Two Households, Both Alike In Dignity

Geometry and Topology in Machine Learning

Bastian Rieck

SPIRES 2024 · Oxford

UNI FR

Goals of this talk

❶ Get people interested in topology. ✓

❷ Provide an introduction to modern machine learning on graphs.

❸ Show that machine learning benefits from different perspectives.

Machine learning needs more maths.

Machine learning

What is a neural network?

Perspective I

Theorem (Universal function approximation)

Let $\sigma \in C(\mathbb{R}, \mathbb{R})$ be a non-polynomial activation function. For every $n, m \in \mathbb{N}$, every compact subset $K \subseteq \mathbb{R}^n$, every function $f \in C(K, \mathbb{R}^m)$ and $\epsilon > 0$, there exist $k \in \mathbb{N}$, $\mathbf{A} \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, and $\mathbf{C} \in \mathbb{R}^{m \times k}$ such that

$$\sup_{x \in K} \|f(x) - g(x)\| < \epsilon,$$

where $g(x) = \mathbf{C}\sigma(\mathbf{A}x + b)$.

A. Pinkus, *Approximation theory of the MLP model in neural networks*, *Acta Numerica* 8, 1999, pp. 143–195

What is a neural network?

Perspective II

Geometry already received a lot of attention…

M. M. Bronstein, J. Bruna, T. Cohen and P. Veličković, *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*, 2021, arXiv: 2104.13478

…but not all of its paradigms are required in practice

AlphaFold 3 predicts the structure and interactions of all of life's molecules

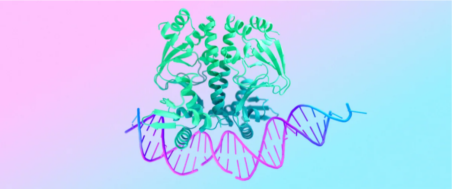
May 08, 2024
6 min read

Introducing AlphaFold 3, a new AI model developed by Google DeepMind and Isomorphic Labs. By accurately predicting the structure of proteins, DNA, RNA, ligands and more, and how they interact, we hope it will transform our understanding of the biological world and drug discovery.

Google DeepMind
AlphaFold team

Isomorphic Labs

Share



We find that ***no invariance or equivariance*** *with respect to global rotations and translation of the molecule are required in the architecture […]*

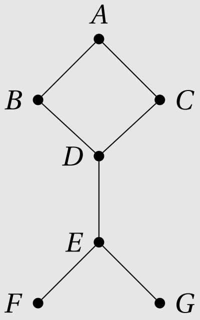
Geometry in Machine Learning

Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

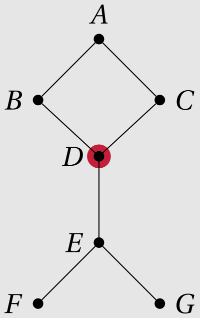


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

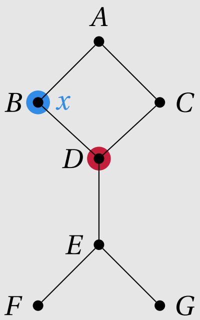


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

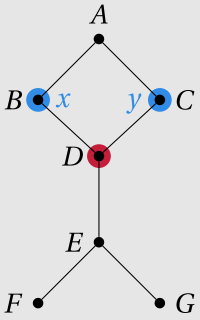


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

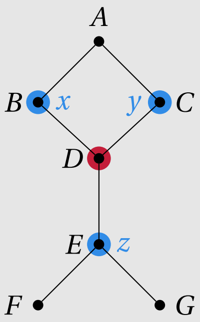


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

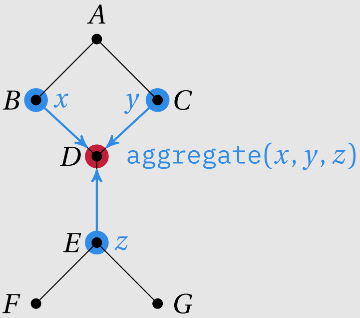


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.



Message passing

Example

Setup

- $(G, \mathbf{X})$: graph with n vertices and node attributes in $\mathbb{R}^d$
- $h_i^{(t)}$: hidden attributes of vertex i at step t

Message passing

Example

Setup

- $(G, \mathbf{X})$: graph with n vertices and node attributes in $\mathbb{R}^d$
- $h_i^{(t)}$: hidden attributes of vertex i at step t

A simple message-passing neural network

$$h_i^{(t)} = \sigma\left(\mathbf{W_1^{(t)}} h_i^{(t-1)} + \mathbf{W_2^{(t)}} \sum_{j \sim i} h_j^{(t-1)}\right)$$

Message passing

Example

Setup

- $(G, \mathbf{X})$: graph with n vertices and node attributes in $\mathbb{R}^d$
- $h_i^{(t)}$: hidden attributes of vertex i at step t

A simple message-passing neural network

$$h_i^{(t)} = \sigma\left(\mathbf{W_1^{(t)}} h_i^{(t-1)} + \mathbf{W_2^{(t)}} \sum_{j \sim i} h_j^{(t-1)}\right)$$

Generalisations

Message passing also works on topological domains like cell complexes, CW complexes, or simplicial complexes!

Is Message Passing the Right Paradigm?

Do graph neural networks learn the *right* things?

Do graph neural networks learn the *right* things?

> *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*, Proceedings of the 41st International Conference on Machine Learning, ed. by R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett and F. Berkenkamp, vol. 235, Proceedings of Machine Learning Research, PMLR, 2024, pp. 3284–3304

Are our data sets *representative*?

Are our data sets *representative*?

> *Our first finding is that standard benchmark graphs […] cover only a small region of this graph space that* `GraphWorld` *is able to cover via synthetic graph generation.*

J. Palowitch, A. Tsitsulin, B. Mayer and B. Perozzi, `GraphWorld`*: Fake Graphs Bring Real Insights for GNNs*, Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD), 2022, pp. 3691–3701

Are the *edges* of the graph relevant?

Are the *edges* of the graph relevant?

> *More recently, there is a trend to decouple the input graph from the graph used for information propagation.*

J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong and M. M. Bronstein, *Understanding over-squashing and bottlenecks on graphs via curvature*, International Conference on Learning Representations, 2022

Does knowledge of the *graph topology* help?

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

- Providing a new layer for use in graph neural networks.

Does knowledge of the *graph topology* help?

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

- Providing a new layer for use in graph neural networks.
- Create new synthetic data sets with pronounced topological features.

Does knowledge of the *graph topology* help?

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

- Providing a new layer for use in graph neural networks.
- Create new synthetic data sets with pronounced topological features.
- Getting high predictive performance on these data sets. 🥳

Does knowledge of the *graph topology* help?

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

- Providing a new layer for use in graph neural networks.
- Create new synthetic data sets with pronounced topological features.
- Getting high predictive performance on these data sets. 🥳
- Getting mediocre performance on other benchmark data sets. 😢

Does knowledge of the *graph topology* help?

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

- Providing a new layer for use in graph neural networks.
- Create new synthetic data sets with pronounced topological features.
- Getting high predictive performance on these data sets. 🥳
- Getting mediocre performance on other benchmark data sets. 😢
- Getting better performance if node features are randomised. 😱

Does knowledge of the *graph topology* help?

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

- Providing a new layer for use in graph neural networks.
- Create new synthetic data sets with pronounced topological features.
- Getting high predictive performance on these data sets. 🥳
- Getting mediocre performance on other benchmark data sets. 😢
- Getting better performance if node features are randomised. 😱

On data sets with pronounced topological structures, we found that our method helps GNNs obtain substantial gains in predictive performance.

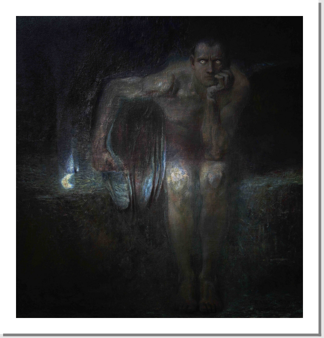
Going back to the data

C. Coupette, J. Vreeken and **B. Rieck**, *All the World's a (Hyper)Graph: A Data Drama*, *Digital Scholarship in the Humanities* 39.1, 2024, pp. 74–96, arXiv: `2206.08225 [cs.LG]`, URL: `https://hyperbard.net`

We introduce a novel dataset,
With full documentation as Appendix.
Raw data stem from all of Shakespeare's plays,
We model them as graphs in many ways,
And demonstrate representations matter.
The data readily accessible,
All code is publicly available.
What follows, to avoid redundancy,
Conveys our main ideas, as you will see
A tragedy in the Community.

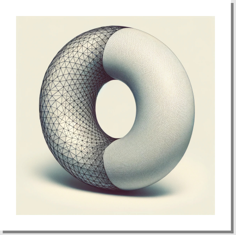
Taking stock

- Our graphs might be wrong.
- Our data sets are not representative.
- Our models may not be learning the 'right' things.



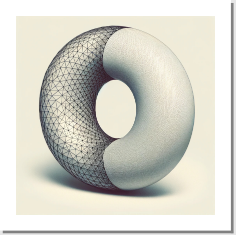
Rethinking the Foundations

Geometry and topology are *dual*



Geometry	=	fine details	+	quantitative answers
Topology	=	fundamental properties	+	qualitative answers

Geometry and topology are *dual*



Geometry	=	fine details	+	quantitative answers
Topology	=	fundamental properties	+	qualitative answers

Data has shape, shape has meaning, and meaning begets understanding.

(Paraphrasing Gunnar Carlsson's paradigm)

What is representation learning anyway?

Slogan

Mapping *things* into *vectors* in $\mathbb{R}^n$ via maps that are often *learnable*.

What is representation learning anyway?

Slogan

Mapping *things* into *vectors* in $\mathbb{R}^n$ via maps that are often *learnable*.

- T. Mikolov, K. Chen, G. Corrado and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, Preprint, 2013, arXiv: `1301.3781 [cs.CL]`
- A. Grover and J. Leskovec, *`node2vec`: Scalable Feature Learning for Networks*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864
- C. Hacker, *`k-simplex2vec`: A Simplicial Extension of `node2vec`*, 'Topological Data Analysis and Beyond' Workshop at NeurIPS, 2020

Using cochains for representation learning

A simplicial perspective

Using cochains for representation learning

A simplicial perspective

Suggestion

- Each simplex is represented by *evaluating* all cochains.

Using cochains for representation learning

A simplicial perspective

Suggestion

- Each simplex is represented by *evaluating* all cochains.
- Cochains are *feature maps* for the representations.

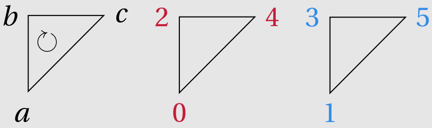
Using cochains for representation learning

A simplicial perspective

Suggestion

- Each simplex is represented by *evaluating* all cochains.
- Cochains are *feature maps* for the representations.

Example



$$\begin{matrix} a \\ b \\ c \end{matrix} \begin{pmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{pmatrix}$$

Using higher-order cochains with differential forms

K. Maggs, C. Hacker and **B. Rieck**, *Simplicial Representation Learning with Neural k-forms*, International Conference on Learning Representations, 2024, arXiv: `2312.08515 [cs.LG]`

Kelly

Celia

Transcending message passing

- Need a simplicial complex S with node embeddings in $\mathbb{R}^n$.
- Extend node embeddings to $\phi\colon S \to \mathbb{R}^n$.
- Learn differential k-forms on the ambient space.
- Obtain representations by *integration*.

In a nutshell

$$\underbrace{\Omega^k(\mathbb{R}^n) \xrightarrow{\int} C^k_{\text{sing}}(\mathbb{R}^n;\mathbb{R})}_{\text{de Rham map}} \underbrace{\xrightarrow{\phi^*} C^k_{\text{simp}}(S;\mathbb{R})}_{\text{Restriction map}}$$

Dramatis personæ

- $\Omega^k(\mathbb{R}^n)$: k-forms on $\mathbb{R}^n$
- $C^k_{\text{sing}}(\mathbb{R}^n;\mathbb{R})$: singular cochains
- $C^k_{\text{simp}}(S;\mathbb{R})$: simplicial cochains
- $\phi^*\gamma(\sigma) = \gamma(\phi|_\sigma)$ for a singular cochain $\gamma \in C^k_{\text{sing}}(\mathbb{R}^n;\mathbb{R})$.

Slogan

We replace the evaluation of feature cochains by the *integration* of differential feature forms.

How to learn differential k-forms?

Decomposition

Every k-form $\omega \in \Omega^k(\mathbb{R}^n)$ decomposes as

$$\omega = \sum_I f_I dx_{i_1} \wedge dx_{i_2} \wedge \ldots \wedge dx_{i_k}.$$

Neural k-forms

Given a neural network $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k}}$, its *neural k-form* is

$$\omega(\sigma) = \sum_I \sigma_I dx_{i_1} \wedge dx_{i_2} \wedge \ldots \wedge dx_{i_k}.$$

Neural k-forms = MLPs

In practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha\colon S_\alpha \to \mathbb{R}^n\}$.

In practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.
- Use MLP to learn $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k} \times \ell}$.

In practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.
- Use MLP to learn $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k} \times \ell}$.
- Integrate (S_α, ϕ_α), building the *integration matrix* $X(\alpha, \sigma) := \left[\int_{\phi_\alpha} \omega(\sigma)\right] \in \mathbb{R}^{|S_\alpha| \times \ell}$.

In practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.
- Use MLP to learn $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k} \times \ell}$.
- Integrate (S_α, ϕ_α), building the *integration matrix* $X(\alpha, \sigma) := \left[\int_{\phi_\alpha} \omega(\sigma) \right] \in \mathbb{R}^{|S_\alpha| \times \ell}$.
- Reduce $X(\alpha, \sigma)$ via $\operatorname{diag}(X(\alpha, \sigma)^T X(\alpha, \sigma))$, for instance.

In practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.
- Use MLP to learn $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k} \times \ell}$.
- Integrate (S_α, ϕ_α), building the *integration matrix* $X(\alpha, \sigma) := \left[\int_{\phi_\alpha} \omega(\sigma) \right] \in \mathbb{R}^{|S_\alpha| \times \ell}$.
- Reduce $X(\alpha, \sigma)$ via $\mathrm{diag}(X(\alpha, \sigma)^T X(\alpha, \sigma))$, for instance.

Implementation details

- k, ℓ are *hyperparameters*.

In practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.
- Use MLP to learn $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k} \times \ell}$.
- Integrate (S_α, ϕ_α), building the *integration matrix* $X(\alpha, \sigma) := \left[\int_{\phi_\alpha} \omega(\sigma) \right] \in \mathbb{R}^{|S_\alpha| \times \ell}$.
- Reduce $X(\alpha, \sigma)$ via $\mathrm{diag}(X(\alpha, \sigma)^T X(\alpha, \sigma))$, for instance.

Implementation details

- k, ℓ are *hyperparameters*.
- Approximate integrals using *Riemann sums*.

Why?

	Parameters	BACE	BBBP	HIV
EGNN	1M	74.62 ± 2.58	82.67 ± 0.54	68.25 ± 6.74
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
NkF (ours)	9K	**83.50 ± 0.55**	**86.41 ± 3.64**	**76.70 ± 2.17**

Why?

	Parameters	BACE	BBBP	HIV
EGNN	1M	74.62 ± 2.58	82.67 ± 0.54	68.25 ± 6.74
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
NkF (ours)	9K	**83.50 ± 0.55**	**86.41 ± 3.64**	**76.70 ± 2.17**

Properties

- We can prove a *universal approximation theorem* for neural k-forms.

Why?

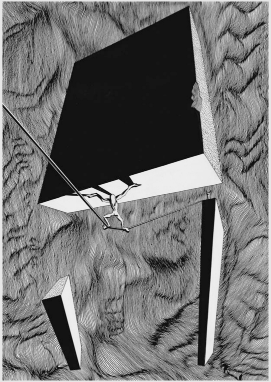
	Parameters	BACE	BBBP	HIV
EGNN	1M	74.62 ± 2.58	82.67 ± 0.54	68.25 ± 6.74
GAT	135K	69.52 ± 17.52	76.51 ± 3.36	56.38 ± 4.41
GCN	133K	66.79 ± 1.56	73.77 ± 3.30	68.70 ± 1.67
GIN	282K	42.91 ± 18.56	61.66 ± 19.47	55.28 ± 17.49
NkF (ours)	9K	**83.50 ± 0.55**	**86.41 ± 3.64**	**76.70 ± 2.17**

Properties

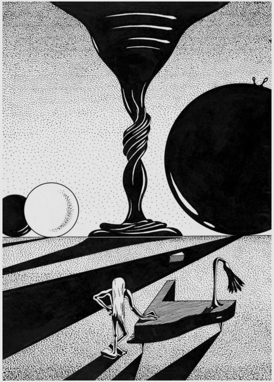
- We can prove a *universal approximation theorem* for neural k-forms.
- Our integration matrices can be easily integrated into *any* architecture.

Looking back…

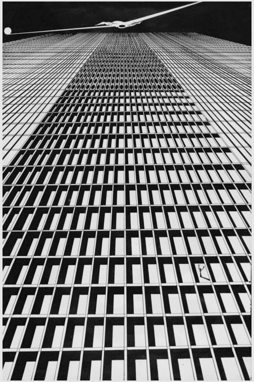
Challenges from a 2019 talk



Improving performance



Escaping flatland



First-class architectures

Images used with kind permission from Prof. A. T. Fomenko; these drawings are also found in the marvellous book *Homotopic Topology*.

Where are we now?

Performance

Persistent Homology Transform

How?

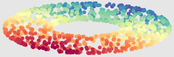
Calculate filtration of a shape $M \subset \mathbb{R}^d$ for a 'height' $r \in \mathbb{R}$ as $M(v, r) := \{x \in M \mid \langle x, v\rangle \leq r\}$, where $v \in \mathbb{S}^{d-1}$ and $\langle\cdot,\cdot\rangle$ denotes an inner product.

Performance

Persistent Homology Transform

How?

Calculate filtration of a shape $M \subset \mathbb{R}^d$ for a 'height' $r \in \mathbb{R}$ as $M(v,r) := \{x \in M \mid \langle x, v\rangle \leq r\}$, where $v \in \mathbb{S}^{d-1}$ and $\langle \cdot, \cdot \rangle$ denotes an inner product.

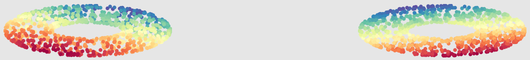


Performance

Persistent Homology Transform

How?

Calculate filtration of a shape $M \subset \mathbb{R}^d$ for a 'height' $r \in \mathbb{R}$ as $M(v, r) := \{x \in M \mid \langle x, v \rangle \leq r\}$, where $v \in \mathbb{S}^{d-1}$ and $\langle \cdot, \cdot \rangle$ denotes an inner product.

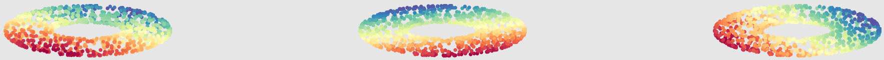


Performance

Persistent Homology Transform

How?

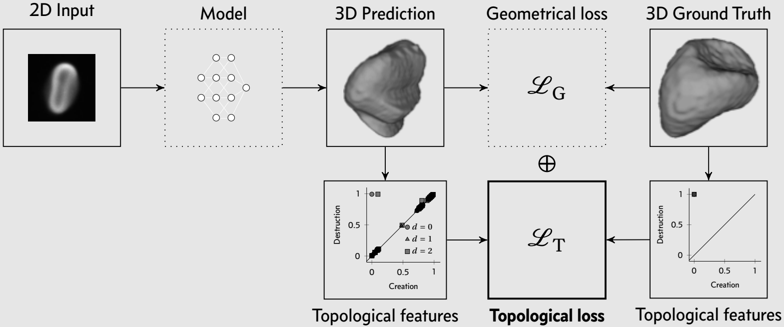
Calculate filtration of a shape $M \subset \mathbb{R}^d$ for a 'height' $r \in \mathbb{R}$ as $M(v, r) := \{x \in M \mid \langle x, v\rangle \leq r\}$, where $v \in \mathbb{S}^{d-1}$ and $\langle \cdot, \cdot \rangle$ denotes an inner product.



- K. Turner, S. Mukherjee and D. M. Boyer, *Persistent Homology Transform for Modeling Shapes and Surfaces*, *Information and Inference* 3.4, 2014, pp. 310–344
- E. Röell and **B. Rieck**, *Differentiable Euler Characteristic Transforms for Shape Classification*, International Conference on Learning Representations, 2024, arXiv: `2310.07630 [cs.LG]`

Escaping flatland (well, sort of...)

Using full 3D information to improve reconstruction tasks. Can we go higher?



D. J. E. Waibel, S. Atwell, M. Meier, C. Marr and **B. Rieck**, *Capturing Shape Information with Multi-Scale Topological Loss Terms for 3D Reconstruction*, Medical Image Computing and Computer Assisted Intervention (MICCAI), 2022, arXiv: `2203.01703 [cs.CV]`

First-class architectures

- C. Hofer, R. Kwitt, M. Niethammer and A. Uhl, *Deep learning with topological signatures*, Advances in Neural Information Processing Systems, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, vol. 30, Curran Associates, Inc., 2017, pp. 1633–1643
- M. Carrière, F. Chazal, Y. Ike, T. Lacombe, M. Royer and Y. Umeda, *PersLay: A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures*, Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, ed. by S. Chiappa and R. Calandra, vol. 108, Proceedings of Machine Learning Research, PMLR, 2020, pp. 2786–2796
- K. Kim, J. Kim, M. Zaheer, J. Kim, F. Chazal and L. Wasserman, *PLLay: Efficient Topological Layer based on Persistent Landscapes*, Advances in Neural Information Processing Systems, ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin, vol. 33, Curran Associates, Inc., 2020, pp. 15965–15977

The Next 20 Years

What we need

Our own data sets.

What we need

Our own data sets.

Harmonised frameworks and reporting.

What we need

Our own data sets.

Harmonised frameworks and reporting.

Users.

What to avoid

Round about the cauldron go;
In the persistent entrails throw.
Diagram that with many a pair
Makes the network look less bare.
Double, double toil and trouble;
GPU burn and cauldron bubble.

What to avoid

Round about the cauldron go;
In the persistent entrails throw.
Diagram that with many a pair
Makes the network look less bare.
Double, double toil and trouble;
GPU burn and cauldron bubble.

The use of topological features should be *justified* and assessed carefully.

One way forward

- Build bridges to ML topics (explainable ML, generative models, …).

One way forward

- Build bridges to ML topics (explainable ML, generative models, …).
- Dismantle obstacles to learning by investing in good explanations.

One way forward

- Build bridges to ML topics (explainable ML, generative models, …).
- Dismantle obstacles to learning by investing in good explanations.
- Focus on intuition and good visualisations.

One way forward

- Build bridges to ML topics (explainable ML, generative models, …).
- Dismantle obstacles to learning by investing in good explanations.
- Focus on intuition and good visualisations.
- **Geometry and topology are not opposed to each other.**

One way forward

- Build bridges to ML topics (explainable ML, generative models, …).
- Dismantle obstacles to learning by investing in good explanations.
- Focus on intuition and good visualisations.
- **Geometry and topology are not opposed to each other.**

P. Bubenik, M. Hull, D. Patel and B. Whittle, *Persistent homology detects curvature*, *Inverse Problems* 36.2, 2020, p. 025008

The end of the Möbius strip

Machine learning needs more mathematics.
Please join!

Our research