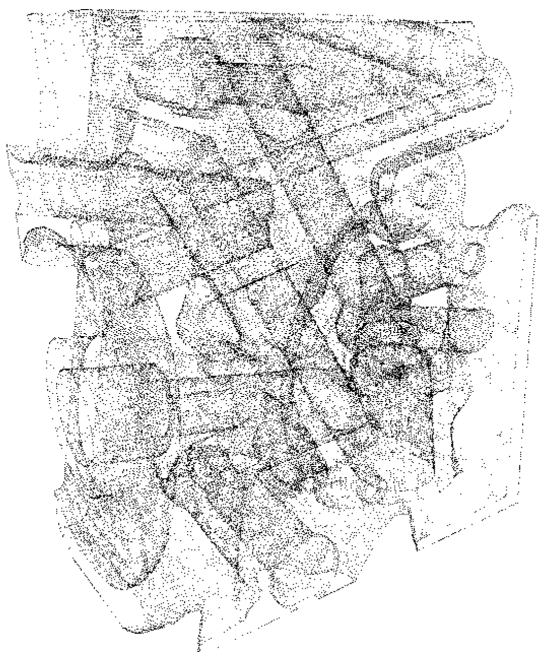
Applied Machine Learning Days at EPFL 2020
AI & Topology track - January 28, 2020

Learning linear representations of persistence diagrams: mathematical aspects and applications

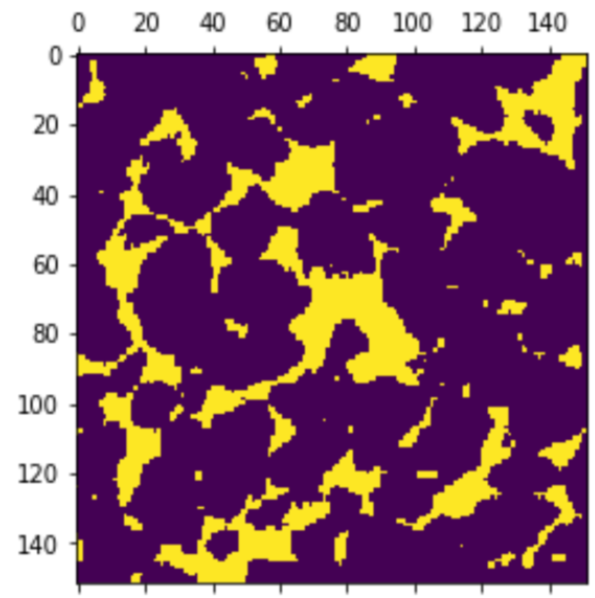
Frédéric Chazal
DataShape - INRIA Saclay - Ile-de-France
frederic.chazal@inria.fr

informatics mathematics

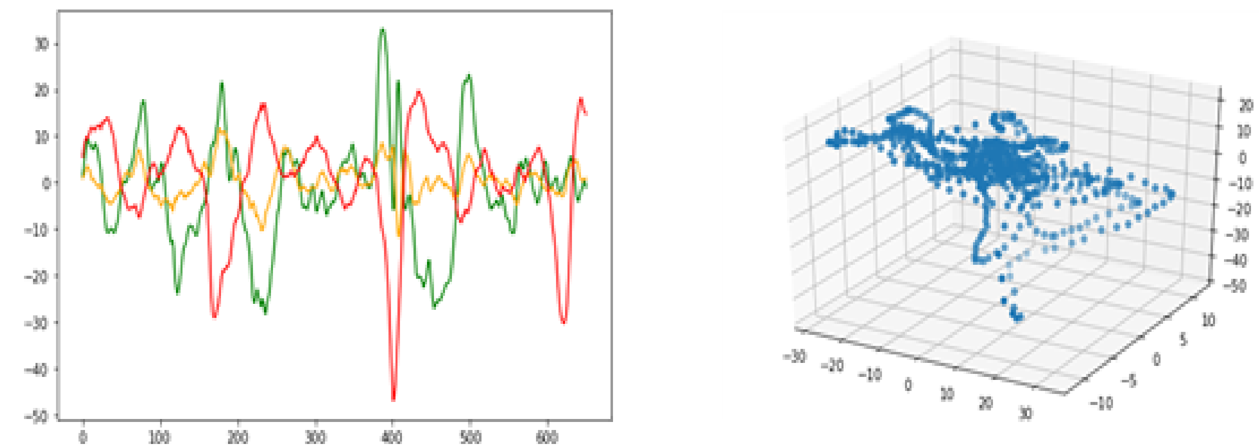
What is Topological Data Analysis (TDA)?



[Scanned 3D object]



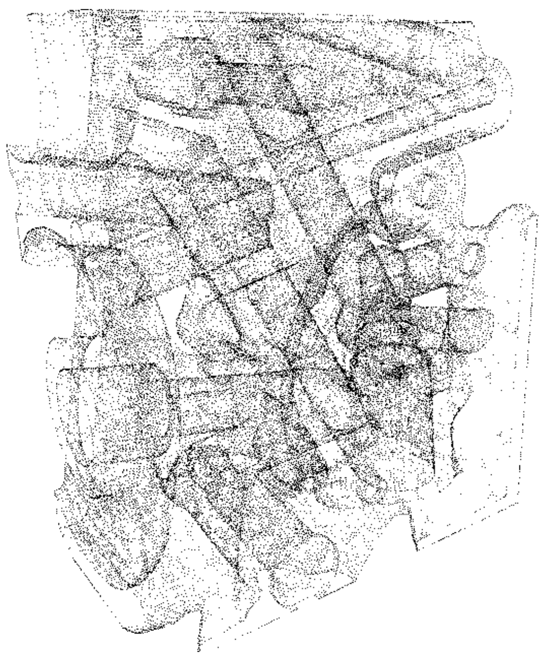
[3D images (porous rocks)]



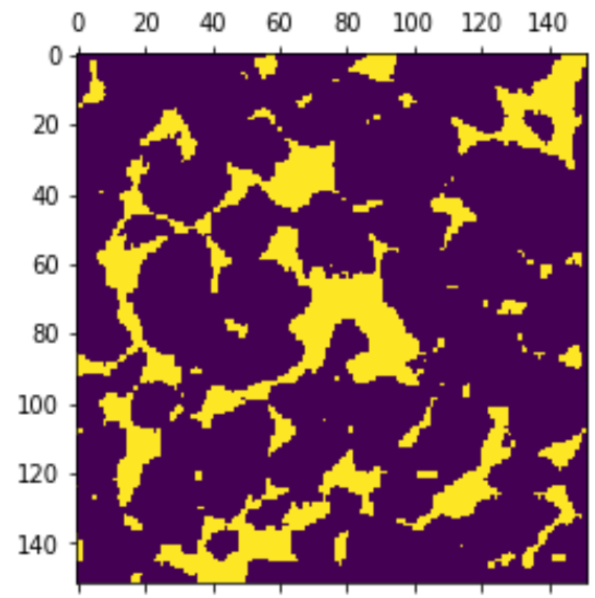
[Sensors (Sysnav courtesy)]

Modern data carry complex, but important, geometric/topological structure!

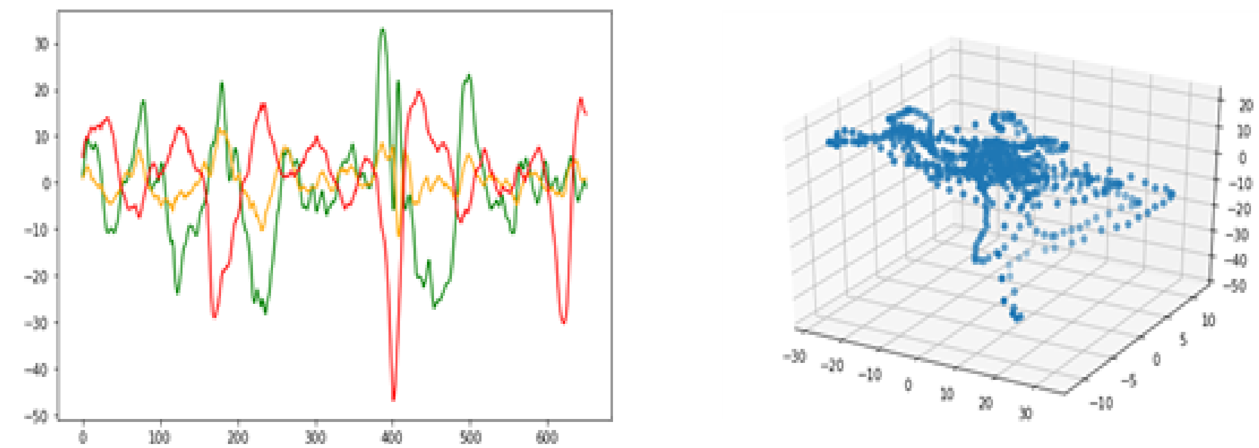
What is Topological Data Analysis (TDA)?



[Scanned 3D object]



[3D images (porous rocks)]

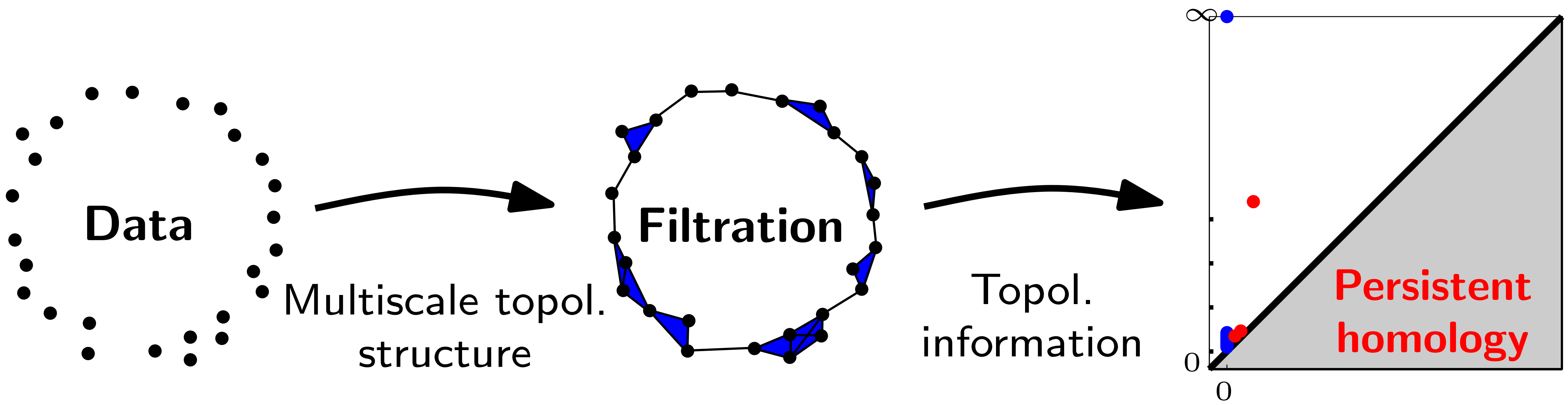


[Sensors (Sysnav courtesy)]

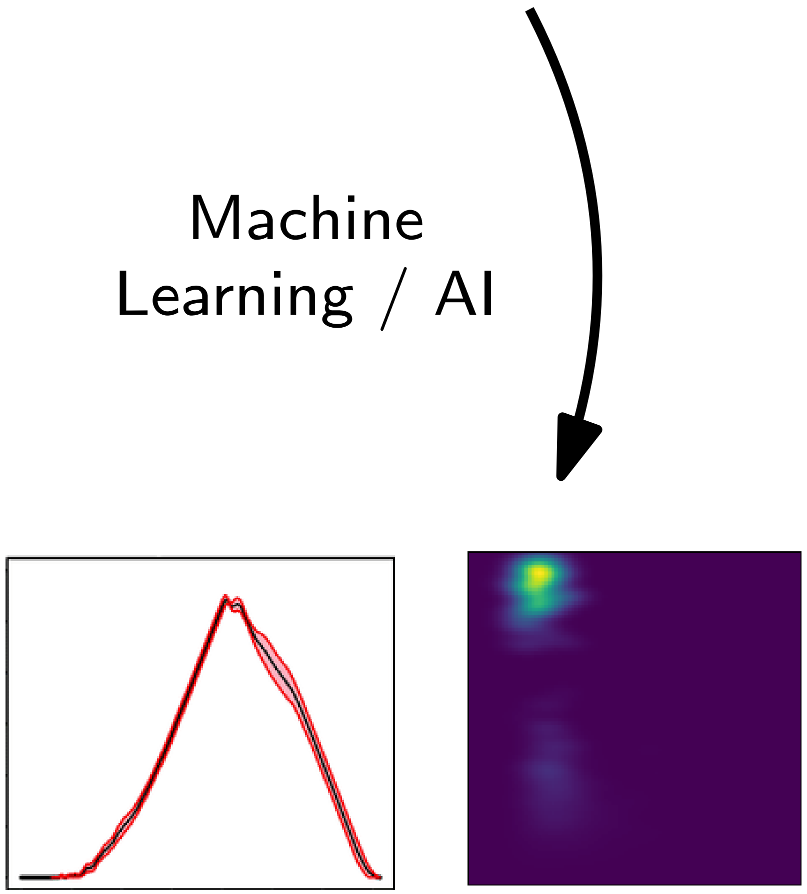
Topological Data Analysis (TDA) is a recent field whose aim is to:

- infer relevant topological and geometric features from complex data,
- take advantage of topological/geometric information for further Data Analysis, Machine Learning and AI tasks:
  - using topological features in ML pipelines,
  - taking advantage of topological information to improve ML pipelines.

A classical TDA pipeline



1. Build a multiscale topol. structure on top of data: filtrations.
2. Compute multiscale topol. signatures: persistent homology
3. Take advantage of the signature for further Machine Learning and AI tasks: Statistical aspects and representations of persistence

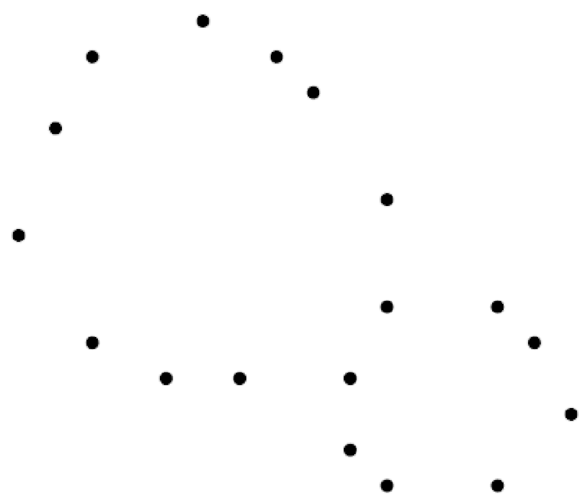


Representations of persistence

Persistent homology

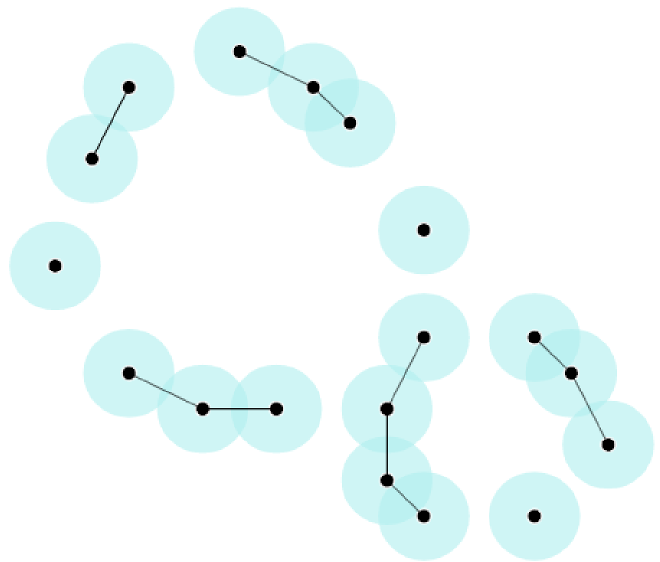
- $90's$: size theory (P. Frosini et al), framed Morse complex and stability (S.A. Barannikov).
- $2002 - 2005$: persistent homology (H. Edelsbrunner et al, Carlsson et al).
- important mathematical and practical developments since the 2000's.

Persistent homology for point cloud data



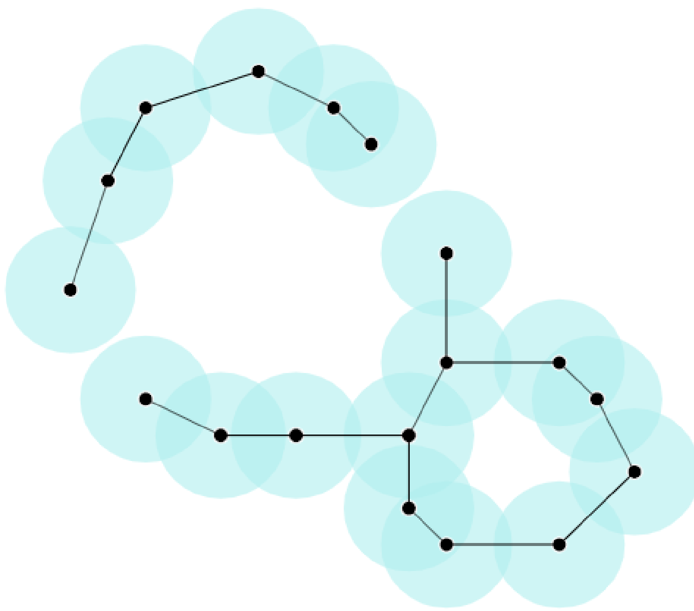
- Filtrations allow to construct "shapes" representing the data in a multiscale way.
- Persistent homology: encode the evolution of the topology across the scales $\rightarrow$ multi-scale topological signatures.

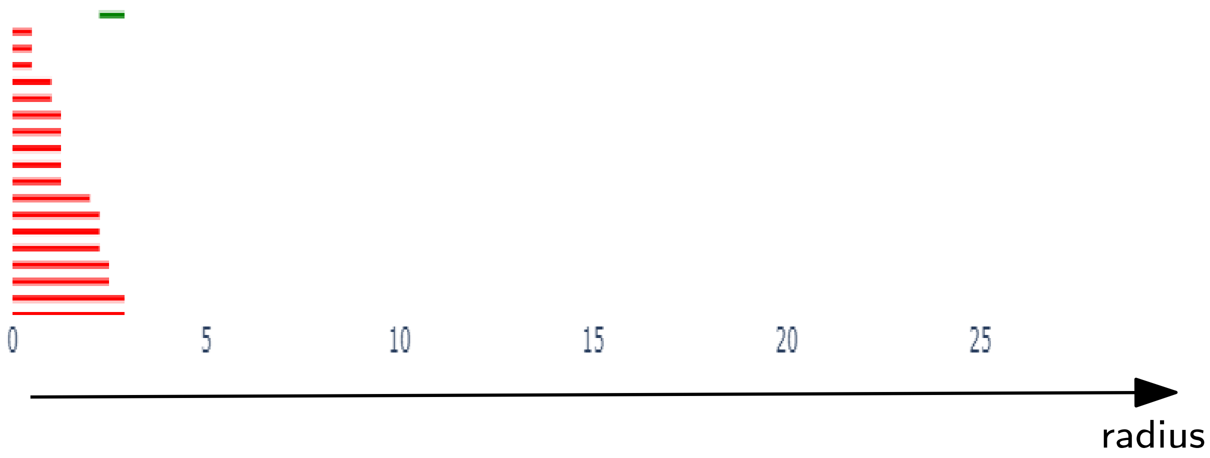
Persistent homology for point cloud data



- Filtrations allow to construct "shapes" representing the data in a multiscale way.
- Persistent homology: encode the evolution of the topology across the scales $\rightarrow$ multi-scale topological signatures.

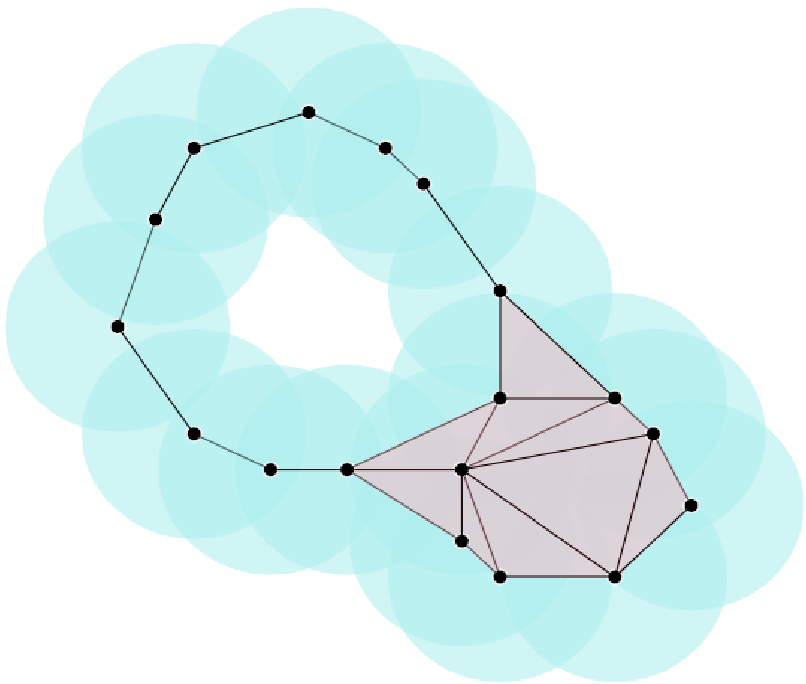
Persistent homology for point cloud data





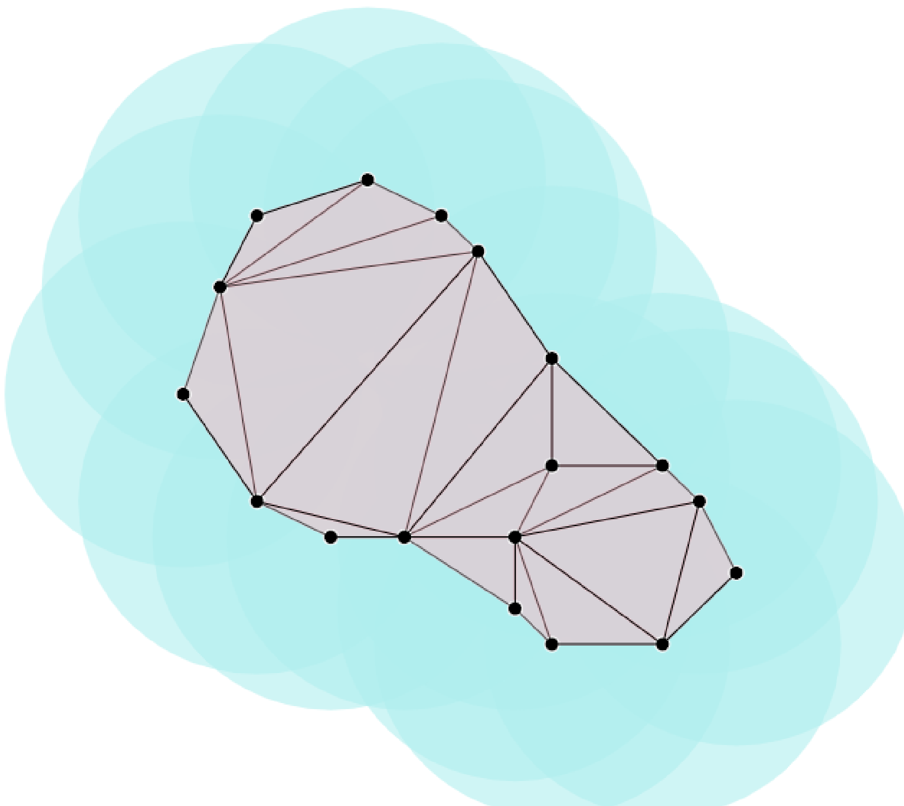
- Filtrations allow to construct "shapes" representing the data in a multiscale way.
- Persistent homology: encode the evolution of the topology across the scales $\rightarrow$ multi-scale topological signatures.

Persistent homology for point cloud data

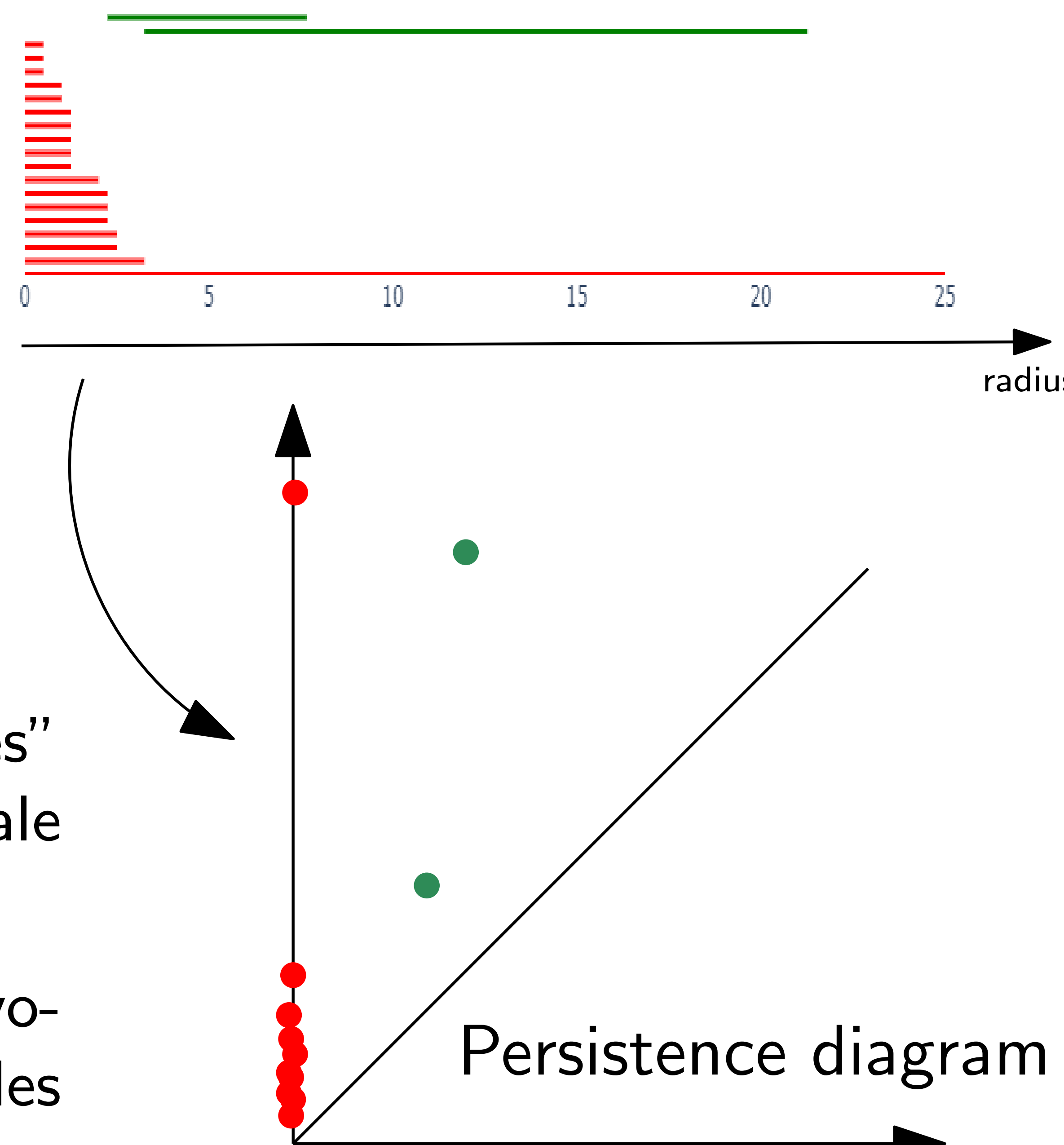


- Filtrations allow to construct “shapes” representing the data in a multiscale way.
- Persistent homology: encode the evolution of the topology across the scales $\rightarrow$ multi-scale topological signatures.

Persistent homology for point cloud data



Persistence barcode



- Filtrations allow to construct "shapes" representing the data in a multiscale way.
- Persistent homology: encode the evolution of the topology across the scales → multi-scale topological signatures.

Stability properties

"Stability theorem": Close spaces/data sets have close persistence diagrams!

[C., de Silva, Oudot - Geom. Dedicata 2013].

If $\mathbb{X}$ and $\mathbb{Y}$ are pre-compact metric spaces, then

$$\mathrm{d_b}(\mathsf{dgm}(\mathrm{Rips}(\mathbb{X})), \mathsf{dgm}(\mathrm{Rips}(\mathbb{Y}))) \leq d_{GH}(\mathbb{X}, \mathbb{Y}).$$

Bottleneck distance

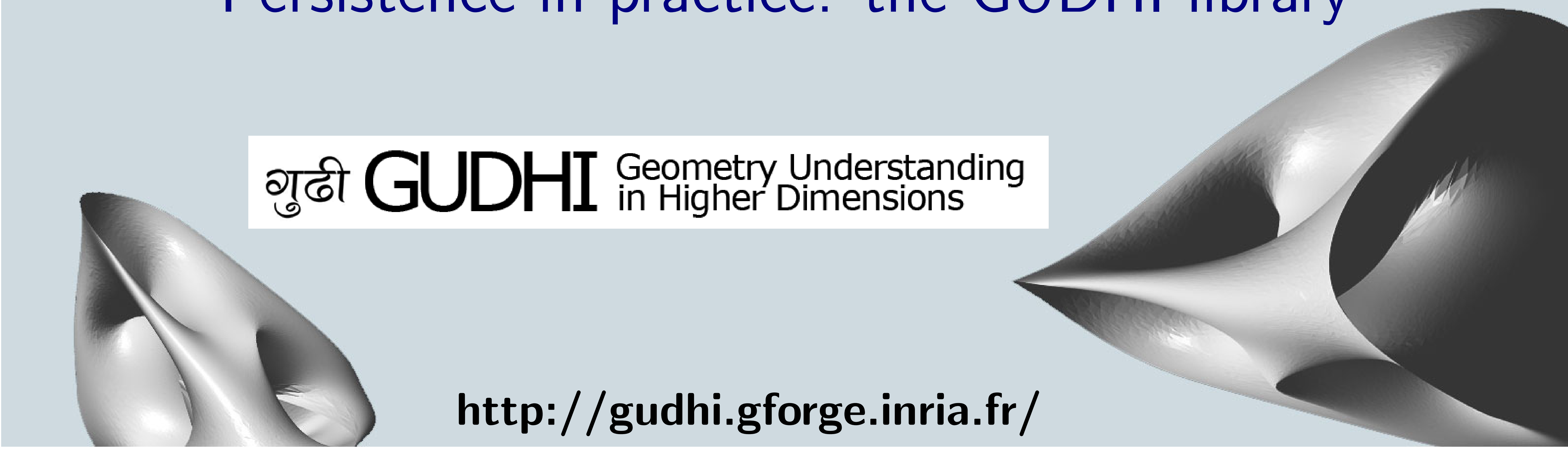
Gromov-Hausdorff distance

$$d_{GH}(\mathbb{X}, \mathbb{Y}) := \inf_{\mathbb{Z}, \gamma_1, \gamma_2} d_H(\gamma_1(\mathbb{X}), \gamma_2(\mathbb{X}))$$

$\mathbb{Z}$ metric space, $\gamma_1 : \mathbb{X} \to \mathbb{Z}$ and $\gamma_2 : \mathbb{Y} \to \mathbb{Z}$ isometric embeddings.

Rem: This result also holds for other families of filtrations (particular case of a more general thm).

Persistence in practice: the GUDHI library

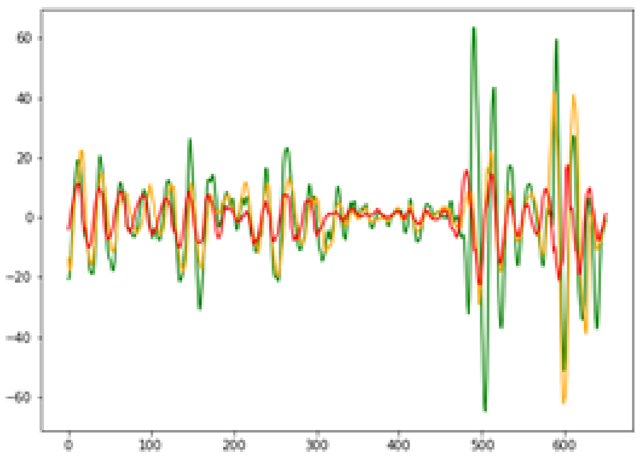


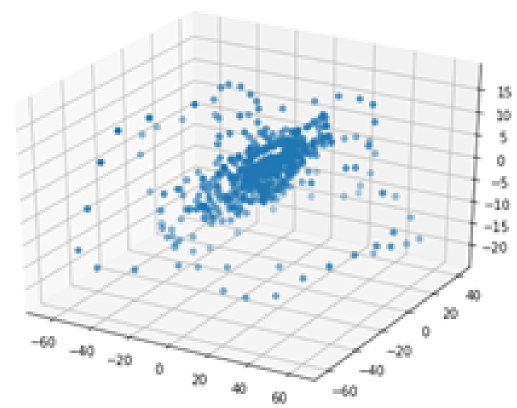
GUDHI :

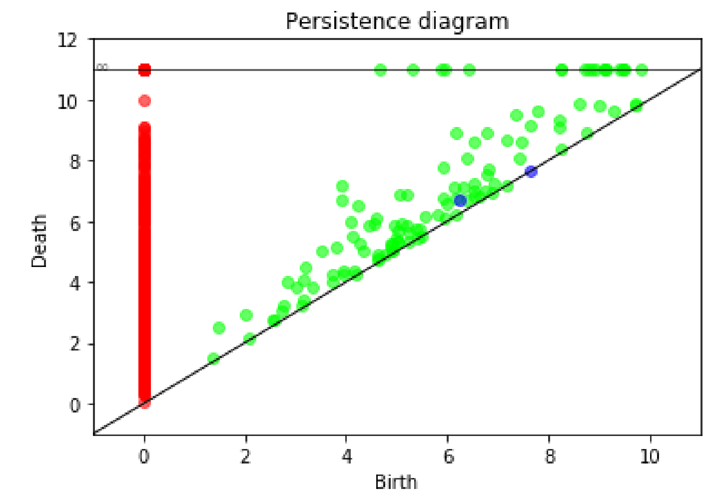
- a C++/Python open source software library for TDA (part of it interfaced to R),
- provides state-of-the-art TDA data structures and algorithms : design of filtrations, computation of pre-defined filtrations, persistence diagrams,...
- a developers team, an editorial board, open to external contributions,
- Python interface for easy use in ML pipeline (see also Giotto, developed at EPFL)

TDA and Machine Learning
Representations of persistent homology

The problem of representation of persistence





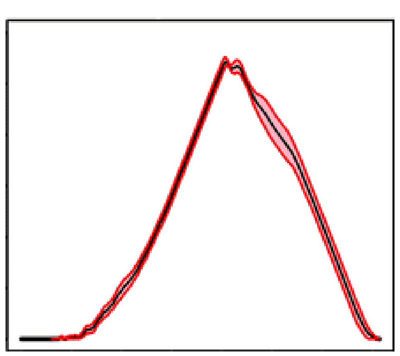


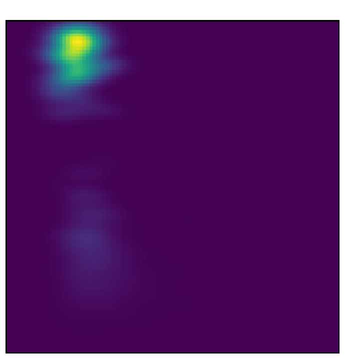
Persistence diagrams are not well-suited for classical ML algorithms (the space of PD is highly non linear)

Not always clear which part of the diagrams carries the relevant information.

An active research area, but still at a very early stage (in particular regarding math. aspects).

Machine Learning / AI





Representations of persistence

A zoo of representations of persistence

(non exhaustive list)

- Collections of 1D functions

  → landscapes [Bubenik 2012]

  → Betti curves [Umeda 2017]

- discrete measures: (interesting statistical properties)

  → persistence images [Adams et al 2017]

  → convolution with Gaussian kernel [Reininghaus et al. 2015] [Chepushtanova et al. 2015] [Kusano Fukumisu Hiraoka 2016-17] [Le Yamada 2018]

  → sliced on lines [Carrière Oudot Cuturi 2017]

  → quantization [C. Ike Royer Umeda 2019]

- finite metric spaces [Carrière Oudot Ovsjanikov 2015]

- polynomial roots or evaluations [Di Fabio Ferri 2015] [Kališnik 2016]

A zoo of representations of persistence

(non exhaustive list)

- Collections of 1D functions

  → landscapes [Bubenik 2012]

  → Betti curves [Umeda 2017]

- discrete measures: (interesting statistical properties)

  → persistence images [Adams et al 2017]

  → convolution with Gaussian kernel [Reininghaus et al. 2015] [Chepushtanova et al. 2015] [Kusano Fukumisu Hiraoka 2016-17] [Le Yamada 2018]

  → sliced on lines [Carrière Oudot Cuturi 2017]

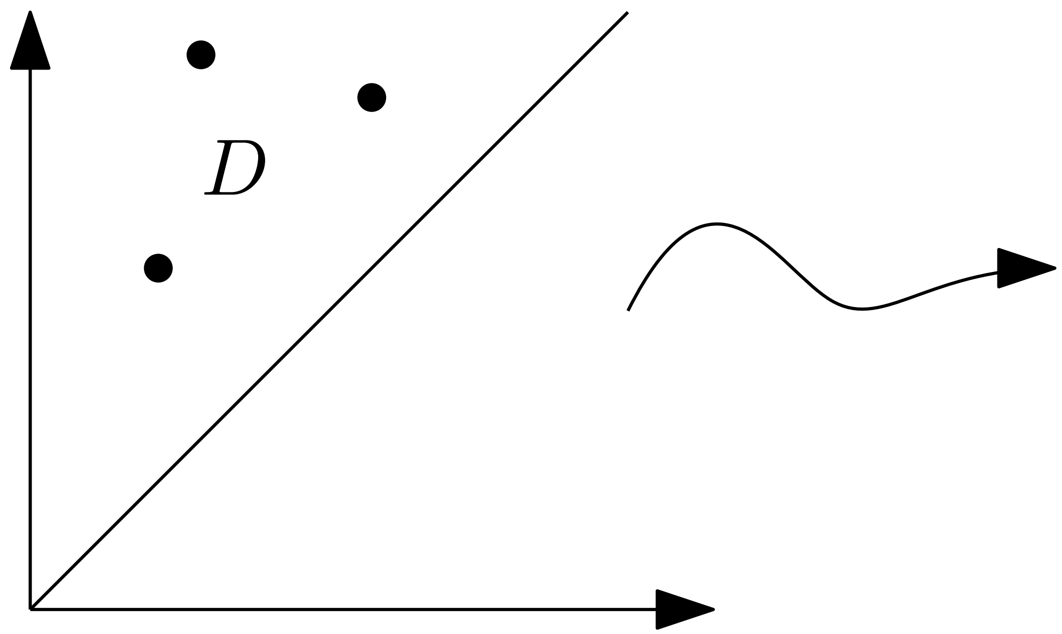
  → quantization [C. Ike Royer Umeda 2019]

- finite metric spaces [Carrière Oudot Ovsjanikov 2015]

- polynomial roots or evaluations [Di Fabio Ferri 2015] [Kališnik 2016]

Problem: How to chose the right representation?

Persistence diagrams as discrete measures



$$D := \sum_{\mathbf{r} \in D} \delta_{\mathbf{r}}$$

Motivations:

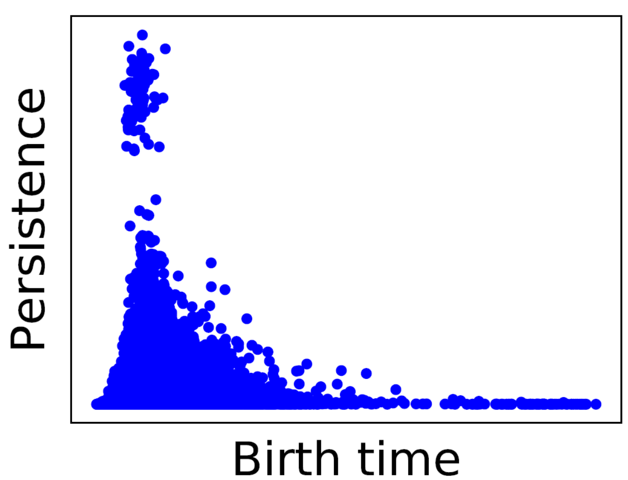
- The space of measures is much nicer that the space of P. D. !
- In the “standard” algebraic persistence theory, persistence diagrams naturally appear as discrete measures in the plane (over rectangles).
  [Chazal, de Silva, Glisse, Oudot 16]
- Many persistence representations can be expressed as

$$D(\phi) = \sum_{\mathbf{r} \in D} \phi(\mathbf{r}) = \int \phi(\mathbf{r}) dD(\mathbf{r})$$

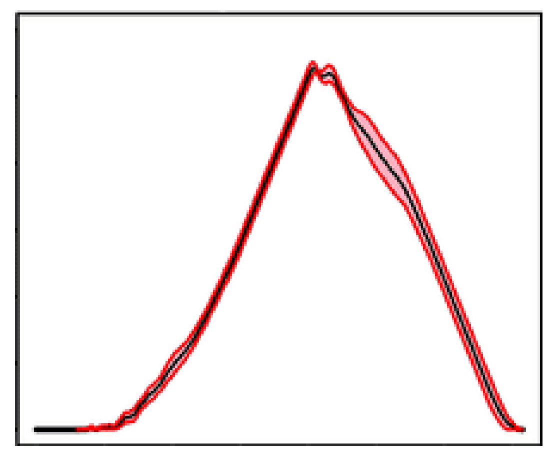
Representations of Persistence diagrams

A representation is called linear if there exists $\phi : \mathbb{R}^2_> \to \mathcal{H}$ such that

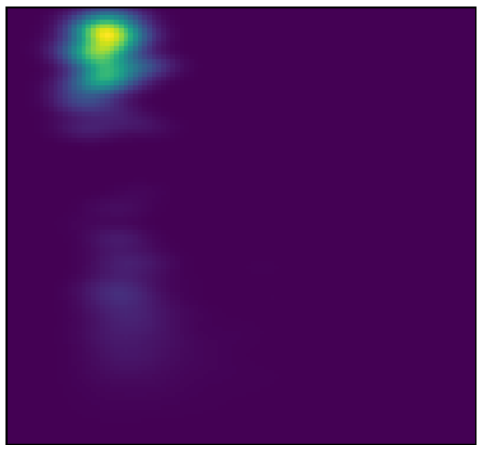
$$\Phi(D) = \sum_{\mathbf{r} \in D} \phi(r) := D(\phi) = \int \phi(\mathbf{r})\, dD(\mathbf{r})$$



Distrib. of life span, total persistence,...



Persistent silhouette
[Chazal & al, 2013]



Persistent surface
[Adams & al, 2016]

Math/Stat behavior of linear representations is well-understood (stability, expectation,...). [C-Divol, JoCG 2020, Divol-Lacombe 2019]

In ML settings, well-suited linear representations of PD can be learnt. [Hofer et al., NeurIPS 2017, Carrière et al, AISTATS 2020]

Deep Set Architecture

Originally defined in [Zaheer et al. 2017]

Tailored to handle sets instead of finite dimensional vectors

Input: $\{x_1, ..., x_n\} \subset \mathbb{R}^d$ instead of $x \in \mathbb{R}^d$

Deep Set Architecture

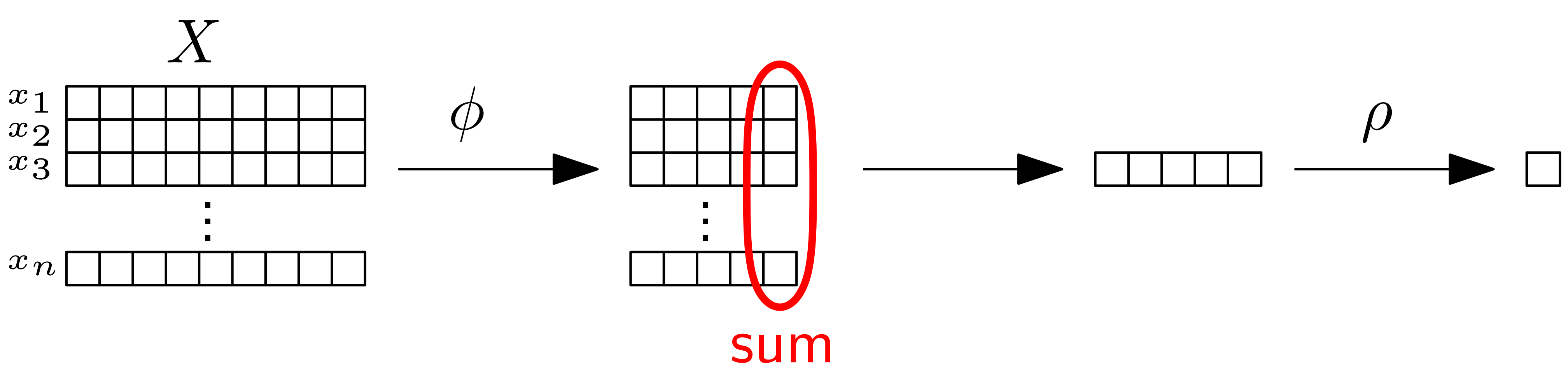
Originally defined in [Zaheer et al. 2017]

Tailored to handle sets instead of finite dimensional vectors

Input: $\{x_1, ..., x_n\} \subset \mathbb{R}^d$ instead of $x \in \mathbb{R}^d$

Network is *permutation invariant*: $F(X) = \rho\left(\sum_i \phi(x_i)\right)$

$$\Rightarrow F(\{x_1, ..., x_n\}) = F(\{x_{\sigma(1)}, ..., x_{\sigma(n)}\}),\ \forall\sigma$$



In practice: $\phi(x_i) = W \cdot x_i + b$

Deep Set Architecture

Originally defined in [Zaheer et al. 2017]

Tailored to handle sets instead of finite dimensional vectors

Input: $\{x_1, ..., x_n\} \subset \mathbb{R}^d$ instead of $x \in \mathbb{R}^d$

Network is *permutation invariant*: $F(X) = \rho\left(\sum_i \phi(x_i)\right)$

Universality theorem

Th: [Zaheer et al. 2017]

A function f is permutation invariant iif $f(X) = \rho\left(\sum_i \phi(x_i)\right)$ for some ρ and ϕ, whenever X is included in a *countable* space

PersLay: adaptation to persistence diagrams

[Carrière, C., Ike, Lacombe, Royer, Umeda, AISTATS 2020]

Permutation invariant layers generalize several TDA approaches

$\rightarrow$ persistence images $\rightarrow$ silhouettes $\rightarrow$ Betti curves

But not all of them since $\mathbb{R}^2$ is not countable

Using any permutation invariant operation (such as max, min, kth largest value) allows to generalize to other persistence representations.

$$\mathrm{PersLay}(dgm) = \rho\left(\mathsf{op}\{w(p)\cdot\phi(p)\}_{p\in dgm}\right)$$

Permutation-invariant operation

Weight function

Point transformation $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^k$

Code already available: https://github.com/GUDHI/TDA-tutorial - to be released in GUDHI

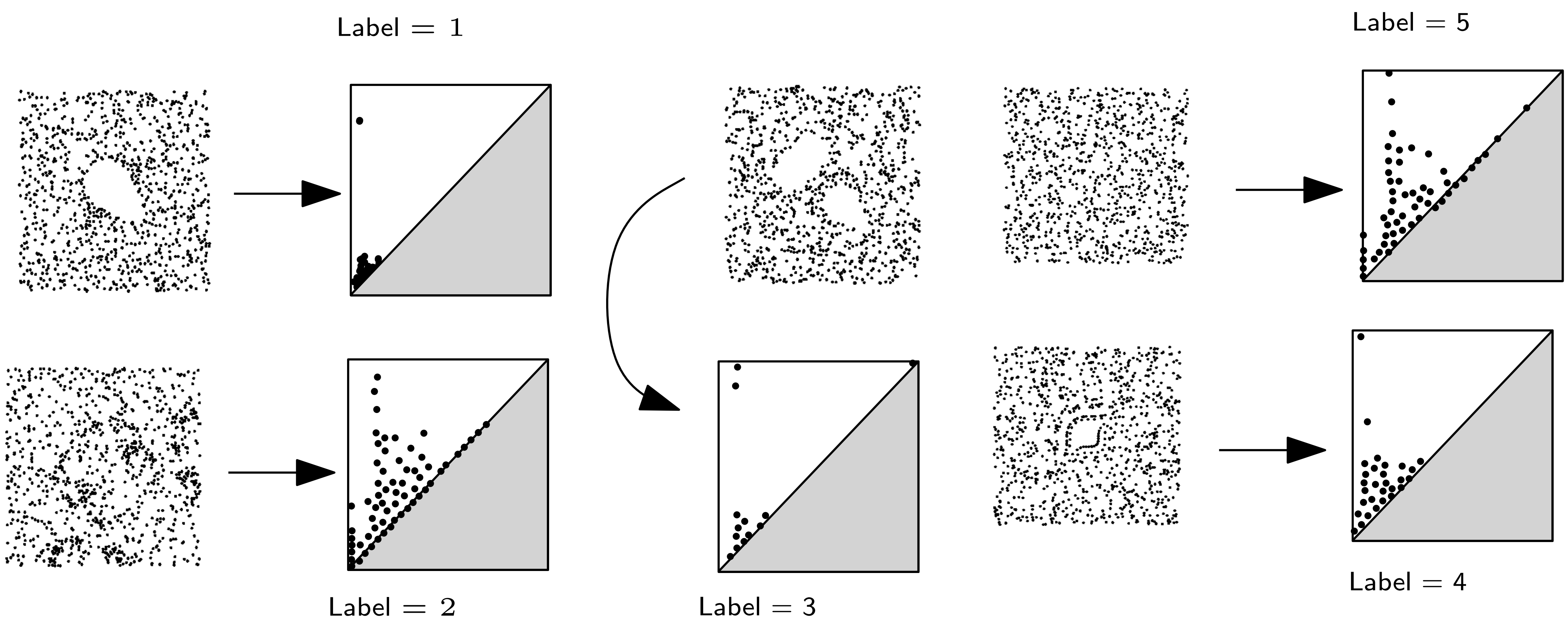
PersLay: adaptation to persistence diagrams

[Carrière, C., Ike, Lacombe, Royer, Umeda, AISTATS 2020]

Example: classify orbits of *linked twisted map*

Orbits described by (depending on parameter r):

$$\begin{cases} x_{n+1} &= \quad x_n + r\, y_n(1 - y_n) \mod 1 \\ y_{n+1} &= \quad y_n + r\, x_{n+1}(1 - x_{n+1}) \mod 1 \end{cases}$$

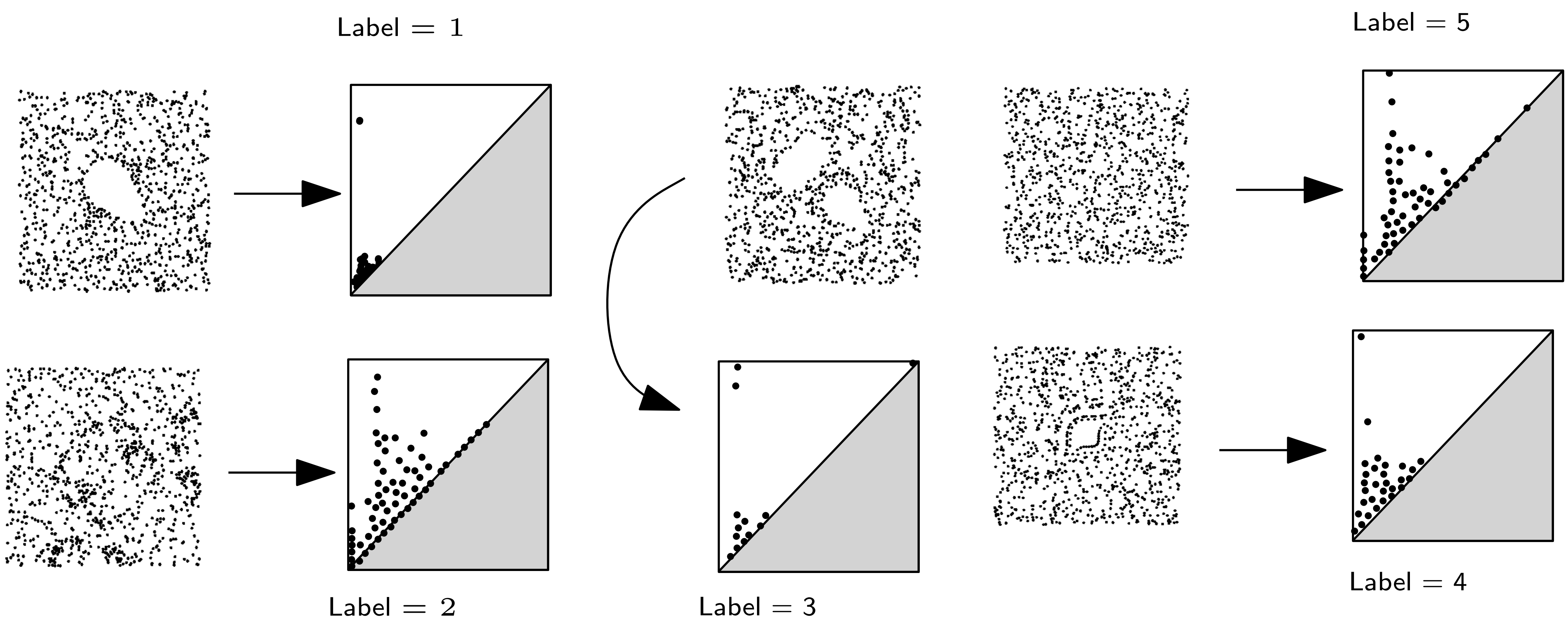


PersLay: adaptation to persistence diagrams

[Carrière, C., Ike, Lacombe, Royer, Umeda, AISTATS 2020]

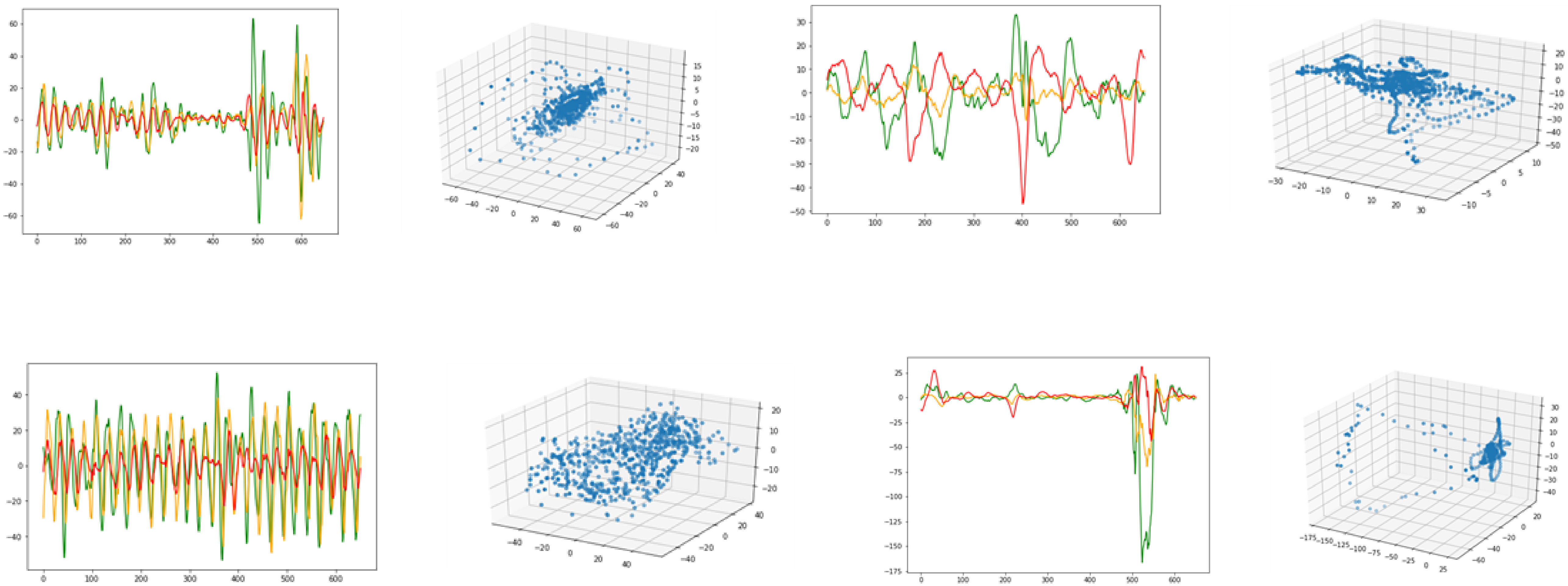
Example: classify orbits of *linked twisted map*

Dataset	PSS-K	PWG-K	SW-K	PF-K	PersLay
ORBIT5K	72.38(±2.4)	76.63(±0.7)	83.6(±0.9)	85.9(±0.8)	**87.7(±1.0)**
ORBIT100K	—	—	—	—	**89.2(±0.3)**



TDA and Machine Learning:
some illustrative examples on industrial applications

TDA and Machine Learning for sensor data



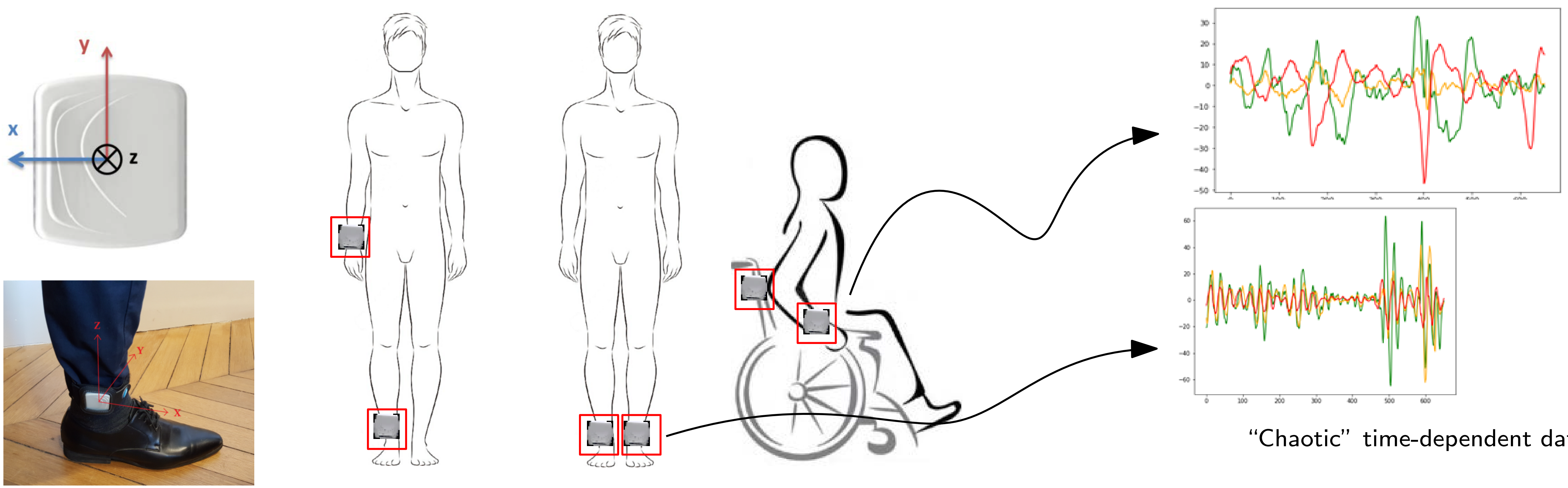
(Multivariate) time-dependent data can be converted into point clouds:
sliding window, time-delay embedding,...

With landscapes: patient monitoring

A joint industrial research project between

sysnav NAVIGATION TECHNOLOGIES and Inria informatics mathematics

A French SME with innovating technology to reconstruct pedestrian trajectories from inertial sensors (ActiMyo)



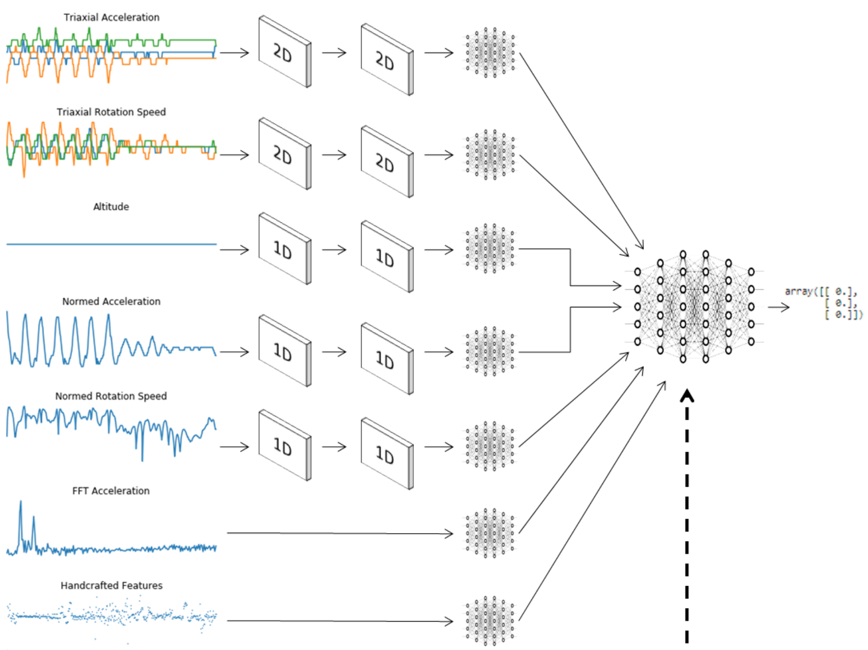
"Chaotic" time-dependent data

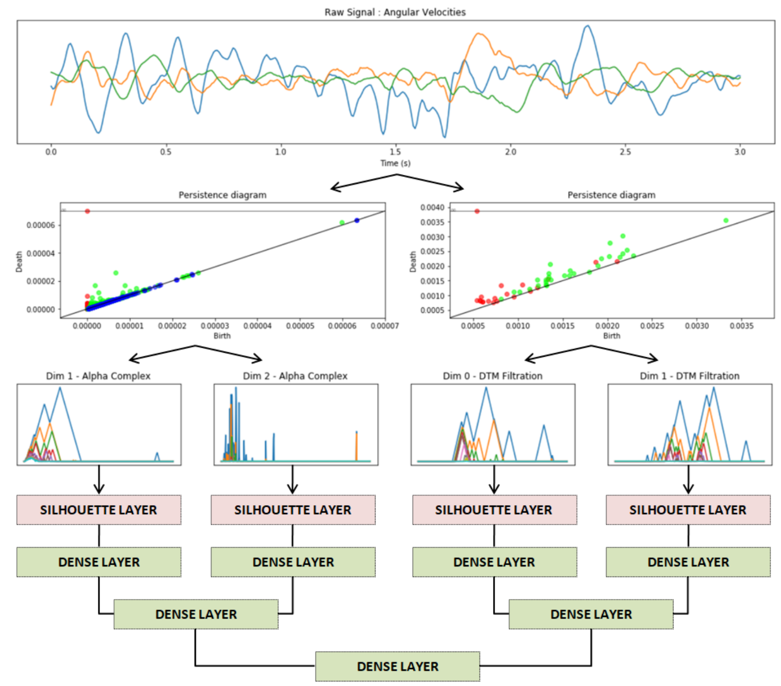
Objective: precise analysis of movements and activities of pedestrians.

Applications: personal healthcare; medical studies; defense.

With landscapes: patient monitoring

Example: Dyskinesia crisis detection and activity recognition:





Class	Naive	Multi	FEA	QUA	TDA
Walking	97.6	98.4	99.3	99.0	**99.5**
Upstairs	97.2	**99.8**	97.8	98.0	97.7
Downstairs	99.6	**99.7**	99.0	98.4	98.3
Sitting	87.1	93.1	89.7	91.8	**96.5**
Standing	87.0	97.7	97.2	97.2	**98.1**
Laying	92.4	**100.**	99.8	99.9	**100.**
Stand-Sit	90.8	**95.6**	89.1	91.3	93.4
Sit-Stand	**100.**	99.9	**100.**	**100.**	**100.**
Sit-Lie	87.1	81.1	84.2	90.0	**95.1**
Lie-Sit	81.4	81.8	85.9	**91.8**	87.9
Stand-Lie	74.2	**87.6**	86.5	87.4	81.5
Lie-Stand	80.4	72.1	**83.2**	77.7	**83.2**

Multi-channels CNN + TDA neural network

Results on publicly available data set (HAPT) - improve the state-of-the-art.

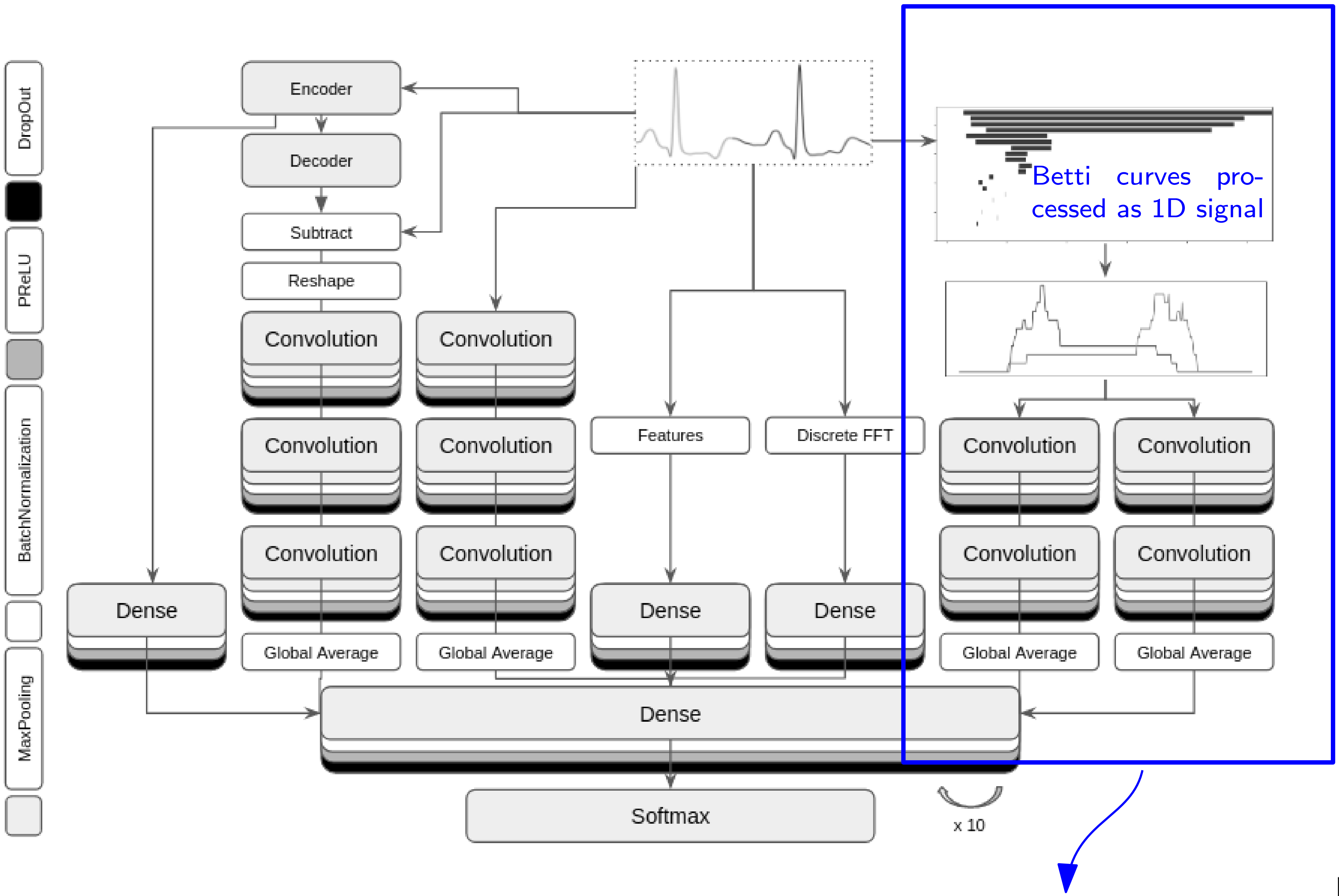
- Data collected in non controlled environments (home) are very chaotic.
- Data registration (uncertainty in sensors orientation/position).
- Reliable and robust information is mandatory.
- Events of interest are often rare and difficult to characterize.

sysnav NAVIGATION TECHNOLOGIES

Inria informatics mathematics

TDA-DL pipeline for arrhythmia detection

Objective: Arrythmia detection from ECG data.



- Improvement over state-of-the-art.
- Better generalization.

	Accuracy[%]
UCLA (2018)	93.4
Li et al. (2016)	94.6
Inria-Fujitsu (2018)*	**98.6**

FUJITSU Inria informatics mathematics

Thank you for your attention!