ElasticRegNet: An Unsupervised Network for Joint Temporal Alignment

Chao Chen[1] and Anuj Srivastava[2]

Department of Statistics, Florida State University, USA
cc16w@fsu.edu

Abstract. The problem of aligning objects is essential in shape, and functional data analysis. The observations of functions, curves, or time-series data are generally misaligned. Direct processing of such data can lose underlying structure, artificially inflate with-in class variance, and decrease overall classification performance. This paper introduces an unsupervised, learning-based framework called Elastic Registration Network (*ElasticRegNet*), that registers curve data efficiently with high accuracy and that generalizes well beyond the training data. Furthermore, this architecture can be trained on simulated data, and with little retraining, using the so-called *transfer learning* can perform well on real test data. It uses multiple convolution layers to learn constrained diffeomorphism functions that help align given curves. The training is based on minimizing an objective function motivated by the elastic Riemannian metric and square-root velocity representation. We demonstrate the efficacy of this architecture using various public datasets and compare them to the current state-of-the-art approaches.

Keywords: Functional Registration · Temporal Alignment · Time-Series Data · Deep Learning · Riemannian Framework.

1 Introduction

The problem of registering objects is omnipresent in statistical analysis of shapes, functions, and any time series. Comparing shapes of curves or surfaces, or computing statistical summaries from shape data, requires pairwise dense registrations of points across objects. Registration is a well-known bottleneck in many fields such as computational anatomy, functional data analysis, activity recognition and shape classification. Data collected from images and videos often come under arbitrary registrations and coordinate systems. Using such unregistered data artificially inflates the data, often modifying the class structures. Furthermore, summarizing misaligned data loses some sharp features underlying given data. For instance, Figure 1 shows some simulated data where each function is essentially a time-warped unimodal function. A standard analysis of these functions loses the unimodal structure. As another example, in activity recognition, the same activity performed by different actors may not be temporally aligned, increasing distances between them under traditional metrics. In all these cases, the registration of geometric features across curves becomes necessary to perform statistical analysis.

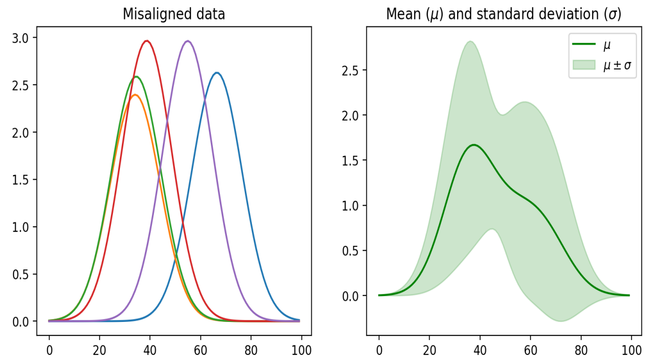

Fig. 1: The left is misaligned unimodal functions, and the right is the cross-sectional mean with one standard deviation band. Without proper alignment, the analysis (e.g., mean, standard deviation) cannot capture the underlying structure of unimodal functions.

In functional and curve data, registration corresponds to appropriate time warping of functions such that their peaks and valleys are tightly aligned. This is also called *phase-amplitude separation* [17, 27], where the aligned functions are called the *amplitudes* and the warping functions called the *phases*. The next question is: What should be an objective function for defining "optimal" alignment? Secondly, what is an efficient computational solution for solving that optimization problem for large data? The data can be large both in terms of the length of the functions and the number of functions. For the first question, a well-developed formulation utilizes an invariant Riemannian metric for deriving an objective function (stated below). The main open issue lies in the second question, deriving an optimization procedure that scales up well with the data size.

Historically, the most common way to align functional data is Dynamic Time Warping (DTW) [24] which utilizes the dynamic programming algorithm at its core. Given two scalar functions $f_1, f_2 : [0, T] \to \mathbb{R}$, this method seeks a warping function, $\gamma : [0, T] \to [0, T]$ that starts with objective function $\|f_1 - f_2 \circ \gamma\|^2$, where $\| \cdot \|$ denotes $\mathbb{L}^2$ norm of functions. However, this objective function is known to be degenerate because one can make it arbitrarily small using some extreme time warpings. This phenomenon is called the *pinching effect* [17, 27]. Past solutions against the pinching effect include adding regularization terms that penalize the roughness of the warping functions [1]. However, this solution has several shortcomings, including inverse inconsistency (the solutions are not consistent if f_1 and f_2 are interchanged) and the need for a hyperparameter tuning. Despite these shortcomings, DTW and its extensions of DTW [3,21,23] are popular.

A more fundamental solution comes from the elastic Riemannian framework [26, 27] which utilizes a different objective function, one that is based on the elastic Riemannian metric and its invariant properties. While this metric has nice theoretical properties, it originally has a form which is too complicated to be useful in the practice, especially on large datasets. This issue is resolved by using the Square-Root Velocity Function (SRVF) representations. For any function $f : [0, T] \to \mathbb{R}$, its SRVF is given by $q(t) = \text{sign}(\dot{f}(t))\sqrt{|\dot{f}(t)|}$ and the SRVF of a time-warped function $f \circ \gamma$ is given by $q \star \gamma \equiv (q \circ \gamma)\sqrt{\dot{\gamma}(t)}$. There are several advantages of using the SRVF representation. First, for any square-integrable q and any time warping γ, we have $\|q \star \gamma\| = \|q\|$. That is, the time-warping in the SRVF space is norm-preserving and prevents any pinching-like effect. Second, the elastic Riemannian distance between f_1 and f_2 equals the $\mathbb{L}^2$ norm between their SRVFs. Consequently, the registration of f_2 to f_1 is performed using their SRVFs q_1 and q_2, using: $\hat{\gamma} = \text{argmin}_\gamma \|q_1 - (q_2 \circ \gamma)\sqrt{\dot{\gamma}}\|^2$. This optimization is performed using the Dynamic Programming algorithm (DPA). If the functions f_1 and f_2 are sampled using T discrete points, then the computational cost of DTA is $O(kT^2)$, where $k << T$ is a constant. If we have to align multiple functions $f_1, f_2, \ldots, f_n$, then we iteratively compute their aligned means μ and align each given function to μ, all in SRVF space. That is, we solve for: $\hat{\gamma}_i = \text{argmin}_\gamma \|\mu - (q_i \circ \gamma)\sqrt{\dot{\gamma}}\|^2$ and use them to update the mean $\mu = \frac{1}{n}\sum_{i=1}^n((q_i \circ \hat{\gamma}_i)\sqrt{\dot{\hat{\gamma}}})$. Thus, each iteration of this optimization requires n calls to the DPA, leading to a total cost $O(nkT^2)$. When n and T become very large, this cost becomes prohibitive.

With recent advances in learning-based solutions, researchers have started exploring the use of deep neural networks for registering functions and curves. If successful, these networks can be trained to handle large data sets. Furthermore, these networks can be applied to future data from similar classes once trained. This application can be very fast without requiring cumbersome DPA. Proper network architectures can be made to generalize solutions to distributions beyond the training classes. One can use ideas such as transfer learning and mild retraining to broaden the applicability of trained networks.

There have been some recent papers on function alignment using deep learning. Jaderberg $et\ al.$ [10] presented a Spatial Transformer Network (STN) that learns invariant spatial warps from the training data and applies these warps to image data to enhance classification. Similarly, Lohit $et\ al.$ [16] introduced a Temporal Transformer Network (TTN) to learn time warping for optimizing classification performance. In these approaches, the ultimate goal is classification, and they obtain alignments as a side product. Some other papers [6, 19] have proposed a *supervised* deep learning approach for curve registration. In contrast, [2, 11, 20, 25] developed *unsupervised* learning-based registration networks that do not require a template in the training stage. Some of these architectures include regularization terms in their objective functions to control warping levels [20, 25]. As mentioned earlier, regularisation terms have limitations and require tuning additional hyperparameters. Thus, despite recent papers, an efficient, training-based network that aligns curve data and generalizes well to unseen data distributions remains elusive.

This paper introduces an Elastic Registration Network (ElasticRegNet) that implements an unsupervised, learning-based registration approach. It combines the strengths of the elastic Riemannian framework with the efficiency of deep neural networks to provide outcomes with excellent mathematical properties and generalizability. Notably, ElasticRegNet adapts neural network architectures that do not need any regularization term. It focuses on reducing the complexity of registration architecture without losing registration performance. Most of the earlier learning-based registration architectures [2,6,16,20] contained three parts. The first part extracts latent features from input data, the second transforms latent features to a different space and the third maps these transformed features into warping functions for registration. Notice that the last part contains non-linear mappings to ensure that warpings are monotonically increasing and satisfy boundary conditions ($\gamma(0) = 0$ and $\gamma(T) = T$). As a result, the trainable parameters are all in the first two parts and ideally should be distributed equally among the two parts. In [2]. the second part's parameter size is disproportionately larger than the first part as that part uses fully-connected layers. In this paper, we replace those layers with one-dimensional convolution (1D-CNN) layers and a global average pooling (GAP) layer. This significantly reduces the number of parameters, while leading to more flexible registration. Furthermore, as demonstrated later, this reduced architecture is particularly suited to transfer learning and can perform sharper registration.

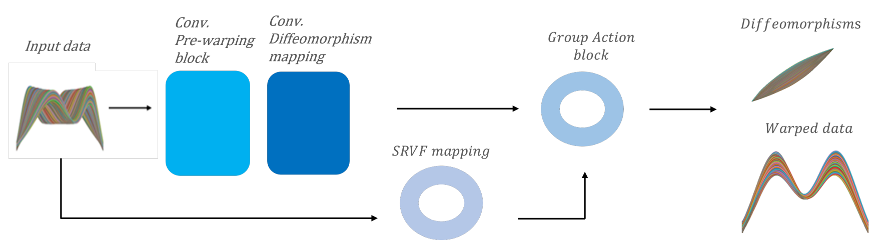

Fig. 2: ElasticRegNet's architecture: The input data goes through conv. pre-warping, diffeomorphism block and SRVF-mapping blocks at the same time. The SRVF mapping computes SRVFs of input data, and the Conv. Diffeomorphism block generates warping functions. These two are combined in the Group action block to output registered data.

2 Proposed Approach

In this section, we introduce the architecture of the ElasticRegNet. This network is composed of several blocks and we introduce them each next.

2.1 Conv. Learnable Pre-warping block

The first block takes input functional data (discretized into T-length vectors) and generates latent features for producing warping functions. This block is composed of three 1D-CNN layers [14] with 16-35-T filters per layer. Additionally, we maintain a consistent series-length by setting the padding size of each 1D-CNN layer to $\frac{k-1}{2}$, where k is the filter size. Each 1D-CNN layer is followed by a rectified linear activation function (ReLU) [18] and one-dimensional batch normalization (1D-BatchNorm) layer [9].

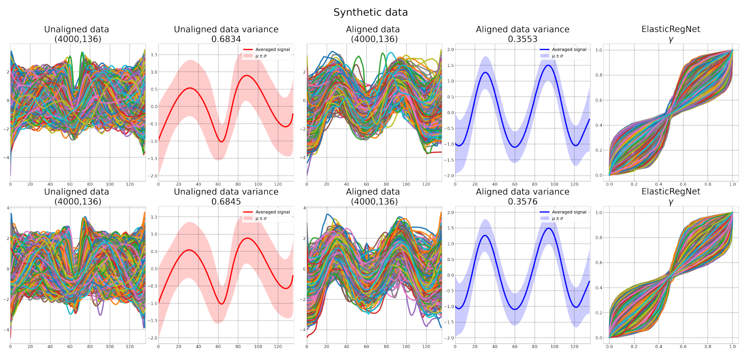

Fig. 3: The top row shows the training data, and the bottom row shows the test data. Each row contains unaligned data, aligned data, the corresponding means with one standard deviation band, and the predicted warping functions.

At the end of the block, there is a GAP layer [15] that is introduced to smoothing and shrinking of latent features using the formula: for $i = 1, \ldots, n, j = 1, \ldots, C, k = 1, \ldots, T$,

$$h^*_{i,j} = \frac{1}{T} \sum_{k=1}^{T} h_{i,j,k} \tag{1}$$

and where the $h_{i,j,k}$ is the latent feature produced by 1D-CNN blocks, n is the batch size, and C is the number of filters.

The Conv. Learnable pre-warping block contributes in extracting relevant features from input data. The 1D-CNN layers learn temporal representation, and the ReLU applies non-linear transformation on latent features; the 1D-BatchNorm layer improves stability of optimization process. Finally, GAP layer plays a role in smoothing latent features, resulting in flexible warping functions later. The input dimensions to this block is $n \times T$, and the output has size $(n \times T)$.

2.2 Diffeomorphism Mapping

The next block – diffeomorphism mapping – transforms the latent features into warping functions for each input. This transformation ensures that warping functions to satisfy properties of positive diffeomorphism – each γ is bijective, smooth, inverse smooth, and $\gamma(0) = 0$, $\gamma(T) = T$. There are no trainable parameters in this stage as it is simply a deterministic mapping according to:

$$\gamma_i(t) = T \frac{\sum_{s=0}^{t} \exp(h^*_i(s))}{\sum_{s=0}^{T} \exp(h^*_i(s))},\ t = 1, \ldots, T. \tag{2}$$

These warping functions $\{\gamma_i\}$ are then applied to the input functional data according to $f_i \mapsto f_i \circ \gamma_i$ (next block) nd evaluated for the registration performance. The output dimension of this mapping is $(n \times T)$.

2.3 SRVF mapping and Objective Function

This mapping computes the SRVF of each input functional data using the formula $q_i(t) = sign(\dot{f}_i(t))\sqrt{|\dot{f}_i(t)|}$. The time warping of the SRVF representations is given by the mapping:

$$Q_i(\gamma_i) = (q_i \star \gamma_i) = (q_i \circ \gamma_i)\sqrt{\dot{\gamma}_i}, i = 1, 2, \ldots, n. \tag{3}$$

The objective function for training the network is given by:

$$E(\gamma_1, \ldots, \gamma_n) = \sum_{i=1}^{N} \|Q_i(\gamma_i) - \bar{Q}\|^2. \tag{4}$$

where $\bar{Q} = \frac{1}{n}\sum_{i=1}^{N} Q_i(\gamma_i)$. Notice that this is a fully unsupervised setting. We train this network using the Adam optimization algorithm [13]. The learning rate is 0.001 and number of epochs is 100. All models are implemented using Pytorch [22] on an i7-10700K CPU@3.80GHz machine equipped with RTX 3060 GPU.

3 Experimental Results

We demonstrate the strengths of the proposed network using both simulated and real data.

3.1 Alignment of A Simulated dataset

We start by simulating data according to

$$\begin{aligned}
&f_i(t) = \alpha_i^2 g(\gamma(t)) + \epsilon_i(t),\\
&where\ g(t) = 2\sin(2\pi t)^2 + 0.05\cos(2\pi t + 1.2),\\
&\gamma_i(t) = \frac{1}{1 + e^{mt}},\\
&m \sim N(0, 5^2), \epsilon_i(t) \sim N(0, (1.2)^2),\\
&\alpha_i \sim N(1, (0.2)^2).
\end{aligned} \tag{5}$$

Figure 3 presents results from registration of this data using the ElasticRegNet. The sample size and data length of simulated training and test datasets are 4000 and 136. The first row shows the training data and second row shows the test data. In each row, we display: (1) the original functions, (2) their cross-sectional means, along with one standard-deviation bands, (3) the aligned functions, (4) the cross-sectional means of the aligned functions with one-standard-deviation bands, and (5) warping functions produced by the ElasticRegNet. We need a way to quantify alignment performance and we use $\frac{1}{N}\sum_{i=1}^{N} \|(f_i \circ \gamma_i) - \frac{1}{N}\sum_{j=1}^{N}(f_j \circ \gamma_j)\|^2$ for this purpose. In figure 3, this quantity for test data drops from 0.6845 to 0.3576 through alignment, and it reveals underlying structure of the functional data. It takes around 0.17 second to register the test data of the size (4000,136). Although the size of this synthetic data is modest, the traditional DTW method- Dynamic Programming Algorithm (DPA) of Duncan $et\ al.$ [5] takes around 2.77 hour to register the training data and needs to perform another run to register the test data.

As mentioned earlier, the DTW-based methods have to revisit the algorithm whenever new data is added. They can not perform registration as training and test scenarios, they take longer to execute registration. In contrast, ElasticRegNet is trained with training data and can be applied to test data without additional training procedures. The time used to train and perform registration is much shorter than the DPA.

3.2 Alignment of Real datasets

An important advantage of the ElasticRegNet architecture lies in the moderate model parameters size. As mentioned earlier, this architecture results in two nice properties: (1) better registration using Transfer Learning (2) higher registration performance.

Transfer learning is a methodology that trains a network with one data domain and then applies this pre-trained model, with some retraining, to another unseen data. The underlying idea is that higher (or earlier) layers tend to learn more generic features, and later layers learn specific task-related features. [2] has demonstrated the power of transfer learning with registration by retraining a new fully-connected layer to form time warping with a small data set. Although it is an alternative for a limited sample size, the number of parameters in a single fully-connected layer that needs to be retrained is still significant. In their work to train a registration with data, the number of time points (T) is 1024, over 95 per cent of parameters come from a single fully-connected layer and will be retrained by small samples, while only less than 4 per cent of parameters used for extracting features are from 1D-CNN layers. To improve this issue, we should manage the parameters' size used for retraining. Our approach utilizes a 1D-CNN layer with a GAP layer for generating warping functions. Notice that a 1D-CNN layer usually contains fewer parameters since it only connects to nearby neurons from the preceding layer within an adjustable size of filters. In contrast, a fully connected layer attaches to every neuron in the preceding layer, causing more training parameters. On top of that, the GAP layer is a transformation without parameters. Hence, the number of retrained parameters reduces significantly, improving performance using transfer learning.

Next, we use the UCR Time Series archive data [4] to evaluate the transfer learning with registration using the ElasticRegNet. We select seven datasets, including ECG200, FaceALL, FiftyWords, GunpointOldVersusYoung, StarlightCurves, and Yoga. We trained our model in the following ways and compared their results.

- **Train from scratch(NonTL)**: We train the ElasticRegNet with training data and perform registration on test data.
- **Registration with transfer learning-Retrain the last layer(TL)**: We train a ElasticRegNet with synthetic data (5) to acquire a pre-trained model. Then, we freeze all 1D-CNN layers, but the last one, and retrain it with training data.
- **Registration with Transfer learning- retrain all layers(TL_All)**: We unfreeze all 1D-CNN layers of the pre-trained model and retrain the model with training

data. This method combines the above procedures(Non_TL and TL) and sometimes gives better performance since TL_all provides different weights initialization, which could benefit the optimization process.

We present visualized registration results using transfer learning approaches from different real-world data. Figure 4, and figure 5 show unaligned and aligned functions performed by models trained with TL, and TL_all approaches, respectively. The datasets we use here are StarLightCurves and ECG200 as their training data size are (152,1024), and (69,96). We can see that both transfer learning approaches(TL, and TL_All) reduce variances significantly compared to the original data. We set epochs and learning rates to be 200 and 0.001. The training times of models are 6min23s, and 68s, and test time, an amount of time to apply registration to the unseen data, are all less than 1 second.

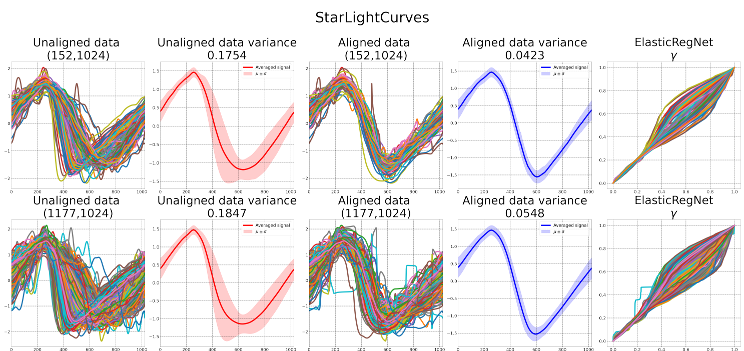

Fig. 4: **(TL)** We adopt the transfer learning method, which retrains the last 1D-CNN layer of the pre-trained ElasticRegNet with training data(top row). The sample size of the training data(152) is smaller than the test data(1177), but the model still produces a clear registration result. Variance of aligned data reduces significantly from 0.1847 to 0.054 in the second row. (152,1140) is (sample size, data length). The first row is training data, and the second row is test data. Each row contains unaligned, and aligned data with the corresponding mean, one-band standard deviations, and predicted warping functions.

To further investigate results of registration between train from scratch (NonTL) and transfer learning(TL, TL_all) approaches, we tested seven real-world time-series datasets, including ECG200, FaceAll, FiftyWords, GunpointOldVersusYoung, MedicalImages, StarLightCurves. For each dataset, we train the ElasticRegNet in NonTL, TL, and TL_all approaches and compare their registration performance in terms of variances. The learning rate and epochs are set to be 0.001 and 100, with the Adam optimization algorithm. Figure 6 compares registration performance using different training approaches. The green bar represents loss, orange, purple, and gray bars represent loss with NonTL, TL, and TL_all training methods, respectively. The loss here is vari-

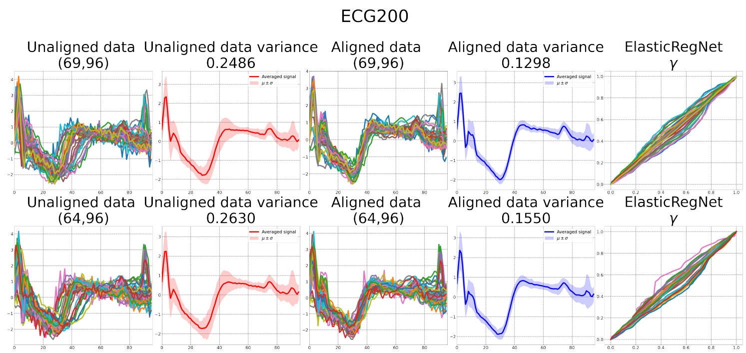

Fig. 5: **(TL_All)** Another transfer learning method, which retrains all 1D-CNN layers of the pre-trained ElasticRegNet. The values of (69,96) correspond to (sample size, data length). The first row is training data, and the second row is test data. We can see that the standard deviation band of the test data shrinks and variances reduce after alignments.

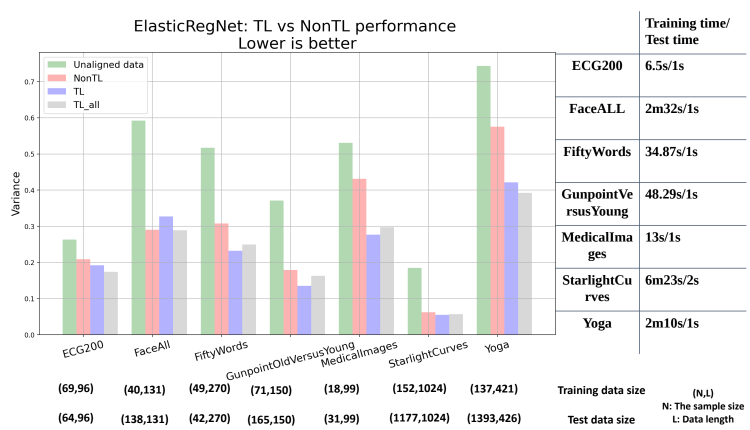

Fig. 6: Each data is registered using NonTL, TL, and TL_all approaches, denoted by red, purple, and gray bars, respectively. In terms of qualitative results, transfer learning methods (purple and gray) could be better since they have similar or lower variances than the NonTL method (red). The training and test time are both very short in all data sizes. The test time is the length of time to register unseen data. We only include the training and test time of NonTL methods since transfer learning approaches usually run even faster.

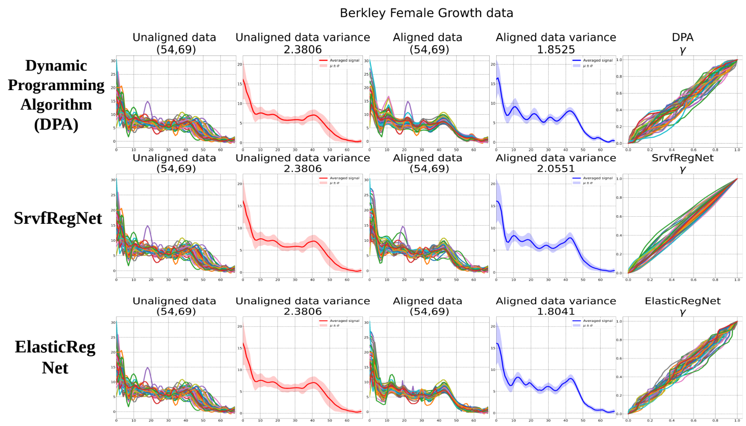

Fig. 7: The first, second, and third rows correspond to Dynamic Programming algorithm(DPA), SrvfRegNet, and ElasticRegNet. The first column is unaligned data from left to right, and the third is aligned data using the corresponding registration model. The second and the fourth columns are variances of unaligned and aligned data. The last column presents warping functions generated by registration models—the warping functions provided by the ElasticRegNet are way more flexible than SrvfRegNet. Moreover, ElasticRegNet gives the lowest variance of registered data among the three models.

ance. The figure shows that transfer learning based approaches generally outperform the model using the train from the scratch approach in all data sets. All methods reduce variance noticeably. These results suggest the ElasticRegNet architecture is suited more to transfer learning than training from scratch in various sample sizes, which can reduce the amount of training time and obtain similar or better performance.

3.3 Flexible and Fast Registration

In order to obtain sharper alignment, it is essential to impose smoothness on warping functions to avoid diffeomorphisms with large slopes. A common way is to include a regularization term to the objective function [19, 20, 25], but this solution loses symmetry property and requires additional effort to tune hyperparameters. [2] makes use of integration as smoothing operation. However, this way harms flexibility of warping functions a bit. Here, we use a global average layer to shrink latent features' dimension, which implicitly introduces smoothness to warping functions without suffering the flaws mentioned above.

We compare our work with DPA, SrvfRegNet by registering Berkely Female growth rate data [8] with a sample size is 54 and series length is 69. The growth rates follow the same trend with slight variation due to individual differences of each research object. Hence, it requires a model to produce flexible warping functions to achieve outstanding registration. Figure 7 shows that ElasticRegNet is a better choice. In terms of quantitative evaluation, the ElasticRegNet provides the lowest variance of aligned data (ElasticRegNet: 1.8041) compared with the other two methods(DPA: 1.8525, SrvfRegNet: 2.0551). Regarding the qualitative result, both DPA and ElasticRegNet register data sharply with flexible warping functions. In contrast, warping functions generated by SrvfRegNet are too smooth to conduct flexible registration.

Lastly, we would like to investigate larger datasets (sample size $>$ 1000) registration using DPA and ElasticRegNet approaches, which are proven to achieve finer registration. We choose Yoga and StarLightCurves datasets. Each dataset has its training and test part. The training part of StarLightCurves and Yoga are (152,1024) and (71, 150), respectively, while test sizes are (1177, 1024) and (1393, 426) respectively. We trained the ElasticRegNet with the training datasets and applied the resulting networks to the corresponding test sets. In contrast, we can only use the DPA to the test data. As the table 1 shows, ElasticRegNet can provide aligned variance comparable to DPA solution but with much faster execution time. Although DPA achieves a slightly lower variance of aligned data, the training and test time of executing registration using ElasticRegNet runs 48 (6min v.s. 4.8hr), 37 times faster (2min v.s. 1.24hr) than DPA method on Yoga and StarLightCurves data, respectively. Both methods reduce variance significantly when compared to the original data.

		DPA		ElastciRegNet	
Data	Var. (unaligned data)	Var. (aligned data)	Time (Test)	Var. (aligned data)	Time (Train/Test)
Yoga	0.7436	0.2182	4.8hr	0.3924	6min2s/2s (**x48 times faster**)
StarLightCurves	0.1847	0.0285	1.24hr	0.0548	2min10s/1s (**x37 times faster**)

Table 1: ElasticRegNet and the DPA methods reduce variance significantly after alignment. While the DPA method may achieve a slightly lower variance, it takes hours to complete the task. In contrast, ElasticRegNet performs registration in minutes, at least 37 times faster, and provides comparable registration performance.

3.4 Comparison with State-of-Art

We compare our approach to another unsupervised, deep learning-based registration network, Diffeomorphic Temporal Alignment Net (DTAN) [25]. We can implement DTAN in two ways, DTAN with a smoothness prior and DTAN without a smooth prior. Regarding DTAN with a smoothness prior, there are two additional hyper-parameters λ_{smooth} and λ_{var}, and we select them to be 0.01 and 0.05 respectively by default. All three approaches, ElasticRegNet, DTAN with and without smoothness, are trained on training data and apply them to test data foe evaluations. The training epochs is 100 and learning rate is 0.01 with Adam optimizer. We provide registration results in both quantitative and qualitative way on StarLightCurves, ECG200, and FaceAll test

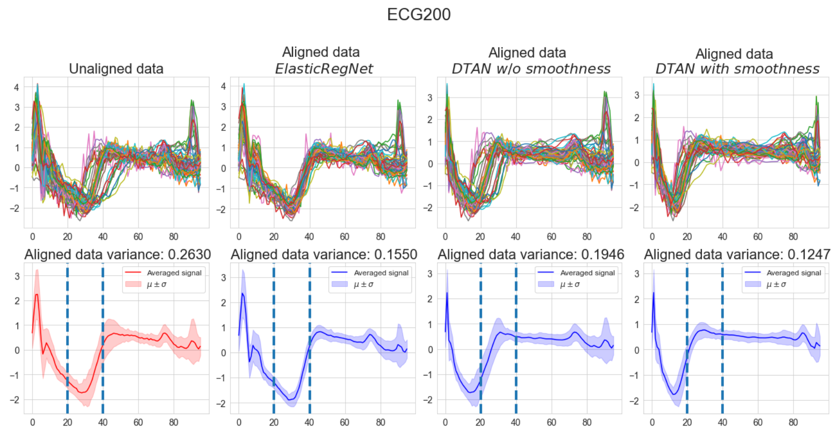

Fig. 8: The top row shows unaligned and aligned data under different approaches. The second row displays corresponding means with one standard-deviation band. From left to right: unaligned data, aligned data registered by ElasticRegNet, DTAN w/o smoothness, and DTAN with smoothness. We can observe that shapes and valleys' position along the x-axis of aligned data registered by ElasticRegNet are well-maintained (in the dashed line interval). On the contrary, those registered by DTAN with and w/o smoothness are dragged leftward (outside the dashed line interval).

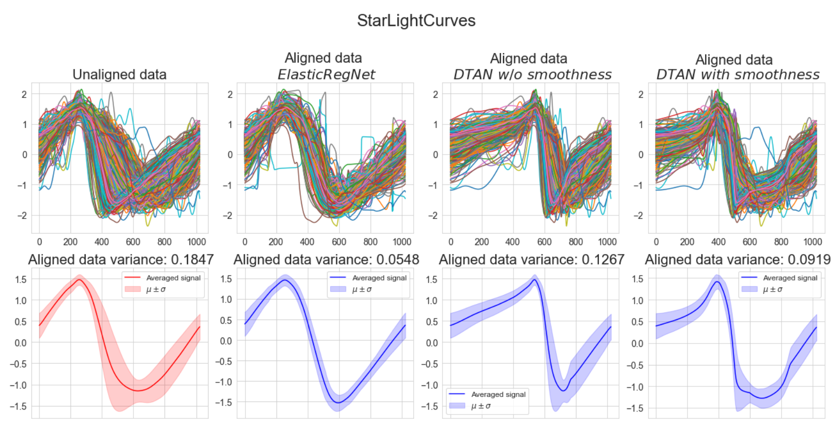

Fig. 9: Shapes of aligned data using DTAN with and w/o smoothness are squeezed and distorted. On the contrary, shapes of aligned data using ElasticRegNet are well-preserved.

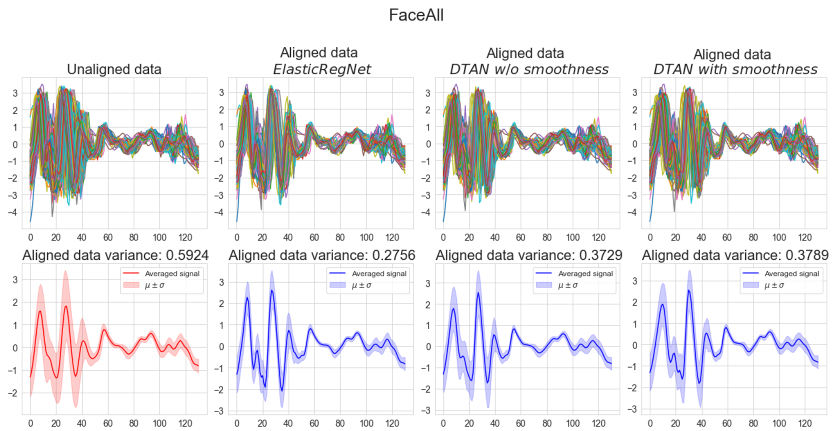

Fig. 10: The Variance of aligned data warped by ElasticRegNet falls more than that warped by DTAN methods. All methods preserve shapes of aligned data well.

datasets. Note that there are more registration models using neural networks proposed [7, 12, 20], but their codes are not open to public so that we cannot include them for comparisons.

Figure 8, 9, and 10 display unaligned and aligned data registered by $ElasticRegNet$, $DTAN\ w/o\ smoothness$, and $DTAN\ with\ smoothness$. In terms of quantitative evaluation, ElasticRegNet performs better on FaceAll, StarLightCurves while DTAN with smoothness did better on ECG200. All three methods reduce variance significantly compared to the original data. With regard to qualitative evaluation, ElasticRegNet is superior to other two methods because it can better preserve the shapes of data after alignment. That is, the shapes of aligned data are not disturbed compared to original data. We can observe this through the bottom row in the figure 8. The shapes and valley of aligned data, registered by ElasticRegNet, remains unchanged compared with original data (1^{st} and 2^{nd} plots from left at bottom row). In contrast, valleys of aligned data registered by DTAN with and w/o smoothness slightly pull to the left compared with original data (3^{rd} and 4^{th} plots from left at bottom row). The shapes of input data are changed by registration approaches. In the figure 9, shapes of aligned data registered by ElasticRegNet (2^{nd} plot from left at bottom row) is well-maintained compared with original data. However, shapes of aligned data (3^{rd} and 4^{th} plots from left at bottom row) are distorted by DTAN with and w/o smoothness algorithms. In Figure 10, although three models maintain shapes of aligned data, ElasticRegNet reduces variance more than DTAN methods. Therefore, ElasticRegNent can be an appropriate registration approach in terms of qualitative and quantitative evaluations.

4 summary

We propose ElasticRegNet, a deep learning model combined with the strength of elastic Riemannian framework for alignment. This light architecture can perform sharper and fast registration in both small and large datasets and generalize well to unseen datasets. Additionally, our architecture prefers more transfer learning with registration than training the model from scratch. This can be helpful in both reducing training time and obtaining nicer registration when the sample size is limited.

References

1. Bertsekas, D.: Dynamic programming and optimal control: Volume I, vol. 1. Athena scientific (2012)
2. Chen, C., Srivastava, A.: Srvfregnet: Elastic function registration using deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4462–4471 (2021)
3. Cuturi, M., Blondel, M.: Soft-dtw: a differentiable loss function for time-series. In: International Conference on Machine Learning. pp. 894–903. PMLR (2017)
4. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica **6**(6), 1293–1305 (2019)
5. Duncan, A., Srivastava, A., Descombes, X., Klassen, E.: Geometric analysis of axonal tree structures. DIFF-CV: Differential Geometric Techniques in Computer Vision (2015)
6. Hartman, E., Sukurdeep, Y., Charon, N., Klassen, E., Bauer, M.: Supervised deep learning of elastic srv distances on the shape space of curves. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4425–4433 (2021)
7. Huang, H., Amor, B.B., Lin, X., Zhu, F., Fang, Y.: Residual networks as flows of velocity fields for diffeomorphic time series alignment. arXiv preprint arXiv:2106.11911 (2021)
8. Hunt, J.V., Eichorn, D.H.: Maternal and child behaviors: A review of data from the berkeley growth study. In: Seminars in Psychiatry. Grune & Stratton, Inc. (1972)
9. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
10. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28**, 2017–2025 (2015)
11. Kaufman, I., Weber, R.S., Freifeld, O.: Cyclic diffeomorphic transformer nets for contour alignment. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 349–353. IEEE (2021)
12. Kawano, K., Kutsuna, T., Koide, S.: Neural time warping for multiple sequence alignment. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3837–3841. IEEE (2020)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kiranyaz, S., Ince, T., Hamila, R., Gabbouj, M.: Convolutional neural networks for patient-specific ecg classification. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 2608–2611. IEEE (2015)
15. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)

16. Lohit, S., Wang, Q., Turaga, P.: Temporal transformer networks: Joint learning of invariant and discriminative time warping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
17. Marron, J.S., Ramsay, J.O., Sangalli, L.M., Srivastava, A.: Functional data analysis of amplitude and phase variation. Statistical Science pp. 468–484 (2015)
18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Icml (2010)
19. Nunez, E., Joshi, S.H.: Deep learning of warping functions for shape analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 866–867 (2020)
20. Nunez, E., Lizarraga, A., Joshi, S.H.: Srvfnet: A generative network for unsupervised multiple diffeomorphic functional alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4481–4489 (2021)
21. Palazón-González, V., Marzal, A.: On the dynamic time warping of cyclic sequences for shape retrieval. Image and Vision Computing **30**(12), 978–990 (2012)
22. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
23. Petitjean, F., Ketterlin, A., Gançarski, P.: A global averaging method for dynamic time warping, with applications to clustering. Pattern recognition **44**(3), 678–693 (2011)
24. Sakoe, H.: Dynamic-programming approach to continuous speech recognition. In: 1971 Proc. the International Congress of Acoustics, Budapest (1971)
25. Shapira Weber, R.A., Eyal, M., Skafte, N., Shriki, O., Freifeld, O.: Diffeomorphic temporal alignment nets. Advances in Neural Information Processing Systems **32**, 6574–6585 (2019)
26. Srivastava, A., Klassen, E., Joshi, S.H., Jermyn, I.H.: Shape analysis of elastic curves in euclidean spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(7), 1415–1428 (2010)
27. Srivastava, A., Klassen, E.P.: Functional and shape data analysis, vol. 1. Springer (2016)