PersLay: A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures

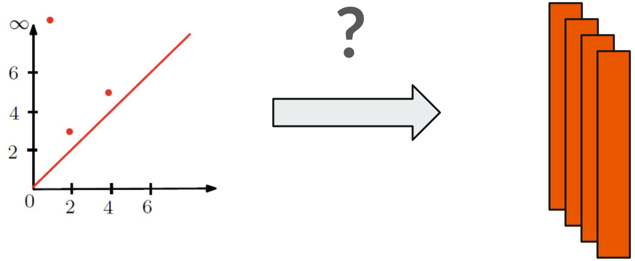
[Mathieu Carriere, Frederic Chazal, Yuichi Ike, Theo Lacombe, Martin Royer, Yuhei Umeda]

Thomas Gravier
Blandine Gorce
Laura Choquet

1. Introduction + Related Work
2. Paper Contributions
3. Network and Training
4. Results
5. Discussion

1. Introduction

Context: persistence diagrams are one of the main topological features, but they are difficult to use in machine learning pipeline due to the lack of structure of the space.



→ Solution: **vectorization** of the persistence diagrams

Related works:

- Real-valued functions on graphs
- Already existing vectorisation
- Deep Sets network [Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., & Smola, A.. *Deep sets.*]

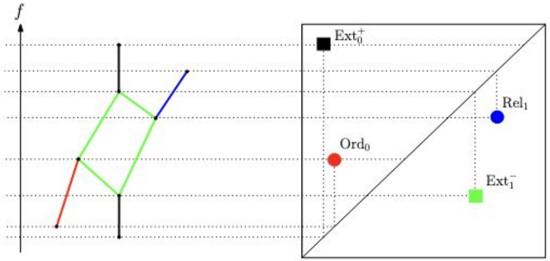
Current limitations:

- Few trainable parameters
- Hard to know choose vectorisation for a specific task
- Kernel methods are not optimized in terms of computation times

1. Introduction

Solution proposed by the paper:

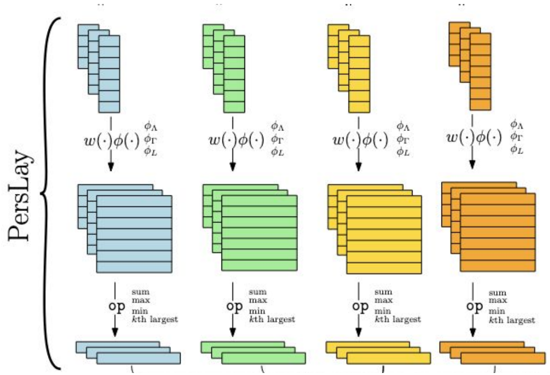
→ A framework to learn the vectorisation adapted to the task generalizing some of the vectorisation already existing.



Extended persistence diagrams

$$hks_{G,t} : \nu \mapsto \sum_{k=1}^{n} exp(-t\lambda_k)\psi_k(\nu)^2$$

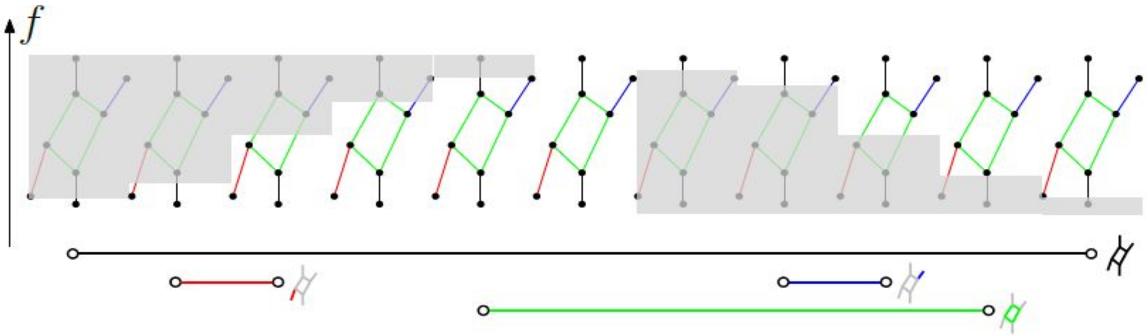
Heat kernel signatures



PersLay: a new neural network layer

1. Introduction and related work
2. Paper contributions
3. Network and Training
4. Results
5. Discussion

2. Extended persistence diagram



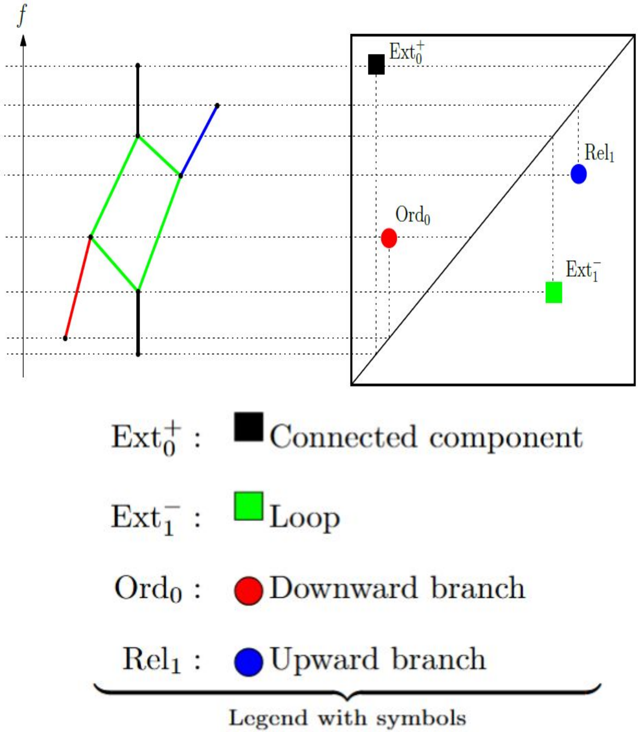
Filtration with up and sub levels

Up level :

$$[G=(V,E)][G_\alpha=(V_\alpha,E_\alpha)][V_\alpha=\{v\in V: f(v)\leq\alpha\}][E_\alpha=\{(v_1,v_2)\in E: v_1,v_2\in V_\alpha\}]$$

Sub level :

$$[X=(V,E)][X^\alpha=\{x\in X: f(x)\geq\alpha\}][E^\alpha=\{(x_1,x_2)\in E: x_1,x_2\in X^\alpha\}]$$



2. Heat Kernel Signatures on graphs

$$L_\omega = I - A^{-1/2} D A^{-1/2}$$

$$hks_{G,t} : \nu \mapsto \sum_{k=1}^{n} exp(-t\lambda_k)\psi_k(\nu)^2$$

λ_k Eigenvalues of the graph Laplacian

ψ_k Eigenfunctions of the graph Laplacian

t Diffusion parameter

Property :

- Stability w.r.t **graph perturbation** $d_B(Dg(G,t), Dg(G',t)) \leq C(G,t)||W||_F$
- Stability w.r.t **parameter t** $d_B(Dg(G,t), Dg(G,t')) \leq 2|t - t'|$

[Sun, J., Ovsjanikov, M., and Guibas, L. A concise and provably informative multi-scale signature based on heat diffusion.]

2. PersLay

$$\textsc{PersLay}(\mathrm{Dg}) := \mathsf{op}\left(\{w(p)\cdot\phi(p)\}_{p\in\mathrm{Dg}}\right)$$

any permutation invariant operation (sum, min...)

weight function (learnable)

point transformation function

Point transformation function:
Pour p un point du diagramme:

- Triangle point transformation: $\Lambda_p : t \to \max\{0,\, y - |t - x|\}$
- Gaussian point transformation: $\Gamma_p : t \to exp\left(-\, \|p - t\|_2^2 / \left(2\sigma^2\right)\right)$
- Line point transformation : $\mathbf{L}_\Delta : p \to \langle p, e_\Delta\rangle + b_\Delta$

Example with a triangle point transformation:

$$\phi : \mathbb{R}^2 \to \mathbb{R}^q$$

$$p \longrightarrow \begin{array}{c}\Lambda_p(t_1)\\ \Lambda_p(t_2)\\ \cdots\\ \Lambda_p(t_q)\end{array}$$

1. Introduction and related work
2. Paper contributions
3. **Network and Training**
4. Results
5. Discussion

3. Network

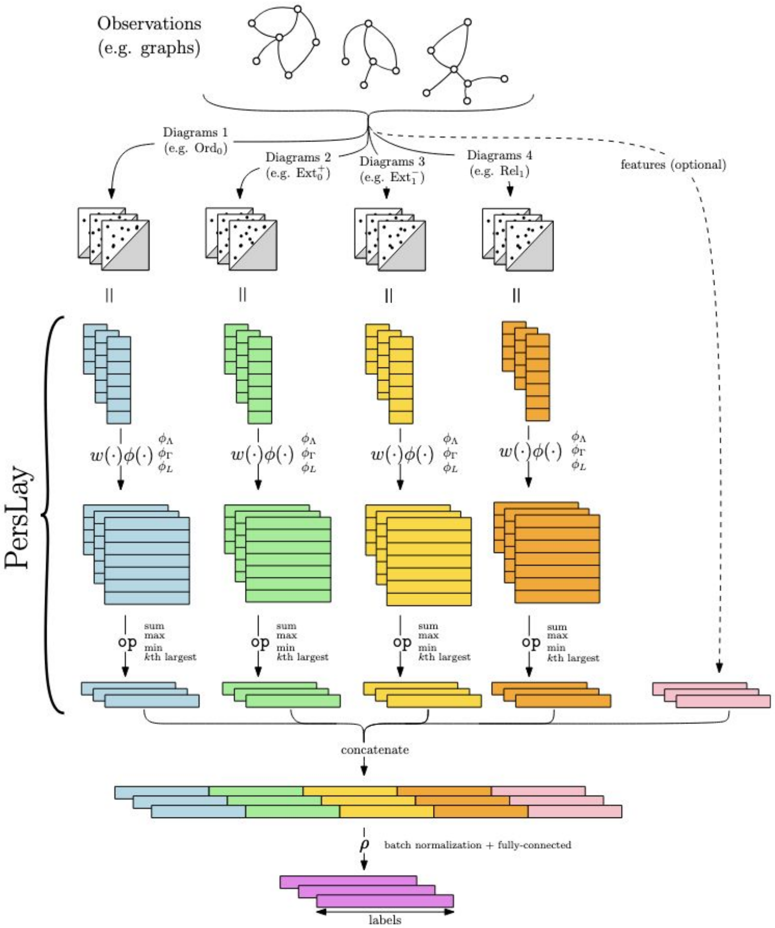
Network:

- 2 layers (PersLay + Fully connected layer)

Hyperparameters set-up:

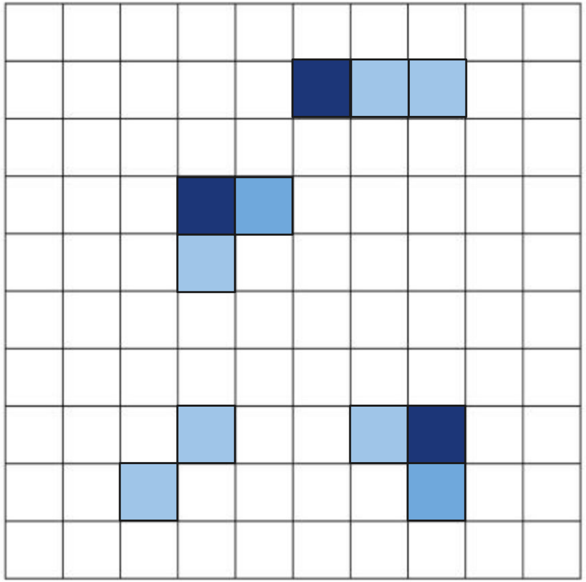
PersLay layer:

- op is the sum
- phi is the sum of the point transformation function



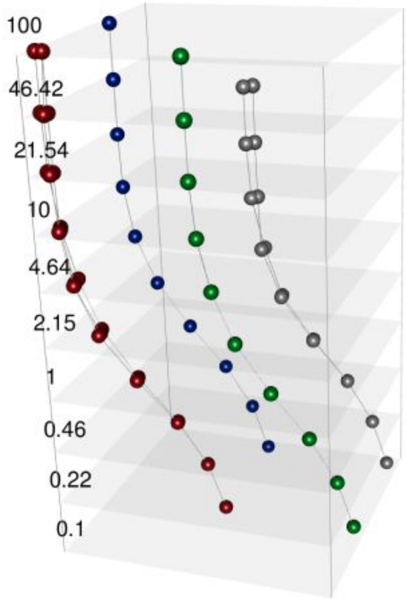
3. Choice of hyperparameters

How to set ω ?



Can we optimize t ?

The choice of t does not affect classification performance



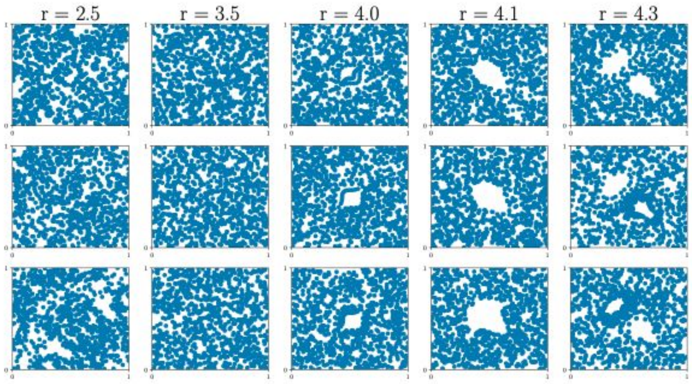
Evolution of $t \mapsto Dg(G,t)$ for one graph from the MUTAD dataset

1. Introduction and related work
2. Paper Contributions
3. Network and Training
4. **Results**
5. Discussion

4. A proof of concept: Perslay results with AlphaComplex filtration

- **ORBIT5K : Improve results** compared to state-of-the-art method
- **ORBIT100K :** Capable of handling large-scale datasets where kernel method fail

Orbits generated by differents choice of r :



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)**

4. Results

Ablation studies :

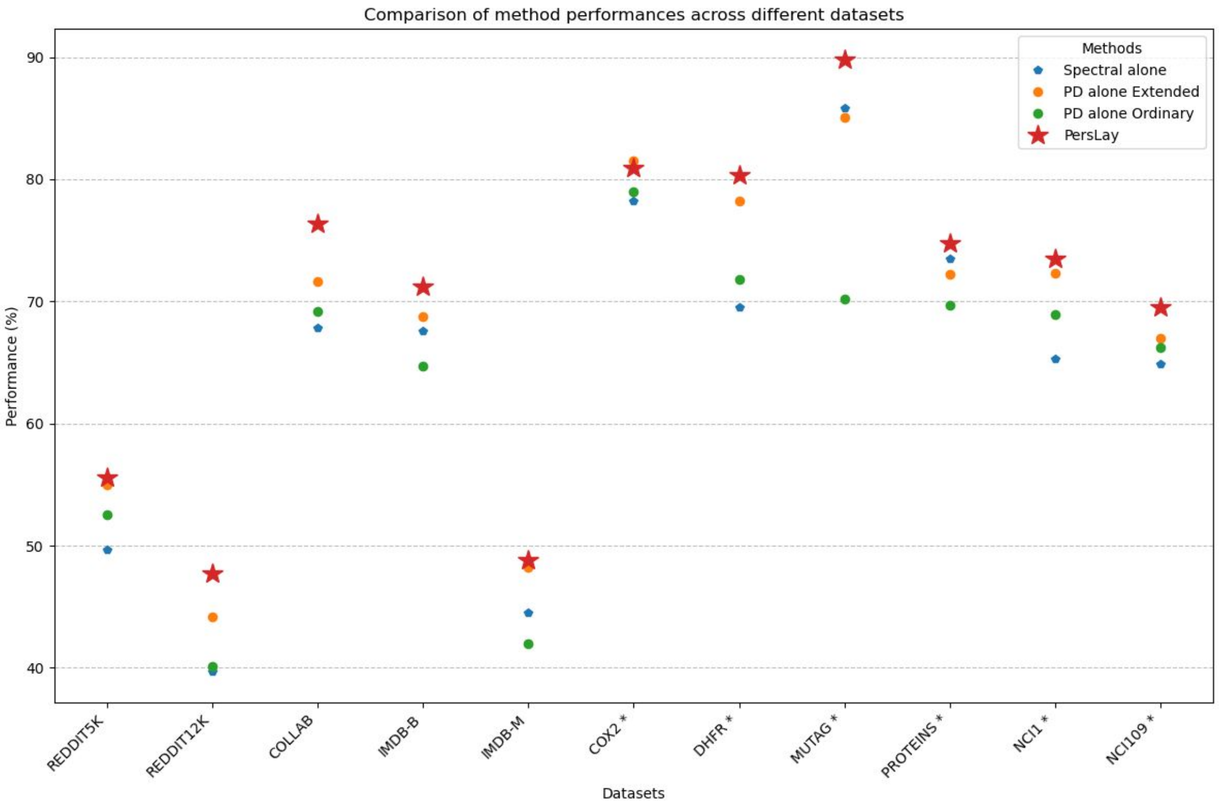
- Choice of the grid size is not a key parameter however too large grid size lead to overfitting
- The best choice of point transformation differs from one dataset to another.
- The sum permutation operator give better results

		Grid size for trainable weights w(p)					
		None	2×2	5×5	10×10	20×20	50×50
MUTAG	Train/Test acc (%)	92.3/88.9	91.1/88.8	91.7/89.6	92.3/**89.9**	93.7/88.3	94.1/87.7
COLLAB	Train/Test acc (%)	76.5/75.3	78.6/75.8	79.0/76.2	80.0/**76.5**	83.5/73.9	94.0/71.3

		Point transformation ϕ			Perm op	
		Gaussian	line	triangle	Sum	Max
MUTAG	Train/Test acc (%)	92.5/**89.7**	89.2/84.2	91.5/85.0	92.3/**89.5**	91.9/87.4
COLLAB	Train/Test acc (%)	79.7/75.3	79.9/**76.1**	79.4/74.7	80.0/**76.4**	78.8/75.0

4. Results

Ablation studies :



5. Discussion

Conclusion :

- Great improvement to add the graph topology for classification
- Important value added with the extended diagram to capture loops geometry
- Vectorization with reduction of the computational cost

To go further:

- The paper focuses on graph only, an extension can be the utilisation of PersLay to other applications.

MERCI

6. Our implementation

Appendice :

The adjacency matrix A of a graph G with vertex set : $V = \{v_1, v_2, ..., v_n\}$

$$A := \left(\mathbf{1}_{(v_i, v_j) \in E}\right)_{i,j}$$

Degree matrix : $D_{i,j} = \sum_j A_{i,j}$

Results - comparing to top five graphs

Dataset	SV[1]	RetGK* [2]	FGSD [3]	GCNN [4]	GIN [5]	PersLay	
						Mean	Max
REDDIT5K	—	56.1	47.8	52.9	57.0	55.6	56.5
REDDIT12K	—	48.7	—	46.6	—	47.7	49.1
COLLAB	—	81.0	80.0	79.6	80.1	76.4	78.0
IMDB-B	72.9	71.9	73.6	73.1	74.3	71.2	72.6
IMDB-M	50.3	47.7	52.4	50.3	52.1	48.8	52.2
COX2*	78.4	80.1	—	—	—	80.9	81.6
DHFR*	78.4	81.5	—	—	—	80.3	80.9
MUTAG*	88.3	90.3	92.1	86.7	89.0	89.8	91.5
PROTEINS*	72.6	75.8	73.4	76.3	75.9	74.8	75.9
NCI1*	71.6	84.5	79.8	78.4	82.7	73.5	74.0
NCI109*	70.5	—	78.8	—	—	69.5	70.1

Learning the weights