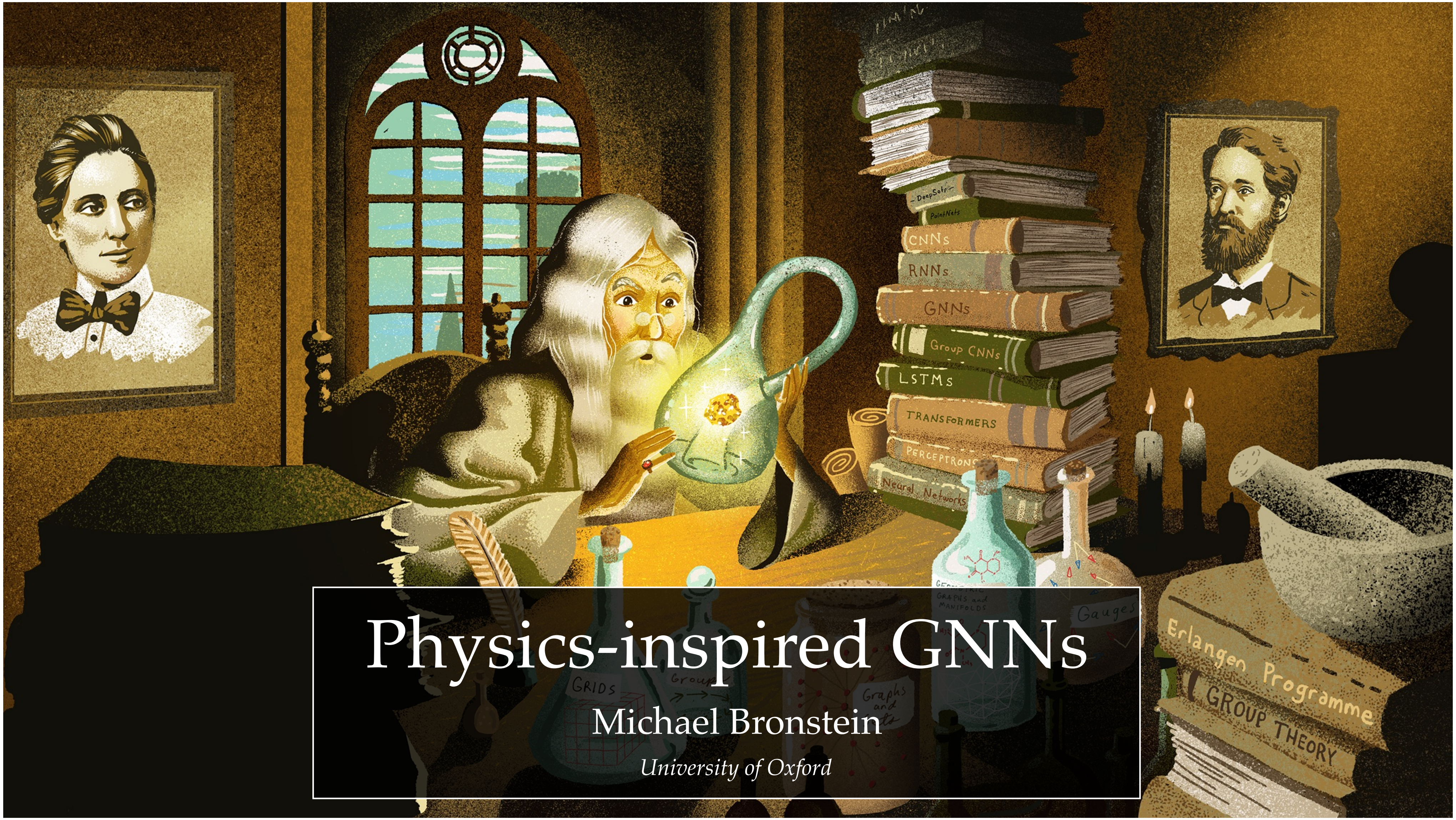


Physics-inspired GNNs

Michael Bronstein
University of Oxford

"Symmetry, as wide or as narrow as you may define its meaning, is one idea by which man through the ages has tried to comprehend and create order, beauty, and perfection"

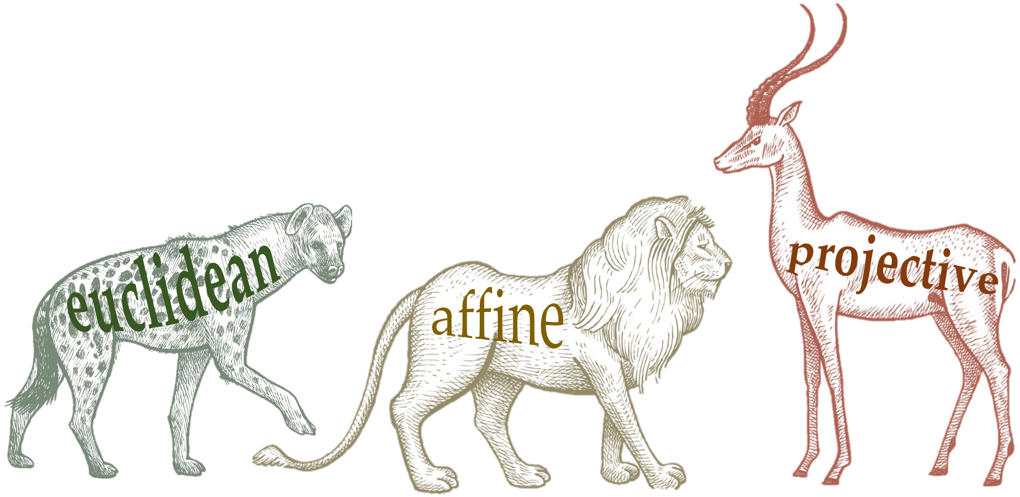
Hermann Weyl

Weyl 1952

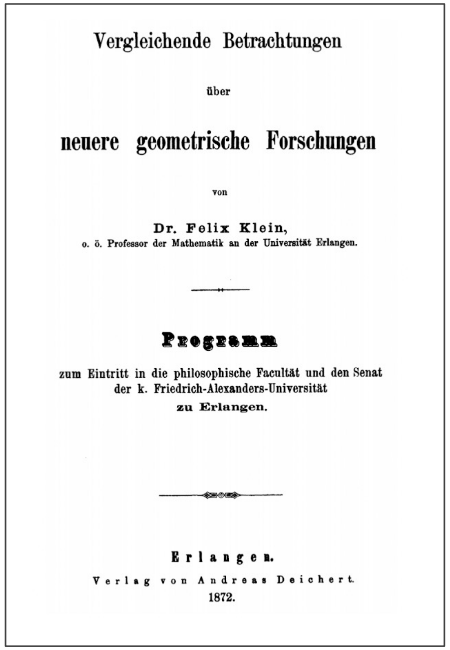
XIX century


euclidean
affine
projective

The Erlangen Programme



Geometry = space + transformation group



Vergleichende Betrachtungen

über

neuere geometrische Forschungen

von

Dr. Felix Klein,

o. ö. Professor der Mathematik an der Universität Erlangen.

Programm

zum Eintritt in die philosophische Facultät und den Senat der k. Friedrich-Alexanders-Universität zu Erlangen.

Erlangen.

Verlag von Andreas Deichert.

1872.

Felix Klein

Klein 1872

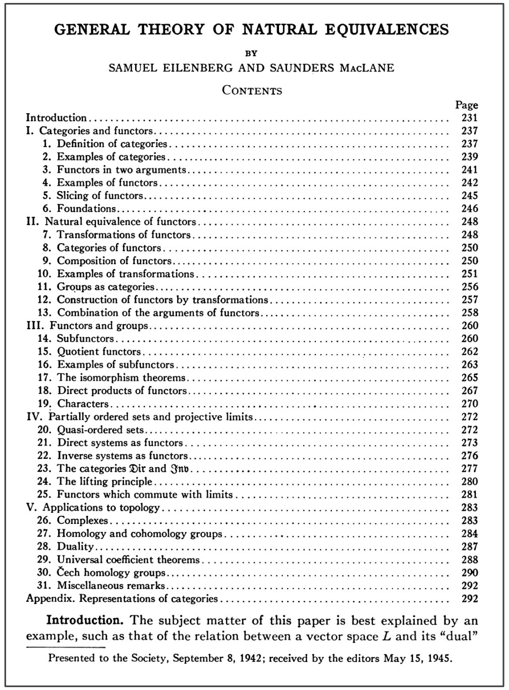
Cultural Impact in Mathematics

E. Beltrami

1868

E. Cartan

1920s



GENERAL THEORY OF NATURAL EQUIVALENCES

BY

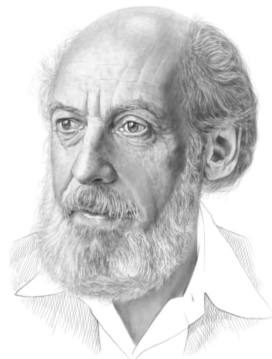
SAMUEL EILENBERG AND SAUNDERS MacLANE

CONTENTS

Introduction. The subject matter of this paper is best explained by an example, such as that of the relation between a vector space L and its "dual"

Presented to the Society, September 8, 1942; received by the editors May 15, 1945.

Category Theory



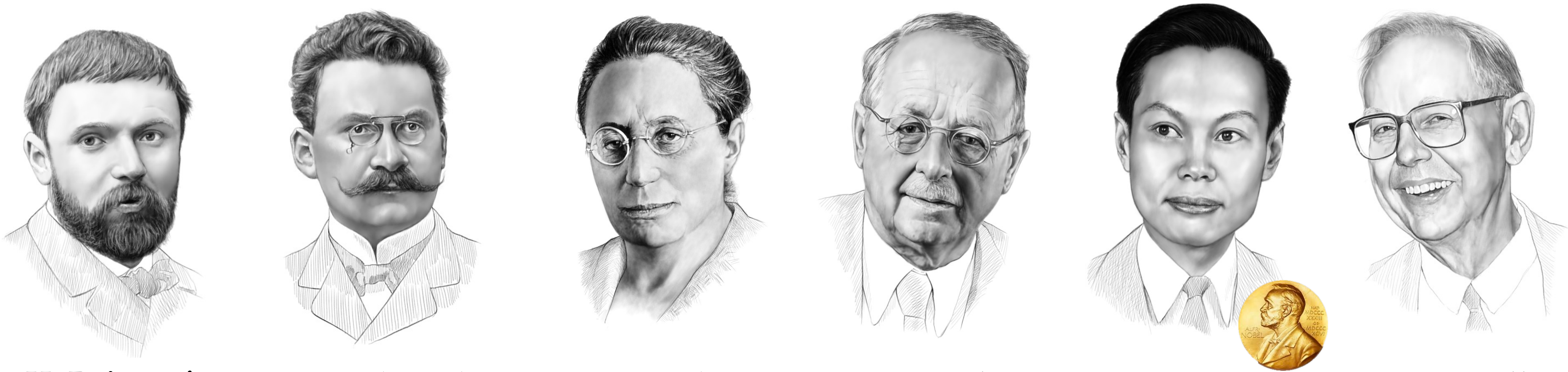
S. Eilenberg

S. Mac Lane

1945

Portraits: Ihor Gorskyi

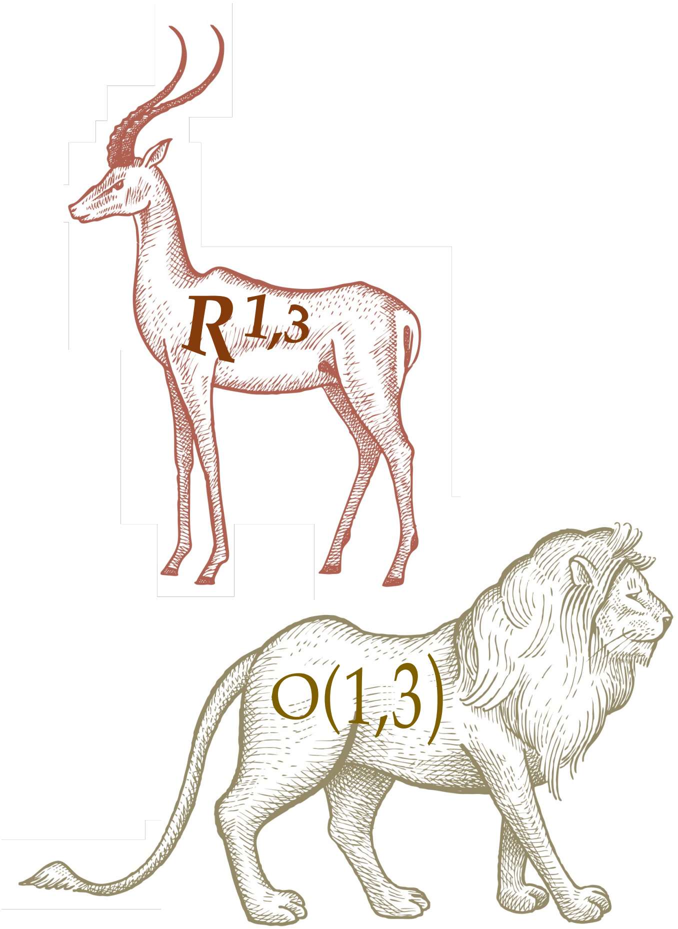
New Physics



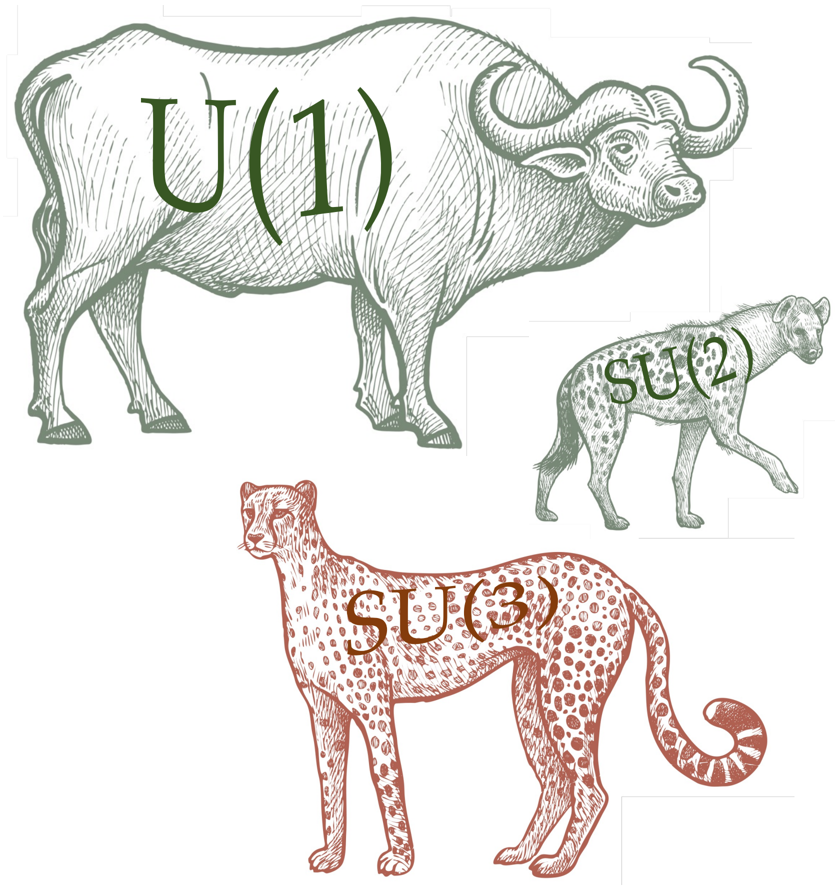
H. Poincaré | **H. Minkowski** | **E. Noether** | **H. Weyl** | **C. N. Yang** | **R. L. Mills**

1904 | 1907 | 1918 | 1929 | 1954

Poincaré 1904; Noether 1918; Weyl 1929; Yang & Mills 1954; Portraits: Ihor Gorskyi



External symmetry



Internal symmetry

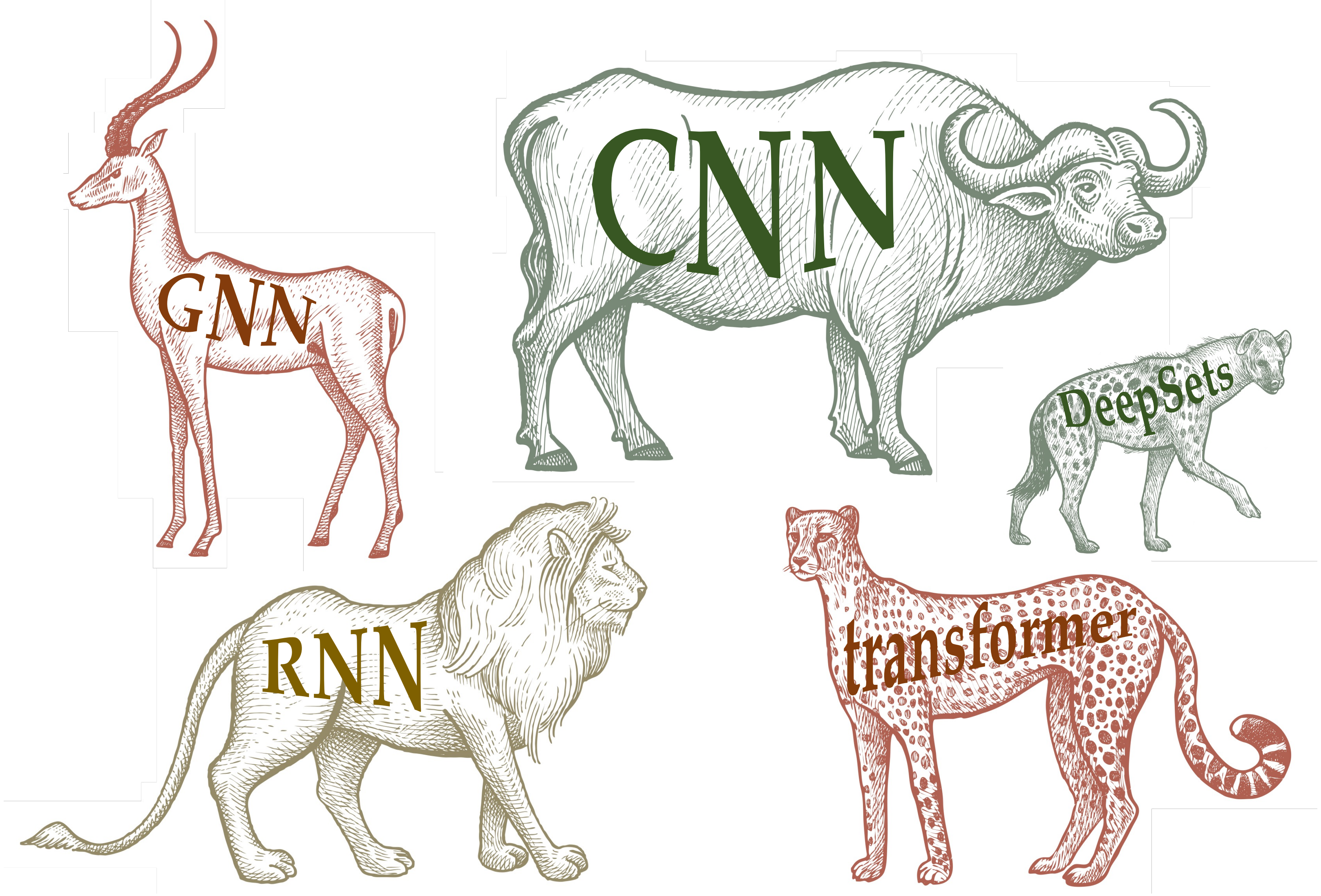
“It is only slightly overstating the case to say that Physics is the study of symmetry”

— “More is different”

P. Anderson

Anderson 1972

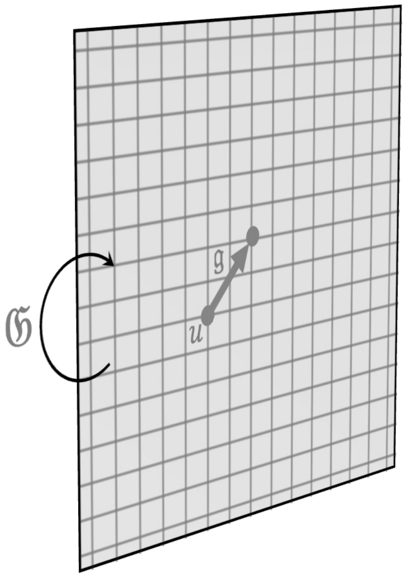
?



GNN
CNN
DeepSets
RNN
transformer

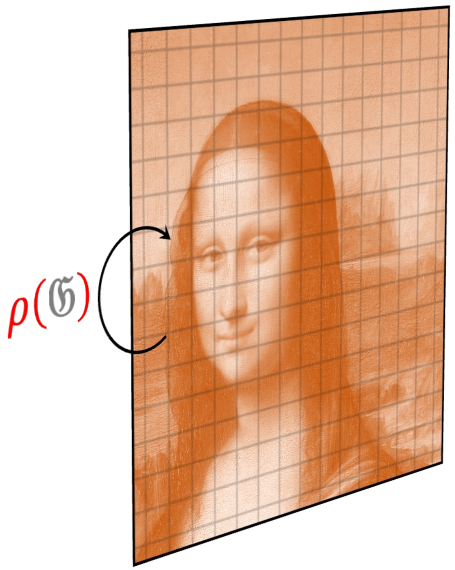
Geometric Deep Learning Blueprint

domain Ω



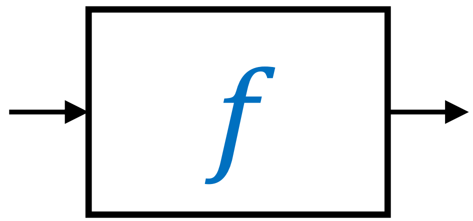
symmetry group $\mathfrak{G}$

signals $\mathcal{X}(\Omega)$



group representation $\rho(\mathfrak{G})$

functions $\mathcal{F}(\mathcal{X}(\Omega))$



Invariance / Equivariance

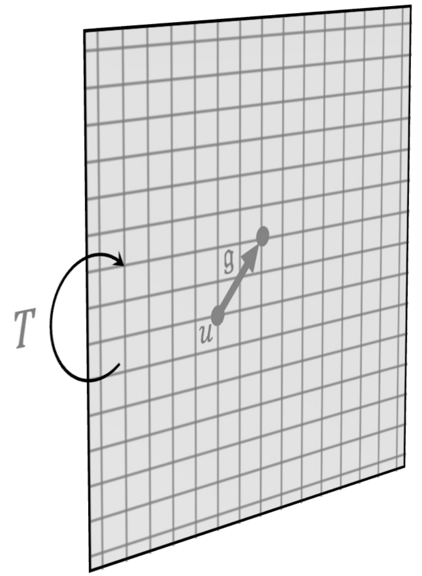
$$f(\rho(\mathfrak{g})x) = f(x)$$

$$f(\rho(\mathfrak{g})x) = \rho(\mathfrak{g})f(x)$$

B et al. 2021

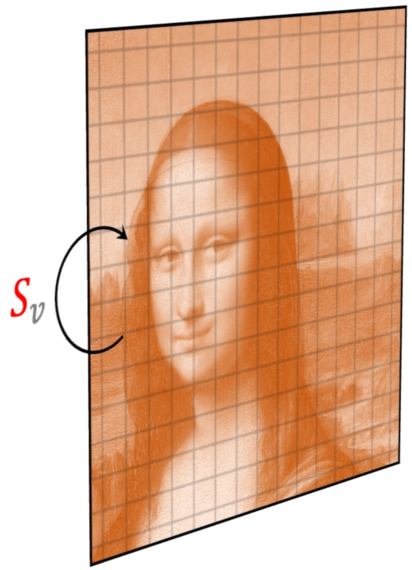
Example: Convolutional Neural Networks

Plane $\mathbb{R}^2$



Translation group $T(2)$

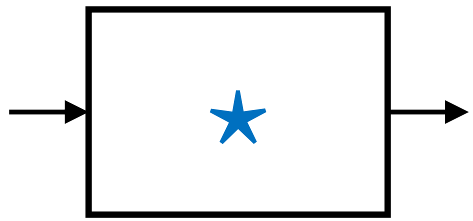
images $\mathcal{X}(\mathbb{R}^2)$



Shift operator S

$$S_v x(u) = x(u - v)$$

functions $\mathcal{F}(\mathcal{X}(\Omega))$

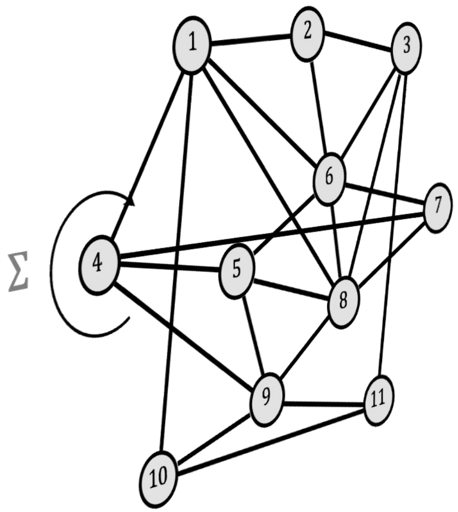


Convolutional layer

$$(Sx \star y) = S(x \star y)$$

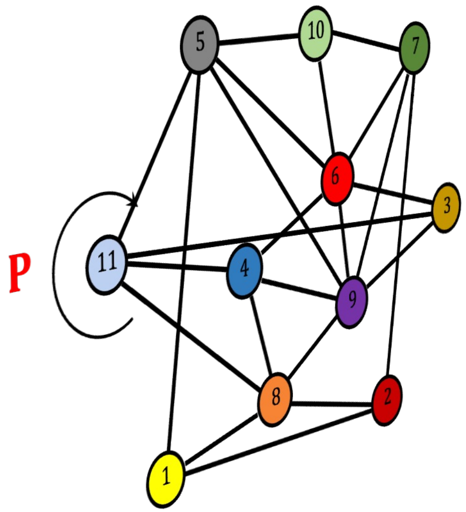
Example: Graph Neural Networks

Graph $G = (V, E)$



Permutation group Σ_n

Node features $\mathcal{X}(G)$



Permutation matrix P

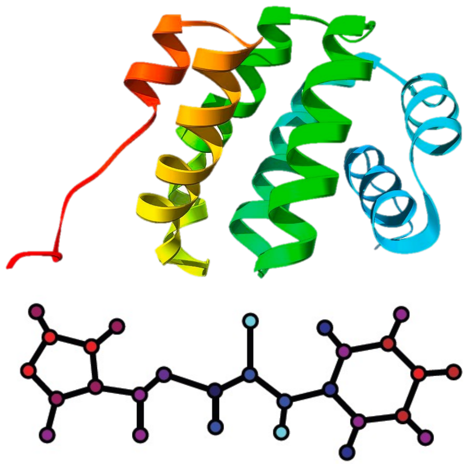
$$\mathbf{PX} = \left(x_{\pi^{-1}(i),j}\right)$$

functions $\mathcal{F}(\mathcal{X}(\Omega))$

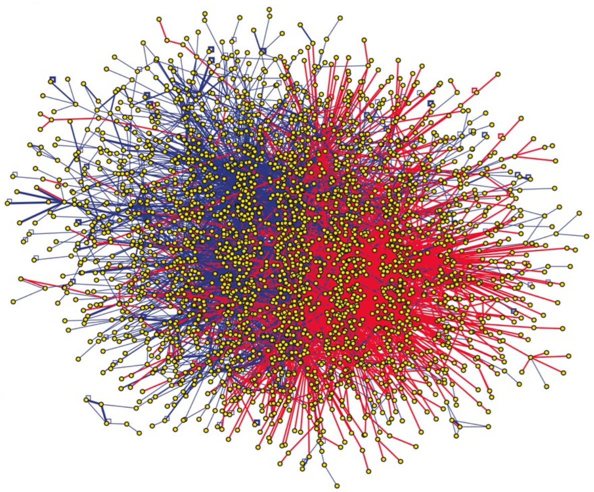
Message passing

$$\mathbf{F}\left(\mathbf{PX}, \mathbf{PAP}^\top\right) = \mathbf{PF}(\mathbf{X}, \mathbf{A})$$

Graphs = Systems of Relations and Interactions

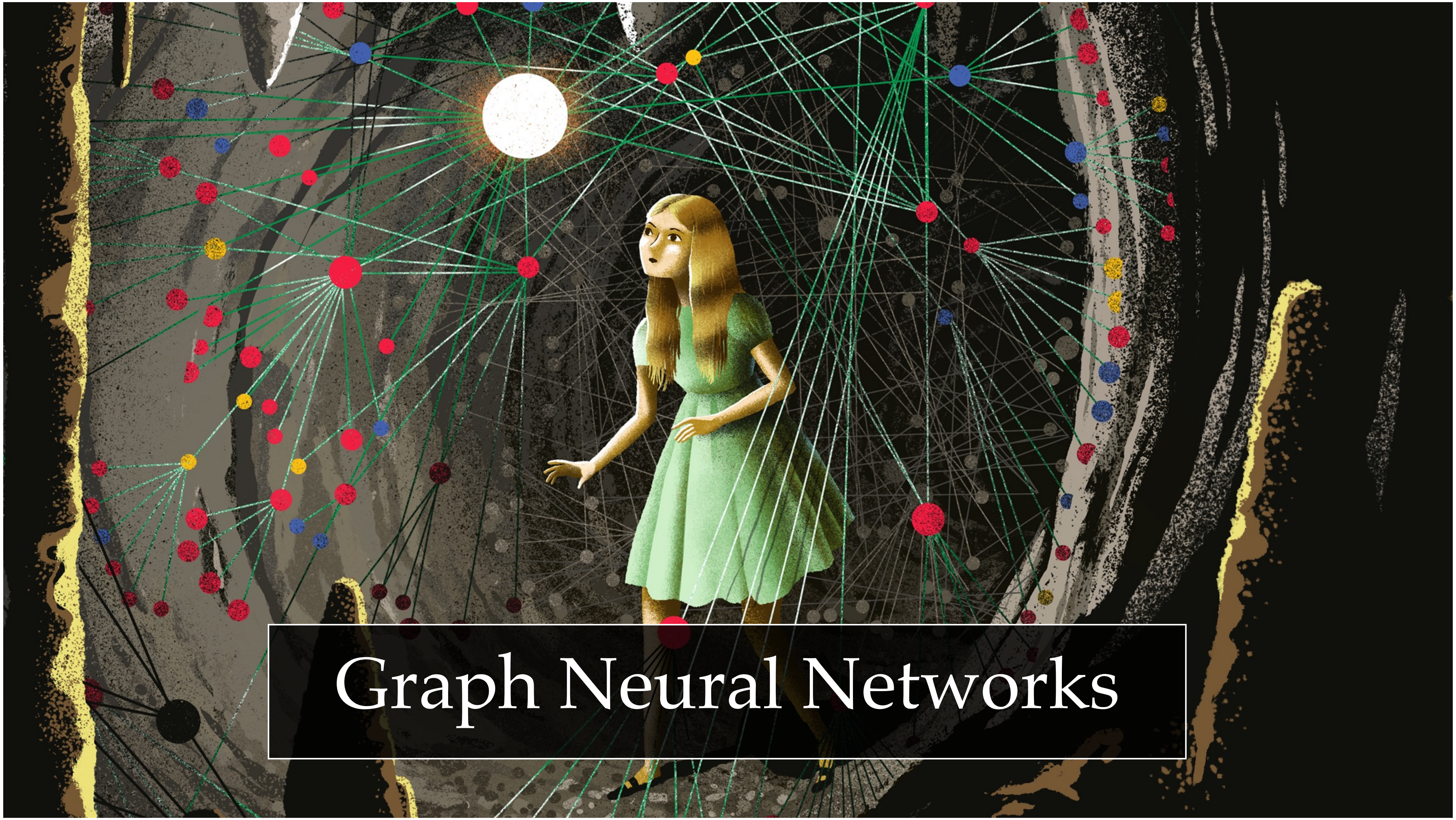


Molecules

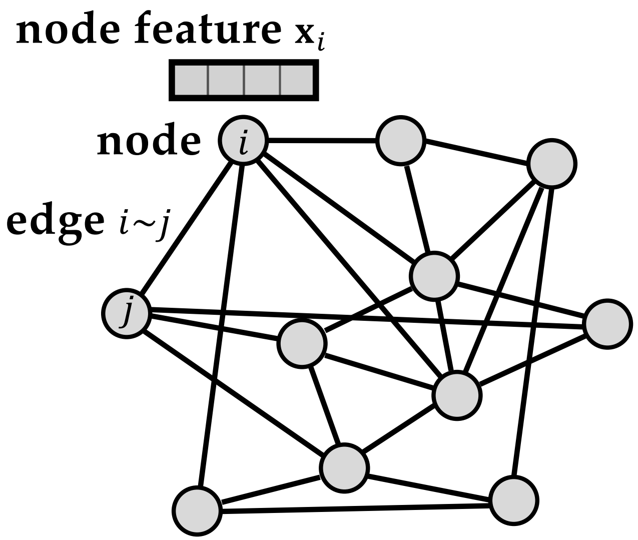


Interactomes

Social networks

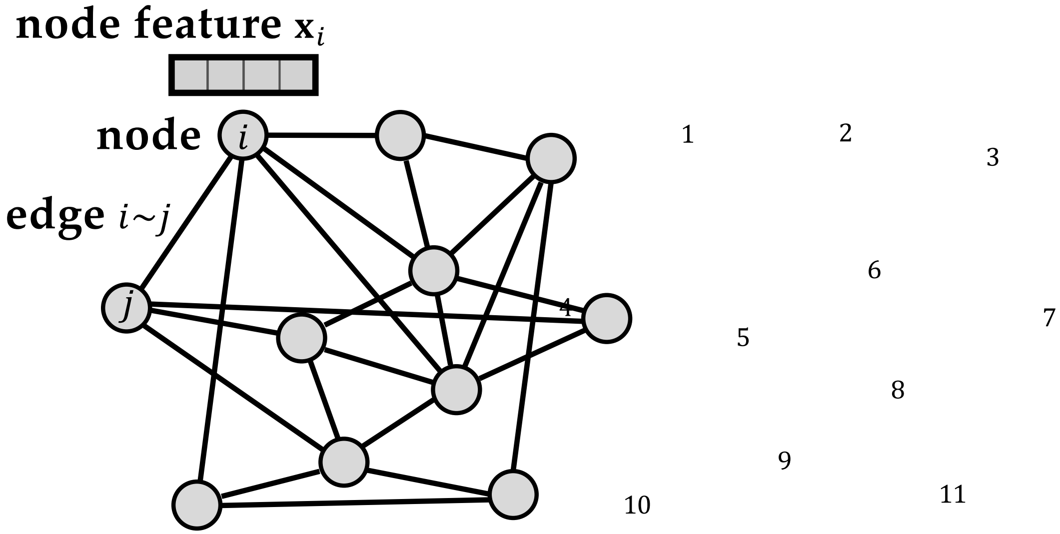

Graph Neural Networks

Graphs: The Basics



graph

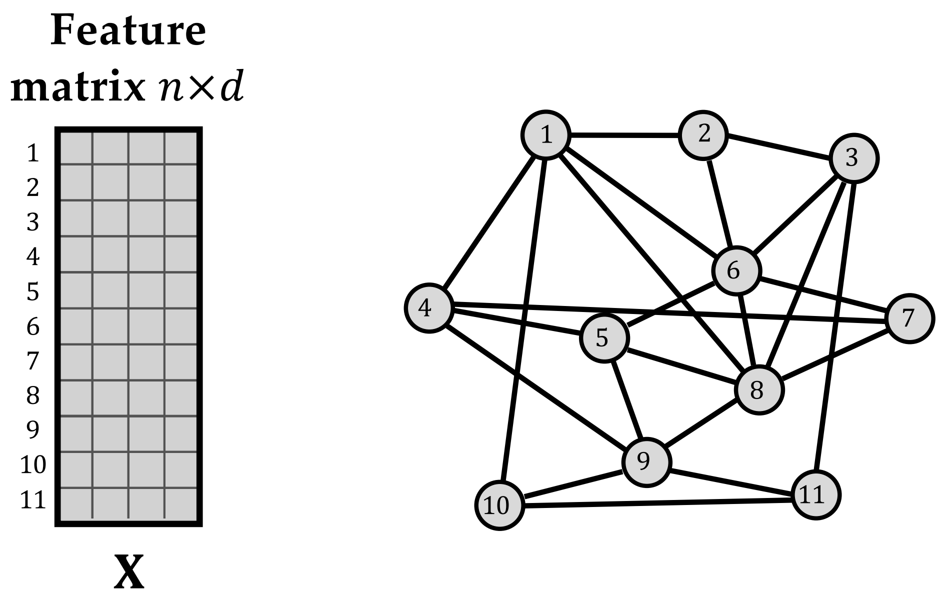
Key Structural Properties of Graphs



graph

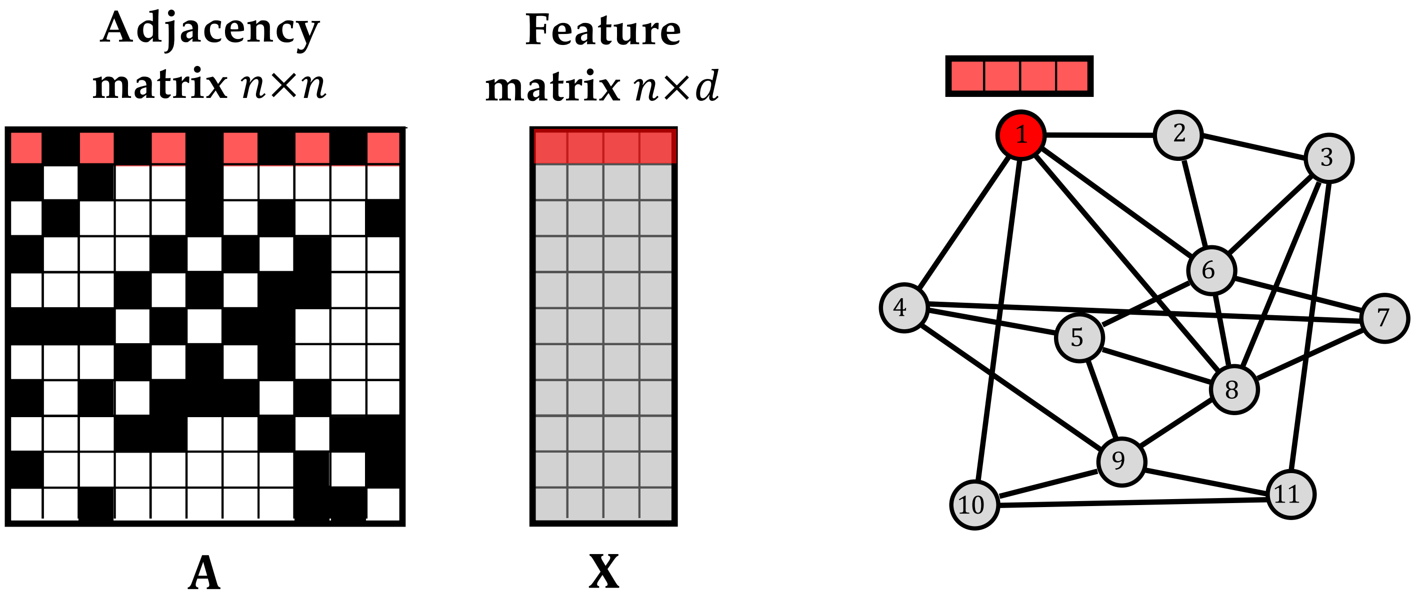
arbitrary ordering of nodes

Key Structural Properties of Graphs



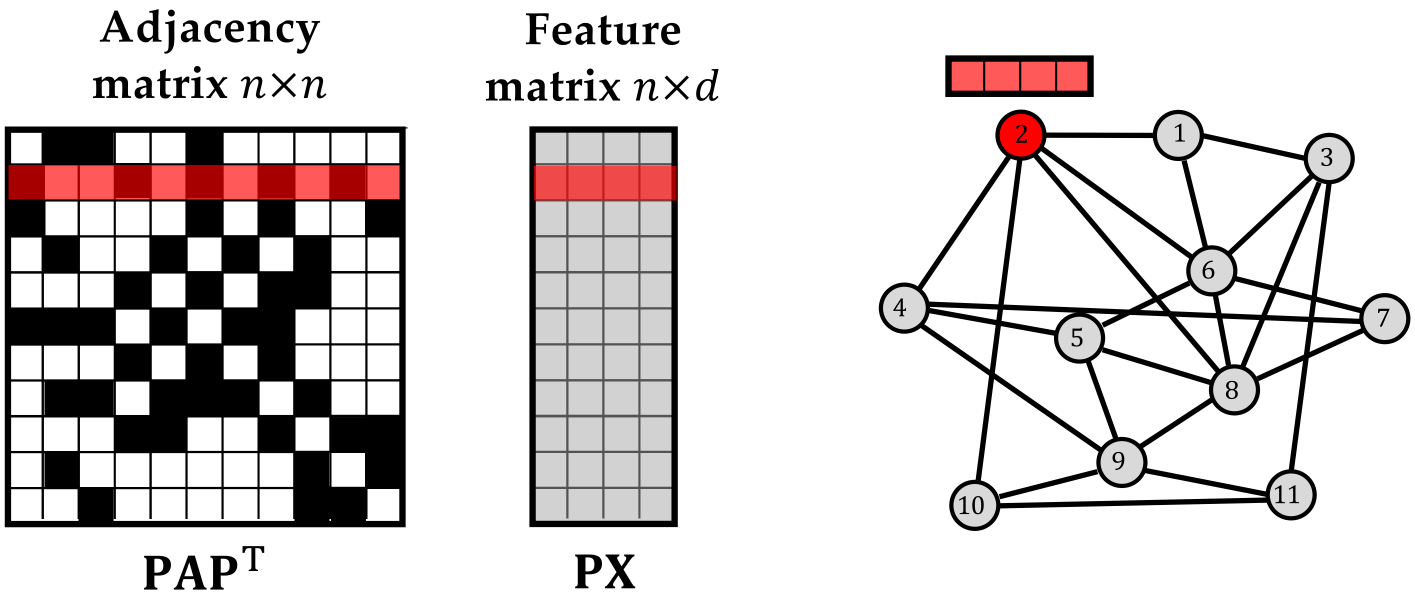
arbitrary ordering of nodes

Key Structural Properties of Graphs



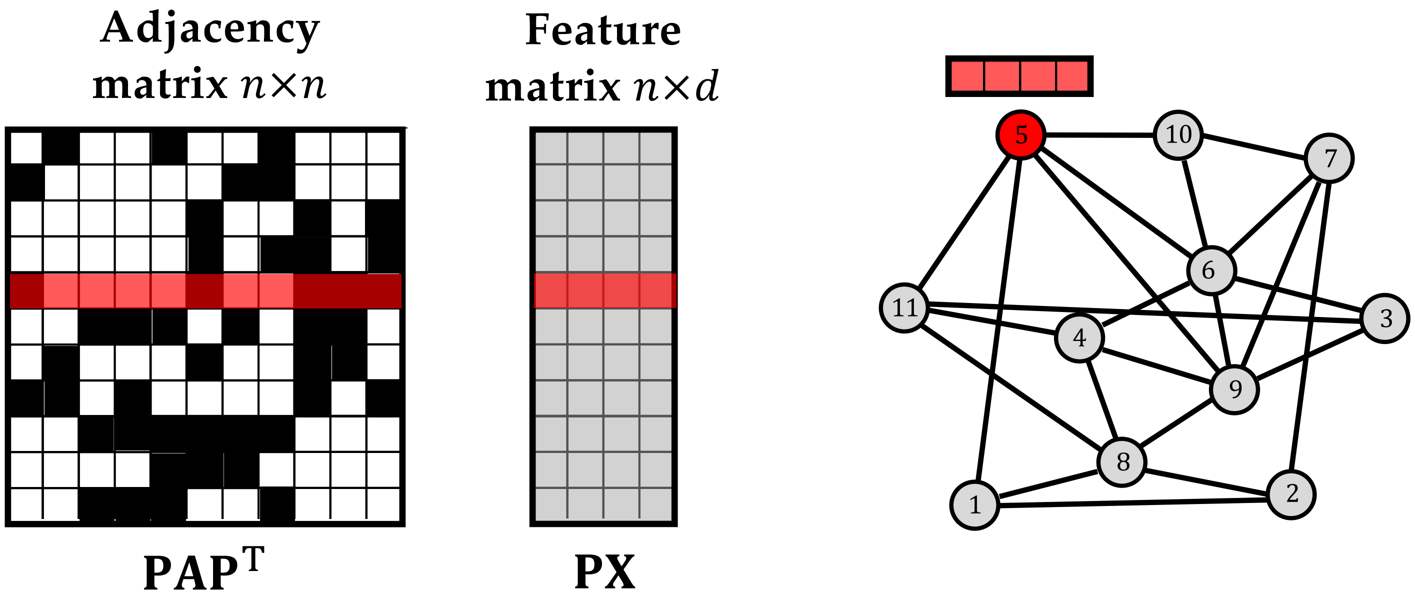
arbitrary ordering of nodes

Key Structural Properties of Graphs



arbitrary ordering of nodes

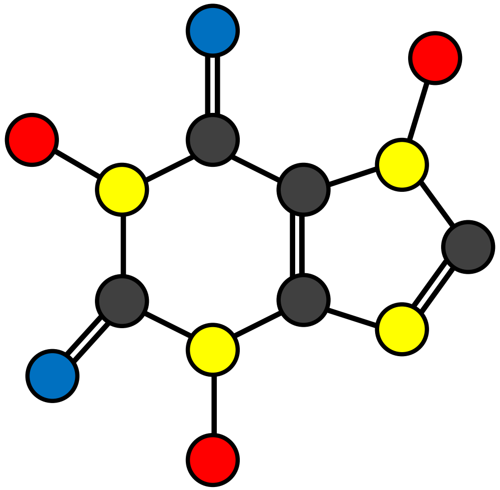
Key Structural Properties of Graphs



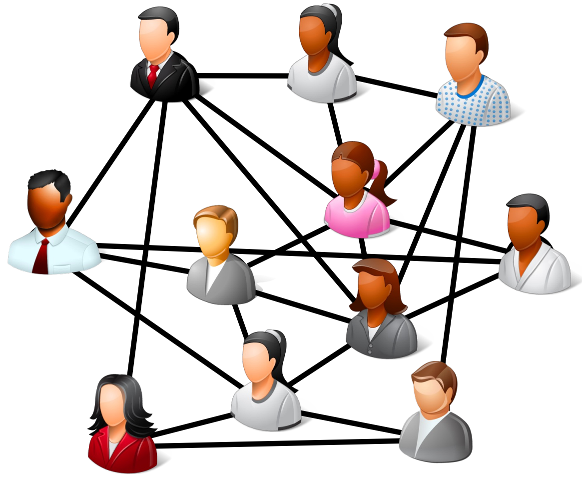
arbitrary ordering of nodes

n! permutations

Invariant vs Equivariant tasks



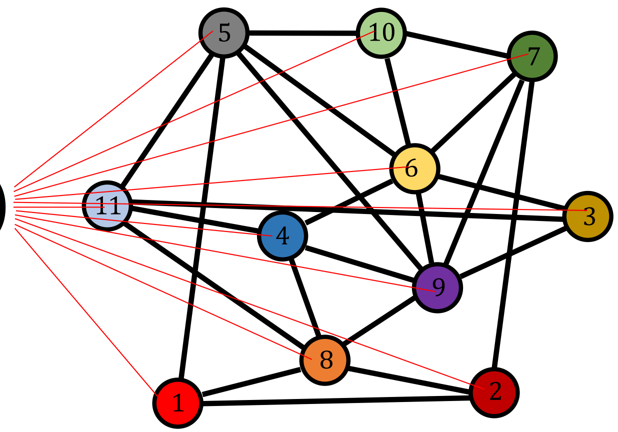
water solubility?



who is a spammer?

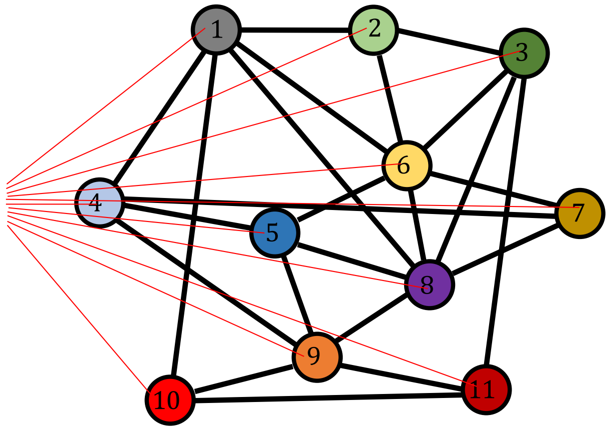
Invariant Graph Functions

graph function $f(\mathbf{X}, \mathbf{A})$



Invariant Graph Functions

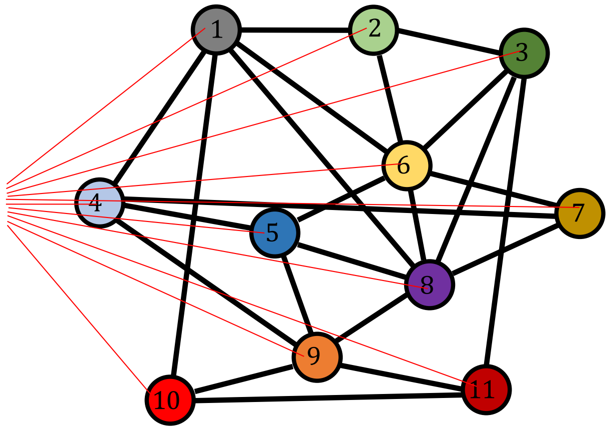
graph function $f(\mathbf{X}, \mathbf{A})$



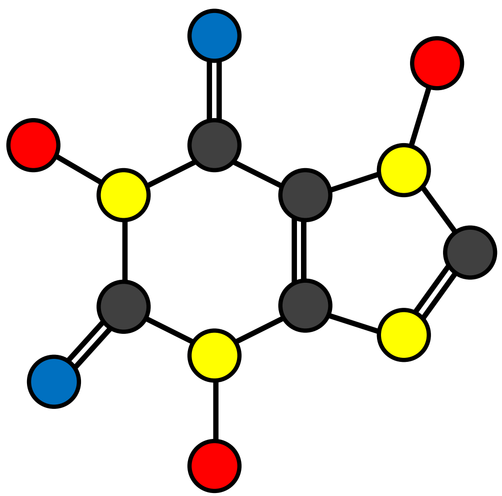
Invariant Graph Functions

permutation-invariant

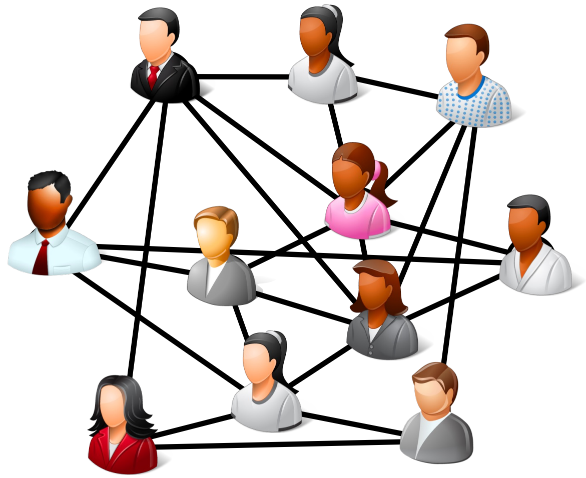
$$f\left(\mathbf{PX}, \mathbf{PAP}^\top\right) = f(\mathbf{X}, \mathbf{A})$$



Invariant vs Equivariant tasks



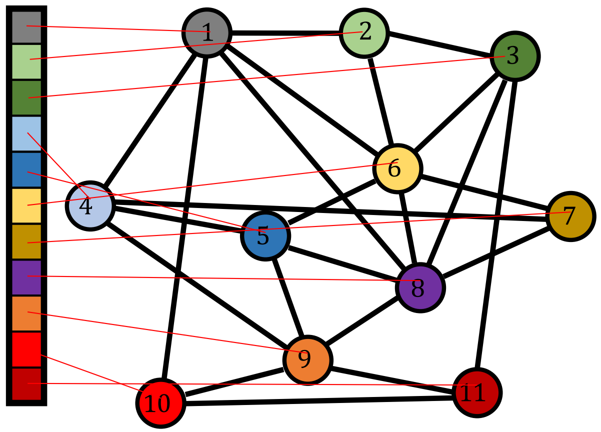
water solubility?



who is a spammer?

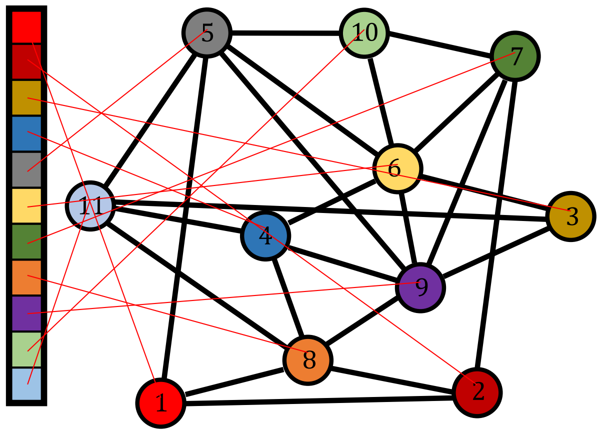
Equivariant Graph Functions

node function $\mathbf{F}(\mathbf{X}, \mathbf{A})$



Equivariant Graph Functions

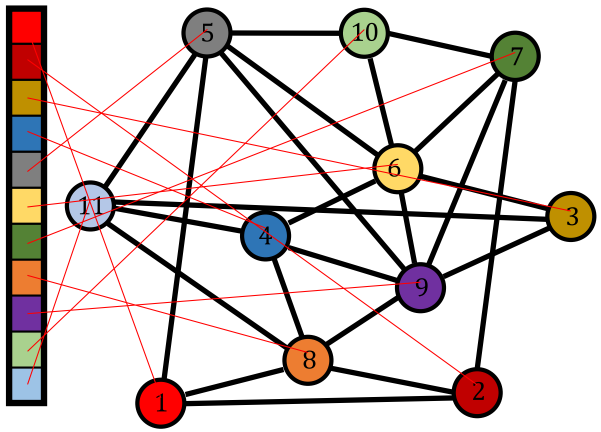
node function $\mathbf{F}(\mathbf{X}, \mathbf{A})$



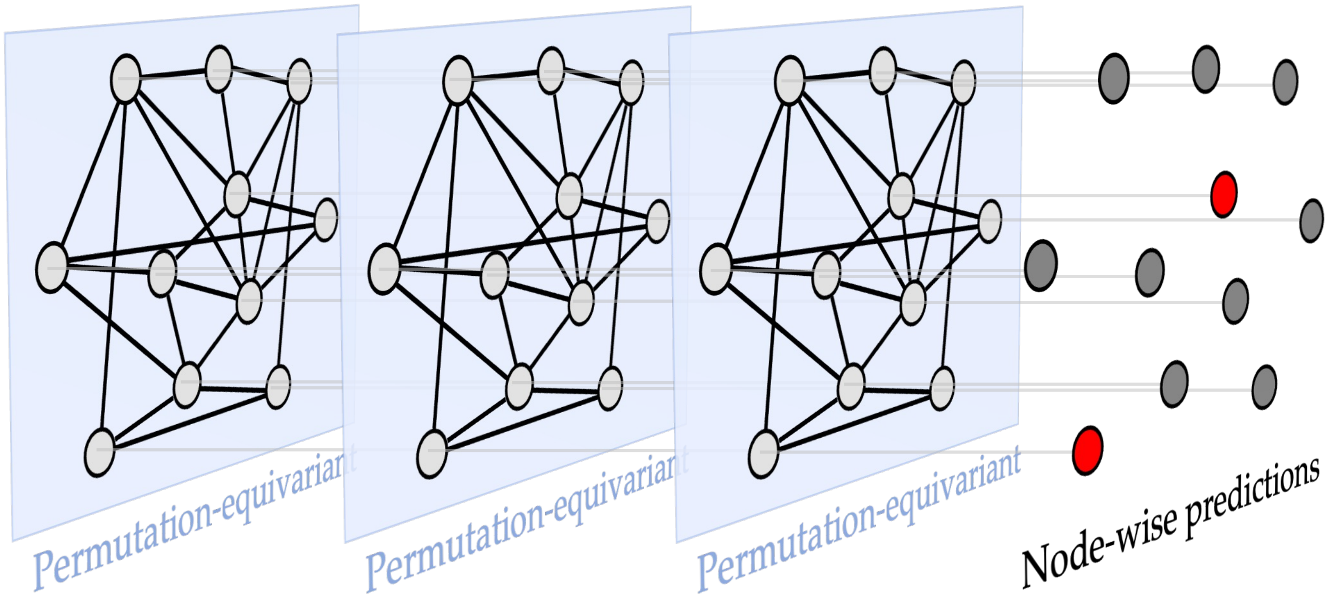
Equivariant Graph Functions

permutation-equivariant

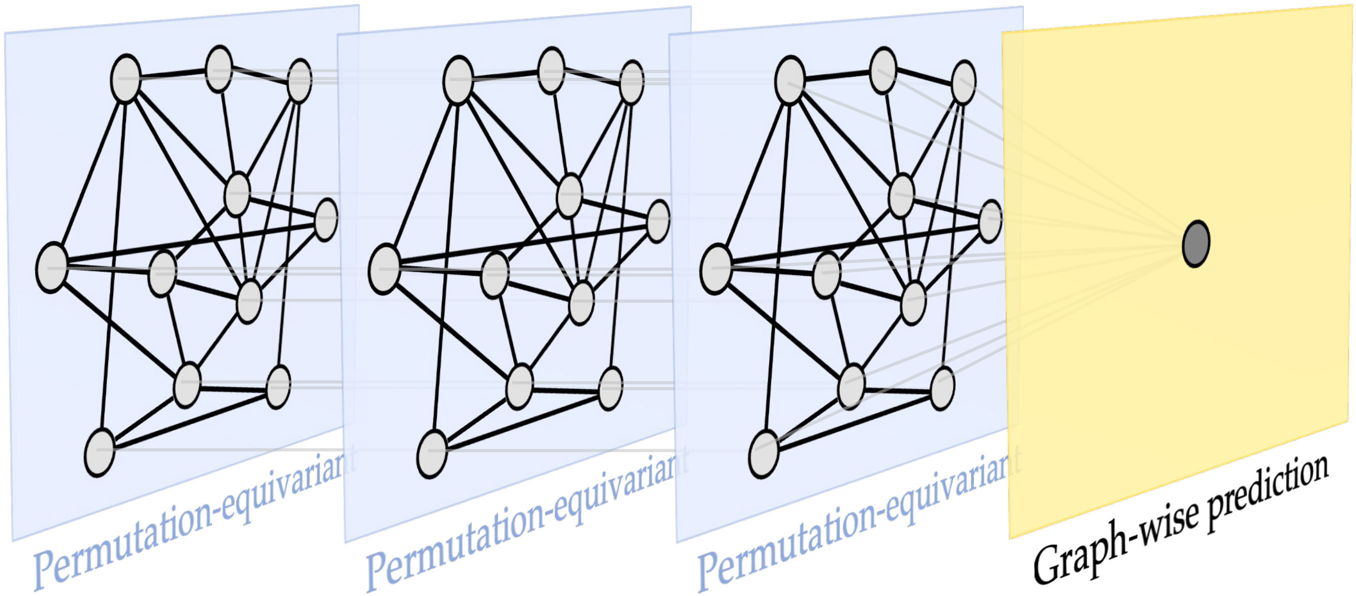
$$\mathbf{F}\left(\mathbf{PX}, \mathbf{PAP}^\top\right) = \mathbf{PF}(\mathbf{X}, \mathbf{A})$$



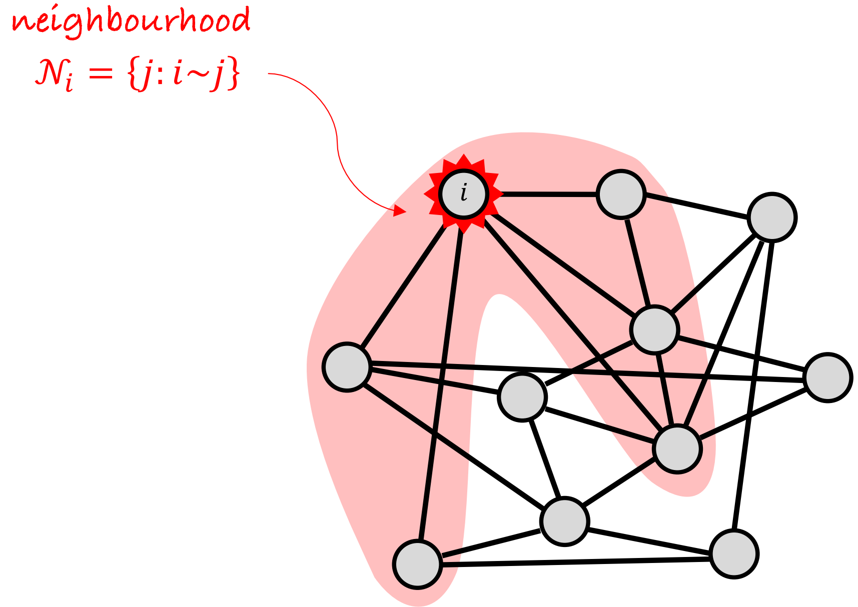
Graph Neural Networks: Node tasks



Graph Neural Networks: Graph tasks

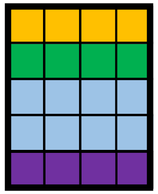


Neighbour Aggregation

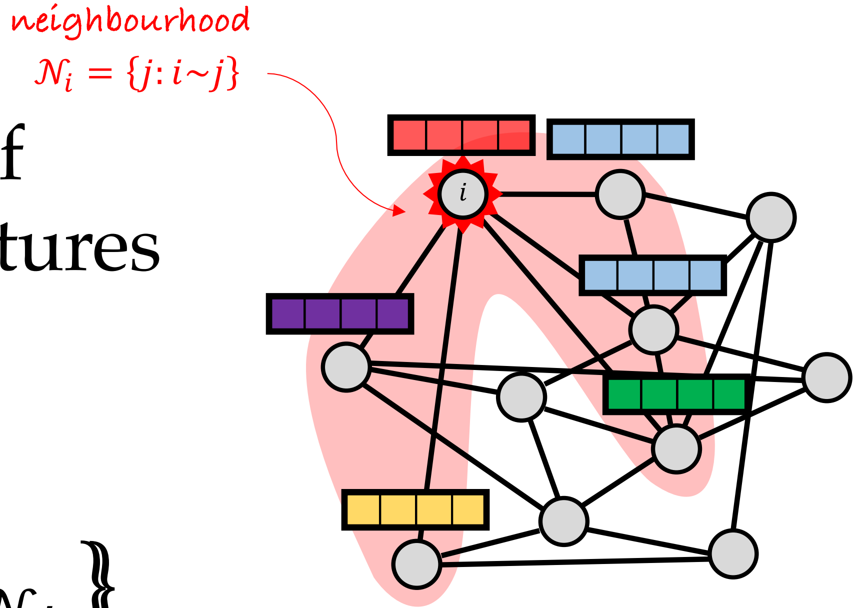


Neighbour Aggregation

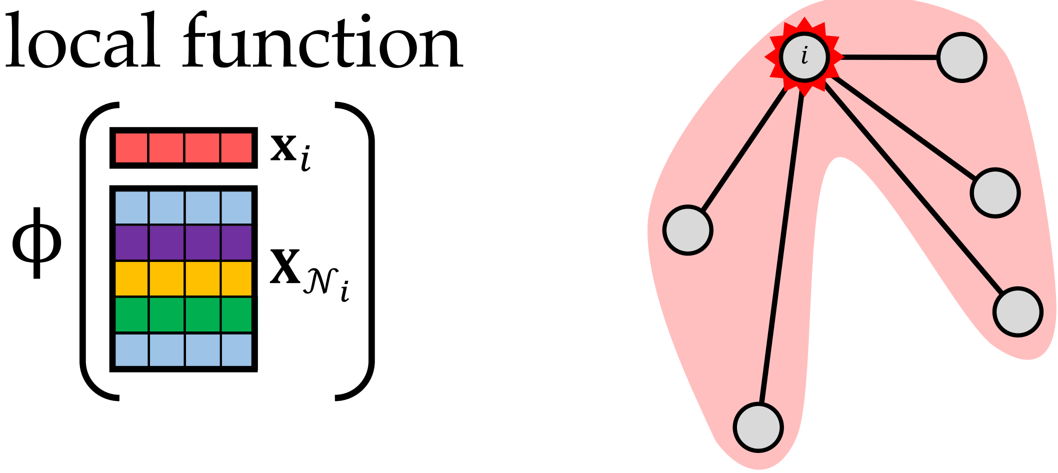
multiset of
neighbour features



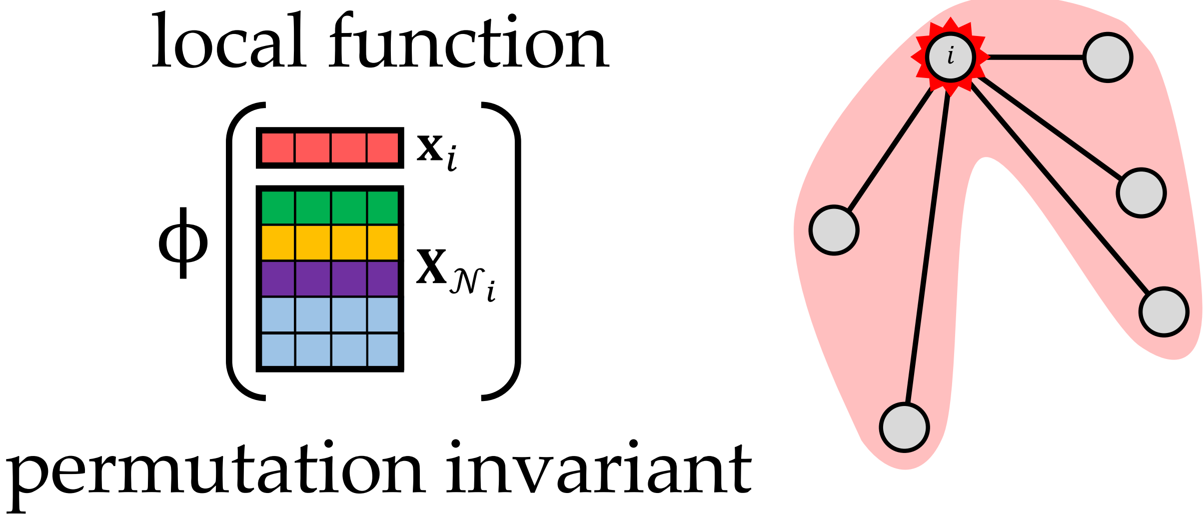
$$\mathbf{X}_{\mathcal{N}_i} = \left\{\mathbf{x}_{j\in\mathcal{N}_i}\right\}$$



Neighbour Aggregation



Neighbour Aggregation

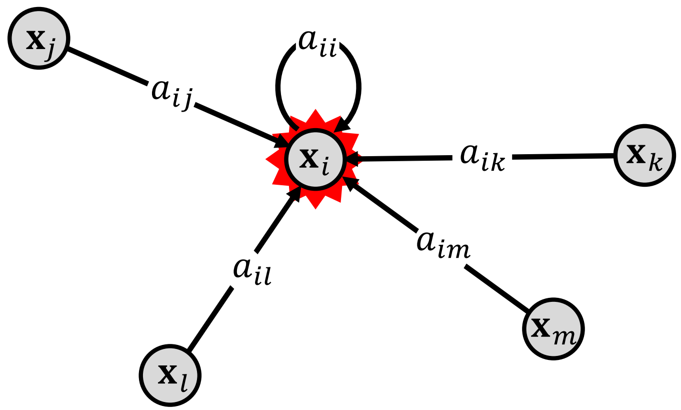


GNN Layer

$$\mathbf{F}(\mathbf{X}, \mathbf{A}) = \begin{bmatrix} - \phi(\mathbf{x}_1, \mathbf{X}_{\mathcal{N}_1}) - \\ \vdots \\ - \phi(\mathbf{x}_i, \mathbf{X}_{\mathcal{N}_i}) - \\ \vdots \\ - \phi(\mathbf{x}_n, \mathbf{X}_{\mathcal{N}_n}) - \end{bmatrix}$$

permutation equivariant

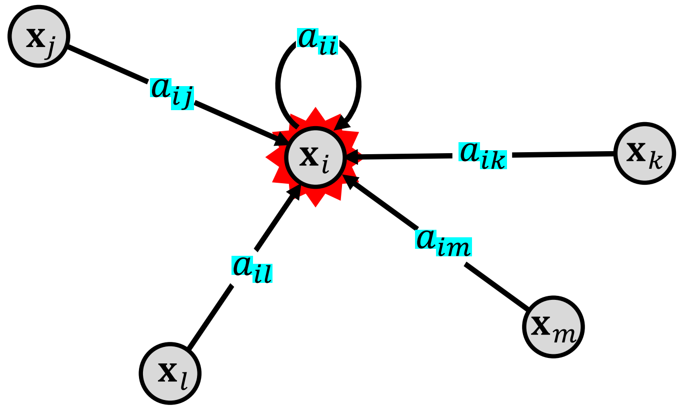
Convolutional GNNs



$$\mathbf{x}_i \leftarrow \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} \psi(\mathbf{x}_j) \right)$$

Defferard et al. 2016; Kipf, Welling 2016 (GCN)

Convolutional GNNs

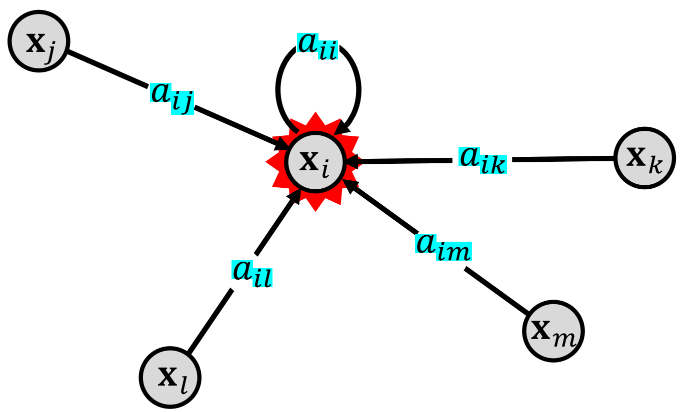


$$\mathbf{x}_i \leftarrow \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} \psi(\mathbf{x}_j) \right)$$

graph adjacency

Defferard et al. 2016; Kipf, Welling 2016 (GCN)

Convolutional GNNs



$$\mathbf{x}_i \leftarrow \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} \psi(\mathbf{x}_j) \right)$$

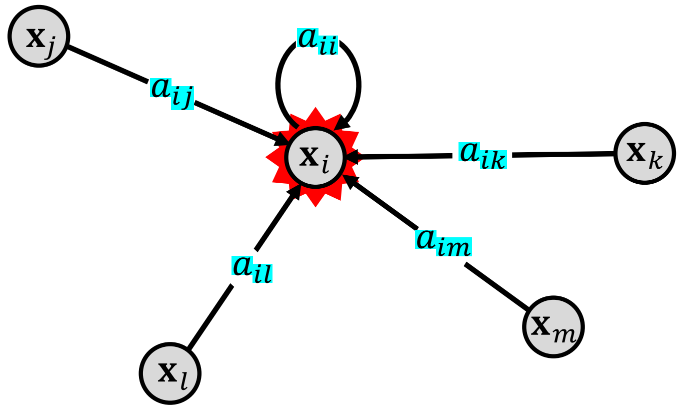
nonlinear activation

graph adjacency

node-wise transformation

Defferard et al. 2016; Kipf, Welling 2016 (GCN)

Convolutional GNNs



$$\mathbf{x}_i \leftarrow \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} \mathbf{W} \mathbf{x}_j\right)$$

nonlinear activation

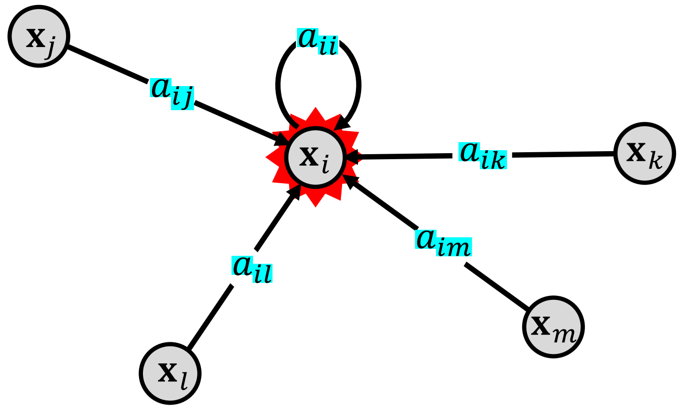
graph adjacency

node-wise *linear* transformation

Defferard et al. 2016; Kipf, Welling 2016 (GCN)

Convolutional GNNs

- Simplest GNN
- Highly scalable
- Industrial use cases
- Folklore: works only on homophilic graphs



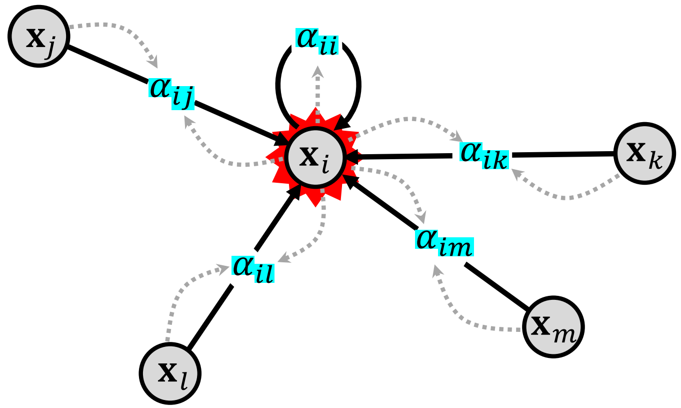
$$\mathbf{X} \leftarrow \sigma(\mathbf{A}\mathbf{X}\mathbf{W})$$

diffusion $n \times n$

channel mixing $d \times d$

Defferard et al. 2016; Kipf, Welling 2016 (GCN)
Rossi, Frasca et B 2020 (SIGN); Ying et al. 2018 (PinSAGE)

Attentional GNNs

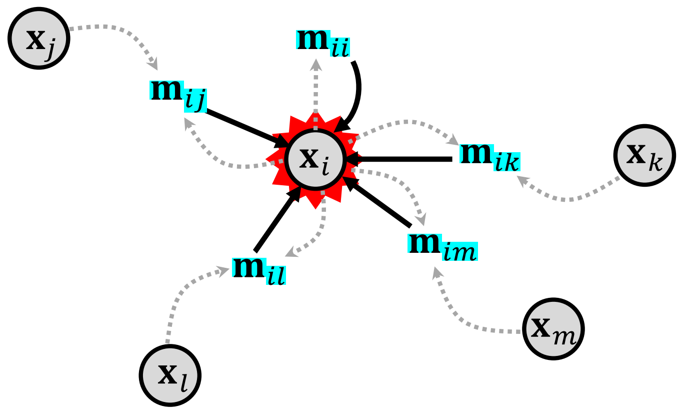


$$\mathbf{x}_i \leftarrow \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} \alpha_{ij}(\mathbf{x}_i, \mathbf{x}_j)\psi(\mathbf{x}_j) \right)$$

learnable attention weights

Monti et al. 2017; Veličković et al. 2018 (GAT)

Message-Passing GNNs

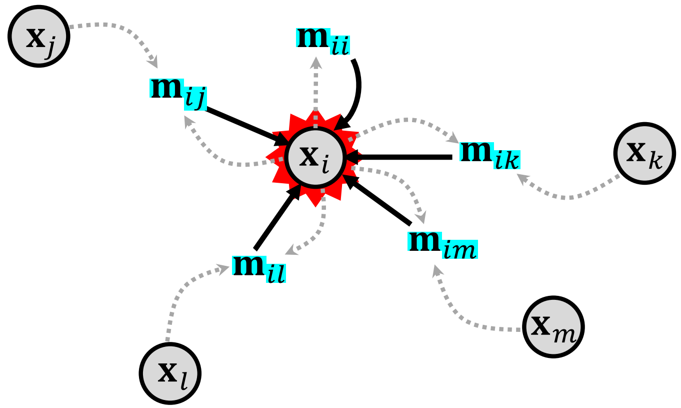


$$\mathbf{x}_i \leftarrow \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} \psi(\mathbf{x}_i, \mathbf{x}_j) \right)$$

message from
node j to node i

Gilmer et al. 2017 (MPNN); Battaglia et al 2018 (Graph Networks)
Wang et B, Solomon 2018 (edgeconv)

Message-Passing GNNs

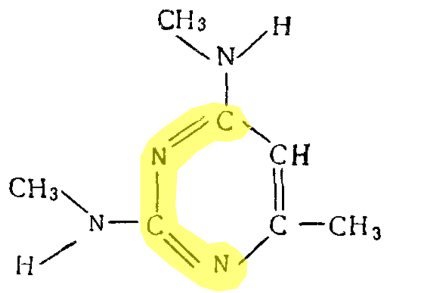


Message Passing GNNs with injective aggregation are equivalent to Weisfeiler-Lehman graph isomorphism test

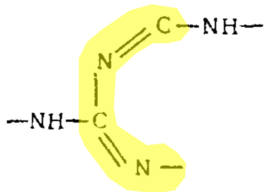
Xu et al. 2019; Morris et al. 2019

Weisfeiler-Lehman Test & Chemical precursors of GNNs

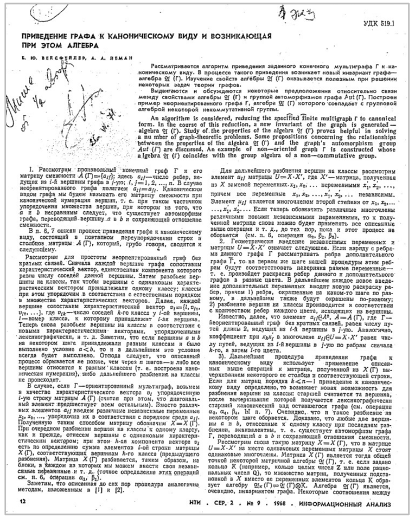
George Vlăduţ

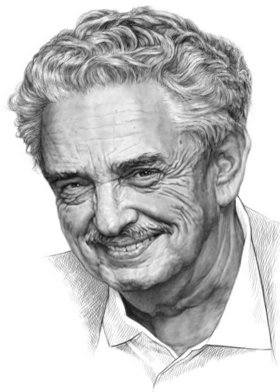


$H^1N^3J^1C^3NC^3J^1MC^3NX3N^3H^1J^1$



$H^1N^3R^1C^3R^1NC^3NR^1N^3H^1R^1$





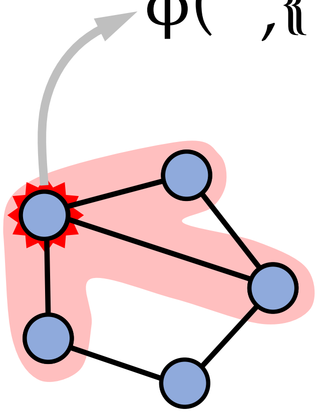
Andrey Lehman

Boris Weisfeiler

Vlăduţ et al. 1959; Weisfeiler, Lehman 1968

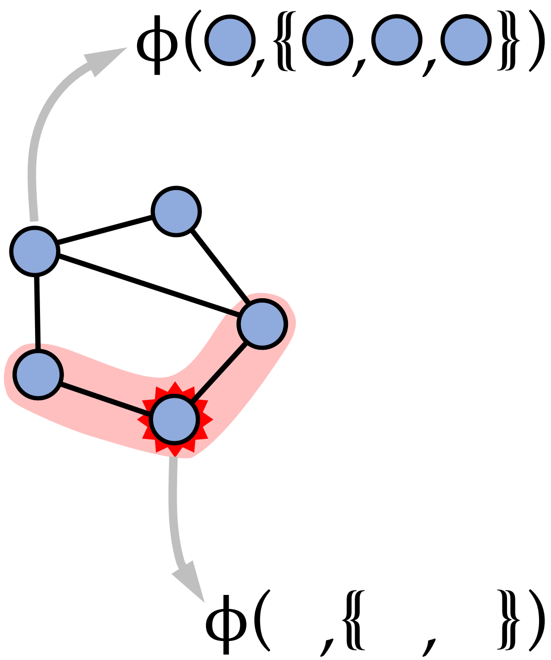
Weisfeiler-Lehman Test

$\phi(\quad,\{\!\{\quad,\quad,\quad\}\!\})$



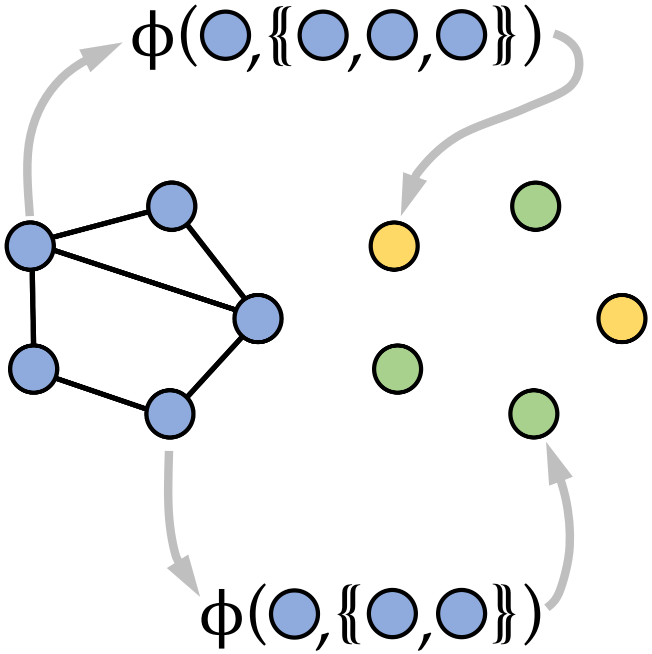
Weisfeiler, Lehman 1968

Weisfeiler-Lehman Test



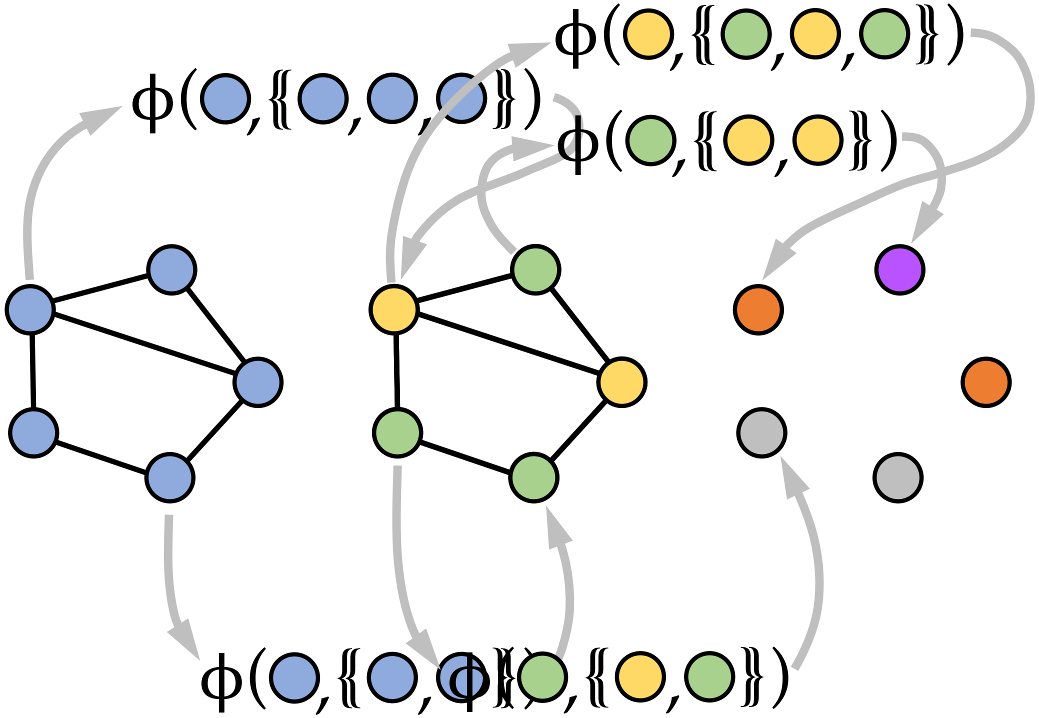
Weisfeiler, Lehman 1968

Weisfeiler-Lehman Test



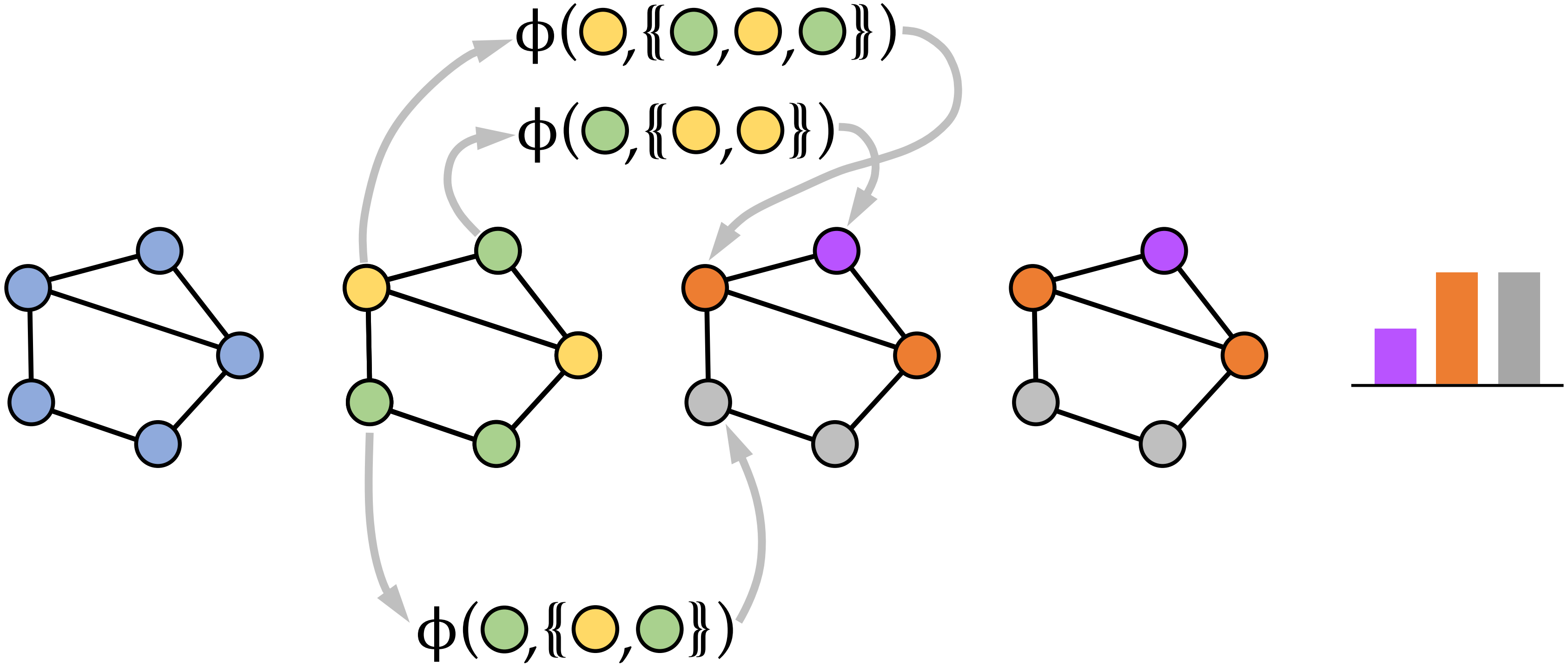
Weisfeiler, Lehman 1968

Weisfeiler-Lehman Test



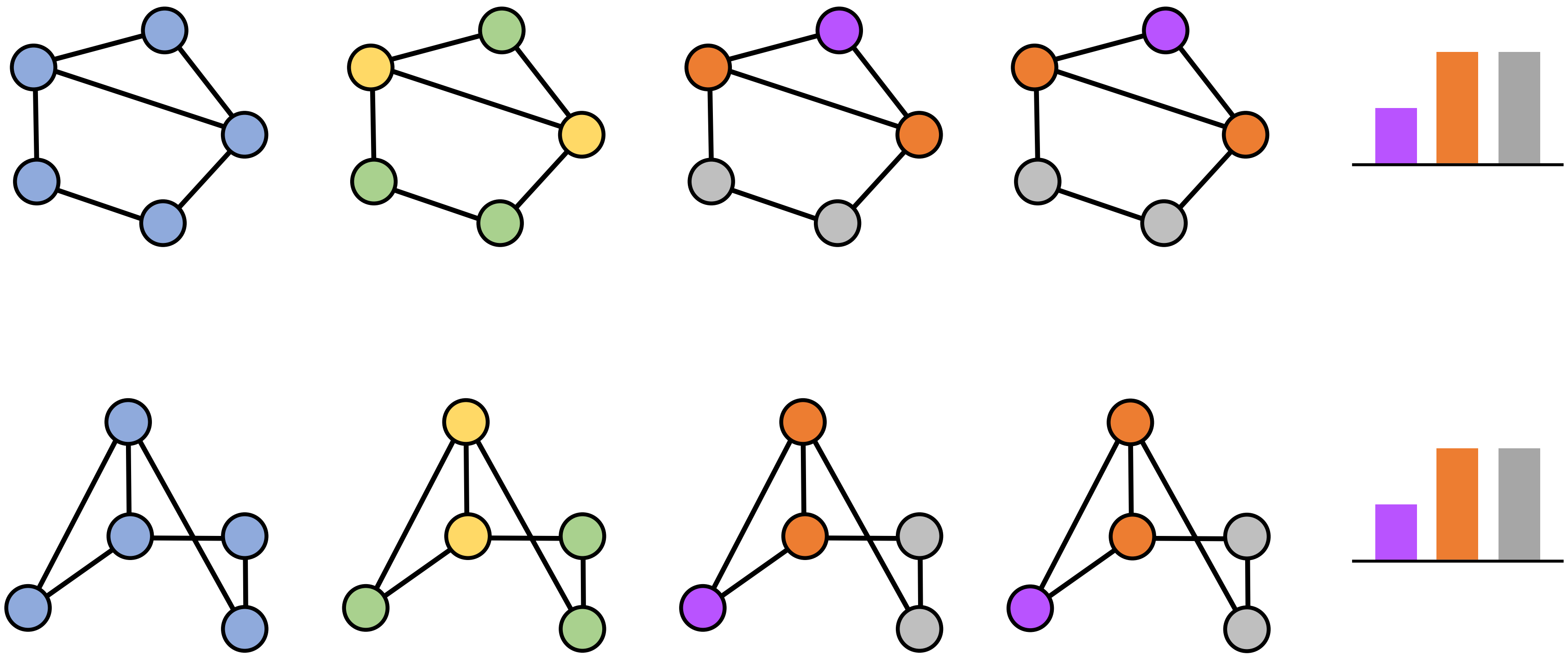
Weisfeiler, Lehman 1968

Weisfeiler-Lehman Test



Weisfeiler, Lehman 1968

Weisfeiler-Lehman Test

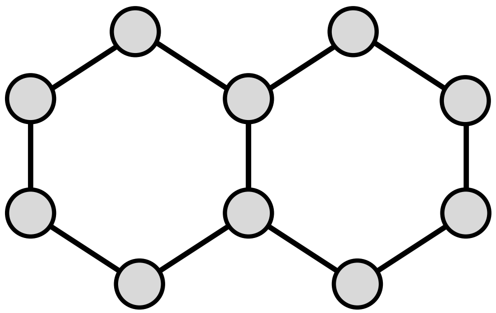


Weisfeiler, Lehman 1968

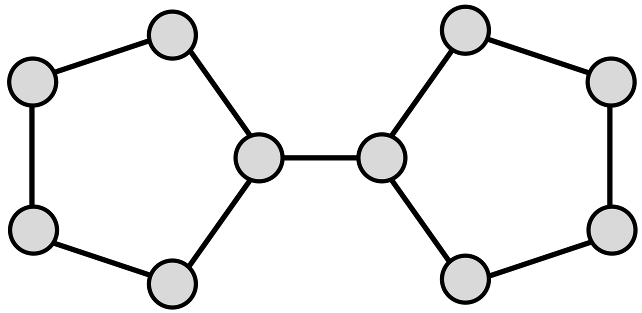
non-isomorphic graphs that are WL-equivalent

Necessary but insufficient condition!

Message-Passing GNNs have limited expressive power!

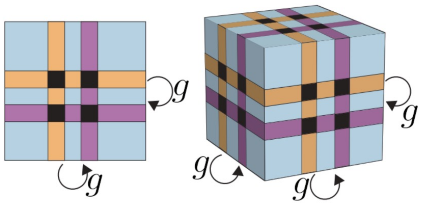


decalin



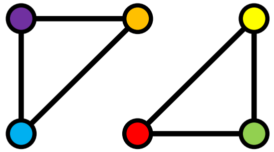
bicyclopnetyl

Towards More Expressive GNNs



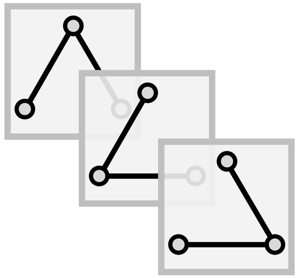
Higher-order
WL tests

Maron et al. 2019
Morris et al. 2019



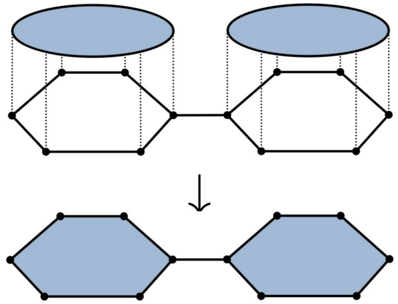
Positional &
Structural encoding

Monti, Otness et B 2018
Sato 2020
Dwivedi et al. 2020
Bouritsas, Frasca et B 2020
…many more



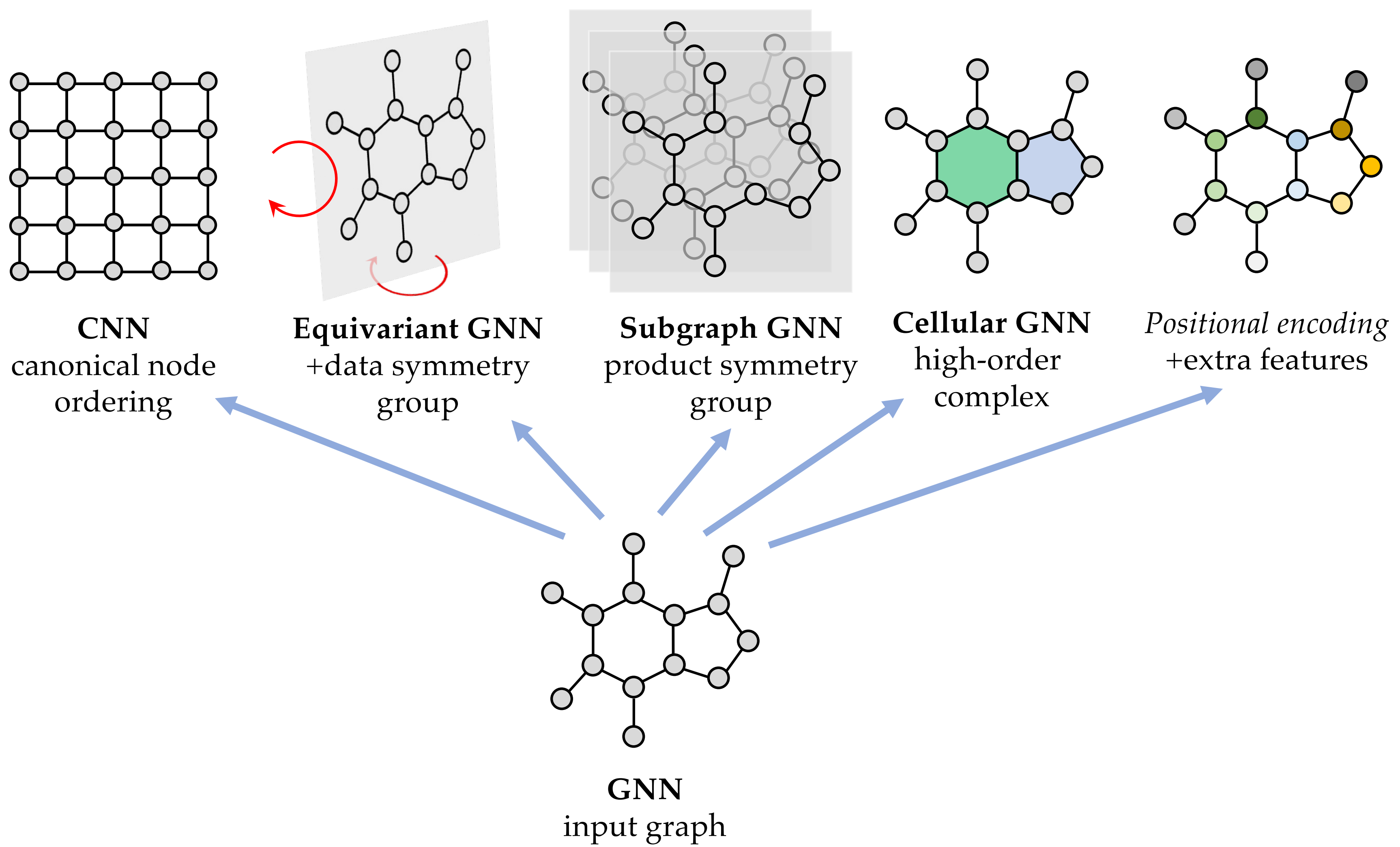
Subgraph
GNNs

Papp et al. 2021
Cotta et al. 2021
Zhao et al. 2021
Bevilacqua, Frasca et B, Maron 2021
Frasca et B, Maron 2022

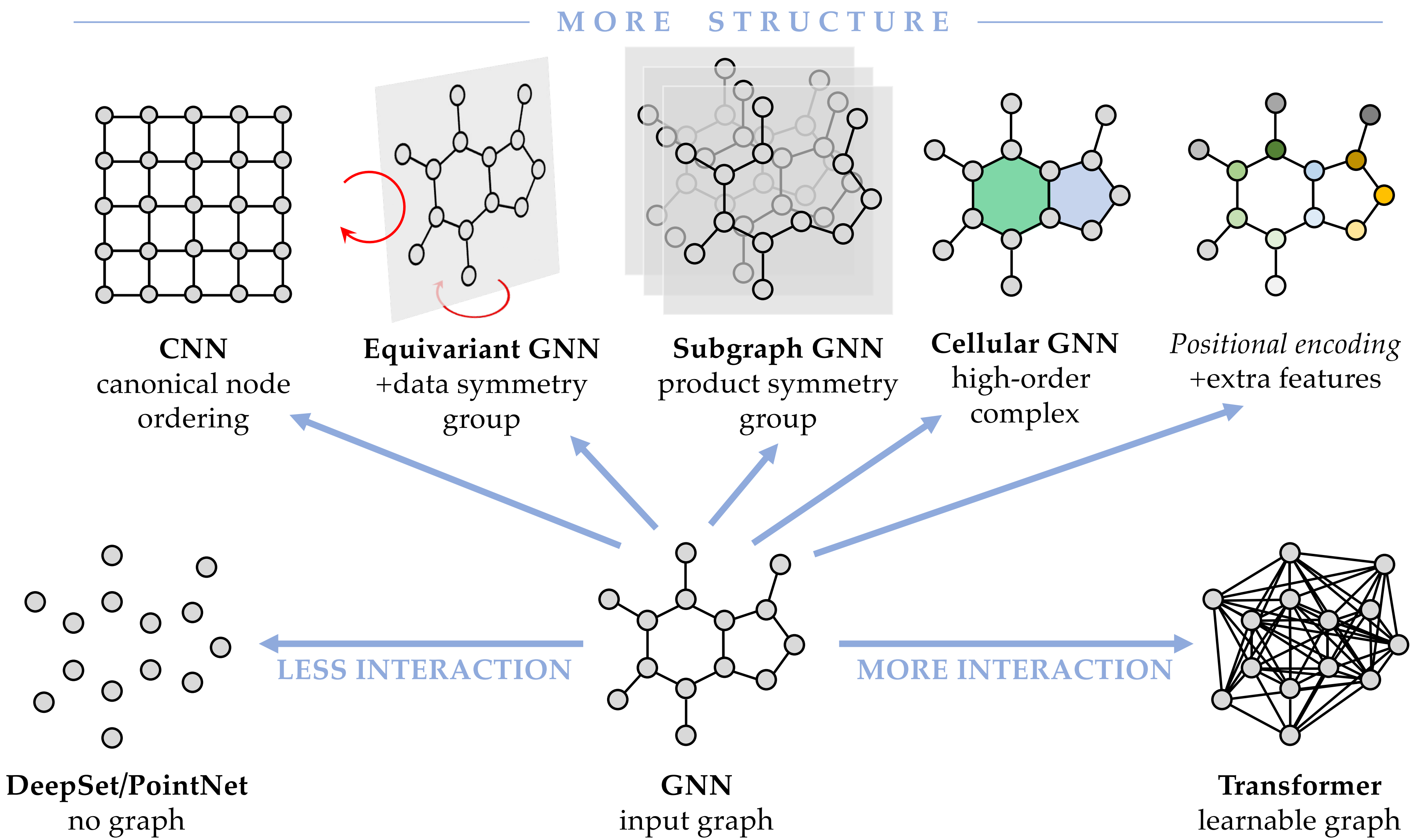


Topological
message passing

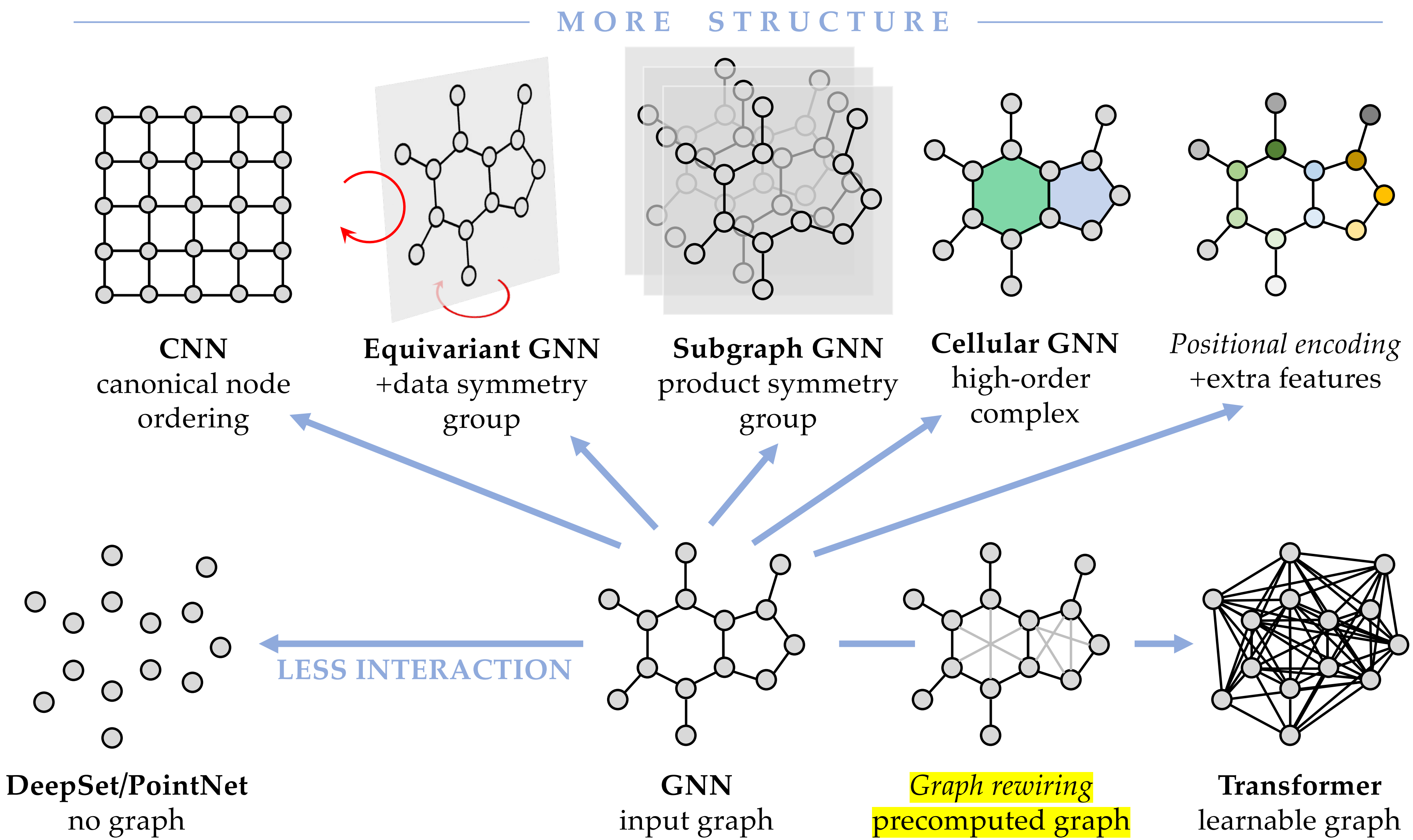
Bodnar, Frasca et B 2021



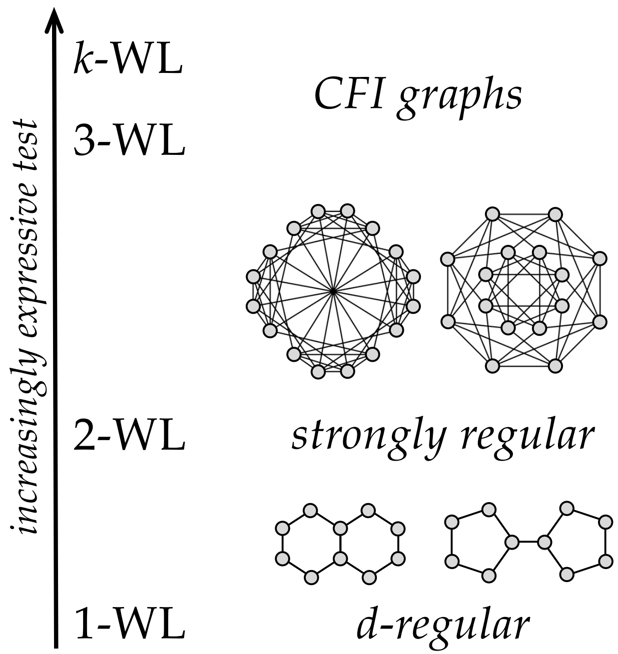
CNN
canonical node ordering
Equivariant GNN
+data symmetry group
Subgraph GNN
product symmetry group
Cellular GNN
high-order complex
Positional encoding
+extra features
GNN
input graph



MORE STRUCTURE
CNN
canonical node ordering
Equivariant GNN
+data symmetry group
Subgraph GNN
product symmetry group
Cellular GNN
high-order complex
Positional encoding
+extra features
LESS INTERACTION
MORE INTERACTION
DeepSet/PointNet
no graph
GNN
input graph
Transformer
learnable graph


MORE STRUCTURE
CNN
canonical node ordering
Equivariant GNN
+data symmetry group
Subgraph GNN
product symmetry group
Cellular GNN
high-order complex
Positional encoding
+extra features
DeepSet/PointNet
no graph
LESS INTERACTION
GNN
input graph
Graph rewiring
precomputed graph
Transformer
learnable graph

Weisfeiler-Lehman hierarchy

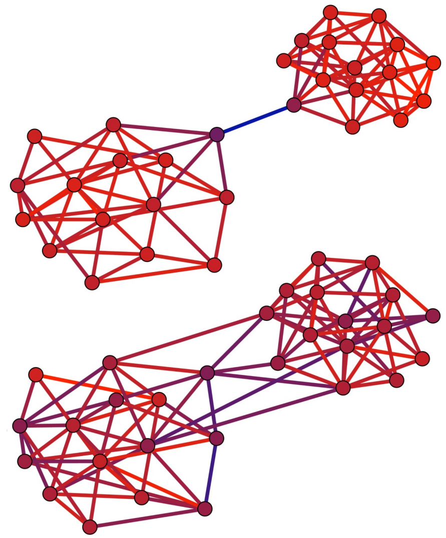


GNN expressive power

Weisfeiler, Lehman 1968 (2-WL); Babai, Mathon 1979 (*k*-WL); Cai, Fürer, Immerman 1992 (CFI graphs)

Gap between Theory & Practice

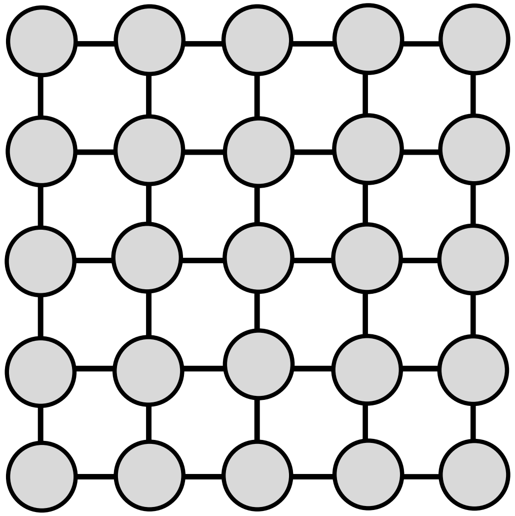
Graphs may be unfriendly for message passing resulting in "bottlenecks"



Graph rewiring

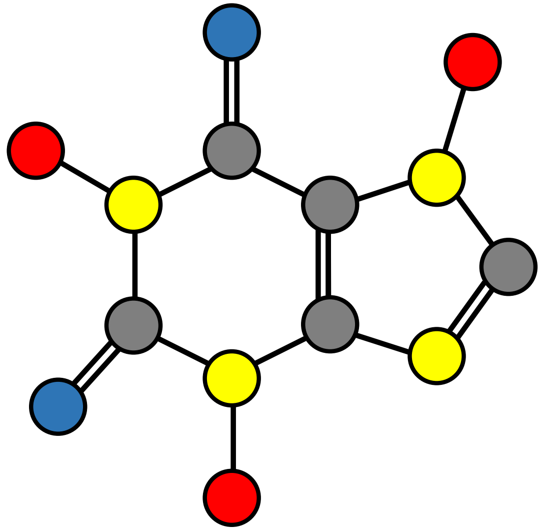
Alon, Yahav 2020 (bottlenecks); Hamilton et al. 2017 (neighbour sampling); Klicpera et al. 2019 (diffusion); Topping, Di Giovanni et B 2022 (Ricci flow); Deac et al. 2022 (expanders)

Graphs vs Meshes vs Grids



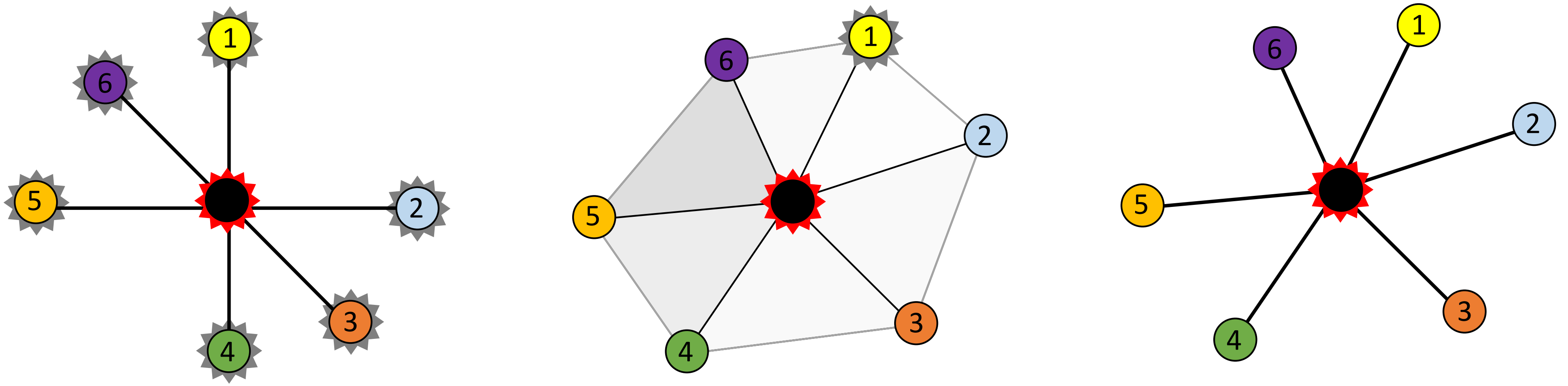
Grid

Mesh



Graph

Graphs vs Meshes vs Grids

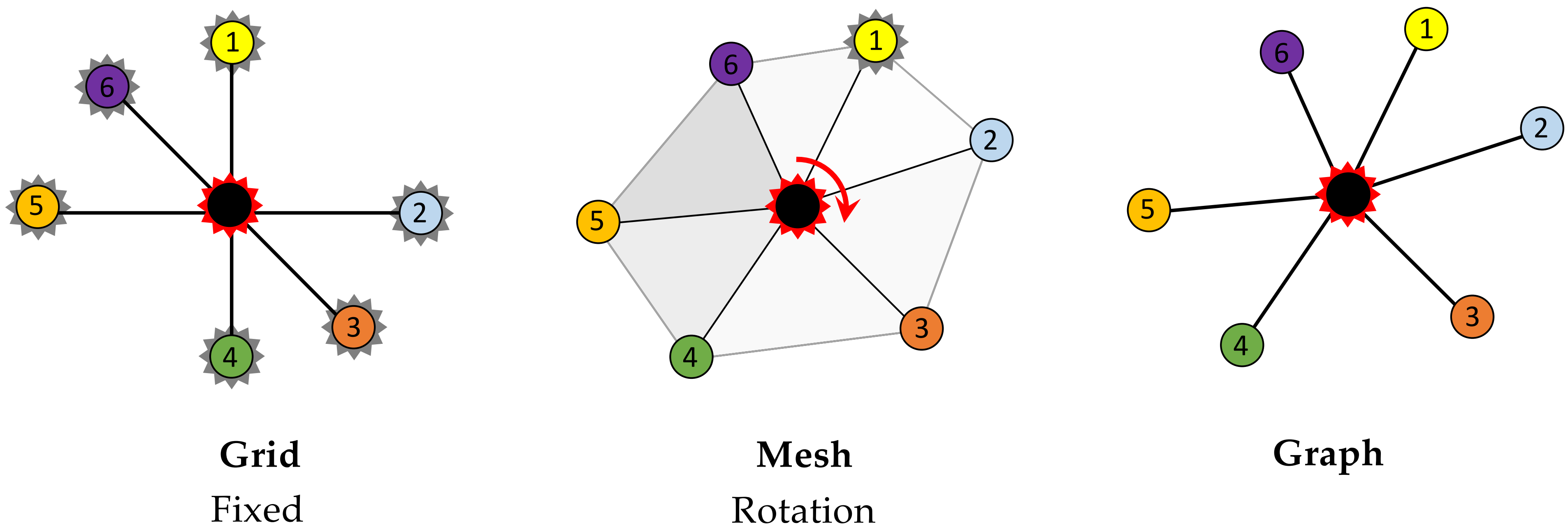


Grid
Fixed

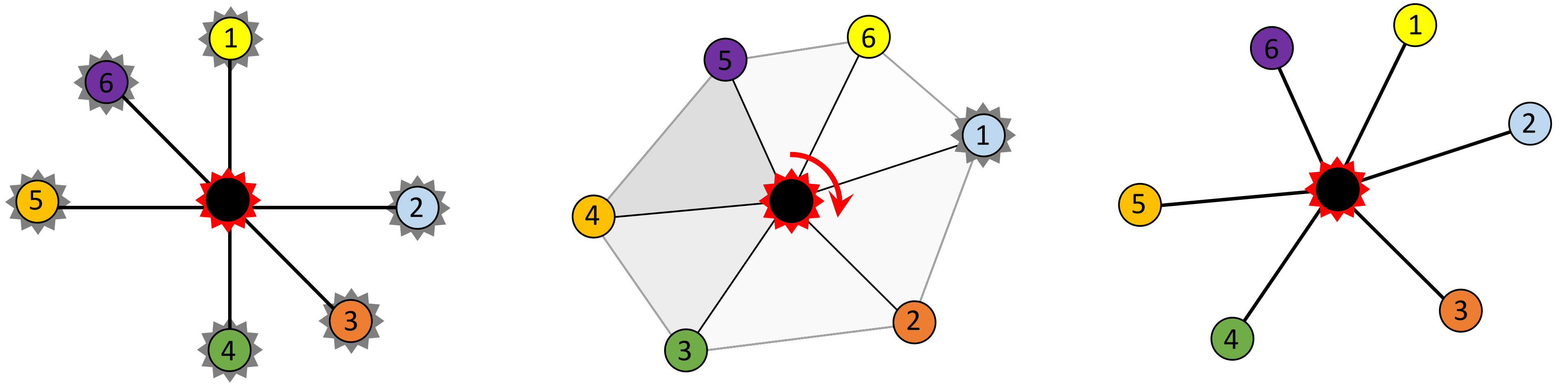
Mesh

Graph

Graphs vs Meshes vs Grids



Graphs vs Meshes vs Grids



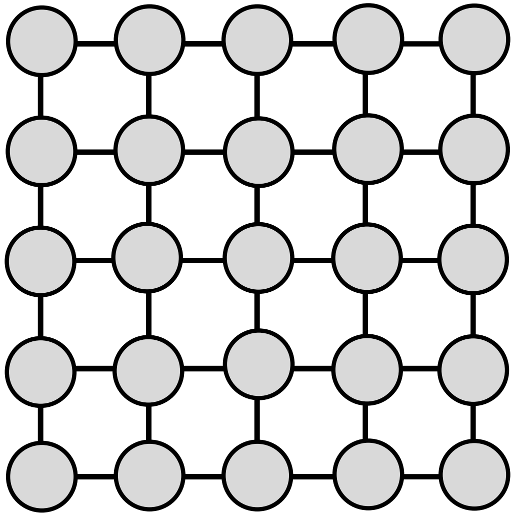
Grid
Fixed

Mesh
Rotation

Graph
Permutation

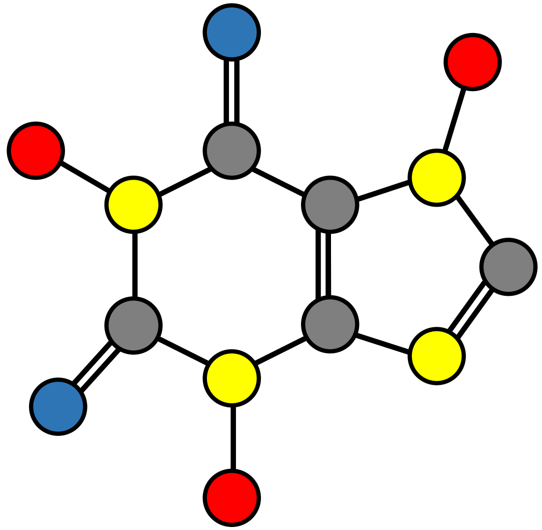
Graphs have the least structure

Graphs vs Meshes vs Grids



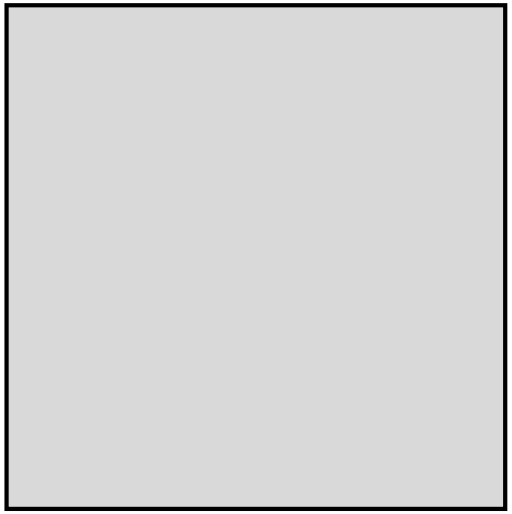
Grid

Mesh



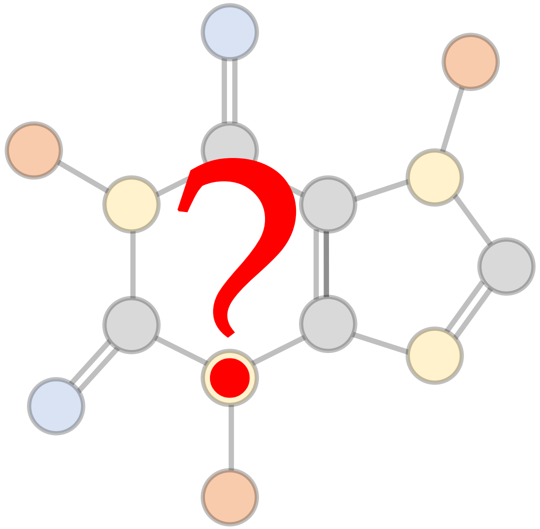
Graph

Graphs vs Meshes vs Grids



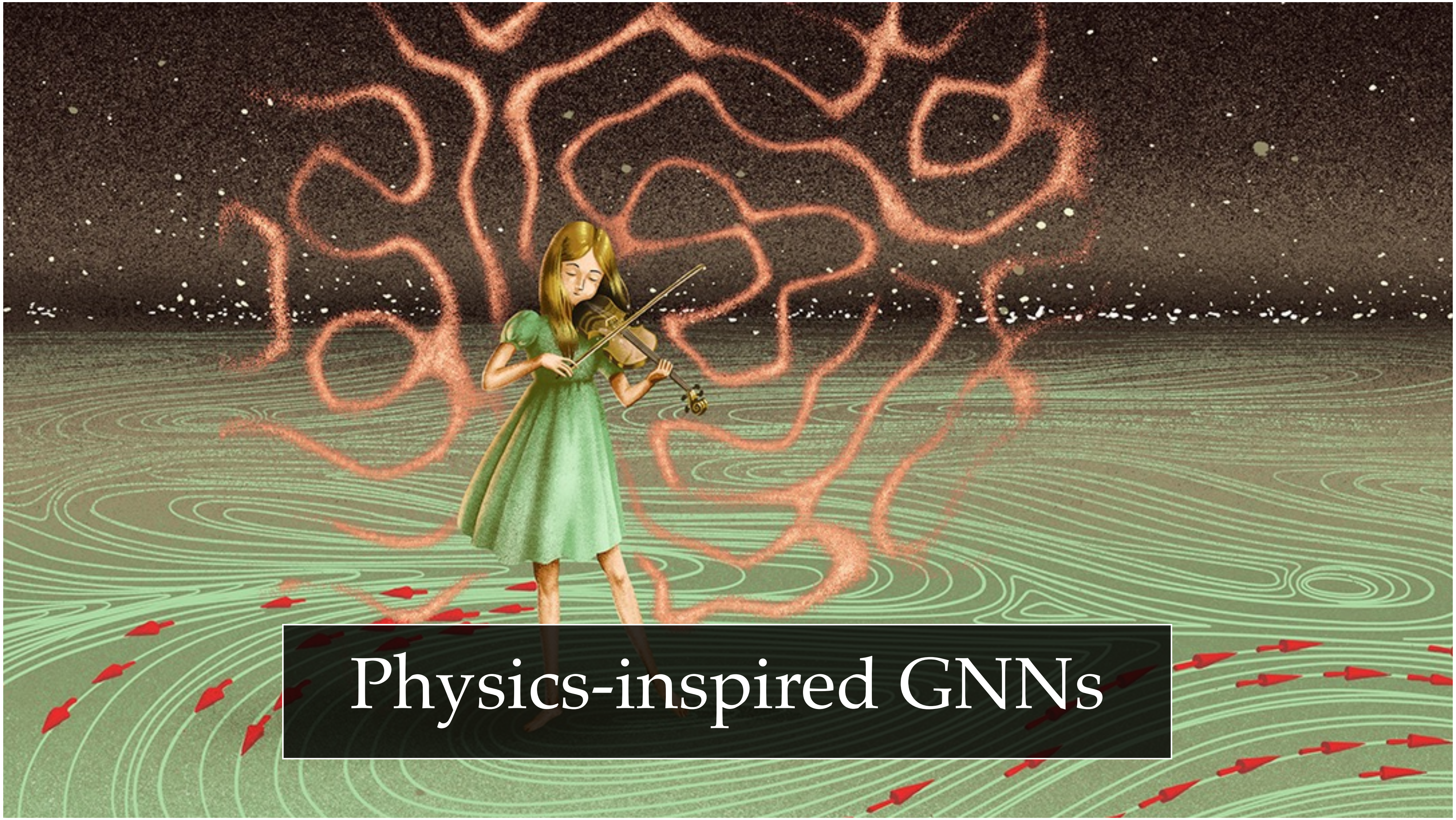
Grid

Mesh



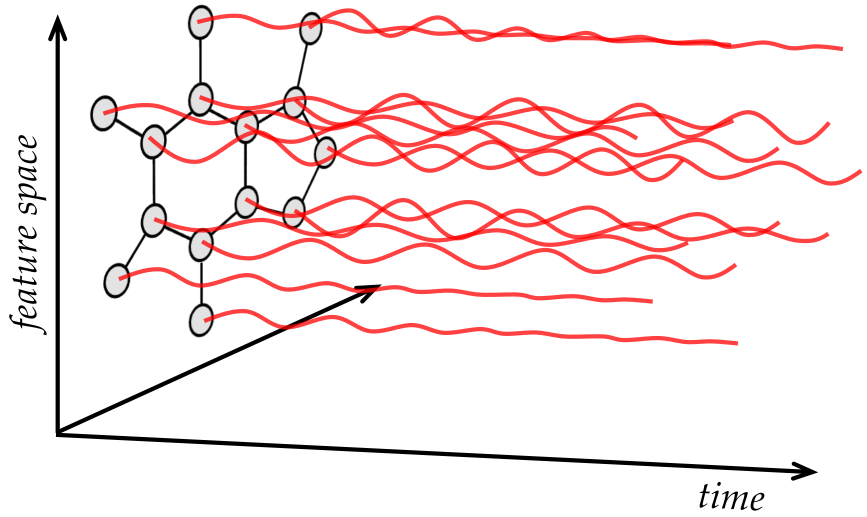
Graph

Continuous models for GNNs?


Physics-inspired GNNs

Physical metaphor of Graph ML

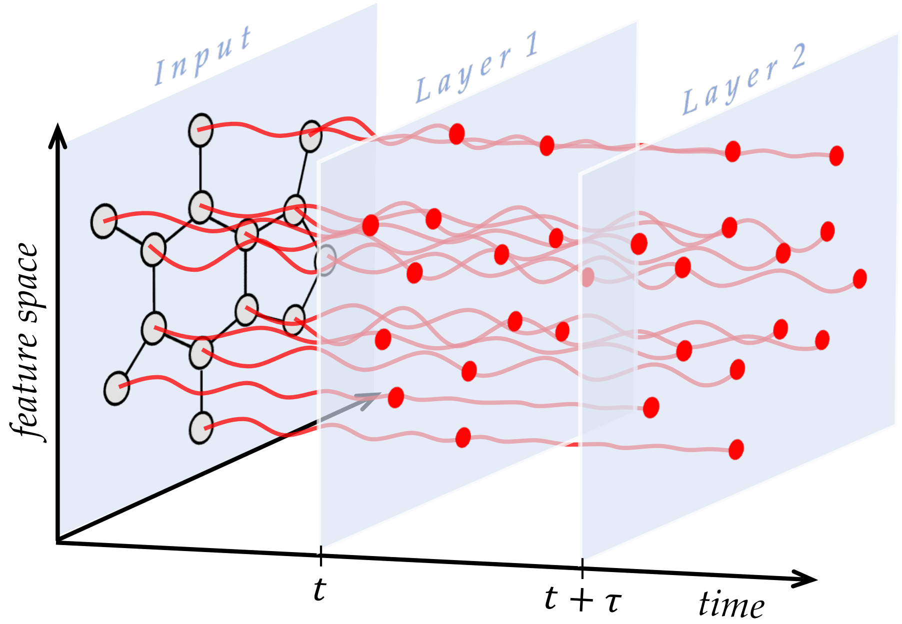
GNN = dynamic system



$$\dot{\mathbf{X}}(t) = \mathbf{F}_{\boldsymbol{\theta}(t)}(\mathbf{X}(t), \mathcal{G})$$

Haber, Ruthotto 2017; Chen et al. 2019 (Neural ODEs); Xhonneau et al. 2020 (CGNN); Chamberlain, Rowbottom, et B. 2021 (GRAND, BLEND)
Eliasof, Haber 2021 (PDE-GCN); Di Giovanni, Rowbottom et B 2022 (GRAFF), Rusch et B 2022 (GraphCON)

Physical metaphor of Graph ML



GNN = dynamic system

layers = discretisation of time

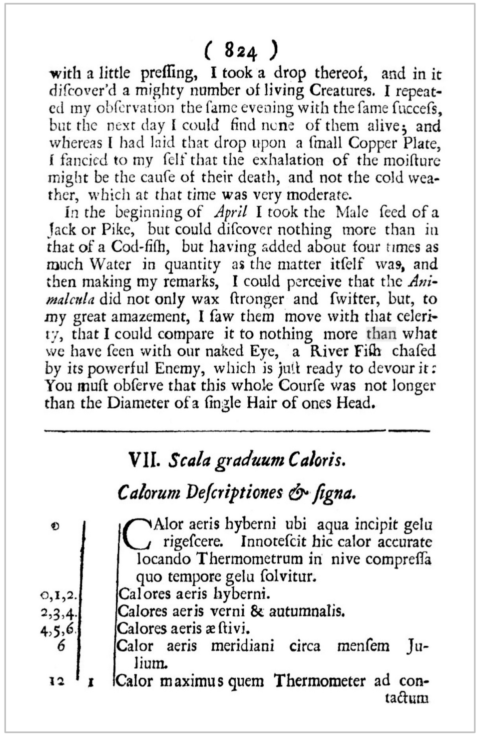
graph = coupling function
(discretisation of space)

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \tau \mathbf{F}_{\boldsymbol{\theta}(t)}(\mathbf{X}(t), \mathcal{G})$$

Haber, Ruthotto 2017; Chen et al. 2019 (Neural ODEs); Xhonneau et al. 2020 (CGNN); Chamberlain, Rowbottom, et B. 2021 (GRAND, BLEND)
Eliasof, Haber 2021 (PDE-GCN); Di Giovanni, Rowbottom et B 2022 (GRAFF), Rusch et B 2022 (GraphCON)

Heat Diffusion

Newton Law of Cooling: “the [temperature] a hot body loses in a given time is proportional to the temperature difference between the object and the environment”



(824)

with a little preſſing, I took a drop thereof, and in it diſcover'd a mighty number of living Creatures. I repeated my obſervation the ſame evening with the ſame ſucceſs, but the next day I could find none of them alive; and whereas I had laid that drop upon a ſmall Copper Plate, I fancied to my ſelf that the exhalation of the moiſture might be the cauſe of their death, and not the cold weather, which at that time was very moderate.

In the beginning of *April* I took the Male ſeed of a Jack or Pike, but could diſcover nothing more than in that of a Cod-fiſh, but having added about four times as much Water in quantity as the matter itſelf was, and then making my remarks, I could perceive that the *Animalcula* did not only wax ſtronger and ſwifter, but, to my great amazement, I ſaw them move with that celerity, that I could compare it to nothing more than what we have ſeen with our naked Eye, a River Fiſh chaſed by its powerful Enemy, which is juſt ready to devour it: You muſt obſerve that this whole Courſe was not longer than the Diameter of a ſingle Hair of ones Head.

VII. *Scala graduum Caloris.*

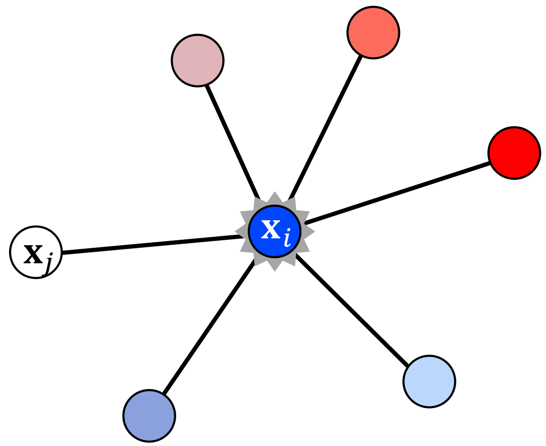
Calorum Deſcriptiones & ſigna.

0		Calor aeris hyberni ubi aqua incipit gelu rigeſcere. Innoteſcit hic calor accurate locando Thermometrum in nive compreſſa quo tempore gelu ſolvitur.
0,1,2.		Calores aeris hyberni.
2,3,4.		Calores aeris verni & autumnalis.
4,5,6.		Calores aeris æſtivi.
6		Calor aeris meridiani circa menſem Julium.
12	1	Calor maximus quem Thermometer ad contactum

Isaac Newton

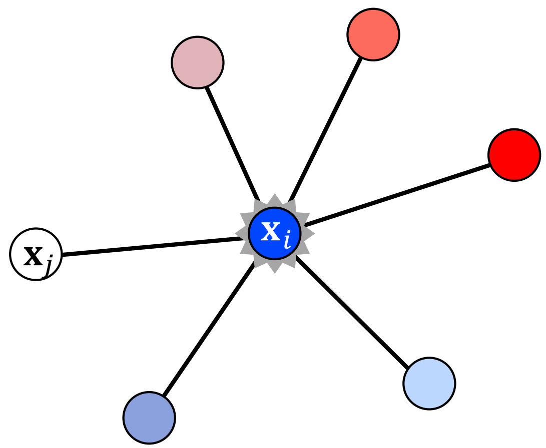
Anonymous 1701

Heat Diffusion Equation on Graphs



$$\dot{\mathbf{x}}_i(t) = \mathbf{x}_i(t) - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} a_{ij} \mathbf{x}_j(t)$$

Heat Diffusion Equation on Graphs



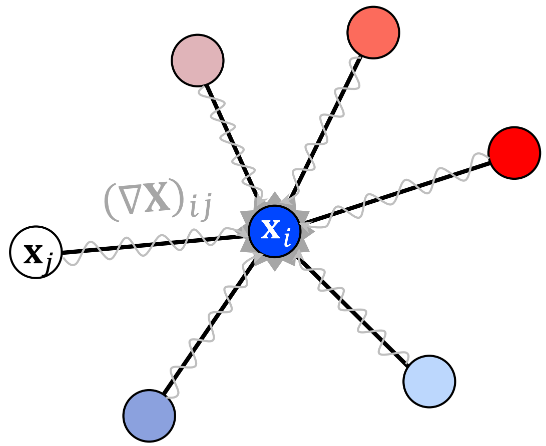
$$\dot{\mathbf{x}}_i(t) = \mathbf{x}_i(t) - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} a_{ij} \mathbf{x}_j(t)$$

rate of temperature change

self temperature

temperature of the environment

Heat Diffusion Equation on Graphs

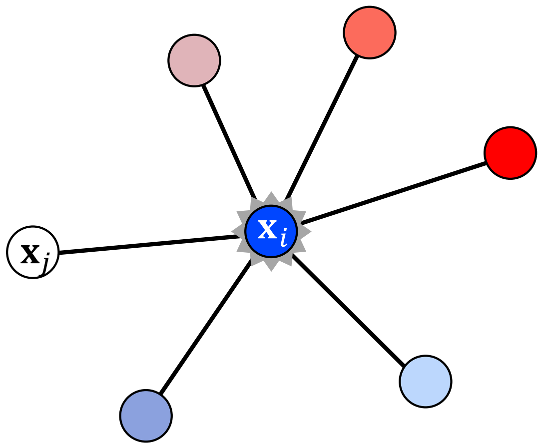


$$\dot{\mathbf{x}}_i(t) = \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} a_{ij} \left(\mathbf{x}_i(t) - \mathbf{x}_j(t) \right)$$

divergence
div

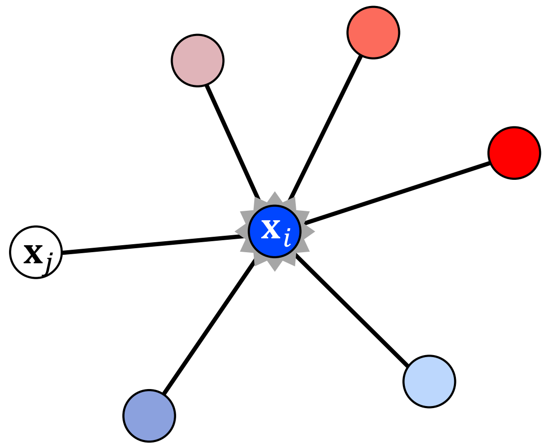
gradient
$-(\nabla \mathbf{X})_{ij}$

Heat Diffusion Equation on Graphs



$$\dot{\mathbf{X}}(t) = -\mathrm{div}\big(\nabla \mathbf{X}(t)\big)$$

Heat Diffusion Equation on Graphs



$$\dot{\mathbf{X}}(t) = \Delta \mathbf{X}(t)$$

Heat Equation as a prototypical Gradient Flow

$$\dot{\mathbf{X}}(t) = -\nabla \mathcal{E}(\mathbf{X}(t))$$

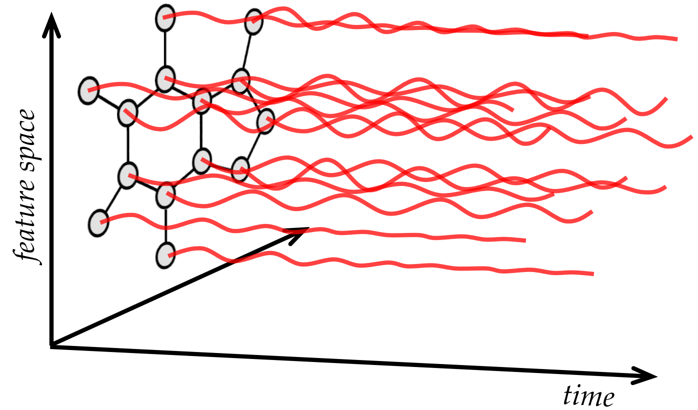
$$\mathcal{E}_{\mathrm{DIR}}(\mathbf{X}) = \frac{1}{2} \sum_{j \in \mathcal{N}_i} \left\| (\nabla \mathbf{X})_{ij} \right\|^2 = \frac{1}{2} \mathrm{trace}\left(\mathbf{X}^{\mathrm{T}} \Delta \mathbf{X}\right)$$

G. Dirichlet

- Heat diffusion equation is the gradient flow of the Dirichlet energy
- "Smoothness" of the node features
- Dirichlet energy decreases along the flow
- In the limit $t \to \infty$ results in "oversmoothing"
- Not very expressive: works only in homophilic graphs ("similar neighbours")

Zhou, Schölkopf 2005 (label propagation); Rossi et B 2021 (feature propagation)

Gradient Flow Framework (GRAFF)



Traditional GNNs

$$\dot{\mathbf{X}}(t) = \mathbf{F}_{\boldsymbol{\theta}(t)}(\mathbf{X}(t), \mathcal{G})$$

- Parametrize evolution equations

GRAFF

$$\dot{\mathbf{X}}(t) = -\nabla \mathcal{E}_{\boldsymbol{\theta}(t)}(\mathbf{X}(t), \mathcal{G})$$

- Parametrize energy
- Derive evolution equation as GF
- Better interpretability

Di Giovanni, Rowbottom et B 2022

Generalised parametric Dirichlet energy

$$\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{X}) = \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{x}_i, \boldsymbol{\Omega}\mathbf{x}_i\rangle - \frac{1}{2}\sum_{j\in\mathcal{N}_i}\bar{a}_{ij}\langle \mathbf{x}_i, \mathbf{W}\mathbf{x}_j\rangle$$

- Energy of a system of particles (nodes) parameterised by $d\times d$ matrices $\boldsymbol{\Omega}$ and $\mathbf{W}$

Di Giovanni, Rowbottom et B 2022

Generalised parametric Dirichlet energy

$$\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{X}) = \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{x}_i, \boldsymbol{\Omega}\mathbf{x}_i\rangle - \frac{1}{2}\sum_{j\in\mathcal{N}_i}\bar{a}_{ij}\langle \mathbf{x}_i, \mathbf{W}\mathbf{x}_j\rangle$$

external energy

- Energy of a system of particles (nodes) parameterised by $d\times d$ matrices $\boldsymbol{\Omega}$ and $\mathbf{W}$
- External energy term acting on all particles

Di Giovanni, Rowbottom et B 2022

Generalised parametric Dirichlet energy

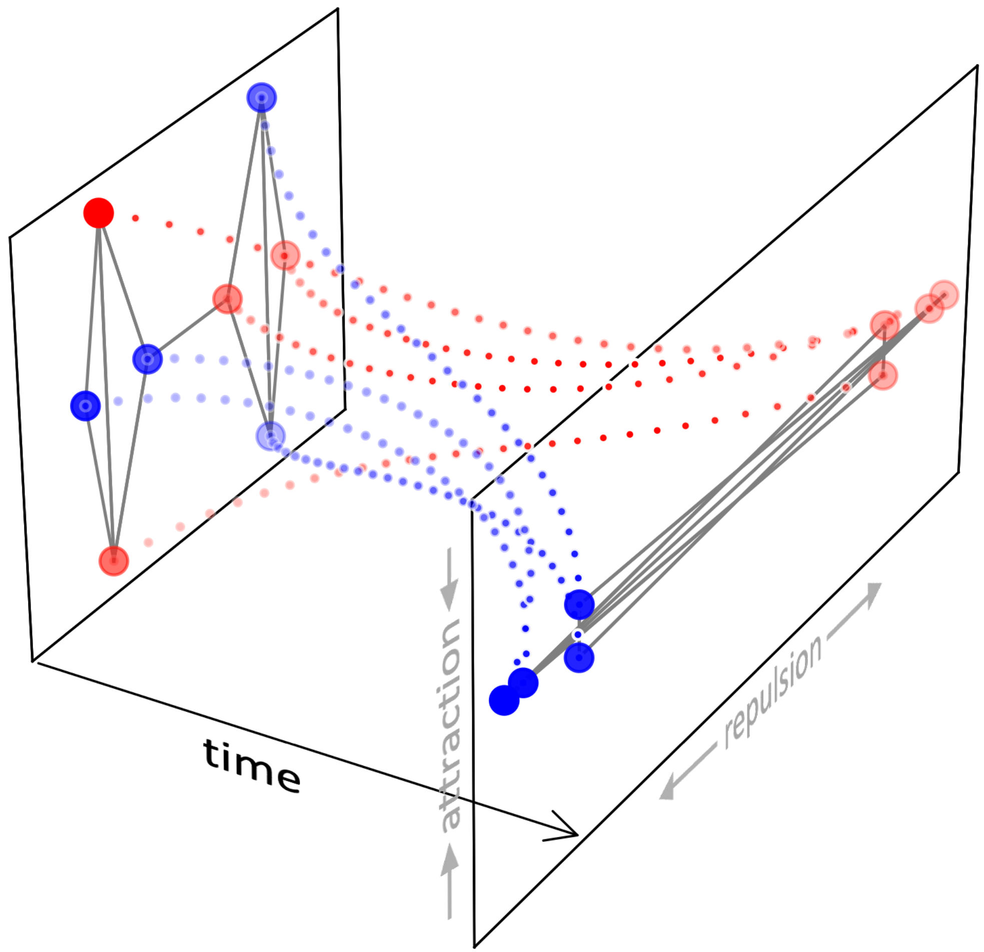
$$\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{X}) = \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{x}_i, \boldsymbol{\Omega}\mathbf{x}_i\rangle - \frac{1}{2}\sum_{j\in\mathcal{N}_i}\bar{a}_{ij}\langle \mathbf{x}_i, \mathbf{W}\mathbf{x}_j\rangle$$

external energy

internal energy
(pair-wise interactions)

- Energy of a system of particles (nodes) parameterised by $d\times d$ matrices $\boldsymbol{\Omega}$ and $\mathbf{W}$
- External energy term acting on all particles
- Internal energy term: interactions between nodes along edges of the graph
  - **Attractive** interactions along positive eigenvectors of $\mathbf{W}$
  - **Repulsive** interactions along negative eigenvectors of $\mathbf{W}$

Di Giovanni, Rowbottom et B 2022



Di Giovanni, Rowbottom et B 2022

Gradient Flow of $\mathcal{E}_{\boldsymbol{\theta}}$

$$\dot{\mathbf{X}}(t) = -\nabla\mathcal{E}_{\boldsymbol{\theta}}\big(\mathbf{X}(t)\big) = -\mathbf{X}(t)\frac{\boldsymbol{\Omega} + \boldsymbol{\Omega}^{\mathrm{T}}}{2} + \bar{\mathbf{A}}\mathbf{X}(t)\frac{\mathbf{W} + \mathbf{W}^{\mathrm{T}}}{2}$$

Di Giovanni, Rowbottom et B 2022

Gradient Flow of $\mathcal{E}_{\boldsymbol{\theta}}$

$$\dot{\mathbf{X}}(t) = -\nabla \mathcal{E}_{\boldsymbol{\theta}}\big(\mathbf{X}(t)\big) = -\mathbf{X}(t)\frac{\boldsymbol{\Omega} + \boldsymbol{\Omega}^{\mathrm{T}}}{2} + \bar{\mathbf{A}}\mathbf{X}(t)\frac{\mathbf{W} + \mathbf{W}^{\mathrm{T}}}{2}$$

matrices appear only in symmetrized form

Di Giovanni, Rowbottom et B 2022

Gradient Flow of $\mathcal{E}_{\boldsymbol{\theta}}$

$$\dot{\mathbf{X}}(t) = -\mathbf{X}(t)\boldsymbol{\Omega} + \bar{\mathbf{A}}\mathbf{X}(t)\mathbf{W}$$

- Symmetric matrices $\boldsymbol{\Omega}$ and $\mathbf{W}$
- Time-independent parameters: $\boldsymbol{\Omega}(t) = \boldsymbol{\Omega}$, $\mathbf{W}(t) = \mathbf{W}$

Di Giovanni, Rowbottom et B 2022

Discretised Gradient Flow of $\mathcal{E}_{\boldsymbol{\theta}}$

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \tau(-\mathbf{X}(t)\boldsymbol{\Omega} + \bar{\mathbf{A}}\mathbf{X}(t)\mathbf{W})$$

- Residual convolutional-type GNN
- Symmetric weights
- Symmetry constraint does not diminish expressive power
- Shared weights across layers

Di Giovanni, Rowbottom et B 2022; Xu, Zagoruyko, Komodakis 2019 (universal approximation in NNs with symmetric weights)

GRAFF

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \tau\bar{\mathbf{A}}\mathbf{X}(t)\mathbf{W}$$

- Residual
- Symmetric weights
- Shared weights
- No nonlinear activation

- Gradient flow (interpretability)
- $d^2/2$ weights
- Can induce both low- and high-frequency dominated dynamics
- Works with heterophilic graphs

Di Giovanni, Rowbottom et B 2022

GCN

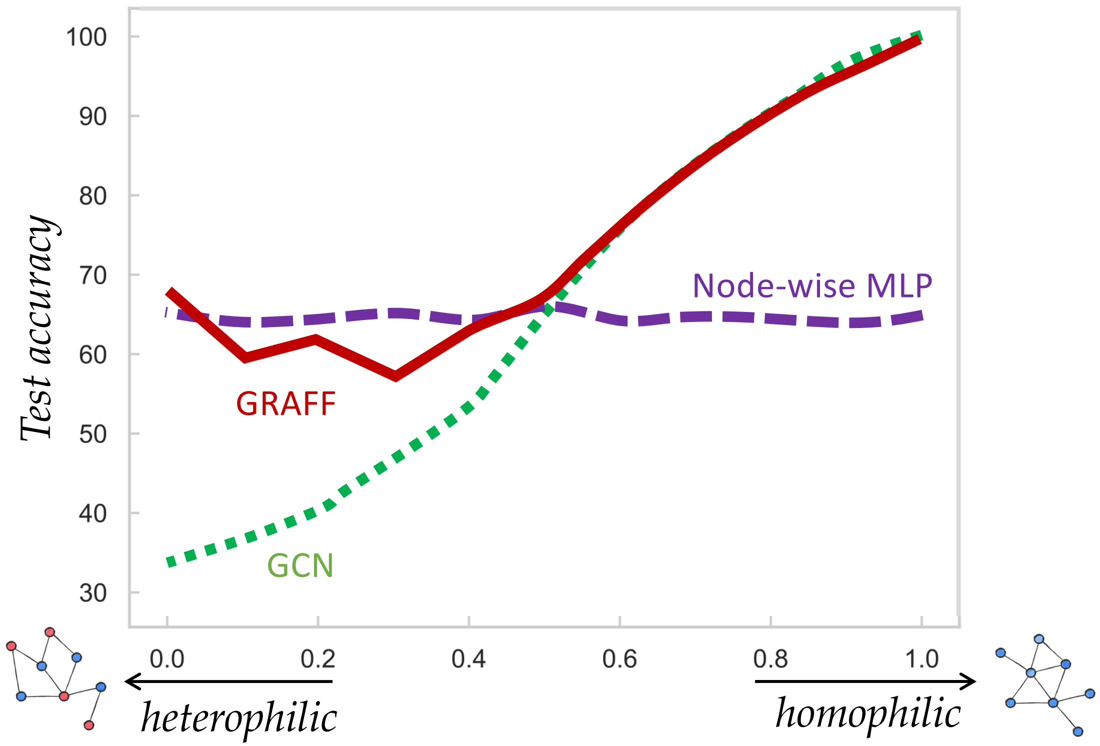
$$\mathbf{X}(t+\tau) = \tau\sigma\left(\bar{\mathbf{A}}\mathbf{X}(t)\mathbf{W}(t)\right)$$

- Non-residual
- Non-symmetric weights
- Different weights per layer
- Nonlinear activation

- Not a gradient flow
- Ld^2 weights
- Only low-frequency dominated dynamics (oversmoothing)
- Only homophilic graphs

Kipf, Welling 2017

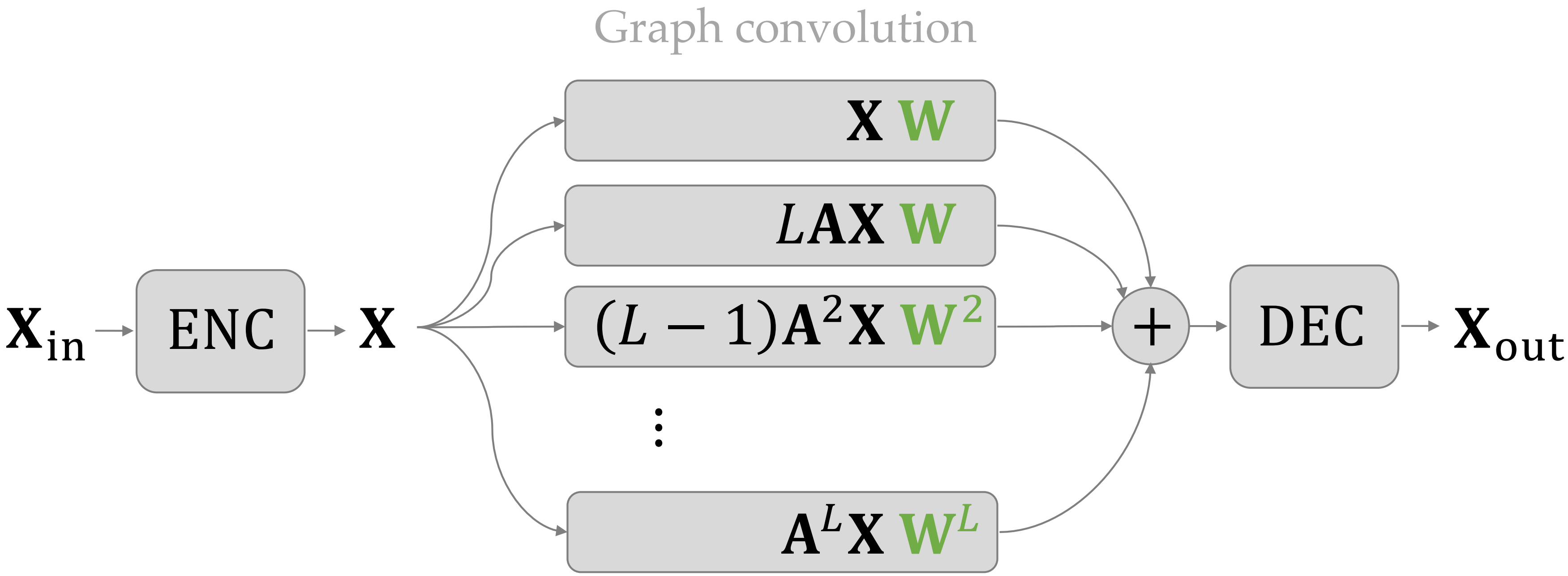
Homophily vs Heterophily



Synthetic Cora node classification task

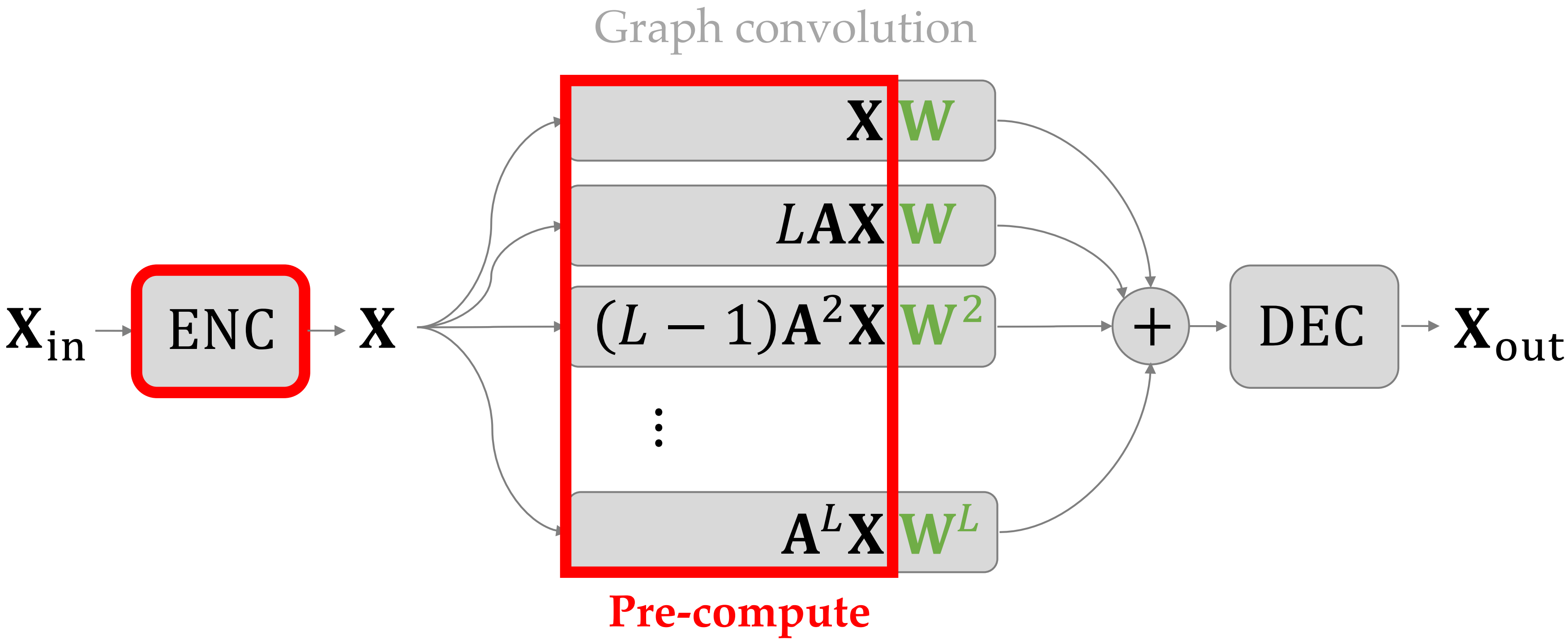
Di Giovanni, Rowbottom et B 2022

Scalability



Rossi, Frasca et B 2020 (SIGN)

Scalability



Rossi, Frasca et B 2020 (SIGN)

Spectral analysis of GRAFF

- $\mathbf{\Delta} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{\mathrm{T}}$ orthogonal eigendecomposition of the graph Laplacian
- $\mathbf{W} = \mathbf{\Psi}\mathbf{M}\mathbf{\Psi}^{\mathrm{T}}$ orthogonal eigendecomposition of channel mixing weights
- Output of L layers GRAFF

$$\mathbf{X}(L\tau) = \sum_{k=1}^{d}\sum_{l=0}^{n-1}\big(1 + \tau\mu_k(1 - \lambda_l)\big)^L \langle \mathbf{X}(0), \boldsymbol{\psi}_l \otimes \boldsymbol{\phi}_l \rangle \boldsymbol{\psi}_l \otimes \boldsymbol{\phi}_l$$

  - **Low frequencies** ($\lambda_l < 1$) magnified by **positive** eigenvalues of **W** ($\mu_k > 0$)
  - **High frequencies** ($\lambda_l > 1$) magnified by **negative** eigenvalues of **W** ($\mu_k < 0$)

Di Giovanni, Rowbottom et B 2022

Spectral analysis of GRAFF

- $\mathbf{\Delta} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{\mathrm{T}}$ orthogonal eigendecomposition of the graph Laplacian
- $\mathbf{W} = \mathbf{\Psi}\mathbf{M}\mathbf{\Psi}^{\mathrm{T}}$ orthogonal eigendecomposition of channel mixing weights
- Output of L layers GRAFF

$$\mathbf{X}(L\tau) = \sum_{k=1}^{d} \sum_{l=0}^{n-1} \left(1 + \tau\mu_k(1 - \lambda_l)\right)^L \langle \mathbf{X}(0), \boldsymbol{\psi}_l \otimes \boldsymbol{\phi}_l \rangle \boldsymbol{\psi}_l \otimes \boldsymbol{\phi}_l$$

If eigenvalues of **W** are sufficiently negative, then GRAFF dynamics is high-frequency dominant: $\mathcal{E}_{\mathrm{DIR}}(\mathbf{X}(t)/\|\mathbf{X}(t)\|) \to \lambda_{\max}/2$

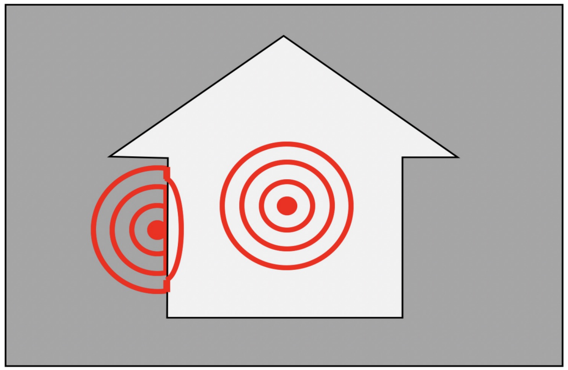
No oversmoothing!

Di Giovanni, Rowbottom et B 2022

Non-homogeneous Diffusion in Image Processing

$$\dot{\mathbf{X}}(t) = -\mathrm{div}\left(\frac{\nabla\mathbf{X}(t)}{1 + c\|\nabla\mathbf{X}(t)\|^2}\right)$$

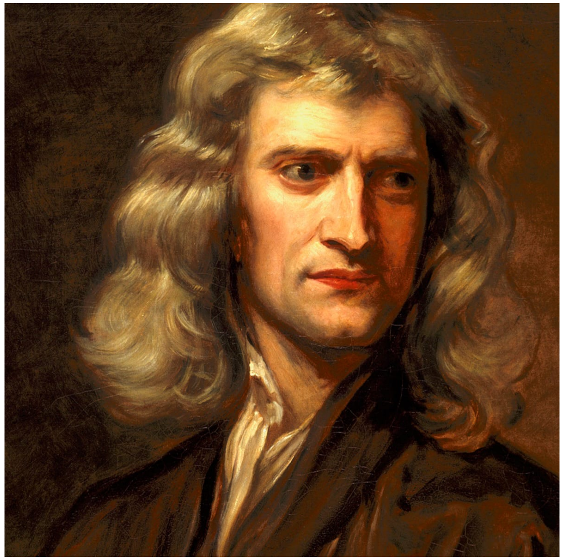
edge indicator

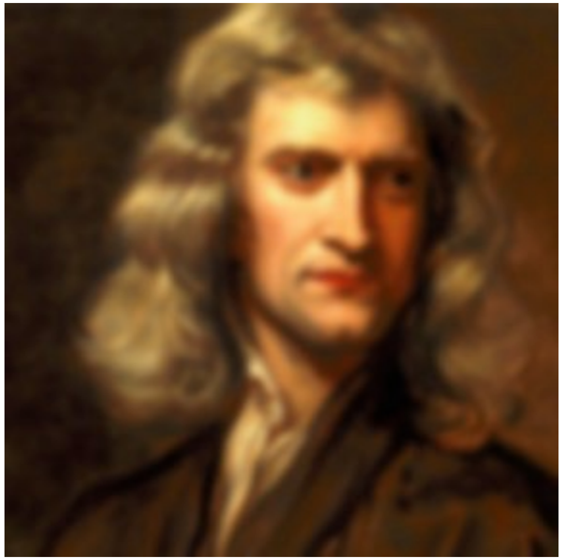


"Do not diffuse across edges"

Perona, Malik 1990

Non-homogeneous Diffusion in Image Processing



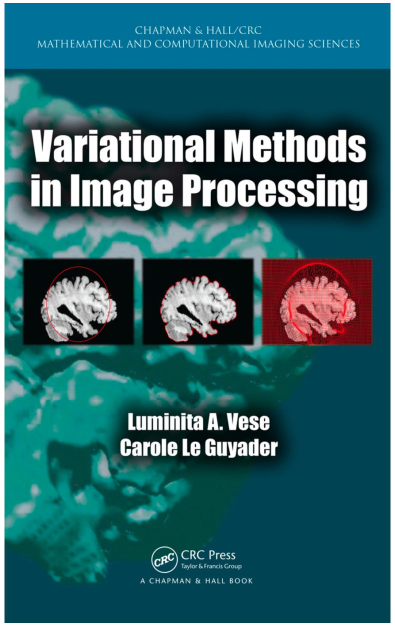


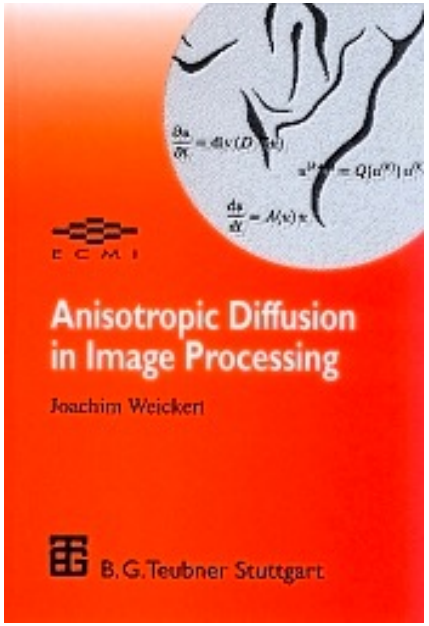
Homogeneous diffusion

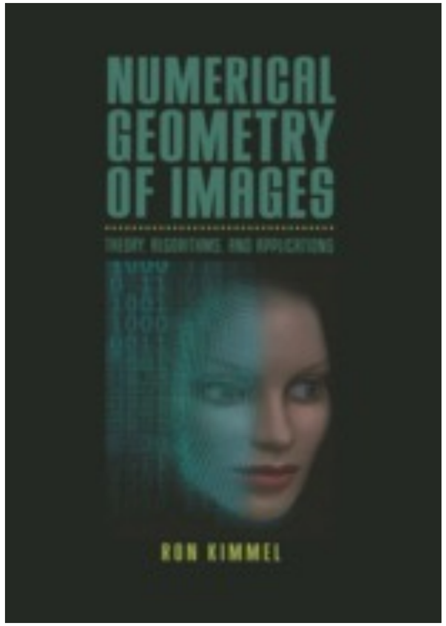
Non-homogeneous diffusion

Perona, Malik 1990; Kimmel et al. 1997; Sochen et al. 1998; Tomasi, Manduchi 1998; Weickert 1998; Buades et al. 2005

Diffusion in Image Processing







Perona, Malik 1990; Kimmel et al. 1997; Sochen et al. 1998; Tomasi, Manduchi 1998; Weickert 1998; Buades et al. 2005

Non-homogeneous Diffusion Equation on Graphs

$$\dot{\mathbf{x}}_i(t) = \sum_{j \in \mathcal{N}_i} a\left(\mathbf{x}_i(t), \mathbf{x}_j(t)\right)\left(\mathbf{x}_j(t) - \mathbf{x}_i(t)\right)$$

Explicit (Forward Euler) discretization with timestep τ:

$$\mathbf{x}_i(t+\tau) = \mathbf{x}_i(t) + \tau \sum_{j \in \mathcal{N}_i} a\left(\mathbf{x}_i(t), \mathbf{x}_j(t)\right)\left(\mathbf{x}_j(t) - \mathbf{x}_i(t)\right)$$

Chamberlain, Rowbottom, et B. 2021

Non-homogeneous Diffusion Equation on Graphs

$$\dot{\mathbf{x}}_i(t) = \sum_{j \in \mathcal{N}_i} a\left(\mathbf{x}_i(t), \mathbf{x}_j(t)\right)\left(\mathbf{x}_j(t) - \mathbf{x}_i(t)\right)$$

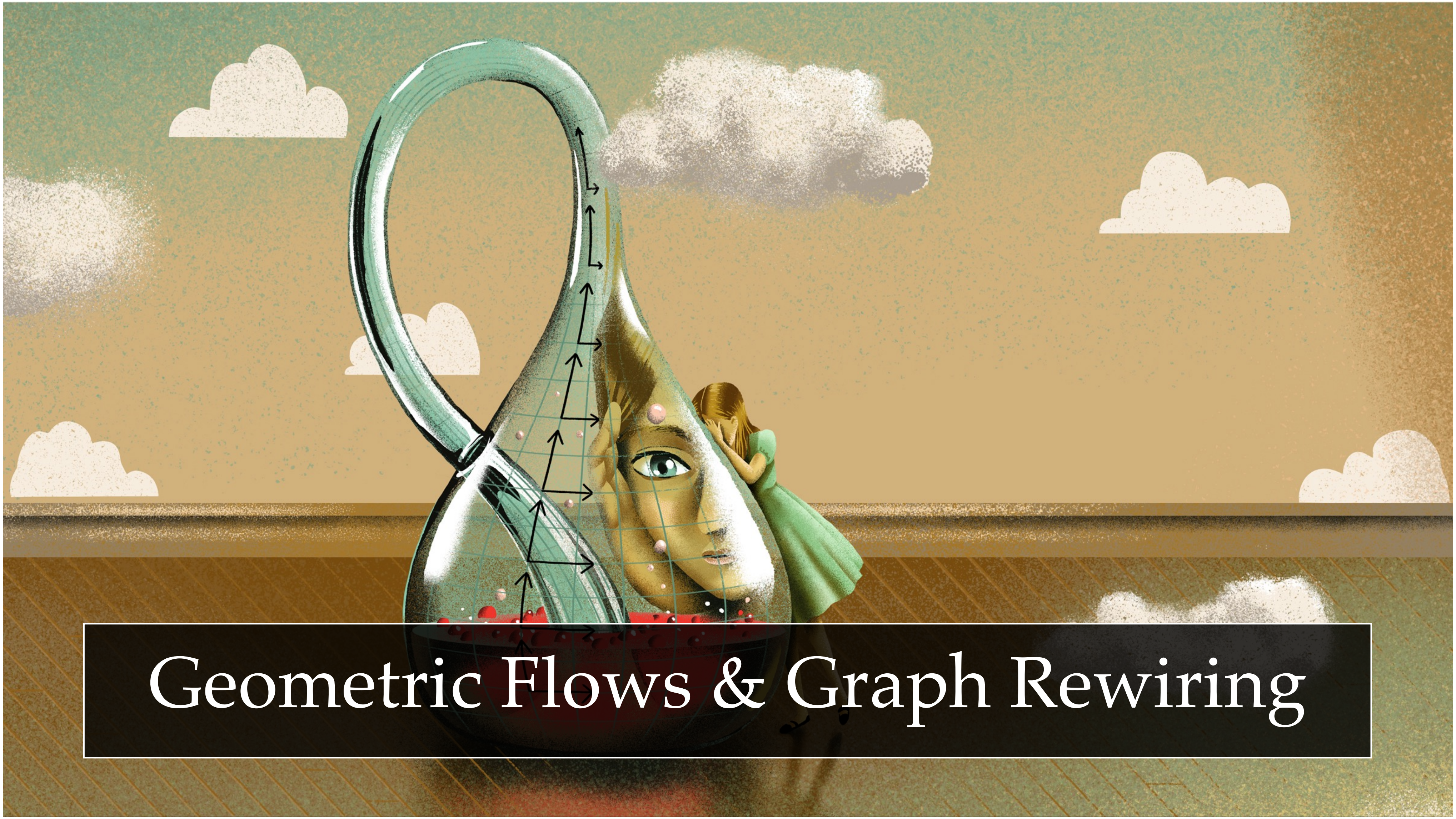
Explicit (Forward Euler) discretization with timestep τ:

$$\mathbf{x}_i(t+\tau) = \sum_{j \in \mathcal{N}_i} a\left(\mathbf{x}_i(t), \mathbf{x}_j(t)\right)\mathbf{x}_j(t)$$

normalised $\sum_j a_{ij} = 1$
unit step $\tau = 1$

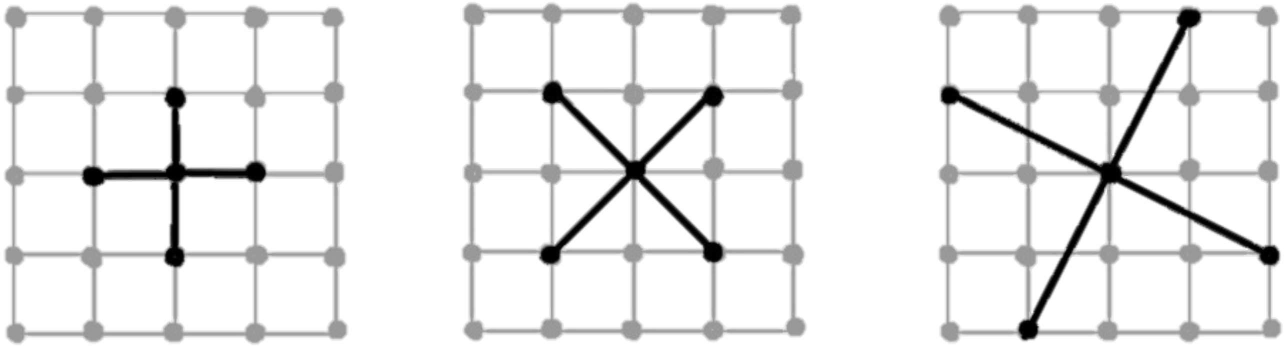
GAT is a particular discretisation of graph diffusion

Chamberlain, Rowbottom, et B. 2021



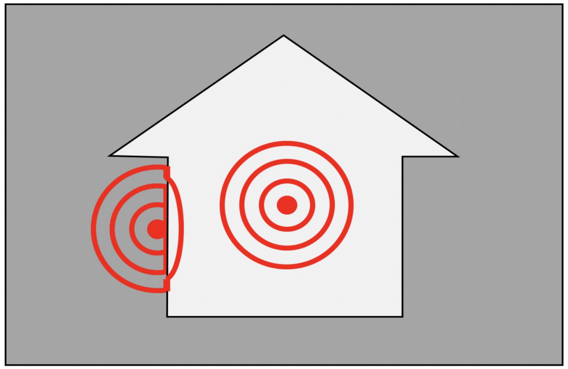
Geometric Flows & Graph Rewiring

Spatial Derivative: Graph Rewiring?



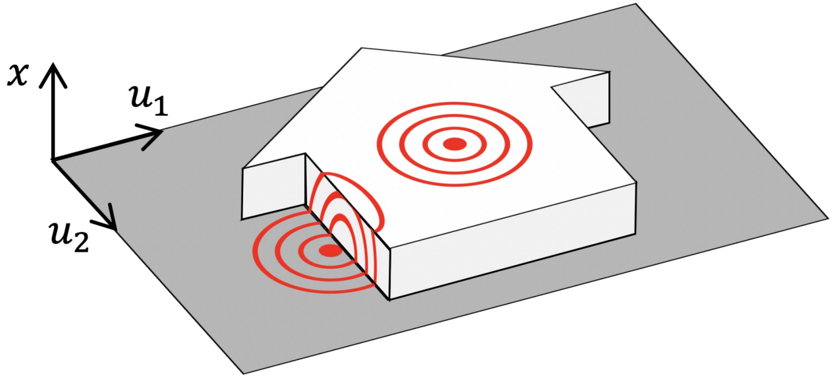
Different discretisations of 2D Laplacian

Images as embedded manifolds



$$\dot{\mathbf{X}} = -\mathrm{div}(a(\mathbf{X})\nabla\mathbf{X})$$

Non-linear diffusion



$$\dot{\mathbf{Z}} = \Delta_{\mathbf{G}}\mathbf{Z}$$

Non-Euclidean diffusion

Kimmel et al. 1997; Sochen et al. 1998

Beltrami flow

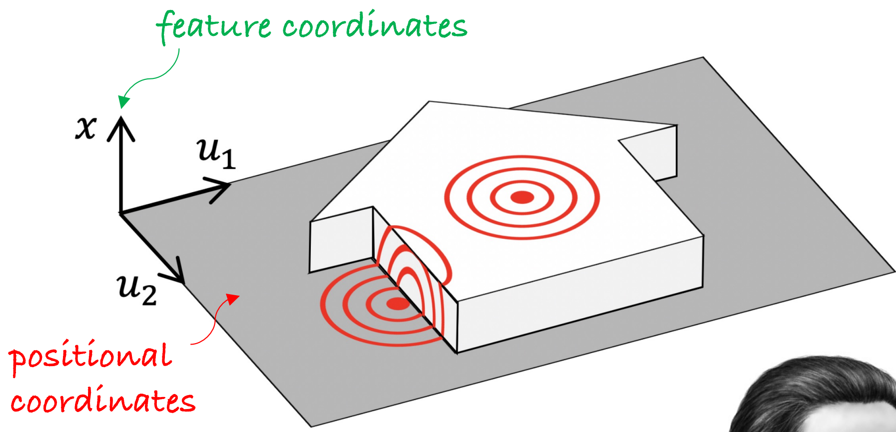
- Consider image as embedded 2-*manifold*

$$\mathbf{Z}(\mathbf{u}) = \big(\mathbf{u}, \alpha\mathbf{X}(\mathbf{u})\big)$$

- *Pullback metric:* 2×2 matrix

$$\mathbf{G} = \mathbf{I} + \alpha^2\big(\nabla_{\mathbf{u}}\mathbf{X}(\mathbf{u})\big)^\top \nabla_{\mathbf{u}}\mathbf{X}(\mathbf{u})$$

- *Beltrami flow* = gradient flow of the *Polyakov energy* (harmonic energy of the embedding used in string theory)



$$\dot{\mathbf{Z}} = \Delta_{\mathbf{G}}\mathbf{Z}$$

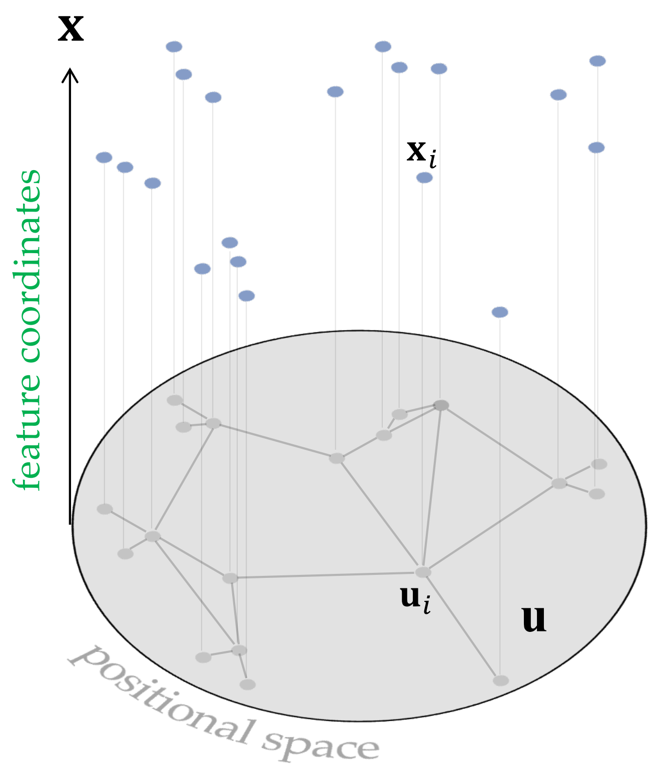
Eugenio Beltrami

Kimmel et al. 1997; Sochen et al. 1998

Graph Beltrami flow

- Graph with positional and feature node coordinates $\mathbf{z}_i = (\mathbf{u}_i, \mathbf{x}_i)$
- **Graph Beltrami flow**

$$\dot{\mathbf{z}}_i(t) = \sum_{j \in \mathcal{N}_i} a\left(\mathbf{z}_i(t), \mathbf{z}_j(t)\right)\left(\mathbf{z}_j(t) - \mathbf{z}_i(t)\right)$$



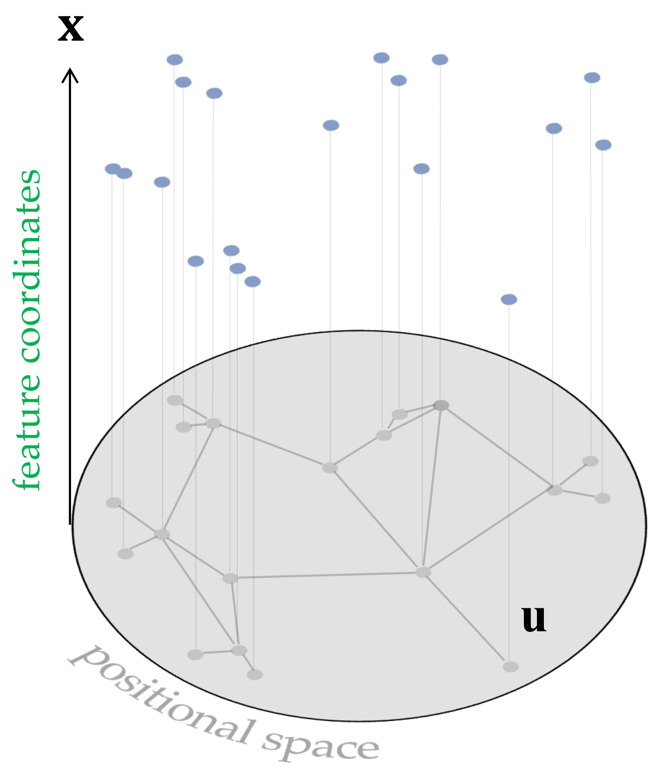
Chamberlain, Rowbottom, et B. 2021

Graph Beltrami flow

- Graph with positional and feature node coordinates $\mathbf{z}_i = (\mathbf{u}_i, \mathbf{x}_i)$
- **Graph Beltrami flow**

$$\dot{\mathbf{z}}_i(t) = \sum_{j \in \mathcal{N}_i} a\Big(\mathbf{z}_i(t), \mathbf{z}_j(t)\Big)\Big(\mathbf{z}_j(t) - \mathbf{z}_i(t)\Big)$$

  - Evolution of $\mathbf{x}$ = feature diffusion



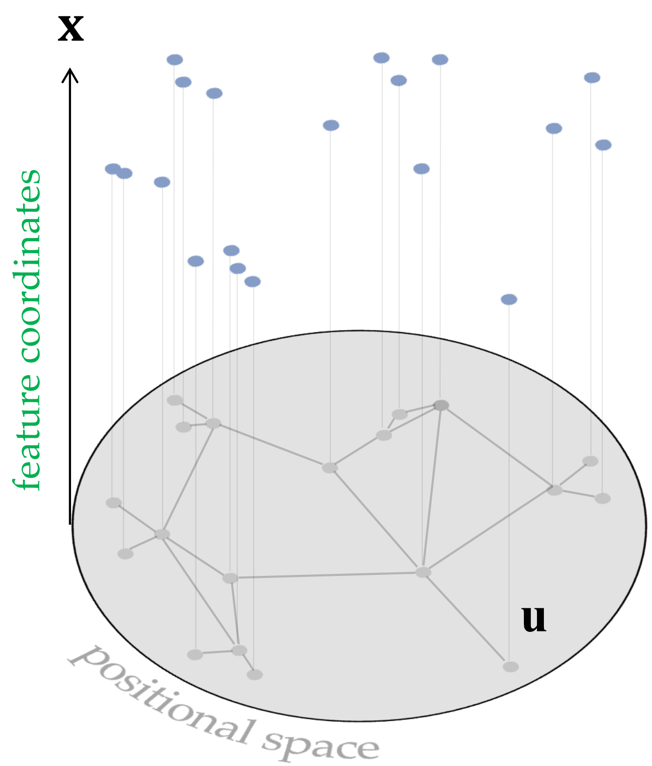
Chamberlain, Rowbottom, et B. 2021

Graph Beltrami flow

- Graph with positional and feature node coordinates $\mathbf{z}_i = (\mathbf{u}_i, \mathbf{x}_i)$
- **Graph Beltrami flow**

$$\dot{\mathbf{z}}_i(t) = \sum_{j \in \mathcal{N}_i} a\Big(\mathbf{z}_i(t), \mathbf{z}_j(t)\Big)\Big(\mathbf{z}_j(t) - \mathbf{z}_i(t)\Big)$$

  - Evolution of $\mathbf{x}$ = feature diffusion
  - Evolution of $\mathbf{u}$ = graph rewiring



Chamberlain, Rowbottom, et B. 2021

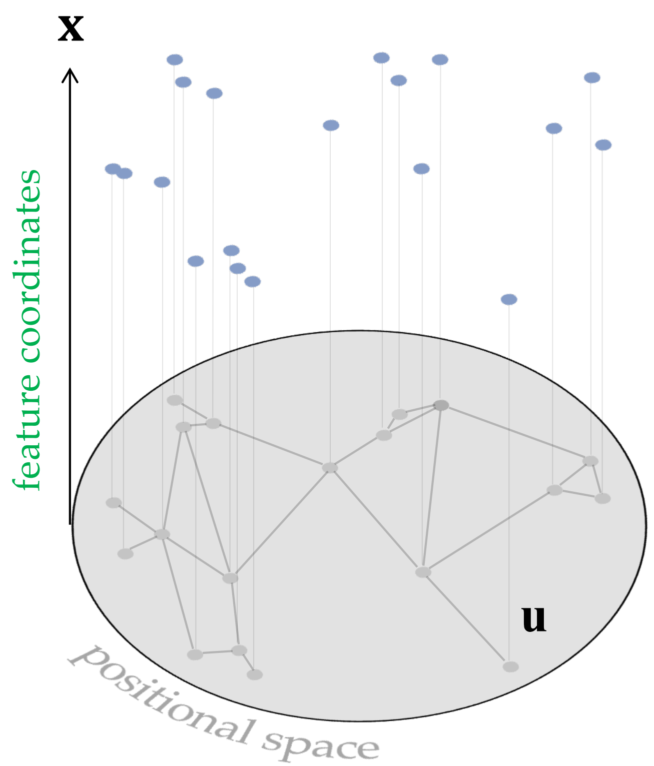
Graph Beltrami flow

- Graph with positional and feature node coordinates $\mathbf{z}_i = (\mathbf{u}_i, \mathbf{x}_i)$
- **Graph Beltrami flow**

$$\dot{\mathbf{z}}_i(t) = \sum_{j \in \mathcal{N}'_i} a\Big(\mathbf{z}_i(t), \mathbf{z}_j(t)\Big)\Big(\mathbf{z}_j(t) - \mathbf{z}_i(t)\Big)$$

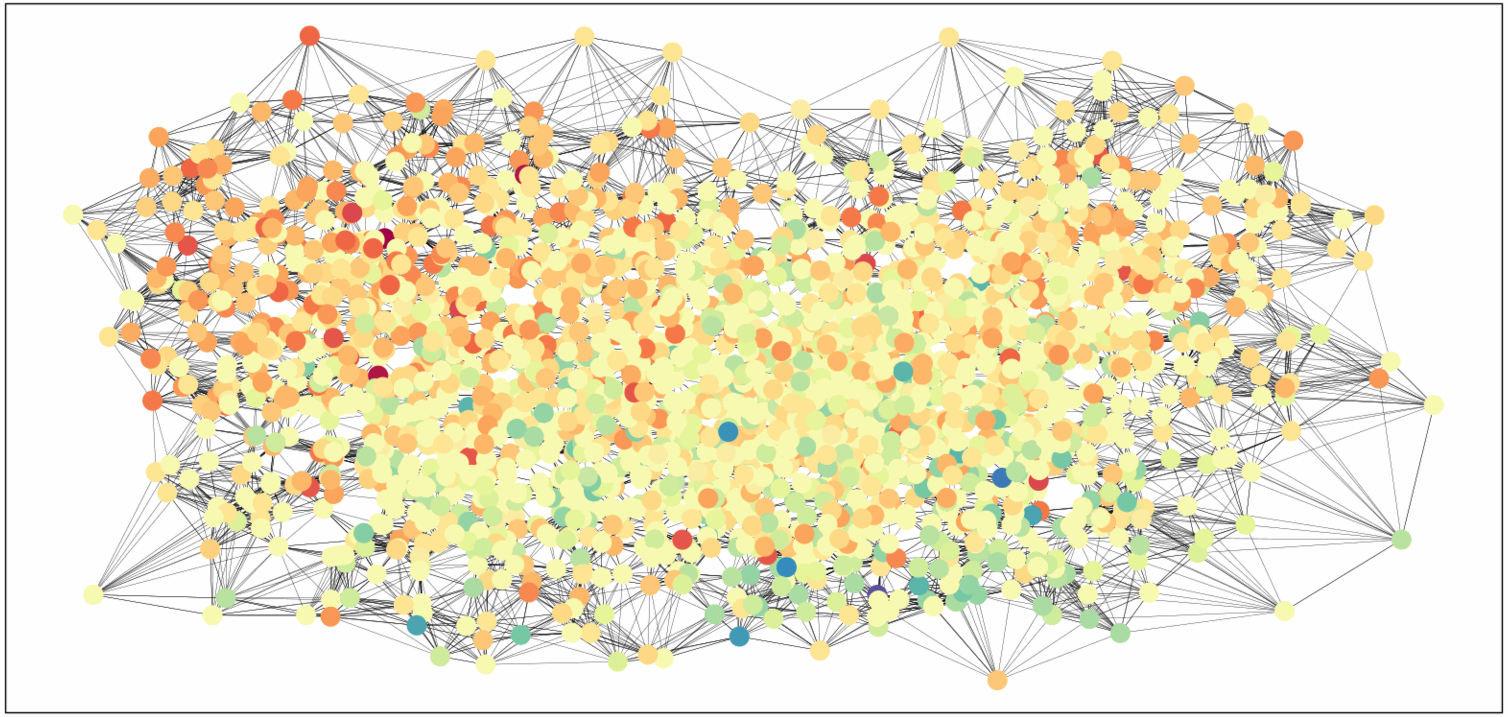
rewired graph

  - Evolution of $\mathbf{x}$ = feature diffusion
  - Evolution of $\mathbf{u}$ = graph rewiring



Chamberlain, Rowbottom, et B. 2021

Graph Beltrami flow



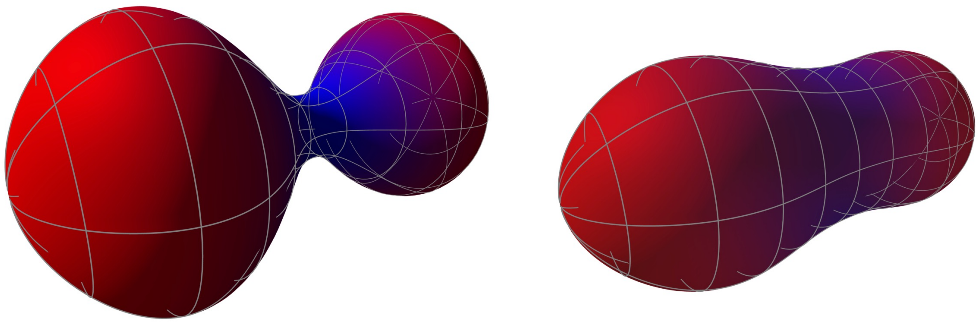
Evolution of positional/feature components + rewiring of the Cora graph

Chamberlain, Rowbottom, et B. 2021

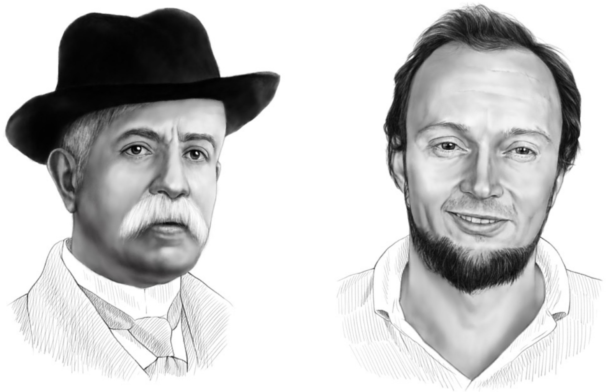
Ricci flow

- **Ricci flow:** "diffusion of the Riemannian metric"

$$\frac{\partial g_{ij}}{\partial t} = R_{ij}$$



Evolution of a manifold under Ricci flow



G. Ricci-Curbastro **R. Hamilton**

Ricci 1903; Hamilton 1988;

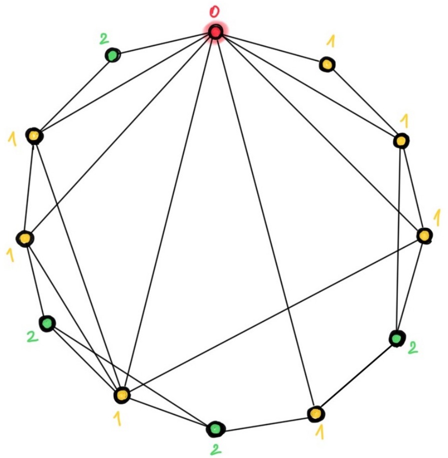
Ricci 1903; Hamilton 1988; Perelman 2003

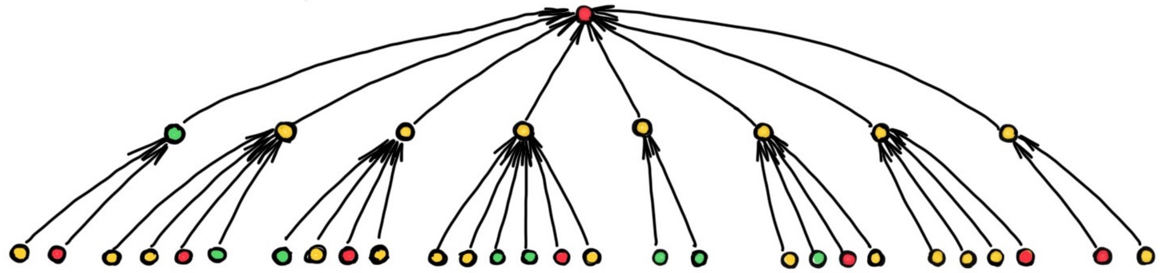
G. Perelman

G. Ricci-Curbastro

R. Hamilton

Over-squashing & Bottlenecks





In small-world graphs metric ball volume $\mathrm{vol}(B_k) = \sum_{j \in B_k} d_j$
grows exponentially with ball radius k

Long-distance dependency + Fast volume growth
= Over-squashing

Alon, Yahav 2020

Over-squashing

- Consider an MPNN of the form

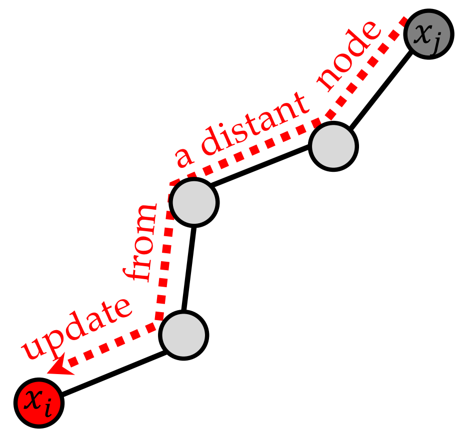
$$\mathbf{x}_i^{(k+1)} = \sigma\left(\mathbf{W}_1\mathbf{x}_i^{(k)} + \sum_j a_{ij}\,\mathbf{W}_2\mathbf{x}_j^{(k)}\right)$$

  - L = *depth* (number of layers)
  - p =*width* (hidden dimension)
  - Nonlinearity σ is c_σ-*Lipschitz-continuous*
  - w = maximum element of weight matrices $\mathbf{W}_1$, $\mathbf{W}_2$

Theorem (Sensitivity bound): For any $i, j \in V$

$$\left\|\frac{\partial \mathbf{x}_i^{(L)}}{\partial \mathbf{x}_j^{(0)}}\right\|_1 \leq (c_\sigma wp)^L(\mathbf{I}+\mathbf{A})_{ij}^L$$

Topping, Di Giovanni et B 2021; Di Giovanni et B 2023



Over-squashing: small Jacobian $\left\|\partial \mathbf{x}_i^{(L)}/\partial \mathbf{x}_j^{(0)}\right\|$ indicates poor information propagation from input node

Over-squashing

- Consider an MPNN of the form

$$\mathbf{x}_i^{(k+1)} = \sigma\left(\mathbf{W}_1\mathbf{x}_i^{(k)} + \sum_j a_{ij}\,\mathbf{W}_2\mathbf{x}_j^{(k)}\right)$$

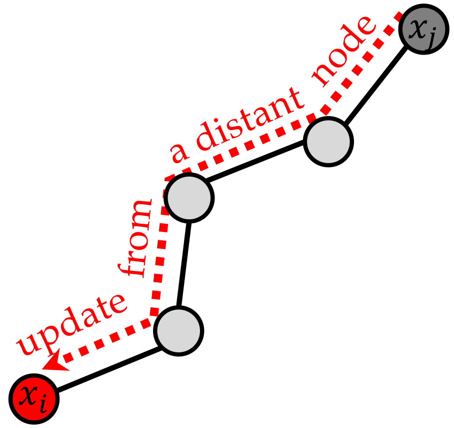
  - L = *depth* (number of layers)
  - p =*width* (hidden dimension)
  - Nonlinearity σ is c_σ-*Lipschitz-continuous*
  - w = maximum element of weight matrices $\mathbf{W}_1$, $\mathbf{W}_2$

Theorem (Sensitivity bound): For any $i, j \in V$

$$\left\|\frac{\partial \mathbf{x}_i^{(L)}}{\partial \mathbf{x}_j^{(0)}}\right\|_1 \leq (c_\sigma w p)^L (\mathbf{I} + \mathbf{A})_{ij}^L$$

model topology

Topping, Di Giovanni et B 2021; Di Giovanni et B 2023



Over-squashing: small Jacobian $\left\|\partial \mathbf{x}_i^{(L)} / \partial \mathbf{x}_j^{(0)}\right\|$ indicates poor information propagation from input node

Preventing over-squashing

$$\left\| \frac{\partial \mathbf{x}_i^{(L)}}{\partial \mathbf{x}_j^{(0)}} \right\|_1 \leq (c_\sigma w p)^L (\mathbf{I} + \mathbf{A})_{ij}^L$$

model topology

- **Width** p helps mitigate over-squashing (potentially at the risk of worse generalization)
- **Depth** L does not help
  - If $L \sim \mathrm{diam}(G)$, over-squashing occurs between distant nodes
  - If $L \gg 1$, we transition from over-squashing to vanishing gradients

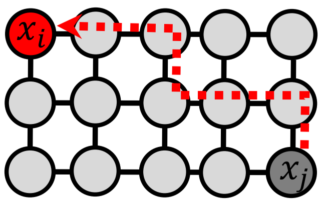
Di Giovanni et B 2023

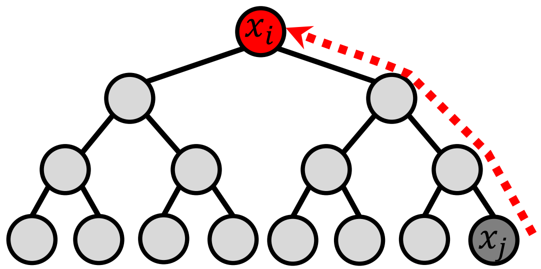
Preventing over-squashing

$$\left\|\frac{\partial \mathbf{x}_i^{(L)}}{\partial \mathbf{x}_j^{(0)}}\right\|_1 \leq (c_\sigma w p)^L (\mathbf{I} + \mathbf{A})_{ij}^L$$

model topology

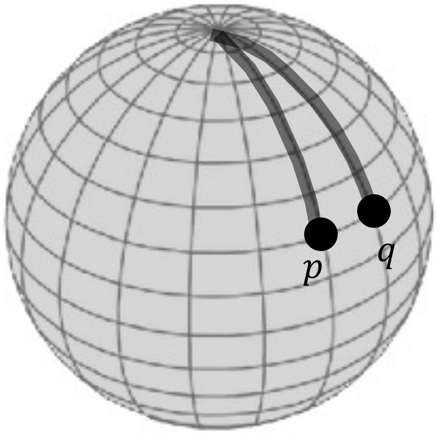
- **Width** p helps mitigate over-squashing (potentially at the risk of worse generalization)
- **Depth** L does not help
  - If $L \sim \mathrm{diam}(G)$, over-squashing occurs between distant nodes
  - If $L \gg 1$, we transition from over-squashing to vanishing gradients
- **Topology** of G has the largest effect on over-squashing



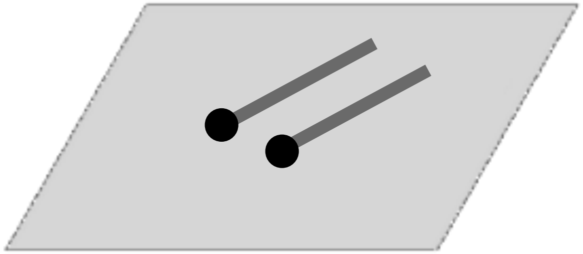


Di Giovanni et B 2023

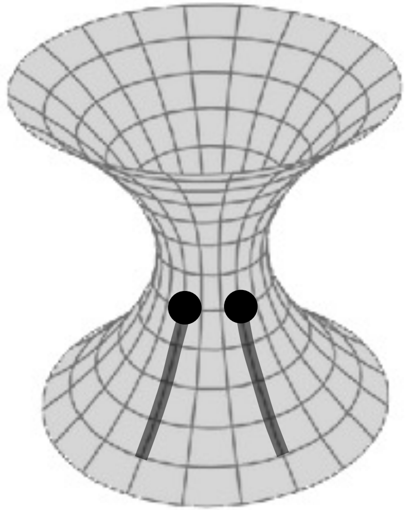
Ricci Curvature on Manifolds



Spherical (>0)



Euclidean (=0)

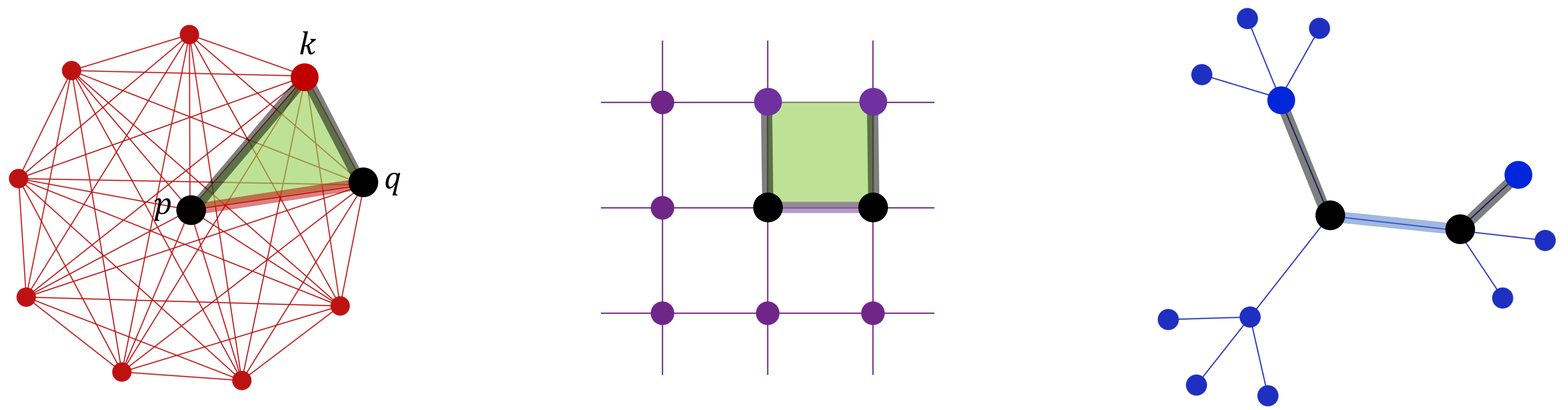


Hyperbolic (<0)

“geodesic dispersion”

Ricci 1903

Ricci Curvature on Graphs



Clique (>0)

Grid (=0)

Tree (<0)

Forman 2003; Ollivier 2007; Topping, di Giovanni, et B. 2021

Balanced Forman Curvature

Balanced Forman Curvature of edge $i \sim j$ in simple unweighted graph $\mathrm{Ric}(i,j) = 0$ if $\min\{d_i, d_j\} = 1$ and otherwise

$$\mathrm{Ric}(i,j) = \frac{2}{d_i} + \frac{2}{d_j} + 2\frac{|\sharp_\Delta(i,j)|}{\max\{d_i,d_j\}} + \frac{|\sharp_\Delta(i,j)|}{\min\{d_i,d_j\}} + \frac{\gamma_{\max}^{-1}}{\max\{d_i,d_j\}}\left(|\sharp_\square^i(i,j)| + |\sharp_\square^j(i,j)|\right) - 2$$

Triangles based at i~j

Max number of 4-cycle based at i~j traversing the same node

Degree of i

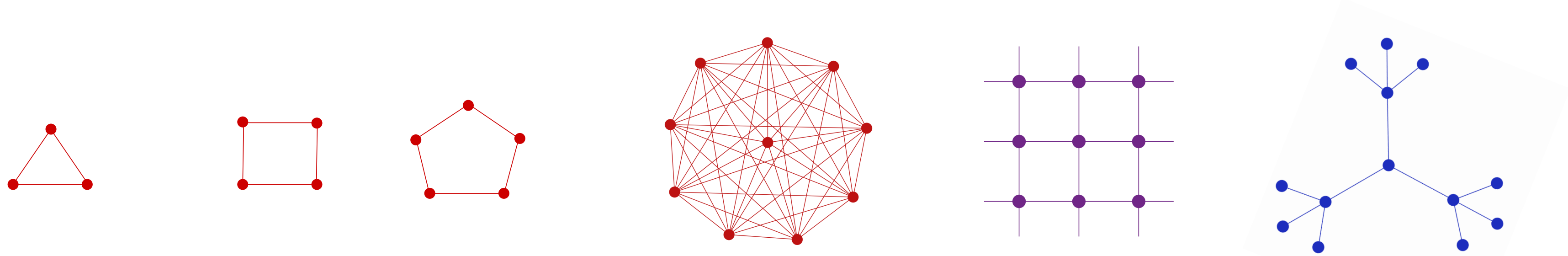
Neighbours of i forming a 4-cycle based at i~j (w/o diagonals)

Forman 2003; Topping, di Giovanni, et B. 2021

Balanced Forman Curvature

Balanced Forman Curvature of edge $i \sim j$ in simple unweighted graph $\mathrm{Ric}(i,j)$ =0 if $\min\{d_i, d_j\} = 1$ and otherwise

$$\mathrm{Ric}(i,j) = \frac{2}{d_i} + \frac{2}{d_j} + 2\frac{|\sharp_\Delta(i,j)|}{\max\{d_i,d_j\}} + \frac{|\sharp_\Delta(i,j)|}{\min\{d_i,d_j\}} + \frac{\gamma_{\max}^{-1}}{\max\{d_i,d_j\}}\left(\left|\sharp_\square^i(i,j)\right| + \left|\sharp_\square^j(i,j)\right|\right) - 2$$



Cycle C_3: $\frac{3}{2}$ C_4: 1 $C_{n\geq 5}$: 0 Clique K_n: $\frac{n}{n-1}$ Grid G_n: 0 Tree T_r: $\frac{4}{r+1} - 2$

Forman 2003; Topping, di Giovanni, et B. 2021

Over-squashing & Bottleneck via Curvature

Theorem 1 (main result): Consider an MPNN with $L \geq 2$ layers and $|\nabla\phi_\ell| \leq \alpha$ and $|\nabla\psi_\ell| \leq \beta$. Let $i \sim j$ with $d_i \leq d_j$ and assume $\exists\delta$ s.t. $0 < \delta < \max\{d_i, d_j\}^{1/2}$, $\delta < \gamma_{\max}^{-1}$ and $\mathrm{Ric}(i,j) \leq -2 + \delta$. Then, there exist nodes $Q \subset \{s\colon d_G(i,s) = 2\}$ of size $|Q| > 1/\delta$ s.t.

$$\frac{1}{|Q|}\sum_{k\in Q}\left|\frac{\partial x_k^{(\ell+2)}}{\partial x_i^{(\ell)}}\right| < (\alpha\beta)^2\delta^{1/4}$$

small δ = negative curvature

more nodes

stronger over-squashing

Over-squashing is caused by strongly negatively-curved edges!

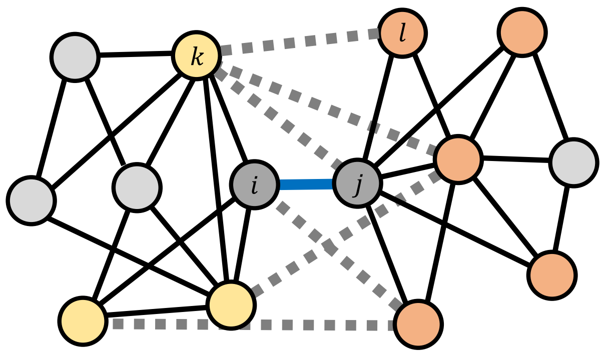
Topping, di Giovanni, et B. 2021

Stochastic Discrete Ricci Flow (SDRF)

Input: graph $G = (V, E)$, temperature $\tau > 0$, (optional C)

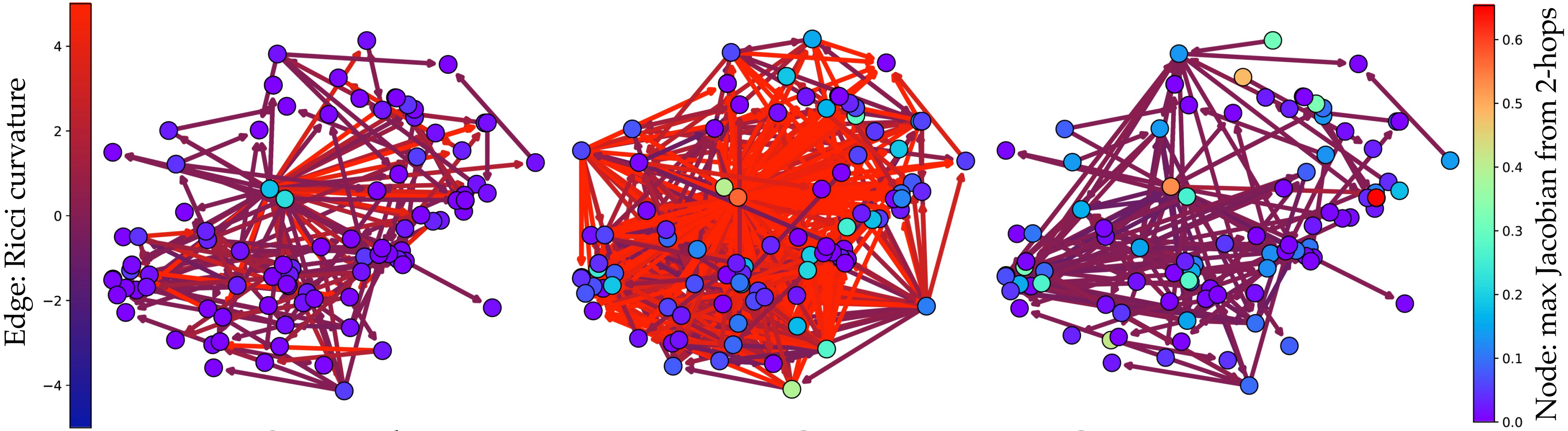
- For edge $i \sim j$ with smallest $\mathrm{Ric}(i, j)$
  - Calculate the improvement $\delta_{kl} = \mathrm{Ric}_{G'}(i, j) - \mathrm{Ric}(i, j)$ from adding edge $k \sim l$ with $k \in B_1(i)$ and $l \in B_1(j)$
  - Sample index k, l with probability $\mathrm{Softmax}(\tau \delta_{kl})$ and add edge $k \sim l$ to E'
- (optional) Remove edge $i \sim j$ with largest $\mathrm{Ric}(i, j) > C$

Output: new graph $G' = (V, E')$



Topping, di Giovanni, et B. 2021

Curvature- vs Diffusion-based Rewiring



Original
Cornell graph

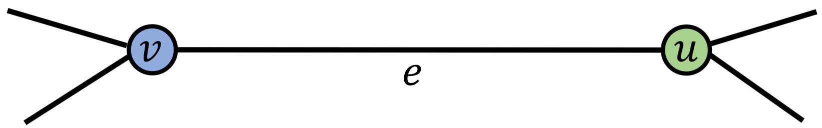
DIGL
+308% edges

Curvature rewiring
+36% / -36% edges

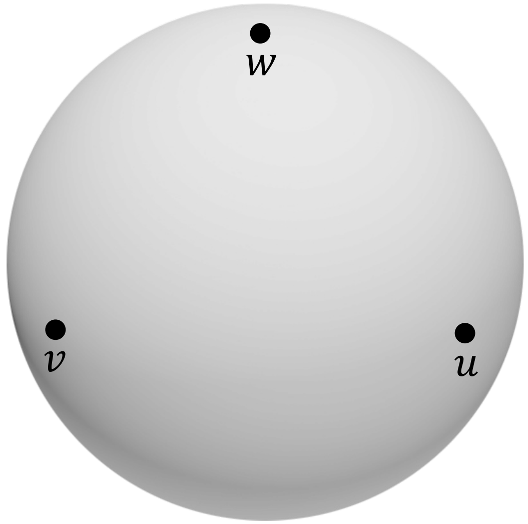
Topping, di Giovanni, et B. 2021; Klicpera et al. 2019 (DIGL)


Even more Exotic Stuff

Cellular Sheaves



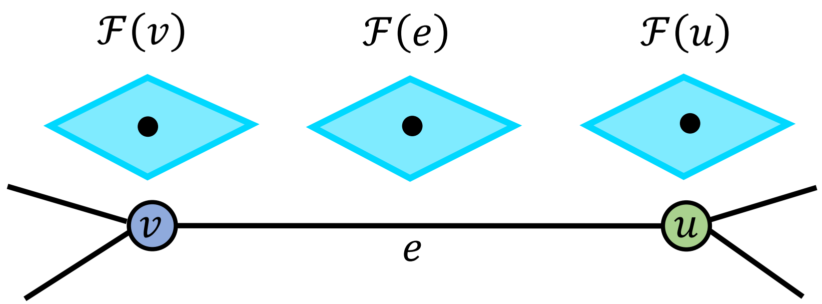
Graph



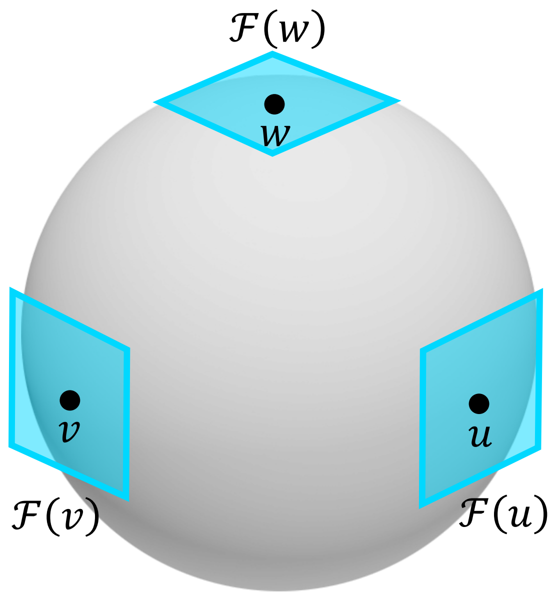
Manifold

Bodnar, di Giovanni, et B. 2022

Cellular Sheaves



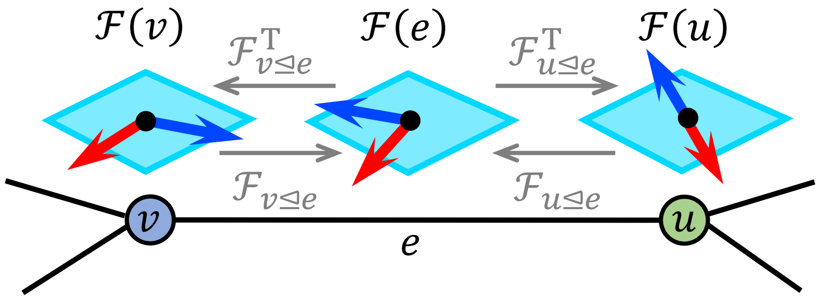
Cellular sheaf $\mathcal{F}$



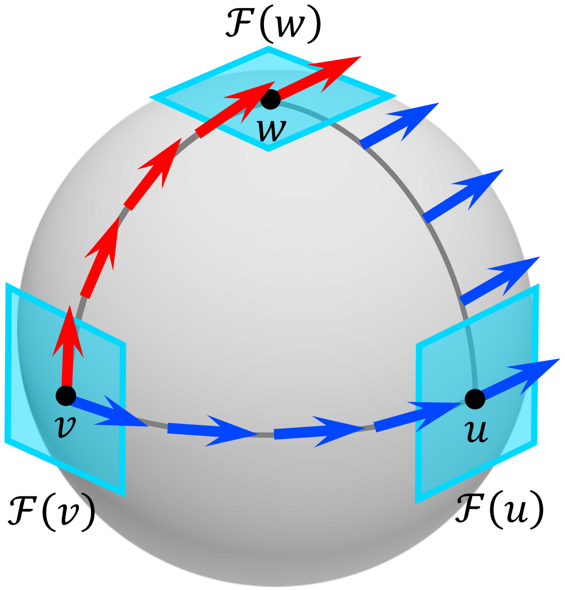
Manifold + Connection

Bodnar, di Giovanni, et B. 2022

Cellular Sheaves



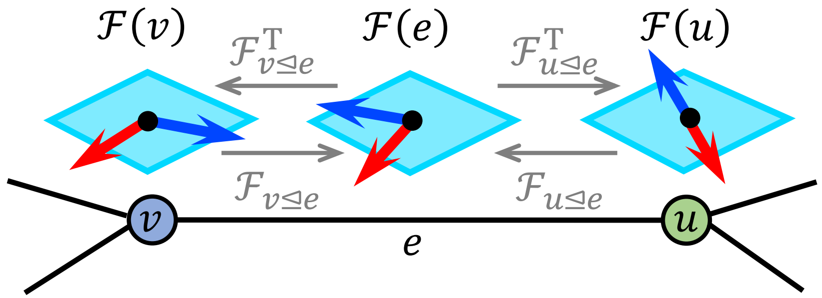
Cellular sheaf $\mathcal{F}$



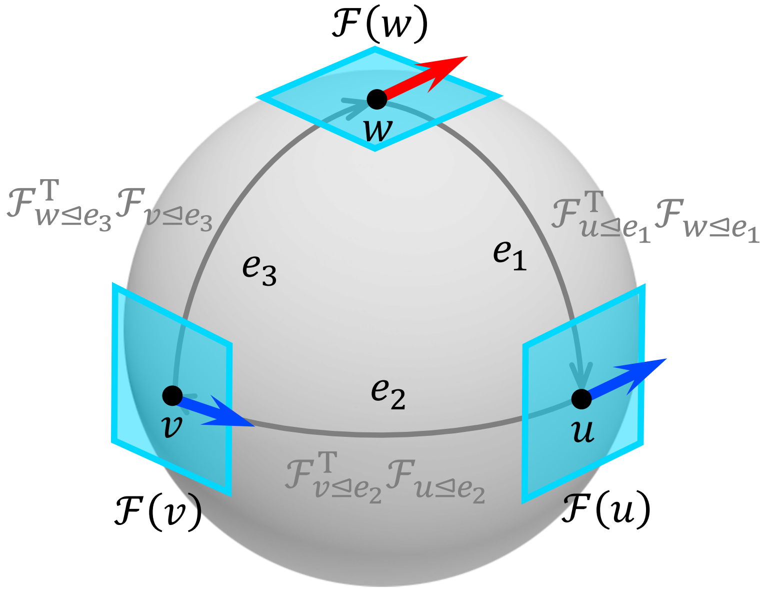
Manifold + Connection

Bodnar, di Giovanni, et B. 2022

Cellular Sheaves



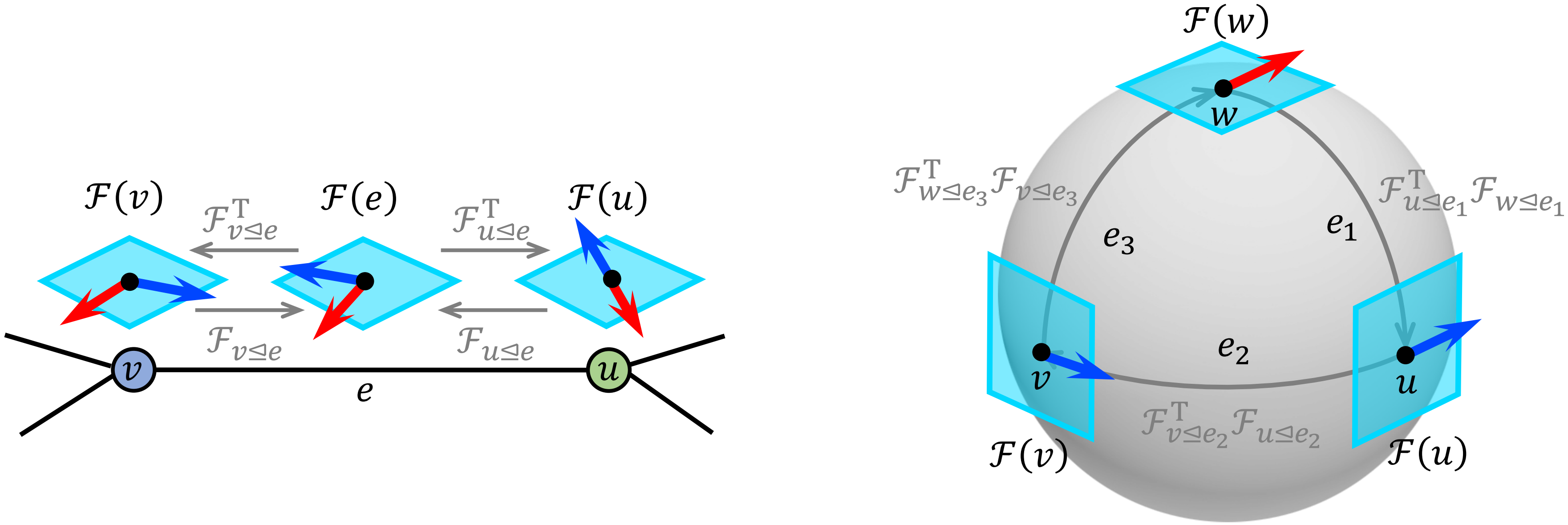
Cellular sheaf $\mathcal{F}$



Analogy to parallel transport
on manifolds

Bodnar, di Giovanni, et B. 2022

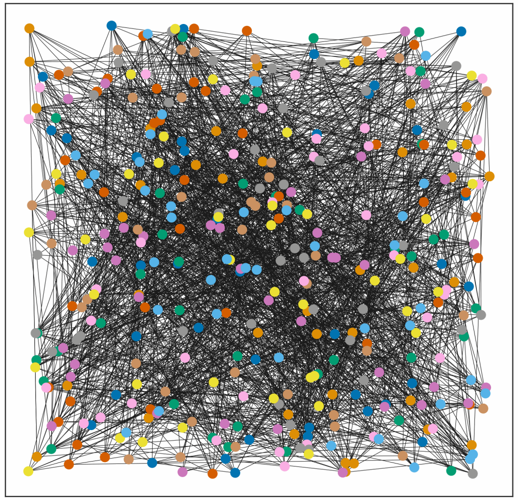
Cellular Sheaves



Endow graph with "geometry" leading to richer diffusion with better separation, ability to cope with heterophily, and no oversmoothing

Bodnar, di Giovanni, et B. 2022

Diffusion on Cellular Sheaves



$$\dot{\mathbf{X}}(t) = \Delta_{\mathcal{F}}\mathbf{X}(t) \quad \text{with i.c. } \mathbf{X}(0) = \mathbf{X}$$

Node classification = limit of sheaf diffusion equation
with an appropriate sheaf

Bodnar, di Giovanni, et B. 2022

Alternative to Weisfeiler-Lehman for expressive power?

Graph type	**# Node classes**	**Sheaf class $\mathcal{F}$**, dim=d	
Homophilic	2	Symmetric d=1	✓
Heterophilic	2	Symmetric d=1	✗
	2	Non-symmetric d=1	✓
	≥3	Non-symmetric d=1	✗
	≤2d	Orthogonal, d={2,4}	✓

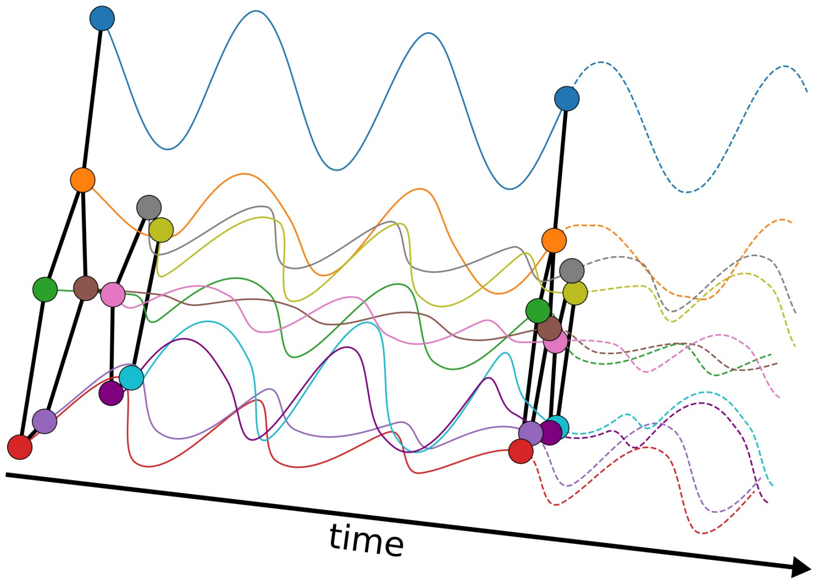
The capability of sheaf diffusion to solve node classification problem in the limit

Bodnar, di Giovanni, et B. 2022

What do we gain from physics-inspired GNNs?

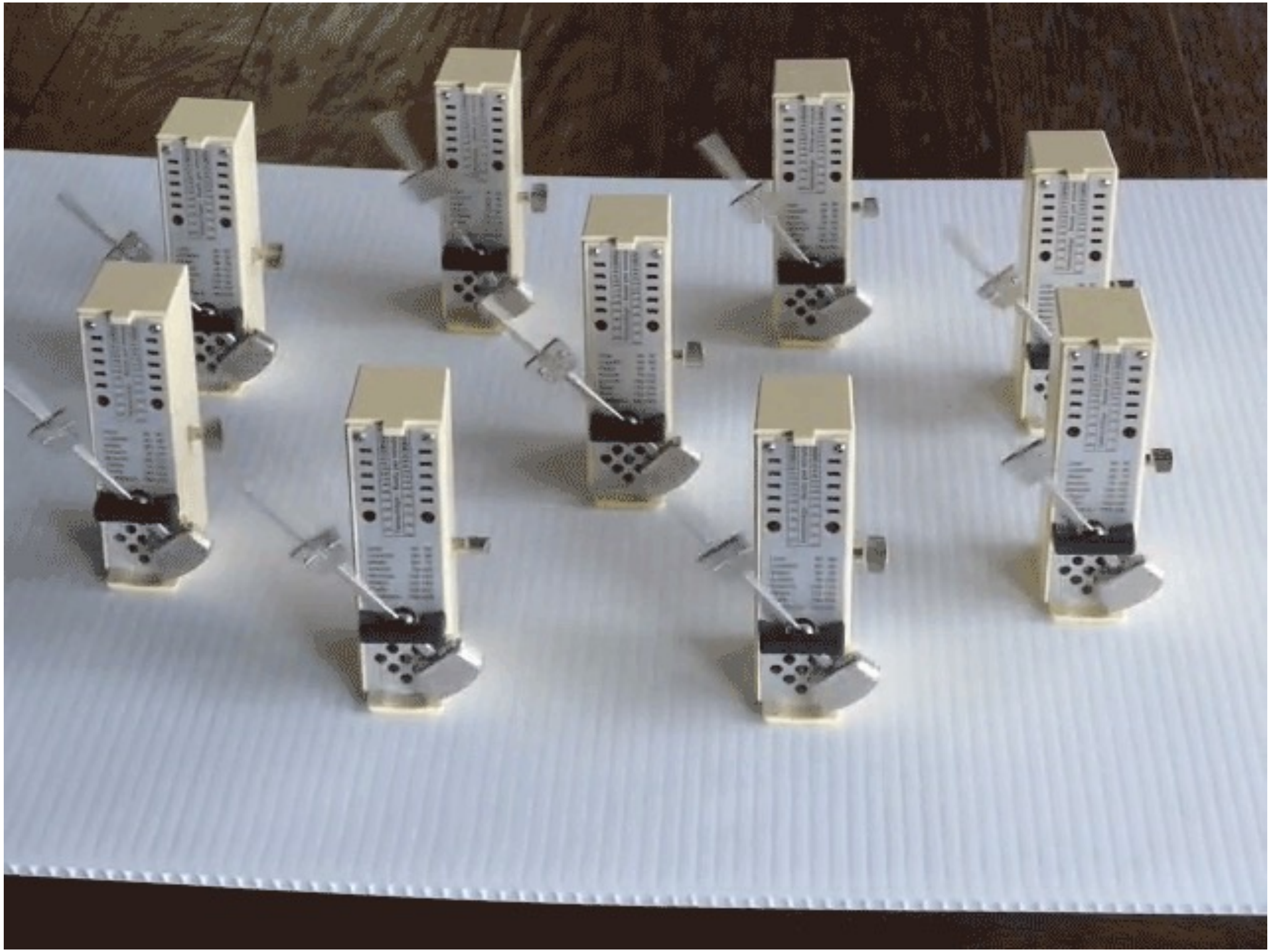
- New perspectives on old problems (e.g. oversmoothing, bottlenecks, etc.)
- New architectures
  - Many GNNs can be formalised as a discretised Graph Diffusion equation
  - More efficient solvers (multistep, adaptive, implicit, multigrid, etc.)
  - Implicit schemes = multi-hop filters
- Principled architectural choices (residual connection, shared symmetric weights)
- Theoretical guarantees (e.g. stability, convergence, expressive power, etc.)
- Deep links to other fields less known in GNN literature (e.g. differential geometry and algebraic topology)
- Other physical models

Graph-Coupled Oscillators



Dynamics of a system of coupled oscillators on a molecular graph

Rusch et B. 2022




Thank you!