Shapes analysis for time series.

Anonymous Author(s)
Affiliation
Address
email

Abstract

Analyzing inter-individual variability of physiological functions is particularly appealing in medical and biological contexts to describe or quantify health conditions. Such analysis can be done by comparing individuals to a reference one with time series as biomedical data. This paper introduces an unsupervised representation learning (URL) algorithm for time series tailored to inter-individual studies. The idea is to represent time series as deformations of a reference time series. The deformations are diffeomorphisms parameterized and learned by our method called TS-LDDMM. Once the deformations and the reference time series are learned, the vector representations of individual time series are given by the parametrization of their corresponding deformation. At the crossroads between URL for time series and shape analysis, the proposed algorithm handles irregularly sampled multivariate time series of variable lengths and provides shape-based representations of temporal data. In this work, we establish a representation theorem for the graph of a time series and derive its consequences on the LDDMM framework. We showcase the advantages of our representation compared to existing methods using synthetic data and real-world examples motivated by biomedical applications.

1 Introduction

Our goal is to analyze the inter-individual variability within a time series dataset, an approach of significant interest in physiological contexts [23, 55, 4, 19]. Specifically, we aim to develop an unsupervised feature representation method that encodes the specificities of individual time series in comparison to a reference time series. In physiology, examining the various "shapes" in a time series related to biological phenomena and their variations due to individual differences or pathological conditions is common. However, the term "shape" lacks a precise definition and is more intuitively understood as the silhouette of a pattern in a time series. In this paper, we refer to the shape of a time series as the graph of this signal.

Although community structures with representatives can be learned in an unsupervised manner [52, 37] using contrastive loss [18, 51, 37] or similarity measures [2, 19, 43, 58], the study of inter-individual variability of shapes within a cluster [40, 48] remains an open problem in unsupervised representation learning (URL), particularly for *irregularly sampled* time series with *variable lengths*.

Our work explicitly focuses on learning shape-based representation of time series. First, we propose to view the shape of a time series not merely as its curve $\{s_t : \ t \in \mathsf{I}\}$, but as its graph $\mathsf{G}(s) = \{(t, s(t)) : \ t \in \mathsf{I}\}$. Then, building on the shape analysis literature [5, 54], we adopt the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework [5, 54] to analyze these graphs. The core idea is to represent each element $\mathsf{G}(s^j)$ of a dataset $(s^j)_{j\in[N]}$ as the transformation of a reference graph $\mathsf{G}(\mathbf{s}_0)$ by a diffeomorphism ϕ_j, i.e. $\mathsf{G}(s^j) \sim \phi_j.\mathsf{G}(\mathbf{s}_0)$. The diffeomorphism ϕ_j is learned by integrating an ordinary differential equation parameterized by a Reproducing Kernel Hilbert Space (RKHS). The parameters $(\alpha_j)_{j\in[N]}$ encoding the diffemorphisms $(\phi_j)_{j\in[N]}$ yield the

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

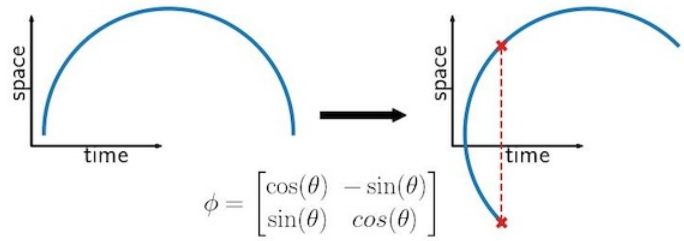


Figure 1: A time series' graph $\mathsf{G} = \{(t, s(t)) :\ t \in \mathsf{I}\}$ can lose its structure after applying a general diffeomorphism $\phi.\mathsf{G}$: a time value can be related to two values on the space axis.

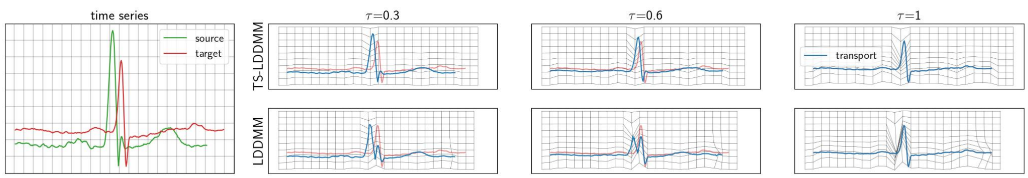


Figure 2: LDDMM and TS-LDDMM are applied to ECG data. We observe that LDDMM, using a general Gaussian kernel, does not learn the time translation of the first spike but changes the space values, i.e., one spike disappears before emerging at a translated position. At the same time, TS-LDDMM handles the time change in the shape. This difference of *deformations* implies differences in features *representations*.

representation features of the graphs $(\mathsf{G}(s^j))_{j\in[N]}$. Finally, these shape-encoding features can be used as inputs to any statistical or machine-learning model.

However, a graph time series transformation by a general diffeomorphism is not always a graph time series, see e.g. Figure 1, thus a graph time series is more than a simple curve [21]. Our contributions arise from this observation: we specify the class of diffeomorphisms to consider and show how to learn them. This change is fruitful in representing transformations of time series graphs as illustrated in Figure 2.

Our contributions can be summarized as follows:

- We propose an unsupervised method (TS-LDDMM) to analyze the inter-individual variability of shapes in a time series dataset (Section 4). In particular, the method can handle multivariate time series *irregularly sampled* and with *variable sizes*.
- We motivate our extension of LDDMM to time series by introducing a theoretical framework with a representation theorem for time series graph (Theorem 1) and kernels related to their structure (Lemma 1).
- We demonstrate the identifiability of the model by estimating the true generating parameter of synthetic data, and we highlight the sensitivity of our method concerning its hyperparameters (**??**), also providing guidelines for tuning (Appendix D).
- We highlight the *interpretability* of TS-LDDMM for studying the inter-individual variability in a clinical dataset (Section 5).
- We illustrate the quantitative interest of such representation on classification tasks on real shape-based datasets with regular and irregular sampling (**??**).

2 Notations

We denote by integer ranges by $[k : l] = \{k, \dots, l\} \subset \mathcal{P}(\mathbb{Z})$ and $[l] = [1 : l]$ with $k, l \in \mathbb{N}$, by $\mathrm{C}^m(\mathsf{I}, \mathsf{E})$ the set of m-times continously differentiable function defined on an open set U to a normed vector space E, by $||u||_\infty = \inf_{x\in\mathsf{U}} |u(x)|$ for any bounded function $u : \mathsf{U} \to \mathsf{E}$, and by $\mathbb{N}_{>0}$ is the set of positive integers.

3 Background on LDDMM

In this part, we expose how to learn the diffeomorphisms $(\phi_j)_{j\in[N]}$ using LDDMM, initially introduced in [5]. In a nutshell, for any $j \in [N]$, ϕ_j corresponds to a differential flow related to a learnable velocity field belonging to a well-chosen Reproducing Kernel Hilbert Space (RKHS).

In the next section, the time series are going to be represented by diffeomorphism parameters $(\alpha_j)_{j\in[N]}$. That's why LDDMM is chosen since it offers a parametrization for diffemorphisms which is sparse and interpretable, two features particularly relevant in the biomedical context.

The basic problem that we consider in this section is the following. Given a set of targets $\mathbf{y} = (y_i)_{i\in[T_2]}$ in $\mathbb{R}^{d'}$[1], a set of starting points $\mathbf{x} = (x_i)_{i\in[T_1]}$ in $\mathbb{R}^{d'}$, we aim to find a diffeomorphism ϕ such that the finite set of points $\mathbf{y}$ is similar in a certain sense to the set of finite sets of transformed points $\phi \cdot \mathbf{x} = (\phi(x_i))_{i\in[T_1]}$. The function ϕ is occasionally referred to as a *deformation*. In general, these sets $\mathbf{x}, \mathbf{y}$ are meshes of continous objects, e.g surfaces, curves, images and so on.

Representing diffeomorpshims as deformations. Such *deformations* ϕ are constructed via differential flow equations, for any $x_0 \in \mathbb{R}^{d'}$ and $\tau \in [0, 1]$:

$$\frac{\mathrm{d}X(\tau)}{\mathrm{d}\tau} = v_\tau(X(\tau)), \quad X(0) = x_0\ , \phi^v_\tau(x_0) = X(\tau), \quad \phi^v = \phi^v_1\ , \tag{1}$$

where the velocity field is $v : \tau \in [0, 1] \mapsto v_\tau \in \mathsf{V}$ and V is a Hilbert space of continuously differentiable function on $\mathbb{R}^{d'}$. If $||\,\mathrm{d}u||_\infty + ||u||_\infty \leq ||u||_{\mathsf{V}}$ for any $u \in \mathsf{V}$ and $v \in \mathrm{L}^2([0, 1], \mathsf{V}) = \{v \in \mathrm{C}^0([0, 1], \mathsf{V}) : \int_0^1 ||v_\tau||^2_{\mathsf{V}}\,\mathrm{d}\tau < \infty\}$, by [20, Theorem 5] ϕ^v exists and belongs to $\mathcal{D}(\mathbb{R}^{d'})$, where we denote by $\mathcal{D}(\mathsf{O})$ the set of diffeomorpshim defined on an open set O to O. Therefore, for any choice of v, ϕ^v defines a valid deformation. This offers a general recipe to construct diffeomorphism given a functional space V.

With this in mind, the velocity field v fitting the data can be estimated by minimizing $v \in \mathrm{L}^2([0, 1], \mathsf{V}) \mapsto \mathscr{L}(\phi^v.\mathbf{x}, \mathbf{y})$, where $\mathscr{L}$ is an appropriate loss function. However, two computational challenges arise. First, this optimization problem is ill-posed, and a penalty term is needed to obtain a unique solution. In addition, we have to find a parametric family $\mathsf{V}_\Theta \subset \mathrm{L}^2([0, 1], \mathsf{V})$, parameterized by Θ, which allows us to solve this minimization problem efficiently.

It has been proposed in [38] to interpret V as a tangent space relative to the group of diffeomorphisms $\mathsf{H} = \{\phi^v : v \in \mathrm{L}^2([0, 1], \mathsf{V})\}$. Following this geometric point of view, geodesics can be constructed on H by using the following squared norm

$$\mathscr{R}^2 : g \in \mathsf{H} \mapsto \inf_{v\in\mathrm{L}^2([0,1],\mathsf{V}):\ g=\phi^v} \int_0^1 ||v_\tau||_{\mathsf{V}}\,\mathrm{d}\tau \tag{2}$$

By deriving differential constraints related to the minimum of (2) and using Cauchy lipshcitz conditions, geodesics can be defined only by giving the starting point and the initial velocity $v_0 \in \mathsf{V}$ [38], as straight lines in Euclidean space. Denoting by $w(v_0)$ the geodesic starting from the identity with inital velocity v_0, the exponential map is defined as $\varphi^{\{v_0\}} \triangleq \phi^v$ and the previous matching problem becomes a *geodesic shooting problem*:

$$\inf_{v_0\in\mathsf{V}} \mathscr{L}(\varphi^{\{v_0\}}.\mathbf{x}, \mathbf{y}). \tag{3}$$

Using $\varphi^{\{v_0\}}$ instead of ϕ^v for any $v \in \mathrm{L}^2([0, 1], \mathsf{V})$ regularizes the problem and induces a sparse representation for the learning diffeomorphisms. Moreover, by setting V as an RKHS, the geodesic shooting problem has a unique solution and becomes tractable, as described in the next section.

Discrete parametrization of diffeomorpshim. In this part, V is chosen as an RKHS [6] generated by a smooth kernel K (e.g., Gaussian). We follow [15] and define a discrete parameterization of the velocity fields to perform geodesics shooting (3). The initial velocity field v_0 is chosen as a finite linear combination of the RKHS basis vector fields, $\mathbf{n}_0$ control points $\mathsf{X}_0 = (x_{k,0})_{k\in[\mathbf{n}_0]} \in (\mathbb{R}^{d'})^{\mathbf{n}_0}$ and momentum vectors $\alpha_0 = (\alpha_{k,0})_{k\in[\mathbf{n}_0]} \in (\mathbb{R}^{d'})^{\mathbf{n}_0}$ are defined such that for any $x \in \mathbb{R}^{d'}$,

$$v_0\,(\alpha_0, \mathsf{X}_0)\,(x) = \sum_{k=1}^{\mathbf{n}_0} K(x, x_{k,0})\alpha_{k,0}\ . \tag{4}$$

[1] Note that we denote by $d' \in \mathbb{N}$ the ambient space

In our applications, the control points $(x_{k,0})_{k\in[\mathbf{n}_0]}$ can be understood as the discretized graph $(t_k, \mathbf{s}_0(t_k))_{k\in[\mathbf{n}_0]}$ of a starting time series $\mathbf{s}_0$. With this parametrization of v_0, **(author?)** [38] show that the velocity field v of the solution of (3) keeps the same structure along time, such that for any $x \in \mathbb{R}^{d'}$ and $\tau \in [0,1]$,

$$v_\tau(x) = \sum_{k=1}^{\mathbf{n}_0} K(x, x_k(\tau))\alpha_k(\tau) \ ,$$

$$\begin{cases} \dfrac{\mathrm{d}x_k(\tau)}{\mathrm{d}\tau} = v_\tau(x_k(\tau)) \ , \quad \dfrac{\mathrm{d}\alpha_k(\tau)}{\mathrm{d}\tau} = -\displaystyle\sum_{k=1}^{\mathbf{n}_0} \mathrm{d}_{x_k(\tau)} K(x_k(\tau), x_l(\tau))\alpha_l(\tau)^\top \alpha_k(\tau) \\ \alpha_k(0) = \alpha_{k,0}, \quad x_k(0) = x_{k,0} \ , k \in [\mathbf{n}_0] \end{cases} \tag{5}$$

These equations are derived from the hamiltonian $H : (\alpha_k, x_k)_{k\in[\mathbf{n}_0]} \mapsto \sum_{k,l=1}^{\mathbf{n}_0} \alpha_k^\top K(x_k, x_l)\alpha_l$, such that the velocity norm is preserved $||v_\tau||_{\mathsf{V}} = ||v_0||_{\mathsf{V}}$ for any $\tau \in [0,1]$. By (5), the velocity field related to a geodesic v^* is fully parametrized by its initial control points and momentum $(x_{k,0}, \alpha_{k,0})_{k\in[\mathbf{n}_0]}$. Thus, given a set of targets $\mathbf{y} = (y_i)_{i\in[T_2]}$ in $\mathbb{R}^{d'}$, a set of starting points $\mathbf{x} = (x_{i,0})_{i\in[T_1]}$ in $\mathbb{R}^{d'}$, a RKHS's kernel $K : \mathbb{R}^{d'} \times \mathbb{R}^{d'} \to \mathbb{R}^{d'\times d'}$, a distance on sets $\mathscr{L}$, a numerical integration scheme of ODE and a penalty factor $\lambda > 0$, the basic geodesic shooting step minimizes the following function using a gradient descent method:

$$\mathcal{F}_{\mathbf{x},\mathbf{y}} : (\alpha_k)_{k\in[T_1]} \mapsto \mathscr{L}\left(\varphi^{\{v_0\}}.\mathbf{x}, \mathbf{y}\right) + \lambda ||v_0||_{\mathsf{V}}^2 \ , \tag{6}$$

where v_0 is defined by (4) and $\varphi^{\{v_0\}}.\mathbf{x}$ is the result of the numerical integration of (5) using control points $\mathbf{x}$ and initial momentums $(\alpha_k)_{k\in[T_1]}$.

Relation to Continuous Normalizing Flows. One particular popular choice to address the problem of learning a diffeomorphism or a velocity field is Normalizing Flows [45, 30] (NF) or their continuous counterpart [11, 22, 46] (CNF). However, we do not rely on this class of learning algorithms for several reasons. Indeed, existing and simple normalizing flows are not suitable for the type of data that we are interested in this paper [17, 14]. In addition, they are primarily designed to have tractable Jacobian functions, while we do not require such property in our applications. Finally, the use of a differential flow solution of an ODE (1) trick is also at the basis of CNF, which then consists of learning a velocity field to address in fitting the data through a loss aiming to address the problem at hand. Nevertheless, the main difference between CNF and LDDMM lies in the parametrization of the velocity field. LDDMM uses kernels to derive closed form formula and enhance interpretability while NF and CNF take advantage of deep neural networks to scale with large dataset in high dimensions.

4 Methodology

We consider in this paper observations which consist in a population of N multivariate time series, for any $j \in [N]$, $s^j \in \mathrm{C}^1(\mathsf{I}_j, \mathbb{R}^d)$. However, we can only access a n_j-samples $\tilde{s}^j = (\tilde{s}_i^j = s^j(t_i^j))_{i\in[n_j]}$ collected at timestamps $(t_i^j)_{i\in[n_j]}$ for any $j \in [N]$. Note that **the number of samples n_j is not necessary the same accross individuals** and the timestamps can be **irregularly sampled**. We assume the time series population is globally homogeneous regarding their "shapes" even if inter-individual variability exists. Intuitively speaking, the "shape" of a time series $s : \mathsf{I} \to \mathbb{R}^d$ is encoded in its graphs $\mathsf{G}(s)$ defined as the set $\{(t, s(t)) : \ t \in \mathsf{I}\}$ and not only in its values $s(\mathsf{I}) = \{s(t) : \ t \in \mathsf{I}\}$ since the time axis is crucial. As a motivating use-case, s^j can be the time series of a heartbeat extracted from an individual's electrocardiogram (ECG), see Figure 2. The homogeneity in a resulting dataset comes from the fact that humans have similar shapes of heartbeat [57, 35].

The deformation problem. In this paper, we aim to study the inter-individual variability in the dataset by finding a relevant representation of each time series. Inspired from the framework of shape analysis [54], addressing similar problems in morphology, we suggest to represent each time series' graph $\mathsf{G}(s^j)$ as the transformation of a reference graph $\mathsf{G}(\mathbf{s}_0)$, related to a time series $\mathbf{s}_0 : \mathsf{I} \to \mathbb{R}^d$, by a diffeomorphism ϕ_j on $\mathbb{R}^{d+1}$, for any $j \in [N]$,

$$\phi_j.\mathsf{G}(\mathbf{s}_0) = \{\phi_j\left(t, \mathbf{s}_0(t)\right), \ t \in \mathsf{I}\} \ . \tag{7}$$

$\mathbf{s}_0$ will be understood as the typical representative shape common to the collection of time series $(s^j)_{j\in[N]}$. As $\mathbf{s}_0$ is supposed to be fixed, then the representation of the time series $(s^j)_{j\in[N]}$ boils down to the one of the transformation $(\phi_j)_{j\in[N]}$. We aim to learn $\mathsf{G}(\mathbf{s}_0)$ and $(\phi_j)_{j\in[N]}$.

Optimization related to (7). Defining the *discretized graphs* of the time series $(s^j)_{j\in[N]}$ and a discretization of the reference graph $\mathsf{G}(\mathbf{s}_0)$ as, for any $j \in [N]$,

$$\mathbf{y}_j = \mathsf{G}(\tilde{s}^j) = (t_i^j, \tilde{s}_i^j)_{i\in[n_j]} \in (\mathbb{R}^{d+1})^{n_j}, \quad \tilde{\mathsf{G}}_0 = (t_i^0, \tilde{s}_i^0)_{i\in[\mathbf{n}_0]} \in (\mathbb{R}^{d+1})^{\mathbf{n}_0} ,$$

with $\mathbf{n}_0 = \mathrm{median}((n_j)_{j\in[N]})$, the representation problem given in (7) boils down solving:

$$\mathrm{argmin}_{\tilde{\mathsf{G}}_0,(\alpha_k^j)_{k\in[\mathbf{n}_0]}^{j\in[N]}} \sum_{j=1}^{N} \mathcal{F}_{\tilde{\mathsf{G}}_0,\mathbf{y}_j}\left((\alpha_k^j)_{k\in[\mathbf{n}_0]}\right) , \tag{8}$$

which is carried out by a gradient descent on the control points $\tilde{\mathsf{G}}_0$ and the momentums $\alpha_j = (\alpha_k^j)_{k\in[\mathbf{n}_0]}$ for any $j \in [N]$, initialized by a dataset's time series graph of size $\mathbf{n}_0$ and by $0_{(d+1)\mathbf{n}_0}$ respectively. The optimization hyperparameter details are given in Appendix E.1. The result of the minimization $\tilde{\mathsf{G}}_0$ is then considered as the $\mathbf{n}_0$-samples of a common time series $\mathbf{s}_0$ and the momentums α_j encoding ϕ_j yields a feature vector in $\mathbb{R}^{d\mathbf{n}_0}$ of s^j for any $j \in [N]$. Finally, the vectors $(\alpha_j)_{j\in[N]}$ can be analyzed with any statistical or machine learning tools such as Principal Components Analysis (PCA), Latent Discriminant Analysis (LDA), longitudinal data analysis and so on.

Nevertheless, (8) ask to define a kernel and a loss in order to perform geodesic shooting 6, which is the purpose of the next subsection.

4.1 Application of LDDMM to time series analysis: TS-LDDMM

In this section, we present our theoretical contribution: we tailor the LDDMM framework to handle time series data. The reason is that applying a general diffeomorphism ϕ from $\mathbb{R}^{d+1}$ to a time series' graph $\mathsf{G}(s)$ can result in a set $\phi.\mathsf{G}(s)$ that does not correspond to the graph of any time series, as illustrated in the Figure 1. Thus, Time series graph have more structure than a simple 1D curve [21] and deserve their special analysis which will prove fruitful as demonstrated in 5.

To address this challenge, we need to identify an RKHS kernel $K : \mathbb{R}^{d+1} \times \mathbb{R}^{d+1} \to \mathbb{R}^{(d+1)^2}$ that generates deformations preserving the structure of the time series graph. This goal motivates us to clarify, in Theorem 1, the specific representation of diffeomorphisms we require before presenting a class of kernels that produce deformations with this representation.

Similarly, selecting a loss function on sets $\mathscr{L}$ that considers the temporal evolution in a time series' graph is crucial for meaningful comparisons with time series data. Consequently, we introduce the oriented Varifold distance.

A representation separating space and time. We prove that two time series graphs can always be linked by a time transformation composed of a space transformation. Moreover, a time series graph transformed by this kind of transformation is always a time series graph. We define $\Psi_\gamma \in \mathcal{D}(\mathbb{R}^{d+1}) : (t, x) \in \mathbb{R}^{d+1} \to (\gamma(t), x)$ for any $\gamma \in \mathcal{D}(\mathbb{R})$ and $\Phi_f : (t, x) \in \mathbb{R}^{d+1} \to (t, f(t, x))$ for any $f \in \mathrm{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$. We have the following representation theorem. All proofs are given in Appendix B.

Denote by $\mathsf{G}(s) \triangleq \{(t, s(t)) : \ t \in \mathsf{I}\}$ the graph of a time series $s : \mathsf{I} \to \mathbb{R}^d$ and $\phi.\mathsf{G}(s) \triangleq \{\phi(t, s(t)) : t \in \mathsf{I}\}$ the action of $\phi \in \mathcal{D}(\mathbb{R}^{d+1})$ on $\mathsf{G}(s)$.

Theorem 1. *Let $s : \mathsf{J} \to \mathbb{R}^d$ and $\mathbf{s}_0 : \mathsf{I} \to \mathbb{R}^d$ be two continuously differentiable time seriess with I, J two intervals of $\mathbb{R}$. There exist $f \in \mathrm{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ and $\gamma \in \mathcal{D}(\mathbb{R})$ such that $\gamma(\mathsf{I}) = \mathsf{J}$ and $\Phi_f \in \mathcal{D}(\mathbb{R}^{d+1})$,*

$$\mathsf{G}(s) = \Pi_{\gamma,f}.\mathsf{G}(\mathbf{s}_0), \ \Pi_{\gamma,f} = \Psi_\gamma \circ \Phi_f .$$

Moreover, for any $\bar{f} \in \mathrm{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ and $\bar{\gamma} \in \mathcal{D}(\mathbb{R})$, there exists a continously differentiable time series $\bar{s}$ such that $\mathsf{G}(\bar{s}) = \Pi_{\bar{\gamma},\bar{f}}.\mathsf{G}(\mathbf{s}_0)$

Remark 2. *that for any $\gamma \in \mathcal{D}(\mathbb{R})$ and $s \in \mathrm{C}^0(\mathsf{I}, \mathbb{R}^d)$,*

$$\{(\gamma(t), s(t)), \ t \in \mathsf{I}\} = \{(t, s \circ \gamma^{-1}(t)) : \ t \in \gamma(\mathsf{I})\} .$$

As a result, Ψ_γ can be understood as a temporal reparametrization and Φ_f encodes the transformation about the space.

Choice for the kernel associated with the RKHS V As depicted on Figure 1-2, we can not use any kernel K to apply the previous methodology to learn deformations on time series' graphs. We describe and motivate our choice in this paragraph. Denote the one-dimensional Gaussian kernel by $K_\sigma^{(a)}(x,y) = \exp(-|x-y|^2/\sigma)$ for any $(x,y) \in (\mathbb{R}^a)^2$, $a \in \mathbb{N}$ and $\sigma > 0$. To solve the geodesic shooting problem (6) on $\mathbb{R}^{d+1}$, we consider for V the RKHS associated with the kernel defined for any $(t,x),(t',x') \in (\mathbb{R}^{d+1})^2$:

$$K_{\mathsf{G}}((t,x),(t',x')) = \begin{pmatrix} c_0 K_{\text{time}} & 0 \\ 0 & c_1 K_{\text{space}} \end{pmatrix} , \tag{9}$$
$$K_{\text{space}} = K_{\sigma_{T,1}}^{(1)}(t,t') K_{\sigma_x}^{(d)}(x,x') \mathrm{I}_d \; , K_{\text{time}} = K_{\sigma_{T,0}}^{(1)}(t,t') \; ,$$

parametrized by the widths $\sigma_{T,0}, \sigma_{T,1}, \sigma_x > 0$ and the constants $c_0, c_1 > 0$. This choice for K_{G} is motivated by the representation Theorem 1 and the following result.

Lemma 1. *If we denote by V the RKHS associated with the kernel K_{G}, then for any vector field v generated by* (5) *with v_0 satisfying* (4)*, there exist $\gamma \in \mathsf{D}(\mathbb{R})$ and $f \in \mathrm{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ such that $\phi^v = \Psi_\gamma \circ \Phi_f$.*

Instead of Gaussian kernels, other types of smooth kernel can be selected as long as the structure (9) is respected.

Remark 3. *With this choice of kernel, the features associated to the time transformation can be extracted from the momentums $(\alpha_{k,0})_{k\in[\mathbf{n}_0]} \in (\mathbb{R}^{d+1})^{\mathbf{n}_0}$ in* (4) *by taking the coordinates related to time. However, the features related to the space transformation are not only in the space coordinates since the related kernel K_{space} depends on time as well.*

In Appendix D, we give guidelines for selecting the hyperparameters $(\sigma_{T,0}, \sigma_{T,1}, \sigma_x, c_0, c_1)$.

Loss This section specifies the distance function $\mathscr{L}$ introduced in the loss function defined in (6).

In practice, we can only access discretized graphs of time series, $(t_i^j, \tilde{s}_i^j)_{i\in[n_j]}$ for any $j \in [N]$, that are potentially of different sizes n_j and sampled at different timestamps $(t_i^j)_{i\in[n_j]}$ for any $j \in [N]$. Usual metrics, such as the Euclidean distance, are not appealing as they make the underlying assumptions of equal size sets and the existence of a pairing between points. Distances between measures on sets (taking the empirical distribution), such as Maximum Mean Discaprency (MMD) [16, 7], alleviate those issues; however, MMD only accounts for positional information and lacks information about the time evolution between sampled points. A classical data fidelity metric from shape analysis corresponding to the distance between *oriented varifolds* associated with curves alleviates this last issue [28]. Intuitively, an oriented varifold is a measure that accounts for positional and tangential information about the underlying curves at sample points. More details and information about *oriented varifolds* can be found in Appendix C.

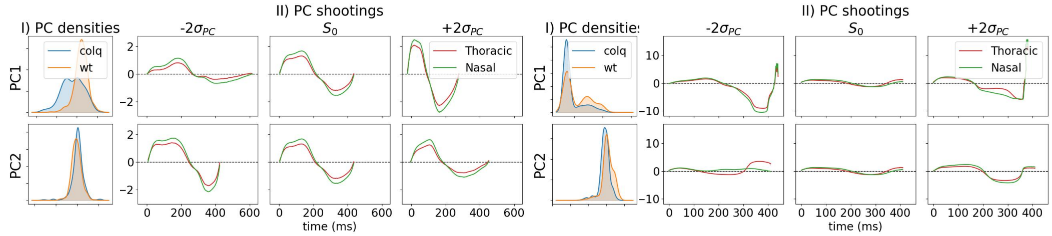
More precisely, given two sets $\mathsf{G}_0 = (g_i^0)_{i\in[T_0]}, \mathsf{G}_1 = (g_i^1)_{i\in[T_1]} \in (\mathbb{R}^{d+1})^{T_1}$ and a kernel[2] $k : (\mathbb{R}^{d+1} \times \mathbb{S}^d)^2 \to \mathbb{R}$ verifying [28, Proposition 2 & 4], for any $\xi \in \{0,1\}$ and $i \in [T_\xi - 1]$, denoting the center and length of the i^{th} segment $[g_i^\xi, g_{i+1}^\xi]$ by $c_i^\xi = (g_i^\xi + g_{i+1}^\xi)/2$, $l_i^\xi = \|g_{i+1}^\xi - g_i^\xi\|$, and $\overrightarrow{v_i}^\xi = (g_{i+1}^\xi - g_i^\xi)/l_i^\xi$, the varifold distance between G_0 and G_1 is defined as,

$$d_{\mathsf{W}^*}^2(\mathsf{G}_0, \mathsf{G}_1) = \sum_{i,j=1}^{T_0-1} l_i^0 k((c_i^0, \overrightarrow{v_i}^0), (c_j^0, \overrightarrow{v_j}^0)) l_j^0 - 2 \sum_{i=1}^{T_0-1} \sum_{j=1}^{T_1-1} l_i^0 k((c_i^0, \overrightarrow{v_i}^0), (c_j^1, \overrightarrow{v_j}^1)) l_j^1$$
$$+ \sum_{i,j=1}^{T_1-1} l_i^1 k((c_i^1, \overrightarrow{v_i}^1), (c_j^1, \overrightarrow{v_j}^1)) l_j^1$$

In practice, we set the kernel k as the product of two anisotropic Gaussian kernels, k_{pos} and k_{dir}, such that for any $(x, \overrightarrow{u}), (y, \overrightarrow{v}) \in (\mathbb{R}^{d+1} \times \mathbb{S}^d)^2$

$$k((x, \overrightarrow{u}), (y, \overrightarrow{v})) = k_{\text{pos}}(x,y) k_{\text{dir}}(\overrightarrow{u}, \overrightarrow{v}) \; .$$

[2] $\mathbb{S}^d = \{x \in \mathbb{R}^{d+1} : \; |x| = 1\}$



(a) TS-LDDMM

(b) LDDMM

Figure 3: Analysis of the two principal components (PC) related to mice' respiratory cycles before exposure for TS-LDDMM Figure 3a, and LDDMM Figure 3b. In both cases, I) displays PC densities according to mice genotype and II) displays deformations of the reference graph S_0 along each PC.

Note that the loss kernel k has nothing to do with the velocity field kernel denoted by K_{G} or K specified in Section 4.1. Finally, we define the data fidelity loss function, $\mathscr{L}$, as a sum of $d^2_{\mathsf{W}^*}$ using different kernel's width parameters σ to incorporate multiscale information such that. $\mathscr{L}$ is indeed differentiable with regards to its first variable. The specific kernels $k_{\mathrm{pos}}, k_{\mathrm{dir}}$ that we use in our experiments are given Appendix C.1. For further readings on curves and surfaces representation as varifolds, readers can refer to [28, 10].

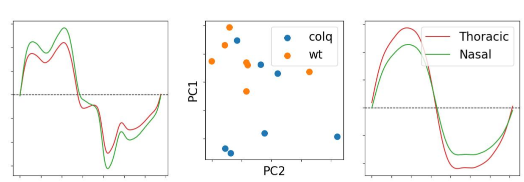
5 Experiments

For consiceness, several experiments are relagated in appendix:

1. **TS-LDDMM representation identifiability, ??:** On synthetic data, we evaluate the ability of our method to retrieve the parameter v_0^* that encodes the deformation $\varphi^{\{v_0^*\}}$ acting on a time series graph G by solving the geodesic shooting problem (6) between G and $\varphi^{\{v_0^*\}}.\mathsf{G}$. **Results** show that TS-LDDMM representations are identifiable or weakly identifiable depending on the velocity field kernel K_G specification.
2. **Robustness to irregular sampling** We compare the robustness of TS-LDDMM representation with 9 URL methods handling irregularly sampled multivariate time series on 15 shape-based datasets (7 univariates & 8 multivariates). We assess methods' classification performances under regular sampling (0% missing rate) and three irregular sampling regimes (30%, 50%, and 70% missing rates), according to the protocol depicted in [29]. **Results** show that our method, TS-LDDMM, outperforms all methods for sampling regimes with missing rates: 0%, 30%, and 50%.
3. **Classification benchmark on regularly sampled datasets, ??:** We compare performances of a kernel support vector machine (SVC) algorithm based on TS-LDDMM representation with 3 state-of-the-art classification methods from shape analysis on 15 shape-based datasets (7 univariates & 8 multivariates). **Results** show that the TS-LDDMM-based method outperforms other methods (best performances over 13 datasets), making TS-LDDMM representation relevant for time series shape analysis.

5.1 Interpretability: analysis of respiratory behavior in mice

This experiment highlights the *interpretability* of TS-LDDMM representation for studying the inter-individual variability in time series biomedical datasets. We consider a multivariate time series dataset that accounts for mice's nasal and thoracic airflow evolution when exposed to an irritant molecule altering respiratory functions [39]. The dataset is divided into two groups, one composed of 7 control mice (**wt**) and the other of 7 mice (**colq**) deficient in an enzyme involved in the control of respiration. For each mouse,



(a) colq cycle (b) PC1 vs PC2 (c) wt cycle

Figure 4: Figure 4a is a **colq** respiratory cycle. Figure 4b displays individual mouse reference respiratory cycle in the TS-LDDMM PC1-PC2 coordinates. Figure 4c is a referent respiratory cycle of a **wt** mouse learned with TS-LDDMM.

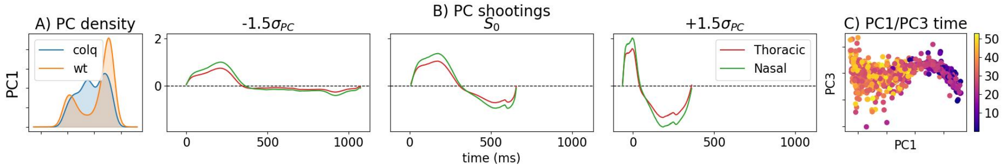


Figure 5: Analysis of the first Principal Component (PC1) related to mice' respiratory cycles before and after exposure for TS-LDDMM. Figure A) displays PC densities according to mice genotype, Figure B) displays deformations of the reference graph S_0 along PC1, and Figure C) displays respiratory cycles with respect to time in PC1 and PC3 coordinates

airflows were recorded for 15 to 20 minutes before exposure to the irritant molecule and then for 35 to 40 minutes. A complete description of the dataset is given in the Appendix F.1.

Result summary. By comparing the shape of individual respiratory cycles, we show that TS-LDDMM features can encode genotype distinctive breathing behaviors and their evolution after exposure to the irritant molecule contrary to LDDMM [21].

Protocol. For both experiments and representations (TS-LDDMM and LDDMM), we derive the reference respiratory cycle's graph S_0 and the representations $(\alpha_j)_{j\in[N]}$ related to N respiratory cycles extracted according to the procedure [19] by solving (8). Then, we perform a kernel PCA on the initial velocity field $(v_0(\alpha_j, S_0))_{j\in[N]} \in \mathsf{V}^{N_1}$ defined in (4) to breathing behavior. The first experiment includes $N_1 = 700$ respiratory cycles collected before exposure. The second experiment includes $N_2 = 1400$ respiratory cycles with 25% (resp. 75%) before (resp. after) exposure. Appendix G.1 describes methods settings. Essentially, varifold losses are identical for both methods, and the velocity field kernels are set to encompass time and space scales.

Breathing behaviors before exposure. We focus on the analysis of the two first Principal Components (PC) for TS-LDDMM (Figure 3a), and LDDMM (Figure 3b). A respiratory cycle is an inspiration (positive flow) followed by an expiration (negative flow); Figure 4c displays an example of **wt** mouse respiratory cycle. Figure 3 shows that principal components learned with TS-LDMM lead to deformations that remain respiratory cycles, while deformations learned with LDDMM are challenging to interpret as respiratory cycles. The LDDMM velocity field kernel is a Gaussian anisotropic kernel that accounts for time and space scales; however, the entanglement of time and space dimensions in the kernel does not guarantee the graph structure, and it makes the convergence of the method difficult (relative varifold loss error: TS-LDDMM: 0.06, LDDMM: 0.11). Regarding TS-LDDMM, Figure 3a shows its principal components refer to different types of deformations. By interpreting Figure 3a: PC1 accounts for time warping, PC2 expresses the trade-off between inspiration and expiration duration. Compared to **wt** mice, the distribution of **colq** mice feature along the PC1 axis has a heavy left tail, and the associated deformation (-2 σ_{PC}) shows an inspiration with two peaks. Figure 4a shows an example of such respiratory cycles, which may be caused by motor impairment due to enzyme deficiency [19]. Finally, Figure 4b shows that PC1 and PC2 capture the main differences between the two groups as their respective reference graphs S^j are located in different parts of the space.

Breathing behaviors' evolution after exposure to the irritant molecule TS-LDDMM representation features are learned on a dataset that includes respiratory cycles before and after exposure. Figure 5 displays the first principal component PC since it encodes the effect of the irritant molecule as depicted in Figure 5-C) (the exposure occurs at 20 minutes). Figure 5-B) shows that the deformation (-1.5 σ_{PC}) leads to longer respiratory cycles that include a pause between inspiration and expiration as observed in [19]. Conjointly, Figure 5-A) shows a bimodal distribution for **wt** mice whereas **wt**' distributions before exposure were unimodal (Figure 3a). Indeed, the irritant molecule inhibits the action of the deficient enzyme, **wt** mice strongly react to the irritant molecule, whereas **colq** mice better adapt due to their deficiency [19].

6 Related Works

Shape analysis focuses on statistical analysis of mathematical objects invariant under some deformations like rotations, dilations, or time parameterization. The main idea is to represent these objects in a complete Riemannian manifold $(\mathcal{M}, \mathbf{g})$ with a metric $\mathbf{g}$ adapted to the geometry of the problem [38]. Then, any set of points in $\mathcal{M}$ can be represented as points in the tangent space of their Frechet mean $\mathbf{m}_0$ [42, 31] by considering their logarithms. The goal is to find a well-suited Riemannian structure according to the nature of the studied object.

LDDMM framework is a relevant shape analysis tool to represent curves as depicted in [21]. However, graphs of time series are a well-structured type of curve due to the inclusion of the temporal dimension that requires specific care (Figure 1). In a similar vein, Qiu *et al* [44] proposes a method for tracking anatomical shape changes in serial images using LDDMM. They include temporal evolution, but not for the same purpose: the aim is to perform longitudinal modeling of brain images.

Leaving the LDDMM representation, the results of [50, 24] addresses the representation of curves with the Square-Root Velocity (SRV) representation. However, the SRV representation is applied after a reparametrization of the temporal dimension on the unit length segment. Consequently, the graph structure of the time series is not respected, and the original time evolution of the time series is not encoded in the final representation. Very recently, in a functional data analysis framework, a paper [56] (Shape-FPCA) improved by representing the original time evolution. Nevertheless, this method is made for *continuous objects* and only applies to time series of *same length*, making the estimation more sensitive to noise.

Balancing between discrete and continuous elements is a challenging task. In the deep learning literature [11, 29, 53, 27, 34, 1], Neural Ordinary Differential Equations (Neural ODEs) [11] learn continuous latent representations using a vector field parameterized by a neural network, serving as a continuous analogue to Residual Networks [59]. This approach was further enhanced by Neural Controlled Differential Equations (Neural CDEs) [29] for handling irregular time series, functioning as continuous-time analogs of RNNs [47]. Extending Neural ODEs, Neural Stochastic Differential Equations (Neural SDEs) introduce regularization effects [34], although optimization remains challenging. Leveraging techniques from continuous-discrete filtering theory, Ansari et al. [1] applied succesfully Neural SDEs to irregular time series. Oh *et al.* [41] improved these results by incorporating the concept of controlled paths into the drift term, similar to how Neural CDEs outperform Neural ODEs. With TS-LDDMM, the representation is also derived from an ODE, but the velocity field is parameterized with kernels and optimized to have a minimal norm, which enhance interpretability.

All these state-of-the-art methods previously mentionned [21, 41, 56, 24] are comapred to TS-LDDMM in **??**.

7 Limitations

While TS-LDDMM performs well on shape-based datasets, it assumes a certain degree of homogeneity with the existence of a reference time series graphG_0. Extending TS-LDDMM to more heterogeneous datasets using clustering is ongoing work. TS-LDDMM employs kernel computations, which require specific libraries (e.g., KeOps [9]) to be efficient and scalable. However, in our experiments, the time complexity of TS-LDDMM is comparable to that of competitors. It is clear that TS-LDDMM needs to be extended to handle very large datasets with high-dimensional time series (such as videos). Additionally, TS-LDDMM requires tuning several hyperparameters, though this is a common requirement among competitors [21, 41, 56, 24]. In future work, adaptive methods are expected to be developed to provide a more user-friendly interface.

8 Conclusion

In this paper, we propose a feature representation method, TS-LDDMM, designed for shape comparison in homogeneous time series datasets. We show on a real dataset its ability to study, with high interpretability, the inter-individual shape variability. As an unsupervised approach, it is user-friendly and enables knowledge transfer for different supervised tasks such as classification. Although TS-LDDMM is already competitive for classification, its performances can be leveraged on more heterogeneous datasets using a hierarchical clustering extension, which is relagated for future work.

References

[1] Abdul Fatir Ansari, Alvin Heng, Andre Lim, and Harold Soh. Neural continuous-discrete state space models for irregularly-sampled time series. In *International Conference on Machine Learning*, pages 926–951. PMLR, 2023.

[2] Asal Asgari. Clustering of clinical multivariate time-series utilizing recent advances in machine-learning. 2023.

[3] 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.

[4] Ziv Bar-Joseph, Anthony Gitter, and Itamar Simon. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8):552–564, 2012.

[5] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61:139–157, 2005.

[6] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[7] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[8] Claudio Carmeli, Ernesto De Vito, Alessandro Toigo, and Veronica Umanitá. Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61, 2010.

[9] Benjamin Charlier, Jean Feydy, Joan Alexis Glaunes, François-David Collin, and Ghislain Durif. Kernel operations on the gpu, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74):1–6, 2021.

[10] Nicolas Charon and Alain Trouvé. The varifold representation of nonoriented shapes for diffeomorphic registration. *SIAM journal on Imaging Sciences*, 6(4):2547–2580, 2013.

[11] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[12] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[13] 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.

[14] Ruizhi Deng, Bo Chang, Marcus A Brubaker, Greg Mori, and Andreas Lehrmann. Modeling continuous stochastic processes with dynamic normalizing flows. *Advances in Neural Information Processing Systems*, 33:7805–7815, 2020.

[15] Stanley Durrleman, Stéphanie Allassonnière, and Sarang Joshi. Sparse adaptive parameterization of variability in image ensembles. *International Journal of Computer Vision*, 101:161–183, 2013.

[16] Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

[17] Shibo Feng, Chunyan Miao, Ke Xu, Jiaxiang Wu, Pengcheng Wu, Yang Zhang, and Peilin Zhao. Multi-scale attention flow for probabilistic time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[18] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

[19] Thibaut Germain, Charles Truong, Laurent Oudre, and Eric Krejci. Unsupervised classification of plethysmography signals with advanced visual representations. *Frontiers in Physiology*, 14:781, 2023.

[20] Joan Glaunes. Transport par difféomorphismes de points, de mesures et de courants pour la comparaison de formes et l'anatomie numérique. *These de sciences, Université Paris*, 13, 2005.

[21] Joan Glaunes, Anqi Qiu, Michael I Miller, and Laurent Younes. Large deformation diffeomorphic metric curve mapping. *International journal of computer vision*, 80:317–336, 2008.

[22] Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, and David Duvenaud. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*, page 7, 2019.

[23] Ella Guscelli, John I Spicer, and Piero Calosi. The importance of inter-individual variation in predicting species' responses to global change drivers. *Ecology and Evolution*, 9(8):4327–4339, 2019.

[24] Tae-Young Heo, Joon Myoung Lee, Myung Hun Woo, Hyeongseok Lee, and Min Ho Cho. Logistic regression models for elastic shape of curves based on tangent representations. *Journal of the Korean Statistical Society*, pages 1–19, 2024.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] Heinz Gerd Hoymann. Lung function measurements in rodents in safety pharmacology studies. *Frontiers in pharmacology*, 3:156, 2012.

[27] Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Irene Kaltenmark, Benjamin Charlier, and Nicolas Charon. A general framework for curve and surface comparison and registration with oriented varifolds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3346–3355, 2017.

[29] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707, 2020.

[30] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

[31] Huiling Le. Locating fréchet means with application to shape spaces. *Advances in Applied Probability*, 33(2):324–338, 2001.

[32] Mathias Lechner and Ramin Hasani. Learning long-term dependencies in irregularly-sampled time series. *arXiv preprint arXiv:2006.04418*, 2020.

[33] Yurim Lee, Eunji Jun, Jaehun Choi, and Heung-Il Suk. Multi-view integrative attention-based deep representation learning for irregular clinical time-series data. *IEEE Journal of Biomedical and Health Informatics*, 26(8):4270–4280, 2022.

[34] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*, 2019.

[35] Putri Madona, Rahmat Ilias Basti, and Muhammad Mahrus Zain. Pqrst wave detection on ecg signals. *Gaceta Sanitaria*, 35:S364–S369, 2021.

[36] Larry Medsker and Lakhmi C Jain. *Recurrent neural networks: design and applications*. CRC press, 1999.

[37] Qianwen Meng, Hangwei Qian, Yong Liu, Yonghui Xu, Zhiqi Shen, and Lizhen Cui. Unsupervised representation learning for time series: A review. *arXiv preprint arXiv:2308.01578*, 2023.

[38] Michael I Miller, Alain Trouvé, and Laurent Younes. Geodesic shooting for computational anatomy. *Journal of mathematical imaging and vision*, 24:209–228, 2006.

[39] Aurélie Nervo, André-Guilhem Calas, Florian Nachon, and Eric Krejci. Respiratory failure triggered by cholinesterase inhibitors may involve activation of a reflex sensory pathway by acetylcholine spillover. *Toxicology*, 424:152232, 2019.

[40] Vit Niennattrakul and Chotirat Ann Ratanamahatana. Inaccuracies of shape averaging method using dynamic time warping for time series data. In *Computational Science–ICCS 2007: 7th International Conference, Beijing, China, May 27-30, 2007, Proceedings, Part I 7*, pages 513–520. Springer, 2007.

[41] YongKyung Oh, Dongyoung Lim, and Sungil Kim. Stable neural stochastic differential equations in analyzing irregular time series data. In *The Twelfth International Conference on Learning Representations*, 2024.

[42] Susovan Pal, Roger P Woods, Suchit Panjiyar, Elizabeth Sowell, Katherine L Narr, and Shantanu H Joshi. A riemannian framework for linear and quadratic discriminant analysis on the tangent space of shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 47–55, 2017.

[43] John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1855–1870, 2015.

[44] Anqi Qiu, Marilyn Albert, Laurent Younes, and Michael I Miller. Time sequence diffeomorphic metric mapping and parallel transport track time-dependent shape changes. *NeuroImage*, 45(1):S51–S60, 2009.

[45] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[46] Hadi Salman, Payman Yadollahpour, Tom Fletcher, and Kayhan Batmanghelich. Deep diffeomorphic normalizing flows. *arXiv preprint arXiv:1810.03256*, 2018.

[47] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[48] Gota Shirato, Natalia Andrienko, and Gennady Andrienko. Identifying, exploring, and interpreting time series shapes in multivariate time intervals. *Visual Informatics*, 7(1):77–91, 2023.

[49] Satya Narayan Shukla and Benjamin M Marlin. Multi-time attention networks for irregularly sampled time series. *arXiv preprint arXiv:2101.10318*, 2021.

[50] Anuj Srivastava, Eric Klassen, Shantanu H Joshi, and Ian H Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE transactions on pattern analysis and machine intelligence*, 33(7):1415–1428, 2010.

[51] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

[52] Patara Trirat, Yooju Shin, Junhyeok Kang, Youngeun Nam, Jihye Na, Minyoung Bae, Joeun Kim, Byunghyun Kim, and Jae-Gil Lee. Universal time-series representation learning: A survey. *arXiv preprint arXiv:2401.03717*, 2024.

[53] Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

[54] Marc Vaillant, Michael I Miller, Laurent Younes, and Alain Trouvé. Statistics on diffeomorphisms via tangent space representations. *NeuroImage*, 23:S161–S169, 2004.

[55] Kai Wang, Youjin Zhao, Qingyu Xiong, Min Fan, Guotan Sun, Longkun Ma, Tong Liu, et al. Research on healthy anomaly detection model based on deep learning from multiple time-series physiological signals. *Scientific Programming*, 2016, 2016.

[56] Yuexuan Wu, Chao Huang, and Anuj Srivastava. Shape-based functional data analysis. *TEST*, 33(1):1–47, 2024.

[57] Can Ye, BVK Vijaya Kumar, and Miguel Tavares Coimbra. Heartbeat classification using morphological and dynamic features of ecg signals. *IEEE Transactions on Biomedical Engineering*, 59(10):2930–2941, 2012.

[58] Lexiang Ye and Eamonn Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956, 2009.

[59] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[60] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in neural information processing systems*, 33:18795–18806, 2020.

A Societal impact

B Proofs

Denote by $\mathsf{G}(s) \triangleq \{(t, s(t)) : \ t \in \mathsf{I}\}$ the graph of a time series $s : \mathsf{I} \to \mathbb{R}^d$ and $\phi.\mathsf{G}(s) \triangleq \{\phi(t, s(t)) : t \in \mathsf{I}\}$ the action of $\phi \in \mathcal{D}(\mathbb{R}^{d+1})$ on $\mathsf{G}(s)$.

Theorem 4. *Let $s : \mathsf{J} \to \mathbb{R}^d$ and $\mathrm{s}_0 : \mathsf{I} \to \mathbb{R}^d$ be two continuously differentiable time seriess with I, J two intervals of $\mathbb{R}$. There exist $f \in \mathrm{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ and $\gamma \in \mathcal{D}(\mathbb{R})$ such that $\gamma(\mathsf{I}) = \mathsf{J}$ and $\Phi_f \in \mathcal{D}(\mathbb{R}^{d+1})$,*

$$\mathsf{G}(s) = \Pi_{\gamma,f}.\mathsf{G}(\mathrm{s}_0), \ \Pi_{\gamma,f} = \Psi_\gamma \circ \Phi_f .$$

Moreover, for any $\bar{f} \in \mathrm{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ and $\bar{\gamma} \in \mathcal{D}(\mathbb{R})$, there exists a continously differentiable time series $\bar{s}$ such that $\mathsf{G}(\bar{s}) = \Pi_{\bar{\gamma},\bar{f}}.\mathsf{G}(\mathrm{s}_0)$

Proof. Let $s : \mathsf{J} \to \mathbb{R}^d$ and $\mathrm{s}_0 : \mathsf{I} \to \mathbb{R}^d$ be two continuously differentiable time seriess with $\mathsf{I} = (a, b), \mathsf{J} = (\alpha, \beta)$ two intervals of $\mathbb{R}$. By setting $\gamma : t \in \mathbb{R} \mapsto (\beta - \alpha)(t - a)/(b - a) + \alpha \in \mathbb{R}$, we have $\gamma(\mathsf{I}) = \mathsf{J}$ and $\gamma \in \mathcal{D}(\mathbb{R})$. By defining $f : (t, x) \in \mathbb{R}^{d+1} \mapsto x - \mathrm{s}_0(t) + s \circ \gamma(t)$, the map $\Phi_f \in \mathcal{D}(\mathbb{R}^{d+1})$, indeed, its inverse is $\Phi_f^{-1} : (t, x) \in \mathbb{R}^{d+1} \mapsto (t, x + \mathrm{s}_0(t) - s(t))$ and is continuously differentiable. Moreover, we have $\Pi_{\gamma,f}.\mathsf{G}(\mathrm{s}_0) = \{(\gamma(t), s \circ \gamma(t)) : \ t \in \mathsf{I}\} = \mathsf{G}(s)$.

Let $\bar{f} \in \mathrm{C}^0(\mathbb{R}^{d+1}, \mathbb{R}^d)$, $\bar{\gamma} \in \mathcal{D}(\mathbb{R})$ and $\mathrm{s}_0 \in \mathrm{C}^0(\mathsf{I}, \mathbb{R}^d)$ with I an interval of $\mathbb{R}$. We have :

$$\begin{aligned} \Pi_{\gamma,f}.\mathsf{G}(\mathrm{s}_0) &= \{(\gamma(t), f(t, \mathrm{s}_0(t))), \ t \in \mathsf{I}\} \\ &= \left\{\left(t, f\left(\gamma^{-1}(t), \mathrm{s}_0(\gamma^{-1}(t))\right)\right), \ t \in \gamma(\mathsf{I}) \right\} . \end{aligned} \tag{10}$$

By defining $\bar{s} : t \in \gamma(\mathsf{I}) \to f\left(\gamma^{-1}(t), \mathrm{s}_0(\gamma^{-1}(t))\right)$, we have $\bar{s} \in \mathrm{C}^0(\gamma(\mathsf{I}), \mathbb{R}^d)$ by composition of continuous functions and $\mathsf{G}(\bar{s}) = \Pi_{\gamma,f}.\mathsf{G}(\mathrm{s}_0)$ by (10), which concludes the proof. □

Lemma 2. *If we denote by V the RKHS associated with the kernel K_{G}, then for any vector field v generated by* (5) *with v_0 satisfying* (4)*, there exist $\gamma \in \mathsf{D}(\mathbb{R})$ and $f \in \mathrm{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ such that $\phi^v = \Psi_\gamma \circ \Phi_f$.*

Proof. Let v be a vector field generated by (5) with v_0 satisfying (4). We remark that the first coordinate of the velocity field v_τ denoted by v_τ^{time} only depends on the time variable t for any $\tau \in [0, 1]$. Thus, when computing the first coordinate of the deformation ϕ^v, denoted by γ, we integrate (1) with v_τ replaced by v_τ^{time}, thus γ is independant of the variable x. Moreover, $\gamma \in \mathcal{D}(\mathbb{R})$

since a Gaussian kernel induced an Hilbert space V satisfying $|f|_V \leq |f|_\infty + |\,\mathrm{d}f|_\infty$ for any $f \in \mathsf{V}$ by [20, Theorem 9]. For the same reason, we have $\phi^v \in \mathcal{D}(\mathbb{R}^{d+1})$, and thus its last coordinates denoted by f belongs to $\mathrm{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$, and by construction $\phi^v = \Psi_\gamma \circ \Phi_f$. □

C Oriented varifold

In this section, we introduce the *oriented varifold* associated with curves. For further readings on curves and surfaces representation as varifolds, readers can refer to [28, 10]. We associate to $\gamma \in \mathrm{C}^1((a,b), \mathbb{R}^{d+1})$ an *oriented varifold* μ_γ, i.e. a distribution on the space $\mathbb{R}^{d+1} \times \mathbb{S}^d$ defined as follows, for any smooth test function $\omega : \mathbb{R}^{d+1} \times \mathbb{S}^d \to \mathbb{R}$,

$$\mathbb{E}_{Y \sim \mu_\gamma}\left[\omega(Y)\right] = \mu_\gamma(\omega) = \int_a^b \omega\left(\gamma(t), \frac{\dot\gamma(t)}{|\dot\gamma(t)|}\right) |\dot\gamma(t)| \,\mathrm{d}t \,.$$

Denoting by W the space of smooth test function, we have that μ_γ belongs to its dual W^*. Thus, a distance on W^* is sufficient to set a distance on oriented varifolds associated to curve and thus on $\mathrm{C}^1((a,b), \mathbb{R}^{d+1})$ by the identification $\gamma \to \mu_\gamma$. Remark that in (TS-LDDMM), γ should be the parametrization of a time series' graph $\mathsf{G}(s)$, i.e. $\gamma : t \in \mathsf{I} \to (t, s(t)) \in \mathbb{R}^{d+1}$ denoting by $s : \mathsf{I} \to \mathbb{R}^d$ the time series. However, in practice, we work with discrete objects. That is why, we set W as an RKHS to use its representation theorem. More specifically [28, Proposition 2 & 4] encourages us to consider a kernel $k : (\mathbb{R}^{d+1} \times \mathbb{S}^d)^2 \to \mathbb{R}$ such that there exist two positive and continuously differentiable kernels k_{pos} and k_{dir}, such that for any $(x, \overrightarrow{u}), (y, \overrightarrow{v}) \in (\mathbb{R}^{d+1} \times \mathbb{S}^d)^2$

$$k((x, \overrightarrow{u}), (y, \overrightarrow{v})) = k_{\mathrm{pos}}(x, y) k_{\mathrm{dir}}(\overrightarrow{u}, \overrightarrow{v}) \,,$$

with moreover $k_{\mathrm{dir}} > 0$ and k_{pos} which admits an RKHS $\mathsf{W}_{\mathrm{pos}}$ dense in the space of continous function on $\mathbb{R}^{d+1}$ vanishing at infinite [8].

Given such a kernel $k : (\mathbb{R}^{d+1} \times \mathbb{S}^d)^2 \to \mathbb{R}$ verifying [28, Proposition 2 & 4], we have that for any $(x, v) \in \mathbb{R}^{d+1} \times \mathbb{S}^d$, $\delta_{(x, \overrightarrow{v})}$ belongs to W^* as a distribution and that the dual metric $\langle \cdot, \cdot \rangle_{\mathsf{W}^*}$ satisfies for any $(x_1, v_1), (x_2, v_2) \in \left(\mathbb{R}^{d+1} \times \mathbb{S}^d\right)^2$,

$$\langle \delta_{(x_1, \overrightarrow{v}_1)}, \delta_{(x_2, \overrightarrow{v}_2)} \rangle_{\mathsf{W}^*} = k((x_1, \overrightarrow{v}_1), (x_2, \overrightarrow{v}_2)) \,.$$

Thus, given two sets of triplets $X = (l_i, x_i, \overrightarrow{v}_i)_{i \in [T_0 - 1]} \in (\mathbb{R} \times \mathbb{R}^{d+1} \times \mathbb{S}^d)^{T_0 - 1}, Y = (l'_i, y_i, \overrightarrow{w}_i)_{i \in [T_1]} \in (\mathbb{R} \times \mathbb{R}^{d+1} \times \mathbb{S}^d)^{T_1 - 1}$ and denoting by

$$\mu_X = \sum_{i=1}^{T_0} l_i \delta_{(x_i, \overrightarrow{v}_i)}, \mu_Y = \sum_{i=1}^{T_1} l'_i \delta_{(y_i, \overrightarrow{w}_i)} \,, \tag{11}$$

we have,

$$|\mu_X - \mu_Y|^2_{\mathsf{W}^*} = \sum_{i,j=1}^{T_0-1} l_i k((x_i, \overrightarrow{v_i}), (x_i, \overrightarrow{v_i}^0)) l_j - 2 \sum_{i=1}^{T_0-1} \sum_{j=1}^{T_1-1} l_i k((x_i, \overrightarrow{v_i}), (y_i, \overrightarrow{w_i})) l'_j + \sum_{i,j=1}^{T_1-1} l'_i k((y_i, \overrightarrow{w_i}), (y_i, \overrightarrow{w_i})) l'_j \,.$$

Then, using the identification $X \to \mu_X, Y \to \mu_Y$, we can define a distance on sets of triplets as $d_{\mathsf{W}^*,3}(X, Y) = |\mu_X - \mu_Y|^2_{\mathsf{W}^*}$.

Now, we aim to discretize the oriented varifold μ_{G} related to a time series' graph $\mathsf{G}(s)$ by using a set of triplets. This is carried out by using a discretized version of $\mathsf{G}(s)$, i.e. $\tilde{\mathsf{G}} = (g_i = (t_i, s(t_i)))_{i \in [T]} \in (\mathbb{R}^{d+1})^T$, in the following way: For any $i \in [T-1]$, denoting the center and length of the i^{th} segment $[g_i, g_{i+1}]$ by $c_i = (g_i + g_{i+1})/2$, $l_i = \|g_{i+1} - g_i\|$, and the unit norm vector of direction $\overrightarrow{g_i g_{i+1}}$ by $\overrightarrow{v_i} = (g_{i+1} - g_i)/l_i$, we define the set of triplets $X(\tilde{\mathsf{G}}) = (l_i, c_i, \overrightarrow{v_i})_{i \in [T-1]}$ and its related oriented varifold $\mu_{X(\tilde{\mathsf{G}})} = \sum_{i=1}^{T-1} l_i \delta_{c_i, \overrightarrow{v_i}}$ as in (11). This is a valid discretization of the oriented varifold μ_{G} according to [28, Proposition 1]: $\mu_{X(\tilde{\mathsf{G}})}$ converges towards μ_{G} as the size of the descretization mesh $\sup_{i \in [T-1]} |t_{i+1} - t_i|$ converges to 0.

Finally, we define a distance on discretized time series' graphs $\tilde{\mathsf{G}}_1, \tilde{\mathsf{G}}_2$ as $d_{\mathsf{W}^*}(\tilde{\mathsf{G}}_1, \tilde{\mathsf{G}}_2) = d_{\mathsf{W}^*,3}(X(\tilde{\mathsf{G}}_1), X(\tilde{\mathsf{G}}_2))$.

C.1 Varifold kernels

Denote the one-dimensional Gaussian kernel by $K_\sigma^{(a)}(x,y) = \exp(-|x-y|^2/\sigma)$ for any $(x,y) \in (\mathbb{R}^a)^2$, $a \in \mathbb{N}$ and $\sigma > 0$. In the implementation, we use the following kernels, for any $((t_1,x_1),(t_2,x_2)) \in (\mathbb{R}^{d+1})^2$, $((w_1,v_1),(w_2,v_2)) \in (\mathbb{S}^d)^2$,

$$k_{\text{pos}}(x,y) = K^{(1)}_{\sigma_{\text{pos},t}}(t_1,t_2) K^{(d)}_{\sigma_{\text{pos},x}}(x_1,x_2), \quad k_{\text{pos}}(x,y) = K^{(1)}_{\sigma_{\text{dir},t}}(w_1,w_2) K^{(d)}_{\sigma_{\text{dir},x}}(v_1,v_2) \ ,$$

where $\sigma_{\text{pos},t}, \sigma_{\text{pos},x}, \sigma_{\text{dir},t}, \sigma_{\text{dir},x} > 0$ are hyperparameters. In practice, we select $\sigma_{\text{pos},x} \approx \sigma_{\text{dir},x} \approx 1$ when the times series are centered and normalized. Otherwise we select $\sigma_{\text{pos},x} \approx \sigma_{\text{dir},x} \approx \bar{\sigma}_s$ with $\bar{\sigma}_s$ the average standard deviation of the time series. We choose $\sigma_{\text{pos},t} \approx \sigma_{\text{dir},t} = m f_e$ with f_e the sampling frequency of the time series and $m \in [5]$ an integer depending on the time change between the starting and the target time series graph. The more significant the time change, the higher m should be. The intuition comes from the fact that the width $\sigma_{\text{pos},t}, \sigma_{\text{dir},t}$ rules the time windows used to perform the comparison, and $\sigma_{\text{pos},x}, \sigma_{\text{dir},x}$ affects the space window. The size of the windows should be selected depending on the variations in the data.

D Tuning the hyperparameters of the TS-LDDMM velocity field kernel

The parameter $\sigma_{T,0}$ should be chosen *large* compared the sampling frequency f_e and compared to average standard deviation $\bar{\sigma}_s$ of the time series, e.g $\sigma_{T,0} = 100$ as $\bar{\sigma}_s \approx f_e \approx 1$. It makes the time transformation smoother. If $\sigma_{T,0}$ is too small, for instance, $\sigma_{T,0} = f_e$, the effect of the time deformation is too localized, and there are not enough samples to make it visible.

The parameter $\sigma_{T,1}$ should be of the same order as f_e: two different points in time can have various space transformations. σ_x should be of the same order of $\bar{\sigma}_s$: two points with a big difference regarding space compared to $\bar{\sigma}_s$ can have very different space transformations.

We take $c_0 \approx 10c_1$, we want to encourage time transformation before space transformation. We take $(c_0, c_1) = (1, 0.1)$ in all experiments.

E Experimental settings

All experiments were performed on a Debian 6.1.69-1 server with NVIDIA RTX A2000 12GB GPU, Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz, and 250 GB of RAM. The source code will be available on Github.

E.1 Optimization details of TS-LDDMM & LDDMM

We implemented TS-LDDMM in Python with the JAX library [3].

Initialization. As initialization of (8), all momentum parameters are set to 0, and the initial graph of reference is picked from the dataset such that its length is equal to the median length observed in the dataset.

Gradient descent. The chosen gradient descent method is "adabelief" [60] implemented in the OPTAX library [4]. The gradient descent has two main parameters: the number of steps (nb_steps) and the maximum stepsize value (η_M). The stepsize has a scheduling scheme:

- Warmup period on $0.1\times$ nb_steps steps: the stepsize increases linearly from 0 to η_M. The goal is to learn progressively the parameters. If the step size is too large at the start, smaller steps at the end cannot make up for the mistakes made at the beginning.
- Fine tuning periode on $0.9\times$ nb_steps : the stepsize decreases from η_M to 0 with a cosine decay implemented in the OPTAX scheduler, i.e. the decreasing factor as the form $0.5(1 + \cos(\pi t/T))$.

By default, we set nb_steps to 400 and η_M to 0.1.

[3] https://github.com/google/jax

[4] https://optax.readthedocs.io/en/latest/

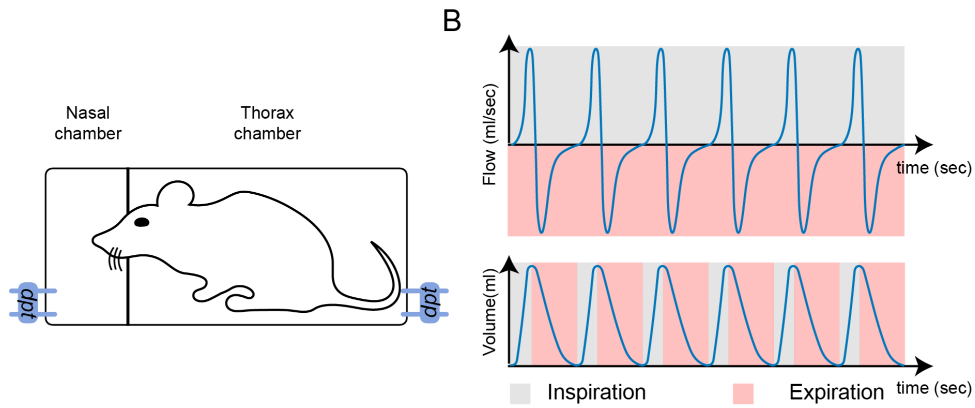


Figure 6: A: Illustration of a double-chamber plethysmograph. The term *dpt* stands for differential pressure transducer which measures the pressure in each compartment, the pressure then being converted to flow. B: Nasal airflow (top) and lung volume (bottom). During inspiration, airflow is positive (grey) and during expiration, airflow is negative (pink).

F Datasets

F.1 Mouse respiratory cycle dataset

Ventilation is a simple physiological function that ensures a vital supply of oxygen and the elimination of CO2. Acetylcholine (Ach) is a neurotransmitter that plays an important role in muscular activity, notably for breathing. Indeed, muscle contraction information passes from the brain to the muscle through the nervous system. Achs are located in synapses of the nervous system (central and peripheral) and skeletal muscles. They ensure the information transmission from nerve to nerve. However, the transmission cannot end without the hydrolysis of Ach by the enzyme Acetylcholinesterase (AchE), allowing nerves to return to their resting state. Inhibition of (AchE) with, for instance, nerve gas, pesticide, or drug intoxication leads to respiratory arrests.

The dataset comes from the experiment [39], where they studied the consequences of partial deficits in AChE and AChE inhibition on mice respiration. AchE inhibition was induced with an irritant molecule called physostigmine (an AchE inhibitor). Mice nasal airflows were sampled at 2000Hz with a Double Chamber plethysmograph [26], as depicted in Figure 6-A). The flow is expressed in $ml.s^{-1}$; it has a positive value during inspiration and a negative value expiration Figure 6-B). Among the mice population, we selected 7 control mice (**wt**) and 7 ColQ mice (**colq**), which do not have AChE anchoring in muscles and some tissues. As described in [39], mice experiments were as follows:

1. The mouse is placed in a DCP for 15 or 20 min to serve as an internal control.
2. The mouse is removed from the DCP and injected with physostigmine.
3. The mouse is placed back into the DCP, and its nasal flow is recorded for 35 or 40 min.

Respiratory cycles were extracted following procedure [19]. We removed respiratory cycles whose duration exceeds 1 second; the average respiratory cycle duration is 300 ms. We randomly sampled 10 respiratory cycles per minute and mouse. It leads to a dataset of 12,732 (time, genotype)-annotated respiratory cycles.

F.2 Shape-based UCR/UEA time series classification datasets

We selected 15 shape-based datasets (7 univariates and 8 multivariates) from the from the University of East Anglia (UEA) and the University of California Riverside (UCR) Time Series Classification

Repository[5] [13, 3]. All datasets were downloaded with the python package aeon[6]. Essential datasets information are summarized in Table 1 and further can be found in [13, 3].

Table 1: UCR/UEA shape-based time series datasets for classification.

	Dataset	Size	Lengh	Number of classes	Number of dimensions	Type
Univariate	ArrowHead	211	251	3	1	IMAGE
	BME	180	128	3	1	SIMULATED
	ECG200	200	96	2	1	ECG
	FacesUCR	2250	131	14	1	IMAGE
	GunPoint	200	150	2	1	MOTION
	PhalangesOutlinesCorrect	2658	80	2	1	IMAGE
	Trace	200	275	4	1	SENSOR
Multivariate	ArticularyWordRecognition	575	144	25	9	SENSOR
	Cricket	180	1197	12	6	MOTION
	ERing	60	65	6	4	SENSOR
	Handwriting	1000	152	26	3	MOTION
	Libras	360	45	15	2	VIDEO
	NATOPS	360	51	6	24	MOTION
	RacketSports	303	30	4	6	SENSOR
	UWaveGestureLibrary	240	315	8	3	SENSOR

G Appendix for experiment: Analysis of respiratory behavior in mice

G.1 Settings

This experiment involves TS-LDDMM and LDDMM [21] methods. Both methods are run twice, first on respiratory cycles before exposure to the irritant molecule to capture mice breathing behavior at rest and on all respiratory cycles to capture the influence of the irritant molecule. Exposure to the irritant molecule leads to significant shape deformation in the respiratory cycles, and the terms must be added to the varifold loss to capture deformations at a large time scale.

TS-LDDMMM parameters.

- **Before exposure:** The velocity field kernel K_G is set to $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 150, 1, 2)$. The varifold loss is the sum of three varifolds to capture shapes variations at different scales with parameters: (Varifold 1,Varifold 2,Varifold 3): $((5, 2, 5, 1), (2, 1, 2, 0.6), (1, 0.6, 1, 0.6))$ and the mapper $(\sigma_{\text{pos},t}, \sigma_{\text{pos},t}, \sigma_{\text{dir},t}, \sigma_{\text{dir},x})$. The optimizer has 800 steps with a maximum stepsize η_M of 0.3.
- **Before/after exposure:** The velocity field kernel K_G is set to $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 220, 1, 2)$. The varifold loss is the sum of four varifolds to capture shapes variations at different scales with parameters: (Varifold 1,Varifold 2,Varifold 3, Varifold 4): $((30, 2, 30, 1), (5, 2, 5, 1), (2, 1, 2, 0.6), (1, 0.1, 1, 0.1))$ and the mapper $(\sigma_{\text{pos},t}, \sigma_{\text{pos},t}, \sigma_{\text{dir},t}, \sigma_{\text{dir},x})$. The optimizer has 800 steps with a maximum stepsize η_M of 0.3.

LDDMMM parameters. Note that varifold losses are unchanged between TS-LDDMM and LDDMM. Compared to TS-LDDMM, the convergence of LDDMM is more sensitive to the maximum stepsize η_m, which must remain small for LDDMM to guarantee the convergence.

- **Before exposure:** The velocity field kernel K_G is an anysotropic Gaussian kernel with parameters $\sigma_T = 150$ for the time dimension and $\sigma_x = 2$ for space dimensions. The varifold loss is the sum of three varifolds to capture shapes variations at different scales with parameters: (Varifold 1,Varifold 2,Varifold 3): $((5, 2, 5, 1), (2, 1, 2, 0.6), (1, 0.6, 1, 0.6))$ and the mapper $(\sigma_{\text{pos},t}, \sigma_{\text{pos},t}, \sigma_{\text{dir},t}, \sigma_{\text{dir},x})$. The optimizer has 800 steps with a maximum stepsize η_M of 0.01.
- **Before/after exposure:** The velocity field kernel K_G is an anysotropic Gaussian kernel with parameters $\sigma_T = 220$ for the time dimension and $\sigma_x = 2$

[5]https://timeseriesclassification.com
[6]https://www.aeon-toolkit.org/en/stable/

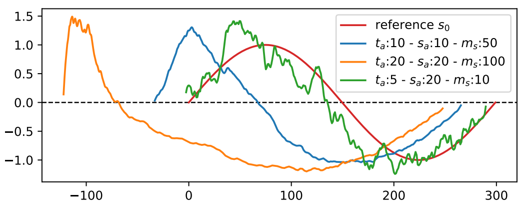


Figure 7: Plots of $\varphi^{\{v_0(\alpha^*,\mathsf{X})\}}.\mathsf{X}$ for different values of α^* according to its sampling parameter t_a, s_a, m_s, taking $\mathsf{X} = \mathsf{G}(s_0)$ with $s_0 : k \in [300] \to \sin(2\pi k/300)$.

Table 2: Values of $\mathscr{L}(\varphi^{\{v_0(\alpha^*,\mathsf{X})\}}.\mathsf{X}, \varphi^{\{\hat{v}_0\}}.\mathsf{X})$ as α^* is sampled according to Gen(10,10,50) and $\hat{v}_0$ is estimated using K_{G} with varying parameters $\sigma_{T,1}, \sigma_x$.

$\sigma_{T,0} \backslash \sigma_x$	1	10	50	100	200	300
0.1	2e+0	3e-4	1e-5	4e-6	7e-4	4e-3
1	4e-2	1e-4	1e-5	4e-6	7e-4	4e-3
100	4e-2	2e-4	1e-5	4e-6	7e-4	4e-3

for space dimensions. The varifold loss is the sum of four varifolds to capture shapes variations at different scales with parameters: (Varifold 1,Varifold 2,Varifold 3, Varifold 4): $((30, 2, 30, 1), (5, 2, 5, 1), (2, 1, 2, 0.6), (1, 0.1, 1, 0.1))$ and the mapper $(\sigma_{\text{pos},t}, \sigma_{\text{pos},t}, \sigma_{\text{dir},t}, \sigma_{\text{dir},x})$. The optimizer has 800 steps with a maximum stepsize η_M of 0.01.

H Appendix for experiment: TS-LDDMM representation identifiability

H.1 Settings

This experiment only involves the TS-LDDMM method in two different settings:

- **The velocity field kernel K_G is well-specified:** The velocity field kernel K_G is set to $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 100, 1, 1)$, the varifold loss kernels (k_{pos}, k_{dir}) are set to $(\sigma_{\text{pos},t}, \sigma_{\text{pos},t}, \sigma_{\text{dir},t}, \sigma_{\text{dir},x}) = (2, 1, 2, 0.6)$, and the optimizer has 400 steps with a maximum stepsize η_M of 0.05.
- **The velocity field kernel K_G is missspecified:** The velocity field kernel K_G is set with $(c_0, c_1, \sigma_{T,1}) = (1, 0.1, 1)$, $\sigma_{T,0}$ ranging in $(1, 5, 10, 50, 100, 200, 300)$, and σ_x ranging in $(0.1, 1, 10, 100)$. The varifold loss kernels (k_{pos}, k_{dir}) are set to $(\sigma_{\text{pos},t}, \sigma_{\text{pos},t}, \sigma_{\text{dir},t}, \sigma_{\text{dir},x}) = (2, 1, 2, 0.6)$, and the optimizer has 400 steps with a maximum stepsize η_M of 0.05.

provided that the hyperparameters and the reference graph are wisely selected, i.e., the parameter v_0^* generating a deformation $\varphi^{\{v_0^*\}}$ of a time series graph G can be estimated from the data $\mathsf{G}, \varphi^{\{v_0^*\}}.\mathsf{G}$ by solving the geodesic shooting problem (6).

First, we show the model identifiability when the kernel K_G is well specified: the estimated parameter is a good approximation of the generating parameter when the generation and the estimation procedure use the same hyperparameters for the RKHS kernel K_{G}. All the hyperparameter values for generation and estimation are given in **??**. We fix the initial control points as $\mathsf{X} = (x_k = (k, \sin(2\pi k/300)))_{k\in[300]}$. Given $m_s \in \mathbb{N}_{>0}$ and $t_a, s_a > 0$, we randomly generate initial momentums $\alpha^* = (\alpha_k^*)_{k\in[\mathbf{n}_0]}$ with the following sampling, called Gen(m_s, t_a, s_a): For any $k \in [\mathbf{n}_0]$, α'_k is sampled according to a Gaussian normal distribution $\mathcal{N}(0_{d+1}, I_{d+1})$. Then, $(\alpha'_k)_{k\in[\mathbf{n}_0]}$ is regularized by a rolling average of size m_s, we get $\bar{\alpha}' = (\bar{\alpha}'_k)_{k\in[\mathbf{n}_0]}$. Finally, we normalize $\bar{\alpha}'$ to derive α^* such that $|([\alpha_k^*]_t)_{k\in[\mathbf{n}_0]}| = t_{\text{amp}}$ and $|([\alpha_k^*]_s)_{k\in[\mathbf{n}_0]}| = s_{\text{amp}}$ for any $k \in [\mathbf{n}_0]$, denoting by $[\alpha_k^*]_t, [\alpha_k^*]_s$ the time and space coordinates of α_k^* respectively. Note that the regularizing step $(\alpha'_k)_{k\in[\mathbf{n}_0]} \to \bar{\alpha}'$ is necessary to obtain realistic deformations which take into account the regularity induced by the RKHS V.

Then, using $v_0(\alpha^*, \mathsf{X})$ as defined in (4) with initial momentums α^* and control points X, we apply the induced deformation $\varphi^{\{v_0\}}$ by (5) to X and obtain $\varphi^{\{v_0\}}.\mathsf{X}$. Finally, we solve (6) to recover an estimation $\hat{\alpha}$ of α^* and report the average relative error (ARE) $|v_0(\hat{\alpha}, \mathsf{X}) - v_0(\alpha^*, \mathsf{X})|_{\mathsf{V}} / |v_0(\alpha^*, \mathsf{X})|_{\mathsf{V}}$ on 50 repetitions. This procedure is performed for any $m_s, t_a, s_a \in \{10, 50, 100\} \times \{5, 10, 15, 20\}^2$. Mean, standard deviation, and maximum of the ARE on all these hyperparameters choices are respectively $\mathbf{0.10}, \mathbf{0.03}, \mathbf{0.17}$. Therefore, the estimation procedure (6) offers a good approximation of the true parameter when the kernel K_{G} is well specified. We observe that the estimation is difficult when $t_a \ll s_a$ because the time series can be very noisy as illustrated in Figure 7: this impacts the Varifold loss which is sensitive to tangents.

Secondly, we demonstrate a weak identifiability when the kernel K_{G} is misspecified: we can reconstruct the graph time series' after deformations even if the hyperparameters of K_{G} are different during the generation and the estimation. The hyperparameters of K_{G} during generation are $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 100, 1, 1)$ and we fix $\sigma_{T,1}, c_0, c_1 = (1, 1, 0.1)$ for K_{G} during estimation. We aim to understand the impact of $\sigma_{T,1}, \sigma_x$ on the reconstruction since they are encoding the smoothness of the transformation according to time and space.

For any choice of the hyperparameters $\sigma_{T,1}, \sigma_x \in \{1, 10, 50, 100, 200, 300\} \times \{0.1, 1, 100\}$ related to K_{G} in the estimation, we average $\mathscr{L}(\varphi^{\{v_0(\alpha^*, \mathsf{X})\}}.\mathsf{X}, \varphi^{\{\hat{v}_0\}}.\mathsf{X})$ on 50 repetitions when α^* is sampled according to Gen$(10, 10, 50)$ and $\hat{v}_0 = v_0(\hat{\alpha}, \mathsf{X})$ denoting by $\hat{\alpha}$ the result of the minimization (6). We observe in Table 2 that the reconstruction is almost perfect except in the case when $\sigma_{t,0} = 1$ during estimation, while $\sigma_{t,0} = 100$ during generation. Compared to $\sigma_{T,0}$, σ_x has nearly no impact on the reconstruction. In Appendix C.1-D, we propose guidelines to drive future hyperparameters tuning and further discussions related to $\sigma_{T,1}, c_0, c_1$.

I Appendix for experiment: Robustness to irregular sampling

This experiment is inspired by [41] where the authors perform an extensive comparison of Neural Ordinary Differential Equations (Neural ODEs) methods [29]. We assess the classification performances of several methods under regular sampling (0% missing rate) and three irregular sampling regimes on 15 shape-based datasets (7 univariate & 8 multivariate). Methods and training strategy are taken from its associated Github[7] and described in what follows. We conclude with the results, which show that our method, TS-LDDMM, outperforms all methods for sampling regimes with missing rates: 0%, 30%, and 50%.

I.1 Benchmark methods

In related work, we give an overview of Neurals ODEs methods and their relation with TS-LDDMM.

- RNN-based methods: Baseline reccurent neural networks including RNN [36], LSTM [25], and GRU [12].
- Attention-based methods: Multi-Time Attention Networks (MTAN) [49] and Multi-Integration Attention Module (MIAM) [33]. Both handle multivariate time series irregularly sampled with attention mechanisms.
- Neural ODEs: ODE-LSTM [32] a form of Neural-ODEs used to learn continuous latent representations.
- Neural SDEs: Neural SDE [34] and Neural LNSDE [41] have been proposed to model randomness in time-series using drift and diffusion terms as an extension of Neural-ODEs.
- Shape-Analysis methods: TS-LDDMM (ours) and LDDMM [21]. From shape analysis, both methods learn representations by solving ODEs parametrized with Kernels. While both methods handle multivariate signals irregularly sampled, TS-LDDMM is specifically designed for time series.

I.2 Model architecture

Neural ODEs methods As depicted in [41], any Neural ODEs layer in Appendix I.1 is followed by an MLP with two fully connected layers with `ReLU` activations. The risk of overfitting and the model regularization are handled with a dropout rate of 10% and an early-stopping mechanism, ceasing the training when the validation loss does not improve for 10 successive epochs.

[7] `https://github.com/yongkyung-oh/Stable-Neural-SDEs`

TS-LDDMM and LDDMM

I.3 Protocol

I.4 Results

Methods	Test F1-score			
	Regular	30 % dropped	50 % dropped	70 % dropped
RNN (1999)	0.64 ± 0.21	0.53 ± 0.23	0.48 ± 0.21	0.44 ± 0.21
LSTM (1997)	0.61 ± 0.29	0.57 ± 0.29	0.53 ± 0.25	0.51 ± 0.29
GRU (2014)	0.71 ± 0.26	0.68 ± 0.28	0.66 ± 0.28	$\underline{0.59 \pm 0.28}$
MTAN (2021)	0.59 ± 0.28	0.58 ± 0.28	0.54 ± 0.29	0.51 ± 0.28
MIAM (2022)	0.48 ± 0.35	0.42 ± 0.33	0.47 ± 0.31	0.35 ± 0.31
ODE-LSTM (2020)	0.63 ± 0.24	0.57 ± 0.25	0.51 ± 0.24	0.45 ± 0.23
Neural SDE (2019)	0.48 ± 0.28	0.47 ± 0.26	0.45 ± 0.27	0.45 ± 0.25
Neural LNSDE (2024)	0.7 ± 0.27	0.68 ± 0.29	$\underline{0.67 \pm 0.25}$	$\mathbf{0.66 \pm 0.23}$
LDDMM (2008)	$\underline{0.72 \pm 0.2}$	$\underline{0.7 \pm 0.21}$	0.57 ± 0.25	0.4 ± 0.25
TS-LDDMM (ours)	$\mathbf{0.83 \pm 0.18}$	$\mathbf{0.8 \pm 0.18}$	$\mathbf{0.7 \pm 0.26}$	0.51 ± 0.27

I.5 Appendix for experiment: Classification benchmark on regularly sampled datasets

I.6 Settings

In this section, we compare classification performances of TS-LDDMM with other state-of-the-art methods coming from shape analysis on 15 shape-based datasets of time-series.

Methods We compare TS-LDDMM with a method from function [56]

	Dataset	Shape-FPCA (2024)	TCLR (2024)	LDDMM (2008)	TS-LDDMM (ours)
Univariate	ArrowHead	0.18	0.75	$\underline{0.84}$	**0.91**
	BME	0.16	$\underline{1.00}$	0.82	**1.00**
	ECG200	0.40	0.67	**0.81**	$\underline{0.79}$
	FacesUCR	0.08	$\underline{0.73}$	0.69	**0.86**
	GunPoint	0.93	$\underline{0.97}$	0.83	**1.00**
	PhalangesOutlinesCorrect	0.39	**0.63**	$\underline{0.53}$	0.52
	Trace	0.55	$\underline{1.00}$	0.46	**1.00**
Multivariate	ArticularyWordRecognition	–	–	$\underline{0.98}$	**1.00**
	Cricket	–	–	$\underline{0.77}$	**0.93**
	ERing	–	–	$\underline{0.95}$	**0.98**
	Handwriting	–	–	$\underline{0.22}$	**0.44**
	Libras	–	–	$\underline{0.56}$	**0.60**
	NATOPS	–	–	$\underline{0.82}$	**0.82**
	RacketSports	–	–	**0.83**	$\underline{0.79}$
	UWaveGestureLibrary	–	–	$\underline{0.72}$	**0.81**

Protocole

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Each claim in the introduction is referring to the part where it is tackled.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.

- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have provided a special section for this purpose Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All the proofs are given in Appendix B and each proof state all its arguments in a logical order.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The optimization methodology is described in Section 4 and all numerical details and protocol are given in Section 5 and in **??**-E.1-**??**-**??**.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Synthetic data can be generated, benchmark data are publicly available (Appendix F.2), but the mouse dataset is not. The code is provided as supplementary material and will be made publicly available if the paper is accepted.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experiments setting is clearly presented in Section 5 for the experiments on the mouse dataset. The full details of all experiments are presented in **??**-E.1-**??**-**??**.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are given for the synthetic experiments, but not for the real experiments due to limited computational budget. There is no randomness in the train/test split but there is randomness in the optimization method (Adabelief), but its impact is not significant on the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This is given at the beginning of Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Every authors are funded for their work and preserve research integrity.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: A special section is given on this subject Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All dataset and python library were cited when necessary.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The mouse dataset is described in Appendix F.1.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.