Regularization-free Diffeomorphic Temporal Alignment Nets

Ron Shapira Weber [1] **Oren Freifeld** [1]

Abstract

In time-series analysis, nonlinear temporal misalignment is a major problem that forestalls even simple averaging. An effective learning-based solution for this problem is the Diffeomorphic Temporal Alignment Net (DTAN) (Shapira Weber et al., 2019), that, by relying on a diffeomorphic temporal transformer net and the amortization of the joint-alignment task, eliminates drawbacks of traditional alignment methods. Unfortunately, existing DTAN formulations crucially depend on a regularization term whose optimal hyperparameters are dataset-specific and usually searched via a large number of experiments. Here we propose a regularization-free DTAN that obviates the need to perform such an expensive, and often impractical, search. Concretely, we propose a new well-behaved loss that we call the *Inverse Consistency Averaging Error* (ICAE), as well as a related new triplet loss. Extensive experiments on 128 UCR datasets show that the proposed method outperforms contemporary methods despite not using a regularization. Moreover, ICAE also gives rise to the first DTAN that supports variable-length signals. Our code is available at https://github.com/BGU-CS-VIL/RF-DTAN.

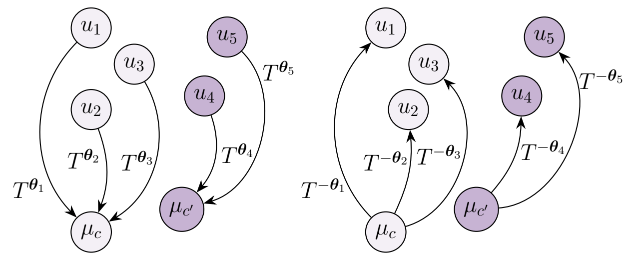
1. Introduction

Nonlinear temporal misalignment between different signals is a major obstacle to time-series statistical analysis. For example, physicians may be interested in the average Electrocardiogram (ECG) signal from a few minutes of recording, but the temporal misalignment across the patient's different heartbeats implies that naively averaging the data will distort the true underlying signal.

A popular attempt to solve the problem relies on **pairwise**

[1]Ben-Gurion University. Correspondence to: Ron Shapira Weber <ronsha@post.bgu.ac.il>, Oren Freifeld <orenfr@cs.bgu.ac.il>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.



(a) Centroids computed using forward warps

(b) The ICAE loss computed using backward warps

Figure 1. The Inverse Consistency Averaging Error loss in a two-class example. (a) The signals u_1, u_2, and u_3 are in class c; u_4 and u_5 are in class c'. Within each class, the centroid (μ_c or $\mu_{c'}$) is obtained by averaging the warped signals ($(u_i \circ T^{\boldsymbol{\theta}_i})_{i\in\{1,2,3\}}$ or $(u_i \circ T^{\boldsymbol{\theta}_i})_{i\in\{4,5\}}$) using the forward warps. (b) The loss is computed using the backward warps; *i.e.*, we measure dissimilarity between each u_i and its class centroid, where the latter is first warped backward ("unwarped") using $T^{-\boldsymbol{\theta}_i}$ (the inverse of $T^{\boldsymbol{\theta}_i}$).

alignments. Let $u_i = (u_i(t))_{t=1}^n$ and $u_j = (u_j(t))_{t=1}^m$ be two real-valued discrete-time signals of lengths n and m, respectively. The optimal pairwise alignment of u_j towards u_i, under some dissimilarity measure D, is defined by

$$T^* = \underset{T\in\mathcal{T}}{\arg\min}\, D(u_i, u_j \circ T) \tag{1}$$

where $\circ$ denotes function composition and $\mathcal{T}$ is a family of *warps* (or warping functions); namely, every $T \in \mathcal{T}$ is a function $T : \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}$ is an interval containing $\{1, \dots, m\}$. For instance, Dynamic Time Warping (DTW) provides the optimal discrete warping path between the time indices of u_i and u_j via dynamic programming, where D is (usually) a Euclidean distance (Sakoe, 1971). More generally, while u_i and u_j are defined over discrete domains (*i.e.*, $\{1, \dots, n\}$ and $\{1, \dots, m\}$), the notation $u_j \circ T$ in Equation 1 implicitly assumes that the value of $u_j(t')$ at every $t' \in \mathbb{R}$ is determined, using interpolation techniques, from (possibly a subset of) the m given values, $(u_j(t))_{t=1}^m$.

In this paper we focus on continuously-defined warps that are order-preserving *diffeomorphisms*. A diffeomorphism (namely, a differentiable invertible function whose inverse is differentiable), is a natural choice for representing time warping (Mumford & Desolneux, 2010). Since spaces of diffeomorphisms are large, and in order to discourage unfavorable solutions, typically some regularization term, denoted by $T \mapsto \mathcal{R}(T;\lambda)$ and parameterized by so-called *hyperparameters* (HP), λ, is added to the objective function; *e.g.*, $\mathcal{R}$ might penalize lack of smoothness (in the machine-learning sense, not calculus) or large deviations from the identity map. Hence, Equation 1 is commonly replaced with

$$T^* = \underset{T\in\mathcal{T}}{\arg\min}\, D(u_i, u_j \circ T) + \mathcal{R}(T;\lambda) \tag{2}$$

where $\mathcal{T}$ is a space of 1D diffeomoprhisms from Ω into $\mathbb{R}$.

In the case of an ensemble of N signals, $(u_i)_{i=1}^N$ where $N > 2$, the pairwise approach usually does not generalize well, is prone to drift errors, and might introduce inconsistent solutions. This motivates approaches for **joint alignment** (JA), also known as global alignment or multiple-sequence alignment. The JA problem is often formulated as

$$(T_i^*)_{i=1}^N, \mu = \underset{(T_i)_{i=1}^N \in\mathcal{T}, u}{\arg\min} \sum_{i=1}^N D(u, u_i \circ T_i) + \mathcal{R}(T_i;\lambda) \tag{3}$$

where $\mathcal{T}$, $\mathcal{R}(\cdot;\lambda)$, and D are as before, T_i is the latent warp associated with u_i, and μ is a latent signal, conceptually thought of as the *average signal* (or *centroid*) of the ensemble. This optimization task may also be amortized via the training of a deep net (*e.g.*, (Shapira Weber et al., 2019; Huang et al., 2021; Martinez et al., 2022)).

We emphasize that *the success of JA methods, including deep-learning (DL) ones, depends crucially on the choice of* $\mathcal{R}(\cdot;\lambda)$ *and, more importantly, the choice of its HP,* λ.

In this work, we propose a regularization-free DL approach based on a new loss, called the **Inverse Consistency Averaging Error (ICAE)**, for time-series JA and averaging. This well-behaved loss, denoted by $\mathcal{L}_{\text{ICAE}}$, alleviates the need for warp regularization and can be used within any JA method, as long as the warps are invertible. Concretely, the ICAE loss encourages both the warps and the latent μ to be consistent with the original signals by warping μ *backward* (also known as *unwarping*) towards each of those signals and then penalizing the difference between each of them and its signal-dependent version of the unwarped μ. That is, letting $\boldsymbol{\theta}_i$ parameterize the i^{th} warp (so $T_i = T^{\boldsymbol{\theta}_i}$), and using the fact that $T^{-\boldsymbol{\theta}_i} = (T^{\boldsymbol{\theta}_i})^{-1}$, we apply $T^{-\boldsymbol{\theta}_i}$ to the current estimate of μ and penalize the difference between $\mu \circ T^{-\boldsymbol{\theta}_i}$ and u_i. See Figure 1 for a conceptual illustration.

Importantly, *the proposed approach frees us from the need to use a regularization* over $(T^{\boldsymbol{\theta}_i})_{i=1}^N$. Another positive aspect of $\mathcal{L}_{\text{ICAE}}$ is that it lends itself immediately to *support variable-length signals*, a capability lacking by existing implementations of leading DL methods for JA and averaging. We demonstrate the validity of the approach on 128 datasets (Dau et al., 2019), and show that when other methods are realistically restricted in their HP search, our method outperforms them by a large margin.

To summarize, **our contributions are:** 1) Introducing the ICAE loss for the JA and averaging task, thereby obviating the need for using a regularization over the predicted warps. 2) A triplet-loss variant of the proposed loss for better inter-class separation. 3) An explicit formulation for JA and averaging of variable-length data. 4) Setting new state-of-the-art (SOTA) results on 128 datasets from the UCR time series classification archive (Dau et al., 2019).

2. Related Work

Dynamic Time Warping (DTW) is a popular distance measure (or discrepancy) between a time-series pair (Sakoe, 1971; Sakoe & Chiba, 1978). Given two signals of lengths n and m, DTW computes the best discrete alignment path in the $n \times m$ pairwise distance matrix. While its complexity is $O(nm)$, enforcing certain constraints on DTW results in a linear complexity. However, generalizing DTW from the pairwise case to the JA of multiple signals is prohibitively expensive since the complexity of finding the optimal discrete alignment between N signals of length n is $O(n^N)$.

To overcome this limitation, several JA methods, working under the DTW geometry, were proposed. The DTW-Barycenter Averaging (DBA) (Petitjean et al., 2011; 2014) employs expectation-maximization (EM) to refine a signal that minimizes the sum of DTW distances from the data; *i.e.*, it alternates between finding μ (while fixing $(T_i)_{i=1}^N$),

$$\mu = \underset{u}{\arg\min} \sum_{i=1}^N D(u, u_i \circ T_i)\,, \tag{4}$$

and finding discretely-defined $(T_i)_{i=1}^N$ (while fixing μ),

$$(T_i^*)_{i=1}^N = \underset{(T_i)_{i=1}^N \in\mathcal{T}}{\arg\min} \sum_{i=1}^N D(\mu, u_i \circ T_i)\,. \tag{5}$$

SoftDTW (Cuturi & Blondel, 2017), a soft-minimum variant of DTW, extends DBA. Instead of using EM, SoftDBA computes μ via gradient-based optimization. SoftDTW has one HP, γ, that controls the smoothness of the alignment ($\gamma = 0$ leads to the original DTW score). SoftDTW-divergence (Blondel et al., 2021) modifies SoftDTW to a proper positive-definite divergence. Both of these optimization-based methods *do not learn* how to find the JA of *new* data; *i.e.*, when new signals arrive, they must be run from scratch in order to achieve JA of the new ensemble. While it is possible to align the new data to the previously-found μ in a pairwise manner, this leads to inferior results (see § 4). Additionally, the time/memory complexity of SoftDTW is $O(mn)$. SoftDTW-div suffers from an even worse complexity for a large n or m; *e.g.*, results on `HandOutlines` (the largest UCR dataset in terms of $n \times N$) were not reported by Blondel et al. (2021), and when we tried to run SoftDTW (using `tslearn` (Tavenard, 2017)) on it, it failed due to memory limitations.

Other methods include the Global Alignment Kernel (GAK) (Cuturi, 2011) on which SoftDTW is based, DTW with Global Invariances which generalizes DTW/SoftDTW to both time and space (Vayer et al., 2020), and Neural Time Warping that relaxes the original problem to a continuous optimization using a neural net (albeit limited in the number of signals it can jointly align) (Kawano et al., 2020).

Spaces of Diffeomorphisms are often used for modeling warping paths between sequences; *e.g.*, Srivastava et al. (2010; 2011) proposed differomoprhisms based on the square-root velocity function (SRVF) representation. However, the employment of diffeomorphisms in DL used to be hindered by the associated expensive computations and/or approximation/discretization schemes. For example, this is why diffeomorphisms could not initially be used effectively within a Spatial Transformer Net (STN) (Jaderberg et al., 2015) since training the latter requires a large number of evaluations of both $x \mapsto T^{\boldsymbol{\theta}}(x)$ and $x \mapsto \nabla_{\boldsymbol{\theta}} T^{\boldsymbol{\theta}}(x)$ (where $\boldsymbol{\theta}$ parameterizes the chosen diffeomorphism family), and these quantities are computed at multiple values of x. This has changed, however, with the emergence of new methods (Skafte Detlefsen et al., 2018; Balakrishnan et al., 2018). In particular, Skafte Detlefsen et al. (2018) built on the CPAB diffeomorphisms (see below) to propose the first diffeomorphic STNs.

CPAB Diffeomorphisms (Freifeld et al., 2015; 2017). The name CPAB, short for CPA-Based, stems from the fact that these parametric diffeomorphisms are based on the integration of Continuous Piecewise-Affine (CPA) velocity fields. Of note, in 1D, the CPAB warp, $x \mapsto T^{\boldsymbol{\theta}}(x)$, has a closed form (Freifeld et al., 2015). While the CPAB warps were proposed by Freifeld et al. (2015) with no relation to DL, it turns out that their expressiveness and efficiency make them an invaluable tool in DL (Hauberg et al., 2016; Skafte Detlefsen et al., 2018; Skafte Detlefsen & Hauberg, 2019; Shapira Weber et al., 2019; Kaufman et al., 2021; Shacht et al., 2021; Schwöbel et al., 2022; Martinez et al., 2022; Neifar et al., 2022) and thus this work uses them too. However, our method is not limited to this choice of $\mathcal{T}$.

A **Temporal Transformer Net (TTN)** is the 1D variant of the STN, where the latter is a DL module which, given a transformation family, predicts and applies a transformation to its input for a downstream task. Lohit et al. (2019) use TTNs with discretized diffeomorphisms for learning rate-invariant discriminative warps. The SRVF framework was integrated into TTNs to either predict DTW-based warping functions (Nunez & Joshi, 2020), learn a generative model over the distribution of SRVF warps (Nunez et al., 2021), and time-series JA (Chen & Srivastava, 2021). However, computations in these nonparametric warps do not scale well with the signal length.

Table 1. Comparing JA/averaging methods. Learning gives the ability to generalize JA to new data. VL indicates whether the method supports variable-length signals.

Method	Reg.-free	Optimization	Learning	VL
Euclidean	✓	*N/A*	✗	✓
DBA	✓	EM	✗	✓
SoftDTW	✗	L-BFGS	✗	✓
DTAN w/ WCSS	✗	DL training	✓	✗
DTAN w/ $\mathcal{L}_{\text{ICAE}}$	✓	DL training	✓	✓

Shapira Weber et al. (2019) propose the **Diffeomorphic Temporal Alignment Net (DTAN)**, a diffeomorphic TTN that, using the parametric and highly-expressive CPAB warps, is an effective learning-based solution for JA and averaging. Shapira Weber et al. (2019) based their DTAN implementation on `libcpab` (Detlefsen, 2018). Recently, Martinez et al. (2022) released another CPAB library, Diffeomorphic Fast Warping (`DIFW`), which, while being similar to `libcpab` (and is, in fact, based on it), is even faster, largely due to the smart discovery of a closed-form gradient (Martinez et al., 2022) for CPAB warps. Together with some other changes and an extensive HP tuning on the test data, this let them propose a DTAN implementation with SOTA results in terms of Nearest Centroid Classification (NCC) accuracy, a standard metric for time-series averaging. Henceforth will refer to the DTAN implementations from Shapira Weber et al. (2019) and Martinez et al. (2022) as $\text{DTAN}_{\text{libcpab}}$ and $\text{DTAN}_{\text{DIFW}}$, respectively. Lastly, ResNet-TW (Huang et al., 2021) also predicts CPAB warps albeit via the Large Deformation Diffeomorphic Metric Mapping framework (Beg et al., 2005).

Warp Regularization. As is typical with diffeomorphisms, CPAB warps too are usually regularized. In particular, the three works above (Shapira Weber et al., 2019; Huang et al., 2021; Martinez et al., 2022), who all use the **within-class-sum-of-squares (WCSS) loss**, also use the following regularization from Freifeld et al. (2015), $\mathcal{R}(T^{\boldsymbol{\theta}_i}; \lambda) = \boldsymbol{\theta}_i^\top \boldsymbol{\Sigma}_{\text{CPA}}^{-1} \boldsymbol{\theta}_i$. The matrix $\boldsymbol{\Sigma}_{\text{CPA}}$ is the covariance of a zero-mean Gaussian smoothness prior over CPA velocity fields and has two HPs: $\lambda_{\boldsymbol{\Sigma}}$, which controls the overall variance, and λ_{smooth}, which controls the smoothness of the fields. Additionally, all these three methods predict a varying number of warps (denoted by N_{warps}), such that their composition yields the final warp.

We conclude the section with Table 1 that summarizes differences between several JA/averaging methods and ours.

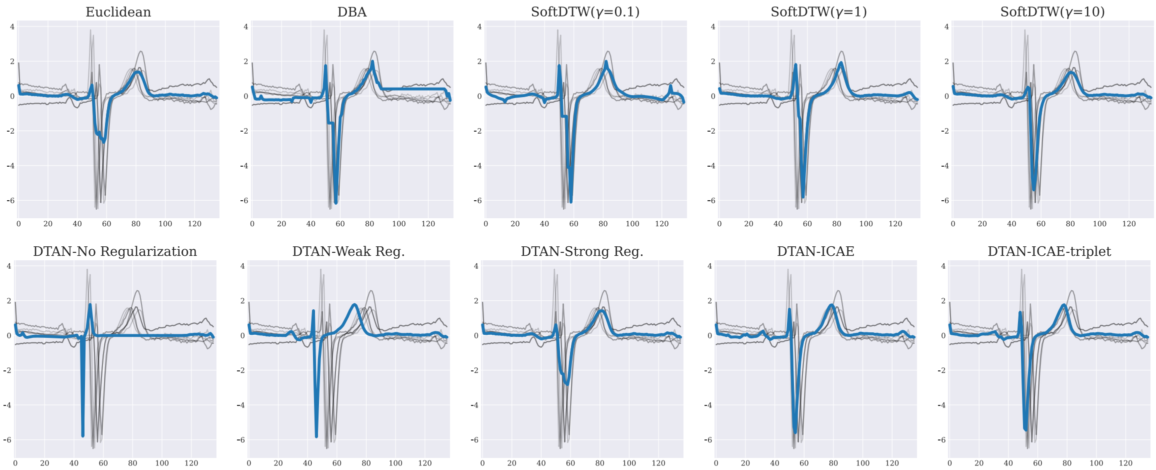


Figure 2. The effect of the regularization HP. The figures shows 10 samples (gray) from the ECGFiveDays dataset with their estimated average (blue), and compares Euclidean averaging, DBA, SoftDTW, and several DTAN methods. DBA requires no HP but falls to poor local minima. SoftDTW's barycenter is severely affected by the choice of its smoothing HP, γ: $\gamma = 0.1$ results in a visible 'pinching' effect while $\gamma = 10$ smoothens out peaks/valleys. DBA and SoftDTW are computed per class and do not learn how to generalize to new data, unlike DTAN which is learning-based and requires a single model for all classes. The regularization often used with DTAN has 2 HPs, $(\lambda_\Sigma, \lambda_{\text{smooth}})$, where a *weak* regularization $(\lambda_\Sigma, \lambda_{\text{smooth}} : .5, .01)$ is insufficient and a *strong* regularization $(\lambda_\Sigma, \lambda_{\text{smooth}} : .001, .1)$, is too restrictive. Our $\mathcal{L}_{\text{ICAE}}$ and $\mathcal{L}_{\text{ICAE-triplet}}$ are regularization-free, yet provide barycenters that represent the data well.

3. Method

We propose a regularization-free approach for time-series JA and averaging using DTAN. Our method leverages the fact that $\mathcal{T}$ is a diffeomorphism family and thus its elements are invertible. Our motivation stems in part from the fact that leading JA methods depend on warp regularization to avoid unrealistic deformation and/or trivial solutions (see Figure 2). *Its optimal HPs, however, are dataset-specific.* As time-series data varies considerably across different application domains (ECG compared with audio recording, for instance), determining a proper value of λ is difficult. For example, Martinez et al. (2022) ran 8064 different experiments (96 different configurations per each of 84 datasets) when evaluating on the UCR archive (Chen et al., 2015) (with such an approach, the 128 datasets of the updated UCR archive (Dau et al., 2019) will require 12288 experiments). Our approach eliminates this issue. The remainder of this section is constructed as follows. In § 3.1, in a presentation that follows Shapira Weber et al. (2019), we briefly explain CPAB warps, and refer to Freifeld et al. (2015; 2017) for more details. in § 3.2 we touch upon the TTN mechanics and our choice of architecture for it. In § 3.3 and § 3.4 we explain our proposed losses, ICAE, and a new triplet-loss variant of it, respectively. § 3.5 details how we handle variable-length data. Finally, in § 3.6 we discuss limitations.

3.1. CPAB Diffeomorphisms

Let Ω be a partition of the signal's time domain into subintervals. Let $\mathcal{V}$ be the linear space of CPA velocity fields w.r.t. Ω, let $d = \dim(\mathcal{V})$, and let $v^{\boldsymbol{\theta}} : \Omega \to \mathbb{R}$, a velocity field parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$, denote the generic element of $\mathcal{V}$, where $\boldsymbol{\theta}$ stands for the coefficient w.r.t. some basis of $\mathcal{V}$. The corresponding space of CPAB warps, obtained via integration of elements of $\mathcal{V}$, is

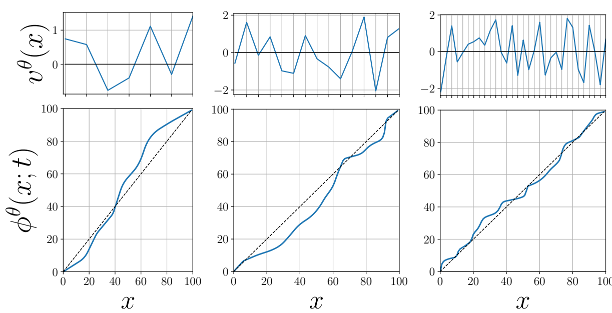


Figure 3. Examples of CPAB warps for three different partitions of Ω. Top: CPA velocity fields. Bottom: The resulting CPAB warps.

$$\mathcal{T} \triangleq \Big\{ T^{\boldsymbol{\theta}} : x \mapsto \phi^{\boldsymbol{\theta}}(x;1) \text{ s.t. } \phi^{\boldsymbol{\theta}}(x;t) \text{ solves}$$
$$\phi^{\boldsymbol{\theta}}(x;t) = x + \int_0^t v^{\boldsymbol{\theta}}(\phi^{\boldsymbol{\theta}}(x;\tau))\,\mathrm{d}\tau \text{ where } v^{\boldsymbol{\theta}} \in \mathcal{V} \Big\}. \tag{6}$$

These order-preserving warps are (C^1) diffeomorphisms (Freifeld et al., 2015; 2017). See Figure 3 for typical CPAB warps. The fineness of Ω determines a trade-off between the expressiveness of $\mathcal{T}$ on the one hand and the computational complexity and dimensionality on the other hand. CPA velocity fields support fast *and* accurate integration methods. Particularly useful in the context of DL is the fact that CPAB warps lend themselves to fast and accurate computation of the so-called CPAB gradient, $x \mapsto \nabla_{\boldsymbol{\theta}} T^{\boldsymbol{\theta}}(x)$. In fact, Martinez et al. (2022), showed that this gradient even has a closed form. Other types of efficient diffeomorphisms (*e.g.*, (Zhang & Fletcher, 2018; Arsigny et al., 2006; Durrleman et al., 2013; Allassonniere et al., 2015)) may also be used in DTAN, provided that there is also an efficient way to evaluate $x \mapsto \nabla_{\boldsymbol{\theta}} T^{\boldsymbol{\theta}}(x)$.

3.2. Temporal Transformer Networks

A TTN, which predicts the warping parameters $\boldsymbol{\theta}$ and applies $T^{\boldsymbol{\theta}}$ to the input signals, consists of three modules. The first is the so-called localization net. This is a neural net, denoted by $f_{\text{loc}}(\cdot)$, which takes as an input a batch of sequences $(u_i)_{i=1}^N$ and predicts the corresponding warping parameters, $(\boldsymbol{\theta}_i)_{i=1}^N$. The second is a grid generator which creates a grid $G \subset \Omega$ of evenly-spaced points which are then warped by $T^{\boldsymbol{\theta}_i}$. Lastly, a grid sampler computes the warped signal $v_i = u_i \circ T^{\boldsymbol{\theta}_i}$ by interpolating its values, using u_i, at $(T^{\boldsymbol{\theta}_i})^{-1}(G)$. See Jaderberg et al. (2015) for details.

In this work, we set f_{loc} to be *InceptionTime* (Ismail Fawaz et al., 2020) instead of a Temporal Convolutional Net (TCN) used in (Shapira Weber et al., 2019; Martinez et al., 2022). Originally designed for time-series classification, InceptionTime was inspired by the Inception-v4 architecture and consists of several Inception modules leveraging the bottleneck design popular in image classification. Notably, we incorporate the Global Average Pooling (GAP) operator before the penultimate layer of f_{loc}, which allows the model to remain fixed in its number of trainable parameters w.r.t. the input size (*e.g.*, we use the same architecture for all UCR datasets). It is also one of the reasons why we can process *variable-length* input at ease (see § 3.5).

3.3. The Inverse Consistency Averaging Error

Christensen & Johnson (2001) introduced the Inverse Consistency Error (ICE) as a regularizer for the task of pairwise image alignment. Given two images, I_1 and I_2, with domains Ω_1 and Ω_2 respectively, the latent spatial maps $f_1 : \Omega_2 \to \Omega_1$ and $f_2 : \Omega_1 \to \Omega_2$ should be consistent; *i.e.*, $f_2 = f_1^{-1}$ and $f_1 = f_2^{-1}$. The ICE, defined as

$$\int_{\Omega_2} \|f_2(f_1(x)) - x\|^2 \mathrm{d}x + \int_{\Omega_1} \|f_1(f_2(x)) - x\|^2 \mathrm{d}x \,, \tag{7}$$

penalizes deviations from that consistency.

We propose a new form of inverse consistency that renders it useful for the JA task as well. Unlike the original ICE, which is pairwise and acts as a regularization term added to the main loss, our proposed $\mathcal{L}_{\text{ICAE}}$ measures the consistency between the estimated average sequence and each of its respective group members. Moreover, rather than being a regularizer term added to another loss, our $\mathcal{L}_{\text{ICAE}}$ is the entire loss by itself. That is, our generalization (of the ICE) stands on its own as a dedicated loss function and results in *consistent JA*. Importantly, and as we will show, it removes the need to use any form of regularization, and this, in turn, removes (trivially) the need to tune regularization HPs.

Following the formulation in (Shapira Weber et al., 2019), let us first recall the previously-used JA loss function in the single- and multi-class cases. For a single class, the loss was the variance of the aligned signals:

$$\mathcal{L}_{\text{data}} \triangleq \frac{1}{N} \sum_{i=1}^{N} \left\| u_i \circ T^{\boldsymbol{\theta}_i} - \mu \right\|_{\ell_2}^2 \tag{8}$$

where $\|\cdot\|_{\ell_2}$ is the ℓ_2 norm, $(f_{\text{loc}}(u_i))_{i=1}^N = (\boldsymbol{\theta}_i)_{i=1}^N$ are the warp parameters predicted by f_{loc}, and

$$\mu = \frac{1}{N} \sum_{i=1}^{N} u_i \circ T^{\boldsymbol{\theta}_i} \tag{9}$$

is the post-alignment average signal. In the multi-class case, the loss was the sum of the within-class variances, often called the within-class sum of squares (WCSS):

$$\mathcal{L}_{\text{data}} \triangleq \sum_{k=1}^{K} \frac{1}{N_K} \sum_{i:y_i=k} \left\| u_i \circ T^{\boldsymbol{\theta}_i} - \mu_k \right\|_{\ell_2}^2 \tag{10}$$

where K is the number of classes, y_i is class label of u_i, N_k is the number of signals in class k, and

$$\mu_k = \tfrac{1}{N_K} \sum\nolimits_{i:y_i=k} u_i \circ T^{\boldsymbol{\theta}_i} \tag{11}$$

is the post-alignment average of class k (this is a semi-supervised problem in the following sense: training is done with known $(y_i)_{i=1}^N$ and unknown $(\boldsymbol{\theta}_i)_{i=1}^N$).

It is clear why, with these losses, the warp-regularization term, $\mathcal{R}(T^{\boldsymbol{\theta}_i}; \lambda)$, is needed. First, the data term, $\mathcal{L}_{\text{data}}$, does not encourage warp consistency. Secondly, it is possible to reduce the variance (even to zero!) by severely distorting the signals, and this issue only worsens due to interpolation artifacts.

However, *optimal regularization is dataset-specific*. For example, penalizing deformations that are too large might not be ideal in many cases. Likewise, with a temporal smoothness prior, it is hard to determine the "right" amount of smoothness. Figure 2 illustrates the critical role of regularization on the barycenter computation using DBA, SoftDTW, and DTAN. Improper values of γ (for SoftDTW) or $\lambda_{\boldsymbol{\Sigma}}, \lambda_{\text{smooth}}$ (for DTAN) usually result in unrealistic warps or overly restrict the warps (*e.g.*, a *strong* prior for DTAN).

Algorithm 1 The JA training with an ICAE loss

Input: N_{epochs}, f_{loc}
Data: $(u_i, y_i)_{i=1}^{N}$
Output: $f_{\text{loc}}(\cdot)$, trained for joint alignment

for *each epoch and each batch* $j \in \{1, \dots, N_{\text{batches}}\}$ **do**
  $\mathcal{L}_{\text{batch}} \leftarrow 0$
  $(u_i, y_i)_{i=1}^{N_j} \leftarrow \text{batch}_j$
  $(\boldsymbol{\theta}_i)_{i=1}^{N_j} \leftarrow (f_{\text{loc}}(u_i))_{i=1}^{N_j}$
  for $k \in \{1, \dots, K\}$ **do**
    $\mu_k = \frac{1}{N_k}\sum_{i:y_i=k}(u_i \circ T^{\boldsymbol{\theta}_i})$
    $\mathcal{L}_{\text{ICAE}} = \frac{1}{N_K}\sum_{i:y_i=k}\|\mu_k \circ T^{-\boldsymbol{\theta}_i} - u_i\|_{\ell_2}^2$
    $\mathcal{L}_{\text{batch}} \mathrel{+}= \mathcal{L}_{\text{ICAE}}$
  Perform an optimization step to minimize $\mathcal{L}_{\text{batch}}$

Instead, we propose a new loss that is minimized when the average sequence is both a minimizer of the variance *and* consistent with its class. Concretely, we propose the Inverse Consistency Averaging Error loss (ICAE), defined as:

$$\mathcal{L}_{\text{ICAE}} \triangleq \sum_{k=1}^{K} \frac{1}{N_K} \sum_{i:y_i=k} \left\| \mu_k \circ T^{-\boldsymbol{\theta}_i} - u_i \right\|_{\ell_2}^2 . \tag{12}$$

$\mathcal{L}_{\text{ICAE}}$ measures how well the average signal, μ_k, fits each signal u_i in its class using the inverse warp $T^{-\boldsymbol{\theta}_i}$. It does so by first aligning all of the signals in class k using the predicted warps, then computing their average μ_k, and finally warping μ_k back toward each u_i using $T^{-\boldsymbol{\theta}_i}$, thereby ensuring consistency between them. *A key insight is that Equation 12 strongly discourages trivial solutions or unrealistic warps as this would result in a poor estimate of μ_k, which in turn would yield a high discrepancy between it and the original signals.* In other words, the loss favors realistic deformations without the need to add a regularization term. The full training procedure is described in Algorithm 1.

3.4. Inverse Consistent Centroids Triplet Loss

While $\mathcal{L}_{\text{ICAE}}$ implies consistency, it is agnostic about the separation between different classes. That said, while metrics such as DTW are completely data-driven, our learning-based can be utilized to learn task-driven representations. As such, we introduce the centroid triplet loss into our framework to encourage inter-class separation. Traditionally, *e.g.* in classification tasks, a triplet loss is defined over a triplet (u_i^a, u_i^p, u_i^n) of an anchor, a positive, and a negative examples, respectively. As our task is intra-class JA and computing class averages (also known as centroids), adopting a centroid-based triplet loss is more adequate here (Doras & Peeters, 2020). We define the *Inverse Consistent Centroids Triplet Loss* over the triplet $(u_i^a, \mu_i^p, \mu_i^n)$ as

$$\begin{aligned} &\mathcal{L}_{\text{ICAE-triplet}}(u_i^a, \mu^p, \mu^n) \triangleq \max(0, \\ &\|u_i^a - \mu^p \circ T^{-\boldsymbol{\theta}_i}\|_{\ell_2}^2 - \|u_i^a - \mu^n \circ T^{-\boldsymbol{\theta}_i}\|_{\ell_2}^2 + \alpha) \end{aligned} \tag{13}$$

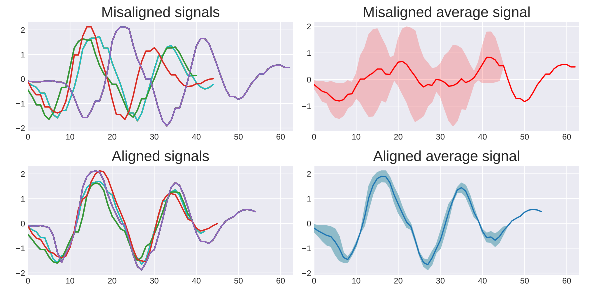


Figure 4. JA of variable-length data (Dataset: *ShakeGestureWiimoteZ*) using the proposed $\mathcal{L}_{\text{ICAE}}$. Shaded area is $\pm$ std. dev.

where μ^p, μ^n are *the* positive and *a* negative class centroids, respectively, and α is the margin between them ($\alpha = 1$ in all our experiments and is dataset-independent). As both μ^p and μ^n are compared via an inverse warp, $\mathcal{L}_{\text{ICAE-triplet}}$ does not break the consistency between samples and their mean. The $\mathcal{L}_{\text{ICAE-triplet}}$ is used in tandem with $\mathcal{L}_{\text{ICAE}}$.

3.5. Variable-Length Joint Alignment

Our proposed $\mathcal{L}_{\text{ICAE}}$ also allows for the JA and averaging of variable-length sequences without having to use a specialized loss function or tweak the boundary conditions on $T^{\boldsymbol{\theta}}$ (as mentioned in (Shapira Weber et al., 2019; Martinez et al., 2022) as a hypothetical possibility). Instead, our formulation (as well as our code) handles both fixed and variable-length data. It does so in the following manner. First, the post-alignment average signal is produced by dividing, at each time step, the sum of the relevant values by the number of non-missing values. That is, for each time step t along the duration of the mean signal μ, we compute:

$$\mu[t] = \frac{1}{N_{\text{valid}}} \sum_{i:(u_i \circ T^{\boldsymbol{\theta}_i})[t] \neq \text{null}}^{N} (u_i \circ T^{\boldsymbol{\theta}_i})[t] \tag{14}$$

where N_{valid} is the number of signals whose domain includes a point mapped to t. Then, when μ is warped backward, Equation 12 is computed with no modifications. See, *e.g.*, Figure 4. From an implementation standpoint, we note that any `null` value in either the input and/or loss would break the computational graph. To avoid `for-loops` and compute back-propagation in batches, it is computationally effective to first pad all samples with zeros (w.r.t. the longest signal) and create an indicator mask for missing values. The mask is also warped by $T^{\boldsymbol{\theta}}$ in Equation 14.

3.6. Limitations

Limitations w.r.t. DTW: DTW-based methods are optimization-based, and thus, when the sample size is very small and/or the signal length is very short, running those methods on such a small *training* data, might be faster than

Table 2. Nearest Centroid Classification Accuracy.

METHOD	OBJECTIVE	$\text{NCC}_{\text{median}}$	NCC_{best}	#CONFIGS	#DATASETS	#EXPERIMENTS
PART 1: ALLOWING HP SEARCH (PREVIOUSLY-REPORTED RESULTS)						
EUCLIDEAN	*N/A*	-	0.611	1	84	84
DBA	DTW	-	0.657	1	84	84
SOFTDTW	SOFTDTW	-	0.703	9	84	756
SOFTDTW	SOFTDTW-DIV	-	0.708	9	84	756
$\text{DTAN}_{\text{libcpab}}$	WCSS + REG	-	0.705	12	84	1008
RESNET-TW	WCSS + REG	-	0.711	20	84	1680
$\text{DTAN}_{\text{DIFW}}$	WCSS + REG	-	**0.749**	96	84	8064
PART 2: SINGLE HP CONFIGURATION IN ALL DATASETS (SAME UCR DATASETS AS REPORTED BY OTHER WORKS ABOVE)						
$\text{DTAN}_{\text{DIFW}}$	WCSS + REG	0.604	0.607	1	84	84
$\text{DTAN}_{\text{DIFW}}$	$\mathcal{L}_{\text{ICAE}}$ (OURS)	0.665	0.694	1	84	84
$\text{DTAN}_{\text{DIFW}}$	$\mathcal{L}_{\text{ICAE-triplet}}$ (OURS)	**0.707**	**0.739**	1	84	84
PART 3: SINGLE HP CONFIGURATION IN ALL DATASETS (INCLUDING ADDITIONAL NEWER FIXED-LENGTH UCR DATASETS)						
$\text{DTAN}_{\text{DIFW}}$	WCSS	0.609	0.65	1	117	117
$\text{DTAN}_{\text{DIFW}}$	WCSS + REG	0.603	0.605	1	117	117
$\text{DTAN}_{\text{DIFW}}$	$\mathcal{L}_{\text{ICAE}}$ (OURS)	0.656	0.686	1	117	117
$\text{DTAN}_{\text{DIFW}}$	$\mathcal{L}_{\text{ICAE-triplet}}$ (OURS)	**0.709**	**0.741**	1	117	117
PART 4: SINGLE HP CONFIGURATION IN ALL DATASETS (FULL UPDATED UCR ARCHIVE, INCLUDING VARIABLE-LENGTH DATASETS)						
$\text{DTAN}_{\text{DIFW}}$	$\mathcal{L}_{\text{ICAE}}$ (OURS)	0.623	0.653	1	128	128
$\text{DTAN}_{\text{DIFW}}$	$\mathcal{L}_{\text{ICAE-triplet}}$ (OURS)	**0.67**	**0.701**	1	128	128

our training time (see Appendix A). We emphasize, however, that if the training data is large (in either dimension) our method is, in fact, usually faster. Additionally, like most learning-based methods, a small train set might result in over-fitting, which will damage performance on test data. Optimization-based methods may not suffer from this issue. Finally, The SoftDTW (Cuturi & Blondel, 2017) smoothness HP, γ, may provide more robustness to amplitude jitter than our method. However, it must be tuned (and this can be expensive or even infeasible), and in practice, the results show that our method still outperforms such methods.

Limitations w.r.t. WCSS loss: During training, our complexity is slightly larger: the proposed $\mathcal{L}_{\text{ICAE}}$ requires two warps per sample (*i.e*., forward and inverse warps), and $\mathcal{L}_{\text{ICAE-triplet}}$ requires 3, while the WCSS requires only the forward warp. Thus, the training times can be slightly longer. However, the difference is small, since the warps are computed very efficiently (using the `DIFW` package (Martinez et al., 2022)) and most of the computation time during training is spent on other parts of the network which are identical regardless which of the losses (WCSS or ICAE) is used. In any case, inference time is identical in both cases since then only a single forward warp is used.

4. Experiments and Results

To evaluate our approach and compare with others, we used the *UCR time-series classification archive* benchmark. The most updated version (Dau et al., 2019) of the UCR archive has 128 datasets with inter-dataset variability in the number of samples, signal length, application domain, and the number of classes. Eleven of those datasets also present intra-dataset variability of the signal length; such datasets are referred to as variable-length (VL) datasets. In all of the experiments, we used the train/test splits provided by the archive. To quantify performances we used, as is customary, the NCC accuracy. This performance index is viewed as an evaluation metric for measuring how well each centroid describes its class members (and thus, implicitly, also measures the JA quality). The NCC framework has 2 steps: 1) compute the centroid, μ_k, for each class of the *train* set; 2) label each *test* sample by the class of its closest centroid. As we explain below, Table 2, which summarizes the NCC results, is divided into several parts. The full results, together with many illustrative figures and computation-time evaluation, appear in our Supplemental Material **(SupMat)**.

Technical details. In all of our DTAN experiments, training was done via the Adam optimizer (Kingma & Ba, 2014) for 1500 epochs, batch size of 64, N_p (the number of subintervals in the partition of Ω) was 16, and the scaling-and-squaring parameter (used by `DIFW`) was 8. These values were previously reported to yield the highest number of *Wins* in (Martinez et al., 2022). As Shapira Weber et al. (2019) used a recurrent variation of DTAN (RDTAN) while Martinez et al. (2022) stacked TCNs, we fixed the number of recurrences to 4 (we did not find it necessary to stack InceptionTime models). The PyTorch `TSAI` implementation of the InceptionTime was taken from (Oguiza, 2022). In the timing experiments (§ 4.3), for DTW, DBA, and SoftDTW we used the `tslearn` package (Tavenard, 2017).

4.1. Nearest Centroid Classification

Part 1: 84 datasets – allowing an extensive HP search (previously-reported results). An older version (Chen et al., 2015) of the UCR archive had only 85 datasets (a subset of the 128 mentioned above). Several previous works reported results on only 84 datasets out of those 85, possibly due to the size of the largest dataset. Part 1 of Table 2 contains the results, on those 84 datasets, obtained by several key methods, as reported by their authors, as well as those obtained by a simple Euclidean averaging (*i.e.*, a no-alignment baseline). The methods are DBA, SoftDTW, $\text{DTAN}_{\text{libcpab}}$, ResNet-TW, and $\text{DTAN}_{\text{DIFW}}$. The regularization-free DBA requires no HP configurations. The SoftDTW methods have one HP for controlling the smoothness. Their results, reported in (Blondel et al., 2021), were obtained by those authors using cross-validation. The other works (Shapira Weber et al., 2019; Huang et al., 2021; Martinez et al., 2022) reported only their best results across different configurations. Shapira Weber et al. (2019) evaluated $\text{DTAN}_{\text{libcpab}}$ using 12 different configurations per dataset (4 configurations for $(\lambda_{\Sigma}, \lambda_{\text{smooth}})$ and 3 different numbers of recurrences). In (Huang et al., 2021), ResNet-TW used the same regularization configurations as in Shapira Weber et al. (2019), but also tested varying numbers of ResNet blocks (4 to 8) per dataset. Martinez et al. (2022) evaluated $\text{DTAN}_{\text{DIFW}}$ using 96 different configurations (various options of $\lambda_{\Sigma}, \lambda_{\text{smooth}}, N_p$, #stacked TCNs, boundary conditions, and the scaling-and-squaring parameter) per dataset. We note that: 1) tuning N_p and the boundary conditions is another form of tweaking the regularization; 2) as stated in supplemental material of (Martinez et al., 2022), their reported results were chosen among those 96 configurations, per dataset, based on the best performance on the test set.

Part 2: Using a single HP configuration in all 84 datasets. Part 1 of Table 2 suggests that increasing the number of tried HP configurations translates to better performance due to the large variability across the UCR datasets. However, the compact summary in Part 1 of Table 2 also hides an ugly truth: there is no `one-size-fits-all` configuration. For example, $\text{DTAN}_{\text{DIFW}}$ produced the best performance but this is largely due to the fact they performed an expensive search over a large number of HP configurations. In fact, inspecting the full results of either $\text{DTAN}_{\text{libcpab}}$, ResNet-TW, or $\text{DTAN}_{\text{DIFW}}$, reveals that the optimal choice of HP varies across the datasets and affects results drastically.

To demonstrate this crucial point, we ran a new set of experiments. We picked the HP configuration that according to Martinez et al. (2022) achieved the highest number of wins among their 96 configurations. Next, using that configuration we ran, on those 84 datasets, exactly the same DTAN but with 3 different losses: 1) WCSS plus the smoothness regularization (λ_{Σ} and λ_{smooth}, 0.001 and 0.1, respectively); 2) our proposed $\mathcal{L}_{\text{ICAE}}$; 3) our proposed $\mathcal{L}_{\text{ICAE}-\text{triplet}}$. In the last 2 cases, which are regularization-free, the values of λ_{Σ} and λ_{smooth} from that configuration were ignored. In all 3 cases, we used $\text{DTAN}_{\text{DIFW}}$ with the same InceptionTime backbone (Oguiza, 2022) (in all 3 cases this gave better results than using a TCN). To account for random initializations and the stochastic nature of DL training, in each of the 3 cases we performed 5 runs on each dataset and report both the median and best results; see part 2 in Table 2. The results illustrate the merits of the proposed method: a single HP configuration for the regularization, even the one stated as the best, does not properly fit the entirety of the UCR datasets. In contrast, dropping the regularization term and using our $\mathcal{L}_{\text{ICAE}}$ increases performance by a large margin, which is only further increased when utilizing $\mathcal{L}_{\text{ICAE}-\text{triplet}}$, which increases separability between class centroids (a feat current DTW-based methods are incapable of) and achieves SOTA results.

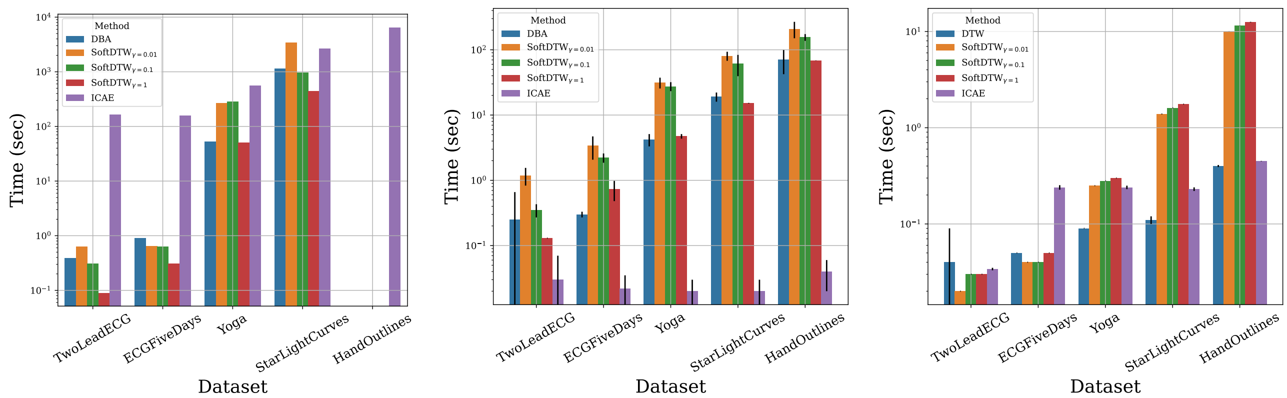
Part 3 & 4: Using a single HP configuration in all of the 128 datasets. To produce the results in part 3 of the table, we again repeated the procedure from part 2, except that 1) we added another case where the loss is only WCSS with no regularization, and 2) the results, on 117 datasets, also take into account additional fixed-length datasets that were added in the newer UCR archive. The results in, and conclusions from, Part 3 are consistent with Part 2. WCSS did slightly better than WCSS+Reg, probably since even though it distorts the signals, it makes it a bit easier (than in the WCSS+Reg case) to differentiate between classes. In any case, our losses outperform both of these methods. **Part 4** extends the results of Part 3 by adding, for the DTANs with our proposed losses, the 11 VL datasets (for a total of 128).

4.2. Ablation Study

An **ablation study** w.r.t. the backbones and losses is presented in Table 3. Note that `SmoothSubspace` dataset (length=15) was omitted for the TCN experiments since it was too short for the MaxPooling operations.

Table 3. Ablation study

Backbone	Objective	NCC	#Datasets
TCN	$\mathcal{L}_{\text{ICAE}}$	0.611	127
TCN	$\mathcal{L}_{\text{ICAE}-\text{triplet}}$	0.632	127
InceptionTime	$\mathcal{L}_{\text{ICAE}}$	0.623	127
InceptionTime	$\mathcal{L}_{\text{ICAE}-\text{triplet}}$	**0.67**	127
InceptionTime	WCSS-triplet	0.642	117
InceptionTime	WCSS-triplet + Reg.	0.603	117
InceptionTime	$\mathcal{L}_{\text{ICAE}}$	0.656	117
InceptionTime	$\mathcal{L}_{\text{ICAE}-\text{triplet}}$	**0.709**	117



(a) Time to compute train set barycenters

(b) Inference - computing barycenters for 30 new samples

(c) Distance to barycenter using the corresponding metric

Figure 5. Timing comparison (the y-axis is log-scaled). See Table 4 in Appendix A for full details.

4.3. Computation-time Comparison

A key advantage of learning-based approaches is fast inference on new data. We performed several timing experiments between DBA, SoftDTW (whose HP, $\gamma \in \{0.01, 0.1, 1\}$, must be searched in each dataset), and DTAN, trained with the proposed $\mathcal{L}_{\mathrm{ICAE}}$. We used a machine with 12 CPU-cores, 32Gb RAM, and an RTX 3090 graphic card. We chose a subset of the UCR archive, spanning different lengths and sample sizes, and compared the time it took to compute the centroids on the entire train set. Then, since DBA and SoftDTW are optimization-based we provide timing for two approaches: (1) barycenter computation time of a new batch ($N = 30$, average of 5 runs) and (2) computing DTW/SoftDTW between the batch and its barycenter (which, after warping, can be averaged again). For DTAN, this is just the inference time. Figure 5 presents the result (while Appendix A presents the full datasets details). On training data, for smaller datasets (in terms of n, N), SoftDTW/DBA is faster than DTAN, but this trend is reversed for the larger ones. **SoftDTW and DBA runs out of memory** on the largest dataset (`HandOutlines`). During inference, using DTAN is *orders of magnitude faster* (x10–x10^4) than recomputing barycenters, and, on the larger datasets, is x10 faster than computing DTW/SoftDTW .

4.4. Multivariate Data

Joint alignment of multivariate time-series data requires special attention due to the usually-complicated inter-channel relationships. When the channels are highly-correlated, a single warp (*i.e.*, a single $\boldsymbol{\theta}$) may suffice. Otherwise, warping each channel independently is preferable. The proposed loss, $\mathcal{L}_{\mathrm{ICAE}}$, supports both options. While complete analysis of multivariate data is outside the scope of this paper, as a proof of concept we trained DTAN with $\mathcal{L}_{\mathrm{ICAE}}$ (using a single warp for all channels) on the *SpokenArabicDigits* dataset (Bagnall et al., 2018) which contains 13 channels and 10 classes. The NCC accuracy for the baseline and the proposed ICAE are 0.08 and 0.402 respectively, demonstrating the potential efficacy of the approach such data.

5. Conclusion

We have proposed the *Inverse Consistency Averaging Error*, $\mathcal{L}_{\mathrm{ICAE}}$, a novel loss function for regularization-free time-series joint alignment and averaging via diffeomorphic temporal transformer nets. The approach utilizes the invertibility of diffeomorphic warps and yields an effective JA while alleviating the need for extensive HP search. We also proposed the $\mathcal{L}_{\mathrm{ICAE-triplet}}$ which allows for a better inter-class separation using a warp-consistent variant of the triplet centroid loss. Additionally, we introduced a formulation of the joint alignment of variable-length time-series data via the proposed framework. Extensive experiments on 128 datasets demonstrate the validity of our approach, resulting in SOTA performance while requiring no warp regularization. Finally, our approach may also be used in conjunction with another regularization-free method for joint alignment which was suggested in (Erez et al., 2022) for spatial warps that relied on a memory-based formulation or with transformation-invariant clustering (Monnier et al., 2020)

Acknowledgments

This work was supported by the Lynn and William Frankel Center at BGU CS, by the Israeli Council for Higher Education via the BGU Data Science Research Center, and by Israel Science Foundation Personal Grant #360/21. R.S.W was also funded in part by the BGU Kreitman School Negev Scholarship.

References

Allassonniere, S., Durrleman, S., and Kuhn, E. Bayesian mixed effect atlas estimation with a diffeomorphic deformation model. *SIAM Journal on Imaging Sciences*, 2015. 5

Arsigny, V., Commowick, O., Pennec, X., and Ayache, N. A log-euclidean polyaffine framework for locally rigid or affine registration. In *BIR*. Springer, 2006. 5

Bagnall, A., Dau, H. A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., and Keogh, E. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018. 9

Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., and Dalca, A. V. An unsupervised learning model for deformable medical image registration. In *CVPR*, 2018. 3

Beg, M. F., Miller, M. I., Trouvé, A., and Younes, L. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *IJCV*, 2005. 3

Blondel, M., Mensch, A., and Vert, J.-P. Differentiable divergences between time series. In *AISTATS*. PMLR, 2021. 2, 3, 8, 29

Chen, C. and Srivastava, A. Srvfregnet: Elastic function registration using deep neural networks. In *CVPR*, 2021. 3

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., and Batista, G. The UCR time series classification archive, 2015. 4, 8, 29

Christensen, G. E. and Johnson, H. J. Consistent image registration. *IEEE TMI*, 2001. 5

Cuturi, M. Fast global alignment kernels. In *ICML*, 2011. 3

Cuturi, M. and Blondel, M. Soft-dtw: a differentiable loss function for time-series. *arXiv preprint arXiv:1703.01541*, 2017. 2, 7

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 2019. 2, 4, 7, 16, 31

Detlefsen, N. S. libcpab. `https://github.com/SkafteNicki/libcpab`, 2018. 3

Doras, G. and Peeters, G. A prototypical triplet loss for cover detection. In *ICASSP*. IEEE, 2020. 6

Durrleman, S., Allassonnière, S., and Joshi, S. Sparse adaptive parameterization of variability in image ensembles. *IJCV*, 2013. 5

Erez, G., Weber, R. S., and Freifeld, O. A deep moving-camera background model. In *ECCV*. Springer, 2022. 9

Freifeld, O., Hauberg, S., Batmanghelich, K., and Fisher III, J. W. Highly-expressive spaces of well-behaved transformations: Keeping it simple. In *ICCV*, 2015. 3, 4

Freifeld, O., Hauberg, S., Batmanghelich, K., and Fisher III, J. W. Transformations based on continuous piecewise-affine velocity fields. *IEEE TPAMI*, 2017. 3, 4

Hauberg, S., Freifeld, O., Larsen, A. B. L., III, J. W. F., and Hansen, L. K. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *AISTATS*, 2016. 3

Huang, H., Amor, B. B., Lin, X., Zhu, F., and Fang, Y. Residual networks as flows of velocity fields for diffeomorphic time series alignment. *arXiv preprint arXiv:2106.11911*, 2021. 2, 3, 8, 29

Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 2020. 5

Jaderberg, M., Simonyan, K., Zisserman, A., et al. Spatial transformer networks. In *NeurIPS*, 2015. 3, 5

Kaufman, I., Weber, R. S., and Freifeld, O. Cyclic diffeomorphic transformer nets for contour alignment. In *IEEE ICIP*, 2021. 3

Kawano, K., Kutsuna, T., and Koide, S. Neural time warping for multiple sequence alignment. In *IEEE ICASSP*, 2020. 3

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, 2014. URL `http://arxiv.org/abs/1412.6980`. 7

Lohit, S., Wang, Q., and Turaga, P. Temporal transformer networks: Joint learning of invariant and discriminative time warping. In *CVPR*, 2019. 3

Martinez, I., Viles, E., and Olaizola, I. G. Closed-form diffeomorphic transformations for time series alignment. In *ICML*. PMLR, 2022. 2, 3, 4, 5, 6, 7, 8, 29

Monnier, T., Groueix, T., and Aubry, M. Deep transformation-invariant clustering. *NeurIPS*, 2020. 9

Mumford, D. and Desolneux, A. *Pattern theory: the stochastic analysis of real-world signals*. AK Peters/CRC Press, 2010. 1

Neifar, N., Ben-Hamadou, A., Mdhaffar, A., Jmaiel, M., and Freisleben, B. Leveraging statistical shape priors in gan-based ECG synthesis. *arXiv preprint arXiv:2211.02626*, 2022. 3

Nunez, E. and Joshi, S. H. Deep learning of warping functions for shape analysis. In *CVPR Workshops*, 2020. 3

Nunez, E., Lizarraga, A., and Joshi, S. H. Srvfnet: A generative network for unsupervised multiple diffeomorphic functional alignment. In *CVPR*, 2021. 3

Oguiza, I. tsai - a state-of-the-art deep learning library for time series and sequential data. Github, 2022. URL `https://github.com/timeseriesAI/tsai`. 7, 8

Petitjean, F., Ketterlin, A., and Gançarski, P. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 2011. 2

Petitjean, F., Forestier, G., Webb, G. I., Nicholson, A. E., Chen, Y., and Keogh, E. Dynamic time warping averaging of time series allows faster and more accurate classification. In *ICDM*. IEEE, 2014. 2

Sakoe, H. Dynamic-programming approach to continuous speech recognition. *International Congress of Acoustics*, 1971. 1, 2

Sakoe, H. and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1978. 2

Schwöbel, P., Warburg, F. R., Jørgensen, M., Madsen, K. H., and Hauberg, S. Probabilistic spatial transformer networks. In *UAI*, 2022. 3

Shacht, G., Danon, D., Fogel, S., and Cohen-Or, D. Single pair cross-modality super resolution. In *CVPR*, 2021. 3

Shapira Weber, R., Eyal, M., Skafte Detlefsen, N., Shriki, O., and Freifeld, O. Diffeomorphic temporal alignment nets. In *NeurIPS*, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 29

Skafte Detlefsen, N. and Hauberg, S. Explicit disentanglement of appearance and perspective in generative models. *NeurIPS*, 2019. 3

Skafte Detlefsen, N., Freifeld, O., and Hauberg, S. Deep diffeomorphic transformer networks. In *CVPR*, 2018. 3

Srivastava, A., Klassen, E., Joshi, S. H., and Jermyn, I. H. Shape analysis of elastic curves in euclidean spaces. *IEEE TPAMI*, 2010. 3

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., and Marron, J. S. Registration of functional data using fisher-rao metric. *arXiv preprint arXiv:1103.3817*, 2011. 3

Tavenard, R. tslearn: a machine learning toolkit dedicated to time-series data (2017). *URL https://github.com/rtavenar/tslearn*, 2017. 3, 7

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *JMLR*, 2008. 13

Vayer, T., Chapel, L., Courty, N., Flamary, R., Soullard, Y., and Tavenard, R. Time series alignment with global invariances. *arXiv preprint arXiv:2002.03848*, 2020. 3

Zhang, M. and Fletcher, P. T. Fast diffeomorphic image registration via fourier-approximated lie algebras. *IJCV*, 2018. 5

Supplemental Material

The Supplemental Material is organized as follows:

- Appendix A: A computation-time study between time-series averaging methods.
- Appendix B: t-SNE projections.
- Appendix C: An illustration of the different warps obtained by DTW on the one hand, and DTAN trained with the proposed $\mathcal{L}_{\mathrm{ICAE}}$ on the other hand.
- Appendix D: A training procedure illustration between the WCSS and our $\mathcal{L}_{\mathrm{ICAE}}$.
- Appendix E: An illustration of unwarping the class mean to the original samples.
- Appendix F: Joint-alignment results on various datasets.
- Appendix G: A visual comparison of time-series averaging methods.
- Appendix H: UCR archive details.
- Appendix I: Full NCC results for all of the UCR archive datasets.

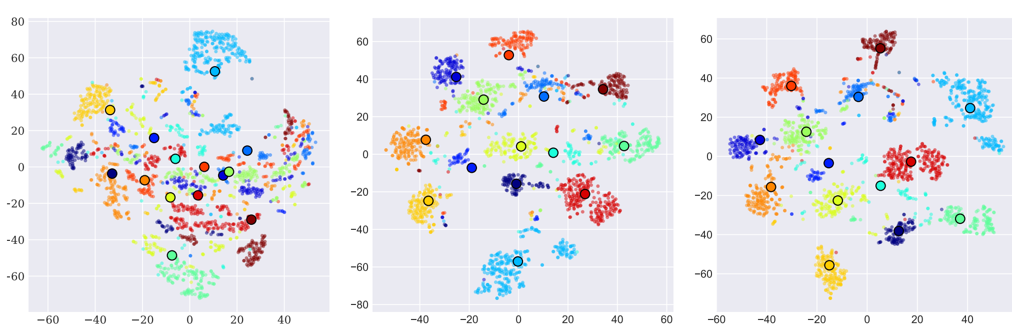
A. Computation Time

Table 4. Timing comparison for several datasets of the UCR archive. (Top) During the fitting/training step, SoftDTW/DBA are computed per class while $\mathrm{DTAN}_{\mathrm{ICAE}}$ uses one model for all classes. (Middle) During inference, 30 new samples are averaged. Soft/DBA needs to be called again as it is optimization-based, while $\mathrm{DTAN}_{\mathrm{ICAE}}$ requires a single forward pass. (Bottom) Finally, each new sample is compared to its train-set barycenter using the corresponding metric. **N/A = Out Of Memory** (on a machine with 12 CPU cores and 32Gb RAM)

.

Dataset	$N_{samples}$	N_{class}	Length	DBA	$\mathrm{SoftDTW}_{\gamma=0.01}$	$\mathrm{SoftDTW}_{\gamma=0.1}$	$\mathrm{SoftDTW}_{\gamma=1}$	ICAE
				Training time - full train set (sec)				
TwoLeadECG	23	2	82	0.39	0.64	0.31	**0.09**	164.78
ECGFiveDays	23	2	136	0.90	0.65	0.64	**0.31**	157.39
Yoga	300	2	426	52.4104	265.493	283.566	**50.3923**	565.65
StarLightCurves	300	3	1024	1140.79	3399.90	964.21	**441.33**	2657.20
HandOutlines	1000	2	2709	N/A	N/A	N/A	N/A	**6483.50**
				Inference time, averaged over 5 runs (sec)				
TwoLeadECG	30	1	82	0.25±0.41	1.19±0.36	0.35±0.08	0.13±0.0	**0.03±0.04**
ECGFiveDays	30	1	136	0.3±0.03	3.39±1.32	2.22±0.36	0.73±0.25	**0.02±0.013**
Yoga	30	1	426	4.21±0.9	31.46±5.92	27.58±4.33	4.73±0.38	**0.02±0.01**
StarLightCurves	30	1	1024	19.08±3.06	80.52±12.71	61.5±22.07	15.2±0.15	**0.02±0.01**
HandOutlines	30	1	2709	70.69±28.39	209.2±58.93	155.53±18.37	68.54±0.3	**0.04±0.02**
				Distance to barycenter using the corresponding metric, averaged over 5 runs (sec)				
TwoLeadECG	30	1	82	0.04±0.05	**0.02±0.0**	0.03±0.0	0.03±0.0	0.034±0.001
ECGFiveDays	30	1	136	0.05±0.0	0.04±0.0	0.04±0.0	0.05±0.0	**0.024±0.0**
Yoga	30	1	426	0.09±0.0	0.25±0.0	0.28±0.0	0.3±0.0	**0.024±0.0**
StarLightCurves	30	1	1024	0.11±0.01	1.39±0.01	1.61±0.0	1.76±0.0	**0.023±0.0**
HandOutlines	30	1	2709	0.4±0.01	10.03±0.01	11.5±0.03	12.54±0.07	**0.045±0.002**

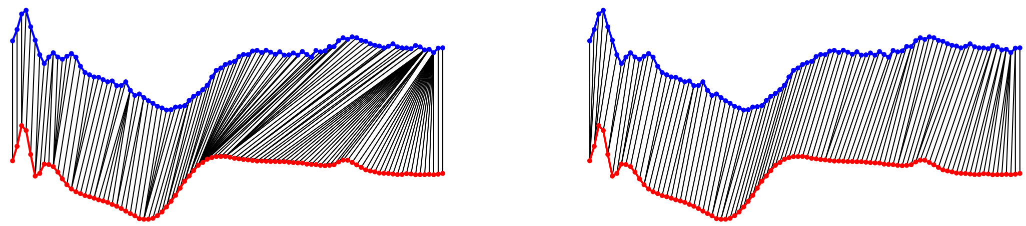
B. t-SNE Projection



(a) Original data (b) $\mathcal{L}_{\mathrm{ICAE}}$ (c) $\mathcal{L}_{\mathrm{ICAE-triplet}}$

Figure 1. Comparison of t-SNE projections (Van der Maaten & Hinton, 2008) of the original and aligned test data (*i.e.*, not embedding) of the 14-class FacesUCR dataset with their respective class centroids. Our proposed $\mathcal{L}_{ICAE}$ decreases the within-class variance, while $\mathcal{L}_{ICAE-triplet}$ increases the inter-class variance further.

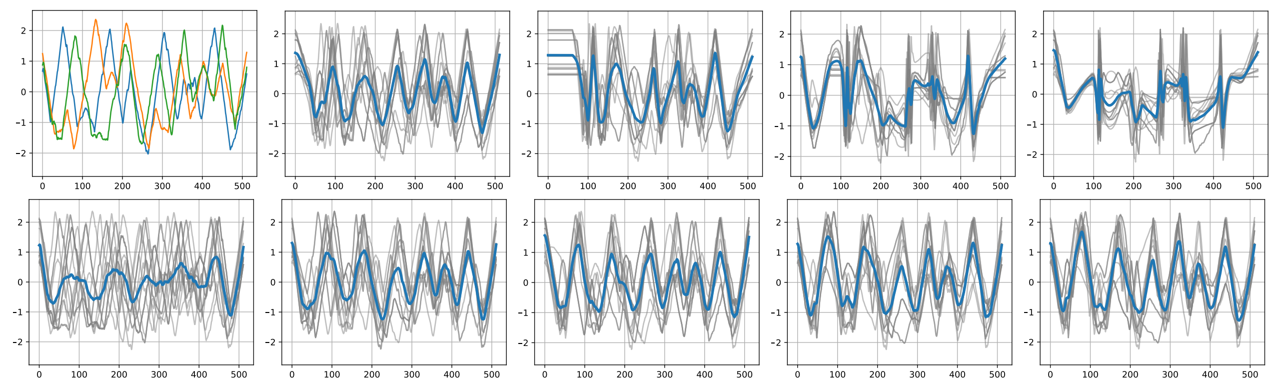
C. DTW vs. DTAN$_{\text{ICAE}}$ warping



(a) DTW (b) DTAN$_{\mathcal{L}_{ICAE}}$

Figure 2. Warping paths computed by Dynamic Time Warping (DTW) and *predicted* by DTAN using the proposed $\mathcal{L}_{ICAE}$, between a test sample (blue) and the class average (red, computed by DTAN). DTW is prone to overfit the signal's noise, whereas our method manages to capture the underlying structure of the time series and provide robust alignment.

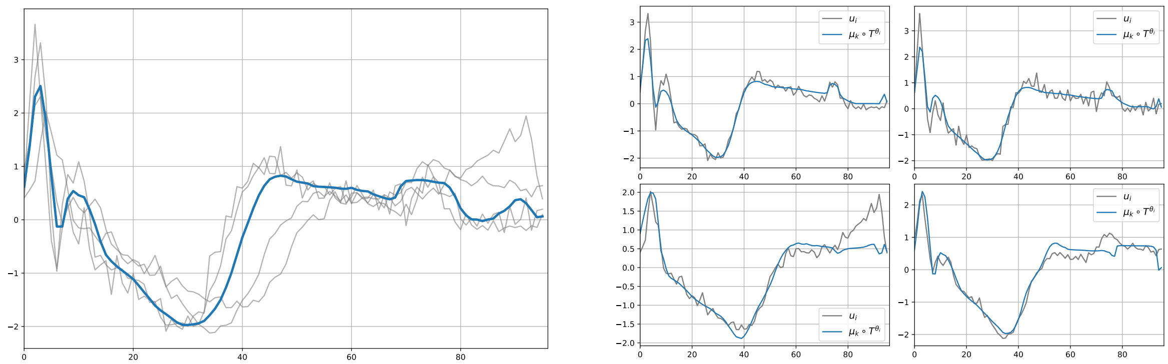
D. Training Procedure Illustration



(a) Input data (b) Epoch 10 (c) Epoch 20 (d) Epoch 50 (e) Epoch 100

Figure 3. Training procedure on the *BeetleFly* dataset. The first column depicts the input data (for better visualization, the top panel shows 3 random signals while the bottom 10 signals and their average are in blue). (**Top**) The Within-Class Sum of Squares (WCSS) loss reduces variance by applying an unrealistic deformation to the data, resulting in visible 'pinching' effect (*i.e.*, bad local minima). (**Bottom**) The proposed $\mathcal{L}_{\text{ICAE}}$, while requiring no regularization, avoids such an undersired solution by maintaining consistency between the average sequence and its class members.

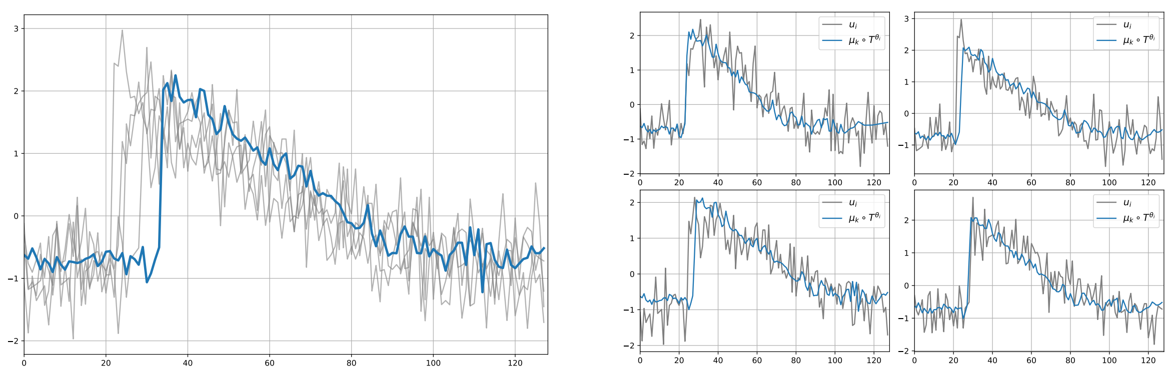
E. Inverse Warping Examples



(a) Class average, μ_k, (blue) with 4 samples, u_i's, (grey).

(b) Uwarping μ_k to each sample (*i.e.*, $\mu_k \circ T^{-\boldsymbol{\theta}_i}$)

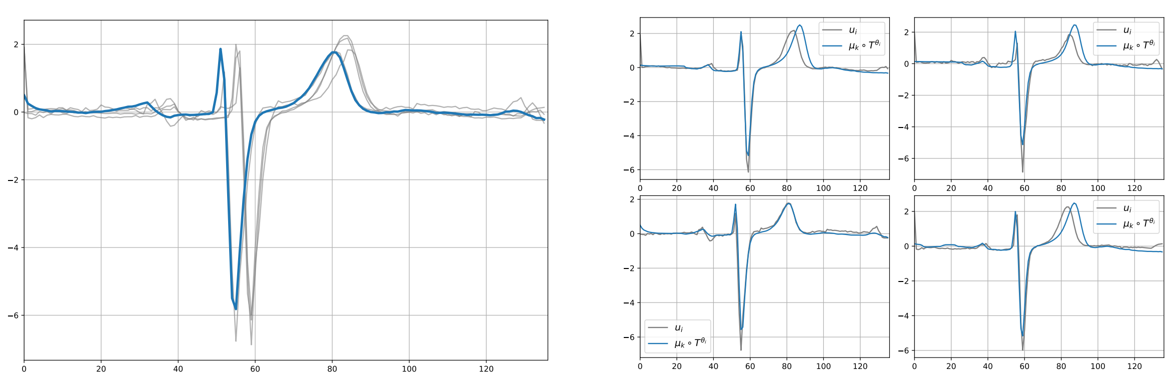
Figure 4. Unwarping the class average to the original data for the *ECG200* dataset.



(a) Class average, μ_k, (blue) with 4 samples, u_i's, (grey).

(b) Uwarping μ_k to each sample (*i.e.*, $\mu_k \circ T^{-\boldsymbol{\theta}_i}$)

Figure 5. Unwarping the class average to the original data for the *CBF* dataset.



(a) Class average, μ_k, (blue) with 4 samples, u_i's, (grey).

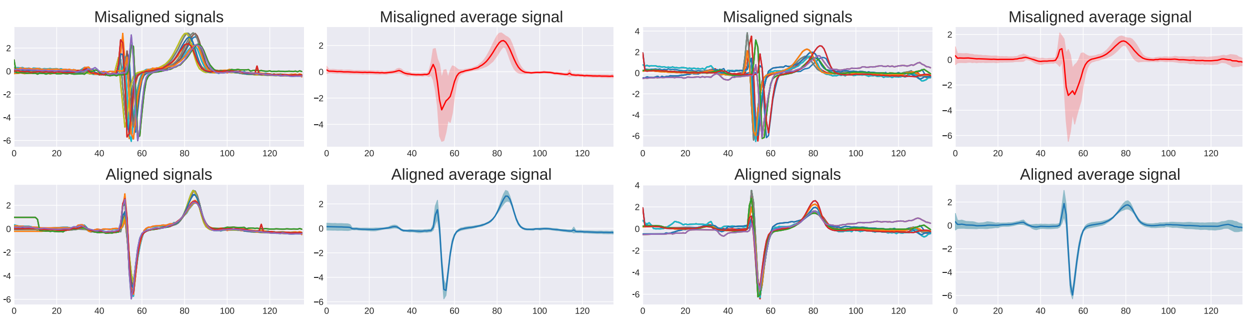
(b) Uwarping μ_k to each sample (*i.e.*, $\mu_k \circ T^{-\boldsymbol{\theta}_i}$)

Figure 6. Unwarping the class average to the original data for the *ECGFiveDays* dataset.

F. Joint Alignment Results

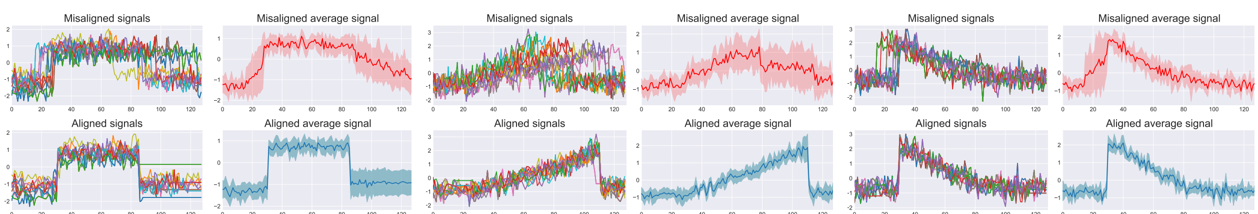
Here we provide additional results for the joint alignment and averaging of various datasets of the UCR time series classification archive (Dau et al., 2019) using our proposed $\mathcal{L}_{\mathrm{ICAE}}$. The results are provided for both the train and test sets.

F.1. Train data



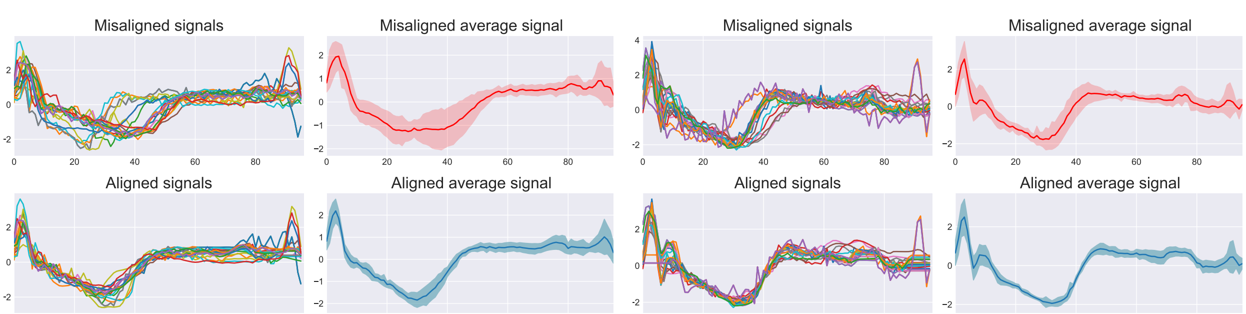
(a) Class 0 (b) Class 1

Figure 7. Joint alignment and averaging of the *ECGFiveDays* dataset. Shaded area corresponds to $\pm\sigma$.



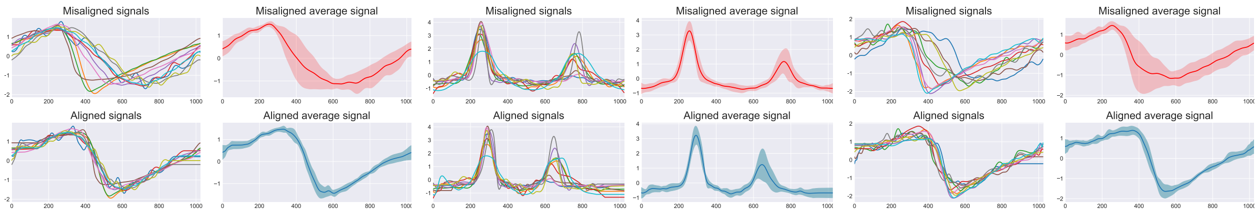
(a) Class 0 (b) Class 1 (c) Class 2

Figure 8. Joint alignment and averaging of the *CBF* dataset. Shaded area corresponds to $\pm\sigma$.



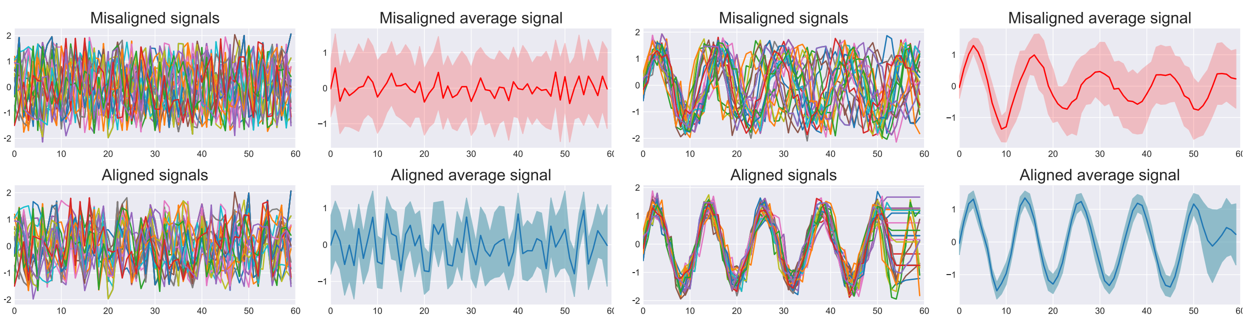
(a) Class 0 (b) Class 1

Figure 9. Joint alignment and averaging of the *ECG200* dataset. Shaded area corresponds to $\pm\sigma$.

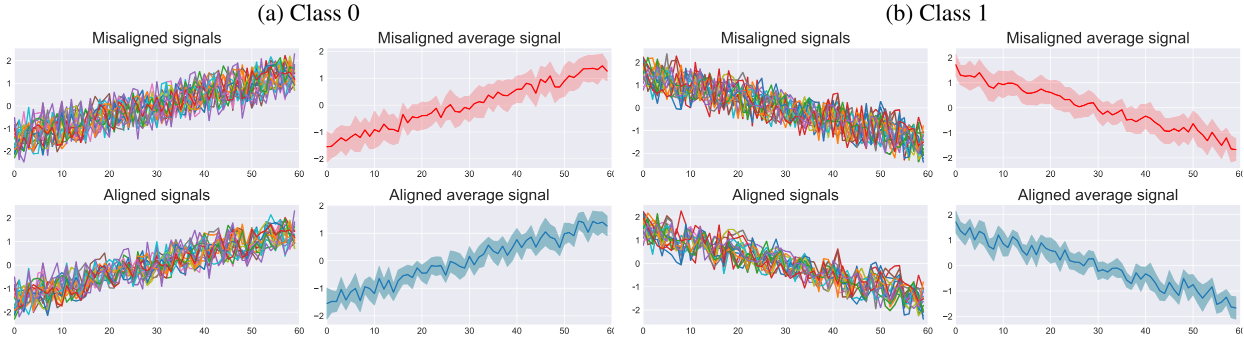


(a) Class 0 (b) Class 1 (c) Class 2

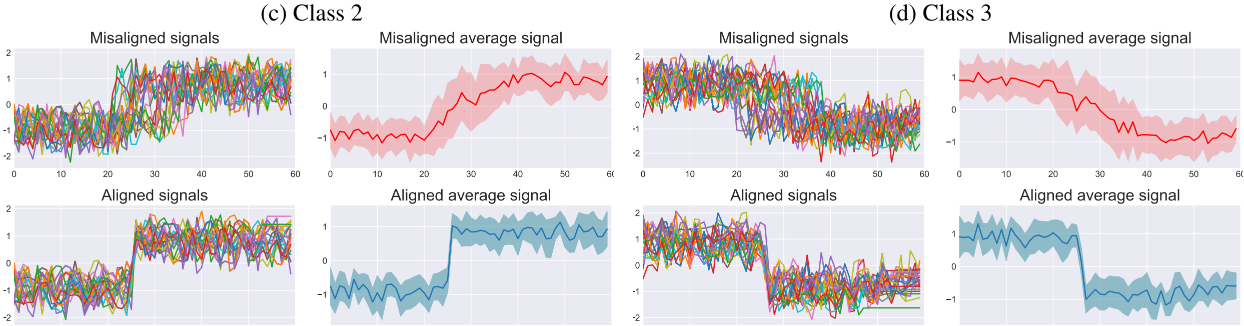
Figure 10. Joint alignment and averaging of the *StarLightCurves* dataset. Shaded area corresponds to $\pm\sigma$.



(a) Class 0 (b) Class 1

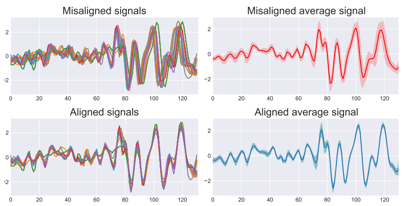


(c) Class 2 (d) Class 3

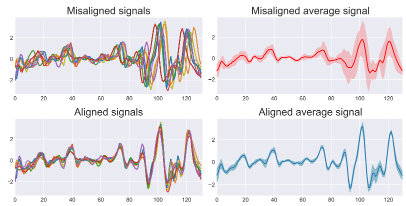


(e) Class 4 (f) Class 5

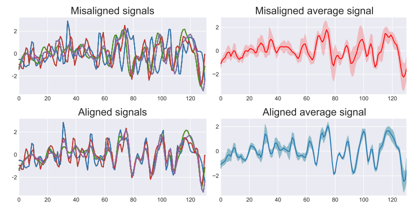
Figure 11. Joint alignment and averaging of the *SyntheticControl* dataset. Shaded area corresponds to $\pm\sigma$.



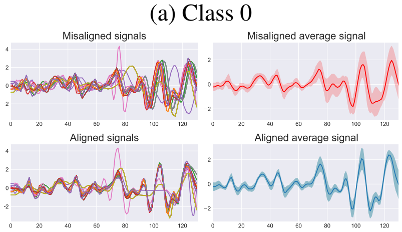
(a) Class 0



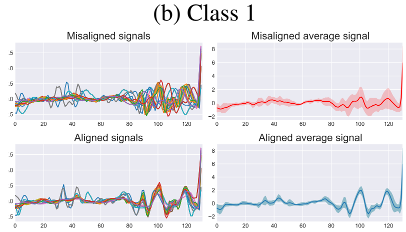
(b) Class 1



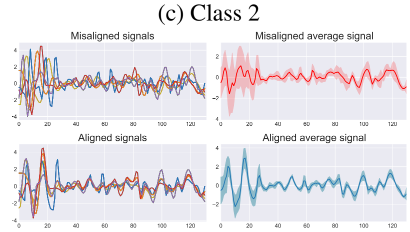
(c) Class 2



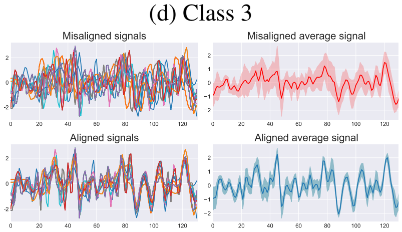
(d) Class 3



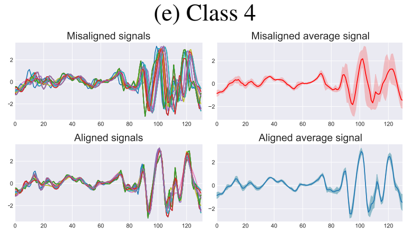
(e) Class 4



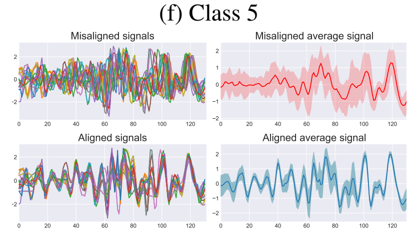
(f) Class 5



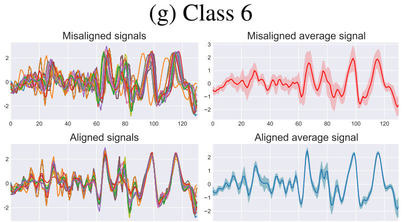
(g) Class 6



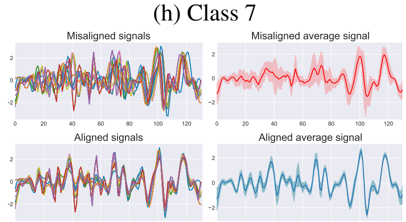
(h) Class 7



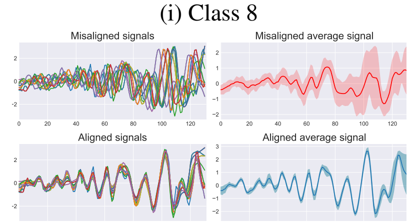
(i) Class 8



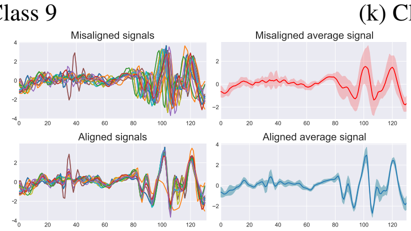
(j) Class 9



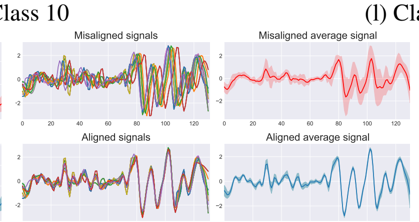
(k) Class 10



(l) Class 11



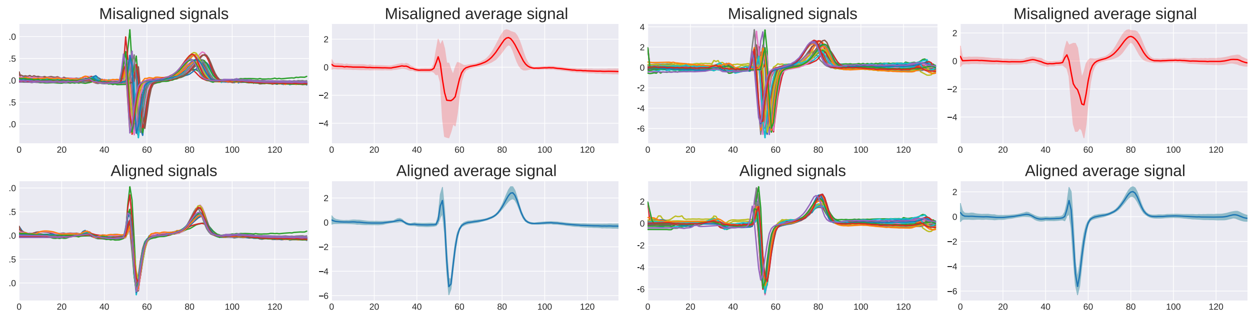
(m) Class 12



(n) Class 13

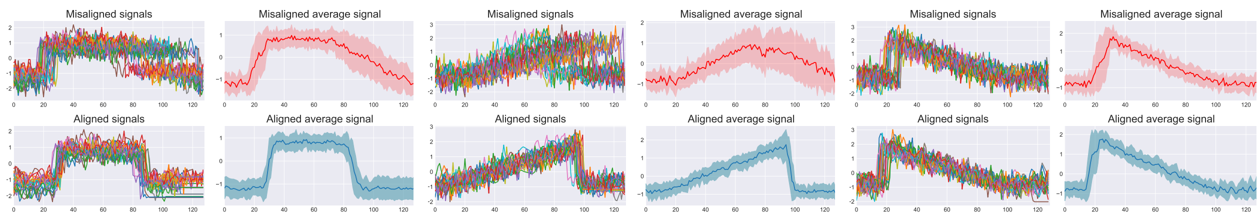
Figure 12. Joint alignment and averaging of the *FacesUCR* dataset. Shaded area corresponds to $\pm\sigma$.

F.2. Test data



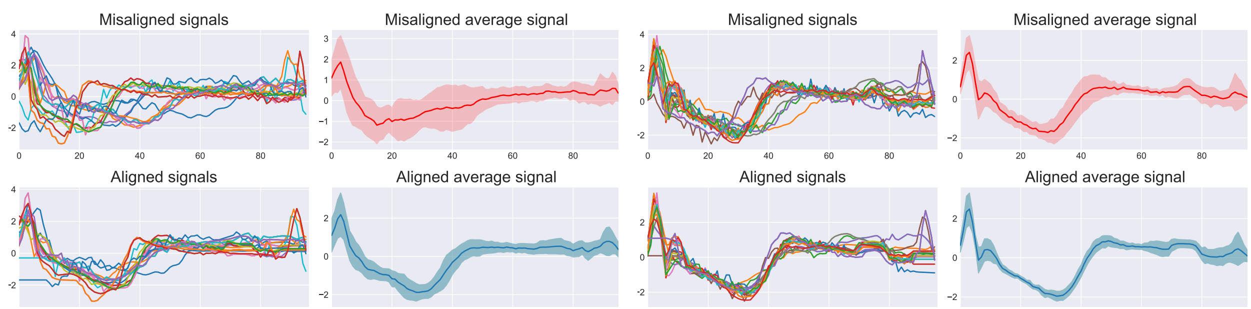
(a) Class 0 (b) Class 1

Figure 13. Joint alignment and averaging of the *ECGFiveDays* dataset. Shaded area corresponds to $\pm\sigma$.



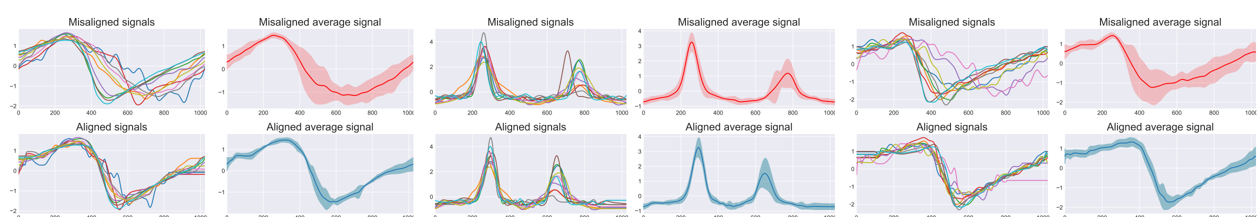
(a) Class 0 (b) Class 1 (c) Class 2

Figure 14. Joint alignment and averaging of the *CBF* dataset. Shaded area corresponds to $\pm\sigma$.



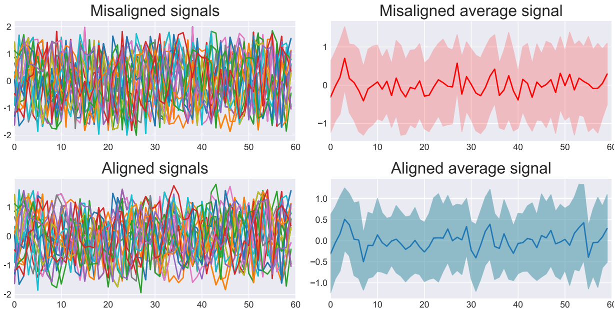
(a) Class 0 (b) Class 1

Figure 15. Joint alignment and averaging of the *ECG200* dataset. Shaded area corresponds to $\pm\sigma$.

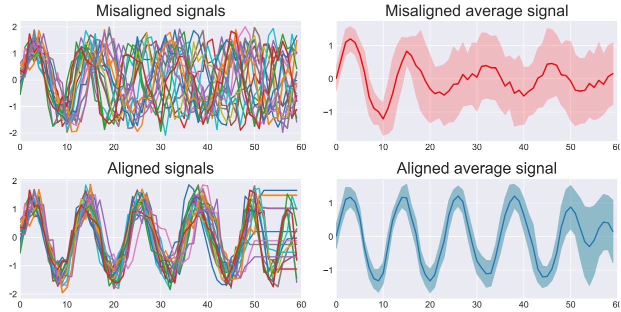


(a) Class 0 (b) Class 1 (c) Class 2

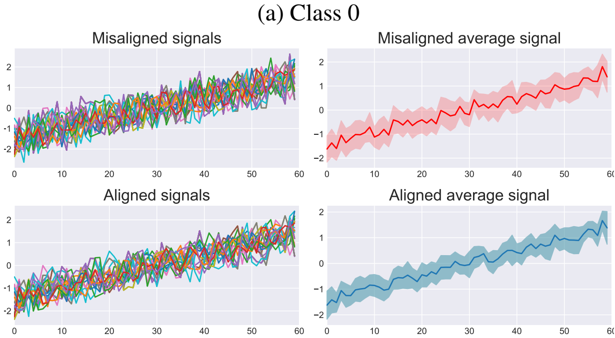
Figure 16. Joint alignment and averaging of the *StarLightCurves* dataset. Shaded area corresponds to $\pm\sigma$.



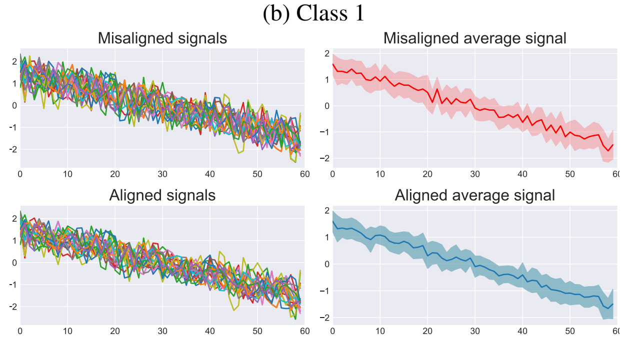
(a) Class 0



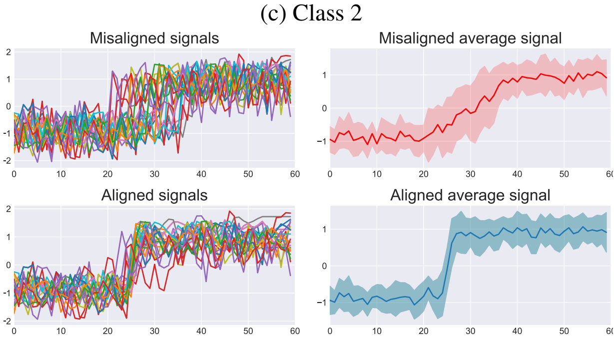
(b) Class 1



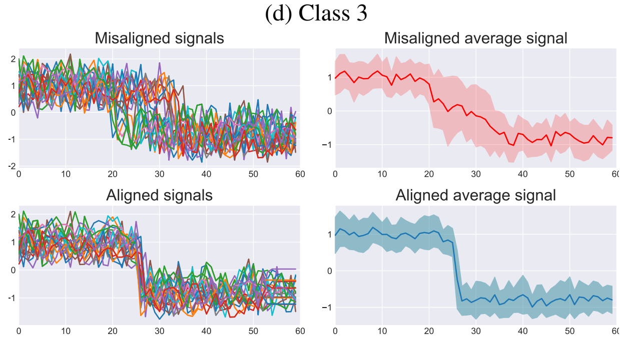
(c) Class 2



(d) Class 3

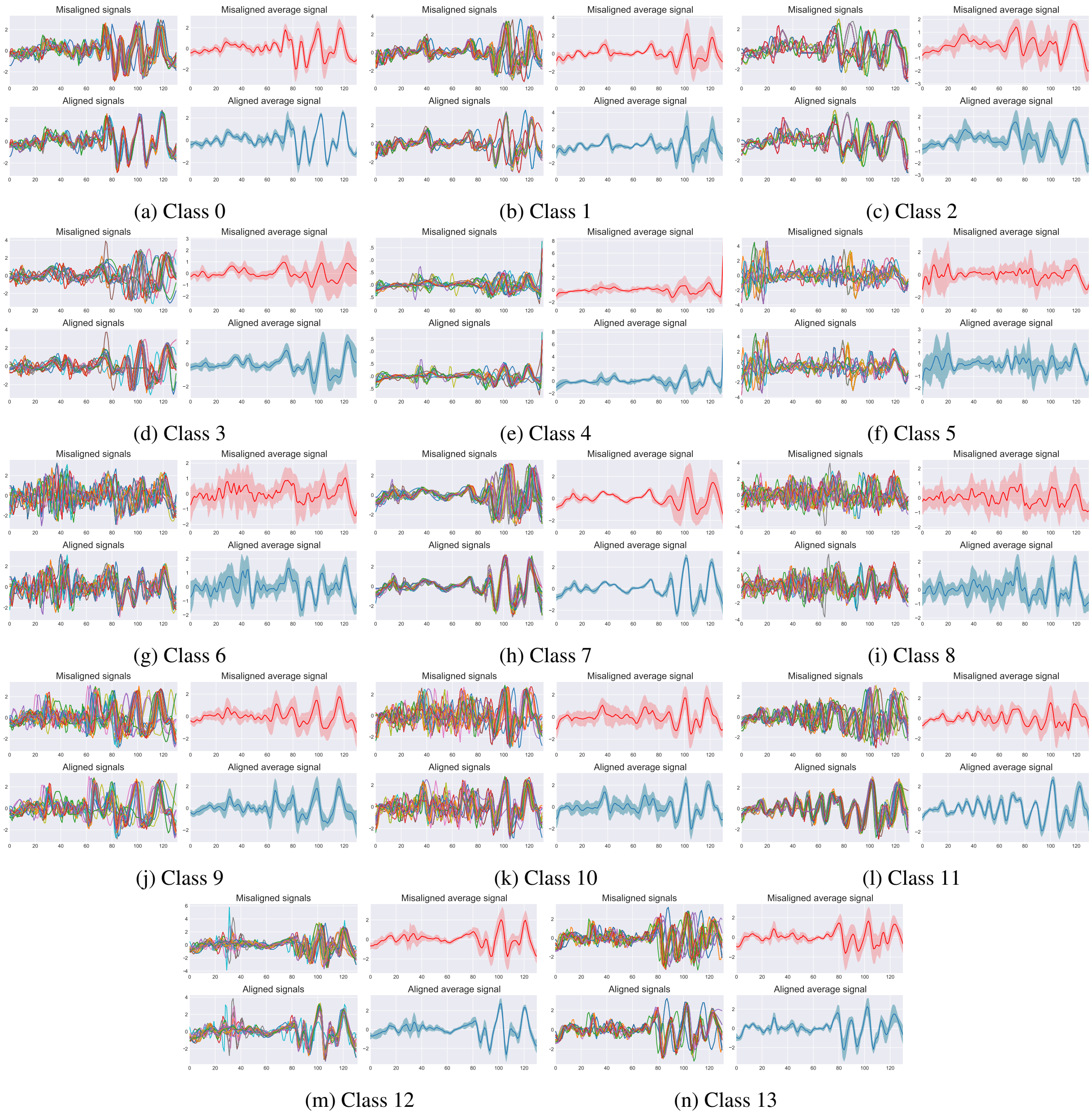


(e) Class 4



(f) Class 5

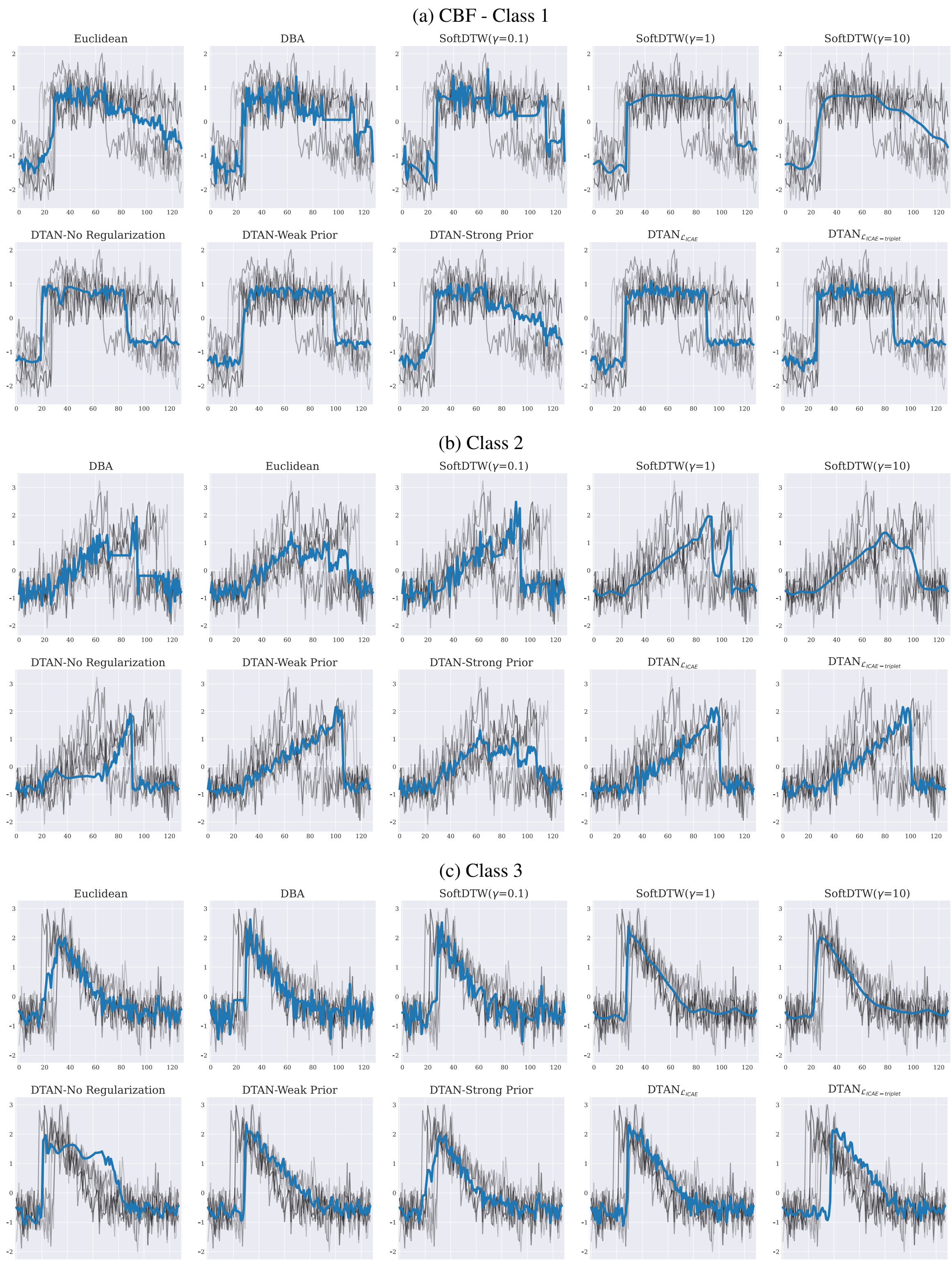
Figure 17. Joint alignment and averaging of the *SyntheticControl* dataset. Shaded area corresponds to $\pm\sigma$.



(a) Class 0 (b) Class 1 (c) Class 2 (d) Class 3 (e) Class 4 (f) Class 5 (g) Class 6 (h) Class 7 (i) Class 8 (j) Class 9 (k) Class 10 (l) Class 11 (m) Class 12 (n) Class 13

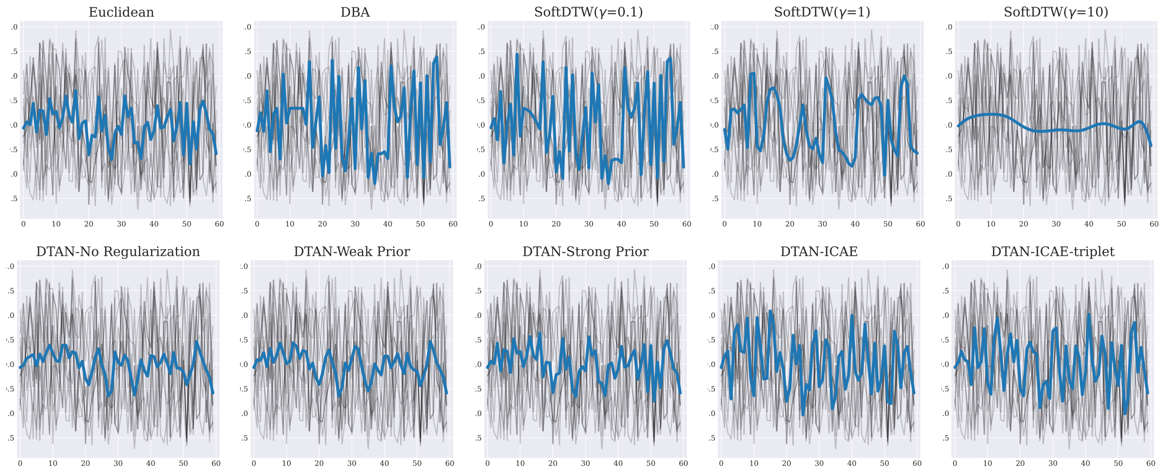
Figure 18. Joint alignment and averaging of the *FacesUCR* dataset. Shaded area corresponds to $\pm\sigma$.

G. Barycenters Comparison

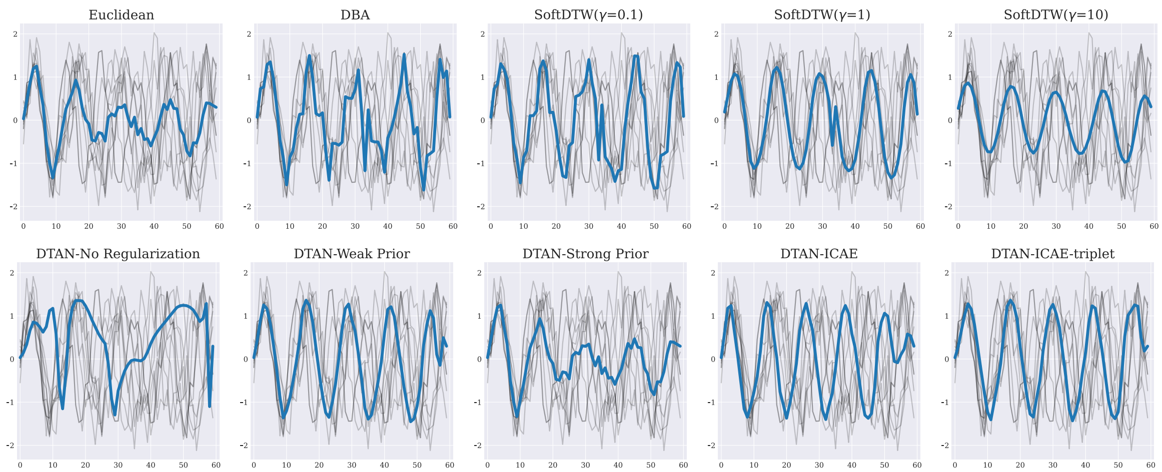


(d) The effect of regularization hyperparameters (HP) on barycenter computation. 10 samples of the CBF dataset and their mean (blue).

(a) SyntheticControl - Class 1

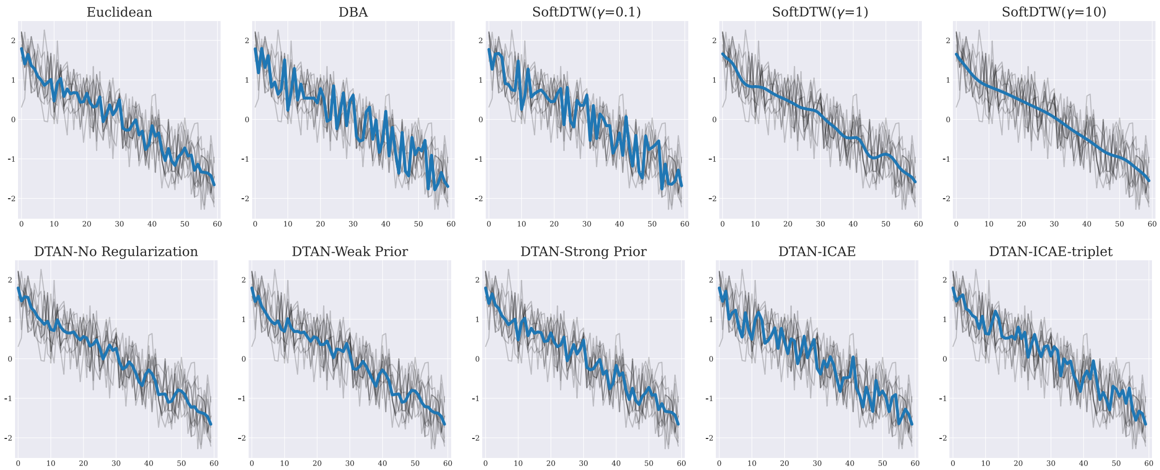


(b) Class 2

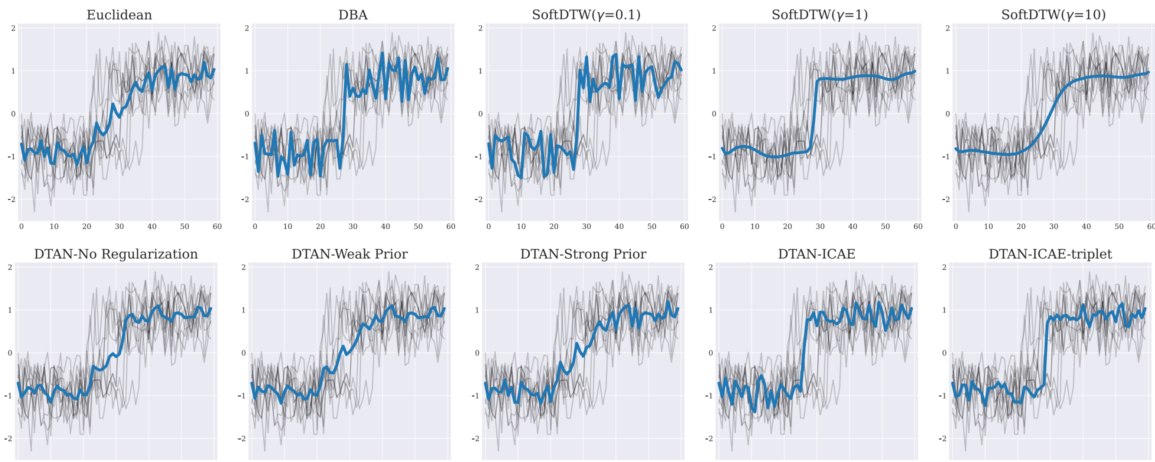


(c) Class 3

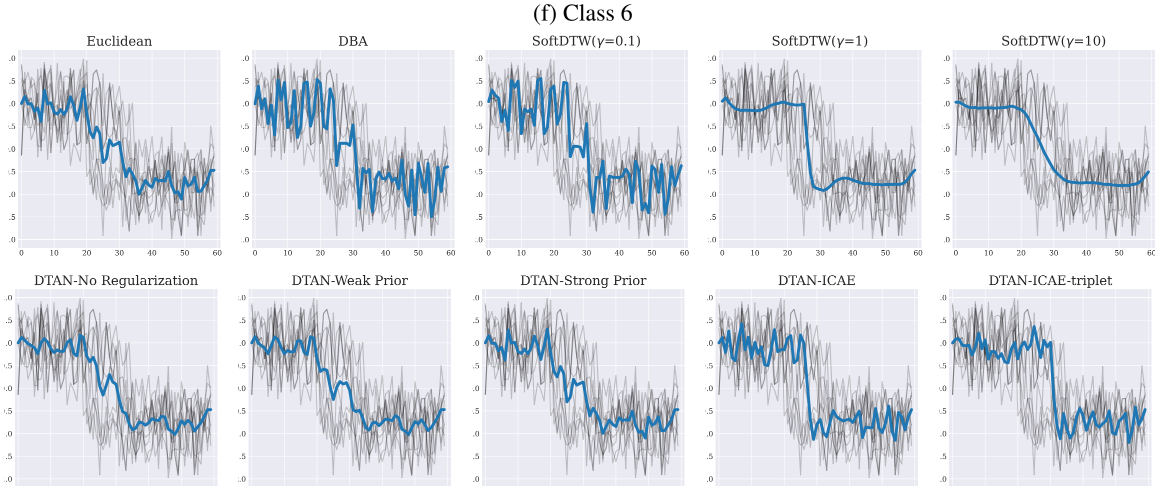
(d) SyntheticControl - Class 4



(e) Class 5

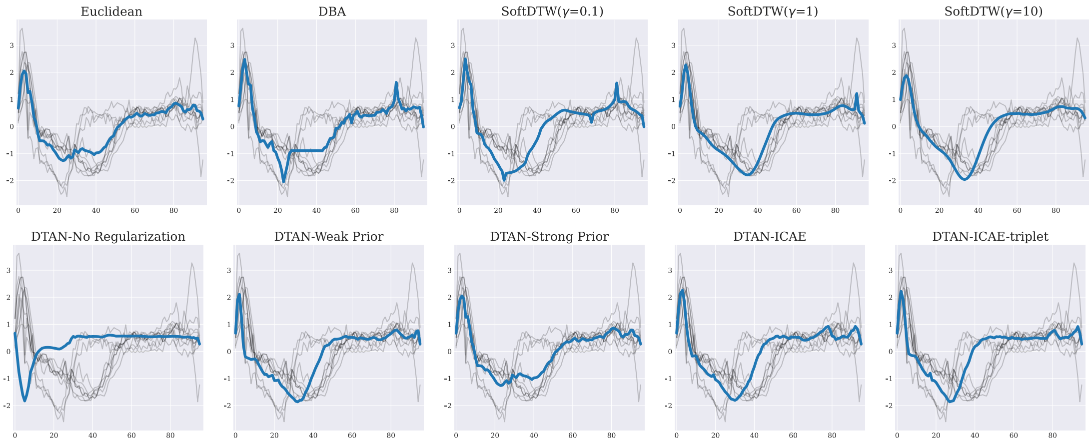


(f) Class 6

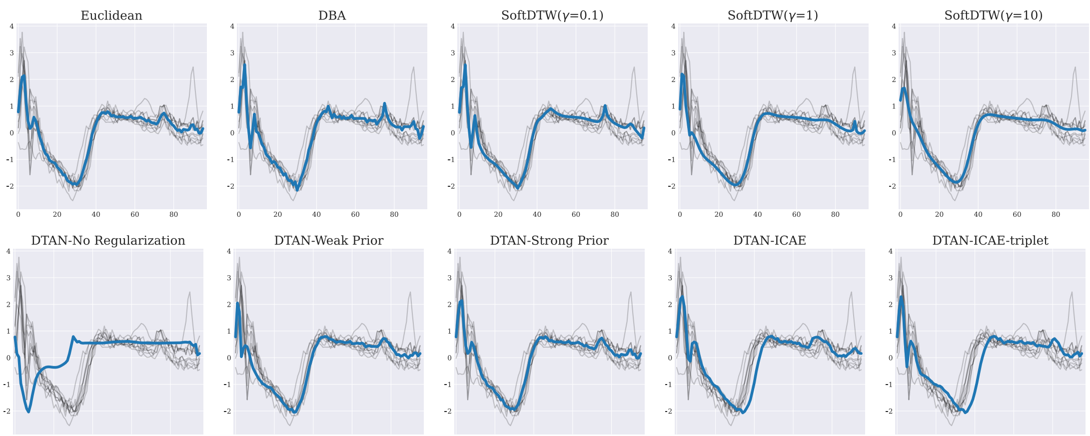


(g) The effect of regularization hyperparameters (HP) on barycenter computation. 10 samples of the SyntheticControl dataset and their mean (blue).

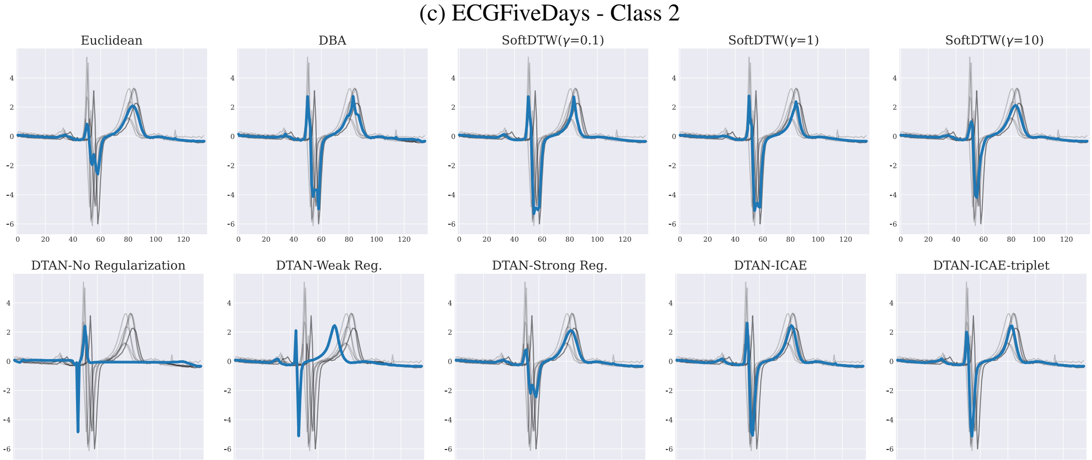
(a) ECG200 - Class 1



(b) Class 2



(c) ECGFiveDays - Class 2



H. UCR time series classification archive details

ID	Type	Name	Train	Test	Class	Length
1	Image	Adiac	390	391	37	176
2	Image	ArrowHead	36	175	3	251
3	Spectro	Beef	30	30	5	470
4	Image	BeetleFly	20	20	2	512
5	Image	BirdChicken	20	20	2	512
6	Sensor	Car	60	60	4	577
7	Simulated	CBF	30	900	3	128
8	Sensor	ChlorineConcentration	467	3840	3	166
9	Sensor	CinCECGTorso	40	1380	4	1639
10	Spectro	Coffee	28	28	2	286
11	Device	Computers	250	250	2	720
12	Motion	CricketX	390	390	12	300
13	Motion	CricketY	390	390	12	300
14	Motion	CricketZ	390	390	12	300
15	Image	DiatomSizeReduction	16	306	4	345
16	Image	DistalPhalanxOutlineAgeGroup	400	139	3	80
17	Image	DistalPhalanxOutlineCorrect	600	276	2	80
18	Image	DistalPhalanxTW	400	139	6	80
19	Sensor	Earthquakes	322	139	2	512
20	ECG	ECG200	100	100	2	96
21	ECG	ECG5000	500	4500	5	140
22	ECG	ECGFiveDays	23	861	2	136
23	Device	ElectricDevices	8926	7711	7	96
24	Image	FaceAll	560	1690	14	131
25	Image	FaceFour	24	88	4	350
26	Image	FacesUCR	200	2050	14	131
27	Image	FiftyWords	450	455	50	270
28	Image	Fish	175	175	7	463
29	Sensor	FordA	3601	1320	2	500
30	Sensor	FordB	3636	810	2	500
31	Motion	GunPoint	50	150	2	150
32	Spectro	Ham	109	105	2	431
33	Image	HandOutlines	1000	370	2	2709
34	Motion	Haptics	155	308	5	1092
35	Image	Herring	64	64	2	512
36	Motion	InlineSkate	100	550	7	1882
37	Sensor	InsectWingbeatSound	220	1980	11	256
38	Sensor	ItalyPowerDemand	67	1029	2	24
39	Device	LargeKitchenAppliances	375	375	3	720
40	Sensor	Lightning2	60	61	2	637
41	Sensor	Lightning7	70	73	7	319
42	Simulated	Mallat	55	2345	8	1024
43	Spectro	Meat	60	60	3	448
44	Image	MedicalImages	381	760	10	99
45	Image	MiddlePhalanxOutlineAgeGroup	400	154	3	80
46	Image	MiddlePhalanxOutlineCorrect	600	291	2	80
47	Image	MiddlePhalanxTW	399	154	6	80
48	Sensor	MoteStrain	20	1252	2	84
49	ECG	NonInvasiveFetalECGThorax1	1800	1965	42	750
50	ECG	NonInvasiveFetalECGThorax2	1800	1965	42	750
51	Spectro	OliveOil	30	30	4	570
52	Image	OSULeaf	200	242	6	427
53	Image	PhalangesOutlinesCorrect	1800	858	2	80
54	Sensor	Phoneme	214	1896	39	1024
55	Sensor	Plane	105	105	7	144
56	Image	ProximalPhalanxOutlineAgeGroup	400	205	3	80
57	Image	ProximalPhalanxOutlineCorrect	600	291	2	80
58	Image	ProximalPhalanxTW	400	205	6	80
59	Device	RefrigerationDevices	375	375	3	720
60	Device	ScreenType	375	375	3	720
61	Simulated	ShapeletSim	20	180	2	500

ID	Type	Name	Train	Test	Class	Length
62	Image	ShapesAll	600	600	60	512
63	Device	SmallKitchenAppliances	375	375	3	720
64	Sensor	SonyAIBORobotSurface1	20	601	2	70
65	Sensor	SonyAIBORobotSurface2	27	953	2	65
66	Sensor	StarLightCurves	1000	8236	3	1024
67	Spectro	Strawberry	613	370	2	235
68	Image	SwedishLeaf	500	625	15	128
69	Image	Symbols	25	995	6	398
70	Simulated	SyntheticControl	300	300	6	60
71	Motion	ToeSegmentation1	40	228	2	277
72	Motion	ToeSegmentation2	36	130	2	343
73	Sensor	Trace	100	100	4	275
74	ECG	TwoLeadECG	23	1139	2	82
75	Simulated	TwoPatterns	1000	4000	4	128
76	Motion	UWaveGestureLibraryAll	896	3582	8	945
77	Motion	UWaveGestureLibraryX	896	3582	8	315
78	Motion	UWaveGestureLibraryY	896	3582	8	315
79	Motion	UWaveGestureLibraryZ	896	3582	8	315
80	Sensor	Wafer	1000	6164	2	152
81	Spectro	Wine	57	54	2	234
82	Image	WordSynonyms	267	638	25	270
83	Motion	Worms	181	77	5	900
84	Motion	WormsTwoClass	181	77	2	900
85	Image	Yoga	300	3000	2	426
86	Device	ACSF1	100	100	10	1460
87	Sensor	AllGestureWiimoteX	300	700	10	Vary
88	Sensor	AllGestureWiimoteY	300	700	10	Vary
89	Sensor	AllGestureWiimoteZ	300	700	10	Vary
90	Simulated	BME	30	150	3	128
91	Traffic	Chinatown	20	343	2	24
92	Image	Crop	7200	16800	24	46
93	Sensor	DodgerLoopDay	78	80	7	288
94	Sensor	DodgerLoopGame	20	138	2	288
95	Sensor	DodgerLoopWeekend	20	138	2	288
96	EOG	EOGHorizontalSignal	362	362	12	1250
97	EOG	EOGVerticalSignal	362	362	12	1250
98	Spectro	EthanolLevel	504	500	4	1751
99	Sensor	FreezerRegularTrain	150	2850	2	301
100	Sensor	FreezerSmallTrain	28	2850	2	301
101	HRM	Fungi	18	186	18	201
102	Trajectory	GestureMidAirD1	208	130	26	Vary
103	Trajectory	GestureMidAirD2	208	130	26	Vary
104	Trajectory	GestureMidAirD3	208	130	26	Vary
105	Sensor	GesturePebbleZ1	132	172	6	Vary
106	Sensor	GesturePebbleZ2	146	158	6	Vary
107	Motion	GunPointAgeSpan	135	316	2	150
108	Motion	GunPointMaleVersusFemale	135	316	2	150
109	Motion	GunPointOldVersusYoung	136	315	2	150
110	Device	HouseTwenty	40	119	2	2000
111	EPG	InsectEPGRegularTrain	62	249	3	601
112	EPG	InsectEPGSmallTrain	17	249	3	601
113	Traffic	MelbournePedestrian	1194	2439	10	24
114	Image	MixedShapesRegularTrain	500	2425	5	1024
115	Image	MixedShapesSmallTrain	100	2425	5	1024
116	Sensor	PickupGestureWiimoteZ	50	50	10	Vary
117	Hemodynamics	PigAirwayPressure	104	208	52	2000
118	Hemodynamics	PigArtPressure	104	208	52	2000
119	Hemodynamics	PigCVP	104	208	52	2000
120	Device	PLAID	537	537	11	Vary
121	Power	PowerCons	180	180	2	144
122	Spectrum	Rock	20	50	4	2844
123	Spectrum	SemgHandGenderCh2	300	600	2	1500
124	Spectrum	SemgHandMovementCh2	450	450	6	1500

ID	Type	Name	Train	Test	Class	Length
125	Spectrum	SemgHandSubjectCh2	450	450	5	1500
126	Sensor	ShakeGestureWiimoteZ	50	50	10	Vary
127	Simulated	SmoothSubspace	150	150	3	15
128	Simulated	UMD	36	144	3	150

I. UCR Nearest Centroid Classification (NCC) Results

I.1. Comparison with results reported in the literature (84 datasets (Chen et al., 2015))

Table 6: NCC results for 84 datasets of the UCR archive. Comparison between our $\mathcal{L}_{\mathrm{ICAE}}$ and $\mathcal{L}_{\mathrm{ICAE-triplet}}$ (titled $\mathcal{L}_{\mathrm{triplet}}$ due to space limitations; median and best results across 5 runs) and various joint alignment and barycenter computation methods in terms of NCC accuracy. Euclidean (Euc.), DBA, SoftDTW (SDTW), and SoftDTW Divergence (SDTW-div) results are taken from (Blondel et al., 2021), ResNet-TW from (Huang et al., 2021), DTAN$_{\mathrm{libcpab}}$ from (Shapira Weber et al., 2019) and DTAN$_{\mathrm{DIFW}}$ from (Martinez et al., 2022).

Dataset	**Euc.**	**DTW**	**SDTW**	**SDTW div**	**DTAN** libcpab	**ResNet-TW**	**DTAN** DIFW	**Median** $\mathcal{L}_{\mathrm{ICAE}}$	$\mathcal{L}_{\mathrm{triplet}}$	**Best** $\mathcal{L}_{\mathrm{ICAE}}$	$\mathcal{L}_{\mathrm{triplet}}$
adiac	0.550	0.471	0.675	0.685	0.696	0.698	0.719	0.696	0.752	0.703	**0.775**
arrowhead	0.611	0.509	0.514	0.577	0.749	0.754	0.726	0.737	0.783	0.754	**0.846**
beef	0.533	0.433	0.467	0.367	0.633	0.633	0.700	0.567	**0.733**	0.600	**0.733**
beetlefly	0.850	0.800	0.700	0.700	0.800	0.800	**0.950**	0.700	0.600	0.850	0.650
birdchicken	0.550	0.600	0.650	0.600	0.800	**0.950**	**0.950**	0.600	0.800	0.750	0.900
car	0.617	0.617	0.700	0.733	0.817	**1.000**	0.989	0.783	0.833	0.833	0.883
cbf	0.763	0.969	0.971	0.971	0.914	0.850	0.982	0.961	0.847	**0.993**	0.857
chlorineconcentration	0.333	0.325	0.352	0.322	0.333	0.352	0.397	0.324	0.779	0.325	**0.812**
cincecgtorso	0.385	0.403	0.719	0.704	0.616	0.543	**0.741**	0.445	0.521	0.514	0.550
coffee	0.964	0.964	0.964	0.964	**1.000**	0.964	**1.000**	0.964	**1.000**	0.964	**1.000**
computers	0.416	0.632	0.516	0.568	0.592	**0.676**	0.616	0.468	0.448	0.480	0.520
cricketx	0.239	**0.577**	0.569	0.567	0.423	0.341	0.428	0.474	0.482	0.526	0.518
crickety	0.349	0.526	0.556	0.549	0.541	0.415	0.513	0.562	0.600	0.572	**0.641**
cricketz	0.305	0.600	**0.610**	0.600	0.421	0.333	0.451	0.518	0.474	0.544	0.556
diatomsizereduction	0.958	0.951	0.967	0.964	0.971	0.974	0.984	0.971	0.974	**0.987**	0.977
distalphalanxoutlineagegroup	0.818	0.840	0.845	0.848	0.848	**0.863**	0.748	0.719	0.712	0.727	0.727
distalphalanxoutlinecorrect	0.472	0.482	0.480	0.473	0.472	0.505	0.775	0.493	0.775	0.518	**0.793**
distalphalanxtw	0.748	0.757	0.745	0.745	0.780	**0.797**	0.683	0.626	0.619	0.647	0.633
earthquakes	0.755	0.581	0.823	0.652	0.773	**0.973**	0.820	0.698	0.683	0.719	0.698
ecg200	0.750	0.750	0.720	0.730	0.790	0.795	0.914	0.790	0.900	0.830	**0.920**
ecg5000	0.860	0.845	0.867	0.860	0.891	0.800	**0.999**	0.854	0.907	0.855	0.912
ecgfivedays	0.690	0.653	0.806	0.834	0.978	0.932	**0.993**	0.859	0.791	0.922	0.947
electricdevices	0.483	0.536	0.571	**0.616**	0.535	0.519	0.574	0.521	0.427	0.549	0.508
faceall	0.492	0.807	0.816	**0.886**	0.805	0.841	0.856	0.738	0.744	0.782	0.825
facefour	0.841	0.830	0.864	0.898	0.830	0.855	**0.920**	0.773	0.830	0.841	0.864
facesucr	0.539	0.792	0.890	**0.911**	0.857	0.857	0.801	0.808	0.808	0.896	0.886
fiftywords	0.516	0.598	0.763	**0.780**	0.653	0.516	0.631	0.609	0.587	0.611	0.622
fish	0.560	0.657	0.811	0.840	0.903	0.903	**0.914**	0.829	0.891	0.891	0.909
forda	0.496	0.556	0.556	0.524	0.605	0.568	0.652	0.574	0.669	0.604	**0.855**
fordb	0.500	0.607	0.476	0.559	0.580	0.566	0.546	0.499	0.515	0.531	**0.623**
gunpoint	0.753	0.680	0.820	0.813	0.880	0.807	0.847	0.913	0.967	0.933	**0.973**
ham	0.762	0.733	0.714	0.752	0.790	0.762	**0.810**	0.790	0.752	0.800	0.790
handoutlines	0.818	0.792	0.824	nan	0.850	0.835	0.908	0.773	0.938	0.800	**0.949**
haptics	0.393	0.357	0.461	0.461	0.458	0.464	**0.487**	0.419	0.377	0.435	0.403
herring	0.547	0.609	0.641	0.641	0.703	0.766	**0.781**	0.625	0.609	0.672	0.656
inlineskate	0.193	0.227	0.234	0.264	0.260	0.244	**0.287**	0.205	0.233	0.242	0.271
insectwingbeatsound	0.601	0.298	0.582	0.586	0.587	0.571	**0.607**	0.533	0.517	0.554	0.536
italypowerdemand	0.918	0.742	0.881	0.905	0.962	0.965	**0.967**	0.939	0.955	0.950	0.964
largekitchenappliances	0.440	0.715	0.720	**0.736**	0.483	0.501	0.517	0.392	0.408	0.421	0.435
lightning2	0.688	0.623	0.672	0.721	0.721	**0.754**	0.738	0.557	0.672	0.623	0.689
lightning7	0.589	0.726	0.781	**0.836**	0.712	0.685	0.726	0.562	0.562	0.562	0.589
mallat	0.967	0.949	0.957	0.948	0.969	0.967	**0.974**	0.957	0.957	0.965	0.959
meat	0.933	0.933	0.850	0.850	0.933	0.933	0.933	**0.933**	0.883	**0.933**	0.917
medicalimages	0.385	0.442	0.404	0.409	0.468	0.474	0.483	0.479	0.563	0.521	**0.613**
middlephalanxoutlineagegroup	0.733	0.725	0.728	0.728	0.738	**0.752**	0.636	0.604	0.578	0.610	0.604
middlephalanxoutlinecorrect	0.552	0.485	0.522	0.528	0.543	0.532	0.698	0.656	0.801	0.670	**0.835**
middlephalanxtw	0.592	0.566	0.582	0.582	0.596	**0.634**	0.539	0.487	0.552	0.506	0.552
motestrain	0.861	0.824	0.904	0.902	0.904	**0.913**	0.875	0.843	0.855	0.857	0.890
noninvasivefetalecgthorax1	0.770	0.701	0.816	0.823	0.853	0.839	0.874	0.844	0.926	0.855	**0.934**
noninvasivefetalecgthorax2	0.802	0.763	0.872	0.877	0.905	0.839	0.917	0.889	0.949	0.891	**0.950**
oliveoil	0.867	0.767	0.833	0.867	0.867	0.867	**0.900**	0.833	0.700	0.867	0.800

osuleaf	0.360	0.459	0.521	0.512	0.463	0.459	**0.933**	0.409	0.426	0.426	0.512
phalangesoutlinescorrect	0.626	0.636	0.637	0.645	0.642	0.663	0.676	0.652	0.837	0.656	**0.845**
phoneme	0.079	0.177	0.201	**0.206**	0.102	0.117	0.101	0.088	0.083	0.093	0.090
plane	0.962	0.990	0.990	0.990	**1.000**	**1.000**	**1.000**	0.981	0.981	**1.000**	**1.000**
proximalphalanxoutlineagegroup	0.820	0.829	0.844	0.844	0.854	**0.873**	**0.873**	0.849	0.844	0.854	0.844
proximalphalanxoutlinecorrect	0.646	0.650	0.650	0.650	0.643	0.687	0.725	0.643	0.911	0.643	**0.928**
proximalphalanxtw	0.708	0.735	0.812	0.815	0.818	**0.823**	0.790	0.756	0.766	0.766	0.780
refrigerationdevices	0.355	0.579	**0.581**	0.552	0.467	0.483	0.485	0.331	0.339	0.339	0.365
screentype	0.443	0.381	0.373	0.400	0.445	0.469	0.461	0.443	0.408	**0.472**	0.459
shapeletsim	0.500	0.617	**0.733**	0.728	0.539	0.589	0.572	0.461	0.483	0.539	0.533
shapesall	0.513	0.622	0.655	**0.687**	0.628	0.682	0.643	0.565	0.577	0.578	0.587
smallkitchenappliances	0.419	0.645	0.680	**0.688**	0.621	0.560	0.592	0.429	0.400	0.435	0.419
sonyaiborobotsurface1	0.812	0.829	0.827	0.829	**0.894**	0.860	0.892	0.699	0.725	0.734	0.742
sonyaiborobotsurface2	0.793	0.766	0.798	0.765	0.811	0.830	**0.875**	0.790	0.826	0.817	0.831
strawberry	0.669	0.612	0.656	0.688	0.843	0.786	0.892	0.654	0.976	0.676	**0.981**
swedishleaf	0.702	0.704	0.794	0.811	0.806	0.837	0.858	0.798	0.827	0.843	**0.862**
symbols	0.864	**0.958**	0.951	0.956	0.857	0.907	0.912	0.865	0.860	0.885	0.882
syntheticcontrol	0.917	0.983	0.980	0.987	0.950	0.950	0.980	0.970	0.983	0.990	**0.993**
toesegmentation1	0.575	0.627	0.733	0.711	0.640	0.654	**0.794**	0.583	0.583	0.618	0.610
toesegmentation2	0.546	**0.869**	0.862	0.854	0.754	0.746	0.785	0.569	0.592	0.669	0.700
trace	0.580	0.980	0.980	0.970	0.780	0.800	0.980	0.780	**1.000**	0.960	**1.000**
twoleadecg	0.555	0.762	0.780	0.831	0.956	0.955	0.989	0.908	0.985	0.942	**0.994**
twopatterns	0.465	0.984	0.987	0.982	0.556	0.701	0.716	0.988	0.999	1.000	**1.000**
uwavegesturelibraryall	0.850	0.835	0.893	0.909	0.921	0.912	**0.944**	0.895	0.887	0.903	0.913
uwavegesturelibraryx	0.631	0.700	0.680	0.697	0.681	**0.722**	0.710	0.685	0.680	0.694	0.697
uwavegesturelibraryy	0.548	0.532	0.613	0.621	0.612	0.617	0.641	0.630	0.628	**0.643**	0.642
uwavegesturelibraryz	0.537	0.606	0.633	0.645	0.642	0.646	**0.652**	0.627	0.617	0.634	0.631
wafer	0.654	0.319	0.688	0.689	0.989	0.983	0.986	0.976	0.993	0.978	**0.997**
wine	0.556	0.537	0.574	0.556	0.574	0.593	**0.833**	0.556	0.778	0.556	0.815
wordsynonyms	0.271	0.343	**0.522**	0.517	0.475	0.502	0.475	0.433	0.414	0.458	0.425
worms	0.215	0.403	0.436	**0.448**	0.260	0.343	0.338	0.351	0.338	0.351	0.351
wormstwoclass	0.541	0.630	0.680	**0.707**	0.619	0.619	0.649	0.494	0.558	0.532	0.571
yoga	0.497	0.600	0.571	0.617	0.632	0.697	0.681	0.620	0.825	0.628	**0.838**

I.2. Comparison between objective functions (128 datasets (Dau et al., 2019))

Table 7: NCC results for 128 datasets of the UCR archive (including ones containing variable length time series). Comparison between our $\mathcal{L}_{\mathrm{ICAE}}$ and $\mathcal{L}_{\mathrm{ICAE-triplet}}$ (titled $\mathcal{L}_{\mathrm{triplet}}$ due to space limitations) and the standard Within-Class Sum of Squares (WCSS) w/o regularization prior. Otherwise, all other parameters, including f_{loc}, are identical. Prior values are set to $\lambda_{\Sigma} = 0.001$ and $\lambda_{\mathrm{smooth}} = 0.1$. Median and best results across 5 runs. N/A in the results refers to datasets containing signals of variable length .

Dataset	Median				Best			
	WCSS	WCSS+reg	$\mathcal{L}_{\mathrm{ICAE}}$	$\mathcal{L}_{\mathrm{triplet}}$	WCSS	WCSS+reg	$\mathcal{L}_{\mathrm{ICAE}}$	$\mathcal{L}_{\mathrm{triplet}}$
acsf1	0.410	0.640	0.500	0.810	0.560	0.700	0.580	**0.830**
adiac	0.660	0.550	0.696	0.752	0.668	0.550	0.703	**0.775**
allgesturewiimotex	N/A	N/A	0.250	0.197	N/A	N/A	**0.307**	0.209
allgesturewiimotey	N/A	N/A	0.390	0.193	N/A	N/A	**0.501**	0.261
allgesturewiimotez	N/A	N/A	0.096	0.126	N/A	N/A	0.129	**0.140**
arrowhead	0.640	0.611	0.737	0.783	0.691	0.611	0.754	**0.846**
beef	0.500	0.533	0.567	**0.733**	0.567	0.533	0.600	**0.733**
beetlefly	0.700	**0.850**	0.700	0.600	0.800	**0.850**	**0.850**	0.650
birdchicken	0.750	0.550	0.600	0.800	**0.950**	0.550	0.750	0.900
bme	0.853	0.647	0.907	0.933	0.880	0.647	0.967	**0.980**
car	0.683	0.617	0.783	0.833	0.800	0.617	0.833	**0.883**
cbf	0.782	0.762	0.961	0.847	0.990	0.762	**0.993**	0.857
chinatown	0.959	0.959	0.980	0.980	0.965	0.959	**0.983**	**0.983**
chlorineconcentration	0.318	0.331	0.324	0.779	0.323	0.333	0.325	**0.812**
cincecgtorso	0.324	0.407	0.445	0.521	0.372	0.408	0.514	**0.550**
coffee	**1.000**	0.964	0.964	**1.000**	**1.000**	0.964	0.964	**1.000**
computers	0.560	0.412	0.468	0.448	**0.640**	0.416	0.480	0.520
cricketx	0.156	0.241	0.474	0.482	0.190	0.241	**0.526**	0.518
crickety	0.174	0.349	0.562	0.600	0.251	0.354	0.572	**0.641**
cricketz	0.195	0.303	0.518	0.474	0.233	0.305	0.544	**0.556**
crop	0.492	0.472	0.549	0.657	0.526	0.472	0.556	**0.658**
diatomsizereduction	0.954	0.958	0.971	0.974	0.958	0.958	**0.987**	0.977
distalphalanxoutlineagegroup	0.719	0.698	0.719	0.712	**0.727**	0.698	**0.727**	**0.727**
distalphalanxoutlinecorrect	0.645	0.688	0.493	0.775	0.663	0.688	0.518	**0.793**
distalphalanxtw	0.612	0.576	0.626	0.619	0.633	0.583	**0.647**	0.633
dodgerloopday	0.450	0.463	0.475	0.438	0.450	0.463	**0.487**	**0.487**
dodgerloopgame	0.812	0.812	0.812	0.826	0.812	0.812	**0.841**	**0.841**
dodgerloopweekend	**0.986**	**0.986**	**0.986**	**0.986**	**0.986**	**0.986**	**0.986**	**0.986**
earthquakes	0.719	0.669	0.698	0.683	**0.763**	0.683	0.719	0.698
ecg200	0.880	0.750	0.790	0.900	0.910	0.750	0.830	**0.920**
ecg5000	0.597	0.860	0.854	0.907	0.614	0.861	0.855	**0.912**
ecgfivedays	0.886	0.747	0.859	0.791	0.908	0.785	0.922	**0.947**
electricdevices	0.342	0.487	0.521	0.427	0.371	0.487	**0.549**	0.508
eoghorizontalsignal	0.213	0.359	0.425	0.434	0.235	0.359	0.428	**0.478**
eogverticalsignal	0.243	0.279	0.304	0.309	0.246	0.279	0.337	**0.365**
ethanollevel	0.262	0.284	0.314	0.834	0.328	0.284	0.316	**0.842**
faceall	0.799	0.495	0.738	0.744	**0.876**	0.496	0.782	0.825
facefour	0.761	0.830	0.773	0.830	0.784	0.841	0.841	**0.864**
facesucr	0.737	0.548	0.808	0.808	0.856	0.549	**0.896**	0.886
fiftywords	0.035	0.516	0.609	0.587	0.116	0.516	0.611	**0.622**
fish	0.680	0.560	0.829	0.891	0.697	0.566	0.891	**0.909**
forda	0.515	0.501	0.574	0.669	0.527	0.504	0.604	**0.855**
fordb	0.486	0.502	0.499	0.515	0.516	0.504	0.531	**0.623**
freezerregulartrain	0.793	0.769	0.768	0.993	0.942	0.769	0.776	**0.995**
freezersmalltrain	0.769	0.763	0.791	0.806	0.782	0.763	0.815	**0.881**
fungi	0.823	0.823	0.823	0.823	**0.828**	0.823	**0.828**	**0.828**
gesturemidaird1	N/A	N/A	0.569	0.608	N/A	N/A	0.600	**0.631**
gesturemidaird2	N/A	N/A	0.562	0.531	N/A	N/A	**0.585**	0.554
gesturemidaird3	N/A	N/A	0.354	0.354	N/A	N/A	0.385	**0.400**
gesturepebblez1	N/A	N/A	0.192	0.192	N/A	N/A	**0.203**	**0.203**
gesturepebblez2	N/A	N/A	0.234	0.228	N/A	N/A	**0.297**	0.285
gunpoint	0.933	0.753	0.913	0.967	0.967	0.753	0.933	**0.973**
gunpointagespan	0.892	0.854	0.642	0.981	0.978	0.854	0.668	**0.987**

Table 7: NCC results for 128 datasets of the UCR archive (including ones containing variable length time series). Comparison between our $\mathcal{L}_{\text{ICAE}}$ and $\mathcal{L}_{\text{ICAE-triplet}}$ (titled $\mathcal{L}_{\text{triplet}}$ due to space limitations) and the standard Within-Class Sum of Squares (WCSS) w/o regularization prior. Otherwise, all other parameters, including f_{loc}, are identical. Prior values are set to $\lambda_{\Sigma} = 0.001$ and $\lambda_{\text{smooth}} = 0.1$. Median and best results across 5 runs. N/A in the results refers to datasets containing signals of variable length .

Dataset	**Median**				**Best**			
	WCSS	WCSS+reg	$\mathcal{L}_{\text{ICAE}}$	$\mathcal{L}_{\text{triplet}}$	WCSS	WCSS+reg	$\mathcal{L}_{\text{ICAE}}$	$\mathcal{L}_{\text{triplet}}$
gunpointmaleversusfemale	0.956	0.690	0.965	**1.000**	0.959	0.690	0.968	**1.000**
gunpointoldversusyoung	0.511	0.775	0.670	0.981	0.565	0.775	0.705	**0.987**
ham	0.667	0.762	0.790	0.752	0.752	0.762	**0.800**	0.790
handoutlines	0.778	0.819	0.773	0.938	0.819	0.819	0.800	**0.949**
haptics	0.364	0.399	0.419	0.377	0.396	0.399	**0.435**	0.403
herring	0.578	0.547	0.625	0.609	**0.672**	0.547	**0.672**	0.656
housetwenty	0.706	0.756	0.706	0.689	0.723	**0.765**	**0.765**	0.714
inlineskate	0.209	0.195	0.205	0.233	0.224	0.196	0.242	**0.271**
insectepgregulartrain	0.622	0.482	0.586	0.719	0.635	0.490	0.671	**0.727**
insectepgsmalltrain	0.618	0.586	0.683	0.651	0.663	0.586	**0.747**	0.695
insectwingbeatsound	0.318	0.604	0.533	0.517	0.364	**0.605**	0.554	0.536
italypowerdemand	0.934	0.920	0.939	0.955	0.948	0.920	0.950	**0.964**
largekitchenappliances	0.456	0.443	0.392	0.408	**0.488**	0.443	0.421	0.435
lightning2	0.689	0.672	0.557	0.672	**0.705**	0.689	0.623	0.689
lightning7	0.671	0.575	0.562	0.562	**0.726**	0.603	0.562	0.589
mallat	0.922	**0.967**	0.957	0.957	0.950	**0.967**	0.965	0.959
meat	0.917	0.933	0.933	0.883	**0.950**	0.933	0.933	0.917
medicalimages	0.261	0.386	0.479	0.563	0.284	0.387	0.521	**0.613**
melbournepedestrian	0.789	0.609	0.733	0.839	0.795	0.609	0.743	**0.845**
middlephalanxoutlineagegroup	0.591	0.571	0.604	0.578	0.597	0.571	**0.610**	0.604
middlephalanxoutlinecorrect	0.608	0.478	0.656	0.801	0.612	0.478	0.670	**0.835**
middlephalanxtw	0.448	0.442	0.487	**0.552**	0.500	0.442	0.506	**0.552**
mixedshapesregulartrain	0.791	0.731	0.839	0.845	0.805	0.731	0.843	**0.851**
mixedshapessmalltrain	0.729	0.729	0.779	0.802	0.773	0.729	0.800	**0.812**
motestrain	0.844	0.861	0.843	0.855	0.850	0.862	0.857	**0.890**
noninvasivefetalecgthorax1	0.737	0.770	0.844	0.926	0.749	0.770	0.855	**0.934**
noninvasivefetalecgthorax2	0.827	0.803	0.889	0.949	0.835	0.803	0.891	**0.950**
oliveoil	0.833	**0.867**	0.833	0.700	**0.867**	**0.867**	**0.867**	0.800
osuleaf	0.360	0.364	0.409	0.426	0.459	0.364	0.426	**0.512**
phalangesoutlinescorrect	0.613	0.626	0.652	0.837	0.628	0.626	0.656	**0.845**
phoneme	0.080	0.080	0.088	0.083	**0.094**	0.082	0.093	0.090
pickupgesturewiimotez	N/A	N/A	0.080	**0.120**	N/A	N/A	0.100	**0.120**
pigairwaypressure	0.019	0.005	0.005	0.024	0.029	0.010	**0.038**	**0.038**
pigartpressure	0.154	0.096	0.197	0.159	0.173	0.096	**0.231**	0.212
pigcvp	0.053	0.038	0.053	0.048	**0.072**	0.038	0.053	0.048
plaid	N/A	N/A	0.069	0.019	N/A	N/A	**0.076**	0.032
plane	**1.000**	0.962	0.981	0.981	**1.000**	0.962	**1.000**	**1.000**
powercons	0.783	0.861	0.889	0.928	0.867	0.861	0.911	**0.944**
proximalphalanxoutlineagegroup	0.839	0.820	0.849	0.844	0.844	0.820	**0.854**	0.844
proximalphalanxoutlinecorrect	0.643	0.646	0.643	0.911	0.643	0.646	0.643	**0.928**
proximalphalanxtw	0.741	0.698	0.756	0.766	0.756	0.698	0.766	**0.780**
refrigerationdevices	0.352	0.355	0.331	0.339	**0.384**	0.365	0.339	0.365
rock	0.660	0.620	0.540	0.780	0.680	0.620	0.620	**0.860**
screentype	0.395	0.443	0.443	0.408	0.397	0.445	**0.472**	0.459
semghandgenderch2	0.655	0.688	0.692	0.833	0.683	0.688	0.697	**0.893**
semghandmovementch2	0.380	0.393	0.393	0.369	0.407	0.411	0.400	**0.467**
semghandsubjectch2	0.556	0.560	0.560	0.638	0.624	0.567	0.580	**0.664**
shakegesturewiimotez	N/A	N/A	0.120	0.160	N/A	N/A	0.160	**0.200**
shapeletsim	0.511	0.494	0.461	0.483	**0.561**	0.517	0.539	0.533
shapesall	0.457	0.513	0.565	0.577	0.470	0.513	0.578	**0.587**
smallkitchenappliances	0.467	0.437	0.429	0.400	**0.547**	0.456	0.435	0.419
smoothsubspace	0.713	0.707	0.713	0.700	**0.873**	0.707	0.747	0.807
sonyaiborobotsurface1	0.754	0.815	0.699	0.725	0.772	**0.822**	0.734	0.742
sonyaiborobotsurface2	0.801	0.792	0.790	0.826	0.812	0.793	0.817	**0.831**
starlightcurves	0.830	0.762	0.845	0.897	0.853	0.762	0.875	**0.938**

Table 7: NCC results for 128 datasets of the UCR archive (including ones containing variable length time series). Comparison between our $\mathcal{L}_{\text{ICAE}}$ and $\mathcal{L}_{\text{ICAE}-\text{triplet}}$ (titled $\mathcal{L}_{\text{triplet}}$ due to space limitations) and the standard Within-Class Sum of Squares (WCSS) w/o regularization prior. Otherwise, all other parameters, including f_{loc}, are identical. Prior values are set to $\lambda_{\Sigma} = 0.001$ and $\lambda_{\text{smooth}} = 0.1$. Median and best results across 5 runs. N/A in the results refers to datasets containing signals of variable length .

Dataset	**Median**				**Best**			
	WCSS	WCSS+reg	$\mathcal{L}_{\text{ICAE}}$	$\mathcal{L}_{\text{triplet}}$	WCSS	WCSS+reg	$\mathcal{L}_{\text{ICAE}}$	$\mathcal{L}_{\text{triplet}}$
strawberry	0.651	0.584	0.654	0.976	0.686	0.584	0.676	**0.981**
swedishleaf	0.770	0.704	0.798	0.827	0.786	0.706	0.843	**0.862**
symbols	0.836	0.865	0.865	0.860	0.848	0.865	**0.885**	0.882
syntheticcontrol	0.943	0.920	0.970	0.983	0.987	0.920	0.990	**0.993**
toesegmentation1	0.583	0.575	0.583	0.583	0.605	0.579	**0.618**	0.610
toesegmentation2	0.608	0.554	0.569	0.592	**0.746**	0.554	0.669	0.700
trace	0.760	0.580	0.780	**1.000**	0.930	0.580	0.960	**1.000**
twoleadecg	0.917	0.556	0.908	0.985	0.930	0.556	0.942	**0.994**
twopatterns	0.256	0.464	0.988	0.999	0.260	0.465	1.000	**1.000**
umd	0.778	0.542	0.806	0.910	0.861	0.542	**0.979**	0.958
uwavegesturelibraryall	0.723	0.850	0.895	0.887	0.762	0.850	0.903	**0.913**
uwavegesturelibraryx	0.595	0.631	0.685	0.680	0.599	0.631	0.694	**0.697**
uwavegesturelibraryy	0.536	0.549	0.630	0.628	0.574	0.549	**0.643**	0.642
uwavegesturelibraryz	0.475	0.538	0.627	0.617	0.545	0.538	**0.634**	0.631
wafer	0.769	0.655	0.976	0.993	0.801	0.655	0.978	**0.997**
wine	0.574	0.556	0.556	0.778	0.630	0.556	0.556	**0.815**
wordsynonyms	0.155	0.271	0.433	0.414	0.183	0.271	**0.458**	0.425
worms	0.273	0.208	**0.351**	0.338	**0.351**	0.208	**0.351**	**0.351**
wormstwoclass	0.468	0.532	0.494	0.558	0.558	0.532	0.532	**0.571**
yoga	0.664	0.497	0.620	0.825	0.683	0.497	0.628	**0.838**