AIDOS LAB
AI FOR DATA-ORIENTED SCIENCE

UNI FR

Learning on Topological Data

Session 3 — Topological and Geometric Deep Learning: Theory, Methods and Applications

Universidad Complutense de Madrid

Inés García Redondo — May 13, 2026

Session 3 — Learning on Topological data

Outline

- Graphs are all around us...
- Towards learning on graphs: learning on sets
  - Permutation invariance and equivariance
- Learning on graphs: message-passing
- Expressivity: what tasks can message-passing solve?
- Extending message-passing to higher-order domains
- Limitations of message-passing
  - Over-squashing and over-smoothing
- Final thoughts on message-passing

Session 3 — Learning on Topological data

Resources and acknowledgements

First part of the talk inspired by Theoretical Foundations of Graph Neural Networks, by Petar Veličković

(https://www.youtube.com/watch?v=uF53xsT7mjc)

Also take a look at:

Bronstein, Michael M., Joan Bruna, Taco Cohen, and Petar Veličković. "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges." *arXiv Preprint arXiv:2104.13478*, 2021.

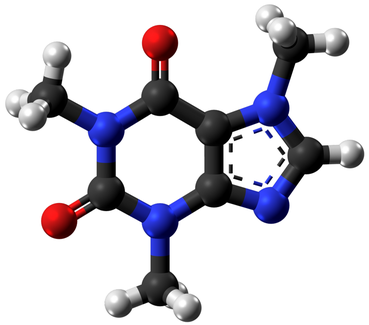
Motivation

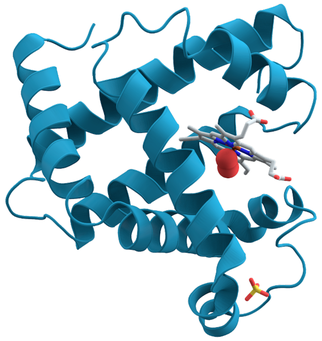
Session 3 — Learning on Topological Data

Inés García Redondo — May 13, 2026

Graphs are all around us...

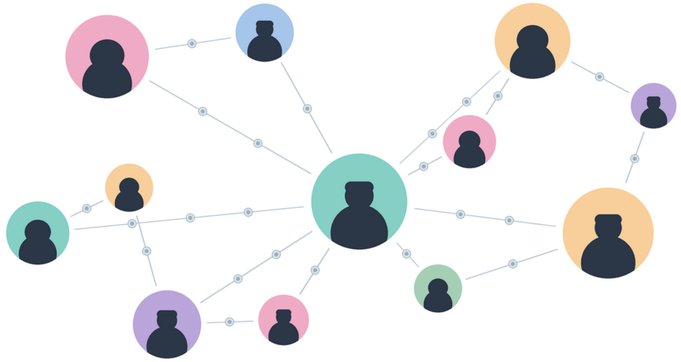
And so the DL grows





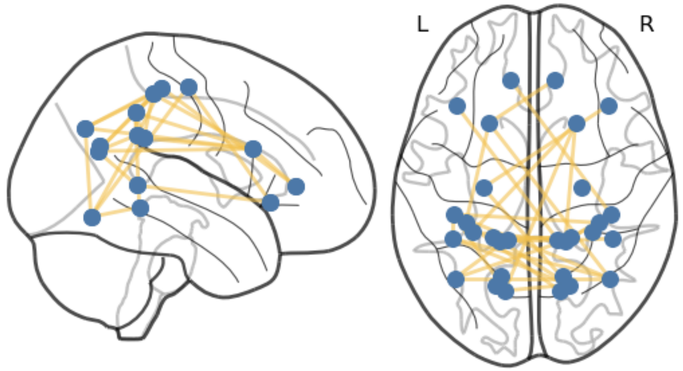
Molecules and proteins

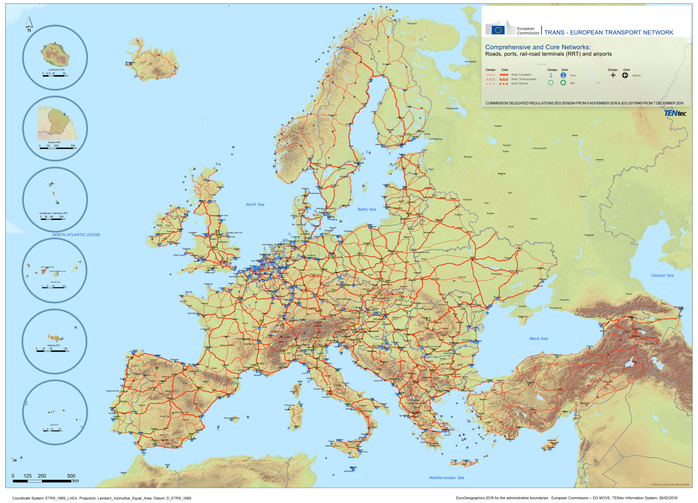
Collaboration networks



Social networks

Brain connectome



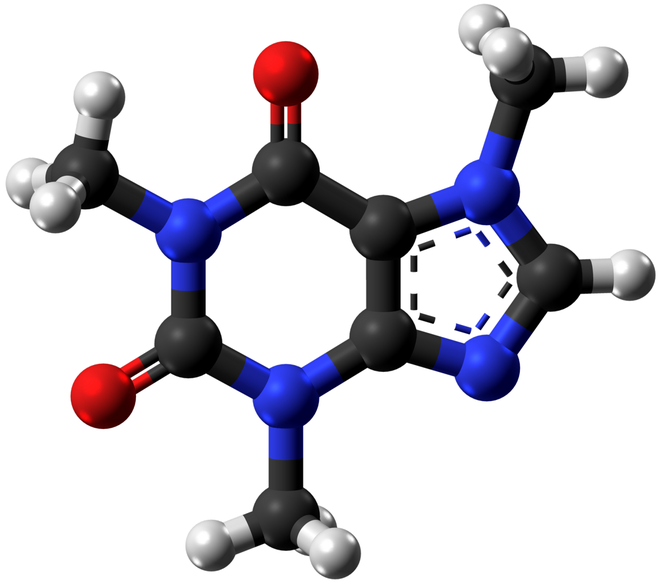


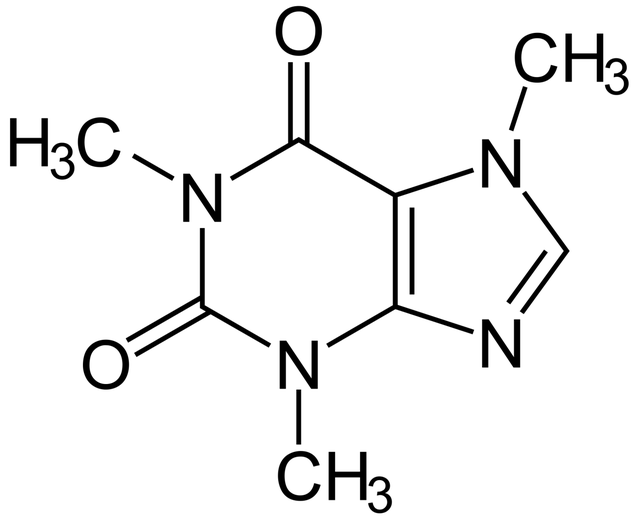
By European Commission
https://ec.europa.eu/transport/infrastructure/tentec/tentec-portal/site/en/maps.html, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=146628735

Molecules are graphs

Caffeine molecule

- Nodes given by atoms, edges by bonds
- Features as atom type, charge, bond type

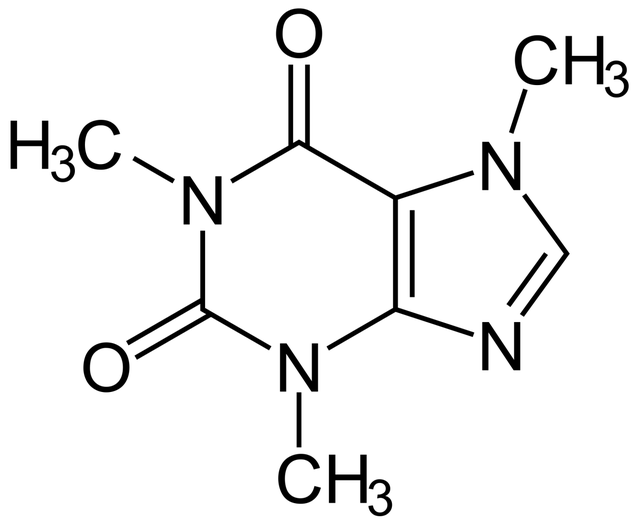




Caffeine molecule ($C_8H_{10}N_4O_2$)

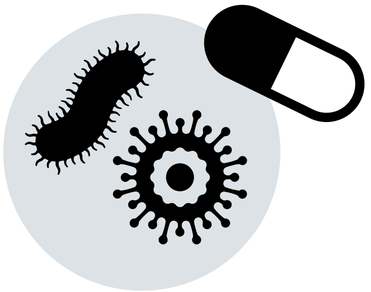
Molecular classification tasks

An example: drug discovery



Caffeine molecule ($C_8H_{10}N_4O_2$)

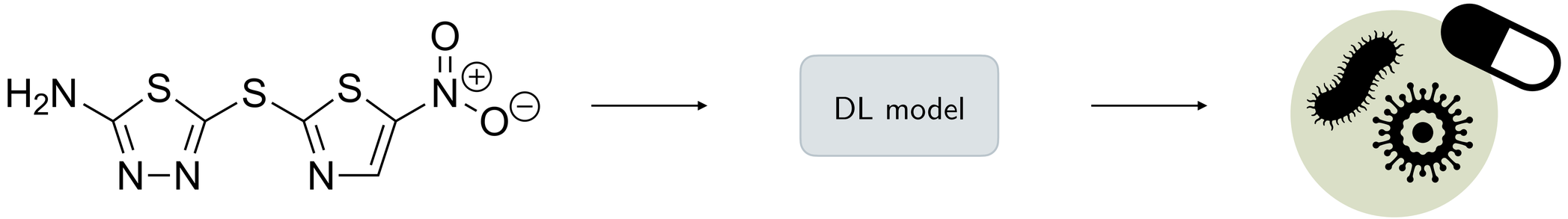
DL model



Does it inhibit E.coli?

Molecular classification tasks

An example: drug discovery



Hallicin

Does it inhibit E.coli?

Stokes, Jonathan M., Kevin Yang, Kyle Swanson, et al. "A Deep Learning Approach to Antibiotic Discovery." *Cell* 180, no. 4 (2020): 688-702.e13.

Transportation networks are graphs

The motorway network in Madrid

Learning on graphs: graph neural networks (GNNs)

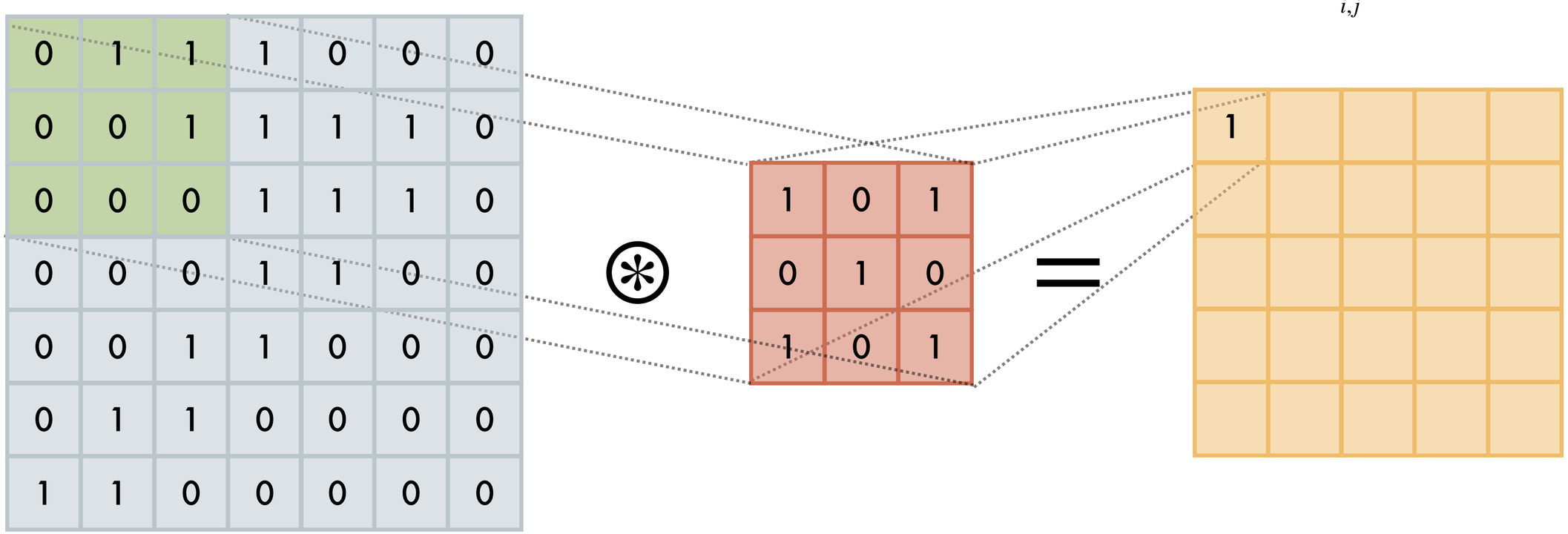
Session 3 — Learning on Topological Data

Inés García Redondo — May 13, 2026

Revisiting CNNs

Locality

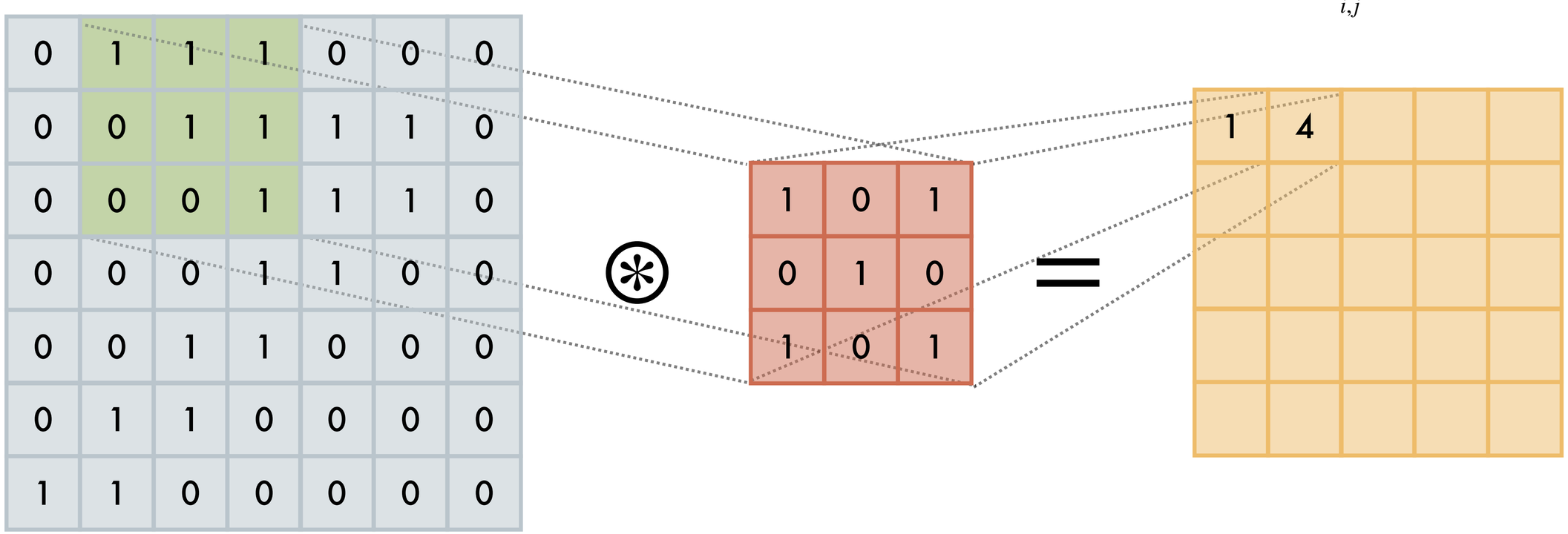
$$(I \circledast K)(x, y) = \sum_{i,j} K(i, j)\ I(x + i, y + j)$$



Revisiting CNNs

Locality

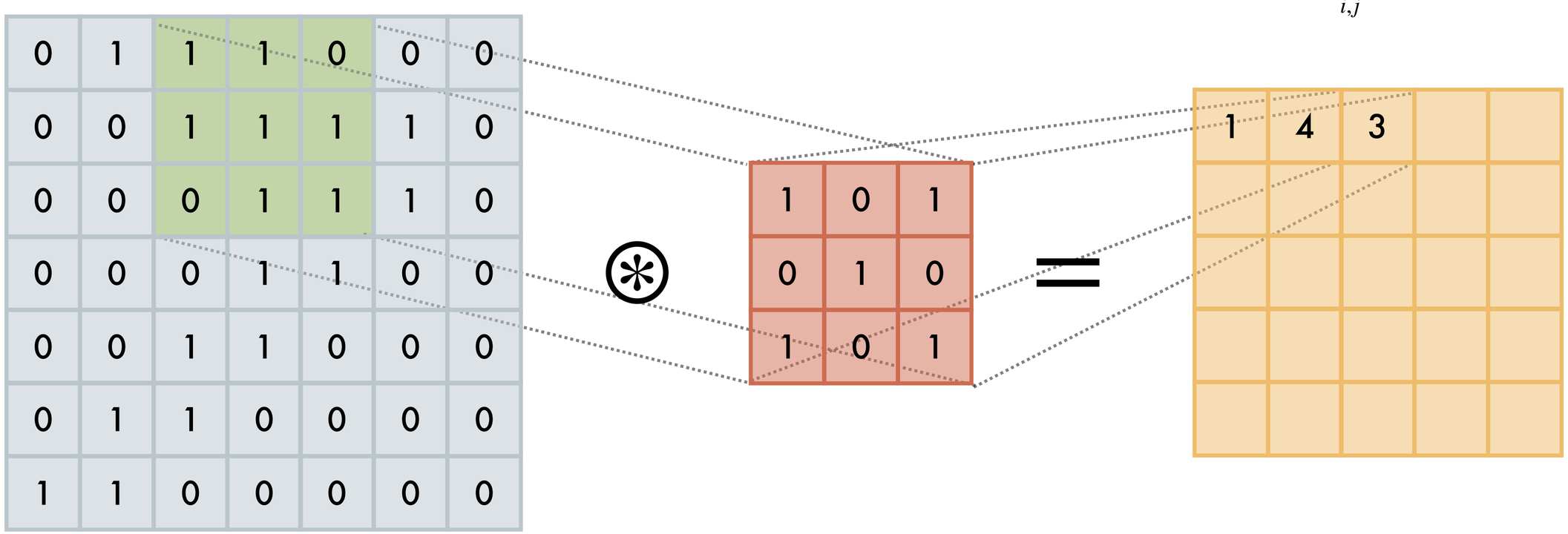
$$(I \circledast K)(x, y) = \sum_{i,j} K(i, j)\ I(x + i, y + j)$$



Revisiting CNNs

Locality

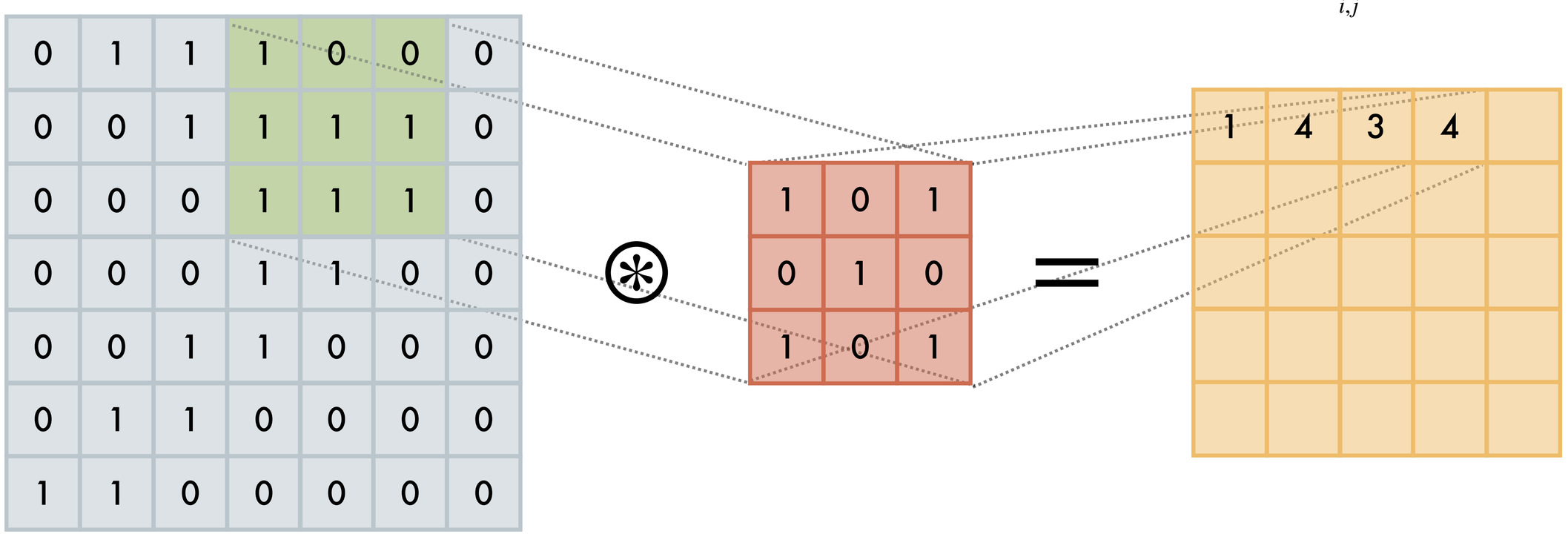
$$(I \circledast K)(x, y) = \sum_{i,j} K(i, j)\ I(x + i, y + j)$$



Revisiting CNNs

Locality

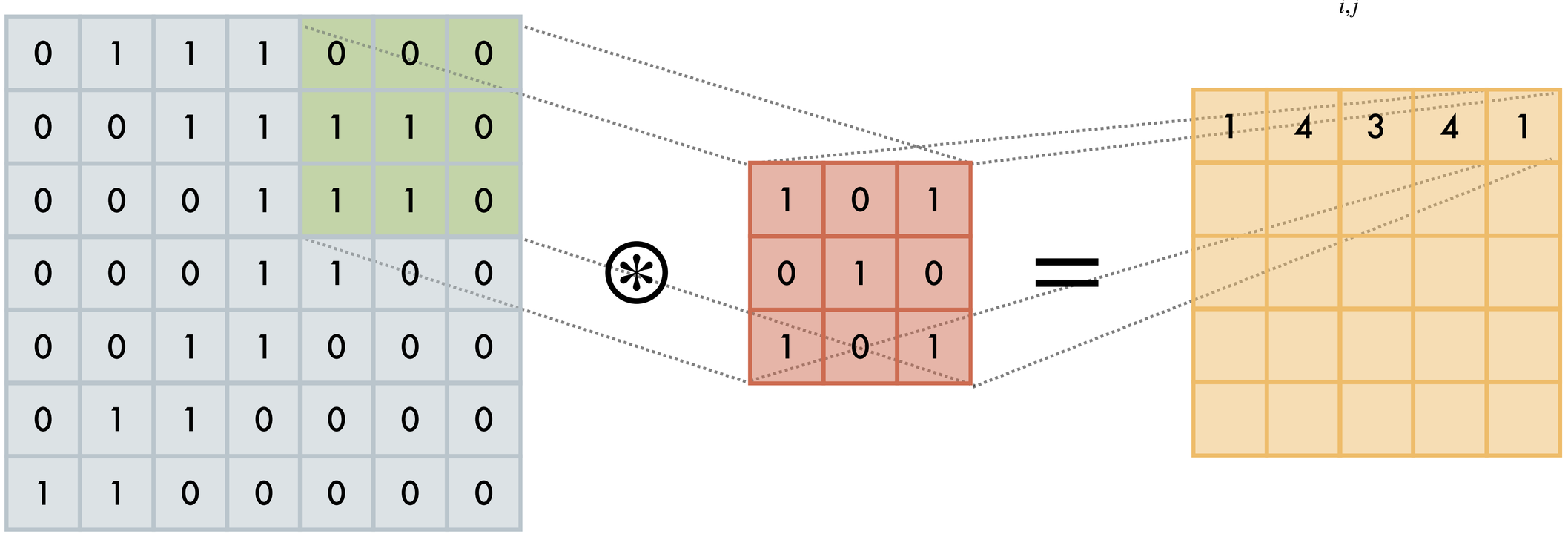
$$(I \circledast K)(x, y) = \sum_{i,j} K(i, j)\ I(x + i, y + j)$$



Revisiting CNNs

Locality

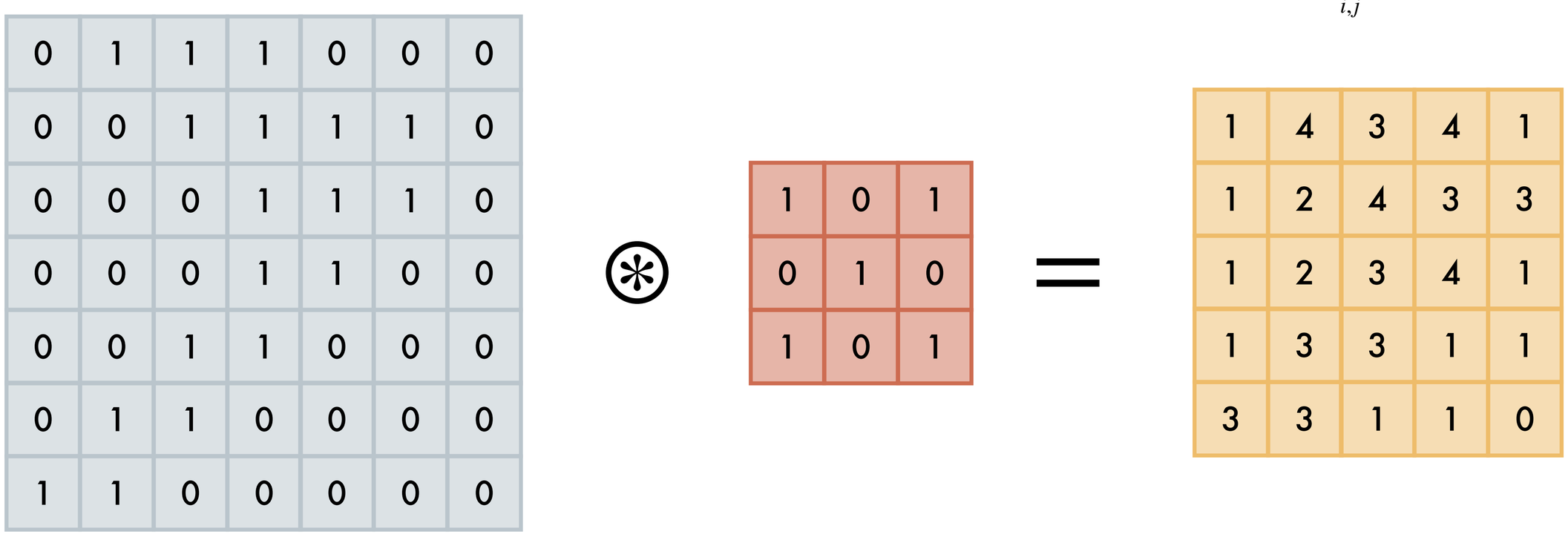
$$(I \circledast K)(x, y) = \sum_{i,j} K(i, j)\ I(x + i, y + j)$$



Revisiting CNNs

Locality

$$(I \circledast K)(x, y) = \sum_{i,j} K(i, j)\ I(x + i, y + j)$$



Revisiting CNNs

Translation equivariance

I

0	1	1	1	0	0	0
0	0	1	1	1	1	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	1	1	0	0	0	0
1	1	0	0	0	0	0

$\circledast$

K

1	0	1
0	1	0
1	0	1

$=$

$I \circledast K$

1	4	3	4	1
1	2	4	3	3
1	2	3	4	1
1	3	3	1	1
3	3	1	1	0

Revisiting CNNs

Translation equivariance

$T(I)$

1	1	1	0	0	0	0
0	1	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	1
1	0	0	0	0	0	1

$\circledast$

K

1	0	1
0	1	0
1	0	1

$=$

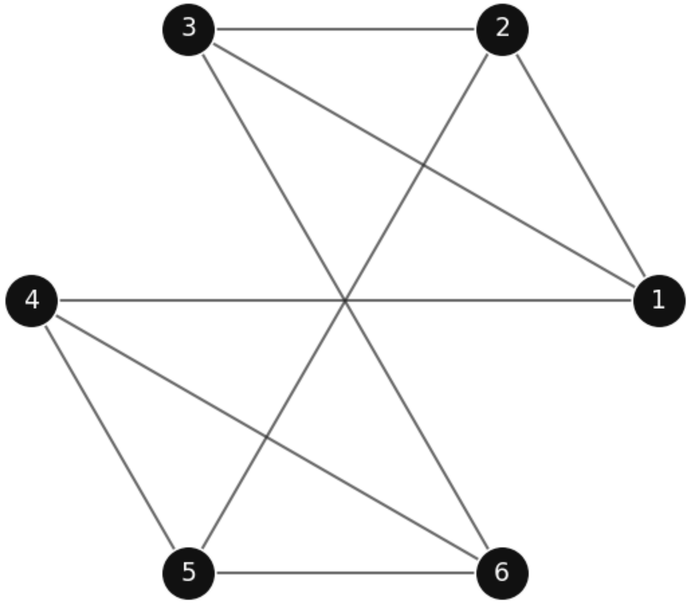
$T(I) \circledast K = T(I \circledast K)$

4	3	4	1	1
2	4	3	3	1
2	3	4	1	1
3	3	1	1	1
3	1	1	0	3

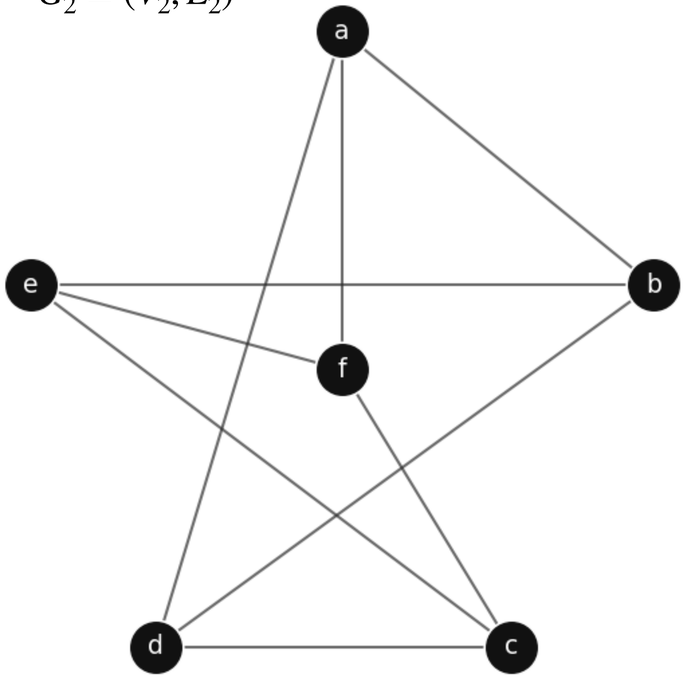
Symmetries and invariances

Isomorphism invariance in graphs

$G_1 = (V_1, E_1)$



$G_2 = (V_2, E_2)$



$$\phi : V_1 \to V_2, \quad (i, j) \mapsto (\phi(i), \phi(j))$$

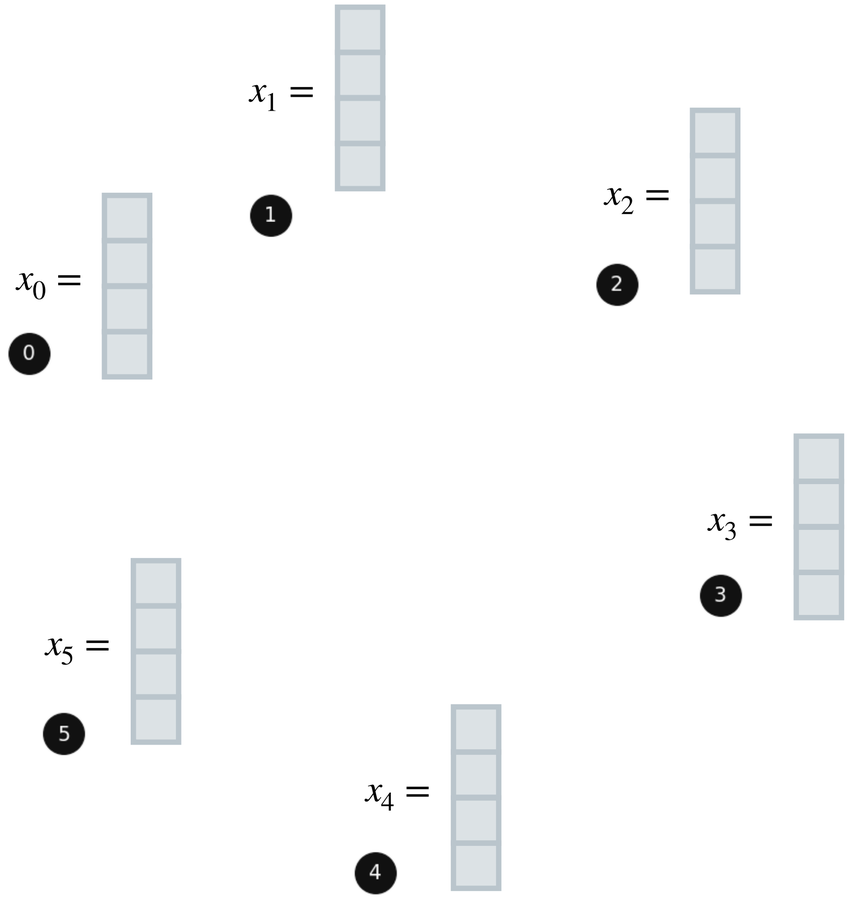
Learning on sets: permutation invariance

Session 3 — Learning on Topological Data

Inés García Redondo — May 13, 2026

Symmetries and invariances

Permutations on nodes



- Start with a set of nodes V and no edges
- Features on the nodes: $x_i \in \mathbb{R}^k$ for $i \in V$
- Stack them: $X = (x_1, \ldots, x_n)^\top \in \mathbb{R}^{n \times k}$

  Already specifies an order in which you are visiting the nodes

  We want our architectures to not depend on this choice

What transformations change the ordering of the nodes?

Permutations $\sigma \in S_n$

Can be represented as *permutation matrices*, which we can apply to X

Example:

$$\sigma = (1\,3\,2\,4) \in S_4$$

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$$

$$P_\sigma = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} x_1^\top \\ x_2^\top \\ x_3^\top \\ x_4^\top \end{pmatrix}, \quad P_\sigma X = \begin{pmatrix} x_3^\top \\ x_4^\top \\ x_2^\top \\ x_1^\top \end{pmatrix}$$

Permutation invariance and equivariance

We want: $f_\theta : \mathbb{R}^{n\times k} \to \mathbb{R}^p, \quad X \mapsto f_\theta(X)$

Graph level tasks: f_θ is **permutation invariant** if for all permutation matrices P

$$f_\theta(PX) = f_\theta(X)$$

Example: DeepSets[1]

$$f(X) = \varphi\left(\sum_{i\in V} \phi(i)\right)$$

You could change the sum to any permutation invariant aggregator $\oplus$ (sum, avg, max)

Node level tasks: f_θ is **permutation equivariant** if for all permutation matrices P

$$f_\theta(PX) = Pf_\theta(X)$$

[1]Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R. Salakhutdinov, and Alexander Smola. "Deep Sets." *Advances in Neural Information Processing Systems* 30 (2017).

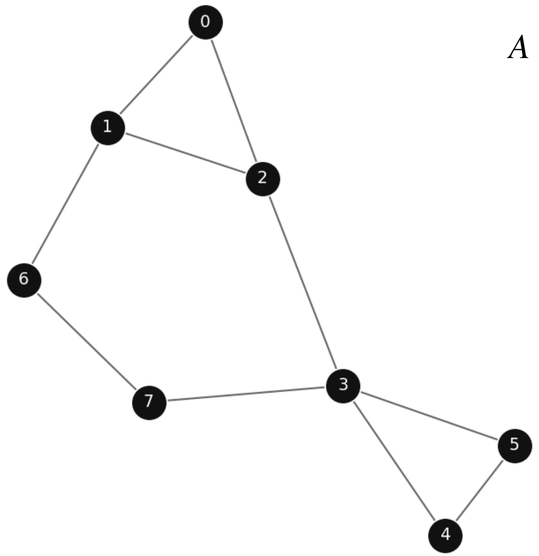
Learning on graphs: message-passing

Session 3 — Learning on Topological Data

Inés García Redondo — May 13, 2026

Including edges into the model

Including the adjacency matrix in the model



$$A \in \mathbb{R}^{n \times n}, \quad A_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Now we want to produce a function that also takes into account node features *and* edges: $f_\theta(X, A)$

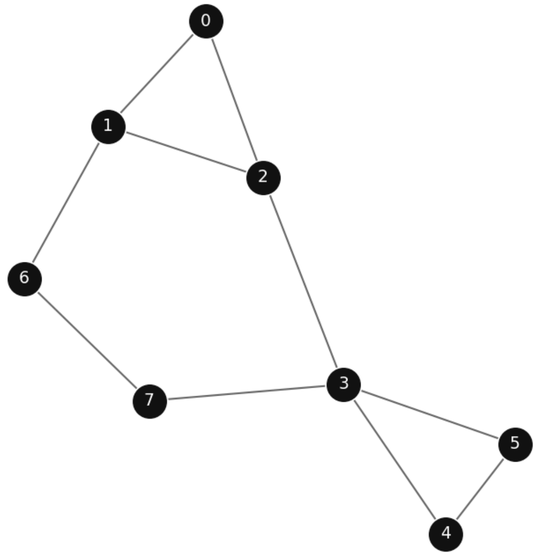
Note: we are not considering features over the edges, but this could be done

If we permute the nodes, the edges are permuted accordingly. This is reflected by transforming the adjacency matrix as $A \mapsto PAP^\top$

- **Permutation invariance:** $f_\theta(PX, PAP^\top) = f_\theta(X, A)$
- **Permutation equivariance:** $f_\theta(PX, PAP^\top) = P f_\theta(X, A)$

Locality on graphs

Neighborhoods



Neighborhood: $N_i = \{ j \in V \mid (i,j) \in E \} = \{ j \in V \mid A_{ij} = 1 \}$

Multiset of **features** in the neighborhood**:** $X_{N_i} = \{\{x_j : j \in N_i\}\}$

Local function: $g(x_i, X_{N_i})$

Similar to the aggregation rule in
a CNN from a kernel

A recipe for GNNs

Putting everything together

Permutation equivariant

A local function g is applied to all nodes and neighborhoods

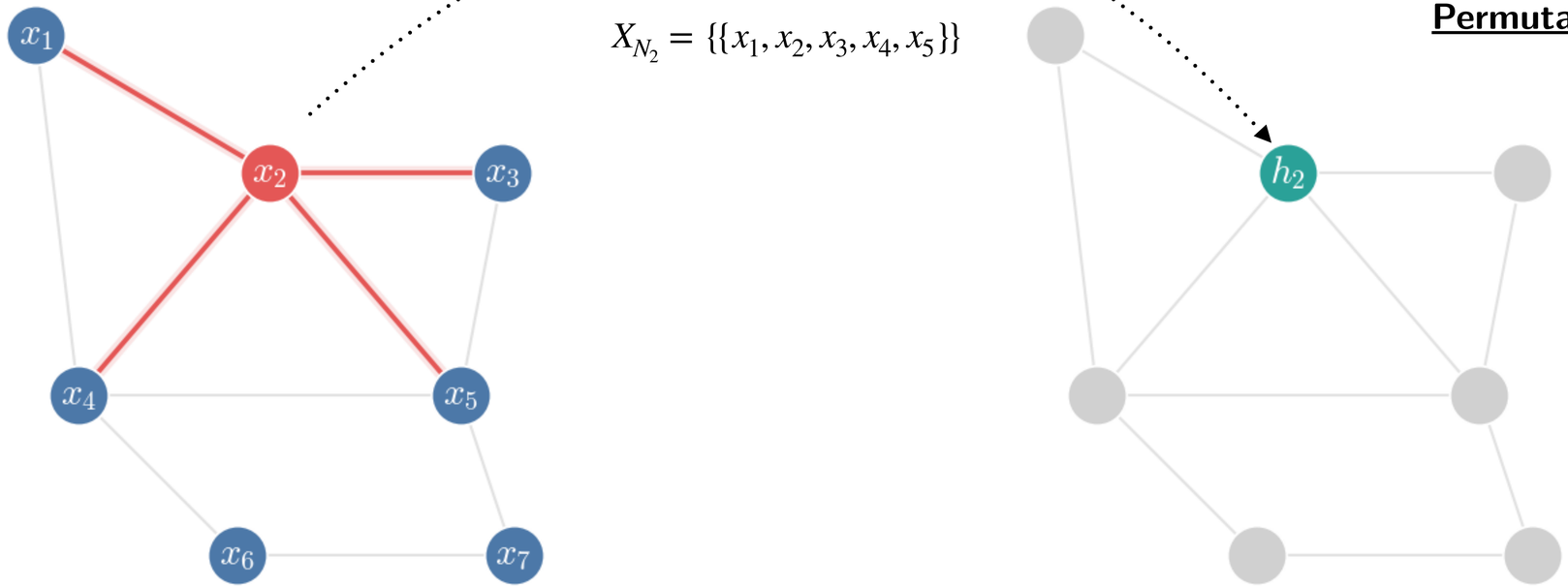
g should not depend on the order of vertices in X_{N_i}

$$f(X, A) = \begin{bmatrix} g(x_1, X_{N_1}) \\ g(x_2, X_{N_2}) \\ \vdots \\ g(x_n, X_{N_n}) \end{bmatrix}$$

Permutation invariant over X_{N_i}

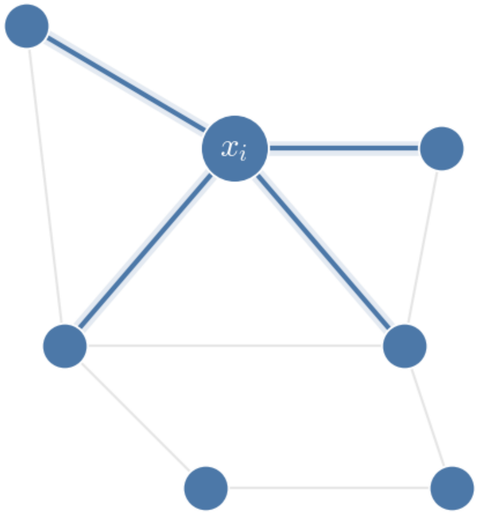
$h_2 = g(x_2, X_{N_2})$

$X_{N_2} = \{\{x_1, x_2, x_3, x_4, x_5\}\}$



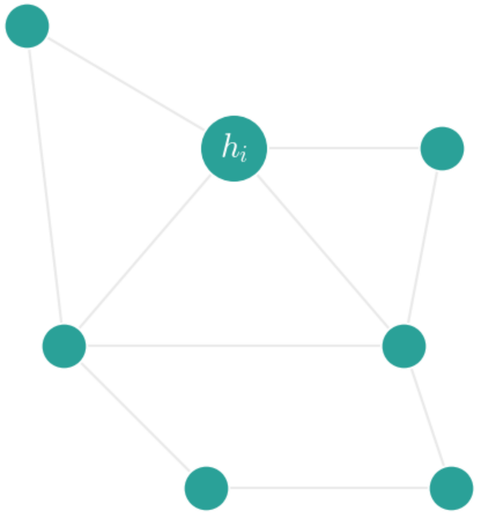
A recipe for GNNs

Solving different tasks



GNN layer

$$f(X, A) = \begin{bmatrix} g(x_1, X_{N_1}) \\ g(x_2, X_{N_2}) \\ \vdots \\ g(x_n, X_{N_n}) \end{bmatrix}$$



Node classification:

$$z_i = f(h_i)$$

Graph classification:

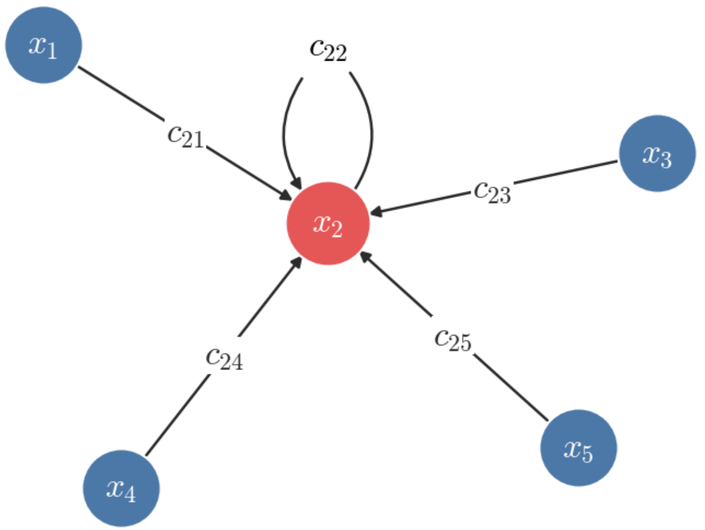
$$z_G = f\left(\bigoplus_{i \in V} h_i\right)$$

Link prediction:

$$z_{ij} = f(h_i, h_j, e_{ij})$$

Message-passing

Instantiating local functions



Convolutional

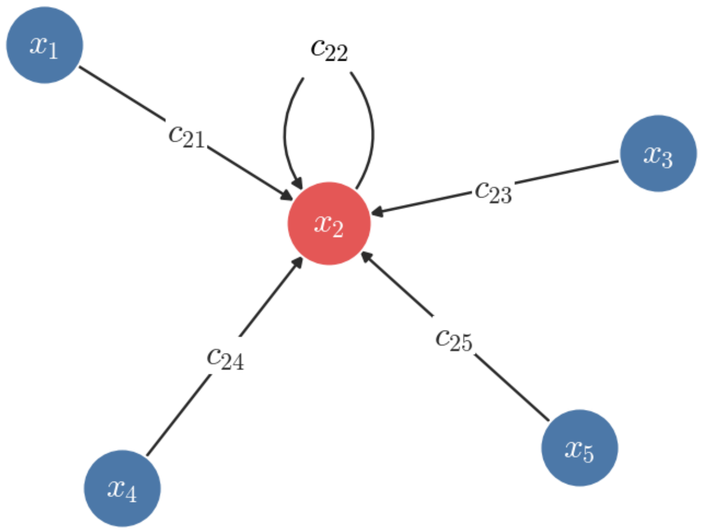
$$h_i = \phi\left(x_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij}\,\psi(x_j)\right)$$

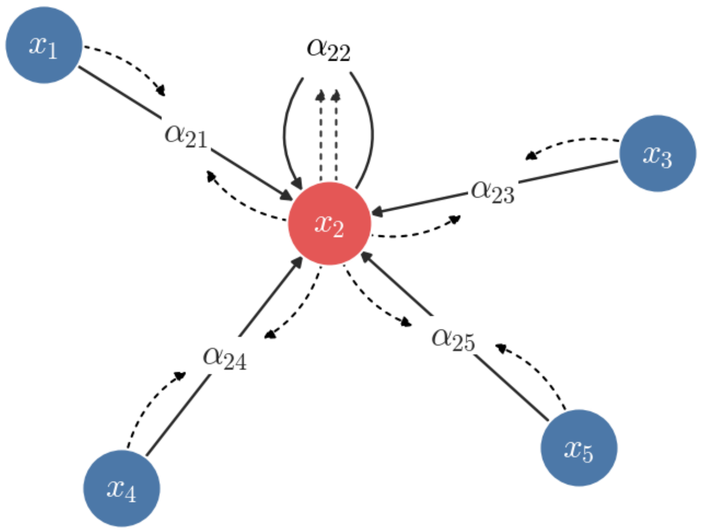
Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst. "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering." *Advances in Neural Information Processing Systems* 29 (2016).

Kipf, Thomas N., and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks." *International Conference on Learning Representations* (2017).

Message-passing

Instantiating local functions





Convolutional

$$h_i = \phi\left(x_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij}\, \psi(x_j)\right)$$

Attentional

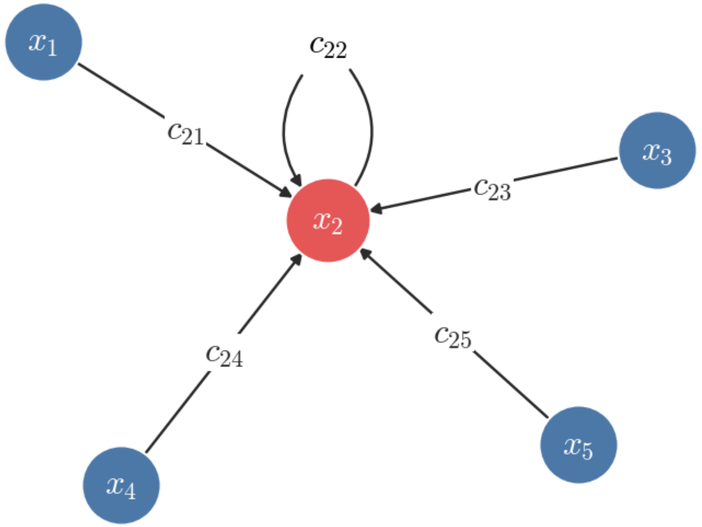
$$h_i = \phi\left(x_i, \bigoplus_{j \in \mathcal{N}_i} \alpha(x_i, x_j)\, \psi(x_j)\right)$$

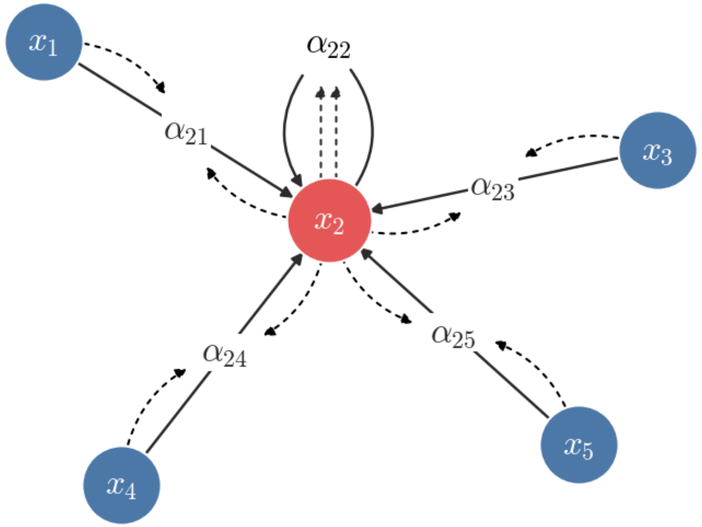
Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks." *International Conference on Learning Representations* (2018).

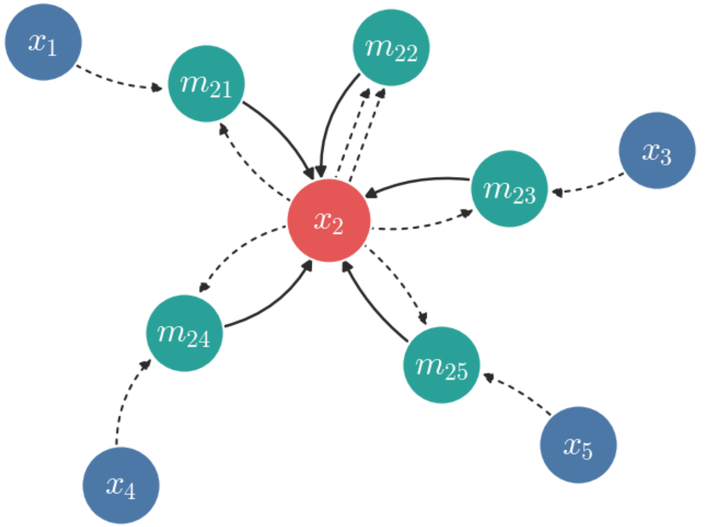
Monti, Federico, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M. Bronstein. "Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs." *IEEE Conference on Computer Vision and Pattern Recognition* (2017), 5115–24.

Message-passing

Instantiating local functions







Convolutional

$$h_i = \phi\left(x_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij}\,\psi(x_j)\right)$$

Attentional

$$h_i = \phi\left(x_i, \bigoplus_{j \in \mathcal{N}_i} \alpha(x_i, x_j)\,\psi(x_j)\right)$$

Message-passing

$$h_i = \phi\left(x_i, \bigoplus_{j \in \mathcal{N}_i} \psi(x_i, x_j)\right)$$

Gilmer, Justin, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. "Neural Message Passing for Quantum Chemistry." *Proceedings of the 34th International Conference on Machine Learning*, July 17, 2017, 1263–72.

Battaglia, Peter W., Jessica B. Hamrick, Victor Bapst, et al. "Relational Inductive Biases, Deep Learning, and Graph Networks." Battaglia, Peter W., Jessica B. Hamrick, Victor Bapst, et al. "Relational Inductive Biases, Deep Learning, and Graph Networks." arXiv:1806.01261. Preprint, arXiv, October 17, 2018.

MPNNs in practice

Useful libraries to train some GNNs

PyTorch Geometric (PyG)

(pyg.org)

- Built over PyTorch
- Largest collection of implemented GNN architectures (GCN, GAT, GIN, GraphSAGE, and 50+ more)
- Includes common benchmark datasets to test your code

NetworkX

(networkx.org)

- The standard library for graph analysis and visualization
- Integrates with PyG for graph construction

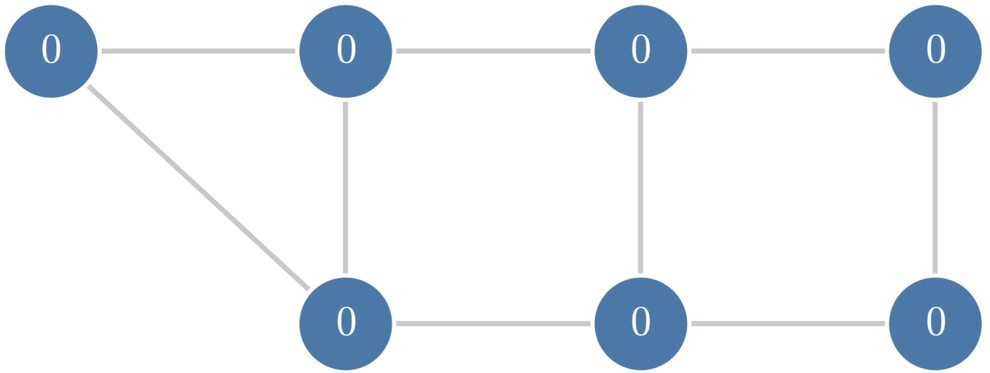
Expressivity: graph isomorphism testing

Session 3 — Learning on Topological Data

Inés García Redondo — May 13, 2026

Weisfeiler–Leman test

An algorithm for testing graph isomorphism



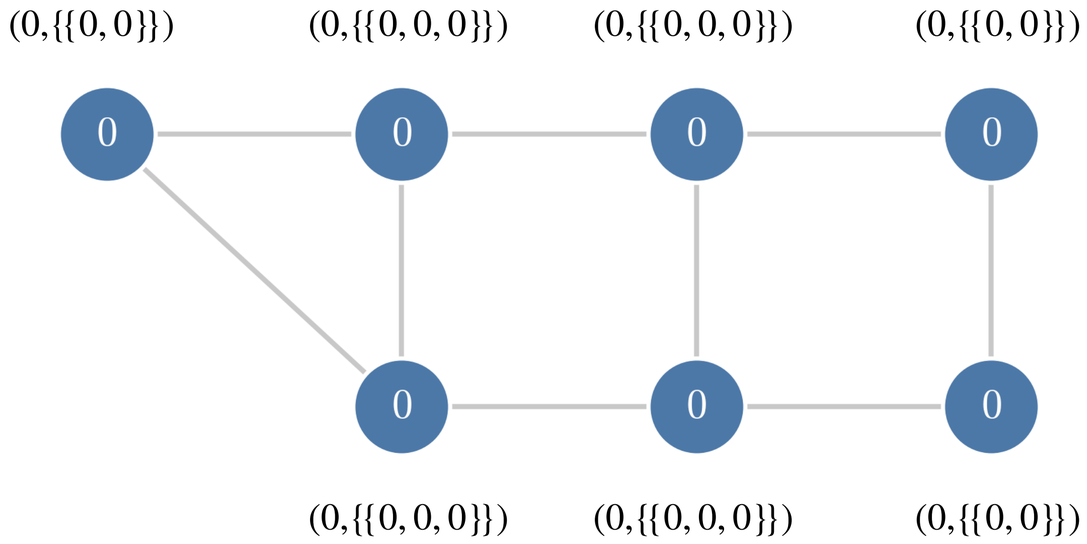
Initialization:

Assign the same label (or color) to all vertices in the graph

Weisfeiler, B. & Leman, A. (1968). "The reduction of a graph to canonical form and the algebra which appears therein". *Nauchno-Technicheskaya Informatsia*, Series 2, 9, pp. 12–16.

Weisfeiler–Leman test

An algorithm for testing graph isomorphism



Iterative step:

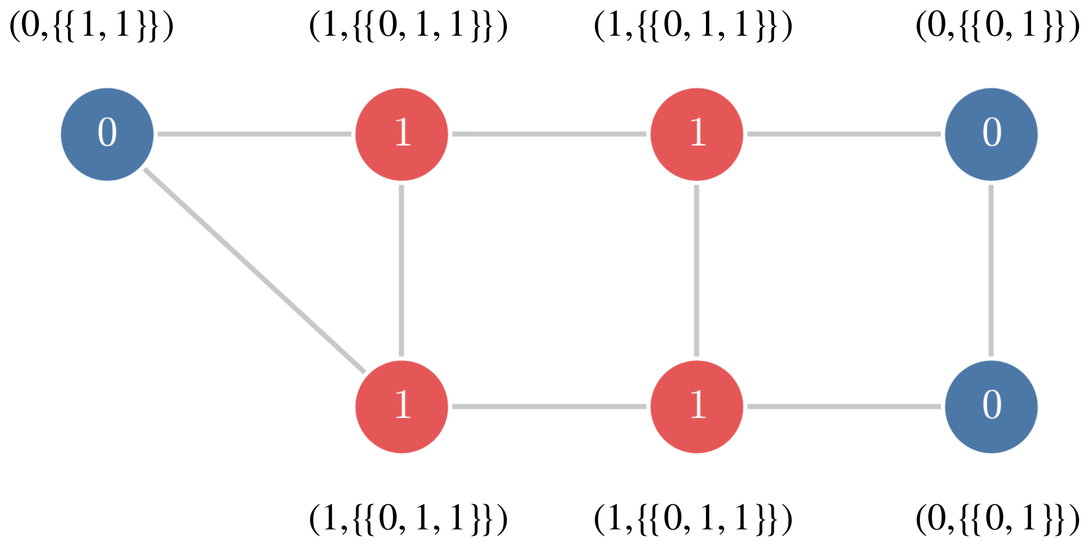
Compute a new label by applying a hash function to the current label and the labels of its neighbors

Weisfeiler, B. & Leman, A. (1968). "The reduction of a graph to canonical form and the algebra which appears therein". *Nauchno-Technicheskaya Informatsia*, Series 2, 9, pp. 12–16.

Weisfeiler–Leman test

An algorithm for testing graph isomorphism

Step 1



Iterative step:

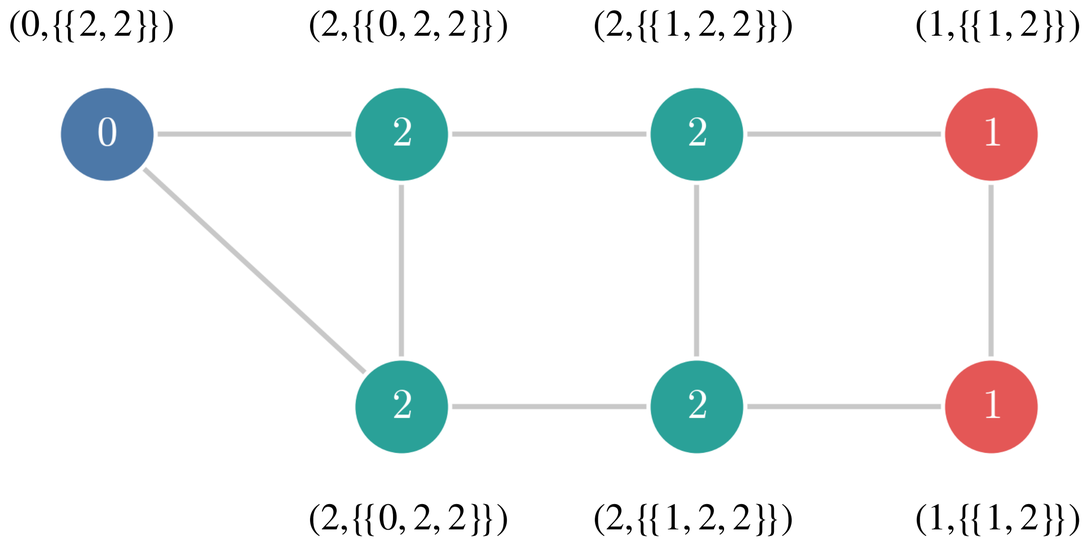
Compute a new label by applying a hash function to the current label and the labels of its neighbors

Weisfeiler, B. & Leman, A. (1968). "The reduction of a graph to canonical form and the algebra which appears therein". *Nauchno-Technicheskaya Informatsia*, Series 2, 9, pp. 12–16.

Weisfeiler–Leman test

An algorithm for testing graph isomorphism

Step 2



Iterative step:

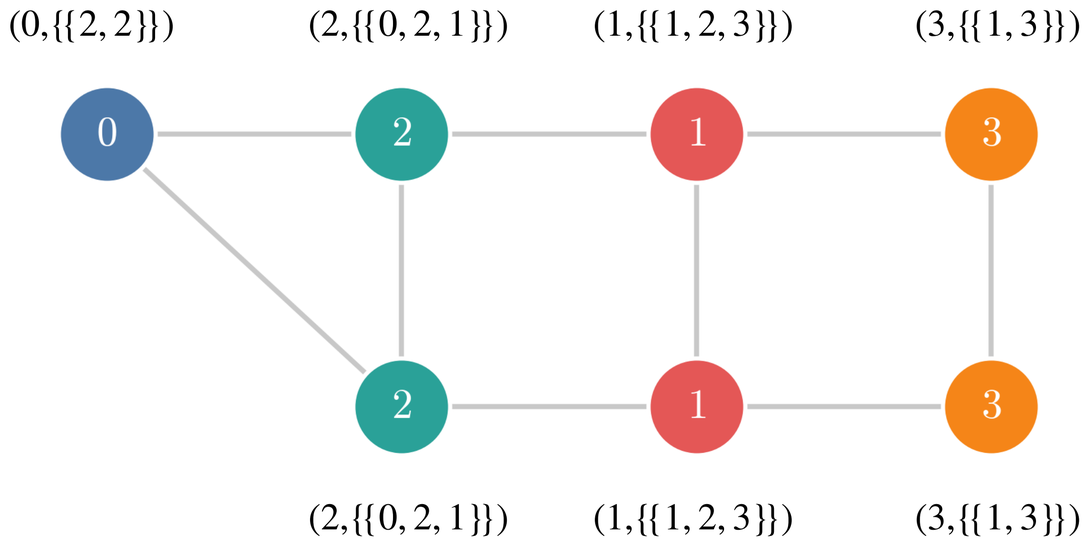
Compute a new label by applying a hash function to the current label and the labels of its neighbors

Weisfeiler, B. & Leman, A. (1968). "The reduction of a graph to canonical form and the algebra which appears therein". *Nauchno-Technicheskaya Informatsia*, Series 2, 9, pp. 12–16.

Weisfeiler–Leman test

An algorithm for testing graph isomorphism

Step 3



Iterative step:

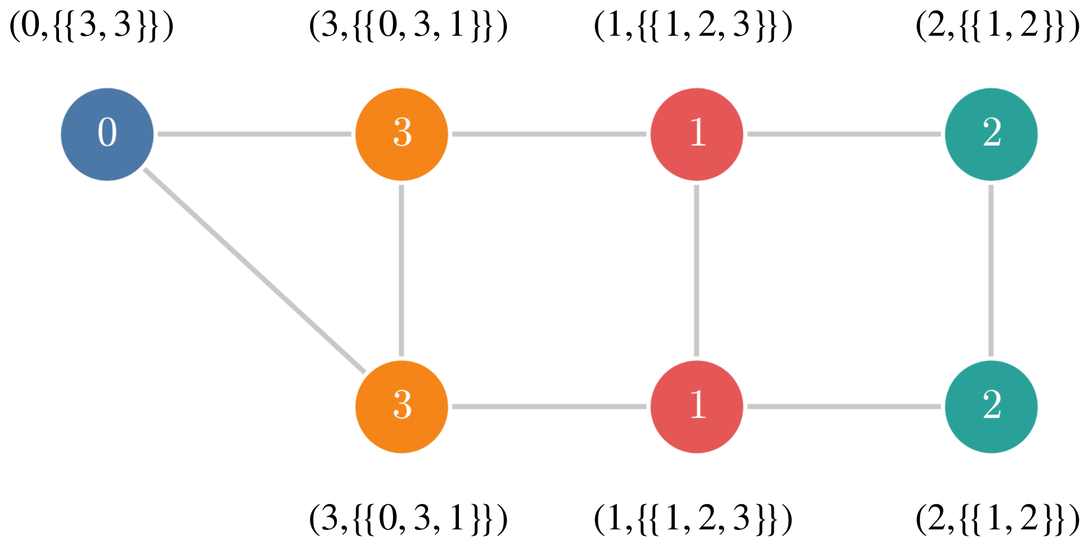
Compute a new label by applying a hash function to the current label and the labels of its neighbors

Weisfeiler, B. & Leman, A. (1968). "The reduction of a graph to canonical form and the algebra which appears therein". *Nauchno-Technicheskaya Informatsia*, Series 2, 9, pp. 12–16.

Weisfeiler–Leman test

An algorithm for testing graph isomorphism

Step 4



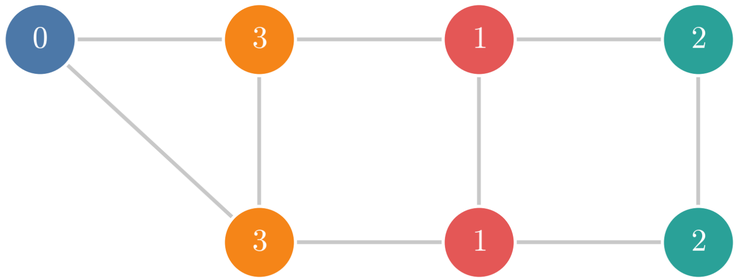
Stopping condition:

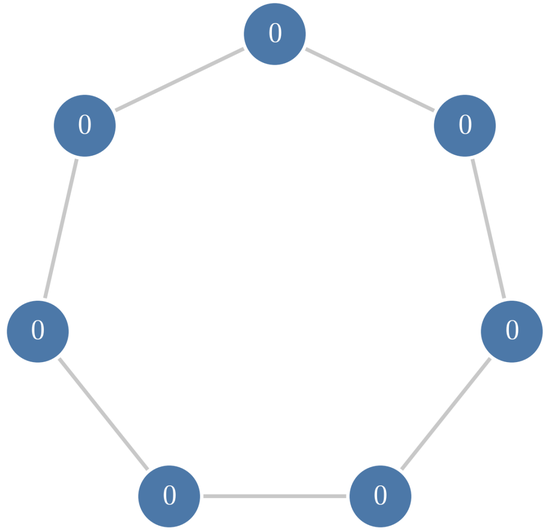
Stop the algorithm when we have converged to a partition of the nodes that does not change in an iteration

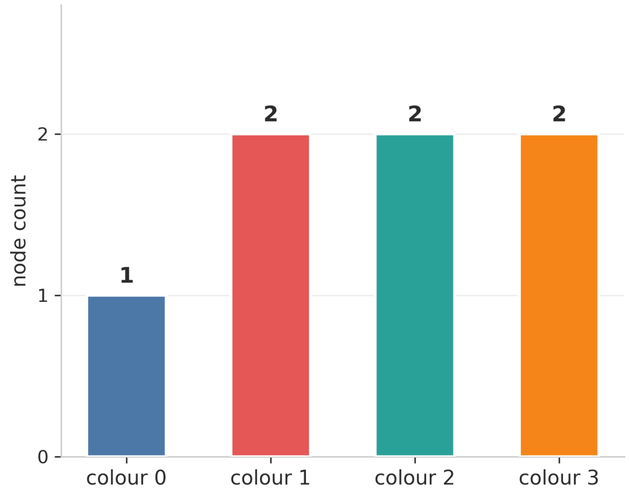
Weisfeiler, B. & Leman, A. (1968). "The reduction of a graph to canonical form and the algebra which appears therein". *Nauchno-Technicheskaya Informatsia*, Series 2, 9, pp. 12–16.

Weisfeiler–Leman test

An algorithm for testing graph isomorphism

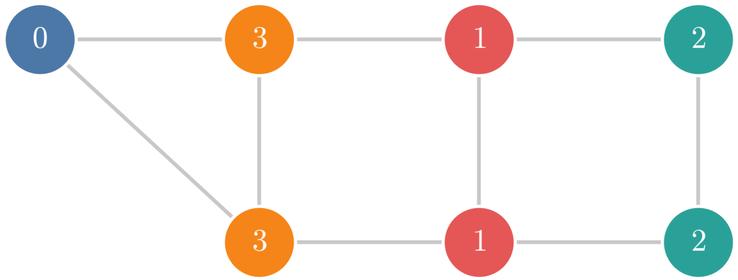


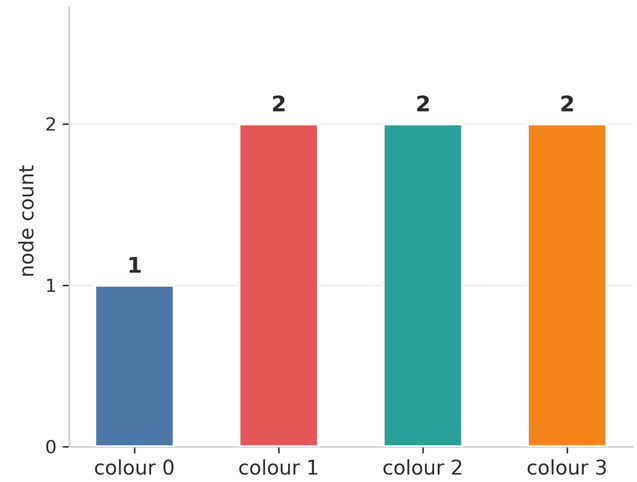


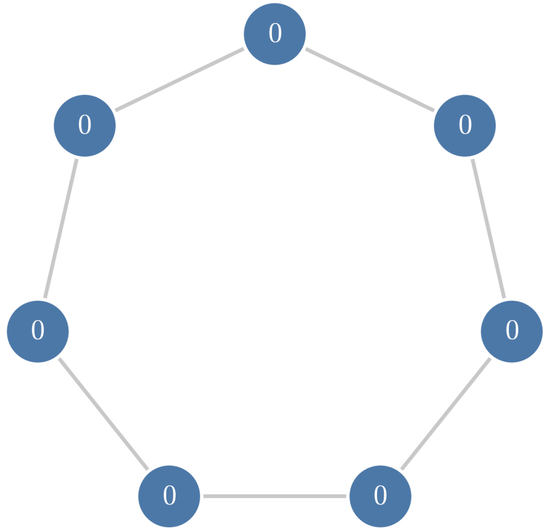


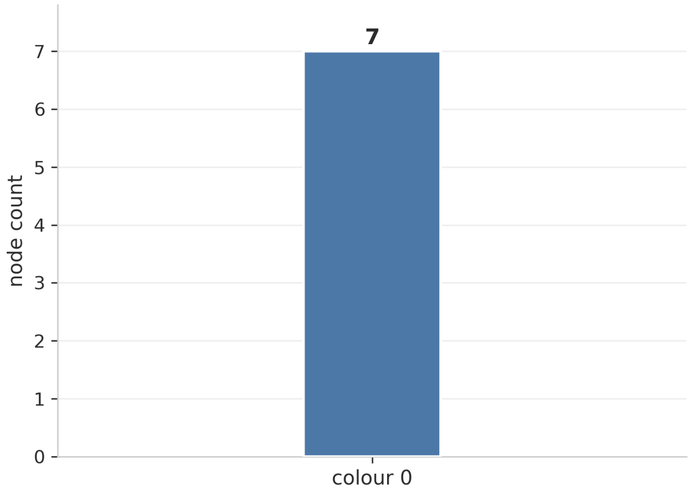
Weisfeiler–Leman test

An algorithm for testing graph isomorphism



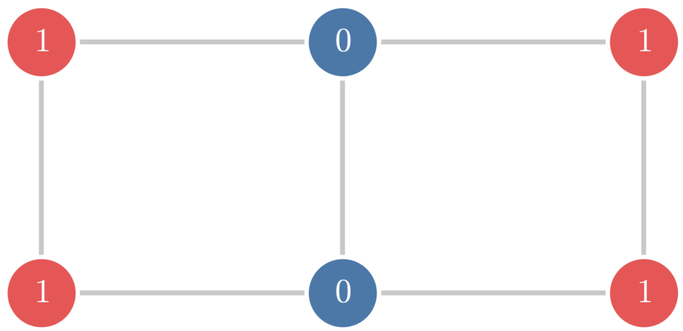


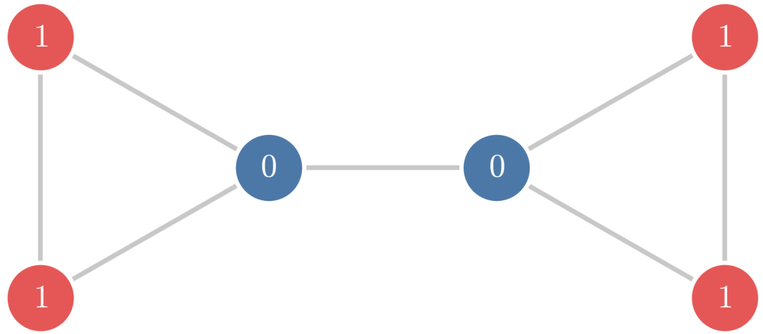


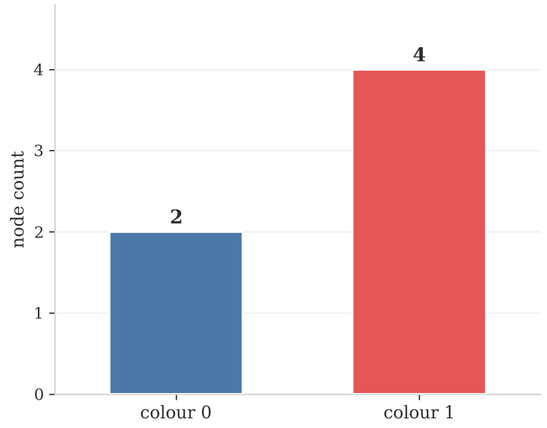


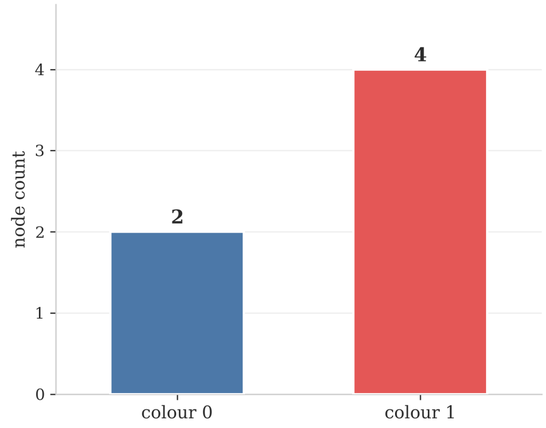
Weisfeiler–Leman test

An algorithm for testing graph isomorphism









Expressivity of message-passing GNNs

How powerful are GNNs?

With a GNN, we want that at least is able to distinguish non-isomorphic graphs, that is: that it produces different embeddings $\mathbf{h}^{G_1} \neq \mathbf{h}^{G_2}$, $i \in V$ for two non-isomorphic graphs $G_1 \ncong G_2$

We can achieve this thanks to the natural connection of message-passing with the WL test (applying an untrained GNN is similar in nature to running the WL algorithm)

Over discrete features, **message-passing GNNs can only be as powerful as the WL test**

You can reach maximum expressivity using an injective aggregator (e.g. the sum) in message-passing

You can try to improve the expressive power of GNNs by looking at the failure modes of the WL test

Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. "How Powerful Are Graph Neural Networks?" *International Conference on Learning Representations* (2018).

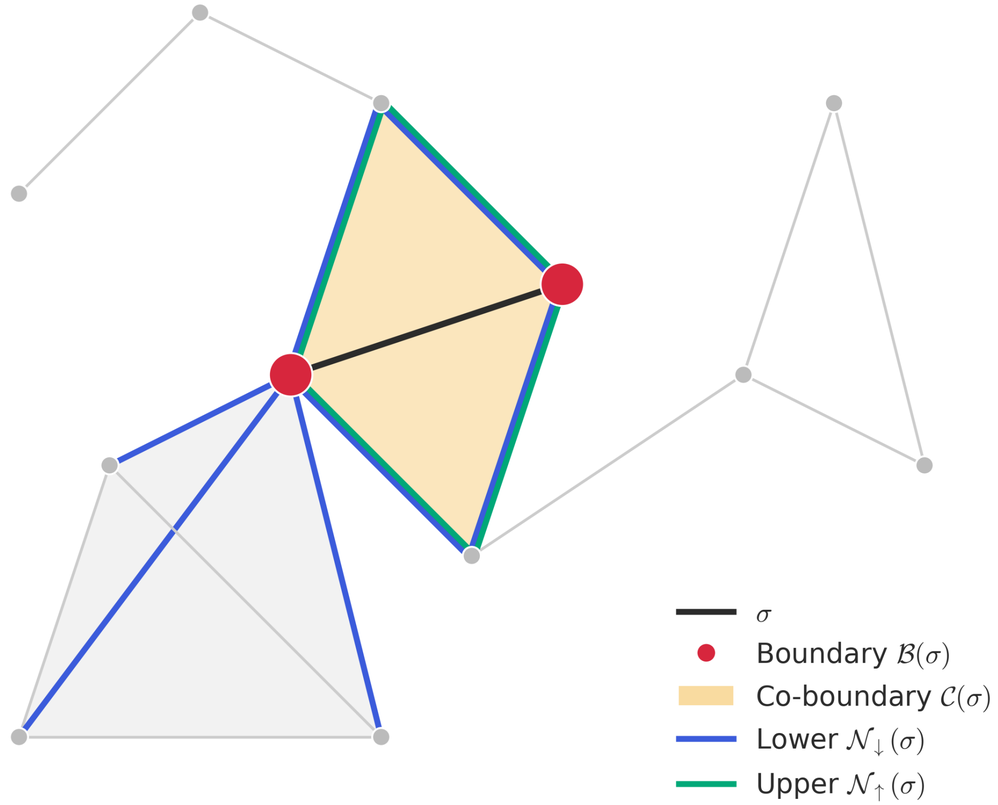
Learning on higher-order domains

Session 3 — Learning on Topological Data

Inés García Redondo — May 13, 2026

Weisfeiler and Leman go topological

Including adjacencies



Face relation:

$\sigma \preceq \tau \iff \sigma \subseteq \tau$

$\sigma \prec \tau \iff \sigma \subset \tau$ and there is no δ such that $\sigma \subset \delta \subset \tau$

Adjacencies in a simplicial complex:

- Boundaries: $\mathcal{B}(\sigma) = \{\tau \in X : \tau \prec \sigma\}$

  Non-zero rows of the boundary operator B_k

- Co-boundaries: $\mathcal{C}(\sigma) = \{\tau \in X : \sigma \prec \tau\}$

  Non-zero rows of $B_{k+1}^{\top}$

- Lower adjacencies: $\mathcal{N}_{\downarrow}(\sigma) = \{\tau \in X : \exists \delta \in X \text{ s.t. } \delta \prec \tau \text{ and } \delta \prec \sigma\}$

  From the down Laplacian $L_k^{\text{down}} = B_k^{\top} B_k$

- Upper adjacencies: $\mathcal{N}_{\uparrow}(\sigma) = \{\tau \in X : \exists \delta \in X \text{ s.t. } \tau \prec \delta \text{ and } \sigma \prec \delta\}$

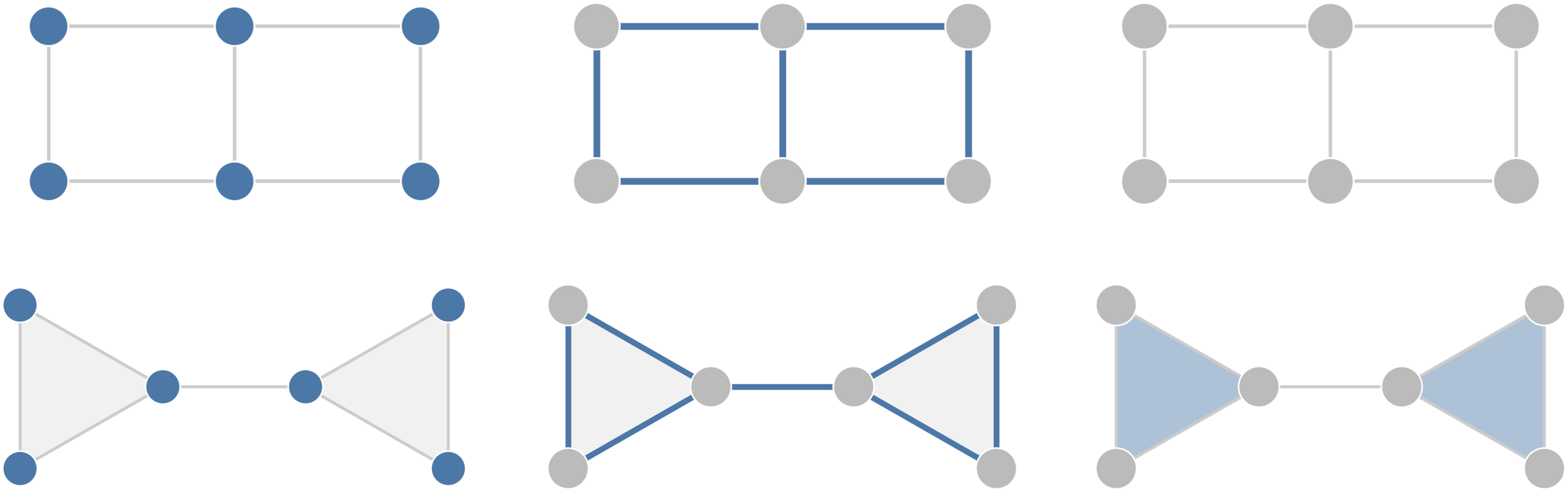
  From the up Laplacian $L_k^{\text{up}} = B_{k+1} B_{k+1}^{\top}$

Bodnar, Cristian, Fabrizio Frasca, Yuguang Wang, et al. "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks." *Proceedings of the 38th International Conference on Machine Learning*, July 1, 2021, 1026–37.

Weisfeiler and Leman go topological

Simplicial WL algorithm

Initialization



Bodnar, Cristian, Fabrizio Frasca, Yuguang Wang, et al. "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks." *Proceedings of the 38th International Conference on Machine Learning*, July 1, 2021, 1026–37.

Weisfeiler and Leman go topological

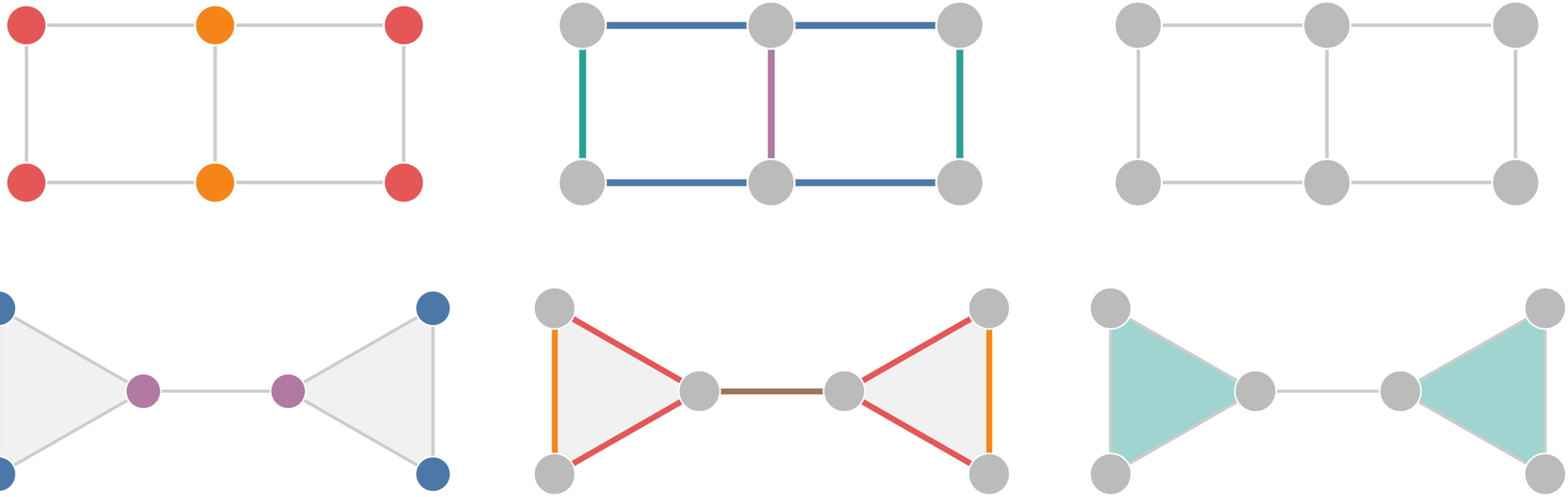
Simplicial WL algorithm

In fact, it is enough to look at the boundary and the upper adjacencies

$$c^t_{\mathcal{B}}(\sigma) = \{\{c^t_\tau : \tau \in \mathcal{B}(\sigma)\}\}$$

$$c^t_{\uparrow}(\sigma) = \{\{(c^t_\tau, c^t_{\tau\cup\sigma}) : \tau \in \mathcal{N}_{\uparrow}(\sigma)\}\}$$

Step 1

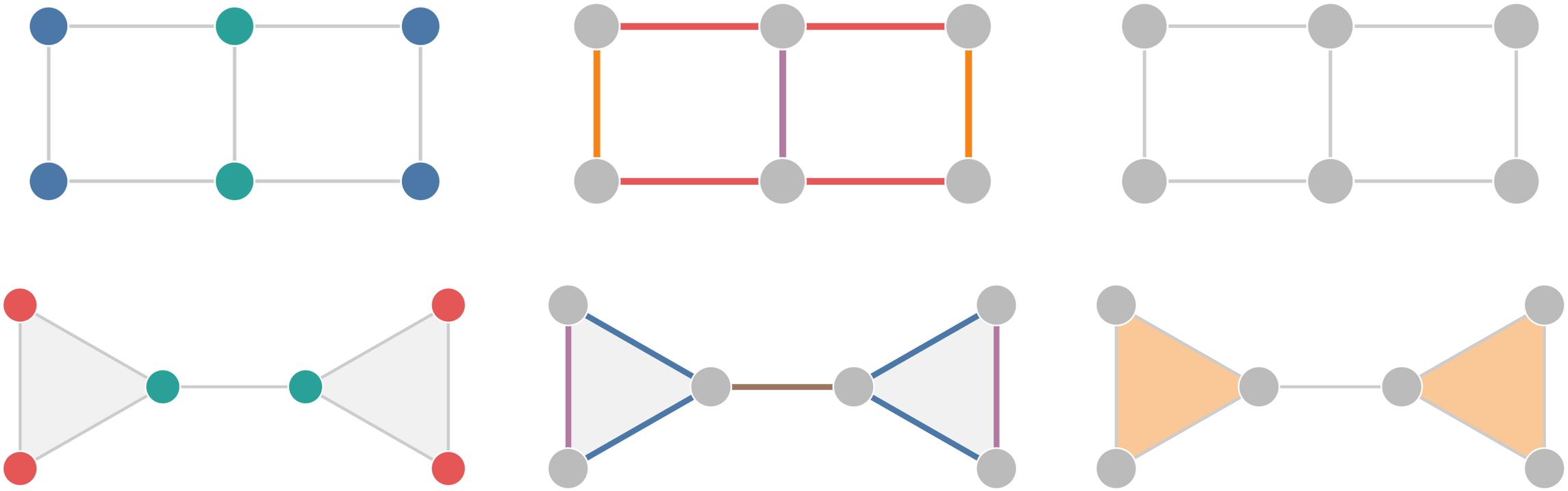


Bodnar, Cristian, Fabrizio Frasca, Yuguang Wang, et al. "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks." *Proceedings of the 38th International Conference on Machine Learning*, July 1, 2021, 1026–37.

Weisfeiler and Leman go topological

Simplicial WL algorithm

Step 2: convergence



Bodnar, Cristian, Fabrizio Frasca, Yuguang Wang, et al. "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks." *Proceedings of the 38th International Conference on Machine Learning*, July 1, 2021, 1026–37.

Weisfeiler and Leman go topological

Expressiveness of SWL and message-passing simplicial NN

SWL is not less powerful than 3-WL, which is strictly more expressive than WL

Message-passing Simplicial Neural Networks:

$$m_{\mathscr{B}}^{t+1}(\sigma) = \bigoplus_{\tau \in \mathscr{B}(\sigma)} \psi(h_\sigma^t, h_\tau^t) \qquad m_{\mathscr{C}}^{t+1}(\sigma) = \bigoplus_{\tau \in \mathscr{C}(\sigma)} \psi(h_\sigma^t, h_\tau^t) \qquad m_{\uparrow}^{t+1}(\sigma) = \bigoplus_{\tau \in \mathscr{N}_{\uparrow}(\sigma)} \psi(h_\sigma^t, h_\tau^t, h_{\tau\cup\sigma}^t) \qquad m_{\downarrow}^{t+1}(\sigma) = \bigoplus_{\tau \in \mathscr{N}_{\downarrow}(\sigma)} \psi(h_\sigma^t, h_\tau^t, h_{\tau\cap\sigma}^t)$$

$$h_\sigma^{t+1} = \phi\left(h_\sigma^t, m_{\mathscr{B}}^{t+1}(\sigma), m_{\mathscr{C}}^{t+1}(\sigma), m_{\uparrow}^{t+1}(\sigma), m_{\downarrow}^{t+1}(\sigma)\right)$$

MPSNs with sufficient layers and injective neighborhood aggregators are as powerful as SWL

Bodnar, Cristian, Fabrizio Frasca, Yuguang Wang, et al. "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks." *Proceedings of the 38th International Conference on Machine Learning*, July 1, 2021, 1026–37.

Weisfeiler and Leman go topological

Expressiveness of SWL and message-passing simplicial NN

SWL is not less powerful than 3-WL, which is strictly more expressive than WL

Message-passing Simplicial Neural Networks:

$$m_{\mathscr{B}}^{t+1}(\sigma) = \bigoplus_{\tau \in \mathscr{B}(\sigma)} \psi(h_\sigma^t, h_\tau^t) \qquad m_{\mathscr{C}}^{t+1}(\sigma) = \bigoplus_{\tau \in \mathscr{C}(\sigma)} \psi(h_\sigma^t, h_\tau^t) \qquad m_{\uparrow}^{t+1}(\sigma) = \bigoplus_{\tau \in \mathscr{N}_{\uparrow}(\sigma)} \psi(h_\sigma^t, h_\tau^t, h_{\tau\cup\sigma}^t) \qquad m_{\downarrow}^{t+1}(\sigma) = \bigoplus_{\tau \in \mathscr{N}_{\downarrow}(\sigma)} \psi(h_\sigma^t, h_\tau^t, h_{\tau\cap\sigma}^t)$$

$$h_\sigma^{t+1} = \phi\left(h_\sigma^t, m_{\mathscr{B}}^{t+1}(\sigma), m_{\mathscr{C}}^{t+1}(\sigma), m_{\uparrow}^{t+1}(\sigma), m_{\downarrow}^{t+1}(\sigma)\right)$$

MPSNs with sufficient layers and injective neighborhood aggregators are as powerful as SWL

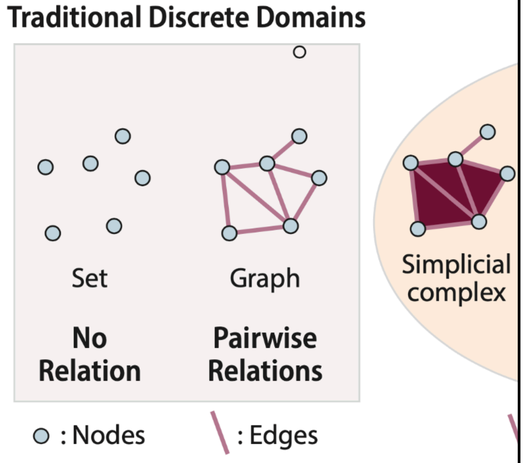
Bodnar, Cristian, Fabrizio Frasca, Yuguang Wang, et al. "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks." *Proceedings of the 38th International Conference on Machine Learning*, July 1, 2021, 1026–37.

SMPNNs in practice

TopoX

TopoNetX (TNX)

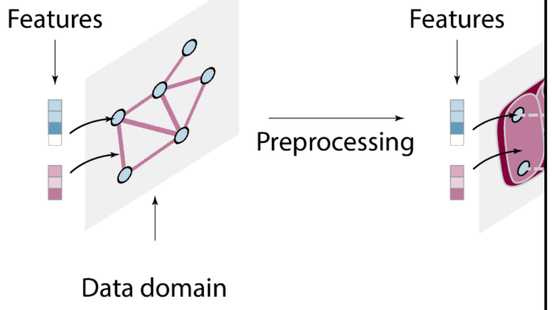
`TopoNetX` is a Python package for computing on topological domains. Topological domains are the natural mathematical structures representing relations between the components of a dataset.



https://pyt-team.github.io/index.html

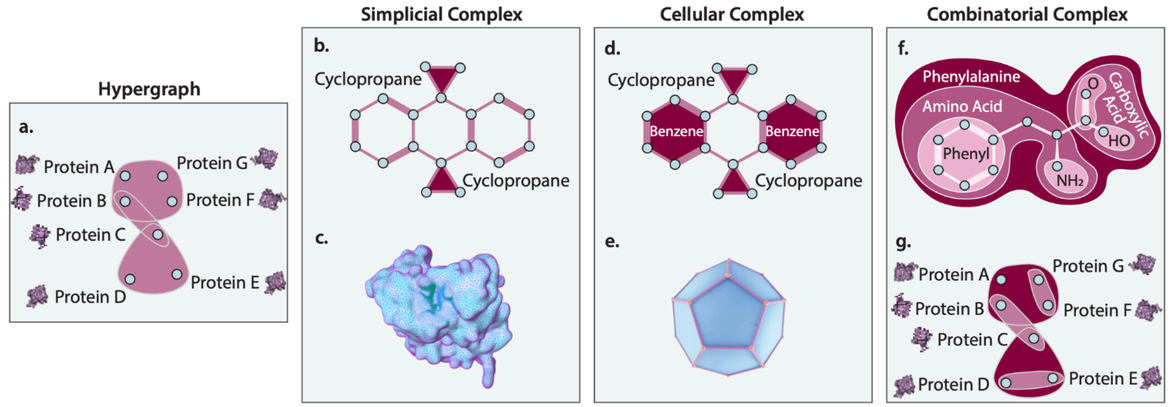
TopoModelX (TMX)

TopoModelX is a Python package for Topological Deep Learning, i.e. Deep Learning on Topological Domains.



TopoEmbedX (TEX)

`TopoEmbedX` (TEX) is a Python package for representation Learning on Topological Domains. Topological domains are the natural mathematical structures representing relations between the components of a dataset.



Papillon, Mathilde, Sophia Sanborn, Mustafa Hajij, and Nina Miolane. "Architectures of Topological Deep Learning: A Survey of Message-Passing Topological Neural Networks." (2024).

Hajij, Mustafa, Ghada Zamzmi, Theodore Papamarkou, et al. "Topological Deep Learning: Going Beyond Graph Data." (2023)

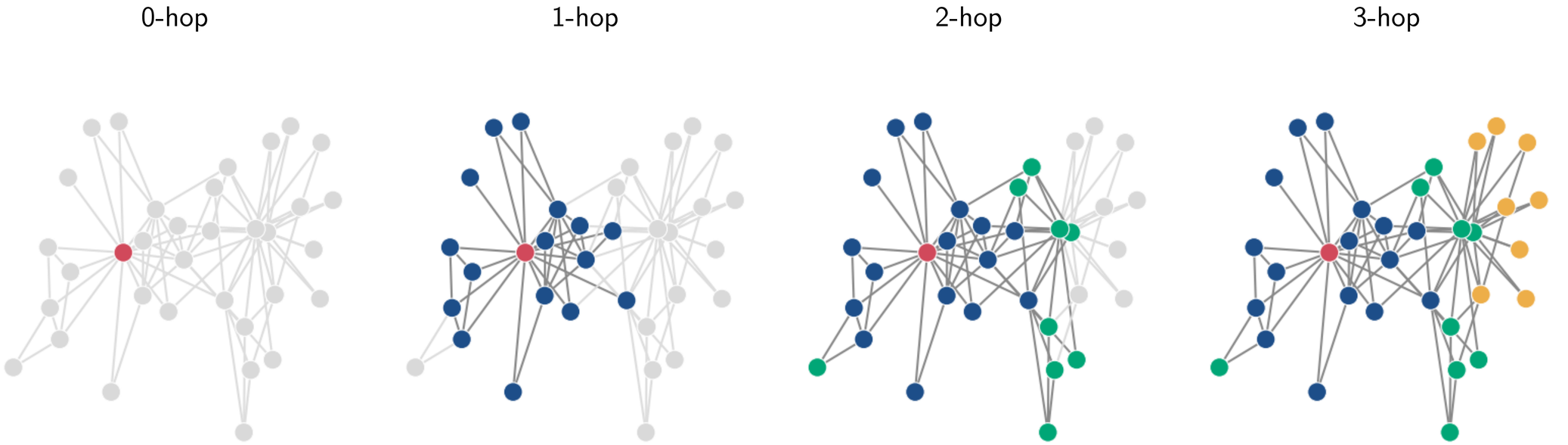
Limitations of message-passing: over-squashing and over-smoothing

Session 3 — Learning on Topological Data

Inés García Redondo — May 13, 2026

Receptive field in MPNNs

With which nodes can I communicate at each layer?

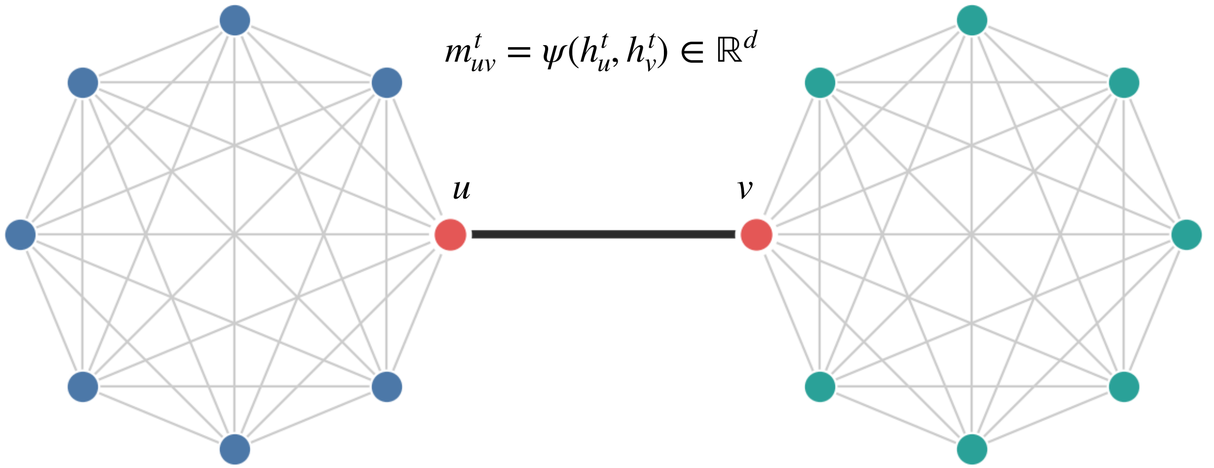


The receptive field grows exponentially with depth

To access information from nodes at distance k, we need at least k layers in our MPNN

Over-squashing

The problem of bottlenecks in message-passing



At layer t, each node receives messages from 2^t nodes

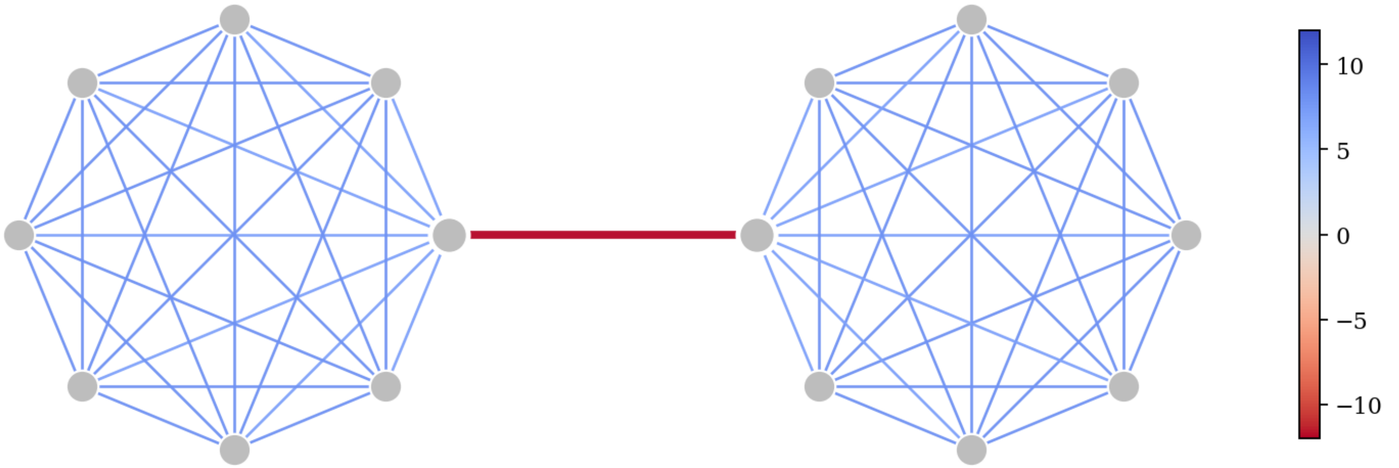
To communicate between the hubs, all the information is compressed in a singe message of fixed size

As a result, GNNs fail to propagate messages originating from distant nodes and perform poorly when the prediction task depends on long-range interaction

Alon, Uri, and Eran Yahav. "On the Bottleneck of Graph Neural Networks and Its Practical Implications." I*nternational Conference on Learning Representations* (2020).

Over-squashing

Characterizing over-squashing via curvature



Negatively curved edges are those causing the graph bottleneck and thus leading to the over-squashing phenomenon

Derive concrete bounds for $\left\| \frac{\partial h_u^t}{\partial x_v} \right\|$ as a proxy for over-squashing

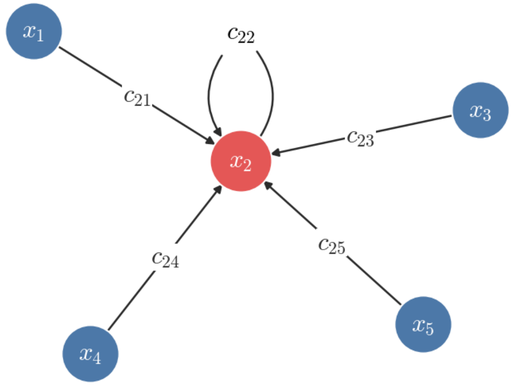
Propose a rewiring technique based on the curvature to address this problem

Topping, Jake, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. "Understanding Over-Squashing and Bottlenecks on Graphs via Curvature." *International Conference on Learning Representations* (2021).

Forman. "Bochner's Method for Cell Complexes and Combinatorial Ricci Curvature." *Discrete & Computational Geometry* 29, no. 3 (2003): 323–74.

Over-smoothing

GCNNs revisited



Update rule of GCN[1]:

$$h_i^{t+1} = \phi\left(h_i^t, \bigoplus_{j\in\mathcal{N}_i} c_{ij}\,\psi(h_j^t)\right) = \sum_{j\in\mathcal{N}_i\cup\{i\}} (d_i+1)^{-1/2}(d_j+1)^{-1/2}\,h_j^t$$

$\phi = \mathrm{Id}$

$c_{ij} = (d_i+1)^{-1/2}(d_j+1)^{-1/2}$

$\bigoplus = \sum$

Augmented adjacency

$$H^{t+1} = D^{-1/2}(A+I)D^{-1/2}H^t$$

Normalized Graph Laplacian

$$\begin{aligned}\hat{L} &= D^{-1/2}LD^{-1/2}\\ &= D^{-1/2}(D-A)D^{-1/2}\\ &= I - D^{-1/2}AD^{-1/2}\end{aligned}$$

Spectrum in [0,2]

Express a signal over the vertices in the eigenbasis:

$\{\Phi_i\}_{i=1}^n \qquad 0 = \lambda_1 \le \lambda_2 \le \ldots \le \lambda_n \le 2$

$h = \sum_{i=1}^n a_i \Phi_i \in \mathbb{R}^n$ (Scalar signal over each node)

Apply the update rule:

$$h^{t+1} = (I-\hat{L})h^t = \sum_{i=1}^n (1-\lambda_i)^t\, a_i \Phi_i$$

$-1 \le 1-\lambda_i \le 0$

All components except for $\lambda_1 = 0$ decay to 0

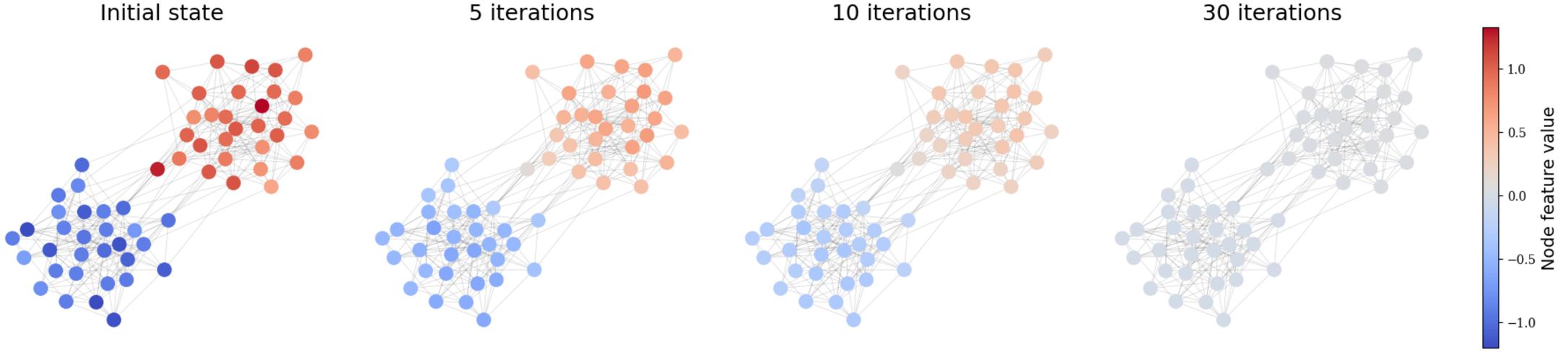
[1]Kipf, Thomas N., and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks." *International Conference on Learning Representations* (2017).

Over-smoothing

Representation collapse

Initialize features in two clusters with values $\{1, -1\}$ + some noise, and let the features propagate with rule

$$H^{t+1} = D^{-1/2}(A + I)D^{-1/2}H^t$$



The tension between over-smoothing and over-squashing

We need many layers for information propagation (even if it is imperfect and subject to bottlenecks)

But too many layers make our representations collapse

How do we strike a balance?

Not a clear solution to these limitations in the literature...

Further thoughts on message-passing

Session 3 — Learning on Topological Data

Inés García Redondo — May 13, 2026

Are GNNs learning 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* (2024).

Are the pairwise interactions even 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)

Are our datasets 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

Final conclusions

Maybe we need to rethink our models and paradigms...

We need to better understand what message-passing is really learning and using from the graph structure

We need to have benchmarks that we trust and where we know what features are important for the task at hand

Coupette, Corinna, Jeremy Wayland, Emily Simons, and Bastian Rieck. "No Metric to Rule Them All: Toward Principled Evaluations of Graph-Learning Datasets." *International Conference of Machine Learning,* 2025.

Ballester, Rubén, Ernst Röell, Daniel Bin Schmid, et al. "MANTRA: The Manifold Triangulations Assemblage." Paper presented at The *Thirteenth International Conference on Learning Representations*. October 4, 2024.

Maggs, Kelly, Celia Hacker, and Bastian Rieck. "Simplicial Representation Learning with Neural k-Forms." Paper presented at *The Twelfth International Conference on Learning Representations*. October 13, 2023.

Thank you for your attention!

AIDOS LAB
AI FOR DATA-ORIENTED SCIENCE

UNI
FR

Session 3 — Learning on topological domains

Inés García Redondo — May 13, 2026