $n \in N$ , we have that  $\phi_{h_1}\phi_{h_2}(n) = \phi_{h_1}(h_2 n h_2^{-1}) = h_1 h_2 n h_2^{-1} h_1^{-1} = (h_1 h_2) n (h_1 h_2)^{-1} = \phi_{h_1 h_2}(n)$ .  $\square$

Now we have all the pieces to form the subgroup required for registration of functions.

**Definition 5.** Using the time-warping subgroup  $N$ , the time-scaling subgroup  $H$ , and the homomorphism  $\phi : H \rightarrow \text{Aut}(N)$ , define the outer semidirect product  $G = N \rtimes_{\phi} H$ , denoted by  $G \subset \Gamma$ , to be the pairing  $(N, H)$  with the following properties:

- 1) Group operation:  $(n_1, h_1) \cdot (n_2, h_2) = (n_1 \circ \phi_{h_1}(n_2), h_1 \circ h_2)$
- 2) Inverse element:  $(n, h)^{-1} = (\phi_{h^{-1}}(n^{-1}), h^{-1})$ .

Using Lemmas 1, 2, and 3 and the definition of the homomorphism  $\phi$ , by definition of an outer semidirect product, we can show that  $G$  is a proper subgroup of  $\Gamma$ .

**Remarks:** By definition, any element in  $G$  can be expressed uniquely as  $g = n \circ h$ . (Therefore,  $G$  also contains functions  $g \in \Gamma$  that take the form  $g(t) = at$  for all  $t > b$ .) Henceforth, we will use this shorthand notation of  $g$  to indicate the time-warping and time-scaling pair  $(n, h)$ .

Since the subgroup  $G$  acts on  $\mathcal{F}_0$  (correspondingly on  $\mathbb{L}_0^2$ ) by isometries, the distance  $d_p$  descends to the quotient space  $\mathcal{F}_0/G$  (correspondingly  $\mathbb{L}_0^2/G$ ) and provides a distance on this quotient space.

**Definition 6.** The shape distance on  $\mathcal{F}_0$  is given by

$$d_s(f_1, f_2) = \inf_{g \in G} d_p(f_1, f_2 \circ g) \quad (3)$$

Similar to the theory developed in [27], [28], [29], this distance  $d_s$  is a proper distance on the quotient space  $\mathcal{F}/G$ .

We illustrate this idea using a simple example in Fig. 3. Here, we first compute the shape distance between a pair of censored functions  $f_1$  and  $f_2$  (with respective censoring points  $c_1$  and  $c_2$ ) by aligning  $f_2$  to  $f_1$  according to Eq. 3. In the first row, the left plot shows the original function pair, the center plot shows the optimally aligned  $f_2$  with respect to  $f_1$ , and the right plot shows the optimal  $g \in G$  needed to perform the time-warping of  $f_2$ . The second row shows the same progression of events but for when the index labeling of the two functions has been reversed. Here, since the shape distance is a proper distance and is hence symmetric, the shape distance is the same in both cases, and the optimal diffeomorphisms in  $G$  are inverses of each other.

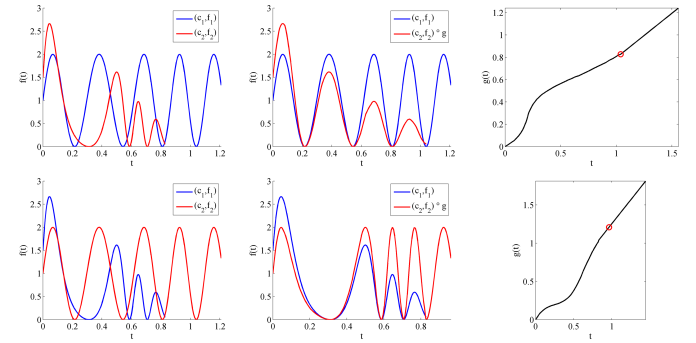


Fig. 3. Alignment of two functions using elastic partial matching. Each row shows an example with the same two functions except that the labels  $f_1$  and  $f_2$  have been reversed in the second row. The first column shows the original functions, the second column shows the aligned functions, and the third column shows the optimal diffeomorphism  $\hat{g} \in G$  used in the alignment. The red circle on each diffeomorphism represents its pivot point, which in this case is the minimum censoring point  $b = \min\{c_1, c_2\}$ .

### 3 OPTIMIZATION DETAILS FOR PAIRWISE SHAPE DISTANCE

Next, we develop techniques for optimally aligning censored functions  $f_1$  and  $f_2$  (with known censoring points  $c_1$  and  $c_2$ ) in order to compute their shape distance as defined in Eq. 3. That is, we develop numerical recipes to find and apply a group element  $\hat{g} \in G$  to  $f_2$  in order to optimally match a fixed  $f_1$  to minimize  $d_p$ . Towards this end, we define an energy:

$$E(g) = \int_0^\infty (q_1(t) - (q_2 * g)(t))^2 dt.$$

We have shown in Section 2.2 that any  $g \in G$  can be identified by the pair  $(a, \gamma) \in \mathbb{R}_+ \times \Gamma_1$  with  $b = \min(c_1, c_2/a)$ . Thus,

$$\begin{aligned} E(a, \gamma) &= \int_0^b (q_1(t) - q_2(ab\gamma(t/b))\sqrt{a\dot{\gamma}(t/b)})^2 dt \\ &\quad + \int_b^\infty (q_1(t) - q_2(at)\sqrt{a})^2 dt. \end{aligned} \quad (4)$$

Then, we solve for  $(\hat{a}, \hat{\gamma}) = \underset{(a, \gamma) \in \mathbb{R}_+ \times \Gamma_1}{\operatorname{argmin}} E(a, \gamma)$ , form  $\hat{g}(t)$  using Eqn. 2, set  $\hat{b} = \min(c_1, c_2/\hat{a})$ , and apply  $\hat{f}_2 = f_2 \circ \hat{g}$ .

### 3.1 Grid-Search Algorithm

As a simpler first idea, we present a grid-search approach for this optimization in the following manner. We start by defining a finite sampling, or “grid,” on the time-scaling parameter  $a \in \mathbb{R}_+$  given by  $\{a_i, i = 1, \dots, J\}$ . Then, for each grid point  $i = 1, \dots, J$ , we fix the time-scaling parameter  $a_i$  and pivot point  $b_i = \min(c_1, c_2/a_i)$  and solve for the optimal time-warping  $\gamma_i \in \Gamma_1$ . One can solve for  $\gamma_i$  by first truncating the two functions to the common overlapping domain  $[0, b_i]$ , rescaling both functions to have domain  $[0, 1]$ , and then performing dense elastic registration over this interval as described in Subsection 2.1. One can perform this function registration using one of several approaches present in the literature. The most commonly used tool is the dynamic programming algorithm [27], but one can also exploit the geometry of the space  $\Gamma_1$  to develop a BFGS-based gradient search [13]. The pair  $(a_i, \gamma_i)$  that yields the lowest value of the energy function  $E$  in Eqn. 4 defines the optimal diffeomorphism  $\hat{g} \in G$  that best matches  $f_2$  to  $f_1$ .

**Given:** Censored function pair  $f_1, f_2 \in \mathcal{F}_0$  with censoring times  $c_1, c_2 \geq 0$ , and time-scaling parameter grid  $\{a_i, i = 1, \dots, J\} \in \mathbb{R}_+$ .

**Result:** Optimally aligned censored functions with associated group element and shape distance.

```

1 Compute SRVFs  $q_1, q_2 \in \mathbb{L}_0^2$ ;
2 for  $i = 1$  to  $J$  do
3   Time-scale  $q_2$  via  $\tilde{q}_2 = \sqrt{a_i} q_2(a_i t)$ ;
4   Compute pivot point  $b_i = \min\{c_1, c_2/a_i\}$ ;
5   Truncate both functions via  $\tilde{q}_1 = q_1 1_{[0, b_i]}$  and
    $\tilde{q}_2 = \tilde{q}_2 1_{[0, b_i]}$ ;
6   Time-scale both  $\tilde{q}_1, \tilde{q}_2$  to have domain  $[0, 1]$  via
    $\tilde{q}_j = \sqrt{1/b_i} \tilde{q}_j(t/b_i)$  for  $j = 1, 2$ ;
7   Solve  $\gamma_i = \operatorname{argmin}_{\gamma \in \Gamma_1} \|\tilde{q}_1 - (\tilde{q}_2 \circ \gamma) \sqrt{\gamma}\|^2$  via
   Dynamic Programming, gradient descent, or
   BFGS;
8   Compute  $E_i = E(a_i, \gamma_i)$  via Eqn. 4;
9 end
10 Let  $\hat{i} = \operatorname{argmin}_i \{E_i, i = 1, \dots, J\}$ ;
11 Form  $g_i$  via Eqn. 2 using  $a_i, b_i, \gamma_i$ ;
12 return  $\hat{g} = g_{\hat{i}}, \hat{f}_2 = f_2 \circ g_{\hat{i}}, \hat{c}_2 = c_2/a_{\hat{i}},$  and
    $d_s = \sqrt{E_{\hat{i}}}$ 

```

**Algorithm 1:** Shape Distance Via Grid Search Over Time-Scaling Parameter

Algorithm 1 describes the grid-search algorithm for elastic partial matching of right-censored functions. The advantages of this algorithm are that: (1) It is relatively straightforward and uses mostly existing tools from the literature, and (2) It provides global solutions depending on the grid resolution. The disadvantage is that it is computationally expensive to solve for an optimal diffeomorphism  $\gamma_i \in \Gamma_1$  for each candidate time-scaling parameter  $a_i$ . A more computationally efficient solution is to perform a gradient search over the joint domains. However, due to the high-dimensionality of the state space and complex nature of the objective function, any gradient-based optimization

will be highly dependent on initialization. Therefore, in practice we use Algorithm 1, with a limited number of grid points, for obtaining a coarse initialization for a gradient-descent algorithm. The gradient descent is then used as a local refinement of the coarse initialization.

### 3.2 Gradient-Based Joint Optimization

In order to derive a gradient-based optimization of  $E$ , we change to a more convenient mathematical representation for elements of the group  $G$ . For an element  $g \in G$  identified by parameters  $(a, \gamma) \in \mathbb{R}_+ \times \Gamma_1$ , consider the map  $M$  given by  $M(a, \gamma) = (\log(a), \sqrt{\gamma})$ , the log-transformation of  $a$  and the SRVF of  $\gamma$ . Let  $\xi \in \mathbb{R}$  be such that  $a = e^\xi$  and let the space of SRVFs of  $\Gamma_1$  be  $\Psi$ . It is easy to see that for any  $\gamma \in \Gamma_1$ , its SRVF  $\psi = \sqrt{\gamma}$  is non-negative and has unit  $\mathbb{L}^2$  norm (please refer to [27]).

**Definition 7.** Define the space of SRVFs of  $\Gamma_1$  as the positive orthant of the unit Hilbert sphere

$$\Psi = \{\psi \in \mathbb{L}^2([0, 1], \mathbb{R}) \mid \|\psi\| = 1, \psi > 0 \text{ a.e.}\}.$$

Define the parameter space  $\mathcal{P} = \mathbb{R} \times \Psi$  as the set of all the transformed variables  $(\xi, \psi)$ . Now, in the context of censored function registration, we can form the group element  $g \in G$  from the parameters  $(\xi, \psi)$ . Since  $(a, \gamma) = M^{-1}(\xi, \psi) = (e^\xi, \int_0^t \psi(s)^2 ds)$  and  $b = \min(c_1, c_2 e^{-\xi})$ , we can rewrite Eqn. 2 as

$$g(t) = \begin{cases} e^\xi b \int_0^{t/b} \psi(s)^2 ds & t \leq b \\ e^\xi t & t > b. \end{cases} \quad (5)$$

Moreover, the group action of  $G$  on  $q_2 \in \mathbb{L}_0^2$  can be written in terms of the parameters  $(\xi, \psi) \in \mathcal{P}$  as such:

$$\begin{aligned} q * g &= (q \circ g) \sqrt{\dot{g}} \\ &= \begin{cases} q(e^\xi b (\int_0^{t/b} \psi(s)^2 ds)) e^{\xi/2} \psi(t/b) & t \leq b \\ q(e^\xi t) e^{\xi/2} & t > b. \end{cases} \end{aligned} \quad (6)$$

Thus, the energy function in Eqn. 4 can be re-written as

$$\begin{aligned} E(\xi, \psi) &= \int_0^b (q_1(t) - q_2(e^\xi b (\int_0^{t/b} \psi(s)^2 ds)) e^{\xi/2} \psi(t/b))^2 dt \\ &\quad + \int_b^\infty (q_1(t) - q_2(e^\xi t) e^{\xi/2})^2 dt, \end{aligned}$$

where  $b = \min(c_1, c_2 e^{-\xi})$ .

What is the reason for changing the representation of elements of  $G$  from  $\mathbb{R}_+ \times \Gamma_1$  to  $\mathcal{P}$ ? The reason is that the Riemannian geometry of  $\mathcal{P}$  is less complex, relatively, in that we know the expressions for tangent spaces, exponential maps and geodesics in  $\mathcal{P}$ . Similar to  $G$ ,  $\mathcal{P}$  is also a Lie group. Note that the group operation is  $(\xi_1, \psi_1) \cdot (\xi_2, \psi_2) = (\xi_1 + \xi_2, (\psi_1 \circ \int_0^t \psi_2(s)^2 ds) \psi_2)$ , the inverse element is given by  $(-\xi, 1/\psi) \in \mathcal{P}$ , and the identity element of  $\mathcal{P}$  is given by  $p_{id} \equiv (\xi_{id}, \psi_{id}) \in \mathcal{P}$ , where  $\xi_{id} = 0$  and  $\psi_{id}(t) = 1$  for all  $t \in [0, 1]$ . This Lie group structure allows us to formulate gradient vectors at each iteration in the tangent space of the identity element and apply these updates to the current estimate in a sequential manner.

**Definition 8.** Define the tangent space of  $\mathcal{P}$  at the identity element  $p_{id} = (\xi_{id}, \psi_{id}) = (0, 1)$  as

$$T_{p_{id}}(\mathcal{P}) = \left\{ v = (y, z) \in \mathbb{R} \times \mathbb{L}^2([0, 1], \mathbb{R}) \mid \int_0^1 z(t) dt = 0 \right\}.$$

**Definition 9.** Define the exponential mapping of  $v = (y, z) \in T_{p_{id}}(\mathcal{P})$  as  $\exp_{p_{id}}(v) \in \mathcal{P}$  via the formula

$$\exp_{p_{id}}(v) = (y, \cos(\|z\|) + \sin(\|z\|) \frac{z}{\|z\|}).$$

In an iterative line-search, at iteration  $k$ , let the current estimate be  $p_k = (\xi_k, \psi_k) \in \mathcal{P}$ , a search direction vector  $v_k$  be in the tangent bundle of  $\mathcal{P}$ , and a step size be  $\alpha_k > 0$ . If  $\mathcal{P}$  were a vector space, the updated estimate would be given simply by  $p_{k+1} = p_k + \alpha_k v_k$ . Since  $\mathcal{P}$  is instead a non-linear manifold, one can compute the update using the exponential mapping  $p_{k+1} = \exp_{p_k}(\alpha_k v_k)$ . However, since  $\mathcal{P}$  is also a Lie group with an identity element and a group action, there is no need to compute search directions in any tangent space except at identity. For any  $k$ , we can compute  $v_k \in T_{p_{id}}(\mathcal{P})$  and then apply a sequential update to  $p_k$  via  $p_{k+1} = p_k \cdot \exp_{p_{id}}(\alpha_k v_k)$ , where  $\cdot$  denotes the group operation of  $\mathcal{P}$ .

Most often, the search direction  $v_k$  is based on the gradient of the objective function, or energy function, evaluated at the current estimate. Therefore, we need to define the energy  $E_k : \mathcal{P} \rightarrow \mathbb{R}$  at step  $k$  and show how to compute its gradient as an element of  $T_{p_{id}}(\mathcal{P})$ . First, define the current censored function as  $q_{2,k} \triangleq (q_2 * g_k)$ , where  $g_k$  is identified by  $(\xi_k, \psi_k) \in \mathcal{P}$  and  $b_k = \min(c_1, c_2 e^{-\xi_k})$ .

**Definition 10.** For an update  $(\xi, \psi) \in \mathcal{P}$  at iteration  $k$ , define the energy of updating the current censored function  $q_{2,k}$  with censoring point  $c_{2,k}$  by  $(\xi, \psi)$  as  $E_k(\xi, \psi)$

$$= \int_0^b (q_1(t) - q_{2,k}(e^{\xi} b (\int_0^{t/b} \psi(s)^2 ds))) e^{\xi/2} \psi(t/b))^2 dt + \int_b^\infty (q_1(t) - q_{2,k}(e^{\xi} t) e^{\xi/2})^2 dt, \quad (7)$$

where  $b = \min(c_1, c_{2,k} e^{-\xi})$ .

### Derivation of Gradient of $E$ :

Next, we compute the gradient  $\nabla E_k$  at identity and define a line-search update direction vector  $v_k \in T_{p_{id}}(\mathcal{P})$  based on this gradient. Before writing the full analytical expression for the gradient, we develop a series of useful results.

**Lemma 4.** For  $q_1, q_2 \in \mathbb{L}_0^2$  with censoring points  $c_1, c_2 \geq 0$  and  $g \in G$  identified by  $(\xi, \psi) \in \mathcal{P}$  with pivot point  $b = \min(c_1, c_2 e^{-\xi})$ , let  $\tilde{q}_2 \triangleq q_2 * g$  according to Eqn. 6. The derivative of  $\tilde{q}_2$  with respect to  $\xi$  evaluated at the identity element  $(\xi_{id}, \psi_{id}) \in \mathcal{P}$  is given by  $\frac{\partial \tilde{q}_2}{\partial \xi}(\xi_{id}, \psi_{id}) = t\dot{q}_2 + \frac{1}{2}q_2$  for all  $t \geq 0$ .

**Proof:** See Appendix A.

**Lemma 5.** Let  $q_1, q_2 \in \mathbb{L}_0^2$  with censoring points  $c_1, c_2 \geq 0$ , let  $b = \min(c_1, c_2)$  and let  $F(\psi) = \int_0^b (q_1(t) - (q_2(b \int_0^{t/b} \psi(s)^2 ds) \psi(t/b)))^2 dt$ . Then, if  $x = t/b$ , the Riemannian gradient at identity  $\nabla F \in T_{\psi_{id}}(\Psi)$  is given by  $\nabla F = w(x) - \int_0^1 w(x) dx$ , with  $w(x) = 4b^2 \int_0^x (q_1(bs) - q_2(bs)) \dot{q}_2(bs) ds - 2b(q_1(bx) - q_2(bx))q_2(bx)$ .

**Proof:** See Appendix B.

Having obtained these useful results, we now derive the expression for gradient of  $E_k$  with respect to an incremental element of  $\mathcal{P}$ .

**Theorem 1.** At iteration  $k$  and for  $q_1, q_{2,k} \in \mathbb{L}_0^2$  with censoring points  $c_1, c_{2,k} \geq 0$ , the gradient of  $E_k$  at identity  $(\xi_{id}, \psi_{id}) \in \mathcal{P}$  is written as the pair  $\nabla E_k(\xi_{id}, \psi_{id}) = (\frac{\partial E_k}{\partial \xi}(\xi_{id}, \psi_{id}), \frac{\partial E_k}{\partial \psi}(\xi_{id}, \psi_{id})) \in T_{p_{id}}(\mathcal{P})$ . The two terms that comprise the gradient vector are defined as follows.

- 1) The partial derivative of  $E_k$  with respect to  $\xi \in \mathbb{R}$  evaluated at identity is given by  $\frac{\partial E_k}{\partial \xi}(\xi_{id}, \psi_{id})$

$$= -2 \int_0^\infty (q_1 - q_{2,k})(t\dot{q}_{2,k} + \frac{1}{2}q_{2,k}) dt \in \mathbb{R}. \quad (8)$$

- 2) Let  $b = \min(c_1, c_{2,k})$ ,  $x = t/b$ , and define the function  $w_k \in T_{\psi_{id}}(\Psi)$  as

$$w_k(x) = 4b^2 \int_0^x (q_1(bs) - q_{2,k}(bs)) \dot{q}_{2,k}(bs) ds - 2b(q_1(bx) - q_{2,k}(bx))q_{2,k}(bx).$$

Then, the partial derivative of  $E_k$  with respect to  $\psi \in \Psi$  evaluated at identity is given by

$$\frac{\partial E_k}{\partial \psi}(\xi_{id}, \psi_{id}) = w_k(x) - \int_0^1 w_k(x) dx. \quad (9)$$

**Proof:** For part (1), define the function  $\tilde{q}_{2,k} = q_{2,k} * g$ , where  $g \in G$  is identified by  $(\xi, \psi) \in \mathcal{P}$  and  $b = \min(c_1, c_{2,k} e^{-\xi})$ . Now, we compute the partial derivative as

$$\frac{\partial E_k}{\partial \xi} = -2 \int_0^\infty (q_1 - \tilde{q}_{2,k}) \frac{\partial \tilde{q}_{2,k}}{\partial \xi} dt.$$

Using Lemma 4, the above expression evaluated at identity becomes the expression given in Eqn. 8.

For part (2), define  $F_k(\psi) = E_k(\xi_{id}, \psi) - \int_b^\infty (q_1(t) - q_{2,k}(t))^2 dt$ . Notice that  $F_k(\psi)$  takes the form of  $F$  as defined in Lemma 5 with  $q_2$  replaced with  $q_{2,k}$ , and by construction  $\nabla F_k(\psi_{id}) = \frac{\partial E_k}{\partial \psi}(\xi_{id}, \psi_{id})$ . Thus, according to Lemma 5, the partial derivative is equal to the expression given in Eqn. 9.  $\square$

In order to implement a backtracking line-search method based on the Armijo-Goldstein condition, we must first define an inner product on  $T_{p_{id}}(\mathcal{P})$ .

**Definition 11.** For  $v_1, v_2 \in T_{p_{id}}(\mathcal{P})$  with  $v_1 = (y_1, z_1)$  and  $v_2 = (y_2, z_2)$ , the chosen inner product on  $\mathcal{P}$  is given by  $\langle v_1, v_2 \rangle = y_1 y_2 + \langle z_1, z_2 \rangle$ , where  $\langle \cdot, \cdot \rangle$  is the standard  $\mathbb{L}^2$  inner product, and the corresponding norm is given by  $\|v\|_{\mathcal{P}} = \sqrt{\langle v, v \rangle}$ .

With this inner product and norm for elements of the tangent space, we can define the Armijo-Goldstein condition for the backtracking line-search method.

**Definition 12. (Armijo-Goldstein Condition)** For a candidate update  $p = (\xi, \psi) \in \mathcal{P}$ , a scalar  $\beta \in (0, 1)$ , a search direction  $v_k \in T_{p_{id}}(\mathcal{P})$ , and a stepsize  $\delta > 0$ , the Armijo-Goldstein condition is given by  $E_k(p) \leq E_k(p_{id}) + \frac{\beta \delta}{\|v_k\|_{\mathcal{P}}} \langle \nabla E_k(p_{id}), v_k \rangle$ .

One can see that in the special case of gradient descent where  $v_k = -\nabla E_k(p_{id})$ , the above condition simplifies to  $E_k(p) \leq E_k(p_{id}) - \beta \delta \|\nabla E_k(p_{id})\|_{\mathcal{P}}$ . In addition to the Armijo-Goldstein condition, we also need the condition that

$(\psi_k * \psi)(t) \geq 0$  for all  $t \in [0, 1]$  in order to ensure that the incremental update of  $\psi_k$  remains in  $\Psi$ . If either of the two conditions are not satisfied, then one must reduce the stepsize by a factor of  $\tau \in (0, 1)$  by updating  $\delta \mapsto \tau\delta$  until both are satisfied or until  $\delta$  becomes too small.

Algorithm 2 outlines the gradient descent method for elastic partial matching of functions.

**Given:** A pair of censored functions  $q_1, q_2 \in \mathbb{L}_0^2$  with censoring points  $c_1, c_2 \geq 0$ , tolerance  $\epsilon > 0$ , maximum number of iterations  $maxit$ , scalar parameters  $\beta, \tau \in (0, 1)$ , and stepsize  $\delta > 0$ .

**Result:** Optimally aligned censored functions.

```

1 Initialize variables  $k = 0, p_0 = p_{id}$ , and  $q_{2,0} = q_2$ ;
2 Compute  $E_0(p_{id})$  via Eq. 7 and  $\nabla E_0(p_{id})$  via Eqs. 8 and 9;
3 while  $\|\nabla E_k(p_{id})\|_{\mathcal{P}} > \epsilon$  and  $k < maxit$  do
4   Let  $v_k = -\nabla E_k(p_{id})$ ;
5   Let  $\delta_k = \delta$ ;
6   Generate candidate update  $p = \exp_{p_{id}}(\delta_k v_k)$ ;
7   Compute  $E_k(p)$  via Eq. 7;
8   while  $E_k(p) > E_k(p_{id}) - \beta\delta_k \|\nabla E_k(p_{id})\|_{\mathcal{P}}$  or  $(\psi_k * \psi) < 0$  anywhere do
9     Let  $\delta_k = \tau\delta_k$ ;
10    Generate candidate update  $p = \exp_{p_{id}}(\delta_k v_k)$ ;
11    Compute  $E_k(p)$  via Eq. 7;
12  end
13  Update  $q_{2,k+1} = q_{2,k} * g$  with  $g$  identified by  $p = (\xi, \psi)$  and  $b = \min(c_1, c_{2,k}e^{-\xi})$ , and update  $p_{k+1} = p_k \cdot p$ ;
14  Update  $k = k + 1$ ;
15 end
16 Form  $g_k$  from  $p_k = (\xi_k, \psi_k)$  and  $b = \min(c_1, c_2 e^{-\xi_k})$  via Eq. 5. return  $\hat{q}_2 = q_{2,k}$  and  $\hat{g} = g_k$ .
```

**Algorithm 2:** Gradient Descent for Alignment of Right-Censored Functions.

### 3.3 Modification of Energy Function

The energy defined above (Definition 10) depends in part on the  $\mathbb{L}^2$  norm of the extra portion of the longer function (the second term). Therefore, if that piece has a large  $\mathbb{L}^2$  norm, then it tends to dominate the total energy. In that case, an optimal alignment pushes the endpoints of the two functions to be closer together, and the results look more similar to the standard elastic registration with identical boundaries. In order to control the influence of this unmatched part, we modify the energy functional by multiplying the second term in  $E$  by a constant  $\lambda > 0$ , which results in:

$$E(\xi, \psi) = \int_0^b (q_1(t) - q_2(e^{\xi}b(\int_0^{t/b} \psi(s)^2 ds)))e^{\xi/2}\psi(t/b))^2 dt + \lambda \int_b^\infty (q_1(t) - q_2(e^{\xi}t)e^{\xi/2})^2 dt, \quad (10)$$

where  $b = \min(c_1, c_2 e^{-\xi})$ . The resulting gradient expressions are altered in only a minor fashion and for the sake of brevity are not repeated here.

We note that for  $\lambda \neq 1$ , the energy  $E$  is no longer the square of a proper metric, and some of the nice mathematical properties of a quotient space metric are lost. The energy minimization process results in a measure of “dissimilarity” rather than a proper distance squared. One can still perform a clustering analysis with these tools, as we demonstrate in the next Section.

## 4 EXPERIMENTAL RESULTS

In this section we present some experiments demonstrating the strengths of this framework. We first introduce the two functional data sets – one simulated and one real – that we use to demonstrate and compare methods.

**Dataset 1:** The simulated data set consists of 51 functions, each with 100 sample points, and separated into three classes of 17 functions each. The functions are all based on a mixture of three Gaussians on the interval  $[0, 1]$  with means at  $t = 0.3$ ,  $t = 0.5$ , and  $t = 0.7$  and with equal variance. The functions within each class have the same mixture coefficients, where, for class 1, 2 and 3, we use the coefficient pairs  $(0.2, 0.5, 0.8)$ ,  $(0.5, 0.5, 0.5)$ , and  $(0.8, 0.5, 0.2)$ , respectively. To generate a function  $f$  in the simulated dataset, we start with the appropriate Gaussian mixture on  $[0, 1]$  and alter it in the following manner: (1) select  $b \sim \text{Uniform}[0.625, 1]$  and truncate  $f$  to the interval  $[0, b]$ ; (2) select  $a \sim \text{Uniform}[0.9, 1.1]$  and apply the time-scaling function  $h(t) = at$  to  $f$ ; and (3) apply a random time-warping diffeomorphism in  $\Gamma_{b/a}$  to  $f$ . Fig. 4 shows the resulting simulated data set, colored according to class label with original uncensored functions plotted as dashed black lines.

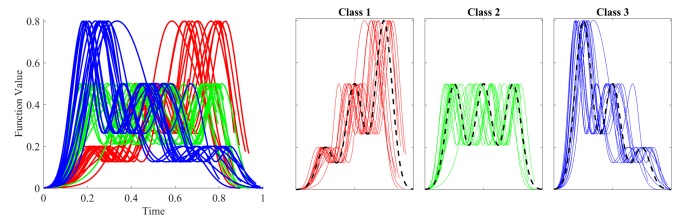


Fig. 4. Simulated data set with three classes. The left-most panel shows all functions plotted in the same window and colored according to class label. The remaining three panels plot each of the three classes separately.

**Dataset 2:** The real data used here comes from daily COVID-19 infection rates for each of the 50 United States plus the District of Columbia and Puerto Rico plus 47 European countries. The daily new case data is preprocessed via the following procedure. We translate each curve so that  $t = 0$  represents the day of the first recorded infection. Thus,  $t \in [0, \infty)$  represents the number of days since a state or country’s first case. Then, we apply a seven day moving average to smooth the data, and then we re-sample each curve via spline interpolation to have 100 uniformly spaced time sample points. Finally, we normalize the rate curve so that it has integral 1. We execute this process three times, each on raw data truncated at different ending dates – July 31, September 30, and November 30 of the year



2020 – to create three datasets of normalized infection rate curves. Fig. 5 shows a plot of all 99 normalized infection rate curves for the United States and Europe. The first two columns separate the US and European curves, respectively, for visual purposes.

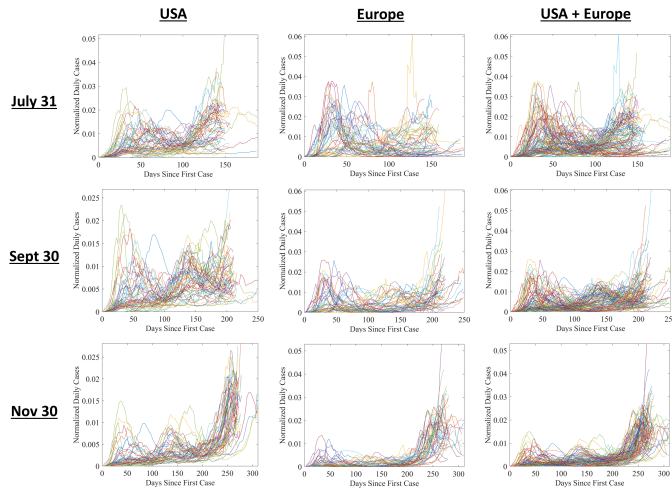


Fig. 5. United States and European COVID-19 normalized infection rate curves truncated at three different right boundaries – July 31, September 30, and November 30 of the year 2020. For each panel, the time axis indicates the number of days since the first recorded case. In the first and second columns, the US and European data is plotted separately for ease of visualization, and in the third column they are plotted together.

#### 4.1 Pairwise Alignment Examples

Fig. 6 shows a few examples of pairwise alignments for both the simulated and the real data sets. For each function pair we compute both the standard elastic alignment with fixed and matched endpoints and our novel elastic partial matching with floating endpoints. For the partial matching, we use the modified energy function in Eq. 10 with  $\lambda = 0.01$  for the simulated data and  $\lambda = 0.25$  for the COVID-19 data. In each panel, we plot the original censored functions  $f_1$  and  $f_2$  in blue and red, respectively; the elastically registered version of  $f_2$  in yellow with matched right endpoint; and the partially matched version via the methodology developed in Sections 2 and 3 in purple. In each case, the alignments are all noticeably different visually, with our elastic partial matching methodology providing a more natural registration than the standard elastic methodology. For some examples, we see that the curve with standard elastic registration is similar to our partially matched curve on a portion of the domain. However, as it approaches the endpoint of the fixed blue curve, the elastic registration tends to unnaturally compress or stretch the yellow curve in order to force the right endpoints to match. In other cases, the freedom offered by our partial matching methodology allows us to find a completely different matching across the entire domain. In particular, the elastic partial matching of North Dakota to Sweden's infection rate curve allows for a surprisingly similar shape matching that the standard elastic registration was unable to uncover.

Below each pairwise matching result, we also show the corresponding optimal diffeomorphisms for the two elastic

methods. Using the same color scheme and the same domain (time axis) scale, we plot the optimal  $\hat{\gamma} \in \Gamma_b$  and  $\hat{g} \in G$  associated with the two elastic registrations. The circle along the purple colored diffeomorphism  $\hat{g} \in G$  represents the point  $\hat{g}(\hat{b})$ , where  $\hat{b} = \min\{c_1, c_2\}$  is the pivot point. Recall that beyond this point the function  $\hat{g}$  is linear with slope  $\hat{g}(\hat{b})/\hat{b}$  and extends to infinity. In cases where the optimally matched second function (purple) is shorter than the first (blue), we do not plot  $\hat{g}(t)$  beyond the pivot point  $t = \hat{b}$ .

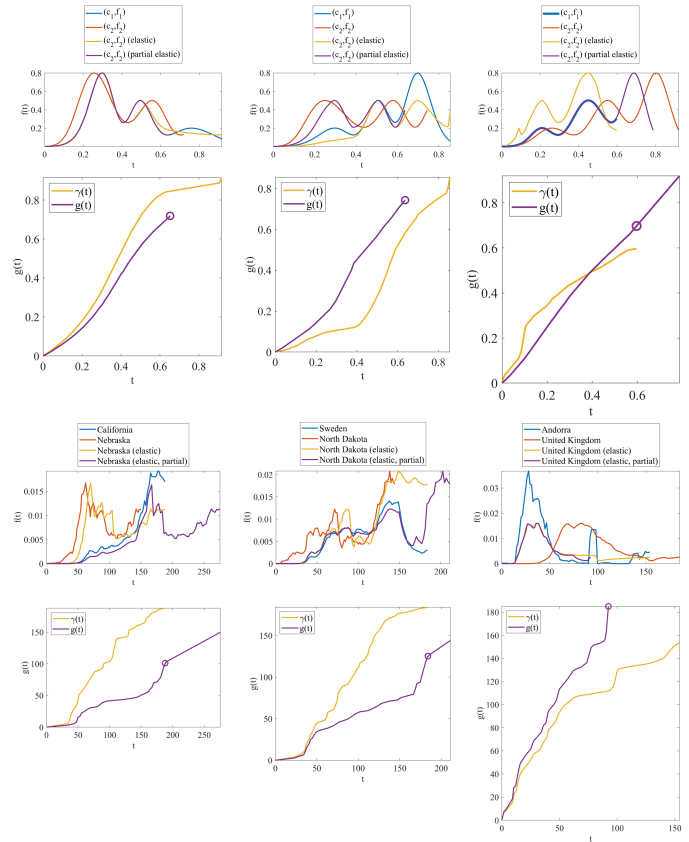


Fig. 6. Examples of pairwise elastic function registrations. The first row shows three pairwise alignment examples for the simulated dataset. In each panel, the blue and red curves are the original censored functions  $f_1$  and  $f_2$ , the yellow curve is the aligned version of  $f_2$  using standard elastic registration with fixed and matched endpoints, and the purple curve is the aligned version of  $f_2$  using our novel elastic partial matching methodology with floating right endpoint. The second row provides the associated optimal diffeomorphisms that achieve the alignments, plotted in the same color scheme and time axis scale. The third and fourth rows are the same as the first and second but instead using examples from the COVID-19 data set. Here, we restrict example pairs to be from the set of US states and European countries truncated at July 31.

#### 4.2 Algorithm Performance and Parameter Selection

##### Performance Analysis:

Here, we present four numerical experiments to demonstrate the numerical performance of the elastic partial matching algorithm under various settings.

- Experiment (1): Using the COVID dataset, we compute and plot the average computation time versus the number of function sample points  $N_f$  for three algorithm configurations: (i) Using only the grid-search algorithm; (ii) Using the gradient descent



algorithm initialized with the original  $f_2$ , i.e. “no init”; and (iii) Using the gradient descent algorithm initialized with the grid-search result from (i).

- Experiment (2): For both the COVID and the simulated datasets, we compute the average shape distance  $d_s$  obtained from using the “no init” gradient descent method as well as using the grid-search initialized gradient descent method.
- Experiment (3): For both the COVID and the simulated datasets, we compute the average shape distance  $d_s$  versus the number of grid points  $J$  in the grid-search initialization to the gradient descent method.
- Experiment (4): For both the COVID and the simulated datasets, we test the effect of  $\lambda$  on the partial matching result by computing the average “overlap ratio,” i.e. the proportion of function domain overlap after alignment, for several values of  $\lambda$ .

The parameters used for each experiment are the following. For all implementations of the gradient descent method, we use  $\epsilon = 10^{-3}$ ,  $\delta = 10^{-4}$ ,  $maxit = 300$ ,  $\beta = 0.1$ , and  $\tau = 0.5$ . Also, for all implementations of the grid-search method, for any value of  $J$ , we set the grid  $\{\xi_i, i = 1, \dots, J\}$  to be  $J$  uniformly spaced values between  $\xi_1 = -\log(2)$  and  $\xi_J = \log(2)$ . Note that we form  $a_i = e^{\xi_i}$  for use in Algorithm 1. In Experiments (1), (2), and (3) where  $\lambda$  is fixed, we use  $\lambda = 0.01$  for the simulated dataset and  $\lambda = 0.25$  for the COVID dataset. In Experiments (2), (3), and (4) where  $N_f$  is fixed, we use  $N_f = 100$ . In Experiments (1), (2), and (4) where  $J$  is fixed, we use  $J = 50$ . For Experiments (1), (2), and (3), we average results over 25 uniquely and randomly selected dataset pairs, and for Experiment (4), we use 100 unique and random dataset pairs. For Experiment (1) we use  $N_f = [25, 38, 56, 84, 127, 190, 285, 427, 641]$ , for Experiment (3), we use  $J = [3, 6, 12, 24, 48, 96, 192, 384]$ , and for Experiment (4) we use  $\lambda = [10^{-2}, \dots, 10^1]$  where the exponents are 10 uniformly spaced values between  $-2$  and  $1$ . Fig. 7 plots the results of Experiments (1) through (4) from left to right.

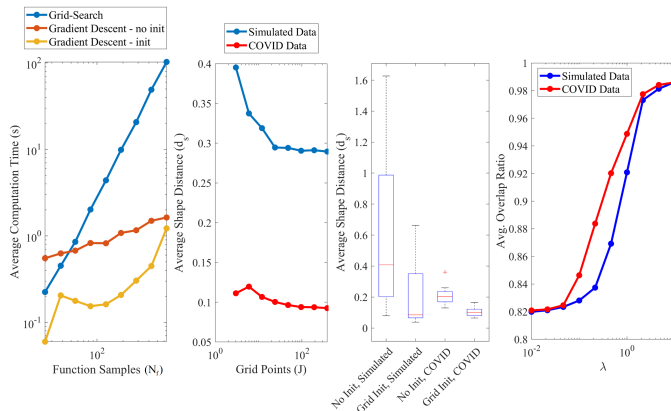


Fig. 7. Four numerical experiments to test our elastic partial matching algorithm performance. The results of Experiments (1) through (4) are shown from left to right.

For Experiment (1), one can see from the left-most plot of Fig. 7 that the gradient descent computation time scales well in  $N_f$  compared to that of the grid-search algorithm.

The complexity of the grid-search algorithm is determined almost entirely by Step 7 of Algorithm 1, the Dynamic Programming algorithm. One can also see that the gradient descent time is reduced for any value of  $N_f$  when provided with a good initialization. In fact, Experiment (2) shows that not only is the computation time reduced, but also the solution quality improves with a good initialization. Since the optimization is complex and over a high dimensional space, the gradient descent method returns a highly localized solution. However, the grid-search method using the Dynamic Programming subroutine achieves a globally optimal solution based on the selected time-scaling grid points and function discretization. Thus, using the gradient descent method as a local refinement to the grid-search result can yield near globally optimal results with large enough number of grid points and function samples. Experiment (3) explores the trade-off between the number of grid points  $J$  over a fixed range and the solution quality for each dataset. In both cases, the average shape distance improves only slightly after  $J = 24$ . Ultimately, since the computation time is not overwhelmingly large for  $N_f = 100$ , we select a larger value of  $J = 50$  for the remainder of our results.

*Remark:* Another important setting for the grid-search algorithm is the range of the grid points. Here, we use a range that allows for proposed censoring points to lie between  $c_2/2$  and  $2c_2$ , i.e. between half and twice the original censoring point. Of course, since the optimal solution of Algorithm 1 serves as the initialization for the gradient descent algorithm, the final censoring point  $\hat{c}_2$  could lie anywhere on  $\mathbb{R}_+$ . However, since there are typically many local solutions to the energy functional, the grid range tends to influence the range of  $\hat{c}_2$  greatly.

#### Selection of $\lambda$ :

In addition to the range of the grid points, the parameter  $\lambda$  in Eq. 10 is a free parameter that controls the amount of time-scaling freedom, or endpoint disparity, allowed by the elastic partial matching algorithms. A lower value of  $\lambda$  allows for a potentially larger difference in right endpoints as the optimal solution; whereas, a higher value of  $\lambda$  will tend to force the endpoints to be closer together. The selection of  $\lambda$  could be guided by the application at hand and any prior knowledge of its physical limitations. For example, if the dataset is of human growth rate curves measured up to a particular age, as in the Berkeley growth curves [24], it make senses to reduce endpoint disparity. Individuals’ biological clocks vary but not by a lot. However, with states’ virus infection rate curves, there are many more factors at play that could accelerate or delay virus reproduction over time, such as population density and local policies. For this reason, it makes sense to allow for more endpoint disparity for the COVID-19 application than for human growth rate curves.

Experiment (4) helps to guide us to a data-driven selection of  $\lambda$  for each dataset using the overlap ratio, which is defined for a pairwise alignment as  $\rho = \min(\{c_1, \hat{c}_2\}) / \max(\{c_1, \hat{c}_2\})$ . As shown in the right-most panel of Fig. 7, which plots the average overlap ratio  $\bar{\rho}$  versus  $\lambda$ , the curves resemble logistic-type S-curves with an upper asymptote of 1 and lower asymptote of about 0.82 in both cases. With this plot, we select an appropriate  $\lambda$  for

each dataset as follows. For the simulated data, we select a relatively small value of  $\lambda = 0.01$  according to the  $\bar{\rho}$  vs  $\lambda$  plot to allow for more flexible endpoints. For the COVID dataset, we anticipate smaller flexibility and select a  $\lambda = 0.25$  that yields an intermediary value of  $\bar{\rho}$ , the inflection point of the  $\bar{\rho}$  vs  $\lambda$  S-curve.

### 4.3 Bayesian Clustering of Functional Data

Next, we perform unsupervised clustering of the data using three different pairwise distance (or dissimilarity) measures and compare the results. The three methods are: (a) the standard  $\mathbb{L}^2$  metric with fixed and matched right endpoints, (b) the standard elastic metric with fixed and matched right endpoints, and (c) our novel elastic partial matching of SRVFs with floating right endpoints. Note that method (a) operates on the function values themselves while methods (b) and (c) use SRVFs for comparison. For method (c), we use the grid-search method (Algorithm 1) to initialize the gradient descent method (Algorithm 2). For Algorithm 1, we use the same grid that we used for Experiments (1), (3), and (4) given in Subsection 4.2; additionally, we use the same gradient descent input parameters for Algorithm 2 as stated in that same Subsection. Finally, as stated earlier, for method (c) we use a weighting coefficient of  $\lambda = 0.01$  for the simulated data and  $\lambda = 0.25$  for the COVID data, and thus the pairwise comparison in this method is not a proper distance but rather a more general dissimilarity measure.

We use the clustering technique described in [34] to cluster the data for each case. This clustering method is Bayesian in nature and uses only a pairwise similarity matrix  $S$  as input – not the functional data itself – to determine the optimal number of clusters and the cluster members in an MCMC approach. It assumes a Wishart prior on the similarity matrix and uses a variant of the Chinese Restaurant Process to help determine the number of clusters. In the associated clustering software there are several hyperparameters that must be set; however, we simply use all default settings provided in the code, with an initial number of clusters set to 3 for the simulated data and 10 for the real data, to obtain clustering results. Here, we use the conversion formula  $S_{ij} = 1 - D_{ij} / \max\{D\}$ , where  $D$  is the pairwise distance (or dissimilarity) matrix. For method (c), the entries  $D_{ij}$  are equal to the square root of the energy functional after alignment. This conversion formula ensures that all entries of the similarity matrix  $S$  are scaled similarly between 0 and 1.

**Dataset 1:** Fig. 8 shows the result of the Bayesian clustering on the simulated dataset for each of the three methods: (a), (b), and (c). From left to right, we show the pairwise similarity matrices, the color-coded block diagonal class inclusion matrices, the function clusters with consistent color scheme, and the associated cluster means for each cluster. In order to form block diagonal matrices, the data set indices have been optimally permuted as a result of the clustering algorithm. One can see that the number of clusters (3) is correct only for the elastic partial matching method shown in row (c). Furthermore, the cluster labeling is 100% correct for this method.

In both methods (b) and (c), the tiled cluster display shows functions that have been jointly aligned according

to their respective alignment methods. For method (b), the joint alignment is a standard one, but for method (c) we use the following iterative approach. The idea is to find a template with the most “information” and then align each function to this template, in each iteration. How is such a template found? For any two functions, we first align them using our method and label the one with the larger censoring point (after the registration) as the one with more information. For a set of  $K$  functions, we select the initial template to be the function with the largest censoring point and then perform  $K - 1$  pairwise alignments to that function. If after the alignments the function with the largest censoring point is the same function as the template, we have found the function with the most information and terminate the iterations; otherwise, we repeat the process with the new template function set as the function with the largest censoring point after the alignments. One can see that the functions align well within the clusters for our method, showcasing the three distinct classes. Since the data is partially observed, the other two methods tend to mislabel and/or divide the three classes unnecessarily into further subclasses.

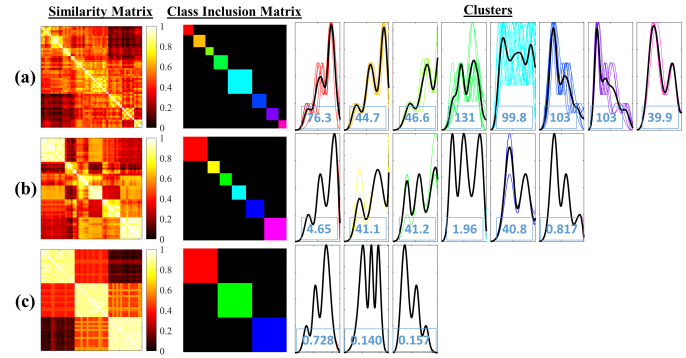


Fig. 8. Simulated data set classes resulting from the implementation of the Bayesian clustering algorithm. The top row shows the results for method (a), the  $\mathbb{L}^2$  metric with fixed and identical endpoints; the middle row shows the results for method (b), the standard elastic metric with fixed and identical endpoints; and the bottom row shows results for method (c), our novel elastic partial matching. The first two columns show the pairwise similarity matrices and the corresponding block diagonal class inclusion matrices, colorized according to class label. The remaining panels are the individual clusters plotted separately, with each  $y$ -axis scaled so that the maximum value is equal to that of the maximum  $y$ -value of the associated cluster. For methods (b) and (c), plots of individual classes show function members that have been mutually aligned within classes. The number in the bottom of each panel here represents the average cross-sectional variance for that cluster ( $\times 10^{-4}$ ).

The right-most panels in Fig. 8 show the cluster members as well as the cross-sectional pointwise mean for each cluster. Additionally, we compute the average cross-sectional variance and display that value in the bottom of each panel. As is evident from these numbers, the standard  $\mathbb{L}^2$  analysis leads to artificially inflated variances due to lack of alignment. The standard elastic analysis helps to bring down the variance within clusters by aligning peaks and valleys, but since the data is partially observed, the restriction of identical right endpoints still drives up the variance unnecessarily. The average cross-sectional variances are  $80.5 \times 10^{-4}$  for method (a),  $21.8 \times 10^{-4}$  for method (b), and  $0.342 \times 10^{-4}$  for method (c).

**Dataset 2:** Next, we apply the clustering algorithm on the real COVID-19 datasets. Figs. 9, 10, and 11 show the results of the Bayesian clustering using methods (a), (b), and (c), respectively. The organization of each of the three figures is the following. Groups of three rows show results for data up to July 31, September 30, and November 30, respectively. The first group of three rows shows the color-coded country/state maps, the permuted pairwise similarity matrices, and the associated block-diagonal class inclusion matrices with the same color scheme as their associated maps. The second group of three rows shows the resulting function clusters along with their cross-sectional mean functions in black and average cross-sectional variance value. Similar to Fig. 8 with methods (b) and (c), the cluster members in Figs. 10 and 11 are mutually aligned within clusters in the tiled cluster display. Also similarly, the vertical axis has been scaled in each cluster tile so that the maximum value is equal to the maximum function value in each cluster set.

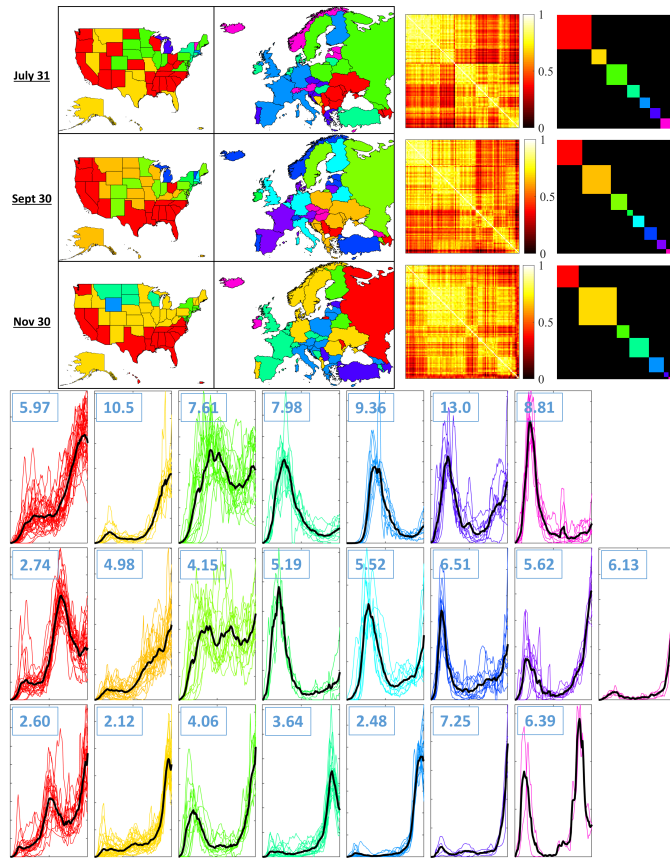


Fig. 9. Clustering results on the COVID-19 data set for the  $L^2$  metric, method (a). Rows 1 and 4 correspond to the July 31 dataset, rows 2 and 5 correspond to the September 30 dataset, and rows 3 and 6 to the November 30 dataset. The first three rows show a US state & European country map colorized according to class label, the pairwise similarity matrix, and the pairwise class inclusion matrix. The next three rows show the corresponding clusters and their cross-sectional mean curves in black. The average cross-sectional variance ( $\times 10^{-6}$ ) is shown in the upper left corner of each cluster panel.

One can draw the following inferences from the COVID-19 results. Firstly, there is a high degree of geographical correlation of cluster members; i.e. neighboring states/countries are typically more likely to be of the same

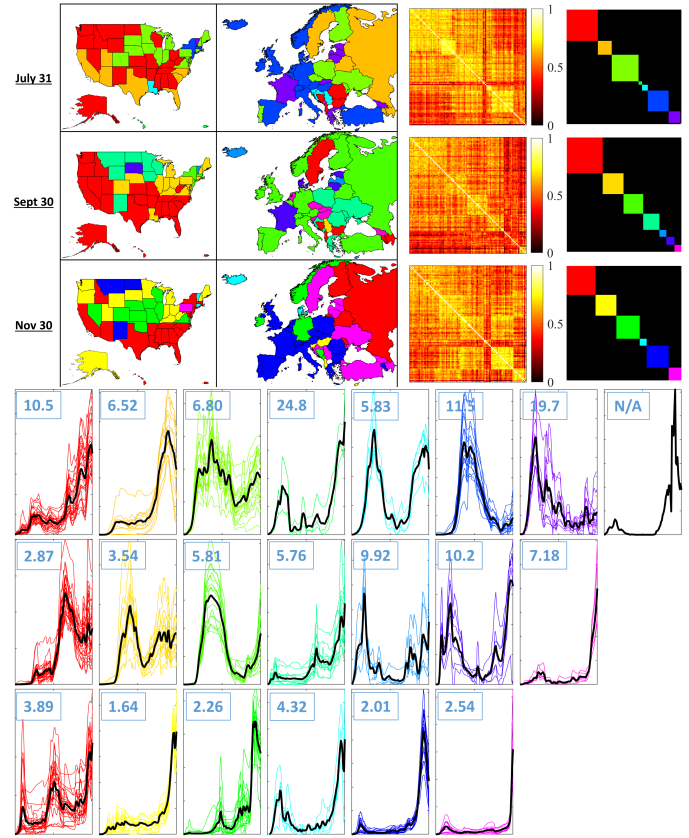


Fig. 10. Clustering results on the COVID-19 data set for the elastic metric with fixed endpoints, method (b). The figure description is the same as that of Fig. 9.

cluster member than not. This geographical correlation is stronger for method (c) compared to others. Secondly, method (c) yields fewer clusters than the other two methods on average due to its relative invariance to partial observations. Since method (c) tends to eliminate clusters with redundant shapes, it preserves true virus trajectory features, and groups states/countries more accurately compared to methods (a) and (b). Note that shape-based features that discriminate between clusters present themselves as the number and relative intensity of waves of virus transmission. Finally, we can see that method (c) yields the lowest within-class cross-sectional variance on average per cluster, suggesting that these clusters are the tightest overall of the three methods. The mean value over all clusters for each of the three methods are the following: (a)  $6.03 \times 10^{-6}$ , (b)  $7.38 \times 10^{-6}$ , and (c)  $5.82 \times 10^{-6}$ .

Another important observation that we can make is that the COVID-19 data clusters are more similar to each other with respect to method (c) than with other methods. This phenomenon is in stark contrast to that of the simulated data, where clusters are very distinct and separable. Fig. 12 provides evidence of the above claim. Here, each panel shows the kernel density estimated probability density functions for within-class similarity values (blue) and across-class similarity values (red) for each dataset and method. The number shown in each panel is our measure of density separability, which is computed as  $\cos^{-1}(\langle \sqrt{f_{in}}, \sqrt{f_{out}} \rangle)$ , where  $f_{in}$  is the estimated within-class similarity value



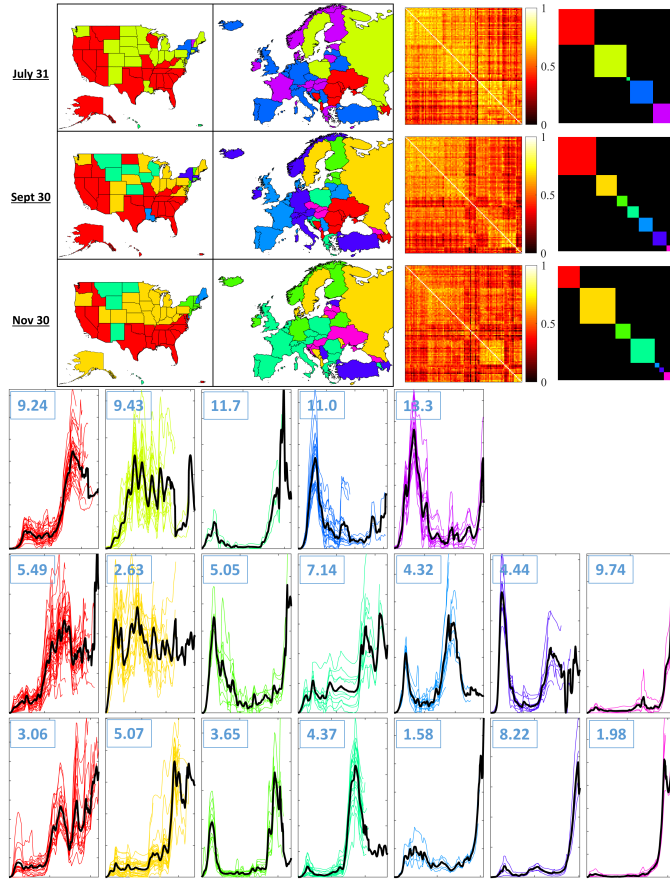


Fig. 11. Clustering results on the COVID-19 data set for the elastic metric with partial matching, method (c). The figure description is the same as that of Fig. 9.

density and  $f_{out}$  is the estimated across-class similarity value density (see [27]). The separability measure for the simulated data increases for methods (a), (b), and (c), respectively, with values of 1.03, 1.33, and 1.57; contrarily, for the COVID data the separability values decrease for methods (a), (b), and (c), respectively, with values of 0.904, 0.684, and 0.634. This result provides evidence that states and countries have overall more similar virus trajectories with respect to method (c) than the other methods, and since the opposite effect is observed in simulated data, this phenomenon is due to the input data and not an artifact of the methodology used.

## 5 SUMMARY & FUTURE WORK

This paper presents a novel approach to overcoming an important limitation of the past works in elastic matching, comparisons, and analysis of functional data, namely that the boundaries are fixed and registered. In addition to exhibiting phase variability within the observation interval, real data often also exhibit boundary variability, and the past methods fail to account for such boundary issues. The proposed elastic partial matching method forms a joint action of time-warping and time-scaling groups and searches for an optimal nonlinear warping along with a sliding right boundary to best match functions. As seen through experiments on simulated and COVID-19 rate curves, this

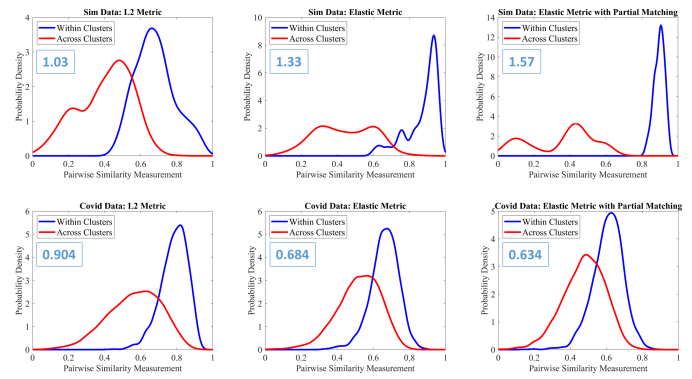


Fig. 12. Estimated density functions of within-class and across-class pairwise similarity values. The top row corresponds to the simulated dataset, and the bottom row to the real COVID-19 datasets. The columns from left to right correspond to methods (a), (b), and (c) respectively. The number value in the top left of each panel indicates the distance measure, or separability, of each pair.

additional freedom allows for more natural alignments and tighter, more visually distinct clusters compared to past methods.

Although the use of a Riemannian elastic framework was motivated from a statistical perspective, the subsequent statistical tools are not developed in this paper due to a lack of space. The shape distance  $d_s$  defined here is a proper distance on the quotient space  $\mathcal{F}_0/G$  and can be used to define sample means, sample covariance, and tangent PCA based statistical models. These models can be used further for testing and classification of functional data.

This framework can easily be applied to the problem of partially matching shapes of curves in Euclidean spaces [1], [11], [26], [33]. Very often, planar curves extracted from image data suffer from partial obscuration and missing parts. The proposed elastic partial framework naturally extends to curves in  $\mathbb{R}^n$  and can be very useful in matching partially observed shapes in that context. Another important application of our elastic partial matching framework is human activity analysis [3], [4], [32]. As stated in these papers, classification of observed activities requires temporal synchronization, modulo variable execution rates, and this matching process can benefit from flexible boundaries.

## ACKNOWLEDGEMENTS

The authors are grateful to Prof. Eric Klassen of FSU for helpful discussions on topics covered in this paper. This research was supported in part by the grants NSF CDS&E DMS 1621787, NIH R01 GM135927, and NSF CDS&E DMS 1953087 to AS; and by the Naval Innovative Science and Engineering (NISE) program at the Naval Surface Warfare Center Panama City Division (NSWC PCD) to DB.

## REFERENCES

- [1] Maheshwari A., Sack JR., Shahbaz K., and Zarrabi-Zadeh H. Improved algorithms for partial curve matching. In *Algorithms – ESA 2011. Lecture Notes in Computer Science*, vol 6942. Springer, Berlin, Heidelberg, 2011.
- [2] H. Alt and M. Godau. Computing the Frechet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5:75 – 91, March 1995.

- [3] B. Ben Amor, J. Su, and A. Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):1–13, 2016.
- [4] R. Anirudh, P. Turaga, J. Su, and A. Srivastava. Elastic functional coding of riemannian trajectories. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 39(5):922–936, 2017.
- [5] Altendji Belkis, Jacques Demongeot, Ali Laksaci, and Mustapha Rachdi. Functional data analysis: Estimation of the relative error in functional regression under random left-truncation. *Journal of Nonparametric Statistics*, 30(2):472–490, 2018.
- [6] K. Buchin, M. Buchin, and Y. Wang. Exact algorithms for partial curve matching via the Frechet distance. In *Proceedings of the 2009 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 645 – 654, January 2009.
- [7] Kong D, Ibrahim JG, Lee E, and Zhu H. FLCRM: Functional linear cox regression model. *Biometrics*, 74(1):109–117, 2018.
- [8] Aurore Delaigle and Peter Hall. Classification using censored functional data. *Journal of the American Statistical Association*, 108(504):1269–1283, 2013.
- [9] Angelo F Elmi. *Curve registration in functional data analysis with informatively censored event-times*. PhD thesis, Univ. of Penn., 2009.
- [10] F Ferraty and P Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer Series in Statistics. Springer New York, 2006.
- [11] T. Funkhouser and P. Shilane. Partial matching of 3d shapes with priority-driven search. In *Eurographics Symposium on Geometry Processing*, 2006.
- [12] Tailen Hsing and Randall Eubank. *Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators*. Wiley, 2015.
- [13] W. Huang, K. A Gallivan, A. Srivastava, and P-A. Absil. Riemannian optimization for registration of curves in elastic shape analysis. *Journal of Mathematical Imaging and Vision*, 54(3):320–343, 2016.
- [14] John P. Klein and Melvin L. Moeschberger. *Survival Analysis: Techniques for Censored and Truncated Data*. Springer Verlag New York, 2003.
- [15] A. Kneip and J. O. Ramsay. Combining registration and fitting for functional models. *Journal of the American Statistical Association*, 103(483), 2008.
- [16] Piotr Kokoszka and Matthew Reimherr. *Introduction to Functional Data Analysis*. Chapman and Hall, 2017.
- [17] Sungwon Lee and Sungkyu Jung. Combined analysis of amplitude and phase variations in functional data. *arXiv:1603.01775v2*, 2017.
- [18] J. S. Marron, J. O. Ramsay, L. M. Sangalli, and A. Srivastava. Statistics of time warpings and phase variations. *Electronic Journal of Statistics*, 8(2):1697–1702, 2014.
- [19] J. S. Marron, J. O. Ramsay, L. M. Sangalli, and A. Srivastava. Functional data analysis of amplitude and phase variation. *Statistical Science*, 30(4):468–484, November 2015.
- [20] J. C. McBride and B. B. Kimia. Archaeological fragment reconstruction using curve-matching. In *2003 Conference on Computer Vision and Pattern Recognition Workshop*, volume 1, 2003.
- [21] M.Cui, J.Femiani, J.Hu, P.Wonka, and A.Razdan. Curve matching for open 2d curves. *Pattern Recognition Letters*, 30:1–10, 2009.
- [22] J. O. Ramsay and X. Li. Curve registration. *Journal of the Royal Statistical Society, Ser. B*, 60:351–363, 1998.
- [23] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis, Second Edition*. Springer Series in Statistics, 2005.
- [24] J.O. Ramsay, R.D. Bock, and T. Gasser. Comparison of height acceleration curves in the fels, zurich, and berkeley growth data. *Annals of Human Biology*, 22(5):413–426, 1999.
- [25] D. Robinson. *Functional data analysis and partial shape matching in square-root velocity framework*. FSU Dissertaion, 2012.
- [26] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003.
- [27] A. Srivastava and E. Klassen. *Functional and Shape Data Analysis*. Springer Series in Statistics, 2016.
- [28] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, 2011.
- [29] A. Srivastava, W. Wu, S. Kurtel, E. Klassen, and J. S. Marron. Registration of functional data using fisher-rao metric. *arXiv*, arXiv:1103.3817, 2011.
- [30] M. Takagishi and H. Yadohisa. Robust curve registration using the t distribution. *Behaviormetrika*, 46:177–198, 2019.
- [31] J. D. Tucker, W. Wu, and A. Srivastava. Generative models for functional data using phase and amplitude separation. *Computational Statistics and Data Analysis*, 61:50–66, 2013.
- [32] A. Veeraraghavan, A. Srivastava, A. K. Roy-Chowdhury, and R. Chellappa. Rate-invariant recognition of humans and their activities. *IEEE Transactions on Image Processing*, 8(6):1326–1339, June 2009.
- [33] Chengzhuan Yang, Hui Wei, and Qian Yu. A novel method for 2d nonrigid partial shape matching. *Neurocomputing*, 275:1160–1176, 2018.
- [34] Z. Zhang, D. Pati, and A. Srivastava. Bayesian clustering of shapes of curves. *Journal of Statistical Planning and Inference*, 166:171 – 186, 2015.



**Darshan Bryner** Darshan Bryner is a research scientist and branch manager at the Naval Surface Warfare Center Panama City Division (NSWC PCD). He obtained his Ph.D. degree in Statistics from Florida State University in 2013, where his dissertation research focused on developing novel statistical shape models to improve image segmentation quality. As a research scientist at NSWC PCD, Dr. Bryner has conducted basic and applied research in the fields of statistical computer vision, functional data analysis, spatial statistics, optimization on nonlinear manifolds, and bioinformatics, leading to several publications in top-tier peer-reviewed journals and conferences. Since 2017, Dr. Bryner has served as the head of the Advanced Signal Processing and Automatic Target Recognition Branch (Code X23) at NSWC PCD.



**Anuj Srivastava** Anuj Srivastava is a Professor and a Distinguished Research Professor at the Florida State University. His research interests include statistical analysis on nonlinear manifolds, statistical computer vision, functional data analysis, and shape analysis. He is a fellow of AAAS, IAPR, IEEE, and ASA. He has held several visiting positions at European universities, including INRIA, France; the University of Lille, France; and Durham University, UK. He has coauthored more than 230 papers in peer-reviewed journals and top-tier conferences, and also several books, including the 2016 Springer textbook on “Functional and Shape Data Analysis”.