Joint Alignment of Multivariate Quasi-Periodic Functional Data Using Deep Learning

Vi Thanh Pham[1], Jonas Bille Nielsen[2], Klaus Fuglsang Kofoed[2,3], Jørgen Tobias Kühl[4], and Andreas Kryger Jensen[1]

[1]Section of Biostatistics, Department of Public Health, Faculty of Health and Medical Sciences, University of Copenhagen

[2]Department of Cardiology and Radiology, Copenhagen University Hospital

[3]Department of Clinical Medicine, Faculty of Health and Medical Sciences, University of Copenhagen

[4]Department of Cardiology, Zealand University Hospital

Abstract

The joint alignment of multivariate functional data plays an important role in various fields such as signal processing, neuroscience and medicine, including the statistical analysis of data from wearable devices. Traditional methods often ignore the phase variability and instead focus on the variability in the observed amplitude. We present a novel method for joint alignment of multivariate quasi-periodic functions using deep neural networks, decomposing, but retaining all the information in the data by preserving both phase and amplitude variability. Our proposed neural network uses a special activation of the output that builds on the unit simplex transformation, and we utilize a loss function based on the Fisher-Rao metric to train our model. Furthermore, our method is unsupervised and can provide an optimal common template function as well as subject-specific templates. We demonstrate our method on two simulated datasets and one real example, comprising data from 12-lead 10s electrocardiogram recordings.

1 Introduction

Statistical analysis of functional data is becoming increasingly essential in the biological and medical research as technologies allow for measuring subjects over long time durations with potential high-frequency sampling times. When applying statistical inference to functional data, it would be advantageous to align the functions, so that the positions of corresponding peaks and valleys are the same across the data set. Failure to align the data correctly can lead to inefficiency of basic statistical summaries like averages, leading to poor estimates of population mean functions that are not representative of the data. In reality, many commonly used functional data analysis techniques present inferior performance when confronted with phase variation (Marron et al., 2015). Furthermore, alignment can also improve prediction accuracy when classification is the goal (Tucker et al., 2013). In practice, functional data is often misaligned and contains variability along both the x axis, called the *phase* variability, and the y axis, called the *amplitude* variability. The process of decomposing the overall variability in the data into these two components is termed elastic phase-amplitude separation and represents the joint alignment of multiple functions. In this paper, we focus on multivariate quasi-periodic functions. We call a function f multivariate if $f(t) = (f_1(t), f_2(t), \ldots, f_J(t)) \in \mathbb{R}^J$, and following Boucheham (2008), we call a function quasi-periodic if it is a concatenation of similar patterns or pseudo-periods.

Examples of such quasi-periodic functional data are continuous glucose monitoring (CGM) data (Klonoff, 2005), which provide information about levels of blood glucose through wearable devices (McDonnell et al., 2022), and electrocardiogram (ECG) recordings (Gregg et al., 2008), which measure the electrical activity of the heart. We aim to study ECGs as they are regularly used together with other tests to diagnose and monitor diseases affecting the cardiovascular system, as well as to inspect symptoms of possible heart conditions. ECGs can assist with detecting arrhythmias and can be used over time to monitor a person with an existing diagnosis, or a person taking medication that can have an influence on the heart. These measurements taken over time through multiple leads placed on the body can be

regarded as multivariate quasi-periodic functions, where the periods are the heart-beats. However, these periods are not exactly the same across the whole recording, and depend on other physiological activities at the moment, turning the data into quasi-periodic.

One can perform statistical analyses to align and compare these observations using tools such as the L^2 distance to find the cross-sectional mean and variance. In addition to these summary measures, it is of great interest to model the variability in the data, and due to the high dimensionality of this data, we would like to analyze them using neural networks. In the example of the ECG data, we might want to align the peaks and segments to extract a template heartbeat while also recording the relative positions of these features.

In this paper, to jointly (simultaneously) model multiple time warping functions and to align multivariate quasi-periodic functions along the x axis, we modify and extend the elastic phase-amplitude separation algorithm for univariate functions from Srivastava and Klassen (2016) and introduce the algorithm for the Joint Alignment of Multivariate quasi-periodic functional data using Deep learning (Deep-JAM). Importantly, DeepJAM provides us with warping functions, one for each observed function, that contain all the information about the phase variability in the data. In the case of multivariate functional data, all the dimensions share the same warping function, because they are all observed on the same sample. Because our approach is unsupervised, we also utilize nice geometric properties of transformed warping functions to extract a unique multivariate functional template. Subsequently, we can use the DeepJAM neural network to easily align new data to this template. Furthermore, DeepJAM can be used as an integrated part of other neural networks in an end-to-end analysis as opposed to two-step modelling, where alignment is only treated as a pre-processing step.

1.1 Related work

A standard approach for alignment of functions is called landmark registration (Ramsay & Silverman, 2005, chap. 7). Landmarks are typically distinctive features of a function, such as minima, maxima, and zero-crossings of the function itself, or its derivatives. The challenge with landmark registration is that it only focuses on the alignment of specific points. An extension to the landmark registration is template registration (Srivastava & Klassen, 2016, chap. 8.2), which aligns whole functions to a specific template. Kneip and Ramsay (2008) suggest aligning func-

tions to the functional principal components instead, with the idea that aligned functions can be represented as the sum of a mean function and a linear combination of a few principal components. This is done through an iterative process of simultaneous estimation of the mean function, warping functions, functional principal components and their scores. An alternative approach also based on functional principal component analysis (FPCA) is presented in Srivastava and Klassen (2016, chap. 8.8).

Dynamic Time Warping (DTW, Müller, 2007, chap. 4) is another well-known technique for finding an optimal alignment between two time series. Building on DTW, Boulnemour and Boucheham (2018) proposed a combination with the shape exchange algorithm, which should be more suitable for handling quasi-periodic time series. However, in applications, we often need joint alignment of multiple functions, because we are interested in patterns regarding the whole data, such as population mean and variance. This could be done, for example, by finding a representative sample in the data, or by constructing a representative template and then using DTW to align the functions to this template. Nonetheless, DTW does not provide us with this template. Srivastava and Klassen (2016) proposed a method to solve this problem using dynamic programming. Regardless of whether the objective is to achieve pairwise or groupwise alignment, the purpose of the above methods using dynamic programming is mainly to serve as a pre-processing step. In the case when we require an end-to-end analysis using neural networks, we need the alignment approach itself to be based on neural networks, or at least another gradient based approach.

In the deep learning universe, the main focus has been on time warping invariant neural networks (Sun et al., 1992), convolutional neural networks (LeCun et al., 2015) that can be invariant to shifts in the input data, and recurrent neural networks (Tallec & Ollivier, 2018). The challenge with using neural networks that are invariant to time warping or shifts is that they mostly focus on amplitude variability. This may or may not be relevant depending on the specific task. Disregarding the phase variability could result in loss of information and inadequate generative models, while using models for both amplitude and phase can improve classification of future data (Tucker et al., 2013).

Recently, research has also been done on pairwise temporal alignment using neural networks. Nunez and Joshi (2020) proposed a convolutional neural network for the alignment of all pairs of functions in the data, which outputs estimated

warping functions. However, the training data in this particular setting were constructed using DTW, the optimal warping functions from DWT were used in the loss function, and as the authors themselves expressed, their primary motivation was demonstrating reduced computational cost compared to DTW. In addition, the neural network was constructed to deal with pairwise alignment of all functional pairs, but not for joint alignment of multiple functions.

Oh et al. (2018) proposed an end-to-end classification model using a Sequence Transformer Network (STN), which transforms the input signals with parameters learned through a Sequence Transformer convolutional neural network, before proceeding with another neural network for classification. However, this STN focuses on linear transformation and does not implement more complicated time warping of the input signal.

In addition to unsupervised single-class learning, Weber et al. (2019) explored the multi-class case with semi-supervised learning. The implemented model is trained using a loss function that involves the empirical variance of the warped signals and uses a regularization term on the warping as a way to ensure identifiability.

Currently, more and more deep learning methods are developed to align multivariate functions using elastic phase-amplitude separation algorithm; see e.g., the works of Lohit et al. (2019), Nunez et al. (2021) and Chen and Srivastava (2021), all of which use a probabilistic simplex transformation for the derivative of the warping functions by calculating the elementwise product of the neural network output and dividing it by the square of its norm. This transformation is, however, not one-to-one.

1.2 Proposed approach

In this paper, we propose a non-parametric approach for joint alignment of multivariate quasi-periodic functions. We deploy neural networks that aim to extract the optimal warping functions that achieve the smallest distance between the warped functions obtained from the observed functions by warping their domain. We use a special one-to-one activation function in the output layer, so that the output of the neural network satisfies the conditions for warping functions. In addition, instead of using the L^2 metric to calculate the distance between functions directly, we use the Fisher-Rao metric and the square-root slope representation to calculate the distance between the warped functions. This metric is used in the loss function

of the neural network. Furthermore, because we take advantage of the differential geometry of the space of warping functions, we can calculate the Karcher mean of the orbits to extract a functional template.

We make use of convolutional layers with multiple channels to account for multivariate functions, and as a result, the output layer returns a single warping function per subject that can be used for warping all the dimensions of the observed function simultaneously, which is desirable as they are all measured on the same subject. Finally, we incorporate a multiscale warping model to handle quasi-periodic functions.

This paper is structured as follows: In Section 2, we present and review the mathematical formalism behind joint alignment of functional data, which we proceed to extend in Section 2.1 to the case of multivariate quasi-periodic functional data. In the same section, we present the architecture of the deep neural warping network employed in the algorithm. In Section 3, we apply our method to simulated univariate and multivariate functions. In Section 4, we carry out the joint alignment on real ECG data, and finally in Section 5, we discuss the results and limitations. An implementation of our method can be found on the first author's GitHub repository.

2   Joint alignment of multivariate data

In this section, we present the mathematical background for joint alignment of a set of functions $\{f_i\} = \{f_i\}_{i=1}^n = \{f_i \in \mathcal{F}_I \mid i = 1, \ldots, n\}$, where $\mathcal{F}_I$ is the set of absolutely continuous functions defined on the interval I, which we without loss of generality take to be $I = [0, 1]$. Further, we consider multivariate functional data $f_i = (f_{i1}, \ldots, f_{iJ}) \colon I \to \mathbb{R}^J$. To jointly align a set of functions, we need to warp the domain of the functions in a certain constrained way. The constraints for the mapping $\gamma \colon I \to I$ performing the domain warp are that $\gamma(0) = 0$ and $\gamma(1) = 1$ (*boundary-preserving*), that γ is invertible, and that both γ and γ^{-1} are differentiable (*diffeomorphism*). We denote the set of such functions Γ_I. Note that the derivative of γ, denoted as $\dot{\gamma}$, is always positive.

With the definition of the two sets, $\mathcal{F}_I$ and Γ_I, we can describe the two main alignment problems: pairwise and multiple alignment (Srivastava & Klassen, 2016, p. 85). In the pairwise alignment problem, given functions $f_1, f_2 \in \mathcal{F}_I$, we wish to find a warping function $\gamma \in \Gamma_I$, such that some energy term $E[f_1, f_2 \circ \gamma]$, where the symbol "$\circ$" represents function composition, i.e., $(f \circ \gamma)(t) = f(\gamma(t))$, is minimized.

In the case of multivariate functions, i.e., $J \geq 2$, we calculate the function composition elementwise, namely, $f \circ \gamma = (f_1 \circ \gamma, \ldots, f_J \circ \gamma)$. The operations defined below are also elementwise, where relevant. We wish to find γ^* as a solution to $\gamma^* = \operatorname{argmin}_{\gamma \in \Gamma_I} E[f_1, f_2 \circ \gamma]$. The function f_1 is then said to be registered to $f_2 \circ \gamma^*$ for any domain value $t \in I$. The multiple alignment problem is an extension of the pairwise alignment problem, where, given a set of functions $\{f_i\}$, we wish to find a set of warping functions $\{\gamma_i\} = \{\gamma_i\}_{i=1}^n = \{\gamma_i \in \Gamma_I \mid i = 1, \ldots, n\}$, such that $f_i \circ \gamma_i$ are said to be registered to each other over i. The $\{\gamma_i\}$ are called the *phases* and $\{f_i \circ \gamma_i\}$ are representatives of their *amplitude*. Finally, the pairwise solution of the multiple alignment, in which all pairs of functions are registered to each other, can be extended to a template-based registration. Here, we first consider a template function μ, and then we align each of the functions $\{f_i\}$ to this template. These steps can be done iteratively to improve the overall alignment as well as the quality of the template (Srivastava & Klassen, 2016, p. 271). First, we calculate the average of the current versions of $\{f_i\}$ under a proper metric to construct the template μ. Then we align the functions $\{f_i\}$ to this template by calculating the optimal warping functions $\{\gamma_i\}$, and update the functions $\{f_i\}$ by $f_i \leftarrow f_i \circ \gamma_i$, and then we iterate these steps.

A natural way to calculate the average of $\{f_i\}$ for constructing the template would be to use the L^2 norm, yielding the cross-sectional mean of $\{f_i\}$. However, as argued by Srivastava and Klassen (2016, chap. 8.2), the cross-sectional mean of $\{f_i\}$ is not a good representation of the template μ, and in addition, also shown by Srivastava and Klassen (2016, pp. 88–90), the standard L^2 norm is not appropriate as the distance measure due to the lack of isometry under warping, pinching effect, and inverse inconsistency. Instead, we use a specific Riemannian metric, called the Fisher-Rao metric (Srivastava & Klassen, 2016, Definition 4.8., p. 105), together with an alternative functional representation called the square-root slope function representation (SRSF, Srivastava & Klassen, 2016, Definition 4.2., p. 91) defined as

$$q(t) = \operatorname{sign}\left(\dot{f}(t)\right) \sqrt{\left|\dot{f}(t)\right|}. \tag{1}$$

from which the original function can be recovered as

$$f(t) = f(0) + \int_0^t q(s)|q(s)| \, ds. \tag{2}$$

Under this representation, the Fisher-Rao metric becomes the standard L^2 met-

ric (Srivastava & Klassen, 2016, Lemma 4.7., p. 105). Note also that the SRSF representation of a warped function $f \circ \gamma$ is (Srivastava & Klassen, 2016, p. 91)

$$\tilde{q}(t) = (q \circ \gamma)(t)\sqrt{\dot{\gamma}(t)} =: (q, \gamma)(t). \tag{3}$$

Srivastava and Klassen (2016, chap. 8) present an alternative estimator of μ which uses the notion of the amplitude of a function in the SRSF space. Given a function $f \in \mathcal{F}_I$ and its associated SRSF representation, q in Equation (1), the amplitude of f in the SRSF space is defined by the orbit, which is the set of all possible domain transformations according to the group action

$$[q] = \text{closure} \left\{ (q, \gamma) = (q \circ \gamma)\sqrt{\dot{\gamma}} \mid \gamma \in \Gamma_I \right\}.$$

With the definition of the amplitude, Srivastava and Klassen (2016) show that the Karcher mean of the set of orbits $\{[q_i]\}$, as described later in this section, is an estimator of the orbit $[\mu_q]$ and that a specific element of this orbit can be used as an estimator of μ_q, which is the SRSF representation of μ. The template-based alignment problem is then reduced to finding the set $\{\gamma_i\}$ that best aligns the functions $\{q_i\}$ to the template μ_q, which can be formally written as

$$\gamma_i = \underset{\gamma \in \Gamma_I}{\arg\inf} \|\mu_q - (q_i, \gamma)\|. \tag{4}$$

In order to estimate μ_q, we first have to calculate the Karcher mean of the set of orbits $\{[q_i]\}$ that is defined as (Srivastava & Klassen, 2016, Definition 8.1., p. 274)

$$[\mu_q] = \arg\inf_{[q] \in \mathcal{A}} \sum_{i=1}^{n} \inf_{\gamma \in \Gamma_I} (\|q - (q_i, \gamma)\|)^2,$$

where $\mathcal{A}$ is a quotient space $\mathcal{F}_I / \tilde{\Gamma}_I$ and $\tilde{\Gamma}_I$ is a set of boundary preserving weakly increasing absolutely continuous functions $\gamma \colon I \to I$.

The Karcher mean of the amplitudes is again an orbit, and we have to find a particular element of this orbit, specifically its center with respect to the set $\{q_i\}$ (Srivastava & Klassen, 2016, Definition 8.2., p. 275). Such an element μ_q satisfies the property that the Karcher mean of the warping functions $\{\gamma_i\}$, which are the solutions to Equation (4), is the identity warp, $\gamma_{\text{id}}(t) = t$.

The algorithm for finding a center of an orbit with respect to the set $\{q_i\}$ follows Srivastava and Klassen (2016, Algorithm 33., p. 277) with some mod-

ifications. In the first step of the algorithm, we select an element $\tilde{\mu}_q$ of the orbit $[\mu_q]$, i.e., the cross-sectional mean of $\{q_i\}$, and we find $\{\tilde{\gamma}_i\}$ by solving $\tilde{\gamma}_i = \mathrm{arginf}_{\gamma \in \Gamma_I}(\|\tilde{\mu}_q - (q_i, \gamma)\|)$. As opposed to using dynamic programming for this problem as in Srivastava and Klassen (2016), we propose to use a convolutional neural network. A second step of the algorithm is calculating the Karcher mean $\mu_{\tilde{\gamma}}$ of the phases $\{\tilde{\gamma}_i\}$ and finding the center of the orbit $[\mu_q]$ with respect to the set $\{q_i\}$ by

$$\mu_q = \left(\tilde{\mu}_q, \mu_{\tilde{\gamma}}^{-1}\right). \tag{5}$$

Srivastava and Klassen (2016) show that for each q_i,

$$\gamma_i = \tilde{\gamma}_i \circ \mu_{\tilde{\gamma}}^{-1} \tag{6}$$

minimizes $\|\mu_q - (q_i, \gamma)\|$, hence these $\{\gamma_i\}$ are a solution to the joint template-based registration problem represented by Equation (4). The center, μ_q, of the orbit is the estimator of the SRSF of the template function μ.

Karcher mean of warping functions. To define the Karcher mean of a set of warping functions γ_i under the Fisher-Rao metric, we will use the differential geometry of Γ_I. Direct analysis on Γ_I is not straightforward due to it being a non-linear manifold, thus we will work with the SRSF representation of γ_i. The SRSF representation of any $\gamma \in \Gamma_I$ has the form $\psi = \sqrt{\dot{\gamma}}$, which is equivalent to Equation (1) because $\dot{\gamma} > 0$ for all domain values $t \in I$. An advantage of using this representation is that $\|\psi\|^2 = \int_0^1 \psi^2(t)\, dt = \int_0^1 \dot{\gamma}(t)\, dt = \gamma(1) - \gamma(0) = 1$, and thus ψ lies in the positive orthant of the unit Hilbert sphere, $\mathbb{S}_+^\infty = \left\{\psi \in \mathrm{L}^2 \,\middle|\, \|\psi\| = 1, \psi > 0\right\}$. On $\mathbb{S}_+^\infty$, any point $\tilde{\psi} \in \mathbb{S}_+^\infty$ can be projected to the tangent space $T_\psi(\mathbb{S}_+^\infty) = \left\{v \in \mathrm{L}^2 \,\middle|\, \int_0^1 \psi(t)v(t)\, dt = 0\right\}$ at ψ by the inverse exponential map (Srivastava & Klassen, 2016, p. 83) by

$$\left(\exp_\psi^{-1} \tilde{\psi}\right)(t) = \frac{\theta}{\sin(\theta)}\left(\psi(t) - \tilde{\psi}(t)\cos(\theta)\right), \quad t \in [0,1], \tag{7}$$

with $\theta = \cos^{-1}\left(\int_0^1 \psi(t)\tilde{\psi}(t)\, dt\right)$. Similarly, points in the tangent space can be projected back to the unit sphere $\mathbb{S}_+^\infty$ at the point ψ by the exponential map (Srivastava & Klassen, 2016, p. 83)

$$\left(\exp_\psi v\right)(t) = \cos\left(\|v\|\right)\psi(t) + \sin\left(\|v\|\right)\frac{v(t)}{\|v\|}, \quad t \in [0,1]. \tag{8}$$

The algorithm for finding the Karcher mean of warping functions under the Fisher-Rao metric follows Srivastava and Klassen (2016, Algorithm 24, p. 238) and is formally described in Algorithm 1, with an initial estimate of the Karcher mean of the warping functions being the normalized cross-sectional mean of their SRSFs. This procedure is based on a fixed-point algorithm, see e.g., Bhattacharya and Bhattacharya (2012, chap. 5).

Algorithm 1: Karcher mean of warping functions

Input: Warping functions $\{\gamma_i\}_{i=1}^n$, stopping criterion c, maximum number of iterations E, step size ϵ

Output: Karcher mean μ_γ

---

1   $\psi_i \leftarrow \mathrm{SRSF}(\gamma_i)$ /* Calculate SRSF representation of γ_i using (1) */
2   $\mu_\psi \leftarrow \frac{1}{n} \sum_{i=1}^n \psi_i$ /* Initialize mean of SRSFs */
3   $\mu_\psi \leftarrow \mu_\psi / \|\mu_\psi\|$ /* Normalize the mean of SRSFs */
4   $\mu_v \leftarrow 0$ /* Initialize mean in the tangent space */
5   $e \leftarrow 0$ /* Initialize the index of iteration */
6   **while** $(\|\mu_v\| \geq c$ **or** $e = 0)$ **and** $e < E$ **do**
7      $e \leftarrow e + 1$ /* Increase the index of iteration */
8      $\mu_\psi \leftarrow \exp_{\mu_\psi} \epsilon \mu_v$ /* Update mean of SRSFs using (8) */
9      $v_i \leftarrow \exp_{\mu_\psi}^{-1} \psi_i$ /* Project ψ_i to tangent space at μ_ψ using (7) */
10     $\mu_v \leftarrow \frac{1}{n} \sum_{i=1}^n v_i$ /* Calculate mean of projections $\{v_i\}$ */
11 **end**
12 $\mu_\gamma \leftarrow \mathrm{ToWarp}(\mu_\psi)$ /* Calculate mean warping function using (2) */

2.1   Joint alignment of multivariate quasi-periodic data

In this section, we describe the extension of the joint alignment algorithm to multivariate quasi-periodic functions. We define a periodic extension of a function $f\colon [0, \tau] \to \mathbb{R}$ as $\mathrm{ext}_{\mathrm{L}^2}^K f\colon [0, K\tau] \to \mathbb{R}$, $K \in \mathbb{N}$ with

$$\left(\mathrm{ext}_{\mathrm{L}^2}^K f\right)(t + k\tau) \coloneqq f(t) \quad \text{and} \quad \left(\mathrm{ext}_{\mathrm{L}^2}^K f\right)(K\tau) \coloneqq f(\tau), \tag{9}$$

for $t \in [0, \tau)$ and $k = 0, \ldots, K - 1$. We also define a periodic extension of a warping function $\gamma\colon [0, \tau] \to [0, \tau]$ on $[0, K\tau]$ by

$$\left(\mathrm{ext}_{\Gamma}^K \gamma\right)(t + k\tau) \coloneqq \gamma(t) + k\tau \quad \text{and} \quad \left(\mathrm{ext}_{\Gamma}^K \gamma\right)(K\tau) \coloneqq K\tau, \tag{10}$$

for $t \in [0, \tau)$ and $k = 0, \ldots, K-1$. As a dual to the periodic extension, we define a split of a function $f \colon [0, K\tau] \to \mathbb{R}$ in its domain into K functions with $t \in [0, \tau]$ by

$$
\begin{aligned}
\left(\mathrm{spl}^K f\right)(t) &= (f(t), \ldots, f(t + (k-1)\tau), \ldots, f(t + (K-1)\tau)) \\
&\coloneqq (f_1(t), \ldots, f_k(t), \ldots, f_K(t))),
\end{aligned}
\tag{11}
$$

for $k = 1, \ldots, K$. After performing the extension or the split of a warping function, we need to linearly transform the image of the resulting warping functions to the same interval as the domain in order to keep the boundary-preserving property, and furthermore, whenever the exponential and inverse exponential maps, Equations (8) and (7), respectively, need to be used, the domain and image of the warping functions need to be transformed to the interval $[0, 1]$.

The observed multivariate quasi-periodic functions $f_i \colon [0, K\tau] \to \mathbb{R}^J$, where $f_i = (f_{i1}, \ldots, f_{iJ})$ with their SRSF representations $q_i = (q_{i1}, \ldots, q_{iJ})$, are assumed to be generated from a multiscale warping model by

$$
\begin{aligned}
f_{ij}(t) &= \left(\mathrm{ext}_{\mathrm{L}^2}^K \mu_j \circ \mathrm{ext}_\Gamma^K \gamma_i^l \circ \gamma_i^g\right)(t) \\
&= \left(\mathrm{ext}_{\mathrm{L}^2}^K \mu_{ij} \circ \gamma_i^g\right)(t) \\
&= \left(\mathrm{ext}_{\mathrm{L}^2}^K \mu_j \circ \gamma_i^t\right)(t),
\end{aligned}
\tag{12}
$$

where $\gamma_i^l \colon [0, \tau] \to [0, \tau]$ is the *local* warping function and $\mathrm{ext}_\Gamma^K \gamma_i^l \colon [0, K\tau] \to [0, K\tau]$ its periodic extension, $\gamma_i^g \colon [0, K\tau] \to [0, K\tau]$ is the *global* warping function, and $\gamma_i^t \colon [0, K\tau] \to [0, K\tau]$ is the *total* warping function defined as the composition $\mathrm{ext}_\Gamma^K \gamma_i^l \circ \gamma_i^g$, see Figure 1. We think of $\mu = (\mu_1, \ldots, \mu_J)$ as the common template and $\mu_i = \mu \circ \gamma_i^l = (\mu_1 \circ \gamma_i^l, \ldots, \mu_J \circ \gamma_i^l) = (\mu_{i1}, \ldots, \mu_{iJ})$ as the subject-specific template on $[0, \tau]$.

On the other hand, given functions γ_i^l and γ_i^g, or γ_i^t, we can extract the periodic extension $\mathrm{ext}_{\mathrm{L}^2}^K \mu_j$ of the common template function μ_j from an observed function f_{ij} by

$$
\begin{aligned}
\left(\mathrm{ext}_{\mathrm{L}^2}^K \mu_j\right)(t) &= \left(f_{ij} \circ (\gamma_i^t)^{-1}\right)(t) \\
&= \left(f_{ij} \circ (\gamma_i^g)^{-1} \circ \left(\mathrm{ext}_\Gamma^K \gamma_i^l\right)^{-1}\right)(t) \\
&= \left(\mathrm{ext}_{\mathrm{L}^2}^K \mu_{ij} \circ \left(\mathrm{ext}_\Gamma^K \gamma_i^l\right)^{-1}\right)(t).
\end{aligned}
\tag{13}
$$

(a) Common template   (b) Subject-specific template   (c) Observed function

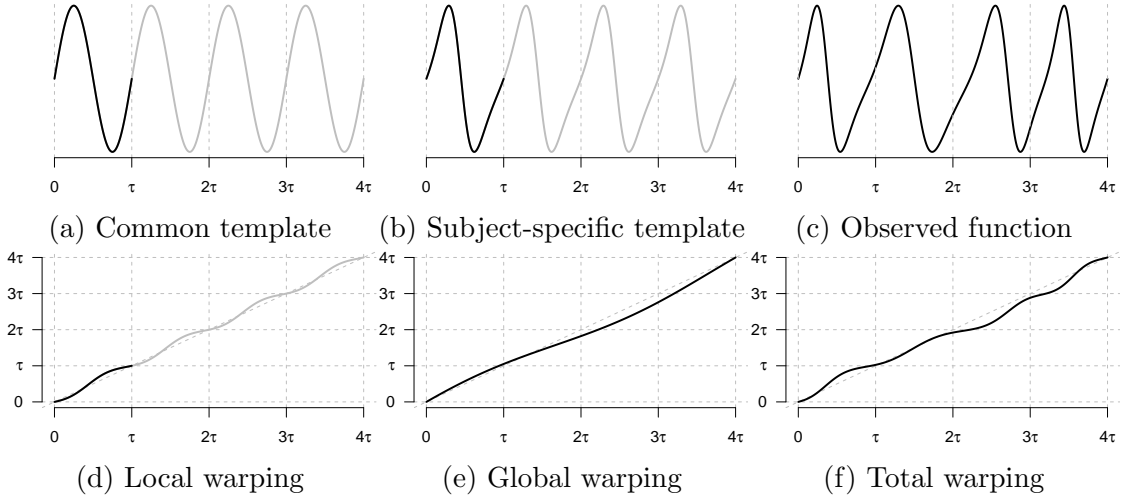(d) Local warping   (e) Global warping   (f) Total warping

Figure 1: Multiscale warping model. The common template is represented by the black line on $[0, \tau]$ in 1a, where the gray line represents the periodic extension of this template on $[0, 4\tau]$. The subject-specific template is shown in 1b, which is a result of warping the common template with the subject-specific local warping function, as represented in 1d. 1c shows the observed function that was obtained from the periodic extension of the subject-specific template warped with the subject-specific global warping function, as represented in 1e. Finally, 1f shows the composition of the subject-specific local and the subject-specific global warping functions.

2.2   Algorithm for joint alignment of multivariate quasi-periodic data using deep learning

The algorithm for adaptive template-based groupwise registration follows Srivastava and Klassen (2016, Algorithm 33., p. 277) with some modifications, and is represented in Algorithm 2, as well as in Figure 2. The most important modifications are replacing dynamic programming (Srivastava & Klassen, 2016, Algorithm 58., p. 437) with a convolutional neural network, and extending the algorithm to handle quasi-periodic multivariate functions.

Subject-specific template. The warping functions $\{\gamma_i\}$ obtained by Algorithm 2 correspond to the inverse of the total warping functions as described in Equations (12) and (13). In order to obtain subject-specific templates, we need to decompose this total warping function into the global and local warping functions, and we propose that this can be done in the following way. Consider the subject i and the corresponding warping function γ_i that aligns the observed function f_i to the periodic extension of the common template, $\mathrm{ext}_{\mathrm{L}^2}^K \mu$. We split the domain of the warping function γ_i into K functions using Equation (11). This yields $(\gamma_{i1}, \ldots, \gamma_{iK})$

12

Algorithm 2: Joint alignment of multivariate quasi-periodic functional data using deep learning: DeepJAM

Input: Observed functions $\{f_i\}_{i=1}^n$ on the interval $I = [0,1]$, number of periods K, number of iterations E, initialized neural network NN

Output: Warping functions $\{\gamma_i\}_{i=1}^n$, SRSF μ_q of the common template

1 $\gamma_i \leftarrow \gamma_{\text{id}}$ /* Initialize γ_i to identity warp */
2 $q_i \leftarrow \text{SRSF}(f_i)$ /* Calculate SRSF representation of f_i using (1) */
3 for e **from** 1 **to** E **do**
4 $\tilde{q}_i \leftarrow (q_i, \gamma_i)$ /* Warp in the SRSF space using (3) */
5 $(\tilde{q}_{i1}, \ldots, \tilde{q}_{iK}) \leftarrow \text{spl}^K \tilde{q}_i$ /* Split the SRSFs using (11) */
6 $\mu_{\tilde{q}} \leftarrow \frac{1}{nK} \sum_{i=1}^n \sum_{k=1}^K \tilde{q}_{ik}$ /* Calculate mean of $\{\tilde{q}_{ik}\}$ */
7 $\mu_{\tilde{q}}^K \leftarrow \text{ext}_{\text{L}^2}^K \mu_{\tilde{q}}$ /* Calculate extension of $\mu_{\tilde{q}}$ using (9) */
8   train $\text{NN}(x = \{q_i\}, y = \mu_{\tilde{q}}^K)$ for one epoch
9 $\tilde{\gamma}_i \leftarrow \text{NN}(x = q_i)$ /* Calculate neural network prediction */
10 $(\tilde{\gamma}_{i1}, \ldots, \tilde{\gamma}_{iK}) \leftarrow \text{spl}^K \tilde{\gamma}_i$ /* Split warping functions using (11) */
11 $\tilde{\gamma}_{ik}' \leftarrow \text{Scale}(\tilde{\gamma}_{ik})$ /* Scale domain and image of $\tilde{\gamma}_{ik}$ to $[0,1]$ */
12 $\mu_{\tilde{\gamma}} \leftarrow \text{KarcherMean}(\{\tilde{\gamma}_{ik}'\})$
         /* Find the Karcher mean of $\{\tilde{\gamma}_{ik}'\}$ as in Algorithm 1 */
13 $\gamma^* \leftarrow \text{ext}_{\Gamma}^K \mu_{\tilde{\gamma}}^{-1}$ /* Calculate extension of $\mu_{\tilde{\gamma}}^{-1}$ using (10) */
14 $\gamma' \leftarrow \text{Scale}(\gamma^*)$ /* Scale domain and image of γ^* to $[0,1]$ */
15 $\gamma_i \leftarrow \tilde{\gamma}_i \circ \gamma'$ /* Calculate warping functions using (6) */
16 end
17 $\gamma' \leftarrow \text{Scale}(\mu_{\tilde{\gamma}}^{-1})$ /* Scale domain and image of $\mu_{\tilde{\gamma}}^{-1}$ to [0,1/K] */
18 $\mu_q \leftarrow (\mu_{\tilde{q}}, \gamma')$ /* Calculate SRSF of common template using (5) */

NN$(x = \{q_i\}, y = \mu_{\tilde{q}}^K)$ and NN$(x = q_i)$ represent a neural network with input $\{q_i\}$ and q_i, respectively, and outcome $\mu_{\tilde{q}}^K$.
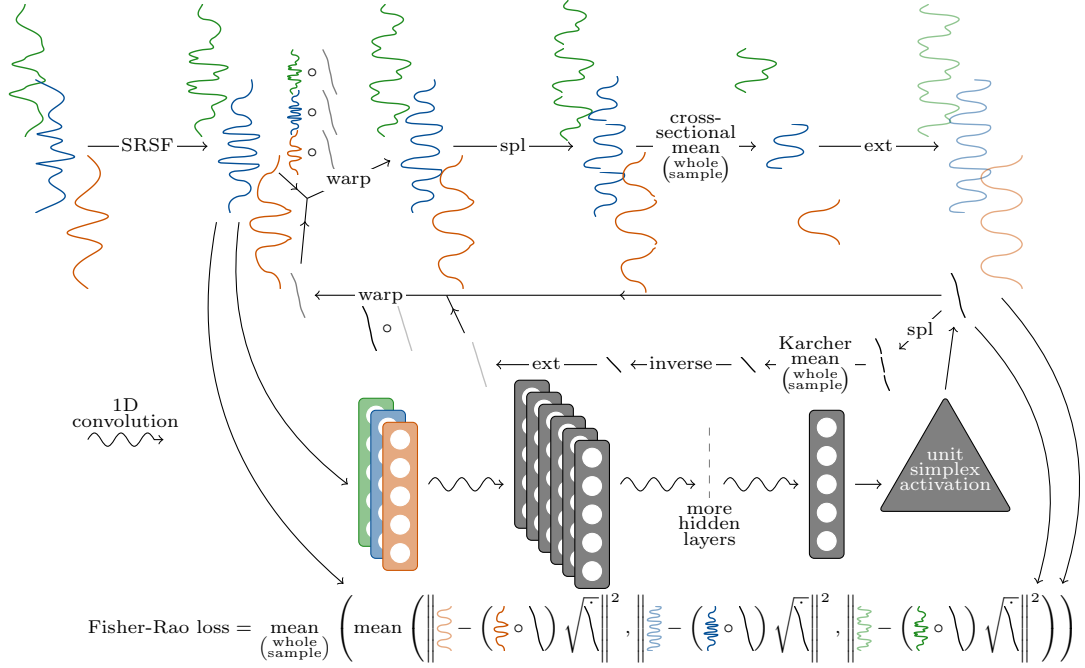
Figure 2: Diagram of the algorithm for joint alignment of multivariate quasi-periodic functions. Note that both the cross-sectional and Karcher mean are taken over the whole sample. Furthermore, the domain and image of the warping functions that are involved in the Karcher mean are first scaled to the interval $[0, 1]$.

which warp the split of q_i, $(q_{i1}, \ldots, q_{iK})$, to an element of an orbit $[\mu_{q_i}]$. We define the SRSF of the subject-specific template as the center of the orbit $[\mu_{q_i}]$ with respect to the set $\{q_{ik}\}_{k=1}^K$. We find this by calculating the Karcher mean μ_{γ_i} of $\{\gamma_{ik}\}_{k=1}^K$ by first scaling their domain and image to the interval $[0, 1]$, and then performing Algorithm 1. We can represent the warping function γ_i as

$$\gamma_i = \gamma_i \circ \gamma_{\mathrm{id}} = \gamma_i \circ \left(\mathrm{ext}_\Gamma^K \mu_{\gamma_i}\right)^{-1} \circ \mathrm{ext}_\Gamma^K \mu_{\gamma_i}, \tag{14}$$

where $\mathrm{ext}_\Gamma^K \mu_{\gamma_i}$ is a periodic extension of μ_{γ_i}, with the domain and image scaled to the interval $[0, 1]$. Furthermore, we have the following identities

$$\gamma_i = \left(\gamma_i^t\right)^{-1}, \quad \gamma_i \circ \left(\mathrm{ext}_\Gamma^K \mu_{\gamma_i}\right)^{-1} = \left(\gamma_i^g\right)^{-1}, \quad \mu_{\gamma_i} = \left(\gamma_i^l\right)^{-1}, \tag{15}$$

where γ_i^t, γ_i^g and γ_i^l are the total, global and local warping functions, respectively, as described in Equation (12).

The SRSF of the common template, μ_q, can be obtained from Algorithm 2, and the common template μ can be obtained from its SRSF using Equation (2) up to a constant $c \in \mathbb{R}$, $\mu(t) = c + \int_0^t \mu_q(s)|\mu_q(s)|\,ds$. The subject-specific template can be

extracted using identities in Equations (12) and (15) by

$$\mu_i = \mu \circ \mu_{\gamma_i}^l = \mu \circ \mu_{\gamma_i}^{-1}. \tag{16}$$

Quasi-periodic functions with amplitude variability. In addition to the multiscale time warping model, the amplitude of the subject-specific template can be further susceptible to amplitude variability, which means that the amplitudes in the different periods might not be the same. Furthermore, amplitudes measured on different subjects do not have to be the same either, thus there is no longer the same relationship between μ_j and μ_{ij} as described in Equation (13), and the subject-specific template cannot be obtained by Equation (16).

To obtain the SRSF of the subject-specific template in the presence of amplitude variability, we propose calculating the cross-sectional mean of the aligned functional data

$$\mu_{q_i}(t) = \frac{1}{K} \sum_{k=1}^{K} (q_{ik}, \gamma_{ik}')(t),$$

where $t \in [0,1]$, $\{(q_{i1}, \ldots, q_{iK})\}$ is the split of q_i using Equation (11), and $\{(\gamma_{i1}', \ldots, \gamma_{iK}')\}$ is the split of $\gamma_i \circ \left(\text{ext}_\Gamma^K \mu_{\gamma_i}\right)^{-1}$ using Equation (11). Finally, the subject-specific template can be obtained up to a constant from μ_{q_i} using Equation (2).

2.3   Deep learning architecture

The general architecture of a warping neural network can be quite flexible. The most important part is that it produces an output that has similar dimension of the discretized input functions and can be transformed into a warping function. In our case, this is achieved by using convolutional layers with a padding such that the layer output has the same length as the layer input. The outputs of the neural network are the warping functions, which do not need to have exactly the same dimension as the input layer. In this case, we can use interpolation to scale the output up or down. Furthermore, to save computation time and because of the specific loss function described below, instead of inputting the original functions, we use their SRSF representation as inputs directly. We implemented the neural network using Keras (Chollet et al., 2015) and TensorFlow (Abadi et al., 2015) in the R packages `keras` (Allaire & Chollet, 2022) and `tensorflow` (Allaire & Tang, 2022).

The output of the neural network uses a special activation function based on the

unit simplex transformation, so that the resulting warping functions are boundary-preserving increasing functions on the interval $[0,1]$. For a given neural network output $y \in \mathbb{R}^P$ before any activation is applied, we calculate new coordinates $z \in \mathbb{R}^P$ by

$$z_p = \text{expit}\left(y_p - \log(P - p + 1)\right), \text{ for } 1 \le p \le P, \tag{17}$$

where $\text{expit}(x) = 1/(1 + \exp(-x))$, ensuring that the new coordinates are all between 0 and 1. The vector z is then used to determine a vector $x \in \mathbb{R}^{P+1}$ by

$$x_p = \begin{cases} z_p, & p = 1, \\ \left(1 - \sum_{p'=1}^{p-1} x_{p'}\right) z_p, & 1 < p \le P, \\ 1 - \sum_{p'=1}^{P} x_{p'}, & p = P + 1, \end{cases}$$

with the property that all the coordinates of x are greater than 0, and that $\sum_{p=1}^{P+1} x_p = 1$. Looking back at Equation (17), the offset $-\log(P - p + 1)$ is added so that the zero vector y is mapped to the simplex $x = (1/(P+1), \ldots, 1/(P+1))$. Finally, we transform this vector x to a discretized warping function $\gamma \in \mathbb{R}^{P+2}$ by

$$\gamma_p = \begin{cases} 0, & p = 1, \\ \sum_{p'=1}^{p-1} x_{p'} & 1 < p \le P + 2. \end{cases}$$

In order to obtain a vector in $\mathbb{R}^P$, we propose linearly interpolating the values of γ, so that $\gamma \in \mathbb{R}^P$.

The loss function of our warping neural network is defined as follows. Let f_i, where $f_i(t) = (f_{i1}(t), \ldots, f_{iJ}(t)) \in \mathbb{R}^J$, be the observed multivariate functions and q_{ij} the SRSF representations of f_{ij}. Let μ_{q_j} be the SRSF representation of the template μ_j, $j = 1, \ldots, J$, and let γ_i be the outputs of the neural network. The Fisher-Rao loss function is then defined as

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{J} \sum_{j=1}^{J} \|\mu_{q_j} - (q_{ij}, \gamma_i)\|^2.$$

The univariate function case is easily obtainable by setting $J = 1$.

We used the Adam optimizer (Kingma & Ba, 2015). The architectural hyper-parameters such as the number of layers, number of filters, kernel size, and learning rate were chosen with the Bayesian optimization algorithm (Snoek et al., 2012) using the expected improvement acquisition function. Finally, the hyperbolic tangent

was used as an activation function of the hidden layers.

3 Simulation study

As a proof-of-concept of our neural network approach, we deploy two simulation studies: one for univariate quasi-periodic functions and one for multivariate quasi-periodic functions. In the case of univariate functions, we did not vary the amplitudes, hence the functions can be perfectly aligned, i.e., the distance between the true template and the aligned functions converges to zero with increasing sample size and complexity of the neural network. In the case of multivariate functions we varied both the inter-subject and intra-subject amplitudes, thus the peaks and valleys of the functions can be temporally aligned to the common template, but the vertical distance between them does not converge to zero.

In both cases, we generated $N = 14,000$ functions and used 8,000 of them as the training data, $2,000$ for hyperparameter tuning, $2,000$ as the validation data, and $2,000$ as the test data.

To evaluate the performance of our method, we use the decrease in the cumulative cross-sectional variance of the observed and aligned data. The cumulative cross-sectional variance is a measure of the average distance of the functions from the mean, defined as follows. Let $\{f_i(t), t \in [0,1]\}_{i=1}^n$ be a functional dataset, then the cumulative cross-sectional variance is

$$\widehat{\mathrm{Var}}\left(\{f_i\}\right) = \frac{1}{n-1} \int_0^1 \sum_{i=1}^n \left(f_i(t) - \mu(t)\right)^2 dt.$$

However, since we do not know the true common template μ for the real application data, we will use the cross-sectional mean in the calculation instead

$$\overline{\mathrm{Var}}\left(\{f_i\}\right) = \frac{1}{n-1} \int_0^1 \sum_{i=1}^n \left(f_i(t) - \frac{1}{n}\sum_{i=1}^n f_i(t)\right)^2 dt.$$

A decrease in the cumulative cross-sectional variance is desirable, because the cumulative cross-sectional variance of the observed data contains both the amplitude and phase variability, whereas the cumulative cross-sectional variance of the aligned data represents only the variability of the amplitude. Furthermore, we calculate the square of the L^2 norm of the distance between the cross-sectional mean and the true

Functions			Mean		
Observed	Aligned	% ↓	Observed	Aligned	% ↓
0.475	0.001	99.81	0.148	$1.89 \cdot 10^{-5}$	99.99

Table 1: Cumulative cross-sectional variance of the observed and aligned simulated univariate data, where the columns "% ↓" show the reduction of the cumulative cross-sectional variance in %. In addition, we calculate the square of the L^2 distance between the cross-sectional mean and the true template.

common template by

$$\left\| \frac{1}{n} \sum_{i=1}^{n} f_i - \mu \right\|^2 = \int_0^1 \left(\frac{1}{n} \sum_{i=1}^{n} f_i(t) - \mu(t) \right)^2 dt.$$

3.1 Scenario 1, univariate functions

In the first study, we simulated the common template to be the sine wave on the interval $[0, 2\pi]$, $\mu_f = \sin(t)$, $t \in [0, 2\pi]$, sampled at 65 equidistant points on each of the $K = 3$ periods, adding up to $P = 193$ equidistant points. Using the described notation, the periodic extension of the template is a sine wave on the interval $[0, 6\pi]$, $\text{ext}_{L^2}^K \mu = \sin(t)$, $t \in [0, 6\pi]$. The domain of the functions were then scaled to the interval $[0, 1]$. We simulated $N = 14,000$ global and $N = 14,000$ local warping functions denoted γ_i^g and γ_i^l using the fdasrvf package (Tucker, 2022), respectively. The warping functions using this package are generated in a way, so that their Karcher mean is the identity function. However, to ensure the identifiability of the subject-specific template, we need to transform the generated warping functions using Equation (14) and the identities in Equation (15). The observed functions are then generated as $f_i(t) = \left(\text{ext}_{L^2}^K \mu \circ (\gamma_i^l)^K \circ \gamma_i^g \right)(t)$.

The optimized architecture consists of an input layer with 193 nodes and one channel, followed by 17 convolutional layers with a kernel of size 64. The hidden layers have 25 filters, while the output layer has only one filter. The learning rate was chosen to be $3.66 \cdot 10^{-7}$.

Figure 3 shows the observed and aligned functions, as well as the corresponding estimated local and global warping functions. We can see that we achieved almost perfect alignment of both the training and test data. Furthermore, we can see that the cross-sectional mean of the observed data still preserves the modality in the data, but the size of the amplitudes is distorted. The cumulative cross-sectional variance of the simulated multivariate data can be found in Table 1.

(a1) Observed   (a2) Aligned   (b1) Observed   (b2) Aligned

(a3) Local warping   (a4) Global warping   (b3) Local warping   (b4) Global warping
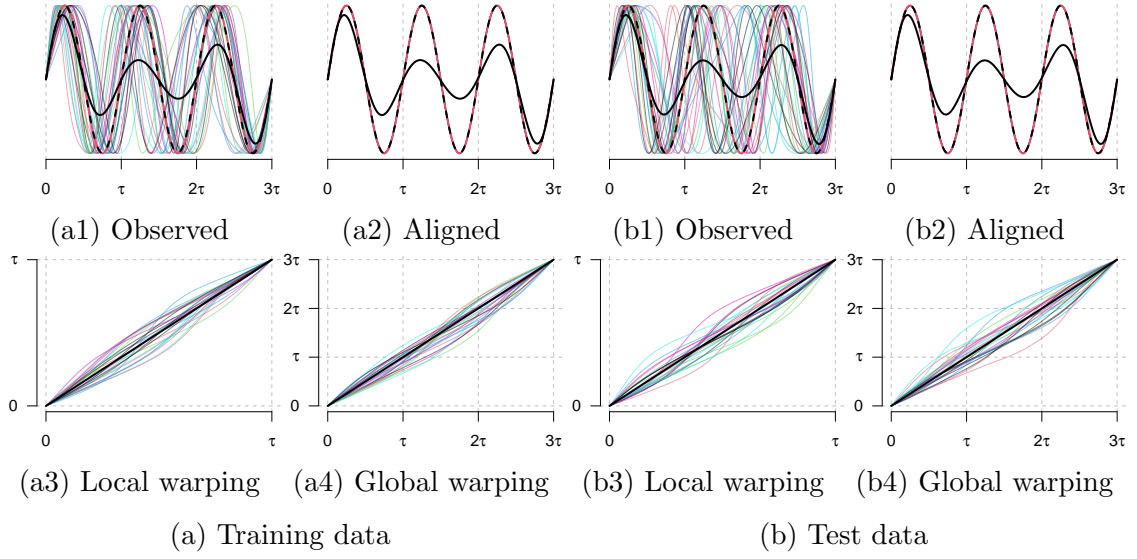
(a) Training data   (b) Test data

Figure 3: Result of the DeepJAM algorithm. The first row shows the following. Figures 3a1 and 3b1 show random 25 observed functions from the training and test data, respectively, while Figures 3a2 and 3b2 show the same functions, but aligned using the DeepJAM neural network, respectively for the training and test data. The thick red lines are the true common template extended to three periods, and the thick black full/dashed lines represent the cross-sectional mean of the full observed/aligned training and test data, separately. The second row shows the estimated local and global warping of these random 25 functions from the training and test data, where the thick black lines are the identity lines.

3.2 Scenario 2, multivariate functions

In this example, the functions to be aligned are f_i where the image of the functions at each t is $\mathbb{R}^3$. We write $f_i(t) = (f_{i1}(t), f_{i2}(t), f_{i3}(t))$, and we think of each function f_{ij}, as a univariate function that is the observed version of y_{ij} generated as follows. Let $z_{ip} \sim \mathcal{N}(1, 0.25^2)$, $p = 1, \ldots, 18$. The functions $y_{i1} : [0, 6\pi] \to \mathbb{R}$ were generated as

$$y_{i1}(t) = \begin{cases} z_{i1} \sin(t) & t \in [0, 2\pi), \\ z_{i2} \sin(t) & t \in [2\pi, 4\pi), \\ z_{i3} \sin(t) & t \in [4\pi, 6\pi]. \end{cases}$$

The common template is then $\mu_1(t) = \sin(t)$, $t \in [0, 2\pi]$. The functions y_{i2} were generated as

$$y_{i2}(t) = \begin{cases} z_{i4}(c_1(x) - c_3(x)) + c_3(x) + z_{i5}(c_2(x) - c_4(x)) + c_4(x), & t \in [0, 6), x = t - 3, \\ z_{i6}(c_1(x) - c_3(x)) + c_3(x) + z_{i7}(c_2(x) - c_4(x)) + c_4(x), & t \in [6, 12), x = t - 9, \\ z_{i8}(c_1(x) - c_3(x)) + c_3(x) + z_{i9}(c_2(x) - c_4(x)) + c_4(x), & t \in [12, 18], x = t - 15, \end{cases}$$

where

$$c_1(x) = e^{-(x-4.5)^2/2}, \qquad\qquad c_2(x) = e^{-(x-1.5)^2/2},$$

$$c_3(x) = \frac{e^{-\frac{4.5^2}{2}} - e^{-\frac{1.5^2}{2}}}{6} x + \frac{e^{-\frac{4.5^2}{2}} + e^{-\frac{1.5^2}{2}}}{2}, \qquad c_4(x) = \frac{e^{-\frac{1.5^2}{2}} - e^{-\frac{4.5^2}{2}}}{6} x + \frac{e^{-\frac{4.5^2}{2}} + e^{-\frac{1.5^2}{2}}}{2}.$$

This transformation is applied to ensure that $y_{i2}(0) = y_{i2}(6) = y_{i2}(12) = y_{i2}(18)$, allowing continuity at the limits of the periods. The common template is therefore $\mu_2(t) = e^{-(t-4.5)^2/2} + e^{-(t-1.5)^2/2}$, $t \in [-3, 3]$. Finally, functions y_{i3} were generated as

$$y_{i3}(t) = \begin{cases} z_{i10}p_2(x) - z_{i11}(p_1(x) - p_1(0)), & t \in [0, 0.5), x = t \\ z_{i10}p_2(x) + z_{i12}(p_3(x) - p_3(0)), & t \in [0.5, 1), x = t \\ z_{i13}p_2(x) - z_{i14}(p_1(x) - p_1(0)), & t \in [1, 1.5), x = t - 1 \\ z_{i13}p_2(x) + z_{i15}(p_3(x) - p_3(0)), & t \in [1.5, 2), x = t - 1 \\ z_{i16}p_2(x) - z_{i17}(p_1(x) - p_1(0)), & t \in [2, 2.5), x = t - 2 \\ z_{i16}p_2(x) + z_{i18}(p_3(x) - p_3(0)), & t \in [2.5, 3), x = t - 2, \end{cases}$$

Functions	Observed	Aligned	% ↓	Mean	Observed	Aligned	% ↓
1	0.506	0.038	92.54	1	0.148	0.001	99.03
2	0.571	0.144	74.76	2	0.194	0.003	98.23
3	0.420	0.033	92.06	3	0.126	0.002	98.75

Table 2: Cumulative cross-sectional variance of the observed and aligned simulated multivariate data, where columns "% ↓" show the reduction of the cumulative cross-sectional variance in %. In addition, we calculate the square of the L² distance between the cross-sectional means and the true templates.

where p_1 is the probability density function of $\mathcal{N}(0.25, 0.1^2)$, p_2 is the probability density function of $\mathcal{N}(0.5, 0.15^2)$ and p_3 is the probability density function of $\mathcal{N}(0.75, 0.1^2)$. The template function is then

$$\mu_3(t) = \begin{cases} p_2(t) - (p_1(t) - p_1(0)), & t \in [0, 0.5], \\ p_2(t) + (p_3(t) - p_3(0)), & t \in [0.5, 1], \end{cases}$$

In addition, functions y_{i2} and y_{i3} were further transformed with an affine transformation so that the image of the template functions μ_2 and μ_3 is in the interval $[-1, 1]$. Finally, the domains of all functions y_{ij}, as well as the extensions of the template functions, $\text{ext}_{L^2}^K \mu_j$, were scaled to the interval $[0, 1]$, and sampled at $P = 193$ equidistant points. The functions were then warped as $f_{ij}(t) = \left(y_{ij} \circ \text{ext}_\Gamma^K \gamma_i^l \circ \gamma_i^g \right)(t)$.

The optimized architecture consists of an input layer with 193 nodes and three channels, followed by 14 convolutional layers with a kernel of size 60. The hidden layers have 56 filters, while the output layer has only one filter. The learning rate was chosen to be $9.35 \cdot 10^{-6}$.

Figure 4 shows the observed and aligned test functions. We can see that we achieved nearly perfect phase alignment. The functions were simulated with amplitude variability, which is retained in the aligned data. Notice how the cross-sectional means of the observed data misrepresent the intrinsic shape of the underlying data generating model. Furthermore, the periodicity is also lost, especially for the cross-sectional mean of the functions $\{f_{i2}\}$. The cumulative cross-sectional variance of the simulated multivariate data can be found in Table 2.

(a) Observed f_{i1} (b) Observed f_{i2} (c) Observed f_{i3}

(d) Aligned $f_{i1} \circ \gamma_i$ (e) Aligned $f_{i2} \circ \gamma_i$ (f) Aligned $f_{i3} \circ \gamma_i$
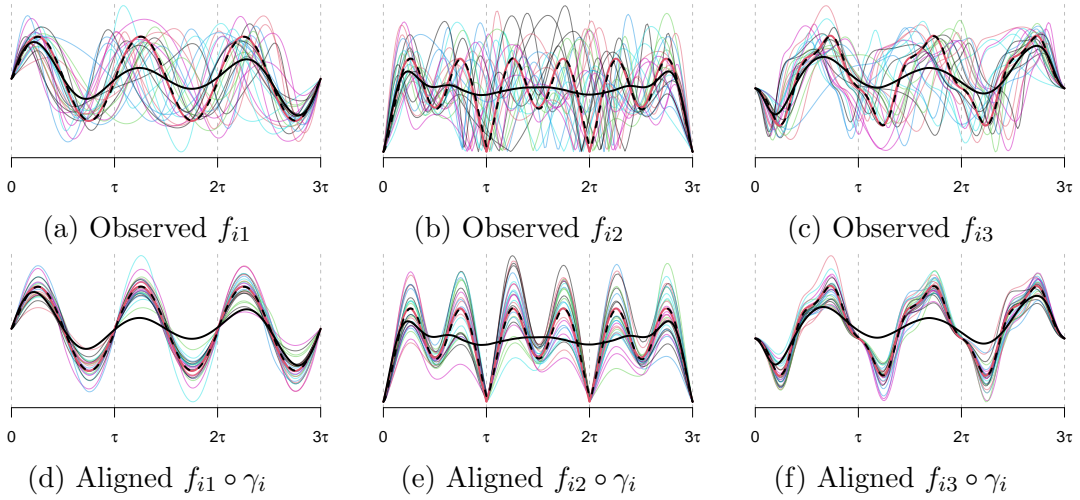
Figure 4: Result of the DeepJAM algorithm. Figures 4a, 4b and 4c show random 25 observed functions from the test data. The observed functions are the functions f_{ij}, $i = 1, \ldots, n$, $j = 1, 2, 3$, as described in Section 3.2. Figures 4d, 4e and 4f show the same functions, but aligned using the DeepJAM neural network. The thick red lines are the true common templates extended to three periods, and the thick black full/dashed lines represent the cross-sectional means of the full observed/aligned test data.

4   Application

For the application, we chose 12-lead 10s ECG recordings. The ECG study population consisted of participants in the Copenhagen General Population Study randomly sampled from the general population in Copenhagen, Denmark (). All ECGs were recorded with the study participant at rest and in supine position prior to a cardiac computed tomography scan. Written informed consent was obtained from all participants, and the study was approved by the local ethics committee (H-KF-01-144/01). The ECGs were preprocessed using the `neurokit2` python module (Makowski et al., 2021). First, the ECGs were filtered to remove noise and to improve the detection of peaks using the function `ecg_clean` with default settings for the method. Afterward, the function `ecg_peaks` was used to detect the location of R-peaks, and for the purpose of this application, we only selected a subset of the ECGs between the second and fifth R-peak, giving rise to a three-period time series, with each period between two subsequent R-peaks. The ECGs were then resampled to $P = 301$ equidistant points. Furthermore, because of the position of the electrodes, four leads can be derived from other leads (Horáček, 2010), thus we will only consider eight leads: I, II, V1–V6. We also only study a subset of the ECGs, resulting in 9,645 ECG recordings. From these, we used 5,511 for training,

(a) Observed lead V3   (b) Aligned lead V3   (c) Cross-sectional means

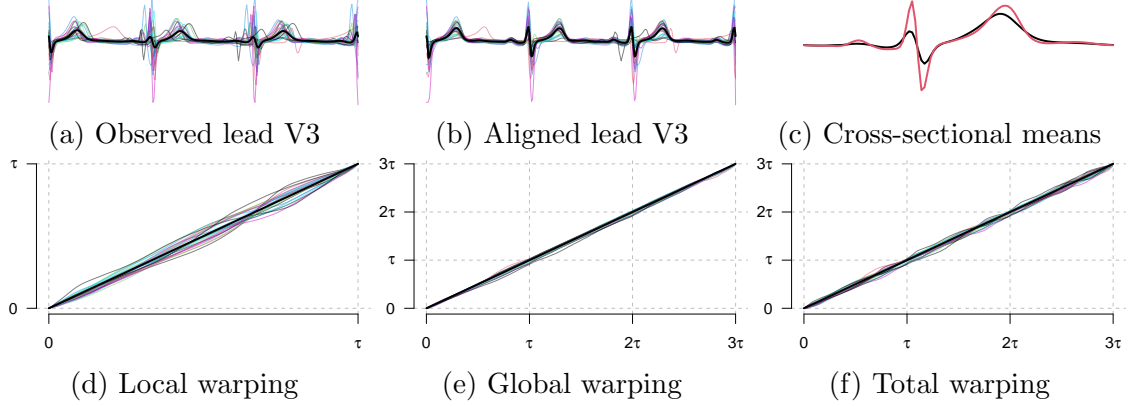(d) Local warping   (e) Global warping   (f) Total warping

Figure 5: Result of the DeepJAM algorithm. Figures 5a and 5b show the observed and aligned lead V3, respectively, of random 25 test ECGs, where the thick black lines represent the cross-sectional means of the full observed and aligned data, respectively. Figure 5c shows again the cross-sectional means of the full observed data (black) and the full aligned data (red), where we only present one period. Figures 5d, 5e and 5f show the local, global, and total warping of the 25 random test ECGs, with the thick black lines representing the identity line.

1,378 for hyperparameter tuning, 1,378 as the validation data and 1,378 as the test data.

The optimized architecture consists of an input layer with 301 nodes and eight channels, followed by 17 convolutional layers with a kernel of size 101. The hidden layers have 30 filters, while the output layer has only one filter. The learning rate was chosen to be $2.84 \cdot 10^{-5}$.

Figure 5 shows the observed and aligned ECG recordings, as well as the corresponding estimated local and global warping functions. In addition, we present a comparison of the cross-sectional means of the observed and aligned data. Table 3 shows the cumulative cross-sectional variance of the observed and aligned data per lead. We can see that alignment using the DeepJAM neural network reduced the cumulative cross-sectional variance by 27.46%–46.32%.

5   Conclusion

In this paper, we have presented a novel method for joint alignment of multivariate quasi-periodic functions, addressing the phase variability in the data that is often overlooked by traditional approaches. We equipped our convolutional neural networks with a unique activation function based on the unit simplex transformation, achieving effective phase-amplitude separation. To extend our approach to multivariate functions, we incorporated a convolutional layer with multiple chan-

Lead	Observed	Aligned	% ↓
I	0.005	0.003	32.87
II	0.017	0.010	41.08
V1	0.010	0.007	27.46
V2	0.024	0.017	29.50
V3	0.029	0.020	30.68
V4	0.026	0.015	41.23
V5	0.020	0.011	45.57
V6	0.014	0.007	46.32

Table 3: Cumulative cross-sectional variance of the observed and aligned ECG data, where column "% ↓" shows the reduction of the cumulative cross-sectional variance in %.

nels in the input layer. The output layer, on the other hand, provides a single warping function per subject, which simultaneously aligns all dimensions of the observed function. We also incorporated the multiscale warping model to accommodate quasi-periodic functions. To train our model, we used a loss function that calculates the Fisher-Rao distance between the square-root slope representations of functions. Notably, our approach is unsupervised and capable of generating common and subject-specific templates.

We conducted experiments on two simulated datasets as well as a real dataset of 12-lead 10s electrocardiogram recordings. We demonstrated the ability of our approach to accurately separate the phase variability from the amplitude variability, with nearly perfect phase alignment in the simulated data and substantial reduction in cumulative cross-sectional variance in the real dataset. We only used very simple convolutional neural networks as a proof of concept. Future work could focus on exploring potential enhancement to the neural network architecture, such as including dropout layers, batch normalization layers, skip layers, as well as different optimizers and activation functions of the hidden layers.

The contribution of our research extends beyond the specific dataset analyzed in this paper. The joint alignment of functions has extensive applications in various fields such as signal processing and computer vision, and medical applications such as neuroscience and data from wearable devices.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/

Allaire, J., & Chollet, F. (2022). *Keras: R interface to 'keras'* [R package version 2.10.0]. https://CRAN.R-project.org/package=keras

Allaire, J., & Tang, Y. (2022). *Tensorflow: R interface to 'tensorflow'* [R package version 2.10.0]. https://CRAN.R-project.org/package=tensorflow

Bhattacharya, A., & Bhattacharya, R. (2012). *Nonparametric inference on manifolds: With applications to shape spaces* (1st). Cambridge University Press. https://doi.org/10.1017/CBO9781139094764

Boucheham, B. (2008). Matching of quasi-periodic time series patterns by exchange of block-sorting signatures. *Pattern Recognition Letters*, *29*(4), 501–514. https://doi.org/10.1016/j.patrec.2007.11.004

Boulnemour, I., & Boucheham, B. (2018). QP-DTW: Upgrading dynamic time warping to handle quasi periodic time series alignment. *Journal of Information Processing Systems*, *14*(4), 851–876. https://doi.org/10.3745/JIPS.02.0090

Chen, C., & Srivastava, A. (2021). SrvfRegNet: Elastic function registration using deep neural networks. *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 4462–4471. https://doi.org/10.1109/CVPRW53098.2021.00503

Chollet, F., et al. (2015). Keras. https://keras.io

Fuchs, A., Kühl, J. T., Sigvardsen, P. E., Afzal, S., Knudsen, A. D., Møller, M. B., et al. (2023). Subclinical coronary atherosclerosis and risk for myocardial infarction in a danish cohort: A prospective observational cohort study. *Annals of Internal Medicine*, *176*(4), 433–442. https://doi.org/10.7326/M22-3027

Gregg, R. E., Zhou, S. H., Lindauer, J. M., Helfenbein, E. D., & Giuliano, K. K. (2008). What is inside the electrocardiograph? *Journal of Electrocardiology*, *41*(1), 8–14. https://doi.org/10.1016/j.jelectrocard.2007.08.059

Horáček, B. M. (2010). Lead theory. In *Comprehensive electrocardiology* (2nd). Springer.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*. https://doi.org/10.48550/ARXIV.1412.6980

Klonoff, D. C. (2005). Continuous glucose monitoring: Roadmap for 21st century diabetes therapy. *Diabetes Care*, *28*(5), 1231–1239. https://doi.org/10.2337/diacare.28.5.1231

Kneip, A., & Ramsay, J. O. (2008). Combining registration and fitting for functional models. *Journal of the American Statistical Association*, *103*(483), 1155–1165. https://doi.org/10.1198/016214508000000517

Kühl, J. T., Nielsen, J. B., Stisen, Z. R., Fuchs, A., Sigvardsen, P. E., Graff, C., et al. (2019). Left ventricular hypertrophy identified by cardiac computed tomography and ECG in hypertensive individuals: A population-based study. *Journal of Hypertension*, *37*(4), 739–746. https://doi.org/10.1097/HJH.0000000000001962

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lohit, S., Wang, Q., & Turaga, P. (2019). Temporal transformer networks: Joint learning of invariant and discriminative time warping. *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12418–12427. https://doi.org/10.1109/CVPR.2019.01271

Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel, C., & Chen, S. H. A. (2021). NeuroKit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, *53*(4), 1689–1696. https://doi.org/10.3758/s13428-020-01516-y

Marron, J. S., Ramsay, J. O., Sangalli, L. M., & Srivastava, A. (2015). Functional data analysis of amplitude and phase variation. *Statistical Science*, *30*(4), 468–484. https://doi.org/10.1214/15-STS524

McDonnell, E. I., Zipunnikov, V., Schrack, J. A., Goldsmith, J., & Wrobel, J. (2022). Registration of 24-hour accelerometric rest-activity profiles and its application to human chronotypes. *Biological Rhythm Research*, *53*(8), 1299–1319. https://doi.org/10.1080/09291016.2021.1929673

Müller, M. (2007). *Information retrieval for music and motion* (1st). Springer.

Nunez, E., & Joshi, S. H. (2020). Deep learning of warping functions for shape analysis. *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 3782–3790. https://doi.org/10.1109/CVPRW50498.2020.00441

Nunez, E., Lizarraga, A., & Joshi, S. H. (2021). SrvfNet: A generative network for unsupervised multiple diffeomorphic functional alignment. *Proceedings of the*

2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 4476–4484. https://doi.org/10.1109/CVPRW53098.2021.00505

Oh, J., Wang, J., & Wiens, J. (2018). Learning to exploit invariances in clinical time-series data using sequence transformer networks. *Proceedings of the 3rd Machine Learning for Healthcare Conference, 85*, 332–347.

Ramsay, J. O., & Silverman, B. W. (2005). *Functional data analysis* (2nd). Springer. https://doi.org/10.1007/b98888

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Proceedings of the Advances in Neural Information Processing Systems, 25*.

Srivastava, A., & Klassen, E. P. (2016). *Functional and shape data analysis* (1st). Springer New York. https://doi.org/10.1007/978-1-4939-4020-2

Sun, G.-Z., Chen, H.-H., & Lee, Y.-C. (1992). Time warping invariant neural networks. *Advances in Neural Information Processing Systems, 5*.

Tallec, C., & Ollivier, Y. (2018). Can recurrent neural networks warp time? *arXiv.* https://doi.org/10.48550/ARXIV.1804.11188

Tucker, J. D. (2022). *fdasrvf: Elastic Functional Data Analysis.* https://CRAN.R-project.org/package=fdasrvf

Tucker, J. D., Wu, W., & Srivastava, A. (2013). Generative models for functional data using phase and amplitude separation. *Computational Statistics & Data Analysis, 61*, 50–66. https://doi.org/10.1016/j.csda.2012.12.001

Weber, R. S., Eyal, M., Skafte, N., Shriki, O., & Freifeld, O. (2019). Diffeomorphic temporal alignment nets. *Advances in Neural Information Processing Systems, 32*.

A Observed and aligned ECG test data

Refer to Figure 6 to see the performance of the DeepJAM algorithm on 8 leads of the ECGs.

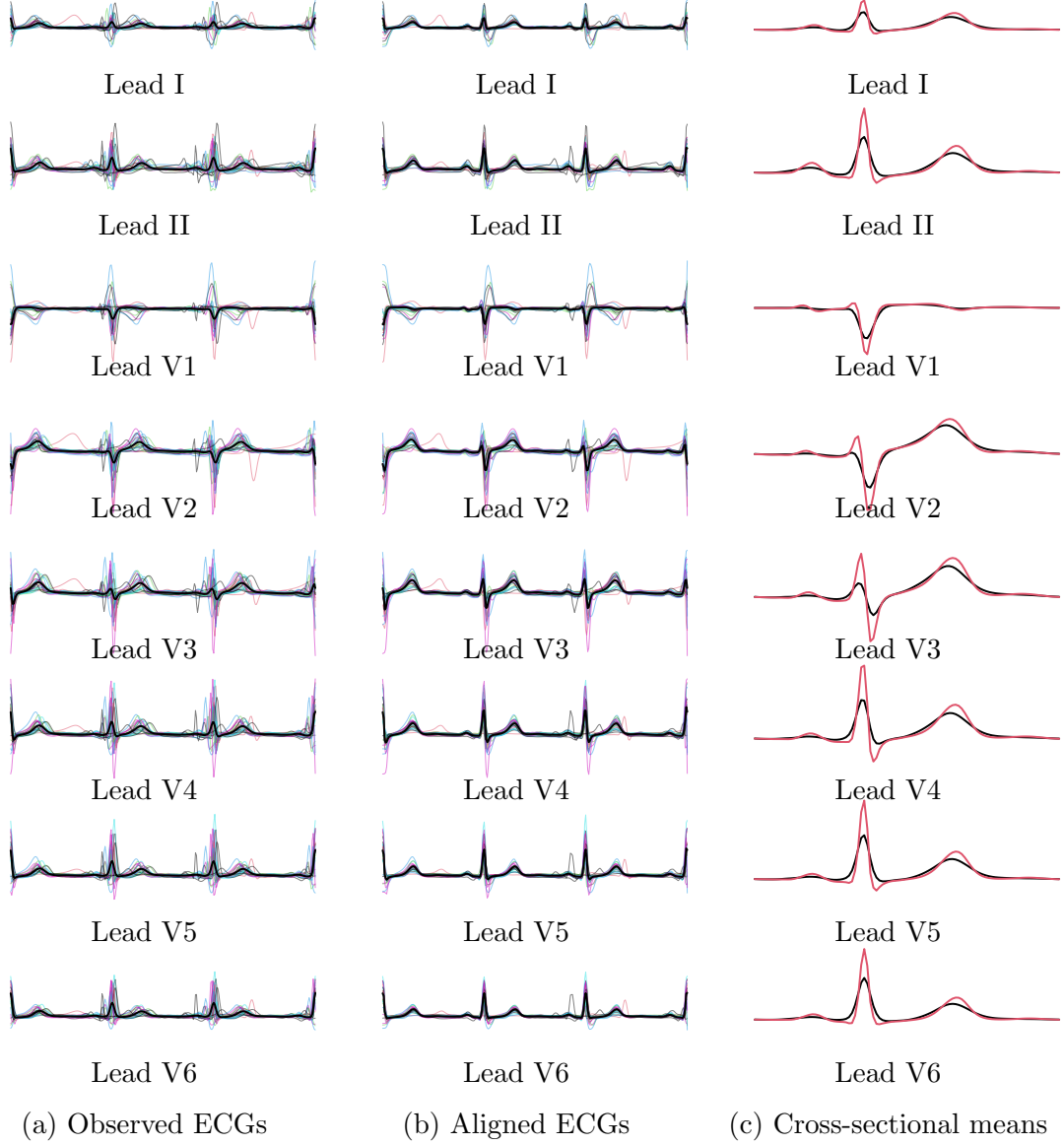|		
(a) Observed ECGs	(b) Aligned ECGs	(c) Cross-sectional means

Figure 6: Result of the DeepJAM algorithm. The first column shows individual leads of random 25 ECG measurements, while the second column shows the same ECGs, but aligned. Finally, the third column shows the cross-sectional means of the full observed dataset (black) and the full aligned dataset (red), where we only show one period.