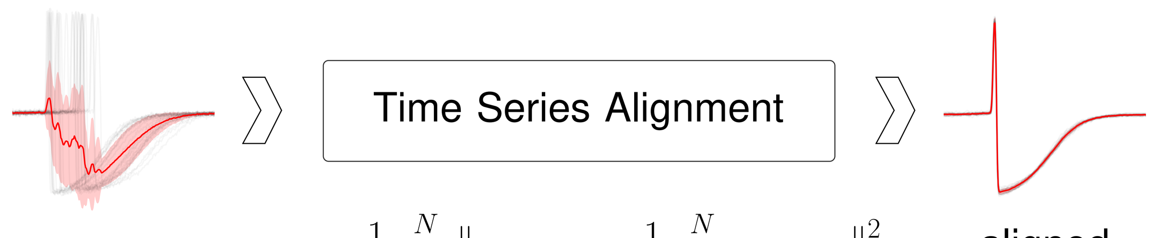
Closed-Form Diffeomorphic Transformations for Time Series Alignment

Iñigo Martinez [1] Elisabeth Viles [2,3] Igor G. Olaizola [1]

[1]Vicomtech, Basque Research and Technology Alliance, Spain [2]TECNUN School of Engineering, University of Navarra, Spain [3]Institute of Data Science and Artificial Intelligence, University of Navarra, Spain

Motivation

- Temporal misalignment impacts recognition and classification performance
- Optimization problem: set of warping functions ϕ that minimize temporal variability
- Preference for highly expressive, differentiable and invertible warping functions



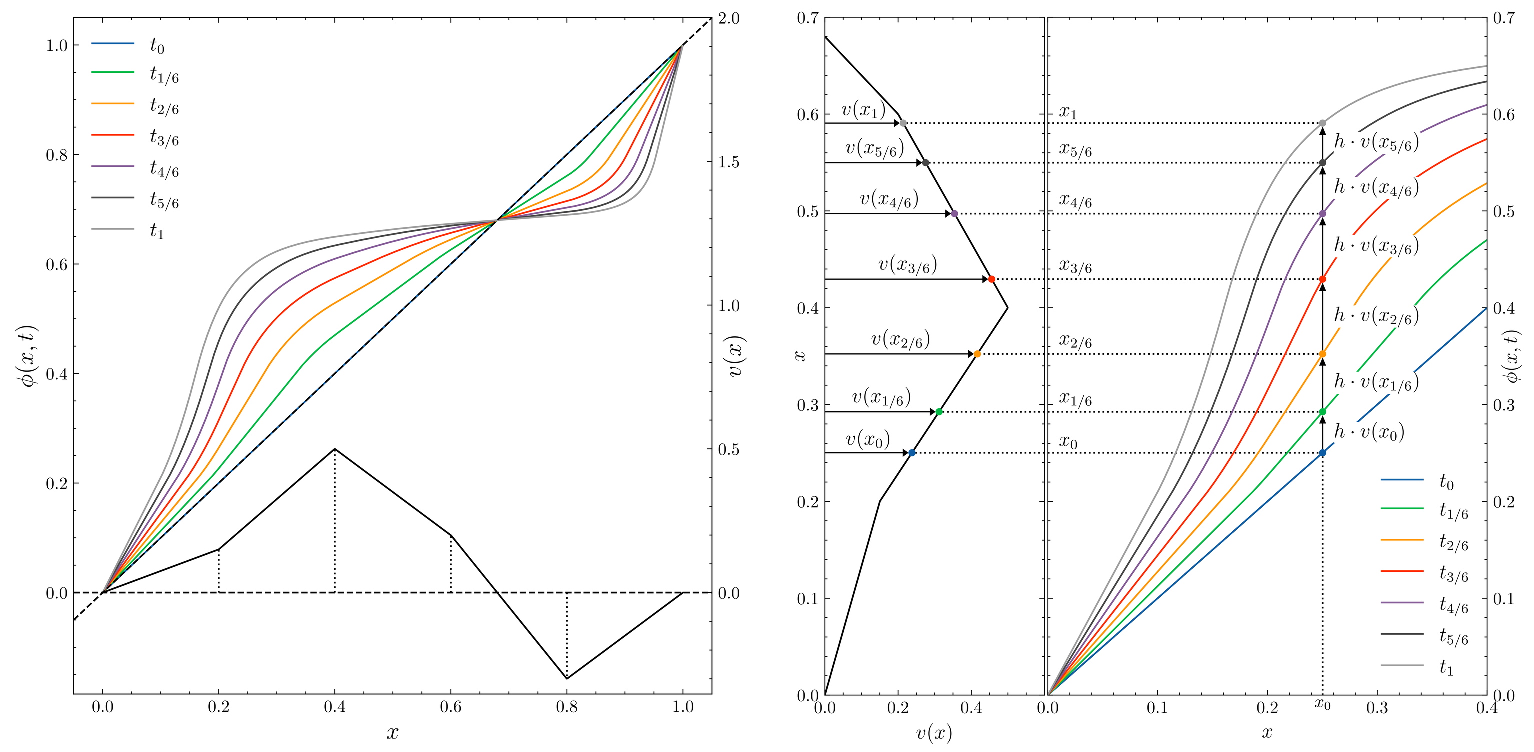
$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left\| \mathbf{y}_i \circ \phi_i - \frac{1}{N}\sum_{j=1}^{N} \mathbf{y}_j \circ \phi_j \right\|_2^2$$

Diffeomorphic Warping Functions

- Generated via integration of stationary or time-dependent velocity fields specified by an ODE with initial condition $\phi(x, 0) = x$ or the equivalent integral equation:

$$\frac{d\phi^\theta(x,t)}{dt} = v^\theta(\phi^\theta(x,t)) \longleftrightarrow \phi^\theta(x,t) = x + \int_0^t v^\theta(\phi^\theta(x,\tau))d\tau$$

x : temporal dimension
t : integration time
v : velocity function
θ : parameters



(a) Continuous piecewise affine velocity function $v(x)$. Flow integration with initial condition $\phi(x, t_0) = x$.

(b) Flow numerical integration: $\phi(x, t + h) = \phi(x, t) + h \cdot v(x)$. Produces increasing functions (invertible 1D functions)

Figure 1. Construction of diffeomorphic curves by integration of velocity functions.

- Problem: Neural networks that include diffeomorphic transformations require to calculate derivatives to the ODE's solution with respect to the model parameters θ.

$$\frac{d\phi^\theta(x,t)}{dt} = v^\theta(\phi^\theta(x,t)) \longrightarrow \frac{\partial\phi^\theta(x,t)}{\partial\theta_k} = ??$$

- Current solutions
  - Adjoint sensitivity methods
  - ResNet's Eulerian discretization
  - Ad-hoc numerical solvers + autodiff

Closed-form Diffeomorphic ODE Solution & Gradient

- We formulate a closed-form expression for the gradient of 1D diffeomorphic transformations under continuous piecewise-affine (CPA) velocity functions.
  - Closed-form ODE solution (forward) $\phi^\theta(x,t)$
  - Closed-form gradient (backward) $\partial\phi^\theta(x,t)/\partial\theta_k$

$$\phi^\theta(x,t) = \left(\psi_{\theta,c_m}^{t_m} \circ \psi_{\theta,c_{m-1}}^{t_{m-1}} \circ \cdots \circ \psi_{\theta,c_2}^{t_2} \circ \psi_{\theta,c_1}^{t_1}\right)(x)$$

$$\phi^\theta(x,t) = \psi^\theta(x = x_m, t = t_m) = \left(x e^{t a_c} + \left(e^{t a_c} - 1\right)\frac{b_c}{a_c}\right)\Bigg|_{\substack{x=x_m\\t=t_m}}$$

- A closed-form solution provides efficiency, speed and precision.
  - Fast computation for iterative gradient descent methods
  - Exact gradient leads to better solutions at convergence
  - Shortens chain of operations and decreases tape overhead

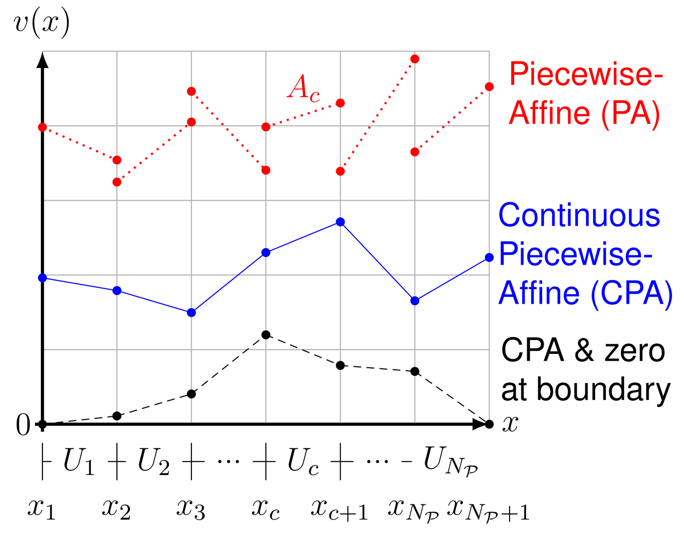


Figure 2. Velocity functions $v(x)$. Each cell U_c in the tessellation $\mathcal{P}$ defines an affine transformation $\mathbf{A}_c = [a_c \; b_c] \in \mathbb{R}^{1\times 2}$.

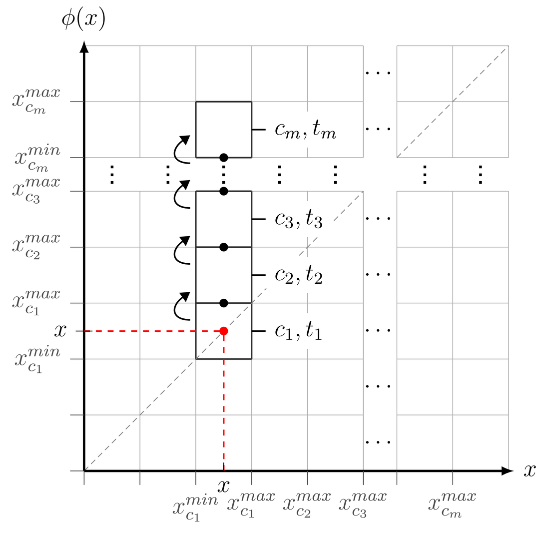


Figure 3. Iterative process of integration: starting at initial cell c_1 and time $t_1 = 1$, several cells are crossed and the process finishes at cell c_m and time t_m.

Results: Performance

- *DIFW* library: optimized implementation of 1D diffeomorphic transformations for CPU (*NumPy and PyTorch with C++*) and GPU (*PyTorch with CUDA*)
- Our closed form method is compared to libcpab numeric solution implementation [3] which is based on [4]
- Speed tests: **x18** / **x260** and **x10** / **x30** improvement on **CPU** / **GPU** over current solutions for forward and backward operations respectively.

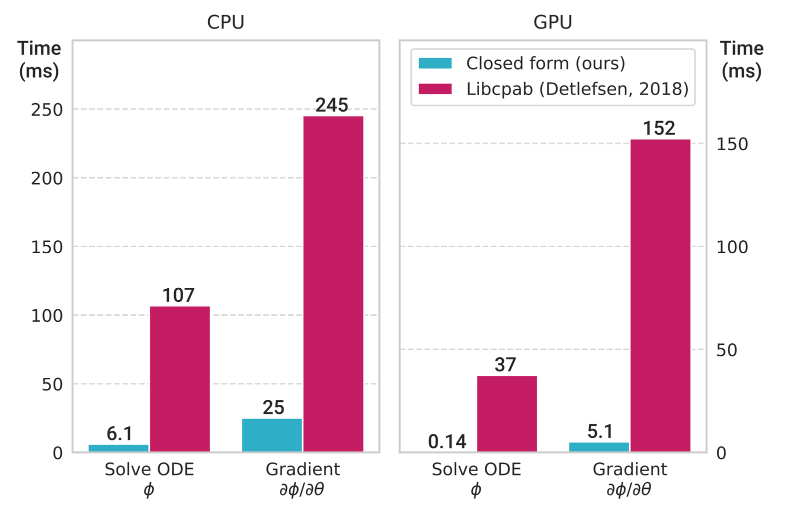


Figure 4. Computation time (ms) for the forward and backward operations on CPU (left) and GPU (right).

Results: Time Series Classification

- Diffeomorphic Temporal Transformer Network for time series alignment and classification
- Data: UCR time-series classification archive, includes 85 real-world datasets [2]
- Benchmark: Nearest Centroid Classifier (NCC). Compared with Euclidean averaging, DTW Barycenter Averaging (DBA) [6], SoftDTW [1], DTAN [7] and ResNet-TW [5]
- Results show significant improvement in terms of efficiency and accuracy

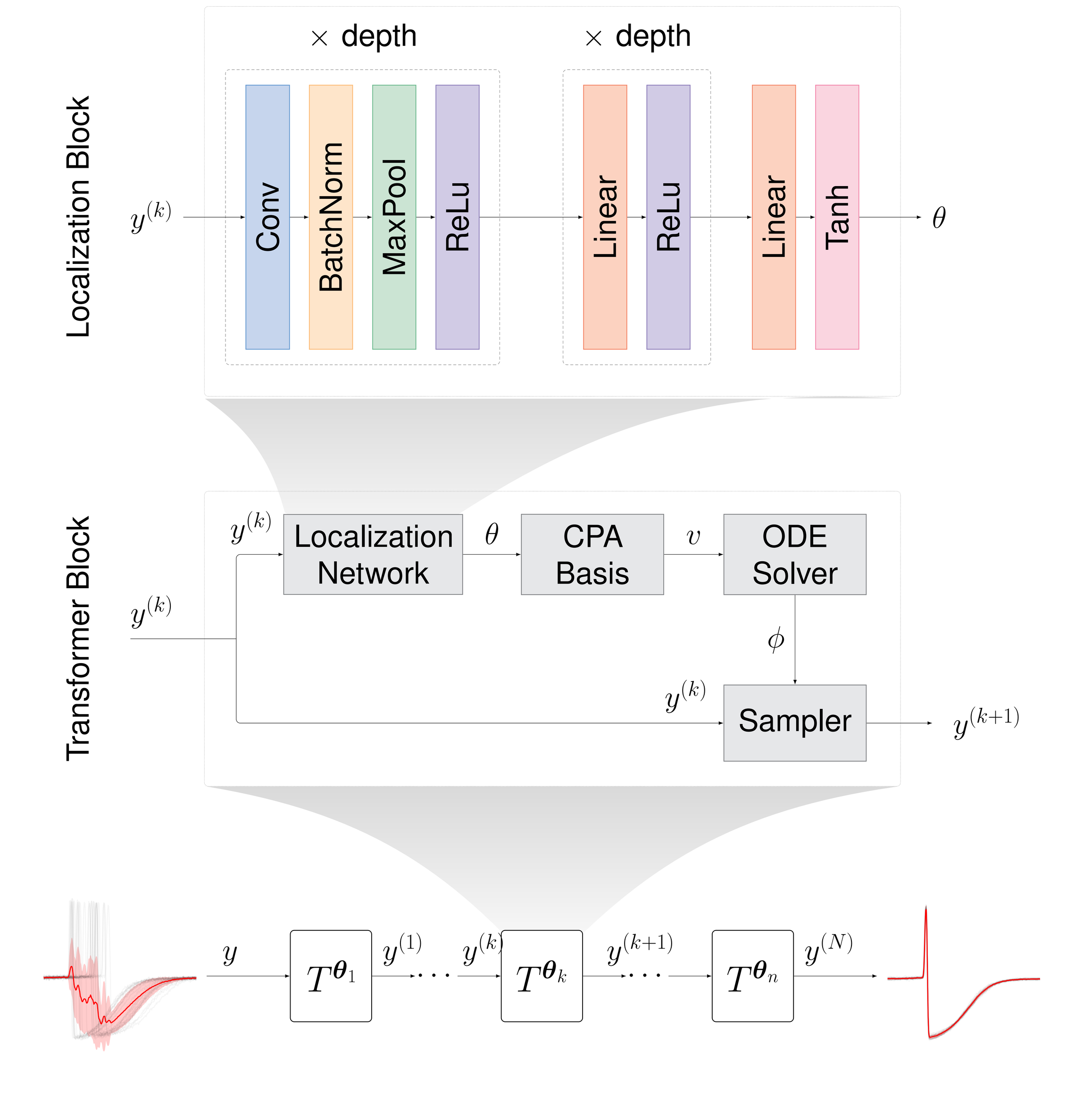


Figure 5. Temporal transformer architecture. **Bottom**: time series y is aligned by applying sequence of transformers T^θ that minimize the empirical variance of warped signals. **Middle**: Transformed data is sampled based on diffeomorphic flow ϕ obtained via integration of velocity function v. **Top**: parameters θ are computed by the localization network based on $y^{(k)}$.

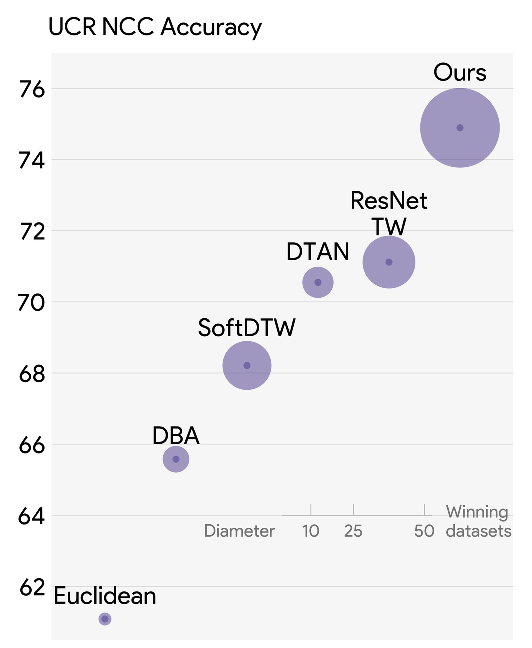


Figure 6. Correct classification rates using NCC on UCR archive [2]. Radius denotes the number of datasets in which each method achieved top accuracy.

$N_\mathcal{P}$ Cells	Training Time	NCC Accuracy
4 → 16	▲ 1.5%	▲ 0.69%
16 → 32	▲ 1.8%	▼ 0.09%
32 → 64	▲ 5.5%	▼ 0.20%
64 → 128	▲ 16.3%	▼ 0.24%
128 → 256	▲ 25.0%	▼ 0.48%

Table 1. Impact of higher expressive CPA functions on training time and NCC

Summary

- A novel closed-form expression for the gradient of CPA-based 1D diffeomorphic transformations, providing efficiency, speed and precision.
- *DIFW*: optimized implementation of 1D diffeomorphic transformations (CPU & GPU).
- Diffeomorphic temporal transformer network resembling [7] for time-series alignment.
- Experiments on 85 datasets from UCR archive [2] show significant improvements both in terms of efficiency and accuracy.

References

[1] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International Conference on Machine Learning*, pages 894–903. PMLR, 2017.

[2] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR Time Series Archive, 2019.

[3] Nicki S. Detlefsen. libcpab. `https://github.com/SkafteNicki/libcpab`, 2018.

[4] Oren Freifeld, Soren Hauberg, Kayhan Batmanghelich, and Jonn W. Fisher. Transformations Based on Continuous Piecewise-Affine Velocity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2496–2509, 2017.

[5] Hao Huang, Boulbaba Ben Amor, Xichan Lin, Fan Zhu, and Yi Fang. Residual Networks as Flows of Velocity Fields for Diffeomorphic Time Series Alignment. 2021.

[6] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[7] Ron Shapira Weber, Matan Eyal, Nicki Skafte Detlefsen, Oren Shriki, and Oren Freifeld. Diffeomorphic temporal alignment nets. *Advances in Neural Information Processing Systems*, 32(NeurIPS), 2019.

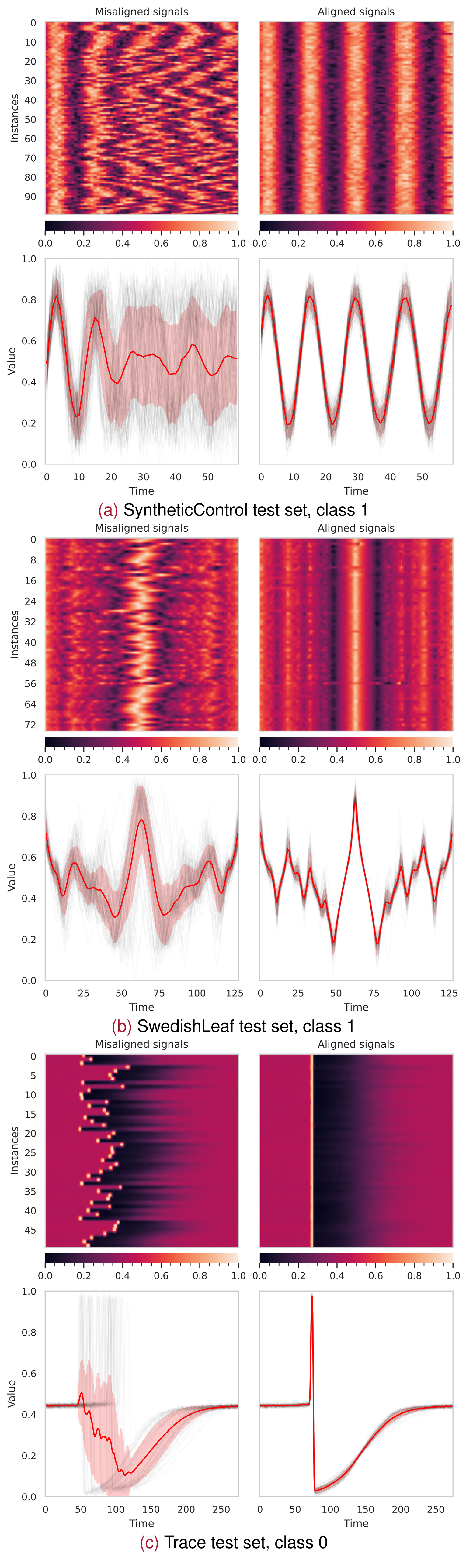


Figure 7. Multi-class time series alignment. **Top**: heatmap of each time series sample (row). **Bottom**: overlapping time series, red line represents Euclidean average. **Left**: original signals. **Right**: after alignment.