FUNCTIONAL AUTOENCODERS FOR AMPLITUDE-PHASE REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Functional data are intrinsically infinite-dimensional, and often exhibit phase variation, where corresponding events occur at different times across observations. Existing linear dimension reduction methods struggle with nonlinear amplitude variation, while functional autoencoders without an explicit warp entangle temporal misalignment with shape. We propose the **Amplitude–Phase Functional Autoencoders (AP-FAE)**, an unsupervised framework for functional data that spans both univariate and multivariate cases, with emphasis on the multivariate setting, and factorizes the latent space into separate amplitude and phase embeddings derived from all channels. A smooth functional decoder reconstructs channel-specific amplitude functions in canonical time, and a shared monotone, endpoint-preserving warp captures phase variation. We prove a bound linking amplitude recovery to registration, reconstruction, and noise errors, and validate it numerically. Across synthetic data and six real-world benchmarks, AP-FAE outperforms state-of-the-art baselines on most clustering and alignment metrics and on all reconstruction metrics. Clustering with amplitude embeddings alone consistently surpasses joint amplitude–phase clustering, confirming the benefit of explicit disentanglement. Code is available at https://anonymous.4open.science/r/APFAE-418C/.

1 INTRODUCTION

Functional data are functions over a continuous domain, such as physiological signals, growth trajectories, or motion recordings (Ramsay & Silverman, 2005; Ferraty & Vieu, 2006). In the multivariate setting, each observation is a vector-valued function $\mathbf{x}_i(t) = (x_{i1}(t), \ldots, x_{iD}(t))^\top$ with D synchronized channels. For example, in UWaveGestureLibrary, a hand-held accelerometer records each gesture along the x-, y-, and z-axes, yielding three synchronized channels (Figure 1). Functional data analysis (FDA) exploits smoothness and dependence structure for estimation, dimension reduction, and learning (Wang et al., 2016). As sensing technologies produce increasingly rich multichannel recordings, functional representation learning becomes important for converting complex trajectories into compact features (Wang et al., 2024).

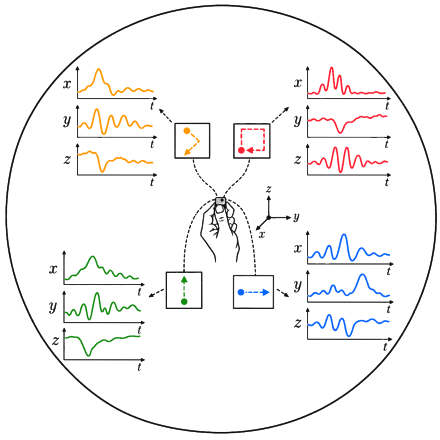


Figure 1: Three synchronized accelerometer channels for four hand gestures (schematic).

A first challenge is phase variation: corresponding events may occur at different times, so curves with similar shapes can appear different (Marron et al., 2015). Ignoring this can make representations reflect timing rather than shape (Kneip & Ramsay, 2008; Tang & Müller, 2008; Tucker et al., 2013). Traditional registration estimates time transformations before modeling (Srivastava et al., 2011), but alignment is then separate from the learning objective, and errors can propagate. Joint approaches such as DeepFRC (Jiang et al., 2026), and NeuralFLoC (Xiong et al., 2026) learn alignment with a modeling

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

objective, but their representations are tied to specific tasks and do not necessarily provide separate compact amplitude and phase embeddings within a reconstruction-based learner.

A second challenge is the intrinsic infinite-dimensionality of functional data. Functional principal component analysis (FPCA) addresses this by projecting onto a low-dimensional linear subspace (Ramsay & Silverman, 2005; Wang et al., 2016), but linearity can be restrictive for nonlinear variation, and channel-wise FPCA ignores cross-channel covariance. Multivariate FPCA (MFPCA) provides joint components across channels but remains linear (Happ & Greven, 2018). Functional autoencoders (FAEs) offer a nonlinear alternative with smooth functional reconstruction (Hsieh et al., 2021; Wu et al., 2024), and a joint multivariate encoder can aggregate cross-channel information. Without an explicit warping mechanism to handle phase variation, however, FAEs must entangle temporal misalignment with amplitude variation.

To address these challenges, we propose the **Amplitude–Phase Functional Autoencoders (AP-FAE)**, an unsupervised framework for temporally misaligned functional data that applies to both univariate and multivariate settings, with a primary focus on the multivariate case. A functional encoder jointly processes all channels and separates the latent representation into amplitude and phase embeddings. The amplitude decoder produces channel-specific trajectories in canonical time, while the phase decoder generates a monotone, endpoint-preserving warp shared across channels. AP-FAE is trained purely by reconstruction, so the learned embeddings are not tied to any downstream task. The amplitude and phase pathways are explicitly disentangled: the amplitude embedding captures shape variation, and the phase embedding captures temporal deformation. This design enables flexible downstream use, such as clustering on amplitude codes alone or regression with both embeddings.

Our main contributions are summarized as follows:

- To the best of our knowledge, we propose the first unsupervised framework that jointly learns nonlinear functional representations and explicitly disentangles amplitude and phase variation through separate embeddings and a shared monotone warp, with a primary focus on the multivariate setting.
- We establish the first theoretical bound quantifying how registration, reconstruction, and noise errors jointly control amplitude recovery in nonlinear functional representation learning, with numerical validation on synthetic data.
- We show that AP-FAE outperforms state-of-the-art baselines on most clustering and alignment metrics and on all reconstruction metrics across synthetic and six real-world benchmarks. Notably, on real data, clustering with amplitude embeddings alone consistently surpasses clustering with joint amplitude–phase embeddings, empirically confirming the benefit of explicit disentanglement for temporally misaligned functional data.

2 Related Work

Functional alignment and amplitude–phase separation. Registration/Alignment estimates time transformations to separate amplitude from phase (Marron et al., 2015). Fisher–Rao and square-root velocity frameworks provide geometric elastic alignment (Srivastava et al., 2011), and neural methods learn registration mappings (Chen & Srivastava, 2021). When used as preprocessing, alignment is separate from representation learning. Joint methods integrate alignment with clustering or classification. For example, JPCCA (Gaffney & Smyth, 2004) combines probabilistic alignment with model-based curve clustering, SRC (Zeng et al., 2019) jointly models registration and clustering, DeepFRC (Jiang et al., 2026) connects registration to supervised classification, and NeuralFLoC (Xiong et al., 2026) uses neural flows for joint registration and clustering. These methods demonstrate the benefits of integrating alignment with a modeling objective, but their representations are tied to specific tasks. AP-FAE instead combines separate amplitude and phase embeddings with smooth multivariate functional decoding under a reconstruction-only objective. Clustering is applied to amplitude embeddings after fitting and does not influence alignment.

Linear functional dimension reduction. FPCA provides optimal linear dimension reduction under squared-error loss (Ramsay & Silverman, 2005; Wang et al., 2016), but may require many components for nonlinear or misaligned curves, and channel-wise FPCA omits cross-channel covariance.

MFPCA extends to vector-valued observations with within- and cross-component covariance (Happ & Greven, 2018), giving a principled multichannel basis. However, standard MFPCA is linear and lacks an explicit subject-specific warp. Nonlinear extensions of FPCA exist but often rely on kernel or manifold assumptions(Chen & Müller, 2012; Song & Li, 2021; Tan et al., 2024), and they do not jointly model temporal deformation. AP-FAE instead learns nonlinear amplitude and phase embeddings under a reconstruction objective, providing a flexible alternative to linear subspace methods.

Autoencoders and functional autoencoders. Autoencoders learn low-dimensional representations through a reconstruction bottleneck (Hinton & Salakhutdinov, 2006). FAEs impose functional smoothness via basis representations of filters and decoded trajectories (Hsieh et al., 2021; Wu et al., 2024). Extending functional projections across channels enables joint multivariate encoding (Chiou et al., 2014; Happ & Greven, 2018), but without explicit temporal deformation, temporal shifts consume latent capacity and mix with shape variation (Tucker et al., 2013; Marron et al., 2015). Some autoencoder variants incorporate temporal regularization or recurrent structures (Srivastava et al., 2015; Malhotra et al., 2016; Sagheer & Kotb, 2019), but they do not separate amplitude from phase. AP-FAE retains the smooth functional encoder–decoder while adding a separate phase code and warping network, so temporal variation is modeled explicitly rather than entangled with shape.

3 The AP-FAE Framework

Overview. We consider D-channel functional data $\mathbf{x}_i(t) = (x_{i1}(t), \ldots, x_{iD}(t))^\top$, where $x_{id} \in L^2([0,1])$, $t \in [0,1]$, and $i = 1, \ldots, N$. We assume the curves exhibit alignment/phase variation, which is common in practice (as shown in the real-world data in Appendix C.9), and model them as

$$x_{id}(t) = f_{id}\big(\omega_i(t)\big) + \epsilon_{id}(t), \qquad d = 1, \ldots, D, \tag{1}$$

where the inverse warp ω_i maps observation time to a common canonical time and is shared across channels, $f_{id}(t)$ is the amplitude function for channel d, and $\epsilon_{id}(t)$ is measurement noise. Following classical FDA formulations (Ramsay & Silverman, 2005), we assume ω_i is a non-decreasing, continuous map on the compact domain $[0,1]$ with boundary conditions $\omega_i(0) = 0$ and $\omega_i(1) = 1$; the domain is set to $[0,1]$ without loss of generality. Our goal is to learn nonlinear representations that disentangle amplitude variation in $f_{id}(t)$ from phase variation in $\omega_i(t)$, yielding finite, compressed, and meaningful features for downstream modeling tasks such as regression, classification, and clustering.

As shown in Figure 2, our framework jointly learns a functional encoder $\mathcal{E}_\theta$ that maps $\mathbf{x}_i(t)$ to an amplitude embedding $\mathbf{a}_i \in \mathbb{R}^r$ and a phase embedding $\mathbf{p}_i \in \mathbb{R}^q$; an amplitude decoder $\mathcal{D}_\phi$ that reconstructs amplitude functions $\{\widehat{f}_{id}\}_{d=1}^D$ from $\mathbf{a}_i$; and a phase network $\mathcal{G}_\psi$ that reconstructs the inverse warp $\widehat{\omega}_i$ from $\mathbf{p}_i$. Here $\widehat{\cdot}$ denotes estimated functions, and θ, ϕ, ψ are the network parameters. We minimize the reconstruction error

$$\theta, \phi, \psi = \arg\min_{\theta,\phi,\psi} \frac{1}{ND} \sum_{i=1}^{N} \sum_{d=1}^{D} \int_0^1 \|x_{id}(t) - \widehat{x}_{id}(t)\|^2 \, \mathrm{d}t, \tag{2}$$

where $\widehat{x}_{id}(t) = \widehat{f}_{id}\big(\widehat{\omega}_i(t)\big)$. The amplitude embedding $\mathbf{a}_i$ and phase embedding $\mathbf{p}_i$ separately encode amplitude and phase patterns. This disentanglement is crucial for downstream modeling: for example, clustering with only $\{\mathbf{a}_i\}$ typically outperforms clustering with both $\{\mathbf{a}_i, \mathbf{p}_i\}$, since the latter entangles phase information with amplitude patterns. We call this framework AP-FAE, short for amplitude–phase functional autoencoders. We next describe its core modules step by step.

Functional encoder. Following classical functional autoencoders (Hsieh et al., 2021; Wu et al., 2024), we encode $\mathbf{x}_i(t)$ into K features $\mathbf{s}_i = (s_{i1}, \ldots, s_{iK})^\top$ using functional weights and inner products:

$$s_{ik} = \sum_{d=1}^{D} \langle x_{id}(t), w_{kd}(t) \rangle = \sum_{d=1}^{D} \int_0^1 x_{id}(t) w_{kd}(t) \, \mathrm{d}t, \qquad k = 1, \ldots, K. \tag{3}$$

Each weight function is expanded in J fixed cubic B-spline bases $b_1(t), \ldots, b_J(t)$ on $[0,1]$:

$$w_{kd}(t) = \sum_{j=1}^{J} \alpha_{kdj} b_j(t), \qquad d = 1, \ldots, D, \tag{4}$$

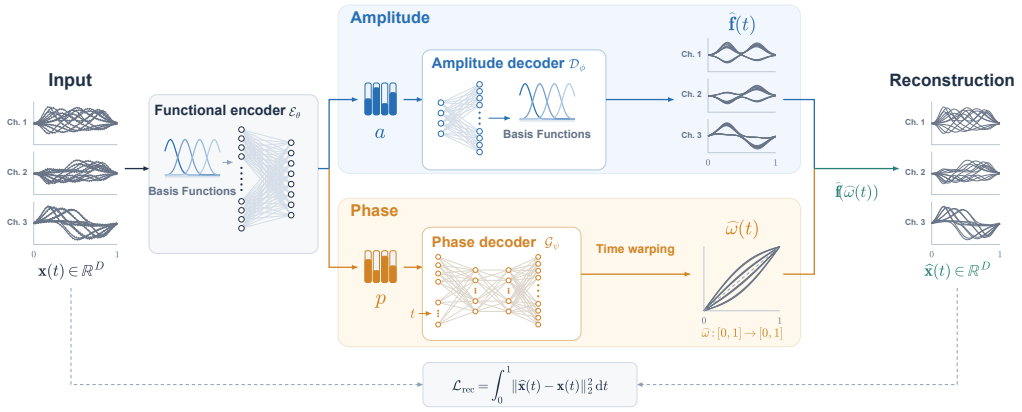


Figure 2: Overview of AP-FAE. A functional encoder maps $\mathbf{x}(t)$ to amplitude and phase embeddings, decoded into channel-specific amplitude functions and a shared inverse warp; modules are trained jointly via reconstruction error to disentangle amplitude and phase variation. We use $D = 3$ for illustration, with "Ch." denoting "Channel".

where α_{kdj} are trainable coefficients. For discretely observed functional data, the inner products are evaluated by numerical quadrature on a common grid $0 = t_1 < \cdots < t_M = 1$ (Fornberg, 2021). We then apply an element-wise ReLU $\sigma(z) = \max\{0, z\}$ followed by an affine map:

$$\mathcal{E}_\theta(\mathbf{x}_i) = W_1 \sigma(\mathbf{s}_i) + \mathbf{b}_1 = \begin{pmatrix} \mathbf{a}_i \\ \mathbf{p}_i \end{pmatrix}, \tag{5}$$

where $W_1 \in \mathbb{R}^{(r+q)\times K}$ and $\mathbf{b}_1 \in \mathbb{R}^{r+q}$ are trainable. The first r elements form $\mathbf{a}_i$, and the remaining q elements form $\mathbf{p}_i$. Both embeddings use information from all D channels. The encoder parameters are $\theta = \{\alpha_{kdj}, W_1, \mathbf{b}_1\}$.

Amplitude decoder. The amplitude decoder maps $\mathbf{a}_i$ to amplitude functions on the canonical time scale: $\mathcal{D}_\phi(\mathbf{a}_i) = (\widehat{f}_{i1}(t), \ldots, \widehat{f}_{iD}(t))^\top$. Using the same fixed B-spline bases as the encoder,

$$\widehat{f}_{id}(t) = \sum_{j=1}^{J} c_{idj} b_j(t), \qquad c_{idj} = \sum_{k=1}^{K} \beta_{dkj} v_{ik}, \qquad d = 1, \ldots, D, \tag{6}$$

where

$$\mathbf{v}_i = (v_{i1}, \ldots, v_{iK})^\top = \sigma(W_2 \mathbf{a}_i + \mathbf{b}_2) \in \mathbb{R}^K. \tag{7}$$

Here $W_2 \in \mathbb{R}^{K\times r}$, $\mathbf{b}_2 \in \mathbb{R}^K$, and β_{dkj} are trainable. Different channels have channel-specific coefficients but share the same amplitude embedding. The phase embedding $\mathbf{p}_i$ is not used by this decoder. The amplitude-decoder parameters are $\phi = \{\beta_{dkj}, W_2, \mathbf{b}_2\}$.

Phase decoder and time warping. To reconstruct raw functional data on the observation time scale, we need the inverse warp $\widehat{\omega}_i(t)$. We use a multilayer perceptron $\mathcal{G}_\psi$ that takes a time coordinate and the phase code $(t, \mathbf{p}_i)$ as inputs, and define

$$\widehat{\omega}_i(t) = \frac{\int_0^t \exp\{\mathcal{G}_\psi(u, \mathbf{p}_i)\} \, \mathrm{d}u}{\int_0^1 \exp\{\mathcal{G}_\psi(u, \mathbf{p}_i)\} \, \mathrm{d}u}. \tag{8}$$

The exponentiation ensures a positive warp density, so $\widehat{\omega}_i$ is strictly increasing and preserves the ordering of observed time points. Normalization fixes the endpoints $\widehat{\omega}_i(0) = 0$ and $\widehat{\omega}_i(1) = 1$. Larger values of $\widehat{\omega}_i(t)$ represent faster progress along the canonical time scale. The network parameters ψ are shared across subjects, while $\mathbf{p}_i$ allows subject-specific warps. For a given subject, the same $\widehat{\omega}_i$ acts on every channel, preserving the common time coordinate across channels. Algorithm 1 in Appendix A.1 summarizes the numerical computation of $\widehat{\omega}_i$; implementation details are provided in Appendix A.2.

Reconstruction and learning. We reconstruct the raw functional data by first mapping t to canonical time $\widehat{\omega}_i(t)$ and then evaluating the decoded amplitude at $\widehat{\omega}_i(t)$:

$$\widehat{x}_{id}(t) = \widehat{f}_{id}\big(\widehat{\omega}_i(t)\big) = \sum_{j=1}^{J} c_{idj} b_j\big(\widehat{\omega}_i(t)\big), \qquad d = 1, \ldots, D. \tag{9}$$

Equivalently, $\widehat{x}_{id} = \widehat{f}_{id} \circ \widehat{\omega}_i$, where $\circ$ denotes function composition. Thus, $\mathbf{a}_i$ determines the curve shape through $\widehat{f}_{id}(t)$, while $\mathbf{p}_i$ determines where that shape is evaluated in time through $\widehat{\omega}_i(t)$. Using the common time grid $t_1, \ldots, t_M$, all parameters are learned jointly by minimizing

$$\mathcal{L}_{\mathrm{rec}}(\theta, \phi, \psi) = \frac{1}{NDM} \sum_{i=1}^{N} \sum_{d=1}^{D} \sum_{m=1}^{M} \left[x_{id}(t_m) - \widehat{x}_{id}(t_m)\right]^2. \tag{10}$$

Algorithm 2 in Appendix A.1 summarizes the joint updates using AdamW(Loshchilov & Hutter, 2019). For downstream tasks, we use $\{\mathbf{a}_i\}$ for amplitude-based clustering, and defer the use of both $\{\mathbf{a}_i, \mathbf{p}_i\}$ to settings where phase variation is predictive; the estimated functions $\{\widehat{f}_{id}, \widehat{\omega}_i\}$ can further visualize the learned amplitude–phase disentanglement.

Computational and space complexity. Training AP-FAE scales linearly in the number of subjects N, channels D, grid points M, and B-spline basis size J. The shared phase network avoids an additional factor of D in the warping pathway. Working memory during minibatch training is dominated by model and optimizer storage together with the activation cache, while full-dataset storage requires $\mathcal{O}(NDM)$. A complete derivation, including per-epoch complexity and memory breakdown, is given in Appendix A.3.

4 THEORY

We bound the error in recovering the true amplitude curves by registration, reconstruction, and measurement-noise errors. Let $\mathbf{x}_i = \mathbf{f}_i \circ \omega_i + \boldsymbol{e}_i$ denote the observed curves and $\widehat{\mathbf{x}}_i = \widehat{\mathbf{f}}_i \circ \widehat{\omega}_i$ the fitted reconstructions, where $\boldsymbol{e}_i \in L^2([0,1]; \mathbb{R}^D)$ is measurement noise. For a D-variate function $\mathbf{u}$, define $\|\mathbf{u}\|_{L^2}^2 = \sum_{d=1}^{D} \int_0^1 u_d(t)^2 \, \mathrm{d}t$.

Theorem 1 (Amplitude error bound)**.** *Assume that (i) $\|\mathbf{f}_i(s) - \mathbf{f}_i(t)\|_2 \le L|s-t|$ for all i and $s, t \in [0,1]$; (ii) the true and fitted inverse warps $\omega_i, \widehat{\omega}_i$ are strictly increasing continuous bijections of $[0,1]$ onto itself; and (iii) each fitted inverse warp $\widehat{\omega}_i$ is C-Lipschitz. Define*

$$\epsilon_g = \max_i \|\widehat{\omega}_i - \omega_i\|_\infty, \qquad \epsilon_x^2 = \frac{1}{N} \sum_{i=1}^{N} \|\widehat{\mathbf{x}}_i - \mathbf{x}_i\|_{L^2}^2, \qquad \tau^2 = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{e}_i\|_{L^2}^2.$$

Then

$$\left\{ \frac{1}{N} \sum_{i=1}^{N} \|\widehat{\mathbf{f}}_i - \mathbf{f}_i\|_{L^2}^2 \right\}^{1/2} \le L\epsilon_g + \sqrt{C}\,(\epsilon_x + \tau). \tag{11}$$

Here ϵ_x is measured against the noisy observations. This deterministic bound requires no independence or zero-mean assumption on the noise.

Remark 1 (Practical implication and feasibility)**.** *Theorem 1 shows that amplitude recovery is controlled by three separately measurable quantities: registration error ϵ_g, reconstruction error ϵ_x, and noise level τ. Thus, small reconstruction error alone does not guarantee accurate amplitude recovery; reliable amplitude embeddings require accurate phase alignment. In practice, ϵ_x is directly minimized during training, τ can be estimated from repeated measurements or a noise model, and ϵ_g can be monitored on synthetic or held-out aligned data. The assumptions are mild for functional registration: the fitted warps are strictly increasing by construction through positive warp densities and endpoint normalization, and bounded-slope piecewise-linear or smooth warps satisfy the C-Lipschitz condition directly. In our simulations, the resulting bound is satisfied across all scenarios. As in functional registration, amplitude and phase are identifiable only relative to a common aligned coordinate; the theorem therefore conditions on registration accuracy in that coordinate rather than deriving it from reconstruction loss alone.*

The proof is given in Appendix B.

5 EXPERIMENTS

Setup. AP-FAE uses $J = 20$ cubic B-spline bases, functional hidden width $K = 16$, amplitude and phase dimensions $r = q = 4$, and a time-conditioned warp MLP with two hidden layers of width 32. Clustering applies K-means with 20 initializations to the amplitude embeddings $\{\mathbf{a}_i\}$. We evaluate on synthetic data and six real datasets (channels 1–8, classes 3–8), reporting clustering accuracy (ACC), adjusted total variance (ATV) for registration, and mean squared error (MSE) for reconstruction; higher ACC and lower ATV/MSE are better. We compare against four baseline categories: (i) *classical functional autoencoders*, including FAE (Hsieh et al., 2021), which learns a functional representation without an explicit warp, and FAEclust (Singh et al., 2025), which combines functional autoencoding with shape-based clustering; (ii) *neural functional registration*, represented by SrvfRegNet (Chen & Srivastava, 2021); (iii) *joint registration and clustering*, including JPCCA (Gaffney & Smyth, 2004) and SRC (Zeng et al., 2019), which estimate temporal alignment together with cluster structure; and (iv) *traditional functional clustering*, including Funclust (Jacques & Preda, 2013) and funHDDC (Bouveyron & Jacques, 2011), which model class structure through functional principal components and group-specific functional subspaces. For fair comparison, all clustering baselines use the true number of classes, and comparisons are partial: only AP-FAE performs clustering, alignment, and reconstruction jointly. Full protocols, metrics, and baseline implementations are given in Appendices C.1 and C.2.

5.1 SIMULATION

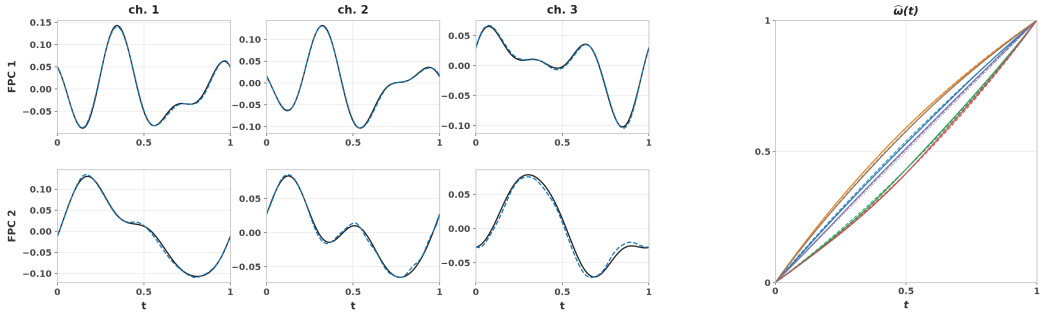
Data generation. We generate three classes of trivariate curves with 100 subjects per class, 100 grid points, class-specific three-harmonic amplitudes, and shared subject-specific exponential warps independent of class. We vary the cross-channel coefficient correlation $\rho \in \{0.2, 0.6\}$ and the phase-distortion range $B \in \{0.75, 1.25\}$; larger ρ induces stronger correlation among channels, while larger B produces more severe phase distortion. The complete design is in Appendix C.3.

Comparison with baselines. AP-FAE achieves the best ACC, ATV, and MSE in all four settings of Table 1, outperforming the strongest competitor SRC (e.g., at $\rho = 0.2, B = 0.75$: ACC 0.924 vs. 0.875, ATV 1.187 vs. 1.347). FAE, which lacks an explicit warp, degrades in clustering as B increases (ACC $0.690 \to 0.586$ at $\rho = 0.2$; $0.688 \to 0.572$ at $\rho = 0.6$), and the alignment-only SrvfRegNet also worsens (ATV $1.502 \to 1.965$ at $\rho = 0.2$), whereas AP-FAE remains stable; these trends confirm that amplitude-based grouping benefits from joint temporal modeling. Increasing ρ mainly lowers ACC while leaving ATV and MSE nearly unchanged, indicating that the metrics capture distinct aspects of the fit. Comparisons are partial: SrvfRegNet reports alignment only; Funclust/funHDDC report clustering only; FAE and FAEclust report clustering and reconstruction but not alignment; JPCCA and SRC report clustering and alignment but not reconstruction. Thus MSE is compared against two autoencoder references, FAE and FAEclust.

Settings	Metric	Funclust	funHDDC	JPCCA	SRC	SrvfRegNet	FAE	FAEclust	AP-FAE
$\rho = 0.2, B = 0.75$	ACC ↑	0.665	0.679	0.585	<u>0.875</u>	–	0.690	0.680	**0.924**
	ATV ↓	–	–	2.540	<u>1.347</u>	1.502	–	–	**1.187**
	MSE ↓	–	–	–	–	–	<u>0.015</u>	0.357	**0.014**
$\rho = 0.6, B = 0.75$	ACC ↑	0.591	0.702	0.608	<u>0.909</u>	–	0.688	0.676	**0.921**
	ATV ↓	–	–	2.435	<u>1.270</u>	1.547	–	–	**1.176**
	MSE ↓	–	–	–	–	–	<u>0.014</u>	0.356	**0.013**
$\rho = 0.2, B = 1.25$	ACC ↑	0.553	0.579	0.566	<u>0.880</u>	–	0.586	0.663	**0.910**
	ATV ↓	–	–	3.544	<u>1.662</u>	1.965	–	–	**1.177**
	MSE ↓	–	–	–	–	–	<u>0.016</u>	0.359	**0.015**
$\rho = 0.6, B = 1.25$	ACC ↑	0.652	0.565	0.570	<u>0.853</u>	–	0.572	0.661	**0.890**
	ATV ↓	–	–	3.628	<u>1.611</u>	1.719	–	–	**1.166**
	MSE ↓	–	–	–	–	–	<u>0.015</u>	0.360	**0.014**

Table 1: Clustering (ACC), alignment (ATV), and reconstruction (MSE) on trivariate synthetic data over B and ρ; means over 50 repetitions. Bold/underline mark the best/second-best distinct values per row, and "–" denotes not applicable.

Recovery of amplitude and phase. The simulation provides ground-truth amplitude functions $\{\mathbf{f}_i\}$ and inverse warps $\{\omega_i(t)\}$, which are withheld from modeling. We assess recovery by comparing the AP-FAE estimates $\{\widehat{\mathbf{f}}_i\}$ and $\{\widehat{\omega}_i(t)\}$ against these truths. Figure 3(a) compares the first two multivariate functional principal components (FPCs) of $\{\mathbf{f}_i\}$ (black solid) and $\{\widehat{\mathbf{f}}_i\}$ (blue dashed); Figure 3(b) compares a subset of true $\{\omega_i(t)\}$ (dashed) and estimated $\{\widehat{\omega}_i(t)\}$ (solid). The close agreement demonstrates effective amplitude and phase recovery. Full reconstruction is shown in Figure A1 in Appendix C.8.



(a) (b)

Figure 3: Amplitude and phase recovery ($\rho = 0.2$, $B = 0.75$). (a) True (black solid) vs. estimated (blue dashed) multivariate functional principal components (FPCs; first two shown) for channels 1–3. (b) True (dashed) vs. estimated (solid) inverse warps for six randomly chosen subjects (distinct colors); the black dotted line is the identity.

Numerical validation of amplitude error bound. We numerically validate Theorem 1 on 50 AP-FAE fits per simulation setting. All 200 fits satisfy the bound, with amplitude errors 0.214–0.253 and bounds 0.434–0.523, confirming the theory while showing the bound is conservative. Full details are in Appendix C.4.

5.2 Real-World Data

Datasets. We evaluate on six benchmark datasets: three multivariate from the UEA archive and three univariate from the UCR archive (Bagnall et al., 2018; Dau et al., 2019). **BasicMotions** ($D = 6$, 4 classes) contains 80 six-channel smartwatch recordings of walking, resting, running, and badminton, each with 100 time points. **ERing** ($D = 4$, 6 classes) comprises 300 four-channel wearable-ring samples of six finger postures, each with 65 time points. **UWaveGestureLibrary** ($D = 3$, 8 classes) includes 4,479 three-axis accelerometer recordings of eight hand gestures, each with 315 time points. **CBF** ($D = 1$, 3 classes) consists of 930 univariate plateau, rising-ramp, and falling-ramp signals with random onset and duration, each of length 128. **Plane** ($D = 1$, 7 classes) contains 210 univariate contour sequences from top-view aircraft images, each of length 144. **Symbols** ($D = 1$, 6 classes) comprises 1,020 univariate recordings of horizontal motion as participants draw displayed symbols, each with 398 observations.

Performance comparison. Table 2 shows that AP-FAE achieves the best ACC on five of six datasets, the best ATV on five of six, and the best MSE on all six; the exceptions are **UWaveGestureLibrary** (SRC best ACC, 0.888 vs. 0.811) and **ERing** (SRC best ATV, 4.773 vs. 5.554). The functional autoencoders FAE and FAEclust, which lack a phase pathway, are the runners-up in MSE on every dataset but are never best in clustering, supporting the benefit of explicit amplitude–phase separation. Alignment and reconstruction quality do not fully determine clustering: SRC attains the best ATV on **ERing** yet is second in ACC, and AP-FAE attains the best ATV on **UWaveGestureLibrary** yet is second in ACC, indicating that the three criteria capture complementary aspects of the fit. Figure 4 visualizes amplitude–phase disentanglement and reconstruction for one AP-FAE fit on channel 1 of **UWaveGestureLibrary**; additional visualizations are in Appendix C.9.

Datasets	Metric	Funclust	funHDDC	JPCCA	SRC	SrvfRegNet	FAE	FAEclust	AP-FAE
BasicMotions	ACC ↑	0.408	0.439	0.250	0.378	–	0.506	0.513	**0.807**
($D = 6$, 4 classes)	ATV ↓	–	–	107.025	113.544	114.108	–	–	**61.116**
	MSE ↓	–	–	–	–	–	17.955	19.252	**15.452**
ERing	ACC ↑	0.219	0.745	0.493	0.903	–	0.862	0.488	**0.943**
($D = 4$, 6 classes)	ATV ↓	–	–	10.317	**4.773**	12.826	–	–	5.554
	MSE ↓	–	–	–	–	–	0.257	0.620	**0.231**
UWaveGestureLibrary	ACC ↑	0.211	0.711	0.184	**0.888**	–	0.707	0.538	0.811
($D = 3$, 8 classes)	ATV ↓	–	–	20.478	14.771	37.213	–	–	**13.209**
	MSE ↓	–	–	–	–	–	0.327	0.562	**0.275**
CBF	ACC ↑	0.345	0.542	0.870	0.817	–	0.596	0.335	**0.911**
($D = 1$, 3 classes)	ATV ↓	–	–	145.374	6.153	16.712	–	–	**4.515**
	MSE ↓	–	–	–	–	–	0.254	0.402	**0.213**
Plane	ACC ↑	0.192	0.810	0.267	0.868	–	0.804	0.734	**0.875**
($D = 1$, 7 classes)	ATV ↓	–	–	0.292	0.297	0.920	–	–	**0.057**
	MSE ↓	–	–	–	–	–	0.075	0.418	**0.039**
Symbols	ACC ↑	0.193	0.564	0.178	0.738	–	0.677	0.626	**0.765**
($D = 1$, 6 classes)	ATV ↓	–	–	3.370	3.915	4.091	–	–	**1.781**
	MSE ↓	–	–	–	–	–	0.059	0.566	**0.038**

Table 2: Clustering (ACC), alignment (ATV), and reconstruction (MSE) on real datasets; D denotes the number of channels. Means over 50 repetitions; bold/underline mark the best and second-best values per row, and "–" denotes not applicable.

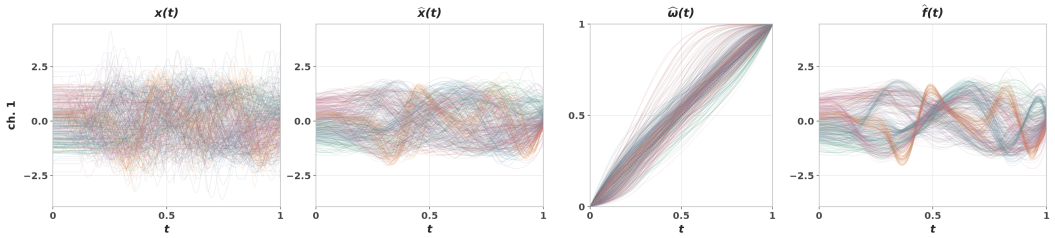


Figure 4: First channel of an AP-FAE fit on **UWaveGestureLibrary** ($D = 3$). Columns show raw $\mathbf{x}(t)$, reconstructed $\widehat{\mathbf{x}}(t)$, estimated inverse warps $\widehat{\omega}(t)$, and estimated amplitudes $\widehat{\mathbf{f}}(t)$; colors denote classes. All three channels appear in Figure A5 in Appendix C.9.

Ablation on embedding usage and size. Clustering is performed on the amplitude embedding $\mathbf{a}$, since the goal is to group curves by shape under temporal misalignment. We also test clustering on the phase embedding $\mathbf{p}$ alone and on both embeddings $[\mathbf{a}, \mathbf{p}]$. As shown in Table 3, clustering on $\mathbf{a}$ alone performs best, while adding $\mathbf{p}$ degrades performance because phase variation confuses the amplitude pattern; clustering on $\mathbf{p}$ alone can still be informative, and phase variation may carry useful signal for other tasks such as regression. These results highlight the importance of disentangling amplitude and phase. We further study the sensitivity to the embedding sizes r and q. Figure 5 reports clustering performance for $(r, q) = (2, 6), (3, 5), (4, 4), (5, 3), (6, 2)$ across all datasets, where $r = q = 4$ is consistently competitive and is used throughout.

Dataset	$\mathbf{a}$	$\mathbf{p}$	$[\mathbf{a}, \mathbf{p}]$
BasicMotions ($D = 6$)	**0.807**	0.800	0.695
ERing ($D = 4$)	**0.943**	0.721	0.840
UWaveGestureLibrary ($D = 3$)	**0.811**	0.536	0.701
CBF ($D = 1$)	**0.911**	0.585	0.821
Plane ($D = 1$)	**0.875**	0.739	0.854
Symbols ($D = 1$)	**0.765**	0.683	0.753

Table 3: Clustering accuracy (ACC) using amplitude $\mathbf{a}$, phase $\mathbf{p}$, or both embeddings.

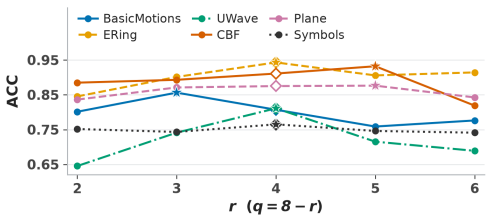


Figure 5: Sensitivity to embedding sizes; stars mark the best.

6 DISCUSSION

Robustness to irregular sampling and missing observations. We tested random missingness and uneven sampling in the trivariate synthetic setting ($B = 0.75$, $\rho = 0.2$) over 50 repetitions. Clustering ACC remained close to the complete-data baseline across all conditions, but clustering stability did not imply accurate reconstruction in sparsely observed regions. Under 80% random missingness, remaining observations and the MSE and ATV were 0.0072 and 1.212, respectively. In contrast, strongly uneven sampling retained all observations but increased the MSE and ATV to 0.0302 and 1.275. Thus, for alignment and reconstruction, what mattered was not merely how many observations were available, but whether they adequately covered the entire time interval. Details and the full table are in Appendix C.5.

Extreme warping or no warping. We tested continuous warps with short plateaus and nondecreasing warps with upward jumps. Mean ACC was nearly unchanged relative to the strictly increasing baseline, while longer plateaus and larger jumps increased ATV and MSE. Since AP-FAE assumes continuous, strictly positive-velocity warps, it cannot exactly represent plateaus or jumps and instead approximates them with small positive velocities and steep transitions. Amplitude adjustments can still preserve clustering and low average MSE, so these metrics do not imply recovery of the true warp. Details are in Appendix C.5. For the no-warping case ($\omega(t) = t$, $B = 0$), AP-FAE performs comparably to FAE (Table A4 in Appendix C.6) and estimates $\widehat{\omega}(t) \approx t$ (Figure A2 in Appendix C.8), confirming applicability whether or not misalignment exists.

Scalability. AP-FAE behaves stably at both ends of the sample-size and resolution ranges we tested. Increasing the number of subjects from 90 to $30{,}000$ lowers reconstruction error monotonically (MSE $0.016 \to 0.011$) while clustering accuracy and alignment remain essentially flat; training time grows approximately linearly in N. Varying the number of grid points from 30 to $1{,}000$ leaves MSE and ACC nearly unchanged, and training time changes by less than 7% over a 33-fold increase in M. The apparent growth in raw ATV with M is an artifact of its approximately $\sqrt{M}$ scaling: after normalization, ATV lies within 0.116–0.118 throughout. Full protocols, timing setup, and per-condition results appear in Appendix C.7.

7 CONCLUSION

We proposed AP-FAE, an unsupervised framework for temporally misaligned functional data that jointly learns nonlinear amplitude and phase representations, applying to both univariate and multivariate settings with a primary focus on the latter. By encoding all channels into separate amplitude and phase embeddings, decoding channel-specific amplitude functions in canonical time, and modeling phase variation with a shared monotone, endpoint-preserving warp, AP-FAE disentangles shape from timing under a reconstruction-only objective. This yields compact, interpretable embeddings that are not tied to a downstream task and support flexible use such as amplitude-based clustering. We also established a theoretical bound linking amplitude recovery to registration, reconstruction, and noise errors, and validated it numerically. Across synthetic and six real-world benchmarks, AP-FAE achieved strong clustering, alignment, and reconstruction, outperforming state-of-the-art baselines on most metrics; notably, clustering on amplitude embeddings alone consistently surpassed clustering on joint amplitude–phase embeddings, confirming the benefit of explicit disentanglement.

Our framework has limitations. It assumes a continuous, positive-velocity warp shared across channels, and its performance degrades when the true warps contain long constant segments or upward jumps, or when observations have severe missingness or strongly uneven sampling; in such cases, stable clustering and low average reconstruction error should not be interpreted as recovery of the true warp. In addition, the amplitude and phase embedding sizes are currently fixed rather than learned from data. Future work includes data-adaptive embedding sizes, richer warp families that accommodate discontinuities and subject-specific grids, and downstream tasks such as functional regression using amplitude embeddings, phase embeddings, or both.

AI USE STATEMENT

In this work, we used generative ChatGPT and Deepseek for polishing the writing of the manuscript and for assisting with code implementation. We have not used generative AI tools for research ideation, method design, or the analysis and interpretation of experimental results, and the remaining required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. Specifically, AI-assisted text edits were limited to improving grammar and clarity, and every edit was checked by the authors to ensure that the technical content and claims were unchanged. All AI-assisted code was manually reviewed, tested, and verified for correctness by the authors before being used in any experiment. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. The full source code of AP-FAE, including training and evaluation scripts, is available at the anonymous repository linked in the abstract. The method is described in Section 3, with algorithms and implementation details in Appendix A.1 and Appendix A.2. Complete proofs and assumptions for the theoretical results in Section 4 are given in Appendix B. The experimental setup, including datasets, baselines, and evaluation metrics, is given in Section 5.2 and Appendix C.1. Model sizes and baseline settings are listed in Appendix C.2, the synthetic data generation procedure is in Appendix C.3.

REFERENCES

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Charles Bouveyron and Julien Jacques. Model-based clustering of time series in group-specific functional subspaces. *Advances in Data Analysis and Classification*, 5(4):281–300, 2011. doi: 10.1007/s11634-011-0095-6.

Chao Chen and Anuj Srivastava. SrvfRegNet: elastic function registration using deep neural networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4457–4466, 2021.

Dong Chen and Hans-Georg Müller. Nonlinear manifold representations for functional data. *The Annals of Statistics*, 40(1):1–29, 2012. doi: 10.1214/11-AOS936.

Jeng-Min Chiou, Yu-Ting Chen, and Ya-Fang Yang. Multivariate functional principal component analysis: a normalization approach. *Statistica Sinica*, 24(4):1571–1596, 2014. doi: 10.5705/ss.2013.305.

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. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. doi: 10.1109/JAS.2019.1911747.

Frédéric Ferraty and Philippe Vieu. *Nonparametric functional data analysis: theory and practice*. Springer, 2006.

Bengt Fornberg. Improving the accuracy of the trapezoidal rule. *SIAM Review*, 63(1):167–180, 2021. doi: 10.1137/18M1229353.

Scott J. Gaffney and Padhraic Smyth. Joint probabilistic curve clustering and alignment. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

Clara Happ and Sonja Greven. Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113 (522):649–659, 2018. doi: 10.1080/01621459.2016.1273115.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647.

Tsung-Yu Hsieh, Yiwei Sun, Suhang Wang, and Vasant G. Honavar. Functional autoencoders for functional data representation learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 666–674. Society for Industrial and Applied Mathematics, 2021. doi: 10.1137/1.9781611976700.75.

Julien Jacques and Cristian Preda. Funclust: a curves clustering method using functional random variables density approximation. *Neurocomputing*, 112:164–171, 2013. doi: 10.1016/j.neucom.2012.11.042.

Siyuan Jiang, Yihan Hu, Wenjie Li, and Pengcheng Zeng. DeepFRC: an end-to-end deep learning model for functional registration and classification. In *International Conference on Learning Representations*, 2026.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. In *ICML Workshop on Anomaly Detection*, 2016.

James Stephen Marron, James O. Ramsay, Laura M. Sangalli, and Anuj Srivastava. Functional data analysis of amplitude and phase variation. *Statistical Science*, 30(4):468–484, 2015. doi: 10.1214/15-STS524.

James O. Ramsay and Bernard W. Silverman. *Functional data analysis*. Springer, 2005.

Alaa Sagheer and Mostafa Kotb. Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems. *Scientific Reports*, 9:19038, 2019. doi: 10.1038/s41598-019-55320-6.

Samuel V. Singh, Shirley Coyle, and Mimi Zhang. Shape-informed clustering of multi-dimensional functional data via deep functional autoencoders. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-5402.

Jun Song and Bing Li. Nonlinear and additive principal component analysis for functional data. *Journal of Multivariate Analysis*, 181:104675, 2021. doi: 10.1016/j.jmva.2020.104675.

Anuj Srivastava, Wei Wu, Sebastian Kurtek, Eric Klassen, and James Stephen Marron. Registration of functional data using Fisher–Rao metric. *arXiv preprint arXiv:1103.3817*, 2011.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 843–852. PMLR, 2015.

Ruoxu Tan, Yiming Zang, and Guosheng Yin. Nonlinear dimension reduction for functional data with application to clustering. *Statistica Sinica*, 34(3):1391–1412, 2024. doi: 10.5705/ss.202021.0393.

Rong Tang and Hans-Georg Müller. Pairwise curve synchronization for functional data. *Biometrika*, 95(4):875–889, 2008. doi: 10.1093/biomet/asn047.

J. Derek Tucker, Wei Wu, and Anuj Srivastava. Generative models for functional data using phase and amplitude separation. *Computational Statistics & Data Analysis*, 61:50–66, 2013. doi: 10.1016/j.csda.2012.12.001.

Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3:257–295, 2016. doi: 10.1146/annurev-statistics-041715-033624.

Shuoyang Wang, Wanyu Zhang, Guanqun Cao, and Yuan Huang. Functional data analysis using deep neural networks. *Wiley Interdisciplinary Reviews: Computational Statistics*, 16(4):e7000, 2024. doi: 10.1002/wics.70001.

Sidi Wu, Cédric Beaulac, and Jiguo Cao. Functional autoencoder for smoothing and representation learning. *Statistics and Computing*, 34(6):203, 2024. doi: 10.1007/s11222-024-10501-w.

Xinyang Xiong, Siyuan Jiang, and Pengcheng Zeng. NeuralFLoC: neural flow-based joint registration and clustering of functional data. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Pengcheng Zeng, Jian Qing Shi, and Won-Seok Kim. Simultaneous registration and clustering for multidimensional functional data. *Journal of Computational and Graphical Statistics*, 28(4):943–953, 2019. doi: 10.1080/10618600.2019.1607744.

A Details of Algorithms and Complexity

A.1 Algorithms

Algorithm 1 Time Warping

Require: Phase embedding $\mathbf{p}_i$; network parameters ψ; grid $0 = t_1 < \cdots < t_M = 1$
Ensure: Inverse warp $\{\widehat{\omega}_i(t_m)\}_{m=1}^M$

1: Compute network outputs $\rho_{im} \leftarrow \mathcal{G}_\psi(t_m, \mathbf{p}_i)$ for $m = 1, \ldots, M$
2: Compute stabilized positive densities $u_{im} \leftarrow \exp\{\rho_{im} - \max_\ell \rho_{i\ell}\}$ for $m = 1, \ldots, M$
3: Compute trapezoidal increments $\Delta_{im} \leftarrow \dfrac{u_{im} + u_{i,m+1}}{2}(t_{m+1} - t_m)$ for $m = 1, \ldots, M-1$
4: Accumulate integrals $H_{i1} \leftarrow 0, \qquad H_{im} \leftarrow \displaystyle\sum_{\ell=1}^{m-1} \Delta_{i\ell}$ for $m = 2, \ldots, M$
5: Normalize $\widehat{\omega}_i(t_m) \leftarrow H_{im}/H_{iM}$ for $m = 1, \ldots, M$
6: **return** inverse warp $\{\widehat{\omega}_i(t_m)\}_{m=1}^M$

Algorithm 2 Training AP-FAE

Require: Observations $\{x_{id}(t_m)\}_{i=1,d=1,m=1}^{N,D,M}$; latent dimensions r, q; number of B-spline bases J
Ensure: Fitted model, reconstructions and embeddings

1: Initialize encoder parameters θ, amplitude-decoder parameters ϕ, and phase-decoder parameters ψ
2: Precompute B-spline basis values
3: **while** stopping criterion is not satisfied **do**
4: $\quad(\mathbf{a}_i, \mathbf{p}_i) \leftarrow \mathcal{E}_\theta(\mathbf{x}_i)$ for $i = 1, \ldots, N$
5: $\quad\widehat{f}_{id} \leftarrow \mathcal{D}_\phi(\mathbf{a}_i)$ for $i = 1, \ldots, N$, $d = 1, \ldots, D$
6: $\quad\widehat{\omega}_i \leftarrow$ MonotoneWarp$(\mathbf{p}_i, \psi)$ using Algorithm 1
7: $\quad\widehat{x}_{id}(t_m) \leftarrow$ LinearInterp$(\widehat{f}_{id}, \widehat{\omega}_i(t_m))$
8: $\quad\mathcal{L}_{\text{rec}} \leftarrow \dfrac{1}{NDM}\displaystyle\sum_{i=1}^{N}\sum_{d=1}^{D}\sum_{m=1}^{M}[x_{id}(t_m) - \widehat{x}_{id}(t_m)]^2$
9: $\quad$Update θ, ϕ, ψ using AdamW and $\nabla\mathcal{L}_{\text{rec}}$
10: **end while**
11: **return** embeddings $\{\mathbf{a}_i, \mathbf{p}_i\}$ and functions $\{\widehat{f}_{id}, \widehat{\omega}_i\}$

A.2 Phase Decoder Implementation

We discretize Eq. equation 8 on the common grid $0 = t_1 < \cdots < t_M = 1$ using cumulative trapezoidal integration. A two-hidden-layer ReLU MLP of width L produces $\rho_{im} = G_\psi(t_m, p_i)$. For numerical stability, we set $u_{im} = \exp\{\rho_{im} - \max_\ell \rho_{i\ell}\}$ and compute

$$\Delta_{im} = \frac{u_{im} + u_{i,m+1}}{2}(t_{m+1} - t_m), \qquad m = 1, \ldots, M-1.$$

The estimated inverse warp is

$$\widehat{\omega}_i(t_1) = 0, \qquad \widehat{\omega}_i(t_m) = \frac{\sum_{\ell=1}^{m-1}\Delta_{i\ell}}{\sum_{\ell=1}^{M-1}\Delta_{i\ell}}, \qquad m = 2, \ldots, M. \tag{A.1}$$

Positive increments ensure strict monotonicity, and normalization gives $\widehat{\omega}_i(1) = 1$. Subtracting the maximum stabilizes exponentiation without changing the normalized warp.

At each $\widehat{\omega}_i(t_m)$, we linearly interpolate the two neighboring values of $\widehat{f}_{id}$ on the amplitude-decoder grid. The interpolation locations and weights are shared across channels, and gradients pass through both the amplitude values and the warp locations.

Our experiments use AdamW with learning rate 10^{-3} and weight decay 10^{-4}. Model sizes and baseline settings are listed in Section C.2. For downstream clustering, the 20 K-means initializations are independent clustering restarts and do not affect neural initialization.

A.3 Computational and Space Complexity

Let N be the number of samples, D the number of functional channels, M the number of grid points, and J the number of B-spline bases. Let K be the number of functional encoder features, S the minibatch size, r and q the amplitude and phase latent dimensions, and L the hidden width of the warping network.

Time complexity. For dense grid-based encoding and coefficient-based amplitude decoding, one epoch costs

$$\mathcal{O}\left(\left\lceil \frac{N}{S} \right\rceil KDJM + N\Big[DK(M+J) + DJM + DM + K(2r+q) + M(Lq+L^2)\Big]\right).$$

The first term evaluates the learned encoder weight functions on the grid after each parameter update. The remaining terms account for functional encoding, amplitude decoding, time-warping evaluation, interpolation, and reconstruction loss. Backpropagation has the same asymptotic order. For fixed grid resolution, basis size, network widths, and minibatch size, the cost scales linearly in both N and D. The shared warping network avoids an additional factor of D in the phase pathway.

Space complexity. Writing P for the number of trainable parameters, minibatch training requires

$$\mathcal{O}\big(P + MJ + DKM + S\big[DM + DJ + ML + K + r + q\big]\big)$$

working memory. This includes model and optimizer storage, basis evaluations, encoder weight functions, and intermediate activations. Storing the full dataset additionally requires $\mathcal{O}(NDM)$ memory. The shared warp needs only $\mathcal{O}(SM)$ interpolation locations and weights, while channel-specific inputs and reconstructions require $\mathcal{O}(SDM)$.

B Details on the Theory

Proof of Theorem 1. Recall that $\mathbf{x}_i = \mathbf{f}_i \circ \omega_i + \boldsymbol{e}_i$ is the observed curve and $\widehat{\mathbf{x}}_i = \widehat{\mathbf{f}}_i \circ \widehat{\omega}_i$ its reconstruction. Here $\mathbf{f}_i$ is the true amplitude curve, ω_i is the true inverse warp, and $\boldsymbol{\nu}_i$ is measurement noise. The quantities in the theorem are

$$\epsilon_g = \max_i \|\widehat{\omega}_i - \omega_i\|_\infty, \qquad \epsilon_x = \left\{\frac{1}{N}\sum_{i=1}^N \|\widehat{\mathbf{x}}_i - \mathbf{x}_i\|_{L^2}^2\right\}^{1/2}, \qquad \tau = \left\{\frac{1}{N}\sum_{i=1}^N \|\boldsymbol{e}_i\|_{L^2}^2\right\}^{1/2}.$$

Thus ϵ_g measures the largest inverse-warp error, while ϵ_x and τ are the root mean squared L^2 reconstruction error and noise level across subjects. The reconstruction error is measured against the noisy observations. The constants L and C are uniform Lipschitz bounds for the amplitude curves $\mathbf{f}_i$ and fitted inverse warps $\widehat{\omega}_i$, respectively. All L^2 norms below are taken over $[0,1]$ and sum over the D channels, as in the theorem.

Write $R_i = \widehat{\mathbf{x}}_i - \mathbf{x}_i$ for the reconstruction residual. Since $\widehat{\omega}_i$ is a strictly increasing continuous bijection of $[0,1]$, substitution of $t = \widehat{\omega}_i^{-1}(s)$ gives

$$\begin{aligned}\widehat{\mathbf{f}}_i(s) - \mathbf{f}_i(s) &= \widehat{\mathbf{x}}_i\{\widehat{\omega}_i^{-1}(s)\} - \mathbf{f}_i(s)\\ &= R_i\{\widehat{\omega}_i^{-1}(s)\} + \boldsymbol{\nu}_i\{\widehat{\omega}_i^{-1}(s)\}\\ &\quad + \mathbf{f}_i\{\omega_i(\widehat{\omega}_i^{-1}(s))\} - \mathbf{f}_i(s).\end{aligned}$$

The last term is bounded using the Lipschitz condition on $\mathbf{f}_i$. Indeed, $s = \widehat{\omega}_i(t)$ ranges over $[0,1]$ as t does, so

$$\sup_{s\in[0,1]} \big|\omega_i\{\widehat{\omega}_i^{-1}(s)\} - s\big| = \sup_{t\in[0,1]} |\omega_i(t) - \widehat{\omega}_i(t)| \le \epsilon_g.$$

Consequently,

$$\begin{aligned}\|\mathbf{f}_i \circ \omega_i \circ \widehat{\omega}_i^{-1} - \mathbf{f}_i\|_{L^2}^2 &= \int_0^1 \|\mathbf{f}_i\{\omega_i(\widehat{\omega}_i^{-1}(s))\} - \mathbf{f}_i(s)\|_2^2\, ds\\ &\le L^2 \int_0^1 \big|\omega_i\{\widehat{\omega}_i^{-1}(s)\} - s\big|^2\, ds \le L^2\epsilon_g^2.\end{aligned}$$

For the residual and noise terms, the C-Lipschitz condition implies that $\widehat{\omega}_i$ is absolutely continuous, with $0 \le \widehat{\omega}_i'(t) \le C$ almost everywhere. Changing variables by $s = \widehat{\omega}_i(t)$ yields, for any $\mathbf{u} \in L^2([0,1];\mathbb{R}^D)$,

$$\begin{aligned}\|\mathbf{u} \circ \widehat{\omega}_i^{-1}\|_{L^2}^2 &= \int_0^1 \|\mathbf{u}\{\widehat{\omega}_i^{-1}(s)\}\|_2^2 \, ds \\ &= \int_0^1 \|\mathbf{u}(t)\|_2^2 \widehat{\omega}_i'(t) \, dt \le C\|\mathbf{u}\|_{L^2}^2.\end{aligned}$$

Applying this estimate to R_i and $\boldsymbol{\nu}_i$ in the decomposition above, and then using the triangle inequality, gives

$$\begin{aligned}\|\widehat{\mathbf{f}}_i - \mathbf{f}_i\|_{L^2} &\le \|R_i \circ \widehat{\omega}_i^{-1}\|_{L^2} + \|\boldsymbol{e}_i \circ \widehat{\omega}_i^{-1}\|_{L^2} + L\epsilon_g \\ &\le \sqrt{C}\,\|R_i\|_{L^2} + \sqrt{C}\,\|\boldsymbol{e}_i\|_{L^2} + L\epsilon_g.\end{aligned}$$

Taking the root mean square over subjects and applying Minkowski's inequality in $\mathbb{R}^N$, we obtain

$$\begin{aligned}\left\{\frac{1}{N}\sum_{i=1}^N \|\widehat{\mathbf{f}}_i - \mathbf{f}_i\|_{L^2}^2\right\}^{1/2} &\le \left\{\frac{1}{N}\sum_{i=1}^N \left(\sqrt{C}\,\|R_i\|_{L^2} + \sqrt{C}\,\|\boldsymbol{e}_i\|_{L^2} + L\epsilon_g\right)^2\right\}^{1/2} \\ &\le \sqrt{C}\left\{\frac{1}{N}\sum_{i=1}^N \|R_i\|_{L^2}^2\right\}^{1/2} + \sqrt{C}\left\{\frac{1}{N}\sum_{i=1}^N \|\boldsymbol{e}_i\|_{L^2}^2\right\}^{1/2} + L\epsilon_g \\ &= L\epsilon_g + \sqrt{C}\,(\epsilon_x + \tau).\end{aligned}$$

This proves the stated bound. The argument holds for each realization of the noise and requires no distributional assumptions on $\boldsymbol{e}_i$. □

C Details of Experiments

C.1 Evaluation Metrics

Let N, M, and D denote the sample size, number of grid points, and number of channels.

Clustering. Let $N_{k\ell} = \sum_i \mathbf{1}\{y_i = k, \widehat{y}_i = \ell\}$, $n_k = \sum_\ell N_{k\ell}$, and $m_\ell = \sum_k N_{k\ell}$. ACC uses optimal one-to-one label matching $\mathcal{P}$:

$$\mathrm{ACC} = \frac{1}{N}\max_{\mathcal{P}} \sum_{(k,\ell)\in\mathcal{P}} N_{k\ell}. \tag{A.2}$$

Registration. For aligned observations $A_i = [x_{id}\{\widehat{\omega}_i^{-1}(t_m)\}]_{m,d}$, define $\overline{A}_k = n_k^{-1}\sum_{i:y_i=k} A_i$ and $S_k = \sum_{i:y_i=k} \|A_i - \overline{A}_k\|_F^2$. We use the pairwise ATV criterion of Jiang et al. (2026),

$$\mathrm{ATV} = \frac{2}{K(K-1)} \sum_{k<\ell} \frac{S_k + S_\ell}{(n_k + n_\ell)\|\overline{A}_k - \overline{A}_\ell\|_F}, \tag{A.3}$$

where K is the number of true classes and the Frobenius norm covers all grid points and channels. ATV uses class labels for evaluation but requires no true warping functions.

Reconstruction. For $\widehat{x}_{id}(t_m) = \widehat{f}_{id}\{\widehat{\omega}_i(t_m)\}$,

$$\mathrm{MSE} = \frac{1}{NMD}\sum_{i=1}^N \sum_{m=1}^M \sum_{d=1}^D [x_{id}(t_m) - \widehat{x}_{id}(t_m)]^2. \tag{A.4}$$

C.2 Baseline Implementation Details

- **FAE:** A functional autoencoder without phase variation, using the reported functional encoder–decoder settings (Wu et al., 2024).

- **FAEclust:** Official PyTorch implementation with elastic distance, architecture 16–8–4–8–16–16–16(Singh et al., 2025).
- **SrvfRegNet:** Local reproduction with a shared-warp multichannel extension and Adam; provides alignment-only entries (Chen & Srivastava, 2021).
- **SRC:** Scalar specialization for univariate examples and a coordinate-list extension of the author-supplied likelihood for trivariate curves (Zeng et al., 2019); both estimate a shared subject-specific warp without class labels.
- **JPCCA:** Spline-mixture reimplementation following the published model (Gaffney & Smyth, 2004).
- **Funclust and funHDDC:** Published R implementations (Jacques & Preda, 2013; Bouveyron & Jacques, 2011).

C.3 Synthetic Data Generation

We adapt the basis-expansion construction of Hsieh et al. (2021), using Gaussian basis coefficients to produce individual curves around class templates and adding a shared time warp. The main comparison contains three classes of trivariate curves ($D = 3$), with 100 subjects per class observed at 100 equally spaced points on $[0, 1]$.

Observed curves. For subject i and channel d,

$$x_{id}(t_j) = f_{id}\{\omega_i(t_j)\} + \epsilon_{idj}, \qquad \epsilon_{idj} \overset{\text{iid}}{\sim} N(0, 0.10^2), \tag{A.5}$$

where f_{id} is the amplitude curve before warping, g_i^{-1} changes its timing, and ϵ_{idj} is independent measurement noise.

Time warping. Each subject has one warp shared by all channels:

$$\omega_i(t) = \begin{cases} \{\exp(b_i t) - 1\}/\{\exp(b_i) - 1\}, & b_i \neq 0, \\ t, & b_i = 0, \end{cases} \qquad b_i \sim \text{Unif}(-B, B). \tag{A.6}$$

As in the model, $\omega_i(t)$ maps observed time to aligned time. The map is increasing and fixes both endpoints. We vary $B \in \{0.75, 1.25\}$ to control the range of time distortions. Warps are independent of the amplitude coefficients and class membership, so timing supplies no class information.

Amplitude curves and basis coefficients. The amplitude curve is a linear combination of three harmonics,

$$f_{id}(t) = \sum_{h=1}^{3} c_{idh}\phi_{dh}(t), \qquad d = 1, \ldots, D. \tag{A.7}$$

For a subject in class k, draw the coefficient vector across channels, $\boldsymbol{c}_{ih} = (c_{i1h}, c_{i2h}, c_{i3h})^\top$, from

$$\boldsymbol{c}_{ih} \sim N_3\big(\beta_{kh}\mathbf{1}_3, (0.20\beta_{kh})^2\Sigma_\rho\big), \qquad \Sigma_\rho = \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix},$$

where $\mathbf{1}_3 = (1, 1, 1)^\top$. Draws are independent across subjects and harmonics. Each coefficient has class-specific mean β_{kh} and standard deviation $0.20|\beta_{kh}|$; coefficients of the same harmonic have correlation ρ across channels. The three class mean vectors are fixed across repetitions:

$$\begin{aligned} \boldsymbol{\beta}_1 &= (0.9175,\ 0.3195,\ -0.1380), \\ \boldsymbol{\beta}_2 &= (0.9175,\ -0.1205,\ 0.1260), \\ \boldsymbol{\beta}_3 &= (0.6150,\ 0.3910,\ -0.2480). \end{aligned}$$

We vary $\rho \in \{0.2, 0.6\}$ while keeping marginal coefficient variances fixed. Here ρ describes coefficient correlation, rather than correlation between observed curves. The channel-specific harmonics are

$$\begin{aligned} \boldsymbol{\phi}_1(t) &= \{\sin(2\pi t),\ \sin(4\pi t),\ \cos(6\pi t)\}, \\ \boldsymbol{\phi}_2(t) &= \{0.75\sin(2\pi t + 0.35),\ 0.95\sin(4\pi t + 0.55),\ 0.80\cos(6\pi t + 0.15)\}, \\ \boldsymbol{\phi}_3(t) &= \{0.90\sin(2\pi t - 0.22),\ -0.65\sin(4\pi t + 0.30),\ 0.50\cos(6\pi t - 0.40)\}. \end{aligned}$$

C.4 Numerical Validation of the Amplitude Recovery Bound

We evaluate Eq.(11) on 50 AP-FAE fits for each of the four simulation settings. Amplitude and registration errors are measured in the original generating time coordinate; reconstruction error is measured against the noisy observations, and τ is computed from the corresponding noise realizations. Table A1 compares the amplitude root mean squared L^2 error with $\epsilon = L\epsilon_g + \sqrt{C}\,(\epsilon_x + \tau)$. All 200 fits satisfy the bound. Across settings, the mean amplitude error ranges from 0.214 to 0.253, compared with mean bounds of 0.434–0.523, so the bound is conservative in these simulations. These results provide direct numerical validation of Theorem 1 and confirm that the derived bound holds in practice.

B	ρ	L	C	Amplitude error	Upper bound ϵ
0.75	0.2	17.402	1.680	0.253	0.523
0.75	0.6	18.006	1.746	0.252	0.523
1.25	0.2	17.402	1.971	0.221	0.451
1.25	0.6	18.006	2.020	0.214	0.434

Table A1: Numerical assessment of the amplitude recovery bound for the four trivariate simulation settings. Entries are means over 50 fits per setting. Amplitude error denotes root mean squared L^2 error.

C.5 Details of the Robustness Experiments

Robustness to Irregular Sampling and Missing Observations. We evaluated robustness to random missingness and unevenly spaced observations in the trivariate synthetic setting with $B = 0.75$ and $\rho = 0.2$. Each experiment was repeated 50 times. Table A2 reports clustering accuracy (ACC), adjusted total variance (ATV), and reconstruction error (MSE) against the noise-free signal on a dense reference grid.

Table A2: Robustness to irregular sampling after retraining ($B = 0.75$, $\rho = 0.2$). Values are means over 50 repetitions.

Sampling condition	ACC ↑	ATV ↓	MSE ↓
Complete uniform grid	0.924	1.187	0.0047
Random missingness: 20%	0.924	1.177	0.0043
Random missingness: 50%	0.935	1.181	0.0048
Random missingness: 80%	0.932	1.212	0.0072
Uneven sampling: $\alpha_{\mathrm{samp}} = 2$	0.922	1.185	0.0056
Uneven sampling: $\alpha_{\mathrm{samp}} = 4$	0.922	1.275	0.0302

Extreme Warping Cases. We further considered continuous warps containing short constant segments and nondecreasing warps containing upward jumps. All curves were observed on the complete grid. Each experiment was repeated 50 times. Table A3 reports ACC, ATV, and reconstruction MSE.

Table A3: Sensitivity to extreme true warps after retraining ($B = 0.75$, $\rho = 0.2$). Values are means over 50 repetitions.

Warp configuration	ACC ↑	ATV ↓	MSE ↓
Strictly increasing baseline	0.924	1.187	0.00470
One plateau: length 0.04	0.933	1.236	0.00523
One plateau: length 0.08	0.920	1.346	0.00756
One plateau: length 0.10	0.929	1.357	0.00833
Two plateaus: length 0.04 each	0.915	1.281	0.00616
One upward jump: $\Delta_{\mathrm{J}} = 0.02$	0.924	1.211	0.00506
One upward jump: $\Delta_{\mathrm{J}} = 0.05$	0.927	1.307	0.00662

Experimental Setup. All robustness experiments used the same synthetic setting: three channels, three classes, 100 subjects per class, $\rho = 0.2$, $B = 0.75$, and 50 repetitions. AP-FAE was re-

trained from scratch under every condition. Training used1500 epochs, minibatch size 32, AdamW with learning rate 10^{-3} and weight decay 10^{-4}, 20 basis functions, hidden width 16, and four-dimensional amplitude and phase codes. Clustering was performed once at the final checkpoint; ACC used true labels only for evaluation.

Irregular Sampling and Missing Observations. The two sampling mechanisms isolate different practical failures. Random missingness models scattered observation loss, such as intermittent sensor or recording failures. Uneven sampling models a valid but highly nonuniform design, such as measurements concentrated near the end of follow-up, and tests whether nominal sample size can mask poor temporal coverage.

Random missingness removed 20%, 50%, or 80% of the coordinates. The two endpoints were always retained.

For uneven sampling, interior coordinates were selected without replacement from a 1001-point candidate grid and combined with the two endpoints. Selection weights were proportional to

$$t^{\alpha_{\mathrm{samp}}-1},$$

where $\alpha_{\mathrm{samp}} = 2$ and 4 generated moderate and strong concentration toward the end of the domain. Within a replication, all subjects and channels shared the observation grid;

Warps with Constant Segments. A constant segment represents a temporary pause in latent progress: observed time advances while canonical time remains unchanged. It can also approximate saturation, rest, or repeated measurements of an unchanged latent state. This setting tests the strict positive-velocity assumption, because the true warp has zero derivative on a nonempty interval. The three single-plateau lengths vary the duration of one pause, while the two-plateau condition contrasts one contiguous pause with two separated pauses having the same total duration of 0.08.

For disjoint plateau intervals $[\ell_k, r_k]$, define

$$h_{\mathrm{P}}(t) = \frac{t - \sum_k \min\{\max(t-\ell_k, 0),\, r_k - \ell_k\}}{1 - L_{\mathrm{P}}}, \qquad L_{\mathrm{P}} = \sum_k (r_k - \ell_k),$$

and set

$$\omega_{\text{plateau}}(t) = \omega(h_{\mathrm{P}}(t))\,.$$

The resulting warp is continuous and nondecreasing, has constant segments, and satisfies $\omega_{\text{plateau}}(0) = 0$ and $\omega_{\text{plateau}}(1) = 1$. We tested one plateau of length 0.04, 0.08, or 0.10, and two plateaus of length 0.04 each. For each subject, the single-plateau centre was sampled from a 0.01-spaced grid in $[0.30, 0.70]$ and held fixed across the three lengths. The two-plateau centres were sampled from the corresponding grids in $[0.20, 0.35]$ and $[0.65, 0.80]$. Locations were generated independently of class, and all three channels shared the subject-specific warp.

Discontinuous Warps. We also tested warps with one upward jump. This represents a sudden advance in latent time and violates the model's continuity assumption. We first modified the time coordinate using

$$h_{\mathrm{J}}(t) = (1 - \Delta_{\mathrm{J}})t + \Delta_{\mathrm{J}}\,\mathbf{1}_{\{t \geq t_{\mathrm{J}}\}}, \qquad \omega_{\text{jump}}(t) = \omega(h_{\mathrm{J}}(t))\,.$$

Here, t_{J} is the jump location and Δ_{J} controls its size. We used $\Delta_{\mathrm{J}} = 0.02$ and 0.05. For each subject, we sampled one jump location between observation times in $(0.30, 0.70)$ and used that same location for both jump sizes. The two conditions therefore differed only in the jump size. The resulting warp remains nondecreasing and satisfies $\omega_{\text{jump}}(0) = 0$ and $\omega_{\text{jump}}(1) = 1$, but it is discontinuous.

C.6 Simulation without phase variation

We set $B = 0$ in the trivariate simulation, so the true warp is the identity, while retaining $\rho \in \{0.2, 0.6\}$ and fixing $K = 3$. Table A4 compares AP-FAE with FAE and FAEclust using ACC, ATV, and MSE. AP-FAE and FAE achieve similar ACC; their MSEs are equal at $\rho = 0.2$, while AP-FAE attains lower MSE at $\rho = 0.6$. Figure A2 illustrates the latter setting.

Settings	Metric	FAE	FAEclust	AP-FAE
$\rho = 0.2, B = 0$	ACC ↑	0.950	0.729	0.946
	ATV ↓	–	–	1.164
	MSE ↓	**0.015**	0.354	**0.015**
$\rho = 0.6, B = 0$	ACC ↑	**0.916**	0.766	<u>0.915</u>
	ATV ↓	–	–	1.139
	MSE ↓	<u>0.013</u>	0.354	**0.011**

Table A4: Trivariate simulation with $B = 0$. Bold and underline mark the best and second-best distinct values when multiple methods report a metric; "–" denotes not applicable.

C.7 Scalability

We examine the effects of sample size and grid resolution on AP-FAE in the trivariate setting with $B = 0.75$ and $\rho = 0.2$. Table A5 shows that increasing the sample size reduces reconstruction error overall, while clustering accuracy and alignment remain relatively stable. Table A6 shows only modest changes in reconstruction error and clustering accuracy across grid resolutions. The increase in raw ATV largely reflects its approximately $\sqrt{M}$-dependent scale (see equation A.3); ATV normalized by $\sqrt{M}$ remains between 0.1160 and 0.1187. Training times for 1500 epochs on one NVIDIA RTX 4090 (batch size 32) include per-epoch reconstruction checks and exclude file I/O. Over the tested range, time grows approximately linearly with sample size but changes little with grid resolution.

Per class	Total N	MSE ↓	ACC ↑	ATV ↓	Time (s)
30	90	0.016	0.949	<u>1.142</u>	19.62
50	150	0.014	**0.963**	**1.115**	31.19
100	300	0.014	0.924	1.187	62.77
1,000	3,000	<u>0.011</u>	<u>0.953</u>	1.166	555.09
10,000	30,000	**0.011**	0.952	1.180	5,422.98

Table A5: AP-FAE across sample sizes for $D = 3$, $B = 0.75$, $\rho = 0.2$, and $M = 100$. Bold and underline indicate the best and second-best distinct values for MSE, ACC, and ATV.

Grid points M	MSE ↓	ACC ↑	ATV ↓	Time (s)
30	**0.013**	<u>0.942</u>	**0.643**	60.82
50	<u>0.013</u>	**0.952**	<u>0.830</u>	60.33
100	0.014	0.924	1.187	62.77
1,000	0.013	0.935	3.670	65.09

Table A6: AP-FAE across grid resolutions for $D = 3$, $B = 0.75$, $\rho = 0.2$, and 100 subjects per class. Bold and underline indicate the best and second-best distinct values for MSE, ACC, and ATV.

C.8 SIMULATION VISUALIZATION

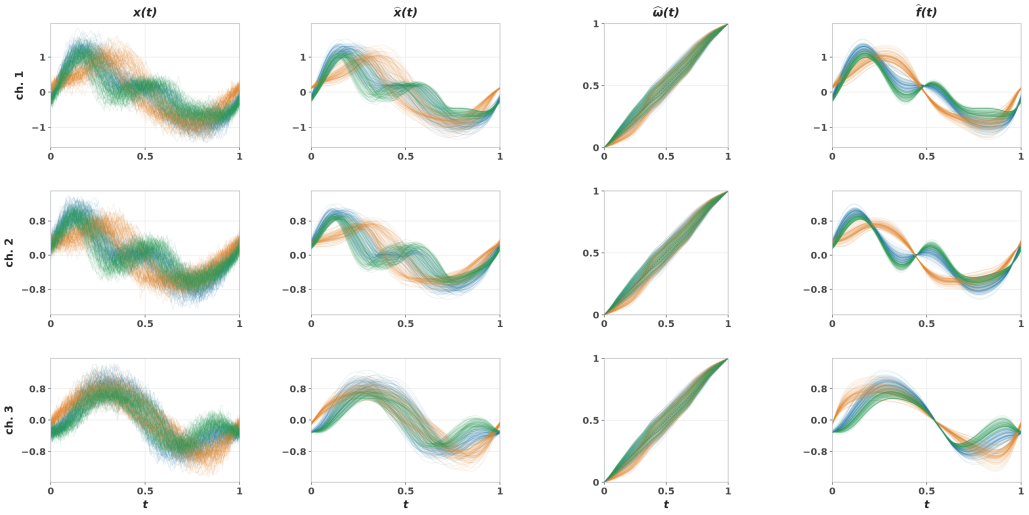


Figure A1: Trivariate AP-FAE fit ($\rho = 0, 2, B = 0.75$). Rows are channels; colours indicate classes and dotted lines denote identity.

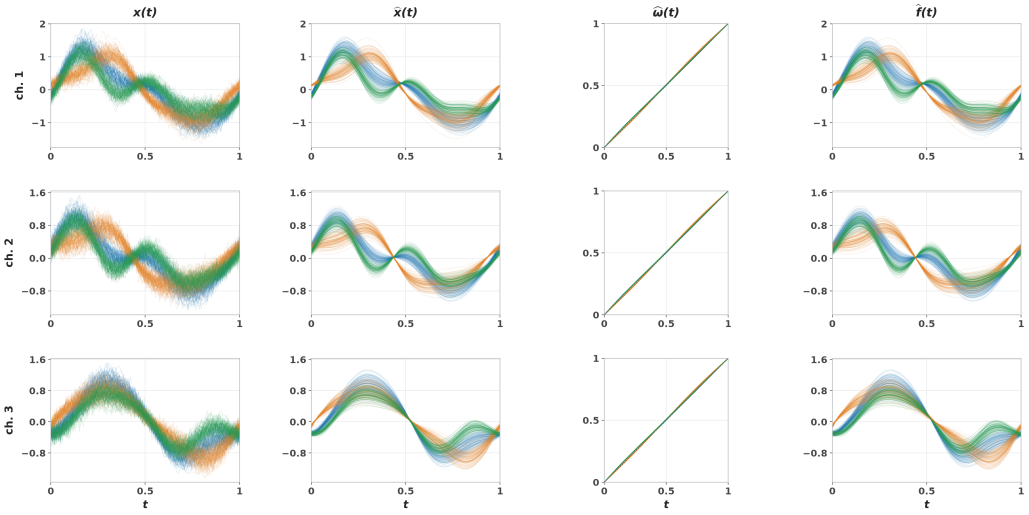


Figure A2: AP-FAE fit with $B = 0$ and $\rho = 0.6$.

C.9 REAL-DATA VISUALIZATIONS

The following figures visualize all the real datasets. Columns show observed curves, reconstructions, learned inverse warps, and aligned curves. Colours indicate classes; dotted lines denote identity.

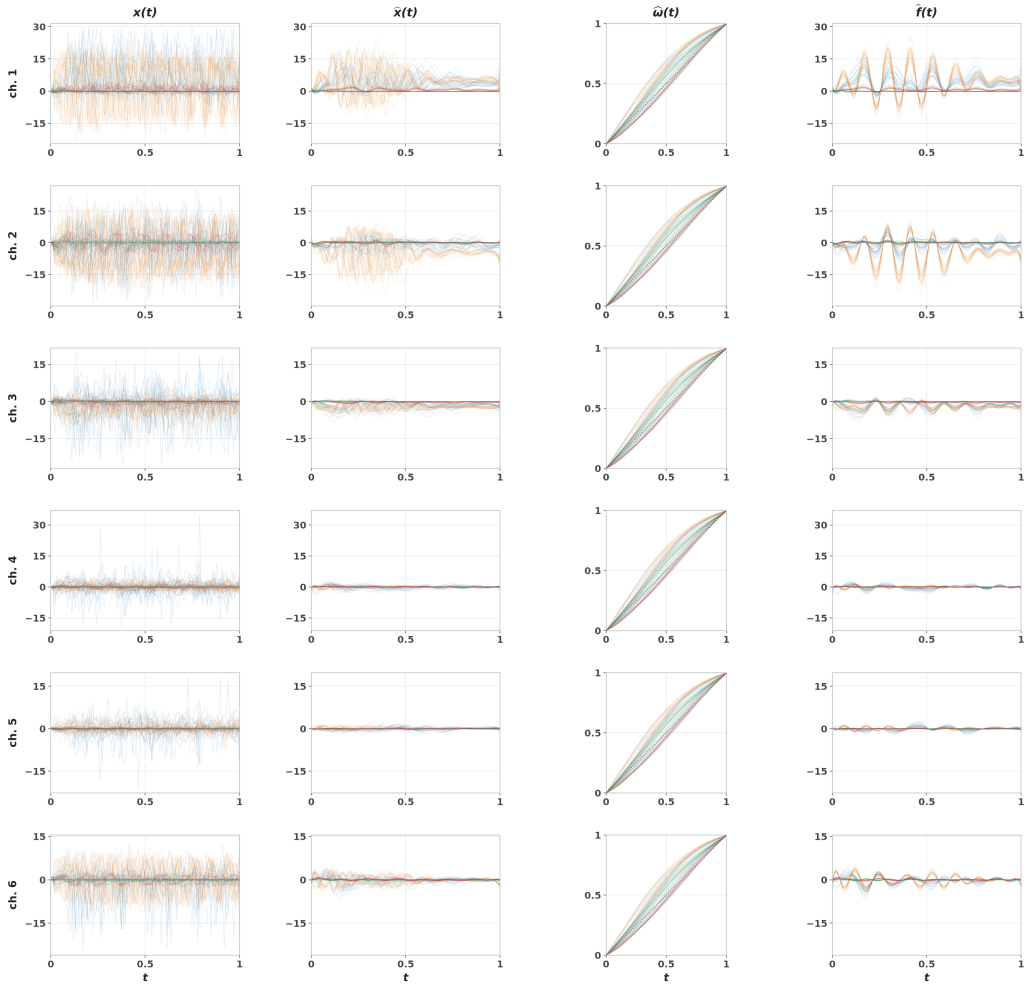


Figure A3: AP-FAE fit on BasicMotions ($D = 6$)

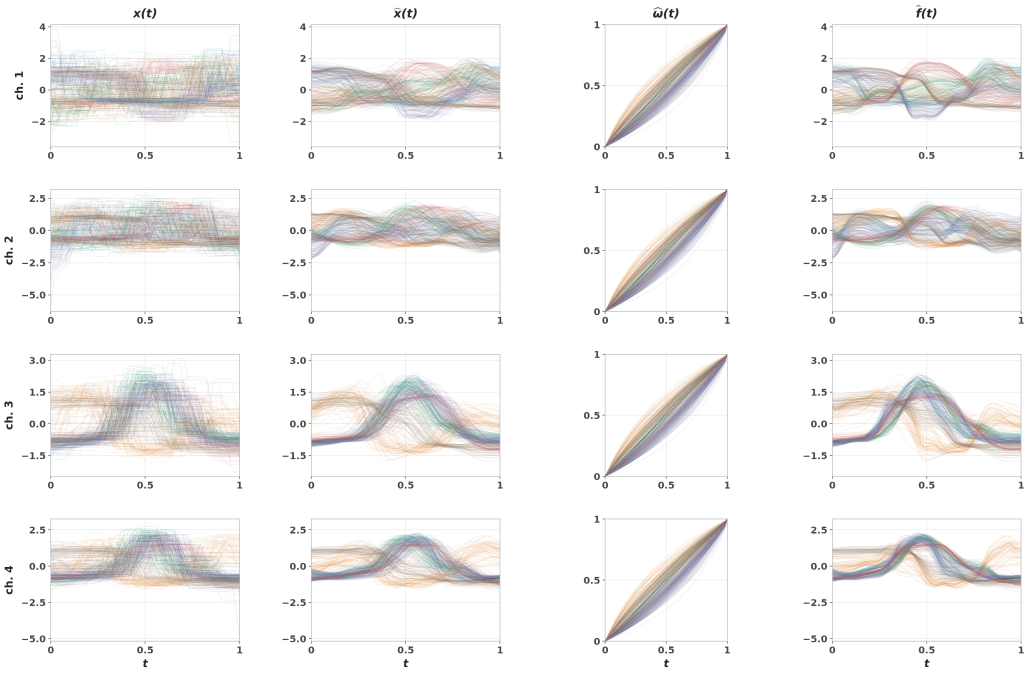


Figure A4: AP-FAE fit on ERing ($D = 4$)

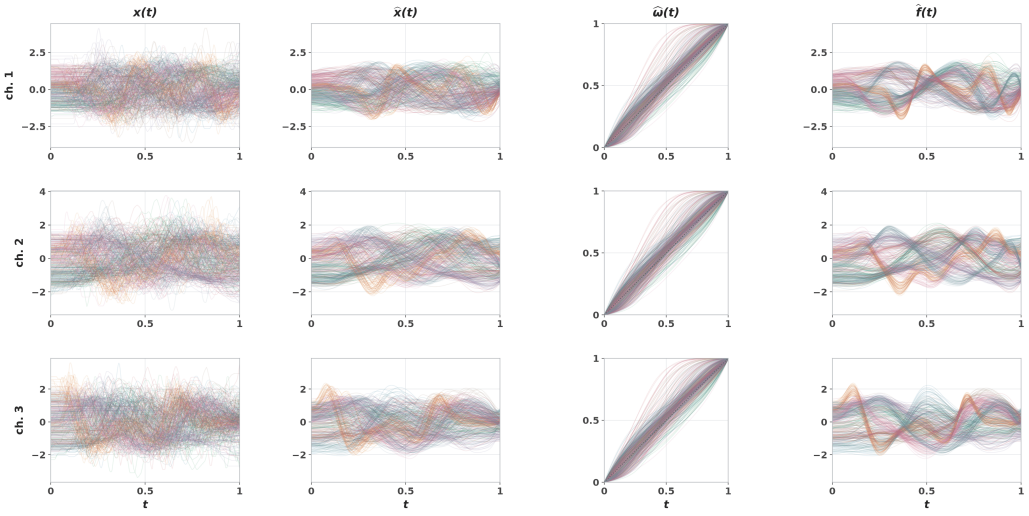


Figure A5: AP-FAE fit on UWaveGestureLibrary ($D = 3$)

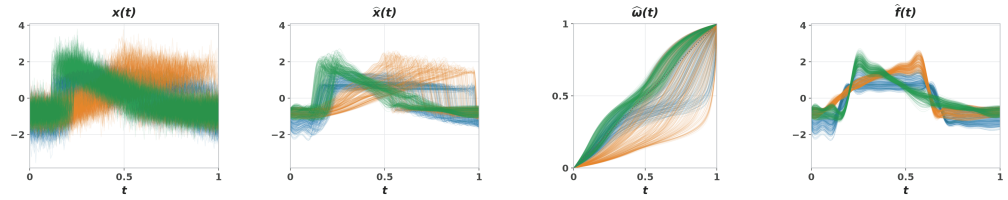


Figure A6: AP-FAE fit on CBF ($D = 1$).

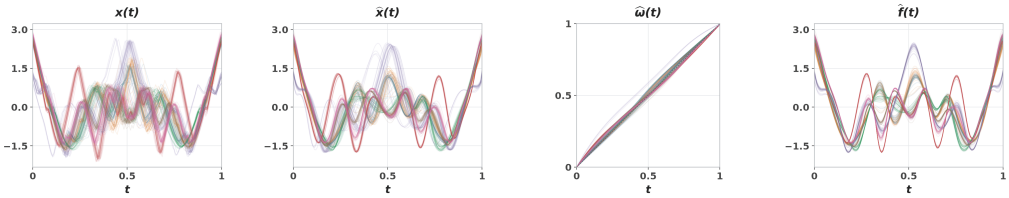


Figure A7: AP-FAE fit on Plane ($D = 1$).

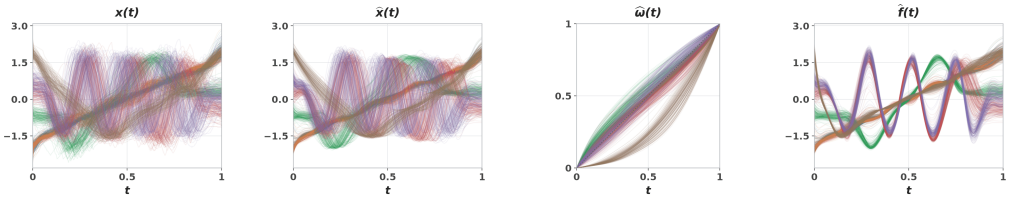


Figure A8: AP-FAE fit on Symbols ($D = 1$).