Topological Deep Learning

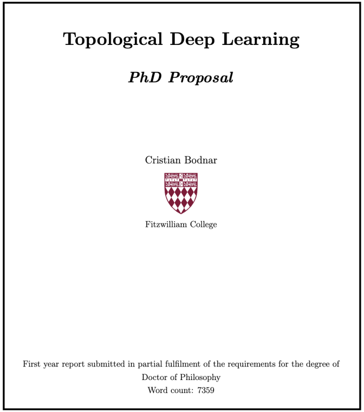
Part 1: Topological Message Passing

Cristian Bodnar
University of Cambridge

First Italian School in Geometric Deep Learning
Pescara, Italy
July 26, 2022

Introduction

Topological Deep Learning


Topological Deep Learning

PhD Proposal

Cristian Bodnar

Fitzwilliam College

First year report submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Word count: 7359

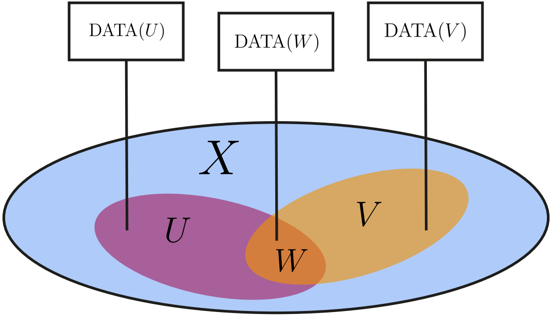
PhD Proposal (2020)

This talk is an attempt to describe the vision for a research programme on Topological Deep Learning.[1]

[1] This term and similar ones have been used informally and in printCarlsson, *Topological Deep Learning*, 2021; Hajij and Istvan, "Topological Deep Learning: Classification Neural Networks", 2021; Rieck, *Topological Representation Learning: A Differentiable Perspective*, 2022

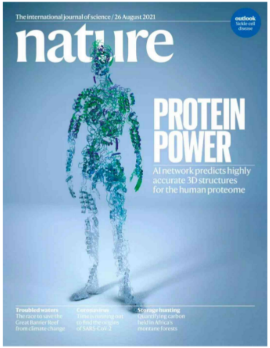
Topological Deep Learning

A research programme studying deep learning on data attached to topological spaces and topological aspects of machine learning models.



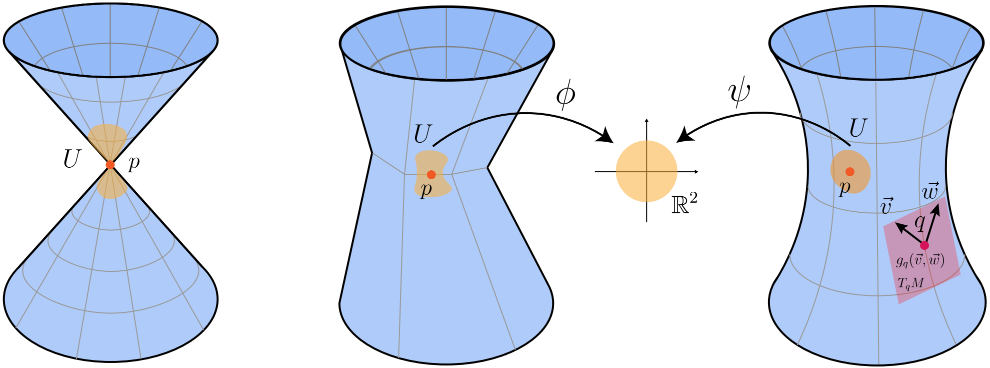
Geometric Deep Learning

Data often resides on structured domains: molecules, meshes, manifolds . . .



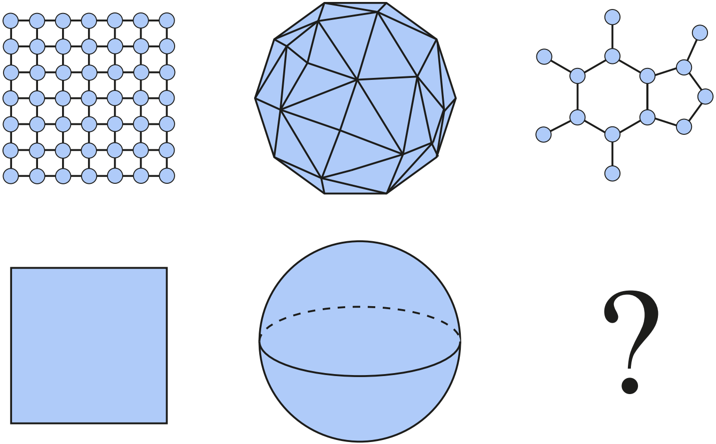
Is geometry all you need?

We must work with a chain of structural dependencies and not all spaces can be equipped with geometrical structure.



Geometry on graphs

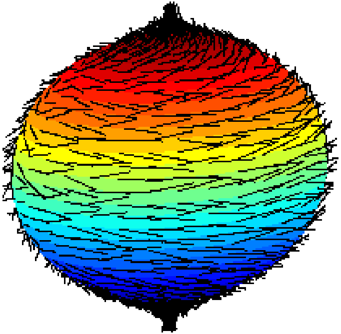
Graphs, the most prevalent space in GDL do not have a "natural" geometric structure[2].



[2] The diagram is inspired from Bronstein, *Graph Neural Networks through the lens of Differential Geometry and Algebraic Topology*, 2022

Topological Obstructions

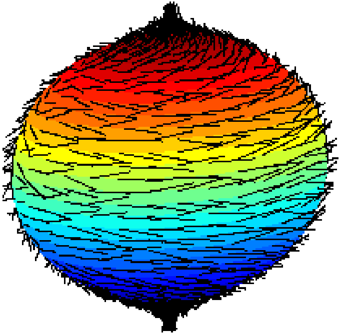
The structure of the topological layer affects the geometrical layer. Therefore, the topology of the space also affect the properties of the models working on it.



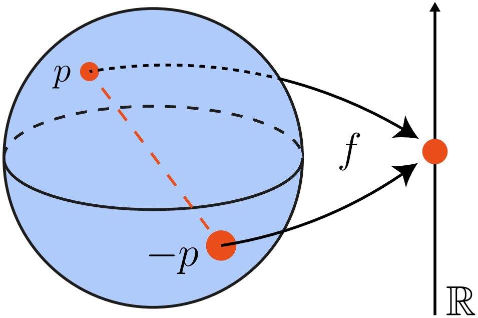
The Hairy Ball Theorem (Source: wikipedia).

Topological Obstructions

The structure of the topological layer affects the geometrical layer. Therefore, the topology of the space also affect the properties of the models working on it.



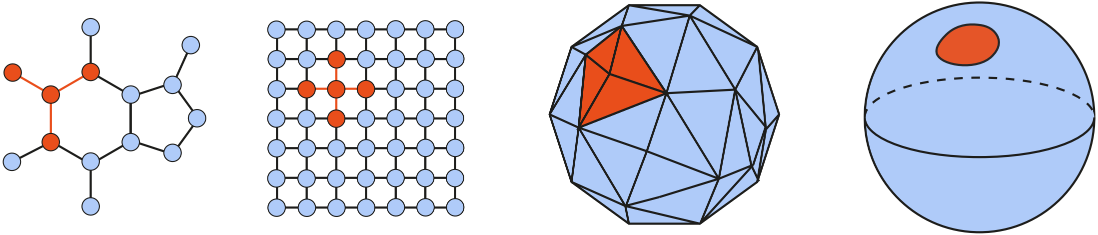
The Hairy Ball Theorem (Source: wikipedia).



The Borsuk–Ulam Theorem.

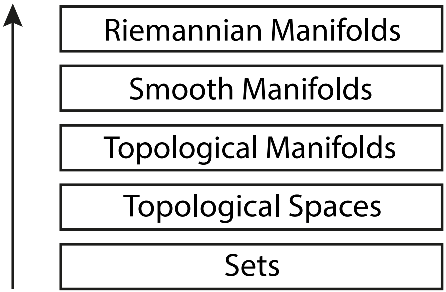
A structure to rule them all

By adopting this very general viewpoint, we can treat all the spaces of interest in a unified manner.



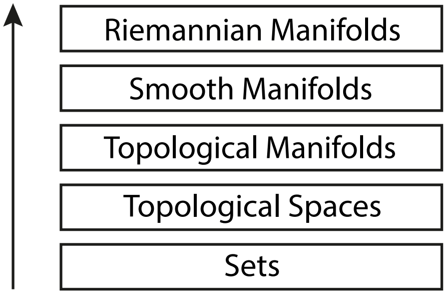
We can understand all types of spaces in terms of its neighbourhood or open set structure.

A bottom up approach



The topological perspective provides us with a bottom-up approach for learning from non-Euclidean data.

A bottom up approach

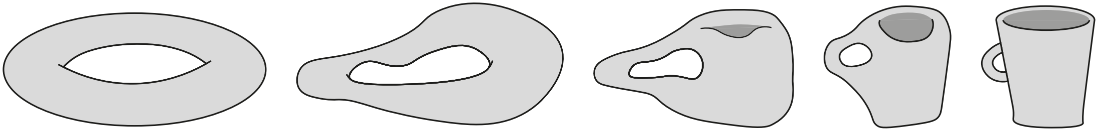


The topological perspective provides us with a bottom-up approach for learning from non-Euclidean data.

We will see that in this way we can recover (to some degree) these higher-level structures even on ill-behaved spaces like graphs.

Noise robustness

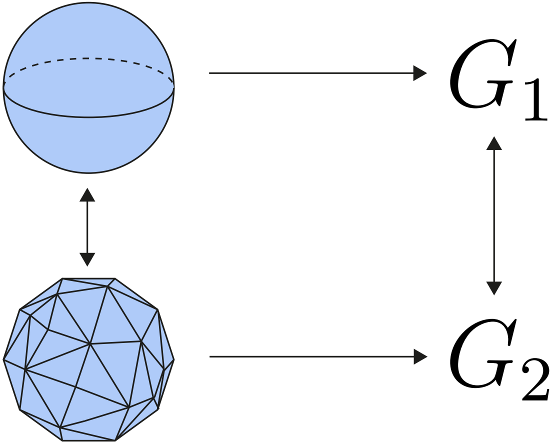
Topological properties are by construction invariant under smooth deformations and therefore, robust to noise. This observation led to Topological Quantum Computers.



To a topologist, a donut and a mug are the same.

A categorical foundation of data processing

Category theory was invented by Samuel Eilenberg and Saunders Mac Lane during their work on algebraic topology. It is a general theory of mathematical structures.

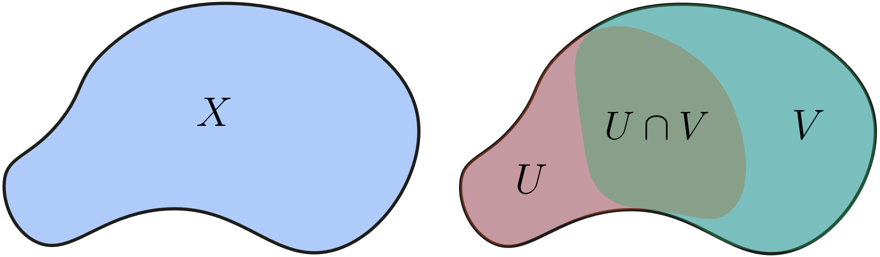


Category theory allows us to translate relations between spaces to relation between groups.

Topological Spaces

A *topological space* is a set X together with a collection $\mathcal{T}$ of subsets of X called the *open sets* of X and satisfying certain axioms:

1. The empty set and X belong to $\mathcal{T}$.
2. Any finite intersection and arbitrary union of open sets is an open set.



A topological space X and its open sets. These sets provide neighbourhood structure for the points of X.

A subfield of Geometric Deep Learning?

Henri Poincaré
(Source: wikipedia).

In "Analysis Situs" Poincaré discusses the group of homeomorphisms of a space and topology as the study of the invariants of the actions of this group. Thus he considered topology as a subfield of (Klein's) geometry.

A subfield of Geometric Deep Learning?

Henri Poincaré
(Source: wikipedia).

In “Analysis Situs” Poincaré discusses the group of homeomorphisms of a space and topology as the study of the invariants of the actions of this group. Thus he considered topology as a subfield of (Klein’s) geometry.

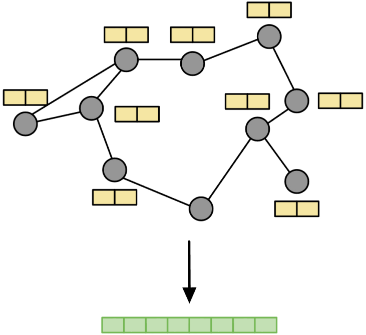
In this sense, Topological Deep Learning can be seen as a subfield of Geometric Deep Learning.

Topological Message Passing

Graph Machine Learning

Two typical tasks showing up in graph ML:

Graph-Level Tasks

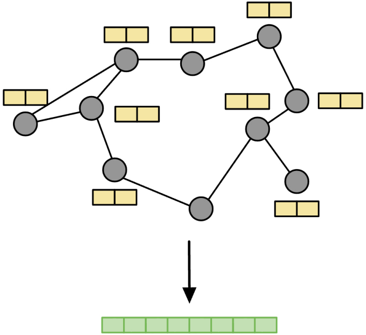


e.g. predicting solubility of molecules

Graph Machine Learning

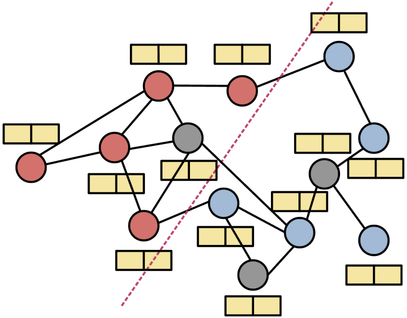
Two typical tasks showing up in graph ML:

Graph-Level Tasks



e.g. predicting solubility of molecules

Node-level Tasks

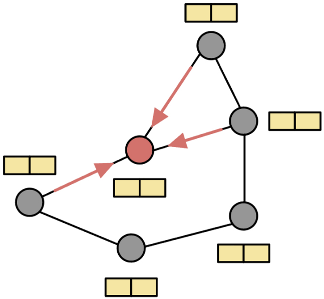


e.g. fraud detection

Message Passing Neural Networks

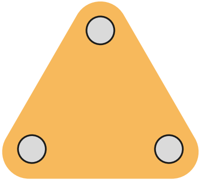
Most Graph Neural Networks (GNNs) can be understood as message passing:

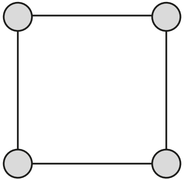
$$m_v^k := \mathrm{AGGREGATE}\big(\{h_u^{k-1} \mid u \in \mathcal{N}(v)\}\big) \qquad h_v^k := \mathrm{COMBINE}\big(h_v^{k-1}, m_v^k\big)$$

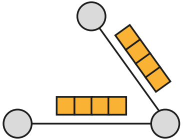


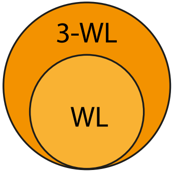
Limitations of Graph Neural Networks

Message Passing GNNs come with a series of limitations...

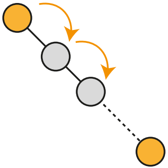

Higher-order interactions


Higher-order structures


Higher-order features



Expressive power

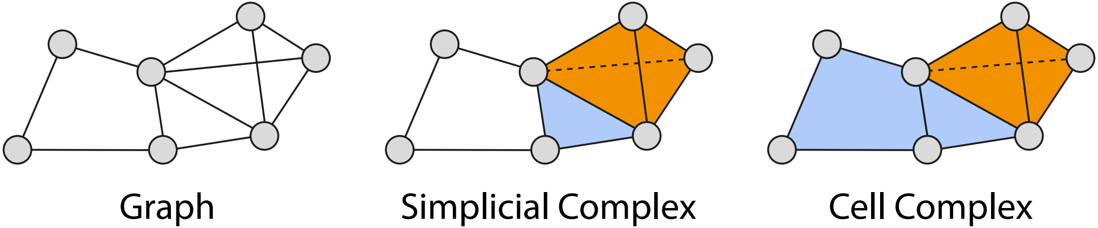

Long-range interactions

Higher-dimensional generalisations of graphs

We will see that these limitations are to some degree related to the underlying space.

Higher-dimensional generalisations of graphs

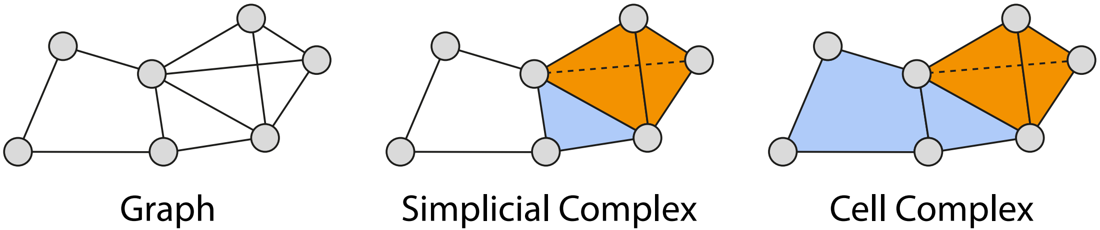
We will see that these limitations are to some degree related to the underlying space.



Graphs are part of a hierarchy of combinatorial topological spaces that are built from d-dimensional cells.

Higher-dimensional generalisations of graphs

We will see that these limitations are to some degree related to the underlying space.



Graphs are part of a hierarchy of combinatorial topological spaces that are built from d-dimensional cells.

Idea

How can we extend message passing to simplicial and cell complexes?

Message Passing Simplicial Networks

Simplicial Complexes

Let V be a non-empty vertex set. A *simplicial complex* K is a collection of nonempty subsets of V, called *simplices*, such that:

Simplicial Complexes

Let V be a non-empty vertex set. A *simplicial complex* K is a collection of nonempty subsets of V, called *simplices*, such that:

1. K contains all the singleton subsets of V.

Simplicial Complexes

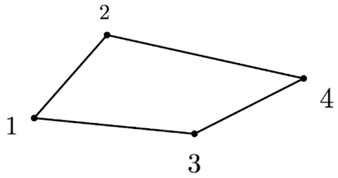
Let V be a non-empty vertex set. A *simplicial complex* K is a collection of nonempty subsets of V, called *simplices*, such that:

1. K contains all the singleton subsets of V.
2. K is closed under taking subsets.

Simplicial Complexes

Let V be a non-empty vertex set. A *simplicial complex* K is a collection of nonempty subsets of V, called *simplices*, such that:

1. K contains all the singleton subsets of V.
2. K is closed under taking subsets.

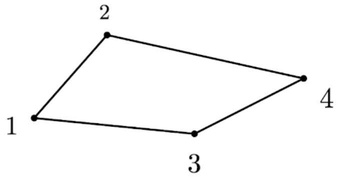


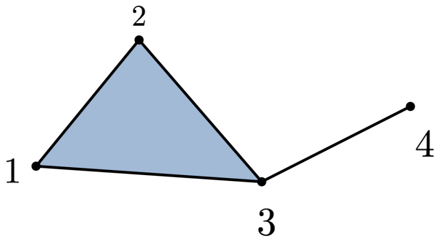
$\{\{1\},\{2\},\{3\},\{4\},\{1,2\},\{1,3\},\{2,4\},\{3,4\}\}$

Simplicial Complexes

Let V be a non-empty vertex set. A *simplicial complex* K is a collection of nonempty subsets of V, called *simplices*, such that:

1. K contains all the singleton subsets of V.
2. K is closed under taking subsets.

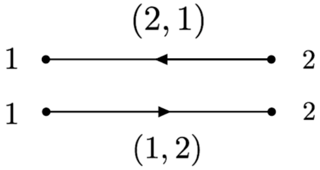




$\{\{1\},\{2\},\{3\},\{4\},\{1,2\},\{1,3\},\{2,4\},\{3,4\}\}$ $\{\{1\},\{2\},\{3\},\{4\},\{1,2\},\{1,3\},\{2,3\},\{3,4\},\{1,2,3\}\}$

Orientations

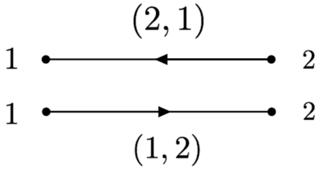
An *oriented* simplex is a simplex with a specified order of its vertices.
These can be visualised as a walk on the simplex in the order specified by the vertices.



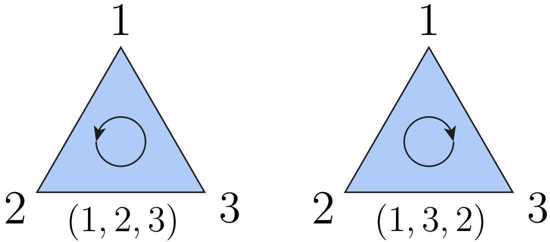
The orientations of the 1-simplex $\{1, 2\}$

Orientations

An *oriented* simplex is a simplex with a specified order of its vertices.
These can be visualised as a walk on the simplex in the order specified by the vertices.



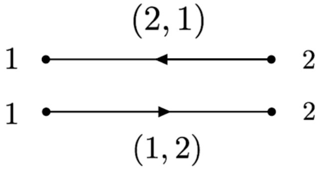
The orientations of the 1-simplex $\{1, 2\}$



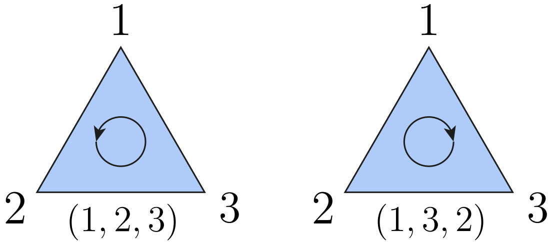
Two orientations of the 2-simplex $\{1, 2, 3\}$

Orientations

An *oriented* simplex is a simplex with a specified order of its vertices.
These can be visualised as a walk on the simplex in the order specified by the vertices.



The orientations of the 1-simplex $\{1, 2\}$

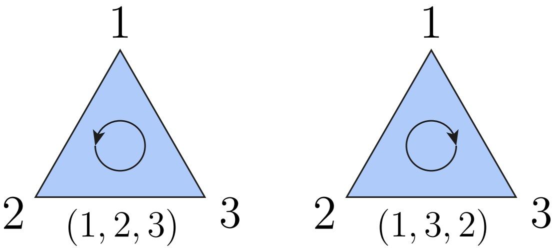


Two orientations of the 2-simplex $\{1, 2, 3\}$

We represent oriented simplices as tuples $(\cdot)$ and unoriented ones as sets $\{\cdot\}$.

Orientations

Ignoring the starting point, each k-simplex with $k > 0$ has two distinct orientations.

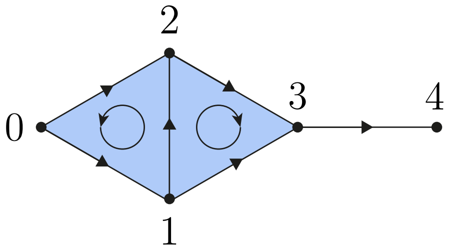


We can choose a representative for each of these two equivalence classes:

$$123 := \{(1,2,3),(2,3,1),(3,1,2)\} \leftarrow \text{Even permutations}$$
$$132 := \{(1,3,2),(2,1,3),(3,2,1)\} \leftarrow \text{Odd permutations}$$

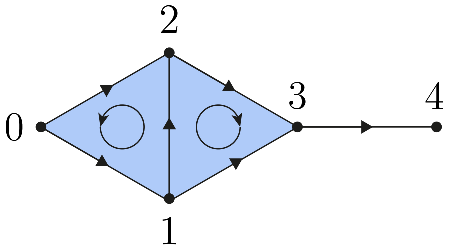
Oriented Simplicial Complexes

An *oriented simplicial complex* is a simplicial complex with a choice of orientation for each of its simplices.



Oriented Simplicial Complexes

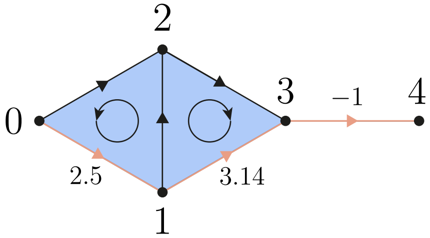
An *oriented simplicial complex* is a simplicial complex with a choice of orientation for each of its simplices.



An easy way to choose an orientation for a complex is to choose a global order for the vertices $[v_0, \ldots, v_n]$ and then use this order for the vertices of any simplex σ.

Chains

The vector space of *k-chains* $C_k(K, \mathbb{R})$ is the vector space with real coefficients having as a basis the oriented k-simplices of K.

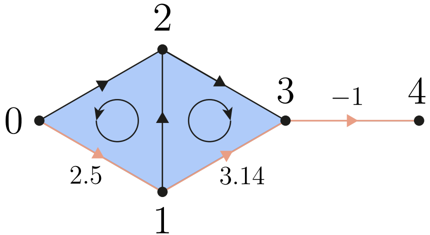


Consider the 1-chain $c_1 \in C_1(K, \mathbb{R})$.

$$
\begin{aligned}
c_1 &= 2.5(0,1) - (3,4) + 3.14(1,3) \\
&= 2.5(0,1) + (4,3) + 3.14(1,3)
\end{aligned}
$$

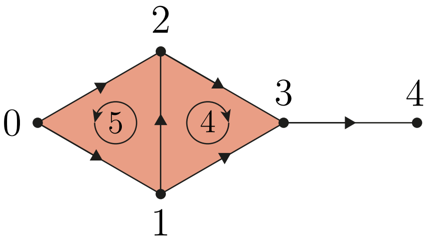
Chains

The vector space of *k-chains* $C_k(K, \mathbb{R})$ is the vector space with real coefficients having as a basis the oriented k-simplices of K.



Consider the 1-chain $c_1 \in C_1(K, \mathbb{R})$.

$$\begin{aligned} c_1 &= 2.5(0,1) - (3,4) + 3.14(1,3) \\ &= 2.5(0,1) + (4,3) + 3.14(1,3) \end{aligned}$$



Consider the 2-chain $c_2 \in C_2(K, \mathbb{R})$.

$$c_2 = 5.0(0,1,2) + 4.0(1,2,3)$$

Boundary operator

Denote by $\sigma_{-i} := (v_0, \ldots, \hat{v}_i, \ldots, v_k)$ the simplex obtained by dropping the vertex v_i.

Boundary operator

Denote by $\sigma_{-i} := (v_0, \ldots, \hat{v}_i, \ldots, v_k)$ the simplex obtained by dropping the vertex v_i.

The *boundary operator* $\partial_k : C_k(K, \mathbb{R}) \to C_{k-1}(K, \mathbb{R})$ is the linear operator:

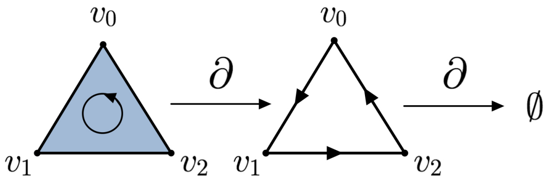
$$\partial_k(v_0, \ldots, v_k) = \sum_{i=0}^{k} (-1)^i (v_0, \ldots, \hat{v}_i, \ldots, v_k)$$

Boundary operator

Denote by $\sigma_{-i} := (v_0, \ldots, \hat{v}_i, \ldots, v_k)$ the simplex obtained by dropping the vertex v_i.

The *boundary operator* $\partial_k : C_k(K, \mathbb{R}) \to C_{k-1}(K, \mathbb{R})$ is the linear operator:

$$\partial_k(v_0, \ldots, v_k) = \sum_{i=0}^{k} (-1)^i (v_0, \ldots, \hat{v}_i, \ldots, v_k)$$

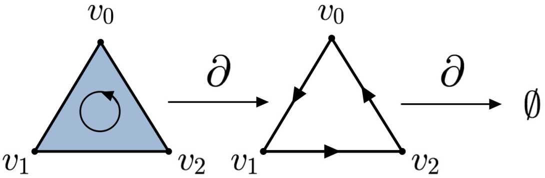


Boundary operator

Denote by $\sigma_{-i} := (v_0, \ldots, \hat{v}_i, \ldots, v_k)$ the simplex obtained by dropping the vertex v_i.

The *boundary operator* $\partial_k : C_k(K, \mathbb{R}) \to C_{k-1}(K, \mathbb{R})$ is the linear operator:

$$\partial_k(v_0, \ldots, v_k) = \sum_{i=0}^{k} (-1)^i (v_0, \ldots, \hat{v}_i, \ldots, v_k)$$



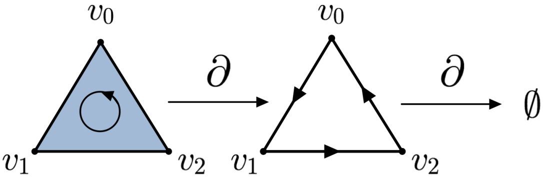
$$\partial(v_0, v_1, v_2) = (v_1, v_2) - (v_0, v_2) + (v_0, v_1) = (v_1, v_2) + (v_2, v_0) + (v_0, v_1) \qquad (1)$$

Boundary operator

Denote by $\sigma_{-i} := (v_0, \dots, \hat{v}_i, \dots, v_k)$ the simplex obtained by dropping the vertex v_i.

The *boundary operator* $\partial_k : C_k(K, \mathbb{R}) \to C_{k-1}(K, \mathbb{R})$ is the linear operator:

$$\partial_k(v_0, \dots, v_k) = \sum_{i=0}^{k} (-1)^i (v_0, \dots, \hat{v}_i, \dots, v_k)$$



$$\partial(v_0, v_1, v_2) = (v_1, v_2) - (v_0, v_2) + (v_0, v_1) = (v_1, v_2) + (v_2, v_0) + (v_0, v_1) \tag{1}$$

$$\partial\partial(v_0, v_1, v_2) = \partial(v_1, v_2) + \partial(v_2, v_0) + \partial(v_0, v_1) \tag{2}$$

$$= (v_2 - v_1) + (v_0 - v_2) + (v_1 - v_0) = 0 \tag{3}$$

Boundary of a Boundary

Is the last result a coincidence? Does it actually work for more complicated cases?

Boundary of a Boundary

Is the last result a coincidence? Does it actually work for more complicated cases?

Proposition

The boundary of a boundary is zero: $\partial_{k-1} \circ \partial_k = 0 \Leftrightarrow \mathrm{im}\ \partial_k \subseteq \ker\ \partial_{k-1}$

Boundary of a Boundary

Is the last result a coincidence? Does it actually work for more complicated cases?

Proposition

The boundary of a boundary is zero: $\partial_{k-1} \circ \partial_k = 0 \Leftrightarrow \mathrm{im}\ \partial_k \subseteq \ker \partial_{k-1}$

Proof.

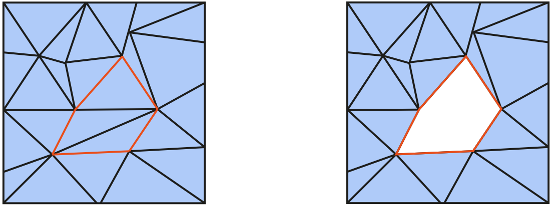
$$(\partial_{k-1} \circ \partial_k)(v_0, \ldots, v_k) = \partial_{k-1} \sum_{i=0}^{k} (-1)^i (v_0, \ldots, \hat{v}_i, \ldots, v_k) \tag{4}$$

$$= \sum_{j<i} (-1)^j (-1)^i \sigma_{-i,-j} + \sum_{j>i} (-1)^{j-1} (-1)^i \sigma_{-i,-j} = 0 \tag{5}$$

□

What is a topological hole?

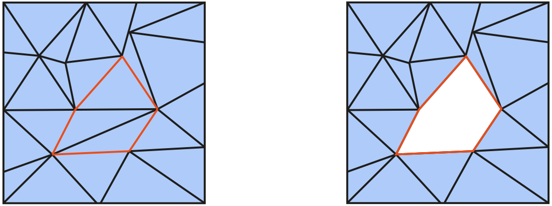
What is the difference between these two 1-chains? One is a hole and the other is not.



Two 1-cochains, c_1 and c_2. The second is a hole, while the first is not.

What is a topological hole?

What is the difference between these two 1-chains? One is a hole and the other is not.

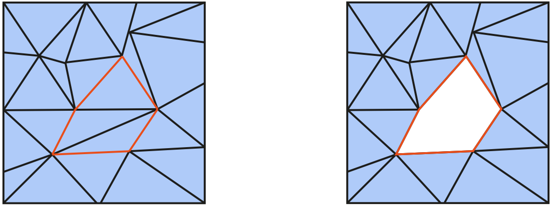


Two 1-cochains, c_1 and c_2. The second is a hole, while the first is not.

First, $\partial_1 c_1 = \partial_1 c_2 = 0$ (they are cycles).

What is a topological hole?

What is the difference between these two 1-chains? One is a hole and the other is not.

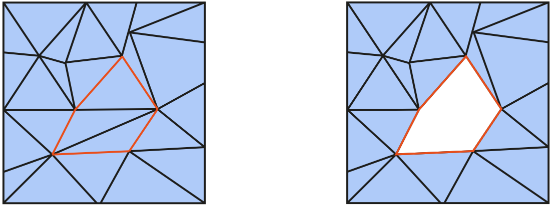


Two 1-cochains, c_1 and c_2. The second is a hole, while the first is not.

First, $\partial_1 c_1 = \partial_1 c_2 = 0$ (they are cycles). But $c_1 \in \mathrm{im}\ \partial_2$ (it is the boundary of a 2-chain).

What is a topological hole?

What is the difference between these two 1-chains? One is a hole and the other is not.



Two 1-cochains, c_1 and c_2. The second is a hole, while the first is not.

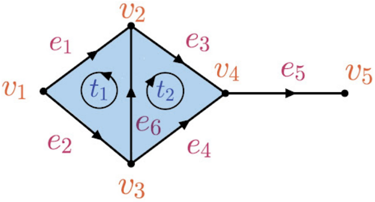
First, $\partial_1 c_1 = \partial_1 c_2 = 0$ (they are cycles). But $c_1 \in \mathrm{im}\ \partial_2$ (it is the boundary of a 2-chain).

This leads to a definition of k-dimensional holes, given by the *k-th Homology group*

$$H_k(K) := \ker\ \partial_k / \mathrm{im}\ \partial_{k+1}$$

Boundary matrices

We can represent the boundary operator for each dimension using a matrix.



$$B_1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ +1 & 0 & -1 & 0 & 0 & +1 \\ 0 & +1 & 0 & -1 & 0 & -1 \\ 0 & 0 & +1 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 \end{bmatrix} \end{array}$$

$$B_2 = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{array} \begin{array}{c} \begin{matrix} t_1 & t_2 \end{matrix} \\ \begin{bmatrix} -1 & 0 \\ +1 & 0 \\ 0 & +1 \\ 0 & -1 \\ 0 & 0 \\ +1 & +1 \end{bmatrix} \end{array}$$

Hodge Laplacian

The *k-th Hodge Laplacian*[3] $\boldsymbol{L}_k : C_k(K, \mathbb{R}) \to C_k(K, \mathbb{R})$ is given by:

$$\boldsymbol{L}_k = \boldsymbol{L}_k^{\downarrow} + \boldsymbol{L}_k^{\uparrow} = \boldsymbol{B}_k^{\top} \boldsymbol{B}_k + \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^{\top}$$

[3] Horak and Jost, "Spectra of combinatorial Laplace operators on simplicial complexes", 2013; Muhammad and Egerstedt, "Control using higher order Laplacians in network topologies", 2006; Lim, "Hodge Laplacians on graphs", 2020; Barbarossa and Sardellitti, "Topological signal processing over simplicial complexes", 2020; Schaub et al., "Random walks on simplicial complexes and the normalized Hodge 1-Laplacian", 2020.

Hodge Laplacian

The *k-th Hodge Laplacian*[3] $\boldsymbol{L}_k : C_k(K, \mathbb{R}) \to C_k(K, \mathbb{R})$ is given by:

$$\boldsymbol{L}_k = \boldsymbol{L}_k^{\downarrow} + \boldsymbol{L}_k^{\uparrow} = \boldsymbol{B}_k^{\top} \boldsymbol{B}_k + \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^{\top}$$

Denote by $\sigma_i \vee \sigma_j$ if σ_i, σ_j share a $(k-1)$-simplex and $\sigma_i \wedge \sigma_j$ if they are on the boundary of the same $(k+1)$-simplex.

[3] Horak and Jost, "Spectra of combinatorial Laplace operators on simplicial complexes", 2013; Muhammad and Egerstedt, "Control using higher order Laplacians in network topologies", 2006; Lim, "Hodge Laplacians on graphs", 2020; Barbarossa and Sardellitti, "Topological signal processing over simplicial complexes", 2020; Schaub et al., "Random walks on simplicial complexes and the normalized Hodge 1-Laplacian", 2020.

Hodge Laplacian

The *k-th Hodge Laplacian*[3] $\boldsymbol{L}_k : C_k(K, \mathbb{R}) \to C_k(K, \mathbb{R})$ is given by:

$$\boldsymbol{L}_k = \boldsymbol{L}_k^{\downarrow} + \boldsymbol{L}_k^{\uparrow} = \boldsymbol{B}_k^{\top} \boldsymbol{B}_k + \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^{\top}$$

Denote by $\sigma_i \vee \sigma_j$ if σ_i, σ_j share a $(k-1)$-simplex and $\sigma_i \wedge \sigma_j$ if they are on the boundary of the same $(k+1)$-simplex.

$$\boldsymbol{L}_k^{\downarrow}(i,j) = \begin{cases} k+1 & \text{if } i = j \\ \pm 1 & \text{if } i \neq j \text{ and } \sigma_i \vee \sigma_j \\ 0 & \text{otherwise} \end{cases}$$

[3] Horak and Jost, "Spectra of combinatorial Laplace operators on simplicial complexes", 2013; Muhammad and Egerstedt, "Control using higher order Laplacians in network topologies", 2006; Lim, "Hodge Laplacians on graphs", 2020; Barbarossa and Sardellitti, "Topological signal processing over simplicial complexes", 2020; Schaub et al., "Random walks on simplicial complexes and the normalized Hodge 1-Laplacian", 2020.

Hodge Laplacian

The *k-th Hodge Laplacian*[3] $\boldsymbol{L}_k : C_k(K, \mathbb{R}) \to C_k(K, \mathbb{R})$ is given by:

$$\boldsymbol{L}_k = \boldsymbol{L}_k^{\downarrow} + \boldsymbol{L}_k^{\uparrow} = \boldsymbol{B}_k^{\top} \boldsymbol{B}_k + \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^{\top}$$

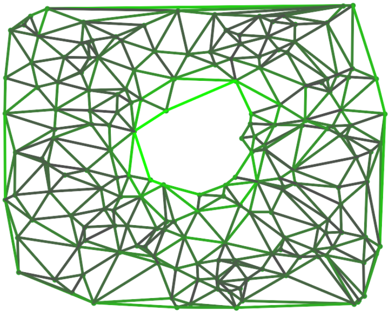
Denote by $\sigma_i \vee \sigma_j$ if σ_i, σ_j share a $(k-1)$-simplex and $\sigma_i \wedge \sigma_j$ if they are on the boundary of the same $(k+1)$-simplex.

$$\boldsymbol{L}_k^{\downarrow}(i,j) = \begin{cases} k+1 & \text{if } i = j \\ \pm 1 & \text{if } i \neq j \text{ and } \sigma_i \vee \sigma_j \\ 0 & \text{otherwise} \end{cases} \quad \boldsymbol{L}_k^{\uparrow}(i,j) = \begin{cases} \deg^{\uparrow}(\sigma_i) & \text{if } i = j \\ \pm 1 & \text{if } i \neq j \text{ and } \sigma_i \wedge \sigma_j \\ 0 & \text{otherwise} \end{cases}$$

Importantly, $\boldsymbol{L}_0 = \boldsymbol{B}_1 \boldsymbol{B}_1^{\top} = \boldsymbol{D} - \boldsymbol{A}$ is the usual graph Laplacian.

[3] Horak and Jost, "Spectra of combinatorial Laplace operators on simplicial complexes", 2013; Muhammad and Egerstedt, "Control using higher order Laplacians in network topologies", 2006; Lim, "Hodge Laplacians on graphs", 2020; Barbarossa and Sardellitti, "Topological signal processing over simplicial complexes", 2020; Schaub et al., "Random walks on simplicial complexes and the normalized Hodge 1-Laplacian", 2020.

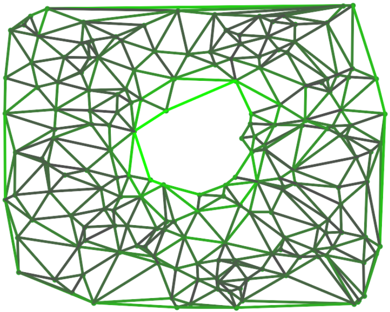
Hodge Theorem



Harmonic eigenvector of $\boldsymbol{L}_1$
Credits to Andrei C. Popescu

Plotting the harmonic eigenvector of $\boldsymbol{L}_1$ we notice that its energy is concentrated around the hole of the complex. What is going on?

Hodge Theorem



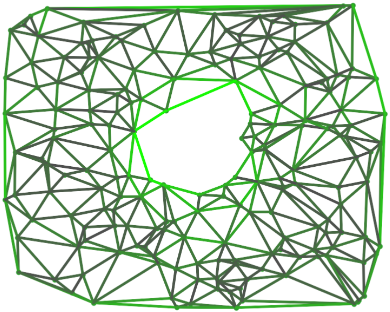
Harmonic eigenvector of $\boldsymbol{L}_1$
Credits to Andrei C. Popescu

Plotting the harmonic eigenvector of $\boldsymbol{L}_1$ we notice that its energy is concentrated around the hole of the complex. What is going on?

Theorem (Hodge Theorem)

$\ker \boldsymbol{L}_k$ *and* $H_k(K)$ *are isomorphic vector spaces.*

Hodge Theorem



Harmonic eigenvector of $\boldsymbol{L}_1$
Credits to Andrei C. Popescu

Plotting the harmonic eigenvector of $\boldsymbol{L}_1$ we notice that its energy is concentrated around the hole of the complex. What is going on?

Theorem (Hodge Theorem)

$\ker \boldsymbol{L}_k$ *and* $H_k(K)$ *are isomorphic vector spaces.*

Moreover, $\beta_k = \dim(\ker \boldsymbol{L}_k) = \dim(H_k)$ gives us the k-th *Betti number*, counting the number of k-dim holes in the complex.

Simplicial Convolutional Networks

Let K be a simplicial complex with normalised Hodge Laplacian $\Delta_k = \alpha \boldsymbol{L}_k$ with $\alpha > 0$, and k-simplex features $\boldsymbol{X} \in \mathbb{R}^{n_k \times d}$.

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.
[5] Cai and Yusu Wang, "A note on over-smoothing for graph neural networks", 2020.

Simplicial Convolutional Networks

Let K be a simplicial complex with normalised Hodge Laplacian $\Delta_k = \alpha \boldsymbol{L}_k$ with $\alpha > 0$, and k-simplex features $\boldsymbol{X} \in \mathbb{R}^{n_k \times d}$.

We can build a simplicial equivalent of GCN[4]: $\boldsymbol{Y} := \sigma\Big((\boldsymbol{I} - \Delta_0)\boldsymbol{X}\boldsymbol{W}\Big)$.

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.

[5] Cai and Yusu Wang, "A note on over-smoothing for graph neural networks", 2020.

Simplicial Convolutional Networks

Let K be a simplicial complex with normalised Hodge Laplacian $\Delta_k = \alpha \boldsymbol{L}_k$ with $\alpha > 0$, and k-simplex features $\boldsymbol{X} \in \mathbb{R}^{n_k \times d}$.

We can build a simplicial equivalent of GCN[4]: $\boldsymbol{Y} := \sigma\Big((\boldsymbol{I} - \Delta_0)\boldsymbol{X}\boldsymbol{W}\Big)$.

Define $\lambda_* = \max_{\lambda_i \neq 0}(1 - \lambda_i)^2$, where λ_i denotes the eigenvalues of Δ_k and assume $\sigma = \mathrm{id}$. Additionally, define the *Dirichlet energy* E of a signal $\boldsymbol{X}$ as $\mathrm{trace}(\boldsymbol{X}^\top \Delta_k \boldsymbol{X})$.

Applying the proof technique of Cai and Wang[5], we have:

Theorem

$E(\boldsymbol{Y}) \leq \lambda_* \|\boldsymbol{W}^\top\|_2^2 E(\boldsymbol{X})$ *and if* α *is sufficiently low, the model converges exponentially fast to* $\ker \Delta_k$.

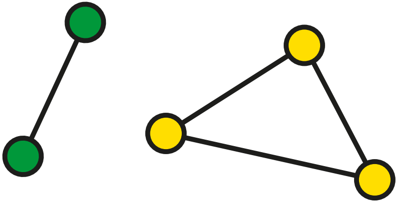
[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.
[5] Cai and Yusu Wang, "A note on over-smoothing for graph neural networks", 2020.

Implications

Main idea

The asymptotic behaviour of Deep Linear GCNs and its simplicial version is topological.

(Linear) GCN



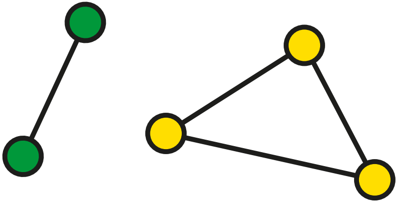
The features converge to a signal depending only on the connected components of the graph and their degrees.

Implications

Main idea

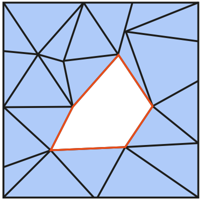
The asymptotic behaviour of Deep Linear GCNs and its simplicial version is topological.

(Linear) GCN



The features converge to a signal depending only on the connected components of the graph and their degrees.

Linear Simplicial CN



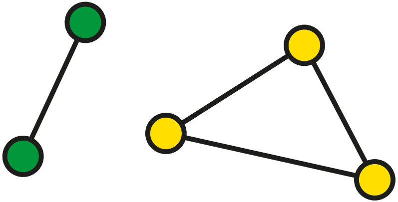
The features converge to a signal whose energy is concentrated around the holes of the complex.

Implications

Main idea

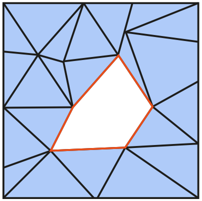
The asymptotic behaviour of Deep Linear GCNs and its simplicial version is topological.

(Linear) GCN



The features converge to a signal depending only on the connected components of the graph and their degrees.

Linear Simplicial CN



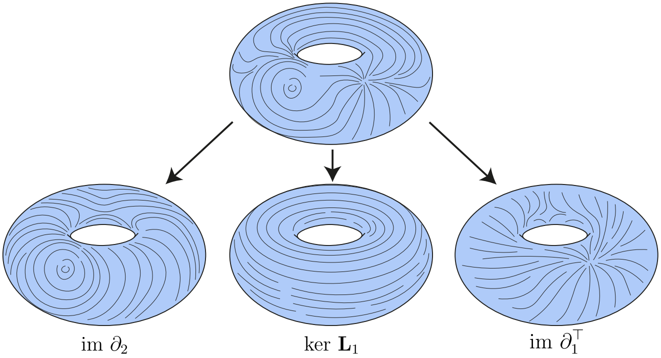
The features converge to a signal whose energy is concentrated around the holes of the complex.

Finally, remark that because β_k is a topological invariant, the dimension of the subspace the model converges to remains invariant under homeomorphisms.

The Hodge Decomposition

Theorem (Hodge Decomposition)

$$C_k(K,\mathbb{R}) = \text{im } \partial_k^\top \bigoplus \text{ker } \boldsymbol{L}_k \bigoplus \text{im } \partial_{k+1}$$



Chain decomposed into rotational, harmonic, and gradient parts. Diagram inspired from K. Crane[6]

[6] Crane et al., "Digital Geometry Processing with Discrete Exterior Calculus", 2013.

The Zoo of Simplicial Convolutional Networks

Equipped with a Laplacian, one can define all sorts of simplicial convolutions[7]. All of these can be seen as a form of message passing[8].

[7] Ebli et al., "Simplicial Neural Networks", 2020; Bunch et al., "Simplicial 2-Complex Convolutional Neural Networks", 2020; Glaze et al., "Principled Simplicial Neural Networks for Trajectory Prediction", 2021; Keros et al., "Dist2cycle: A simplicial neural network for homology localization", 2022; Goh et al., "Simplicial Attention Networks", 2022; Giusti et al., "Simplicial Attention Networks", 2022.

[8] Bodnar, Frasca, Yuguang Wang, et al., "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks", 2021.

The Zoo of Simplicial Convolutional Networks

Equipped with a Laplacian, one can define all sorts of simplicial convolutions[7]. All of these can be seen as a form of message passing[8].

Let $\boldsymbol{X}_0 \in \mathbb{R}^{V\times F}$, $\boldsymbol{X}_1 \in \mathbb{R}^{E\times F}$, $\boldsymbol{X}_2 \in \mathbb{R}^{T\times F}$ be matrices of $0, 1, 2$-chains respectively, $\boldsymbol{W}_i$ a set of weights. Then the convolutions above typically look like below, where $\boldsymbol{L}_1^{\downarrow}, \boldsymbol{L}_1^{\uparrow}, \boldsymbol{B}_1^{\top}, \boldsymbol{B}_2$ can be replaced by any matrix with the same sparsity pattern (e.g. attention matrix).

$$\boldsymbol{Y} = \psi\Big(\underbrace{\boldsymbol{L}_1^{\downarrow}\boldsymbol{X}_1\boldsymbol{W}_1}_{\text{Lower adj.}} + \underbrace{\boldsymbol{L}_1^{\uparrow}\boldsymbol{X}_1\boldsymbol{W}_2}_{\text{Upper adj.}} + \underbrace{\boldsymbol{B}_1^{\top}\boldsymbol{X}_0\boldsymbol{W}_3}_{\text{Boundary adj.}} + \underbrace{\boldsymbol{B}_2\boldsymbol{X}_2\boldsymbol{W}_4}_{\text{Coboundary adj.}} + \boldsymbol{X}_1\boldsymbol{W}_5 \Big)$$

[7] Ebli et al., "Simplicial Neural Networks", 2020; Bunch et al., "Simplicial 2-Complex Convolutional Neural Networks", 2020; Glaze et al., "Principled Simplicial Neural Networks for Trajectory Prediction", 2021; Keros et al., "Dist2cycle: A simplicial neural network for homology localization", 2022; Goh et al., "Simplicial Attention Networks", 2022; Giusti et al., "Simplicial Attention Networks", 2022.

[8] Bodnar, Frasca, Yuguang Wang, et al., "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks", 2021.

The Zoo of Simplicial Convolutional Networks

Equipped with a Laplacian, one can define all sorts of simplicial convolutions[7]. All of these can be seen as a form of message passing[8].

Let $\boldsymbol{X}_0 \in \mathbb{R}^{V\times F}$, $\boldsymbol{X}_1 \in \mathbb{R}^{E\times F}$, $\boldsymbol{X}_2 \in \mathbb{R}^{T\times F}$ be matrices of $0, 1, 2$-chains respectively, $\boldsymbol{W}_i$ a set of weights. Then the convolutions above typically look like below, where $\boldsymbol{L}_1^{\downarrow}, \boldsymbol{L}_1^{\uparrow}, \boldsymbol{B}_1^{\top}, \boldsymbol{B}_2$ can be replaced by any matrix with the same sparsity pattern (e.g. attention matrix).

$$\boldsymbol{Y} = \psi\Big(\underbrace{\boldsymbol{L}_1^{\downarrow}\boldsymbol{X}_1\boldsymbol{W}_1}_{\text{Lower adj.}} + \underbrace{\boldsymbol{L}_1^{\uparrow}\boldsymbol{X}_1\boldsymbol{W}_2}_{\text{Upper adj.}} + \underbrace{\boldsymbol{B}_1^{\top}\boldsymbol{X}_0\boldsymbol{W}_3}_{\text{Boundary adj.}} + \underbrace{\boldsymbol{B}_2\boldsymbol{X}_2\boldsymbol{W}_4}_{\text{Coboundary adj.}} + \boldsymbol{X}_1\boldsymbol{W}_5\Big)$$

This factorisation can be interpreted via the Hodge decomposition.

[7] Ebli et al., "Simplicial Neural Networks", 2020; Bunch et al., "Simplicial 2-Complex Convolutional Neural Networks", 2020; Glaze et al., "Principled Simplicial Neural Networks for Trajectory Prediction", 2021; Keros et al., "Dist2cycle: A simplicial neural network for homology localization", 2022; Goh et al., "Simplicial Attention Networks", 2022; Giusti et al., "Simplicial Attention Networks", 2022.

[8] Bodnar, Frasca, Yuguang Wang, et al., "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks", 2021.

Symmetries: Permutation Equivariance

A simplicial complex of dimention d can be specified by all its boundary matrices $\mathcal{B} = (\boldsymbol{B}_1, \ldots, \boldsymbol{B}_d)$. Similarly define a tuple of permutation matrices $\mathcal{P} = (\boldsymbol{P}_0, \ldots, \boldsymbol{P}_d)$ and denote by $\mathcal{P}\mathcal{B} = (\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top, \ldots, \boldsymbol{P}_{d-1} \boldsymbol{B}_d \boldsymbol{P}_d^\top)$.

Symmetries: Permutation Equivariance

A simplicial complex of dimention d can be specified by all its boundary matrices $\mathcal{B} = (\boldsymbol{B}_1, \ldots, \boldsymbol{B}_d)$. Similarly define a tuple of permutation matrices $\mathcal{P} = (\boldsymbol{P}_0, \ldots, \boldsymbol{P}_d)$ and denote by $\mathcal{P}\mathcal{B} = (\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top, \ldots, \boldsymbol{P}_{d-1} \boldsymbol{B}_d \boldsymbol{P}_d^\top)$.

$f : C_k(K, \mathbb{R})^{F_1} \to C_k(K, \mathbb{R})^{F_2}$ is *permutation equivariant* if $f(\mathcal{P}\mathcal{B}, \boldsymbol{P}_k \boldsymbol{X}) = \boldsymbol{P}_k f(\mathcal{B}, \boldsymbol{X})$

Proposition

The function $f(\mathcal{B}, \mathcal{X}) := \psi\left(\boldsymbol{L}_1^{\downarrow} \boldsymbol{X}_1 \boldsymbol{W}_1 + \boldsymbol{L}_1^{\uparrow} \boldsymbol{X}_1 \boldsymbol{W}_2\right)$ *is permutation equivariant.*

Proof sketch.

Considering only lower adjacencies:

$(\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top)^\top (\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top)(\boldsymbol{P}_1 \boldsymbol{X}_1) \boldsymbol{W}_1 = \boldsymbol{P}_1 \boldsymbol{B}_1^\top \boldsymbol{B}_1 \boldsymbol{X}_1 \boldsymbol{W}$

Symmetries: Permutation Equivariance

A simplicial complex of dimention d can be specified by all its boundary matrices $\mathcal{B} = (\boldsymbol{B}_1, \ldots, \boldsymbol{B}_d)$. Similarly define a tuple of permutation matrices $\mathcal{P} = (\boldsymbol{P}_0, \ldots, \boldsymbol{P}_d)$ and denote by $\mathcal{P}\mathcal{B} = (\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top, \ldots, \boldsymbol{P}_{d-1} \boldsymbol{B}_d \boldsymbol{P}_d^\top)$.

$f : C_k(K, \mathbb{R})^{F_1} \to C_k(K, \mathbb{R})^{F_2}$ is *permutation equivariant* if $f(\mathcal{P}\mathcal{B}, \boldsymbol{P}_k \boldsymbol{X}) = \boldsymbol{P}_k f(\mathcal{B}, \boldsymbol{X})$

Proposition

The function $f(\mathcal{B}, \mathcal{X}) := \psi\left(\boldsymbol{L}_1^\downarrow \boldsymbol{X}_1 \boldsymbol{W}_1 + \boldsymbol{L}_1^\uparrow \boldsymbol{X}_1 \boldsymbol{W}_2\right)$ *is permutation equivariant.*

Proof sketch.

Considering only lower adjacencies:

$$(\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top)^\top (\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top)(\boldsymbol{P}_1 \boldsymbol{X}_1) \boldsymbol{W}_1 = \boldsymbol{P}_1 \boldsymbol{B}_1^\top \boldsymbol{B}_1 \boldsymbol{X}_1 \boldsymbol{W} = \boldsymbol{P}_1 (\boldsymbol{L}_1^\downarrow \boldsymbol{X}_1 \boldsymbol{W})$$

Symmetries: Permutation Equivariance

A simplicial complex of dimention d can be specified by all its boundary matrices $\mathcal{B} = (\boldsymbol{B}_1, \ldots, \boldsymbol{B}_d)$. Similarly define a tuple of permutation matrices $\mathcal{P} = (\boldsymbol{P}_0, \ldots, \boldsymbol{P}_d)$ and denote by $\mathcal{P}\mathcal{B} = (\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top, \ldots, \boldsymbol{P}_{d-1} \boldsymbol{B}_d \boldsymbol{P}_d^\top)$.

$f : C_k(K, \mathbb{R})^{F_1} \to C_k(K, \mathbb{R})^{F_2}$ is *permutation equivariant* if $f(\mathcal{P}\mathcal{B}, \boldsymbol{P}_k \boldsymbol{X}) = \boldsymbol{P}_k f(\mathcal{B}, \boldsymbol{X})$

Proposition

The function $f(\mathcal{B}, \mathcal{X}) := \psi\left(\boldsymbol{L}_1^{\downarrow} \boldsymbol{X}_1 \boldsymbol{W}_1 + \boldsymbol{L}_1^{\uparrow} \boldsymbol{X}_1 \boldsymbol{W}_2\right)$ *is permutation equivariant.*

Proof sketch.

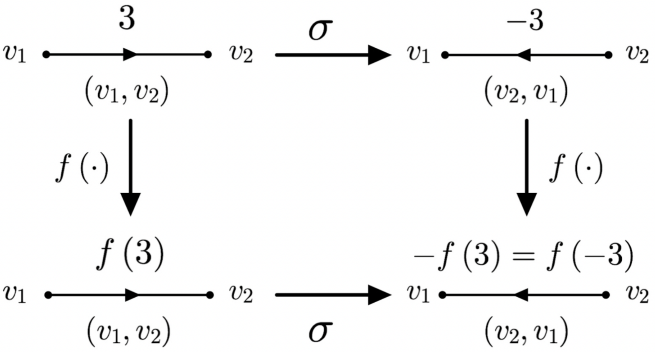
Considering only lower adjacencies:

$(\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top)^\top (\boldsymbol{P}_0 \boldsymbol{B}_1 \boldsymbol{P}_1^\top)(\boldsymbol{P}_1 \boldsymbol{X}_1) \boldsymbol{W}_1 = \boldsymbol{P}_1 \boldsymbol{B}_1^\top \boldsymbol{B}_1 \boldsymbol{X}_1 \boldsymbol{W} = \boldsymbol{P}_1 (\boldsymbol{L}_1^{\downarrow} \boldsymbol{X}_1 \boldsymbol{W})$

Upper adjacencies proceed similarly and the nonlinearity ψ commutes with $\boldsymbol{P}_1$. □

Symmetries: Orientation Equivariance

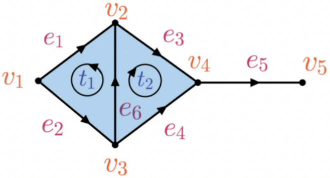
Mathematically, the choice of orientation is irrelevant. Therefore, we would like our model to produce the same outputs up to a change in orientation.



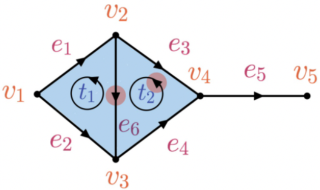
The function f must be odd.

Symmetries: Orientation Equivariance

If a simplex changes its orientation, then it flips its relative orientation with respect to its adjacent neighbours.



$$B_1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ +1 & 0 & -1 & 0 & 0 & +1 \\ 0 & +1 & 0 & -1 & 0 & -1 \\ 0 & 0 & +1 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 \end{bmatrix} \end{array} \qquad B_2 = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{bmatrix} -1 & 0 \\ +1 & 0 \\ 0 & +1 \\ 0 & -1 \\ 0 & 0 \\ +1 & +1 \end{bmatrix} \end{array}$$



$$B_1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ +1 & 0 & -1 & 0 & 0 & -1 \\ 0 & +1 & 0 & -1 & 0 & +1 \\ 0 & 0 & +1 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 \end{bmatrix} \end{array} \qquad B_2 = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{bmatrix} -1 & 0 \\ +1 & 0 \\ 0 & -1 \\ 0 & +1 \\ 0 & 0 \\ -1 & +1 \end{bmatrix} \end{array}$$

This amounts to flipping the sign in the corresponding rows and columns of the boundary matrices.

Symmetries: Orientation Equivariance

Consider a tuple of matrices $\mathcal{T} = (\boldsymbol{T}_0, \ldots, \boldsymbol{T}_d)$, where each $\boldsymbol{T}_i$ is a diagonal matrix with values in $\{\pm 1\}$. Additionally, because vertices always have a positive orientation, we restrict $\boldsymbol{T}_0 = I$. Then denote by $\mathcal{T}\mathcal{B} = (\boldsymbol{T}_0 \boldsymbol{B}_1 \boldsymbol{T}_1, \ldots, \boldsymbol{T}_{d-1} \boldsymbol{B}_d \boldsymbol{T}_d)$

[9] Bodnar, Frasca, Yuguang Wang, et al., "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks", 2021.
[10] Goh et al., "Simplicial Attention Networks", 2022.

Symmetries: Orientation Equivariance

Consider a tuple of matrices $\mathcal{T} = (\boldsymbol{T}_0, \ldots, \boldsymbol{T}_d)$, where each $\boldsymbol{T}_i$ is a diagonal matrix with values in $\{\pm 1\}$. Additionally, because vertices always have a positive orientation, we restrict $\boldsymbol{T}_0 = I$. Then denote by $\mathcal{T}\mathcal{B} = (\boldsymbol{T}_0 \boldsymbol{B}_1 \boldsymbol{T}_1, \ldots, \boldsymbol{T}_{d-1} \boldsymbol{B}_d \boldsymbol{T}_d)$

$f : C_k(K, \mathbb{R})^{F_1} \to C_k(K, \mathbb{R})^{F_2}$ is *orientation equivariant* if $f(\mathcal{T}\mathcal{B}, \boldsymbol{T}_k \boldsymbol{X}) = \boldsymbol{T}_k f(\mathcal{B}, \boldsymbol{X})$

Proposition

$f(\mathcal{B}, \mathcal{X}) := \psi\left(\boldsymbol{L}_1^{\downarrow} \boldsymbol{X}_1 \boldsymbol{W}_1 + \boldsymbol{L}_1^{\uparrow} \boldsymbol{X}_1 \boldsymbol{W}_2\right)$ *is orientation equivariant when* ψ *is odd.*

Proof sketch.
$\psi\left((\boldsymbol{T}_0 \boldsymbol{B}_1 \boldsymbol{T}_1)^{\top} (\boldsymbol{T}_0 \boldsymbol{B}_1 \boldsymbol{T}_1) \boldsymbol{T}_1 \boldsymbol{X}_1 \boldsymbol{W}\right) = \psi\left(\boldsymbol{T}_1 \boldsymbol{L}_1^{\downarrow} \boldsymbol{X}_1 \boldsymbol{W}\right)$. Odd ψ commutes with $\boldsymbol{T}_1$. □

[9] Bodnar, Frasca, Yuguang Wang, et al., "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks", 2021.
[10] Goh et al., "Simplicial Attention Networks", 2022.

Symmetries: Orientation Equivariance

Consider a tuple of matrices $\mathcal{T} = (\boldsymbol{T}_0, \ldots, \boldsymbol{T}_d)$, where each $\boldsymbol{T}_i$ is a diagonal matrix with values in $\{\pm 1\}$. Additionally, because vertices always have a positive orientation, we restrict $\boldsymbol{T}_0 = I$. Then denote by $\mathcal{T}\mathcal{B} = (\boldsymbol{T}_0 \boldsymbol{B}_1 \boldsymbol{T}_1, \ldots, \boldsymbol{T}_{d-1} \boldsymbol{B}_d \boldsymbol{T}_d)$

$f : C_k(K, \mathbb{R})^{F_1} \to C_k(K, \mathbb{R})^{F_2}$ is *orientation equivariant* if $f(\mathcal{T}\mathcal{B}, \boldsymbol{T}_k \boldsymbol{X}) = \boldsymbol{T}_k f(\mathcal{B}, \boldsymbol{X})$

Proposition

$f(\mathcal{B}, \mathcal{X}) := \psi\left(\boldsymbol{L}_1^{\downarrow} \boldsymbol{X}_1 \boldsymbol{W}_1 + \boldsymbol{L}_1^{\uparrow} \boldsymbol{X}_1 \boldsymbol{W}_2\right)$ *is orientation equivariant when* ψ *is odd.*

Proof sketch.

$\psi\left((\boldsymbol{T}_0 \boldsymbol{B}_1 \boldsymbol{T}_1)^\top (\boldsymbol{T}_0 \boldsymbol{B}_1 \boldsymbol{T}_1) \boldsymbol{T}_1 \boldsymbol{X}_1 \boldsymbol{W}\right) = \psi\left(\boldsymbol{T}_1 \boldsymbol{L}_1^{\downarrow} \boldsymbol{X}_1 \boldsymbol{W}\right)$. Odd ψ commutes with $\boldsymbol{T}_1$. □

This idea can be generalised to general simplicial message passing architectures[9], including attention[10].

[9] Bodnar, Frasca, Yuguang Wang, et al., "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks", 2021.
[10] Goh et al., "Simplicial Attention Networks", 2022.

Convolutions on Chains Complexes

The boundary maps produce a *chain complex*, which is a sequence of vector spaces:

$$0 \to C_n(K, \mathbb{R}) \to \cdots \xrightarrow{\partial_{k+1}} C_k(K, \mathbb{R}) \xrightarrow{\partial_k} C_{k-1}(K, \mathbb{R}) \cdots \xrightarrow{\partial_2} C_1(K, \mathbb{R}) \xrightarrow{\partial_1} C_0(K, \mathbb{R}) \to 0$$

Convolutions on Chains Complexes

The boundary maps produce a *chain complex*, which is a sequence of vector spaces:

$$0 \to C_n(K,\mathbb{R}) \to \cdots \xrightarrow{\partial_{k+1}} C_k(K,\mathbb{R}) \xrightarrow{\partial_k} C_{k-1}(K,\mathbb{R}) \cdots \xrightarrow{\partial_2} C_1(K,\mathbb{R}) \xrightarrow{\partial_1} C_0(K,\mathbb{R}) \to 0$$

We can see our convolution works on chain complexes and boundary matrices ensure communication between different dimensions of this chain.

$$\boldsymbol{Y} = \psi\big(\boldsymbol{L}_1^{\downarrow}\boldsymbol{X}_1\boldsymbol{W}_1 + \boldsymbol{L}_1^{\uparrow}\boldsymbol{X}_1\boldsymbol{W}_2 + \boldsymbol{B}_1^{\top}\boldsymbol{X}_0\boldsymbol{W}_3 + \boldsymbol{B}_2\boldsymbol{X}_2\boldsymbol{W}_4 + \boldsymbol{X}_1\boldsymbol{W}_5\big)$$

Convolutions on Chains Complexes

The boundary maps produce a *chain complex*, which is a sequence of vector spaces:

$$0 \to C_n(K,\mathbb{R}) \to \cdots \xrightarrow{\partial_{k+1}} C_k(K,\mathbb{R}) \xrightarrow{\partial_k} C_{k-1}(K,\mathbb{R}) \cdots \xrightarrow{\partial_2} C_1(K,\mathbb{R}) \xrightarrow{\partial_1} C_0(K,\mathbb{R}) \to 0$$

We can see our convolution works on chain complexes and boundary matrices ensure communication between different dimensions of this chain.

$$\boldsymbol{Y} = \psi\big(\boldsymbol{L}_1^{\downarrow}\boldsymbol{X}_1\boldsymbol{W}_1 + \boldsymbol{L}_1^{\uparrow}\boldsymbol{X}_1\boldsymbol{W}_2 + \boldsymbol{B}_1^{\top}\boldsymbol{X}_0\boldsymbol{W}_3 + \boldsymbol{B}_2\boldsymbol{X}_2\boldsymbol{W}_4 + \boldsymbol{X}_1\boldsymbol{W}_5\big)$$

Proposition

ψ must be non-linear for simplices that are more than two dimensions apart to communicate.

Convolutions on Chains Complexes

The boundary maps produce a *chain complex*, which is a sequence of vector spaces:

$$0 \to C_n(K,\mathbb{R}) \to \cdots \xrightarrow{\partial_{k+1}} C_k(K,\mathbb{R}) \xrightarrow{\partial_k} C_{k-1}(K,\mathbb{R}) \cdots \xrightarrow{\partial_2} C_1(K,\mathbb{R}) \xrightarrow{\partial_1} C_0(K,\mathbb{R}) \to 0$$

We can see our convolution works on chain complexes and boundary matrices ensure communication between different dimensions of this chain.

$$\boldsymbol{Y} = \psi\big(\boldsymbol{L}_1^{\downarrow}\boldsymbol{X}_1\boldsymbol{W}_1 + \boldsymbol{L}_1^{\uparrow}\boldsymbol{X}_1\boldsymbol{W}_2 + \boldsymbol{B}_1^{\top}\boldsymbol{X}_0\boldsymbol{W}_3 + \boldsymbol{B}_2\boldsymbol{X}_2\boldsymbol{W}_4 + \ \boldsymbol{X}_1\boldsymbol{W}_5\big)$$

Proposition

ψ must be non-linear for simplices that are more than two dimensions apart to communicate.

Proof sketch.

If ψ is the identity, consider how 2-chains propagate to the 0-chain level:

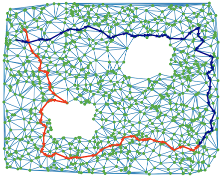
$\boldsymbol{B}_1(\boldsymbol{B}_2\boldsymbol{X}_2\boldsymbol{W}_4^1)\boldsymbol{W}_4^2 = (\boldsymbol{B}_1\boldsymbol{B}_2)\boldsymbol{X}_2\boldsymbol{W}_4^1 = 0$ □

Application: Trajectory Classification

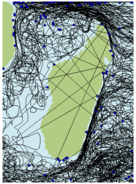
We are interested in classifying trajectories represented as 1-chains. At train time we use a fixed orientation and, at test time, we randomly flip the orientations of the edges.

Method	Synthetic Flow Train	Synthetic Flow Test	Ocean Drifters Train	Ocean Drifters Test
GNN L_0-inv	63.9±2.4	61.0±4.2	70.1±2.3	63.5±6.0
MPSN L_0-inv	88.2±5.1	85.3±5.8	84.6±4.0	**71.5±4.1**
MPSN - ReLU	**100.0±0.0**	50.0±0.0	**100.0±0.0**	46.5±5.7
MPSN - Id	88.0±3.1	82.6±3.0	94.6±0.9	**73.0±2.7**
MPSN - Tanh	97.9±0.7	**95.2±1.8**	**99.7±0.5**	**72.5±0.0**

Trajectory classification accuracy. The tasks are inspired from Schaub et al.[18].



The task is to classify random walks.



The task is to classify ocean drifter trajectories around Madagascar.

[18] Schaub et al., "Random walks on simplicial complexes and the normalized Hodge 1-Laplacian", 2020

Message Passing on Cell Complexes

Cell complexes

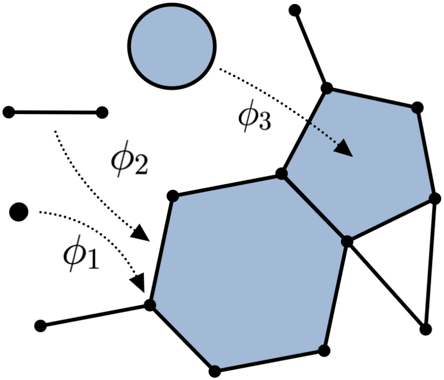
The subset inclusion property of simplicial complexes can be restrictive, but we still want to exploit the topological tools they come with.

Cell complexes

The subset inclusion property of simplicial complexes can be restrictive, but we still want to exploit the topological tools they come with.

A finite (regular) *cell complex* is a topological space X formed of a finite disjoint union of subspaces called *cells* such that:

1. Each cell is homeomorphic to $\mathbb{R}^n$, for some n.
2. The closure of each cell is homeomorphic to a closed ball in $\mathbb{R}^n$.

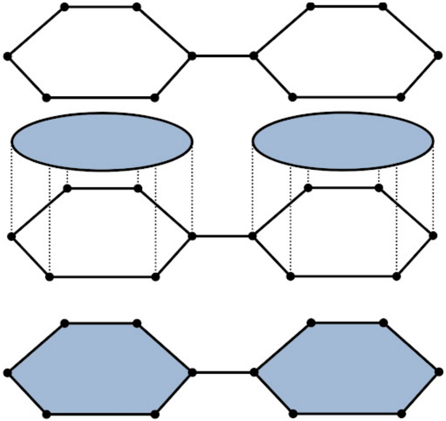


A cell complex X and the corresponding homeomorphisms to the closed balls for three cells of different dimensions in the complex.

Constructing cell complexes

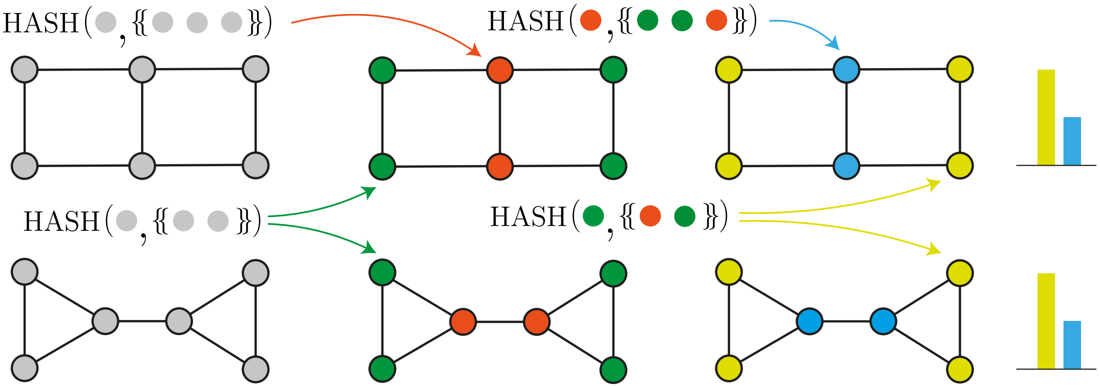
Cell complexes can be constructed hierarhically:

1. Start with a set of vertices.
2. Glue the boundary of a set of line segments to these vertices.
3. Glue the boundary of two-dimensional disks to cycles present in the graph previously obtained.



The Weisfeiler Lehman Test

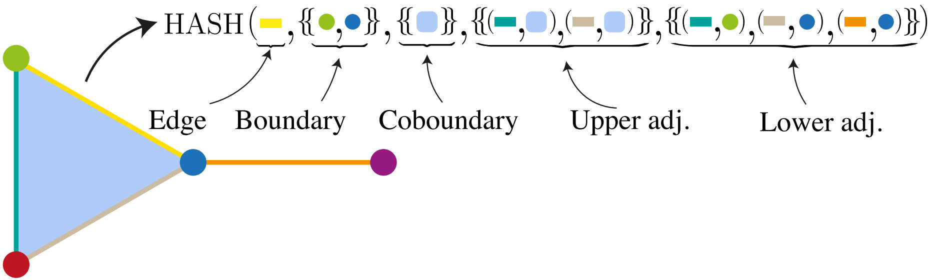
The WL test is an heuristic algorithm for testing the isomorphims of two graphs. It performs iterative colour-refinement.



If the two graphs converge to the same histogram, the test is inconclusive. In this case, the WL test fails to distinguish these non-isomorphic graphs.

The Cellular Weisfeiler Lehman Test

Generalising the Weisfeiler-Lehman algorithm for graphs, we can define a cellular version of the WL test[11]. We call this *cellular WL*.



An example of a colour refinement step of CWL for an edge of the cell complex. This iteration is performed over all the cells in the complex until convergence.

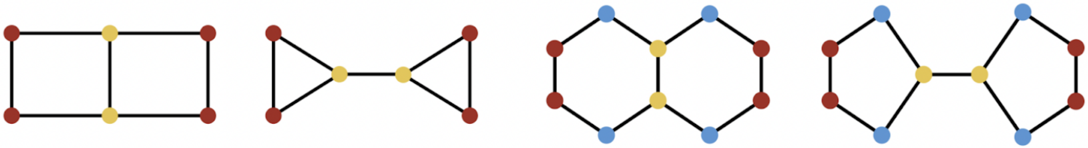
[11] Bodnar, Frasca, Otter, et al., "Weisfeiler and Lehman Go Cellular: CW Networks", 2021.

Expressive power of CWL

Let k-CL, k-IC, k-C be the "lifting" maps attaching cells to all the cliques, induced cycles and simple cycles, respectively, of size at most k.

Theorem

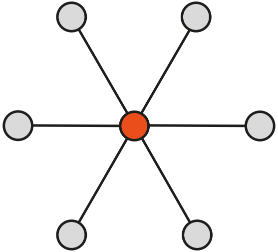
For $k \geq 3$, CWL(k-CL), CWL(k-IC) and CWL(k-C) are strictly more powerful than WL.



Pairs of graphs WL cannot distinguish but CWL can.

Sparse adjacencies

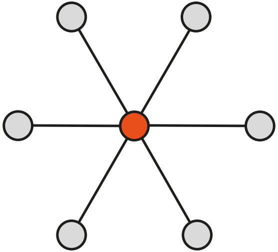
Considering all these adjacencies might make the computational complexity explode.



Even though the star graph is sparse, all the edges are lower adjacent to each other and they exchange messages.

Sparse adjacencies

Considering all these adjacencies might make the computational complexity explode.

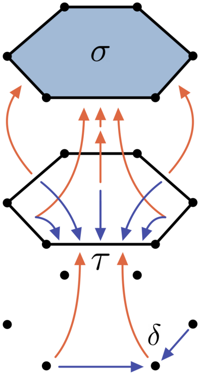


Even though the star graph is sparse, all the edges are lower adjacent to each other and they exchange messages.

Theorem

CWL without coboundary and lower adjacencies is as expressive as CWL with the full set of adjacencies.

Topological Message Passing



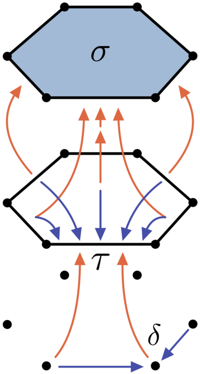
Orange arrows indicate boundary messages received by cells σ and τ, while blue ones show upper messages received by cells τ and δ

The cells receive two types of messages:

$$m_{\mathcal{B}}^{t+1}(\sigma) = \mathrm{AGG}_{\tau \in \mathcal{B}(\sigma)}\Big(M_{\mathcal{B}}\big(h_\sigma^t, h_\tau^t\big)\Big)$$

$$m_{\uparrow}^{t+1}(\sigma) = \mathrm{AGG}_{\tau \in \mathcal{N}_\uparrow(\sigma), \delta \in \mathcal{C}(\sigma,\tau)}\Big(M_{\uparrow}\big(h_\sigma^t, h_\tau^t, h_\delta^t\big)\Big)$$

Topological Message Passing



Orange arrows indicate boundary messages received by cells σ and τ, while blue ones show upper messages received by cells τ and δ

The cells receive two types of messages:

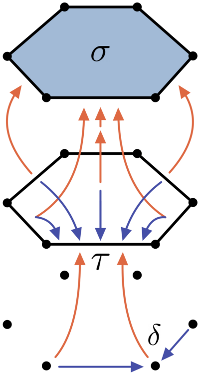
$$m_{\mathcal{B}}^{t+1}(\sigma) = \mathrm{AGG}_{\tau \in \mathcal{B}(\sigma)}\Big(M_{\mathcal{B}}\big(h_\sigma^t, h_\tau^t\big)\Big)$$

$$m_{\uparrow}^{t+1}(\sigma) = \mathrm{AGG}_{\tau \in \mathcal{N}_\uparrow(\sigma), \delta \in \mathcal{C}(\sigma,\tau)}\Big(M_{\uparrow}\big(h_\sigma^t, h_\tau^t, h_\delta^t\big)\Big)$$

The update function takes as input these messages:

$$h_\sigma^{t+1} = U\Big(h_\sigma^t, m_{\mathcal{B}}^t(\sigma), m_\uparrow^{t+1}(\sigma)\Big)$$

Topological Message Passing



Orange arrows indicate boundary messages received by cells σ and τ, while blue ones show upper messages received by cells τ and δ

The cells receive two types of messages:

$$m_{\mathcal{B}}^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{B}(\sigma)}\Big(M_{\mathcal{B}}\big(h_\sigma^t, h_\tau^t\big)\Big)$$

$$m_{\uparrow}^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{N}_{\uparrow}(\sigma), \delta \in \mathcal{C}(\sigma,\tau)}\Big(M_{\uparrow}\big(h_\sigma^t, h_\tau^t, h_\delta^t\big)\Big)$$

The update function takes as input these messages:

$$h_\sigma^{t+1} = U\Big(h_\sigma^t, m_{\mathcal{B}}^t(\sigma), m_{\uparrow}^{t+1}(\sigma)\Big)$$

A readout function computes a final representation:

$$\text{READOUT}\big(\{\!\{h_\sigma^L\}\!\}_{dim(\sigma)=0}, \{\!\{h_\sigma^L\}\!\}_{dim(\sigma)=1}, \{\!\{h_\sigma^L\}\!\}_{dim(\sigma)=2}\big)$$

Message passing on cell complexes has also been considered by Hajij, Istvan, and Zamzmi, “Cell Complex Neural Networks”, 2020

Expressive power

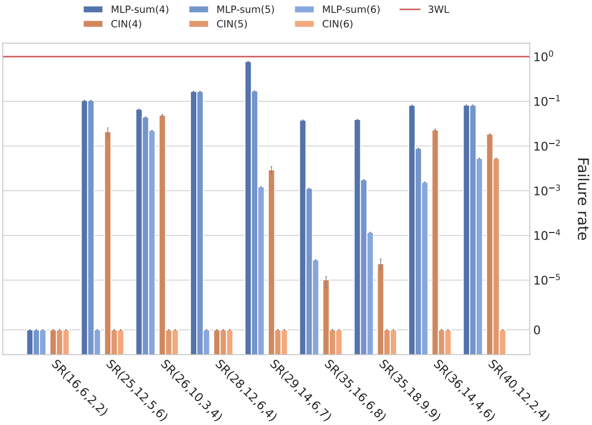
Theorem

When using injective neighbourhood aggregators and a sufficient number of layers, topological message passing is as powerful as CWL.

Expressive power

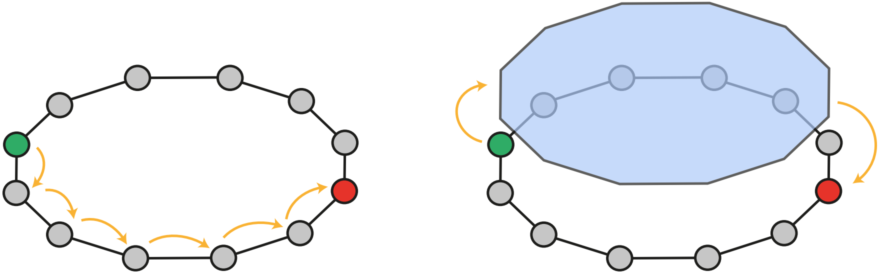
Theorem

When using injective neighbourhood aggregators and a sufficient number of layers, topological message passing is as powerful as CWL.



Long-range interactions

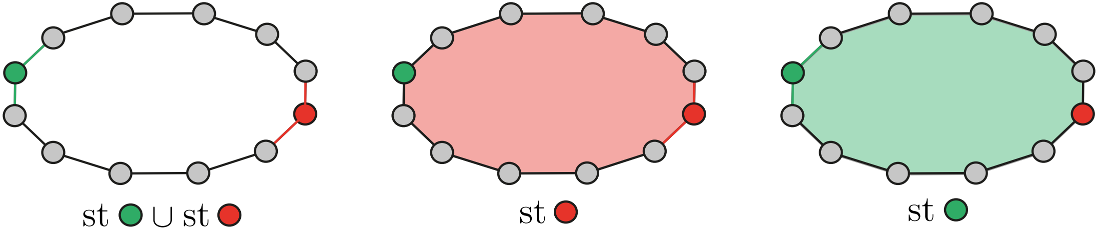
The neighbourhood structure induced by cell complex naturally allows long-range interactions with a reduced number of computational steps.



Comparison between regular message passing on graphs and topological message passing.

A more sophisticated topological structure

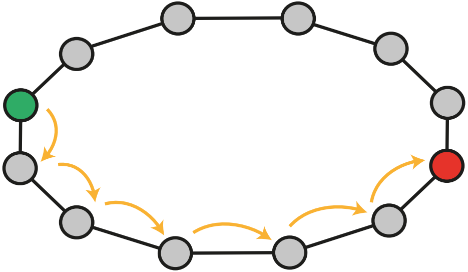
Given a cell σ define *the star* of σ, denoted by $\mathrm{st}\ \sigma$, as the union of all the cells having σ as a face. The stars of all cells form the basis of a topology for the cell complex.



In the graph case (left), the open neighbourhoods of the two nodes do not intersect. In the higher-dimensional cell complex (right), the neighbourhoods of the nodes are significantly expanded.

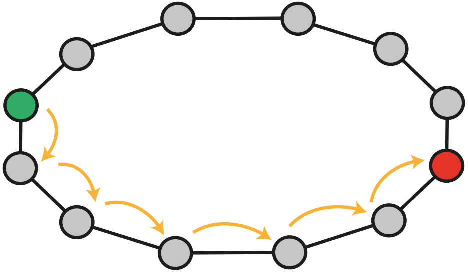
Long-range interactions experiment

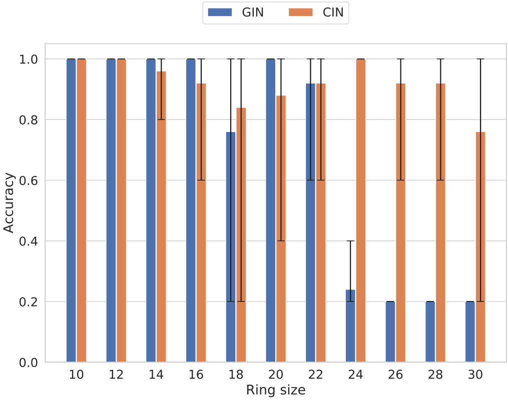
We validated the benefits of long-range interactions with an experiment where the model has to transfer a value from one side of the ring to the other.



Long-range interactions experiment

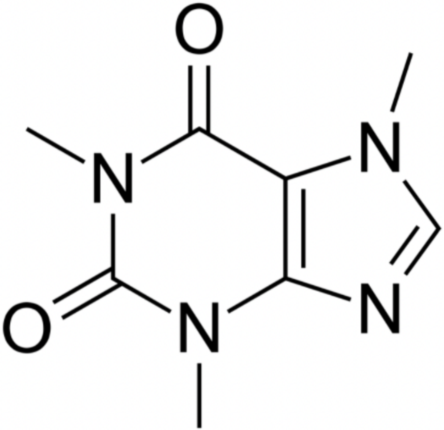
We validated the benefits of long-range interactions with an experiment where the model has to transfer a value from one side of the ring to the other.

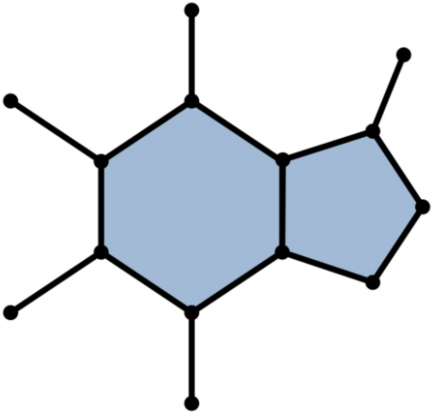




Domain alignment

This type of space aligns well with certain applications such as molecular modelling.





Molecular Property Prediction

ZINC (MAE), ZINC-FULL (MAE) and Mol-HIV (ROC-AUC).

Method	ZINC ↓ No Edge Feat.	ZINC ↓ With Edge Feat.	ZINC-FULL ↓ All methods	MOLHIV ↑ All methods
GCN	0.469±0.002	N/A	N/A	76.06±0.97
GAT	0.463±0.002	N/A	N/A	N/A
GatedGCN	0.422±0.006	0.363±0.009	N/A	N/A
GIN	0.408±0.008	0.252±0.014	0.088±0.002	77.07±1.49
PNA	0.320±0.032	0.188±0.004	N/A	79.05±1.32
DGN	0.219±0.010	0.168±0.003	N/A	79.70±0.97
HIMP	N/A	0.151±0.006	0.036±0.002	78.80±0.82
GSN	0.139±0.007	0.108±0.018	N/A	77.99±1.00
CIN-small (Ours)	0.139±0.008	0.094±0.004	0.044±0.003	80.55±1.04
CIN (Ours)	**0.115±0.003**	**0.079±0.006**	**0.022±0.002**	**80.94±0.57**

Thanks for your attention!

Email: cb2015@cam.ac.uk
Twitter: @crisbodnar