Emerging Topics in AI: Geometry and Topology

Bastian Rieck (@Pseudomanifold)

HELMHOLTZ MUNICH
TUM

An alternative title

Beyond Message Passing

Graphs are ubiquitous...

> *In many ways, graphs are the main modality of data we receive from nature. This is due to the fact that most of the patterns we see, both in natural and artificial systems, are elegantly representable using the language of graph structures. Prominent examples include molecules (represented as graphs of atoms and bonds), social networks and transportation networks.*

P. Veličković, 'Everything is connected: Graph neural networks', *Current Opinion in Structural Biology* 79, 2023, p. 102538

…but graph neural networks are not always up to the task

> *While GNNs have the ability to ignore the graph-structure in such cases, it is not clear that they will. In this work, we show that GNNs actually tend to overfit the graph-structure in the sense that they use it even when a better solution can be obtained by ignoring it.*

M. Bechler-Speicher, I. Amos, R. Gilad-Bachrach and A. Globerson, 'Graph Neural Networks Use Graphs When They Shouldn't', Preprint, 2023, arXiv: `2309.04332 [cs.LG]`

This talk

Rather than ask about *computational mechanisms*, think about *novel descriptors* about graphs.

This talk

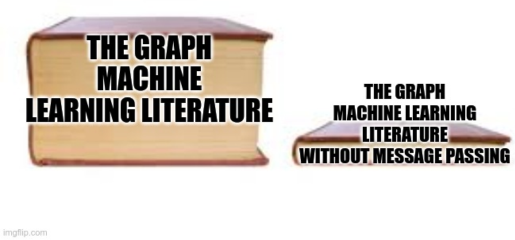
Rather than ask about *computational mechanisms*, think about *novel descriptors* about graphs.

Use concepts from *geometry* and *topology* to harness more information from graphs.

This talk

Rather than ask about *computational mechanisms*, think about *novel descriptors* about graphs.

Use concepts from *geometry* and *topology* to harness more information from graphs.



Collaborators

Celia Hacker

Christian Bock

Josh Southern

Jeremy Wayland

Kelly Maggs

Michael Bronstein

Geometry: Ollivier–Ricci Curvature

What is curvature?

Motivation

Characterise how ‘curved’ an object (a surface, a manifold, a topological space, …) is.
Curvature can be *extrinsic* or *intrinsic*.

What is curvature?

Motivation

Characterise how 'curved' an object (a surface, a manifold, a topological space, ...) is. Curvature can be *extrinsic* or *intrinsic*.

Gaussian curvature

Gaussian curvature K is the product of the *principal curvatures* κ_1, κ_2. It is an intrinsic property of a surface and does not depend on a specific embedding.

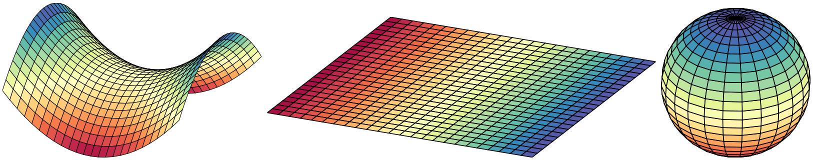
What is curvature?

Motivation

Characterise how ‘curved’ an object (a surface, a manifold, a topological space, ...) is. Curvature can be *extrinsic* or *intrinsic*.

Gaussian curvature

Gaussian curvature K is the product of the *principal curvatures* κ_1, κ_2. It is an intrinsic property of a surface and does not depend on a specific embedding.



$K < 0$ $K = 0$ $K > 0$

Ollivier–Ricci curvature

Let G be a graph with its shortest-path metric d and μ_v be a probability measure on G for node $v \in V$. The *Ollivier–Ricci curvature* of a pair of nodes $i \neq j \in V$ is then defined as

$$\kappa_{\mathrm{OR}}(i,j) := 1 - \frac{\mathrm{W}_1(\mu_i, \mu_j)}{\mathrm{d}(i,j)}, \tag{1}$$

where W_1 refers to the first *Wasserstein distance* between μ_i and μ_j.

History

First introduced by Ollivier for metric (measure) spaces, this notion of curvature was quickly adopted for use in the graph setting.

Y. Ollivier, 'Ricci curvature of Markov chains on metric spaces', *Journal of Functional Analysis* 256.3, 2009, pp. 810–864

How to pick μ_i?

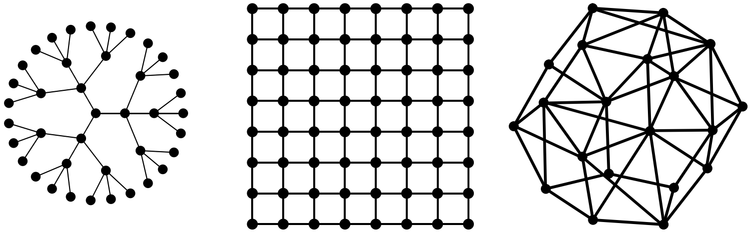
It is common practice to define a version of μ_i based on lazy random walks. Given a laziness parameter $\alpha \in [0, 1]$, we set

$$\mu_i(j) := \begin{cases} \alpha & \text{if } i = j \\ \frac{1-\alpha}{\deg(i)} & \text{if } i \neq j \text{ and } i \sim j \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $\deg(i)$ refers to the degree of node i.

Ollivier–Ricci curvature

Different regimes

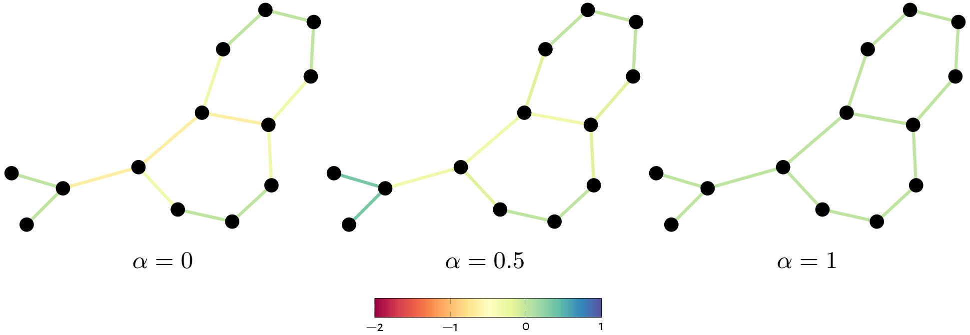


$\kappa_{\mathrm{OR}} < 0$ $\kappa_{\mathrm{OR}} = 0$ $\kappa_{\mathrm{OR}} > 0$

(figure inspired by K. Devriendt and R. Lambiotte, 'Discrete curvature on graphs from the effective resistance', *Journal of Physics: Complexity* 3.2, 2022, p. 025008)

Ollivier–Ricci curvature

Examples



Useful properties of κ_{OR}

Lower bound

It is sufficient to know the values of κ_{OR} for each edge (i, j). If $\kappa_{\mathrm{OR}}(i, j) \geq K$ for edges $(i, j) \in E$, then $\kappa_{\mathrm{OR}}(k, l) \geq K$ for *all* pairs of vertices (k, l).

Curvature characterises graphs

If $\kappa_{\mathrm{OR}}(i, j) \geq K > 0$ for all edges $(i, j) \in E$, then for $i, j \in V$, we have

$$\mathrm{d}(i, j) \leq \frac{\mathrm{W}_1(\delta_i, \mu_i) + \mathrm{W}_1(\delta_j, \mu_j)}{\kappa_{\mathrm{OR}}(i, j)}, \tag{3}$$

where δ_i, δ_j refer to Dirac probability measures centred at node i and j. Thus,

$$\mathrm{diam}(G) \leq \frac{\sup_i \mathrm{W}_1(\delta_i, \mu_i)}{K}. \tag{4}$$

How to leverage this in ~~practice~~ theory?

Success rate (↑) of distinguishing pairs of graphs in the 'BREC' data set when using different probability measures in the OR curvature calculation.

Method		Basic (56)	Regular (50)	STR (50)	Extension (97)	CFI (97)
1-WL		0.00	0.00	0.00	0.00	0.00
3-WL		**1.00**	**1.00**	0.00	**1.00**	**0.59**
κ_{OR}	$\mu_i^{(1)}$	**1.00**	0.96	0.06	0.87	0.00
	$\mu_i^{(2)}$	**1.00**	**1.00**	**0.14**	0.97	0.01
	$\mu_i^{(3)}$	**1.00**	**1.00**	**0.14**	0.99	0.04
	$\mu_i^{(4)}$	**1.00**	**1.00**	**0.14**	**1.00**	0.09
	$\mu_i^{(5)}$	**1.00**	**1.00**	**0.14**	**1.00**	0.19

Ollivier–Ricci curvature with *learnable* probability measures promises to lead to structural insights.

How to leverage this in practice?

Use curvature to evaluate *graph generative models*.

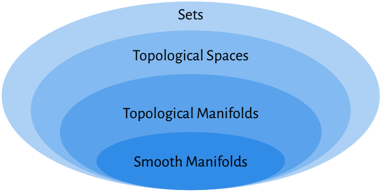
J. Southern*, J. Wayland*, M. Bronstein and **B. Rieck**, 'Curvature Filtrations for Graph Generative Model Evaluation', *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, arXiv: `2301.12906 [cs.LG]`, in press

Use curvature as an efficient *graph descriptor* for graph learning tasks.

L. O'Bray*, **B. Rieck*** and K. Borgwardt, 'Filtration Curves for Graph Representation', *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2021, pp. 1267–1275

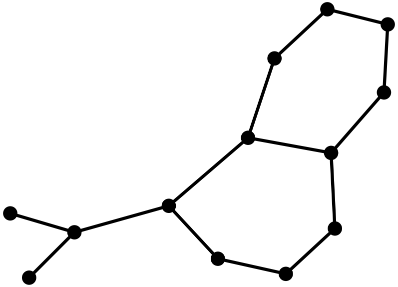
Topology: Multi-Scale Views on a Graph

Why topology?

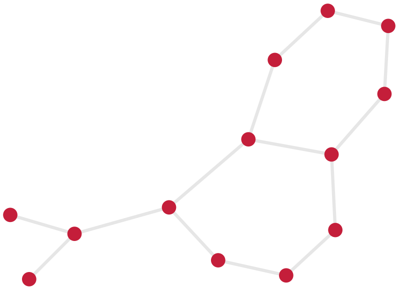


Most of machine learning happens at the level of smooth manifolds. A topological perspective is *more general* but also *coarser*.

Topological features of graphs

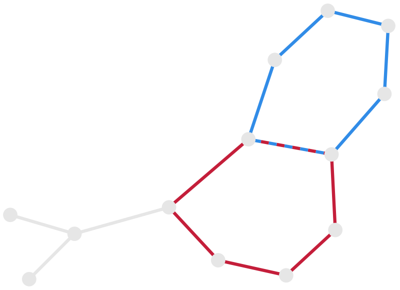


Topological features of graphs



β_0: Connected components

Topological features of graphs

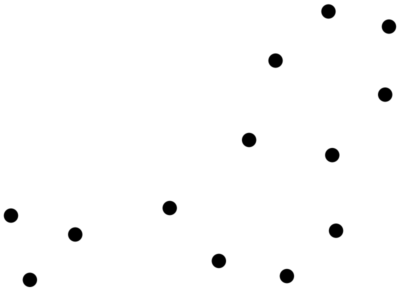


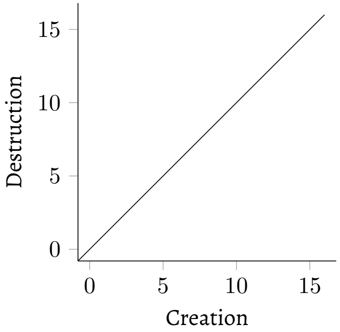
β_1: Cycles

The *Betti numbers* of a graph are easy to calculate but only provide coarse information.

Persistent homology

Intuition

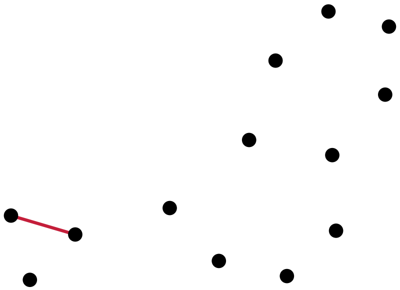


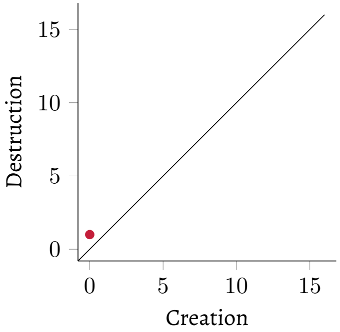


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

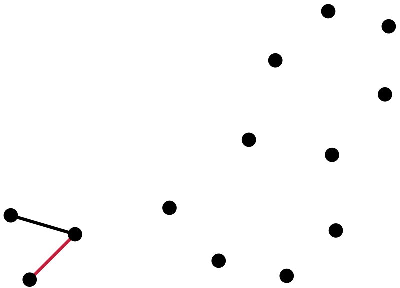


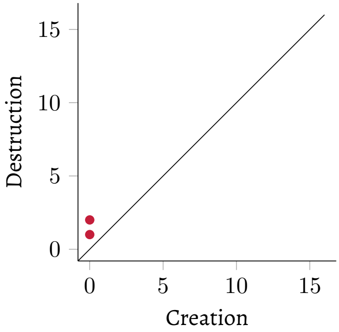


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

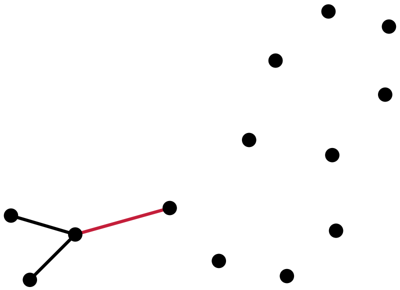


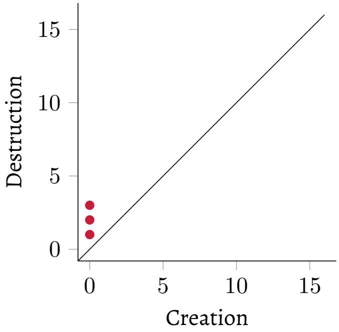


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

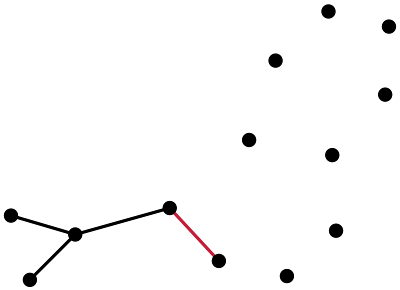


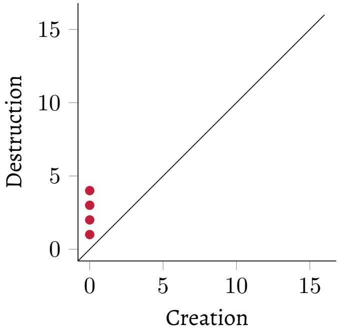


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

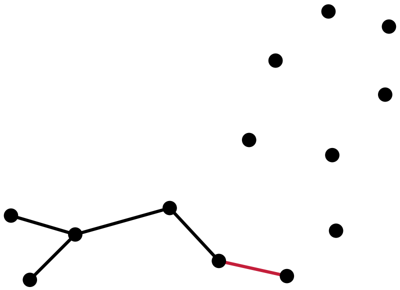


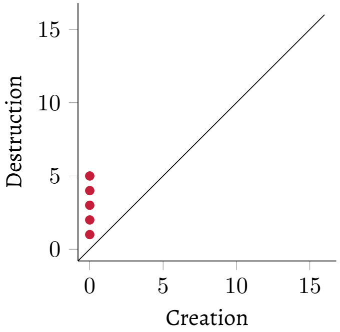


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

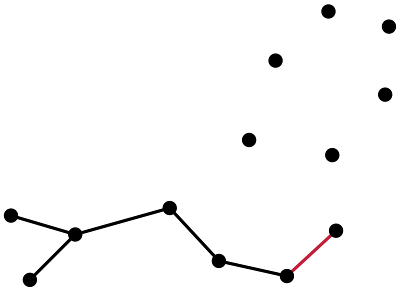


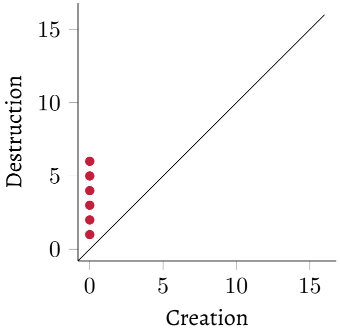


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

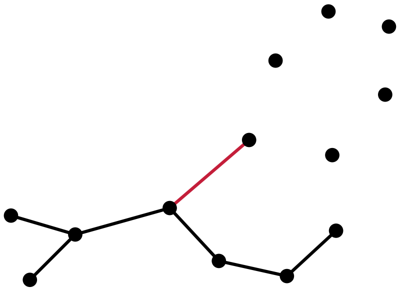


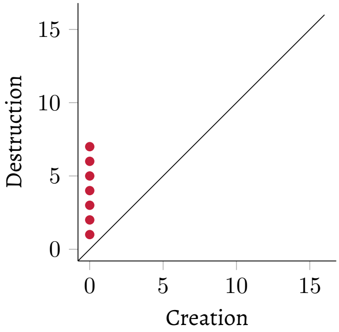


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

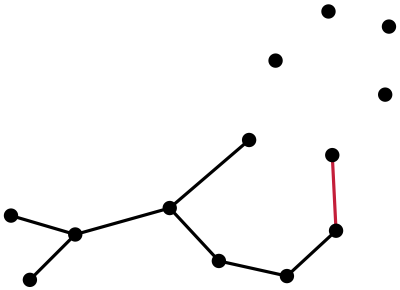


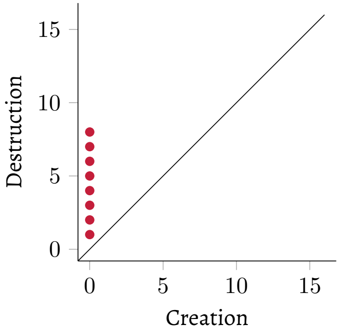


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

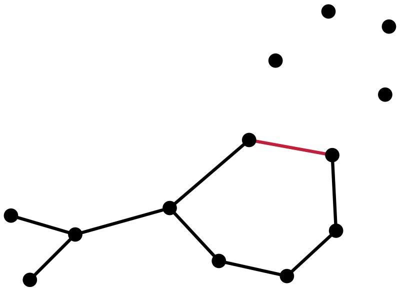


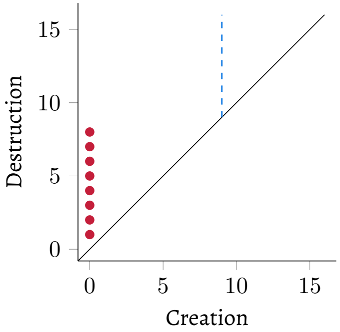


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

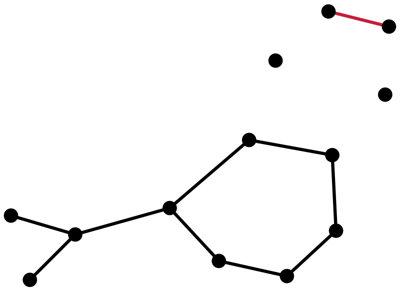


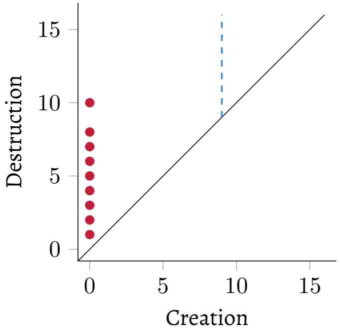


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

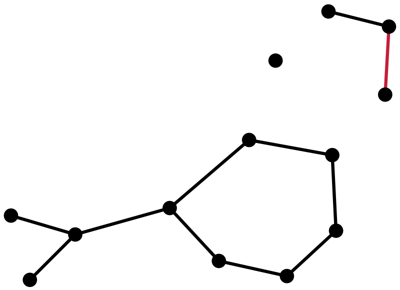


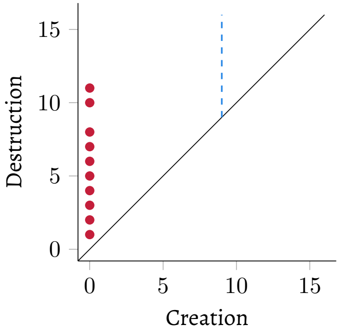


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

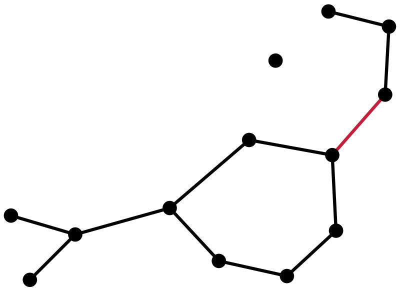


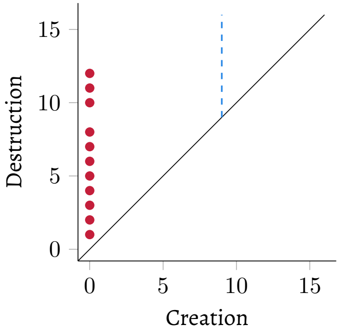


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

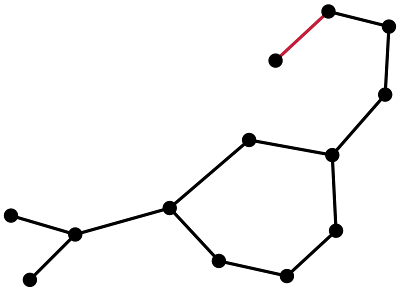


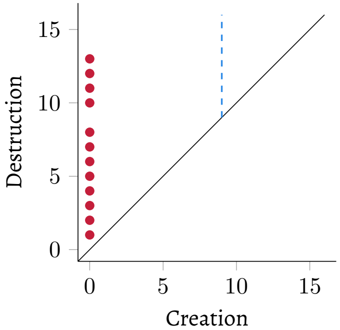


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

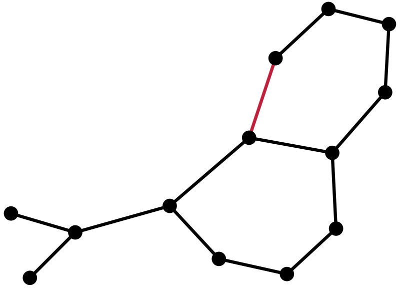


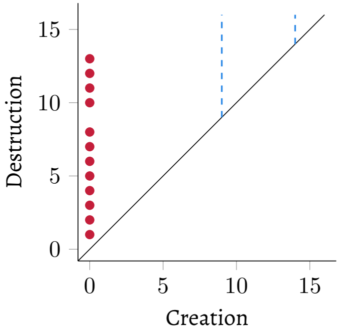


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

Intuition

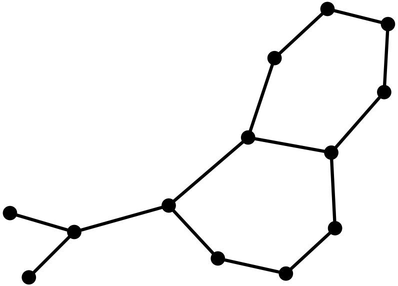


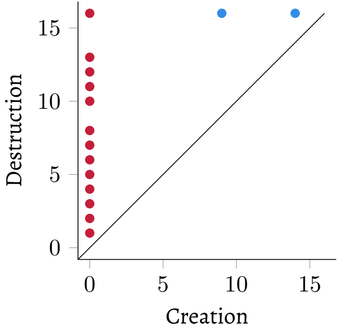


Use a *graph filtration* to observe how topological features of the graph change.

Persistent homology

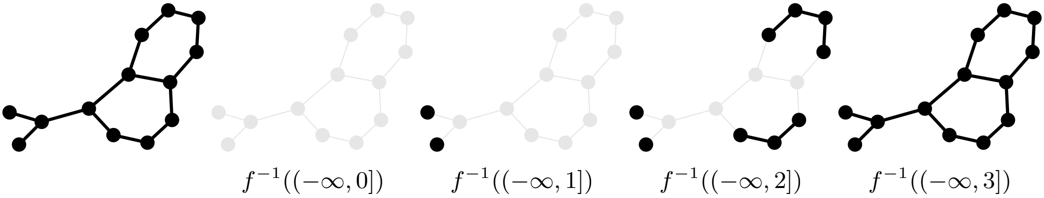
Intuition





Use a *graph filtration* to observe how topological features of the graph change.

More on filtrations



Filtrations can be *learned* and incorporate additional geometrical information about a space.

Why is this important?

Escaping the Weisfeiler–Le(h)man hierarchy

Persistent homology is *at least* as expressive as WL[1], the 1-dimensional Weisfeiler–Le(h)man test for graph isomorphism.

If information about k-cliques is available, persistent homology is *at least as expressive* as k-FWL, the 'folklore' variant of Weisfeiler–Le(h)man.

B. Rieck, 'On the Expressivity of Persistent Homology in Graph Learning', Preprint, 2023, arXiv: `2302.09826 [cs.LG]`

Circumventing message passing by using pre-defined filtrations

	D & D	MUTAG	NCI1	NCI109	PROTEINS	PTC-MR	PTC-FR	PTC-MM	PTC-FM
V-Hist	78.32 ± 0.35	85.96 ± 0.27	64.40 ± 0.07	63.25 ± 0.12	72.33 ± 0.32	58.31 ± 0.27	**68.13 ± 0.23**	66.96 ± 0.51	57.91 ± 0.83
E-Hist	72.90 ± 0.48	85.69 ± 0.46	63.66 ± 0.11	63.27 ± 0.07	72.14 ± 0.39	55.82 ± 0.00	65.53 ± 0.00	61.61 ± 0.00	59.03 ± 0.00
RetGK*	**81.60 ± 0.30**	90.30 ± 1.10	84.50 ± 0.20		75.80 ± 0.60	62.15 ± 1.60	67.80 ± 1.10	67.90 ± 1.40	63.90 ± 1.30
WL	79.45 ± 0.38	87.26 ± 1.42	85.58 ± 0.15	84.85 ± 0.19	**76.11 ± 0.64**	63.12 ± 1.44	67.64 ± 0.74	67.28 ± 0.97	64.80 ± 0.85
Deep-WL*		82.94 ± 2.68	80.31 ± 0.46	80.32 ± 0.33	75.68 ± 0.54	60.08 ± 2.55			
P-WL	79.34 ± 0.46	86.10 ± 1.37	85.34 ± 0.14	84.78 ± 0.15	75.31 ± 0.73	63.07 ± 1.68	67.30 ± 1.50	**68.40 ± 1.17**	64.47 ± 1.84
P-WL-C	78.66 ± 0.32	**90.51 ± 1.34**	85.46 ± 0.16	**84.96 ± 0.34**	75.27 ± 0.38	**64.02 ± 0.82**	67.15 ± 1.09	**68.57 ± 1.76**	**65.78 ± 1.22**
P-WL-UC	78.50 ± 0.41	85.17 ± 0.29	**85.62 ± 0.27**	**85.11 ± 0.30**	75.86 ± 0.78	**63.46 ± 1.58**	67.02 ± 1.29	**68.01 ± 1.04**	**65.44 ± 1.18**

This procedure imbues the Weisfeiler–Le(h)man kernel with additional information about the topology of a graph.
B. Rieck* , C. Bock* and K. Borgwardt, 'A Persistent Weisfeiler–Lehman Procedure for Graph Classification', *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 5448–5458

Augmenting message passing with a topology-based layer

Method	DD	ENZYMES	MNIST	PROTEINS
GCN-4	68.0 ± 3.6	22.0 ± 3.3	76.2 ± 0.5	68.8 ± 2.8
GCN-3-TOGL-1	**75.1 ± 2.1**	**30.3 ± 6.5**	**84.8 ± 0.4**	**73.8 ± 4.3**
GIN-4	75.6 ± 2.8	21.3 ± 6.5	83.4 ± 0.9	**74.6 ± 3.1**
GIN-3-TOGL-1	**76.2 ± 2.4**	**23.7 ± 6.9**	**84.4 ± 1.1**	73.9 ± 4.9
GAT-4	63.3 ± 3.7	21.7 ± 2.9	63.2 ± 9.4	67.5 ± 2.6
GAT-3-TOGL-1	**75.7 ± 2.1**	**23.5 ± 6.1**	**77.2 ± 9.5**	**72.4 ± 4.6**

The GNN part has been disabled by using only random node features. Performance is thus entirely driven by topological structures.

M. Horn*, E. De Brouwer*, M. Moor, Y. Moreau, **B. Rieck**† and K. Borgwardt†, 'Topological Graph Neural Networks', *International Conference on Learning Representations (ICLR)*, 2022, arXiv: `2102.07835 [cs.LG]`

Geometry & Topology: Differential k Forms

Taking stock

Status quo

Representation learning = mapping things into $\mathbb{R}^d$

Taking stock

Status quo

Representation learning = mapping things into $\mathbb{R}^d$

The maps are *learnable*

Taking stock

Status quo

- Representation learning = mapping things into $\mathbb{R}^d$
- The maps are *learnable*
- Message passing is but one way of doing things

Taking stock

Status quo

Representation learning = mapping things into $\mathbb{R}^d$

The maps are *learnable*

Message passing is but one way of doing things

Can we establish *bridges* between geometry and topology?

Cohomology studies maps from objects to $\mathbb{R}$.

Taking stock

Status quo

Representation learning = mapping things into $\mathbb{R}^d$

The maps are *learnable*

Message passing is but one way of doing things

Can we establish *bridges* between geometry and topology?

Cohomology studies maps from objects to $\mathbb{R}$.

Can we learn such maps?

Language

Topologists think in terms of *complexes*, such as *simplicial complexes*.

Language

Topologists think in terms of *complexes*, such as *simplicial complexes*.

A simplicial complex generalises a graph.

Language

Topologists think in terms of *complexes*, such as *simplicial complexes*.

A simplicial complex generalises a graph.

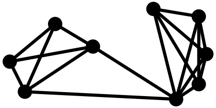
A good mental model is a graph together with its cliques.

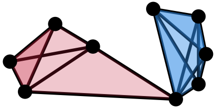
Language

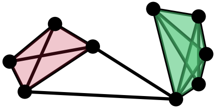
Topologists think in terms of *complexes*, such as *simplicial complexes*.

A simplicial complex generalises a graph.

A good mental model is a graph together with its cliques.







Some heavy lifting

Topology

A *singular chain* is a function $\sigma_i \colon \Delta^k \to \mathbb{R}^d$, assigning a vector to the *standard simplex*.

Some heavy lifting

Topology

A *singular chain* is a function $\sigma_i \colon \Delta^k \to \mathbb{R}^d$, assigning a vector to the *standard simplex*.

Singular cochains are linear functionals over $C_k^{\text{sing}}(\mathbb{R}^d, \mathbb{R}) := \{\sum_i \lambda_i \sigma_i \mid \lambda_i \in \mathbb{R}\}$, the set of all singular chains.

Some heavy lifting

Topology

A *singular chain* is a function $\sigma_i\colon \Delta^k \to \mathbb{R}^d$, assigning a vector to the *standard simplex*.

Singular cochains are linear functionals over $C_k^{\text{sing}}(\mathbb{R}^d, \mathbb{R}) := \{\sum_i \lambda_i \sigma_i \mid \lambda_i \in \mathbb{R}\}$, the set of all singular chains.

Machine learning

Singular cochains play the role of *feature maps* for simplices.

Where's geometry?

Differential k-forms

A differential k-form represents the area of a k-cube in subspace of the tangent space.

Where's geometry?

Differential k-forms

A differential k-form represents the area of a k-cube in subspace of the tangent space.

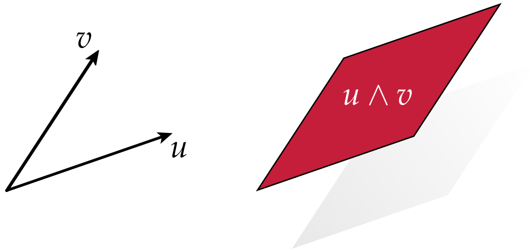
Differential forms generalise the concept of 'infinitesimal areas' to manifolds.

Where's geometry?

Differential k-forms

A differential k-form represents the area of a k-cube in subspace of the tangent space.

Differential forms generalise the concept of 'infinitesimal areas' to manifolds.



(figure modified from K. Crane, F. de Goes, M. Desbrun and P. Schröder, 'Digital Geometry Processing with Discrete Exterior Calculus', *ACM SIGGRAPH 2013 Courses*, ACM, 2013)

Putting everything together

Integrating differential k-forms over a space creates singular cochains.

Putting everything together

Integrating differential k-forms over a space creates singular cochains.

Thus, if we make differential k-forms learnable, we can obtain learnable maps for representing simplicial complexes!

Putting everything together

Integrating differential k-forms over a space creates singular cochains.

Thus, if we make differential k-forms learnable, we can obtain learnable maps for representing simplicial complexes!

No message passing required.

Putting everything together

Integrating differential k-forms over a space creates singular cochains.

Thus, if we make differential k-forms learnable, we can obtain learnable maps for representing simplicial complexes!

No message passing required.

The fact that this works at all is a consequence of *de Rham's theorem*, a deep result that connects geometry and topology.

Preliminary results

Results (mean AUROC and standard deviation of 5 runs) on molecular graph benchmark data sets that exhibit 'geometrical' node features. All models have roughly the same architecture.

	No. Parameters	BACE	BBBP	HIV
GAT	135K	69.52 ± 17.52	76.51 ± 3.36	56.38 ± 4.41
GCN	133K	66.79 ± 1.56	73.77 ± 3.30	68.70 ± 1.67
GIN	282K	42.91 ± 18.56	61.66 ± 19.47	55.28 ± 17.49
Differential k-forms	8.6K	**83.50 ± 0.55**	**86.41 ± 3.64**	**76.70 ± 2.17**

Why should we care?

'Learning more with less.'

Why should we care?

'Learning more with less.'

New paradigms can help advance the field!

Why should we care?

'Learning more with less.'

New paradigms can help advance the field!

Towards more sustainable models?

Conclusion

Our research

https://aidos.group
https://github.com/aidos-lab

AATRN
Applied Algebraic Topology
Research Network

https://aatrn.net