

Structural Priors for Graph Neural Network Design

Fragkiskos D. Malliaros

CentraleSupélec, Inria, Université Paris-Saclay

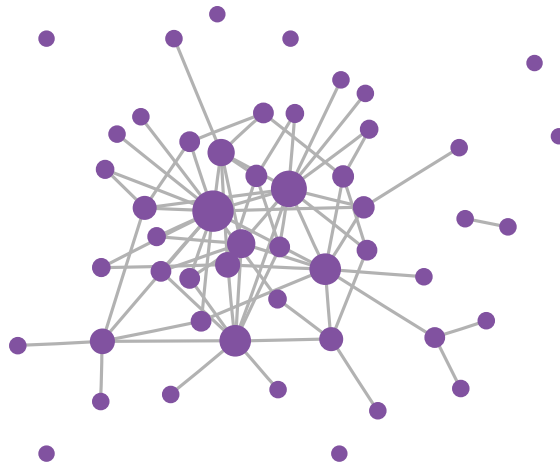
ILLS Bellairs Workshop on Machine Learning and Statistical Signal Processing

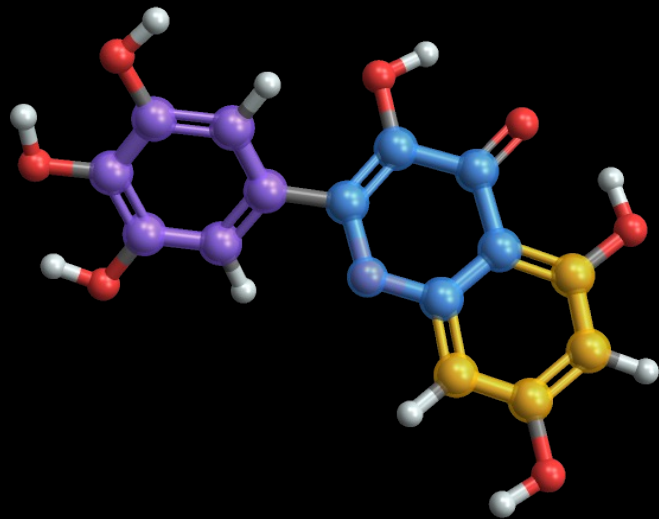
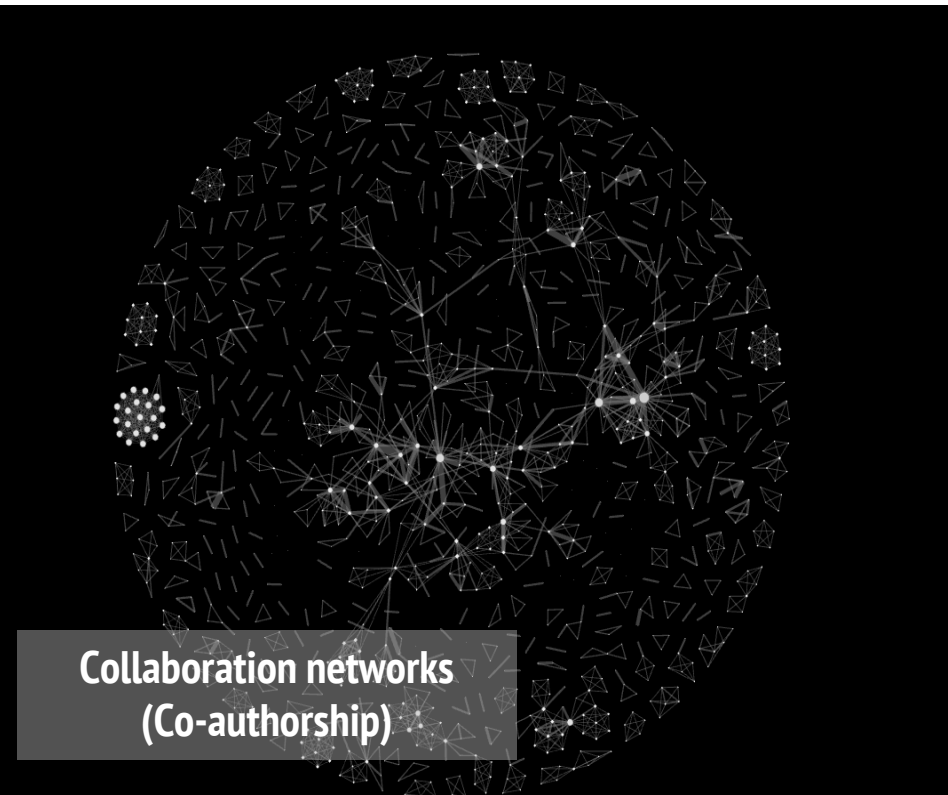
December 16, 2025

interactions + complexity

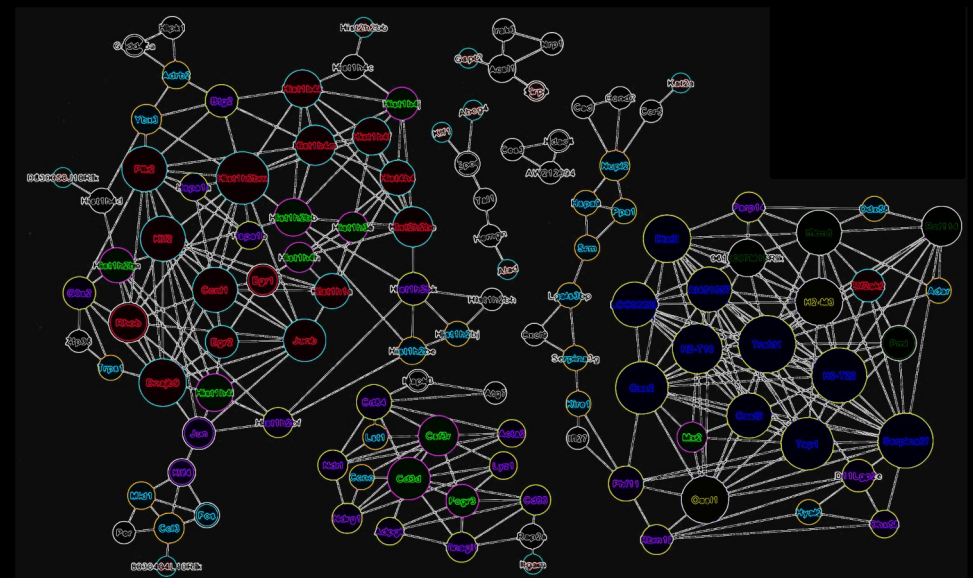
=

complex network





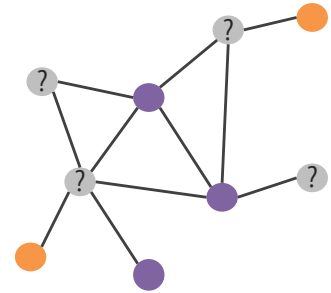
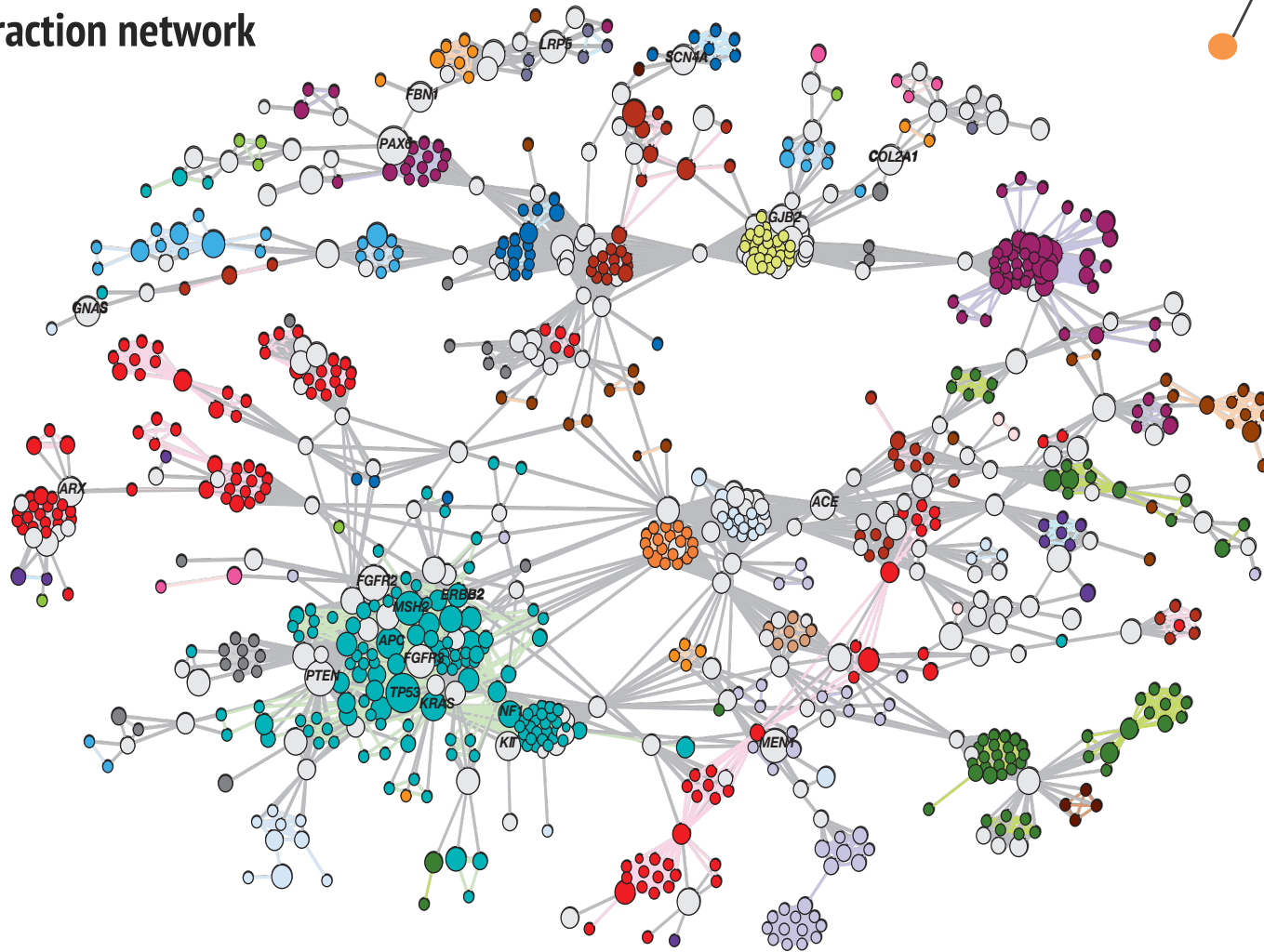
Molecular graphs



Gene co-expression network

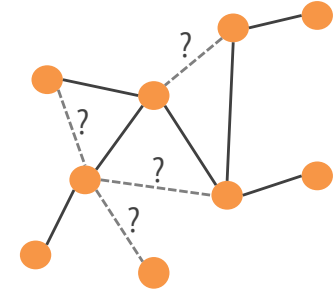
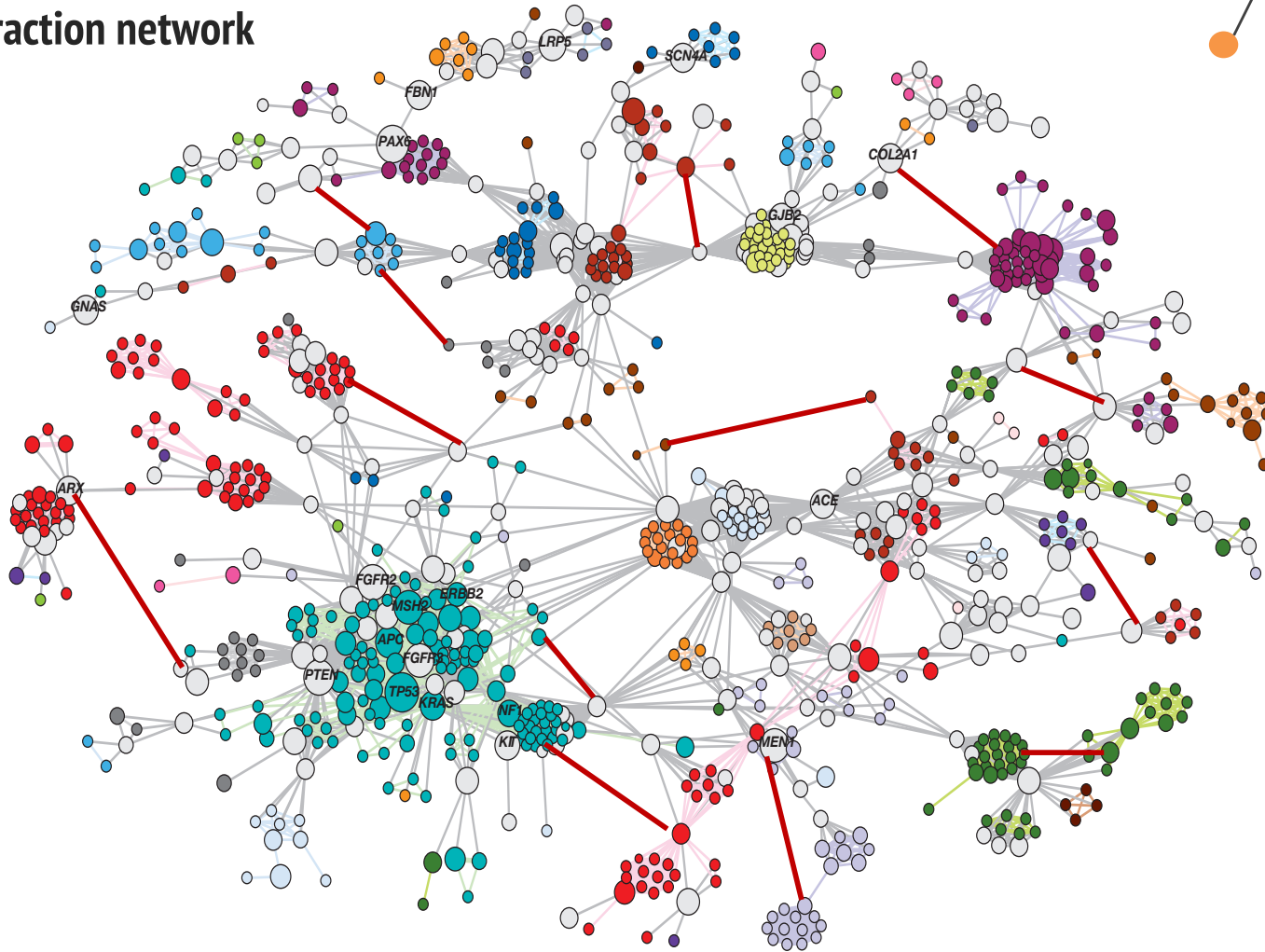
Node Classification

Gene interaction network



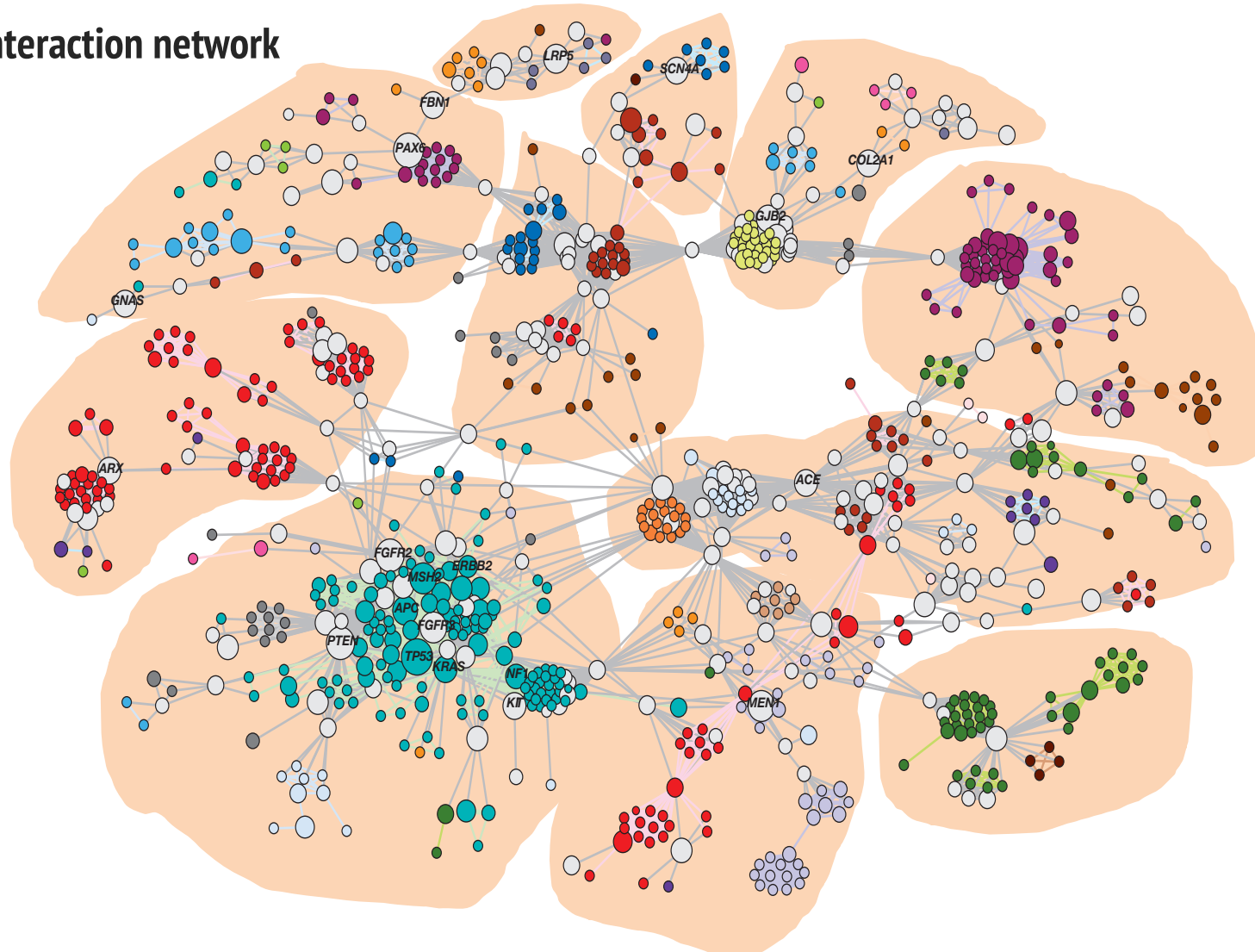
Link Prediction

Gene interaction network



Community Detection (or Graph Clustering)

Gene interaction network



Outline of the Presentation

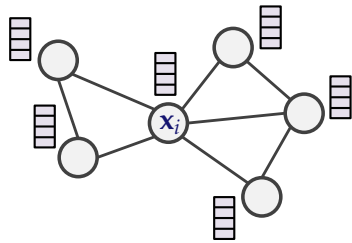
Part I. Brief introduction to Graph Neural Networks (GNN)

Part II. Topics in GNN model design: over-squashing, pooling, generalization

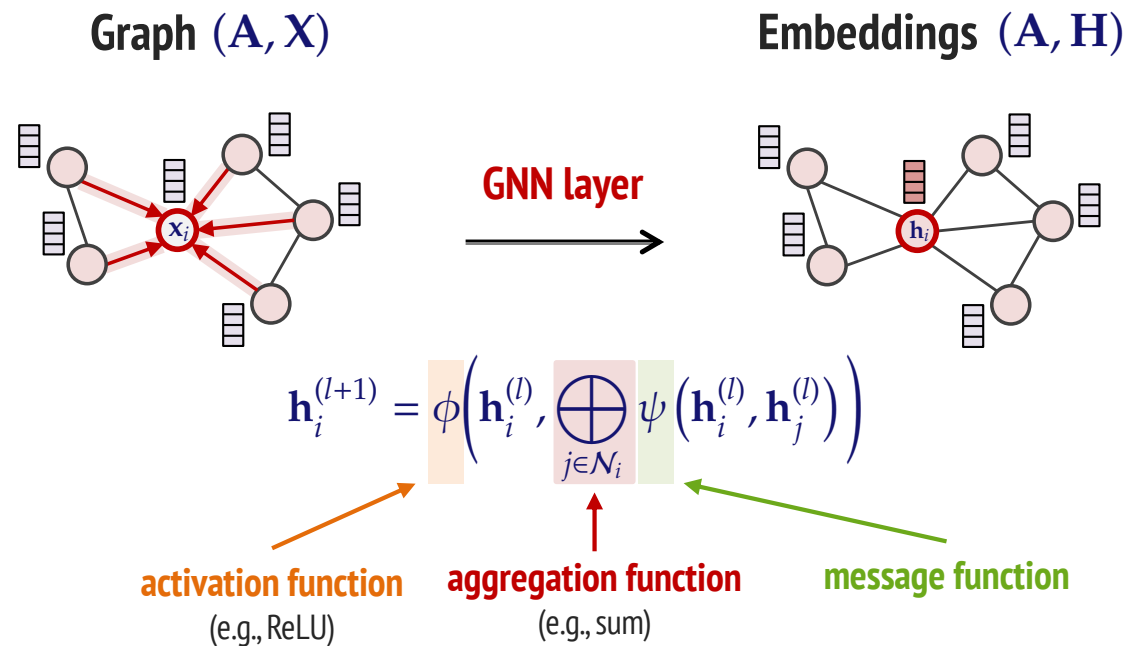
Part III. Perspectives and ongoing work

Graph Neural Networks (GNNs)

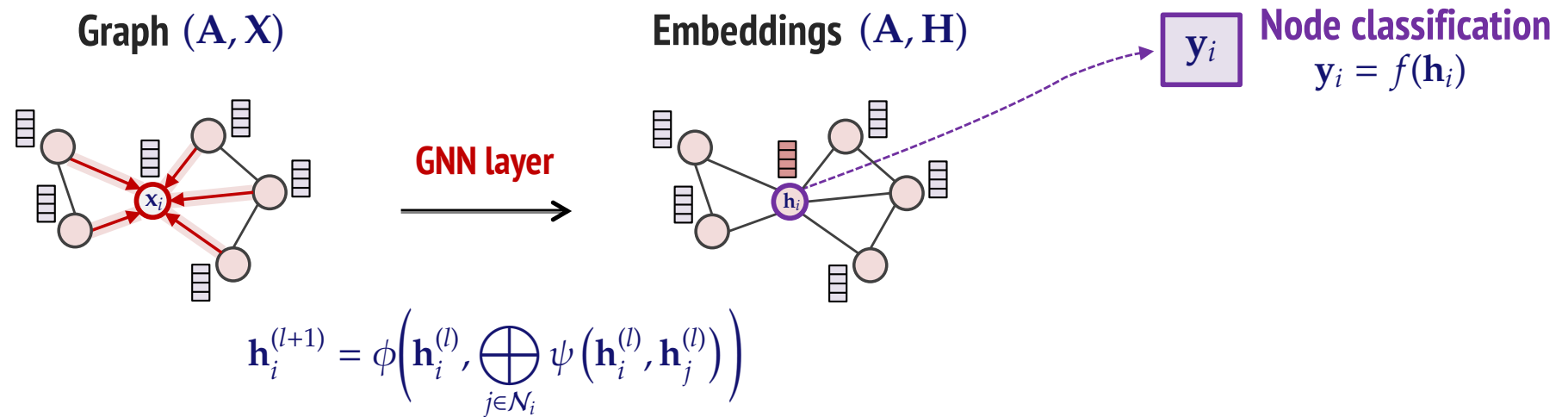
Graph (A, X)



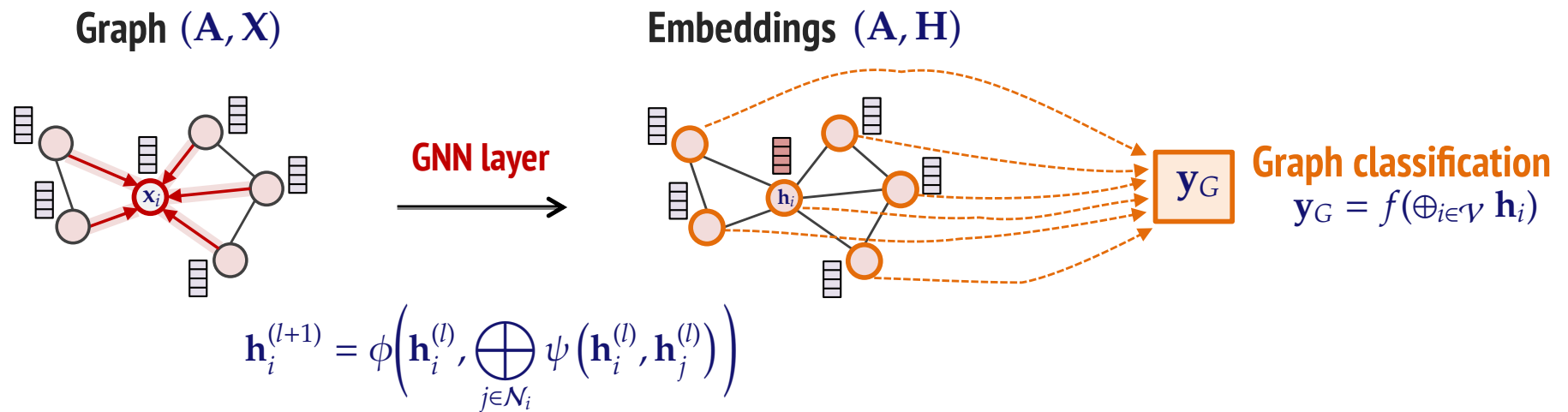
Graph Neural Networks (GNNs)



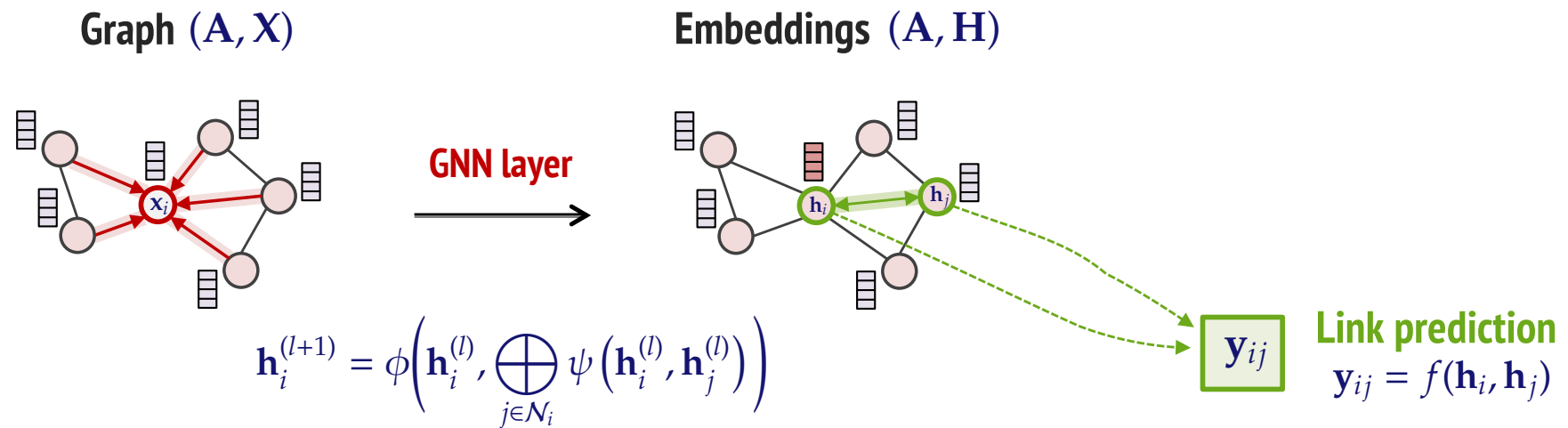
Graph Neural Networks (GNNs)



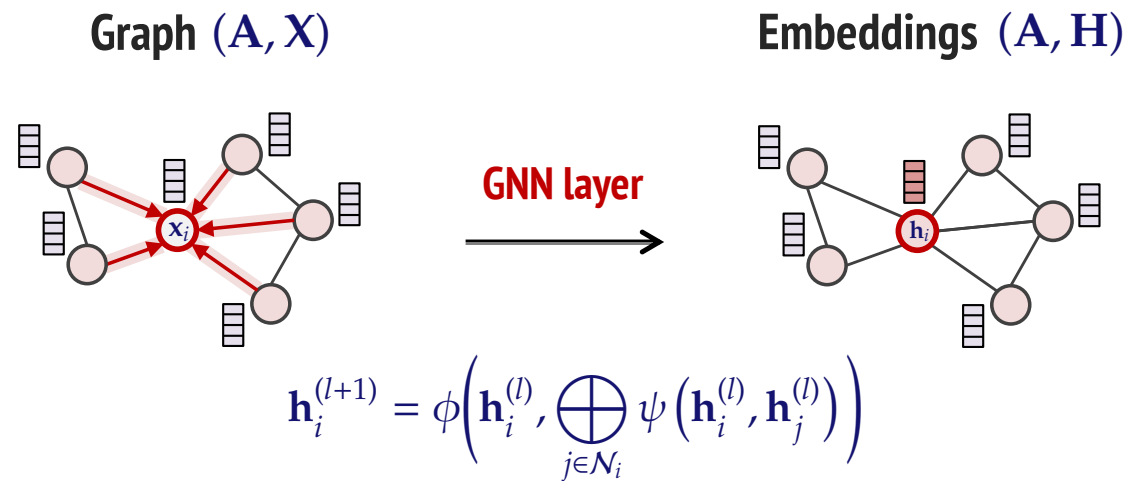
Graph Neural Networks (GNNs)



Graph Neural Networks (GNNs)



Graph Neural Networks (GNNs)



$\mathbf{y}_i$ **Node classification**
 $\mathbf{y}_i = f(\mathbf{h}_i)$

$\mathbf{y}_G$ **Graph classification**
 $\mathbf{y}_G = f(\bigoplus_{i \in \mathcal{V}} \mathbf{h}_i)$

$\mathbf{y}_{ij}$ **Link prediction**
 $\mathbf{y}_{ij} = f(\mathbf{h}_i, \mathbf{h}_j)$

Different instances of GNN layers

Convolutional GNNs (e.g., ChebNet, GCN, SGC)

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in \mathcal{N}_i} c_{ij} \psi(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)})\right)$$

$\uparrow$
node degree

GNNs with Attention (e.g., GAT)

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in \mathcal{N}_i} \alpha(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}) \psi(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)})\right)$$

$\uparrow$
attention mechanism

Challenges in GNN Model Design

1. How to design **deep** GNNs?
 - Graph rewiring to address **over-smoothing** and **over-squashing** (**SJLR**, **TRIGON**)
2. How to compute **graph-level** representations?
 - Hierarchical clustering-based **graph pooling** (**HoscPool**)
3. (if time permits) How to improve **generalization** of GNNs?
 - Framework for graph data **augmentation** (**GRATIN**)

Leverage structural graph priors: connectivity, geometry, and local patterns

Over-smoothing and Over-squashing in GNNs

w/ J.H. Giraldo, H. Attali, K. Skianis, T. Bouwmans, T. Papastergiou, N. Pernelle

CIKM '23, CIKM '25

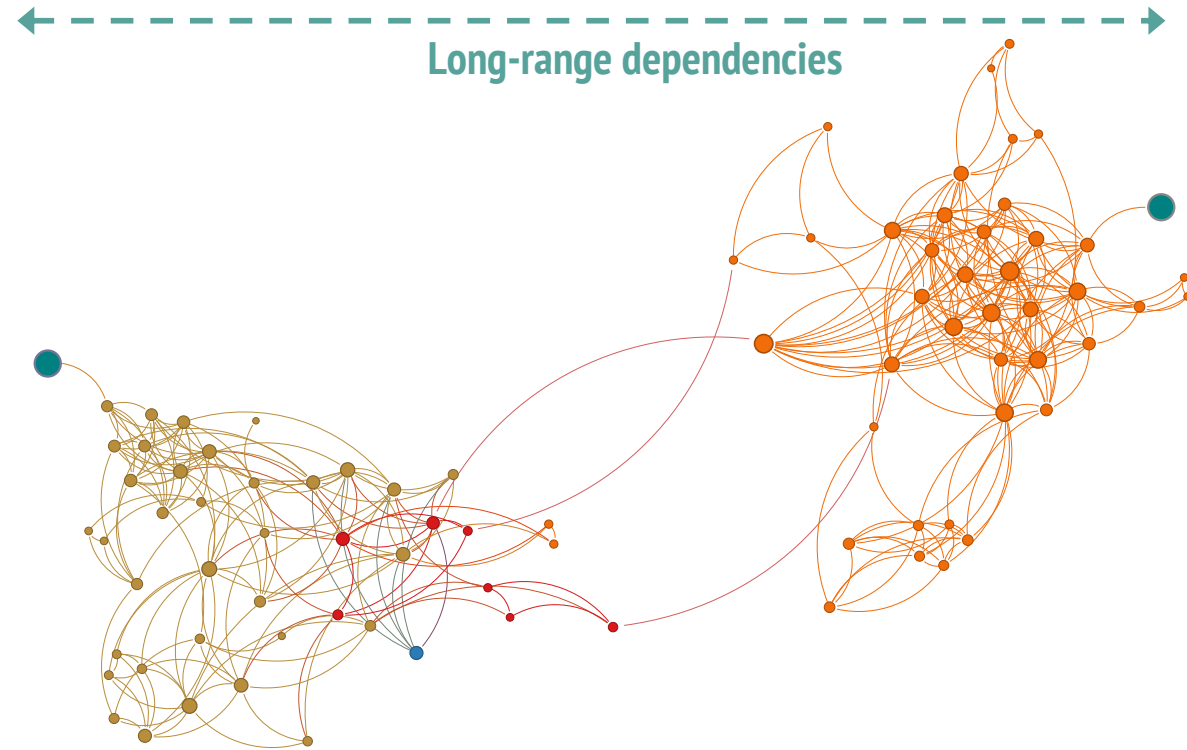


J.H. Giraldo
Télécom Paris

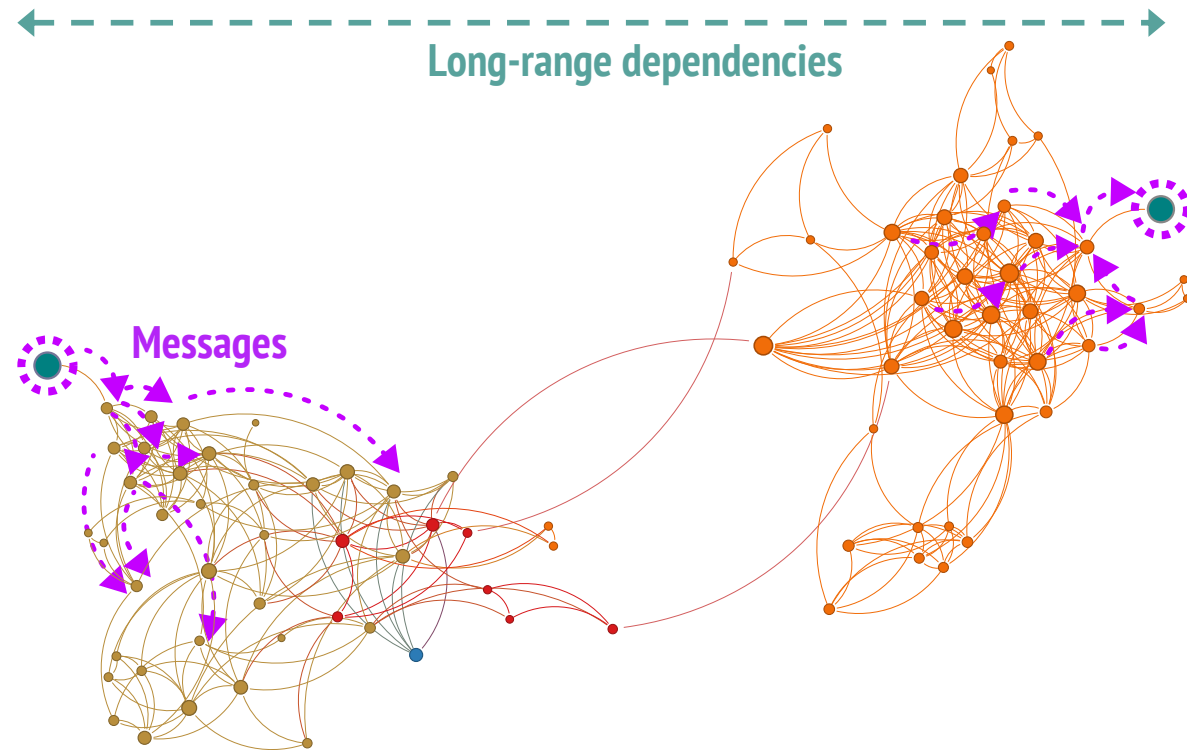


H. Attali
Univ. Sorbonne Paris Nord

Long-range Dependencies and Deep GNNs

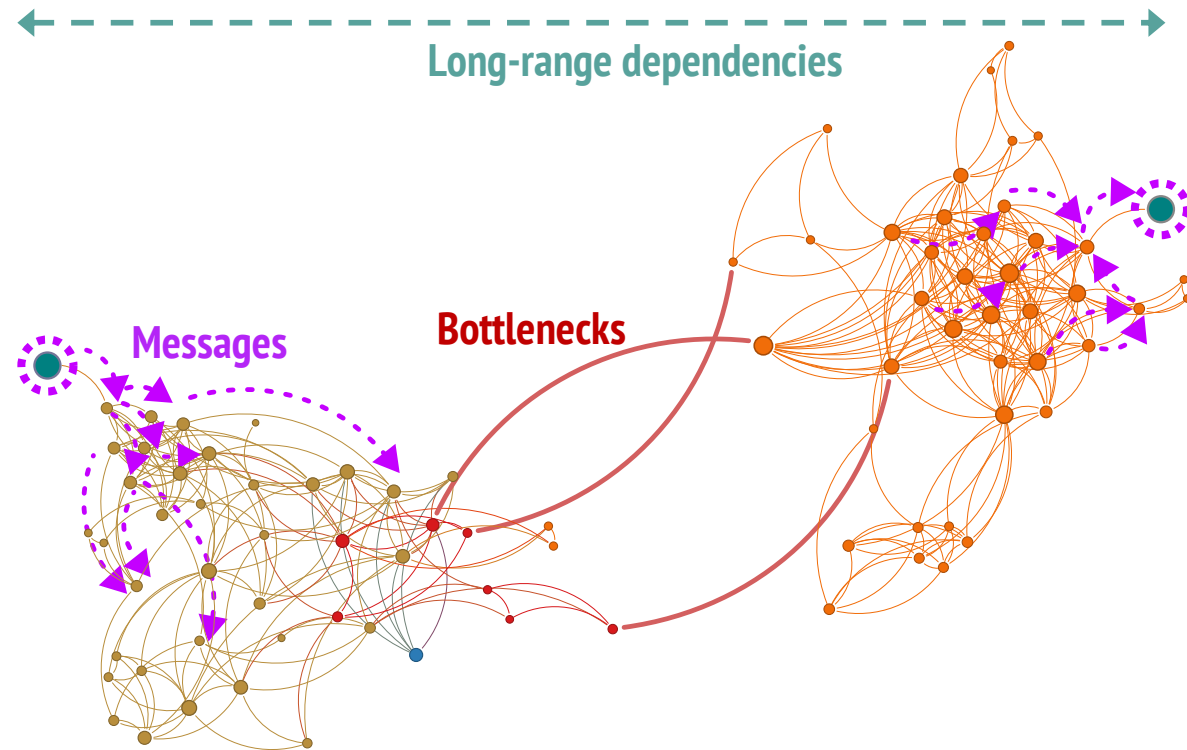


Long-range Dependencies and Deep GNNs



- **Over-smoothing:** node embeddings become **indistinguishable** with more GNN layers

Long-range Dependencies and Deep GNNs



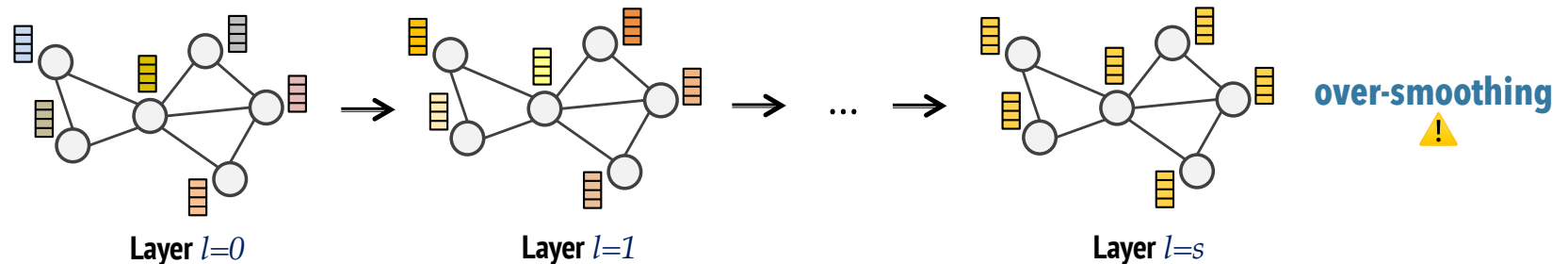
- **Over-smoothing:** node embeddings become **indistinguishable** with more GNN layers
- **Over-squashing:** information from distant nodes is **squeezed** on **bottleneck** edges

Overview of Key Findings

- We establish a fundamental **topological relationship** between over-smoothing and over-squashing in deep GNNs
- We found that the **spectral gap** of a graph is intrinsically related to both problems
- There is an inherent **trade-off** between over-smoothing and over-squashing
- We introduce a **curvature-** and **triangle-based** algorithms to mitigate this trade-off

The **Over-smoothing** – Over-squashing Trade-off

The stationary distribution on graphs



- Consider a simple GNN model without nonlinearities (e.g., SGC)
 - Repeated message passing is equivalent to applying a **random walk operator**
- For a random walk transition matrix $\mathbf{P}$ and initial distribution $\mathbf{f} : \mathcal{V} \rightarrow \mathbb{R}$, we can compute s such that

$$\|\mathbf{f}^\top \mathbf{P}^s - \boldsymbol{\pi}\| \leq e^{-s\lambda_2} \frac{\max_i \sqrt{d_i}}{\min_j \sqrt{d_j}}$$

s : number of GNN layers
 λ_2 : spectral gap of $\mathbf{L}$

- GNNs converge **exponentially fast** to the **stationary distribution** $\boldsymbol{\pi}$ when stacking several layers → **over-smoothing**
- The convergence depends on the **spectral gap** λ_2

The Over-smoothing – Over-squashing Trade-off

Cheeger constant and bottlenecks

- The Cheeger constant h_G of a graph

$$h_G = \min_{S \subset \mathcal{V}, 0 < |S| \leq \frac{|\mathcal{V}|}{2}} \frac{|\partial S|}{\min(\text{vol}(S), \text{vol}(\mathcal{V} \setminus S))}$$

edge boundary
(# of edges crossing the cut)

sum of node
degrees in S

- Captures **structural bottlenecks** in the graph

- Cheeger constant and spectral gap: $2h_G \geq \lambda_2 \geq \frac{h_G^2}{2}$

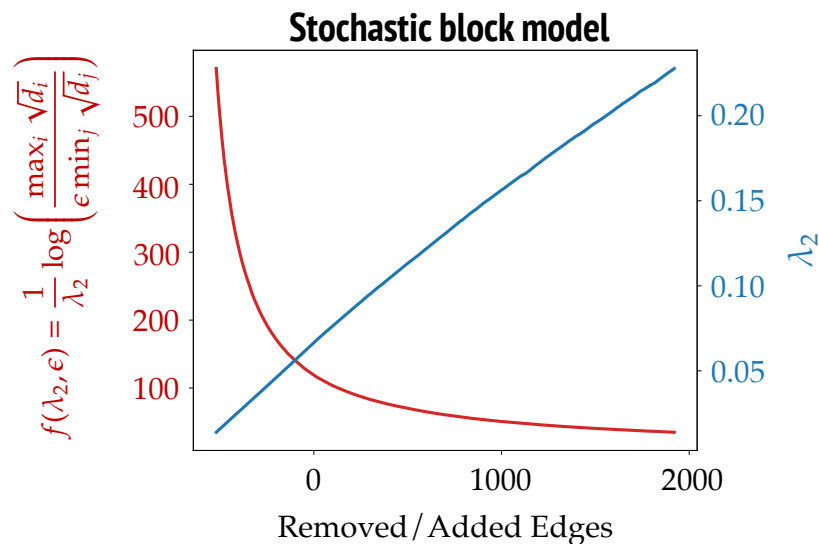
- **Small Cheeger constant** h_G and λ_2 imply bottlenecks $\longrightarrow$ **over-squashing**

The Over-smoothing – Over-squashing Trade-off

The trade-off

$$h_G \geq \frac{1}{2s} \log \left(\frac{\max_i \sqrt{d_i}}{\epsilon \min_j \sqrt{d_j}} \right)$$

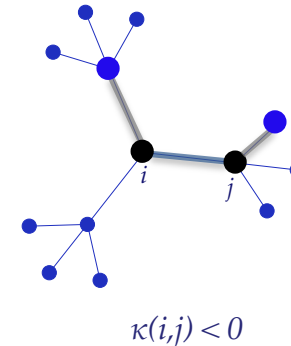
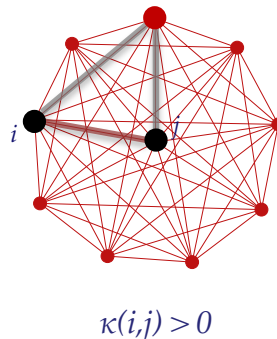
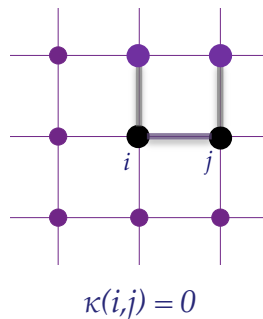
- If $s \rightarrow 0$ then $h_G \rightarrow \infty$: reduce bottlenecks by accelerating convergence to the stationary distribution. **Over-smoothing**.
- If $h_G \rightarrow 0$ then $s \rightarrow \infty$: avoid converging to the stationary distribution by promoting a bottleneck-like structure. **Over-squashing**.



- We can increase mixing time by **removing** some edges
 - Alleviate over-smoothing
- We increase λ_2 by **adding** edges, improving h_G
 - Alleviate over-squashing

The SJLR Algorithm: Key Ingredients

- We target to manipulate the **spectral gap** λ_2 via **graph rewiring**
- We borrow ideas from **graph curvature** $\kappa(i,j)$
 - **Increasing curvature** improves the spectral gap



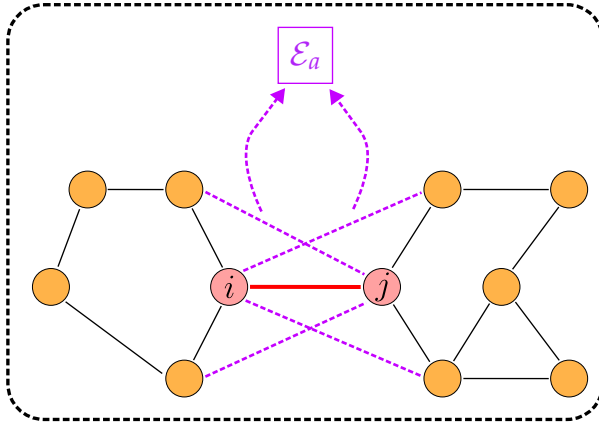
[Bronstein, Physics-inspired GNNs '23]

- **SJLR**: Stochastic Jost and Liu Curvature (JLC) Rewiring
 - JLC: curvature metric based on **triangles**
 - **Greedy algorithm**: adds/removes edges during training to locally improve curvature
 - Graph structure + node features
 - Good performance in graph with both **homophily** and **heterophily**

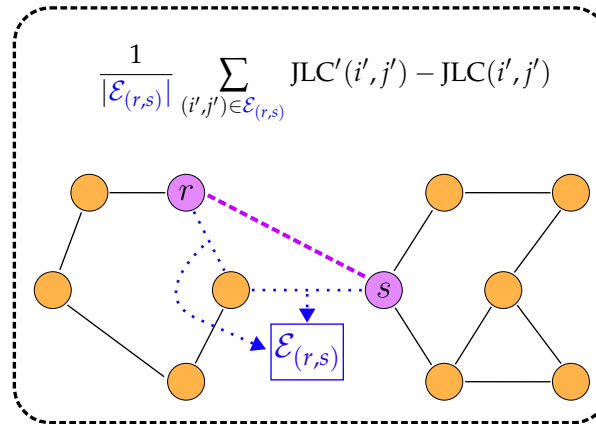
SJLR: The Algorithm

Stochastic Jost and Liu Curvature Rewiring

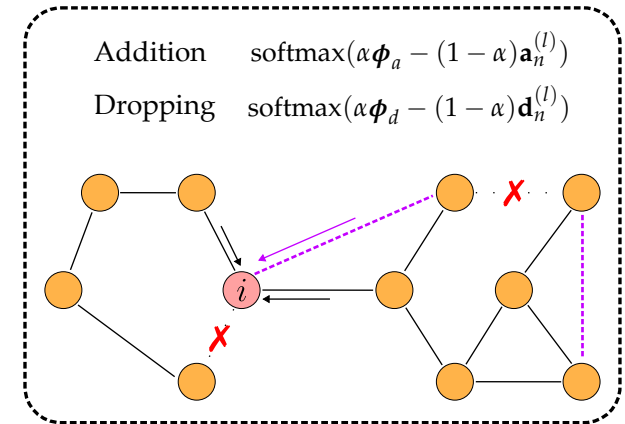
(1) Good edges to add



(2) Score of improvement of adding (r, s)



(3) GNN training



- (1) Compute a bank of **candidate edges** to add $\mathcal{E}_a$
 - Calculate and sort edges (i, j) based on the Jost and Liu Curvature (JLC)
- (2) Associate a score to every edge $(r, s) \in \mathcal{E}_a$
 - Average **improvement of curvature** of adding (r, s) to the graph
- (3) **Graph rewiring** during training
 - **Add** and **drop** edges stochastically based on the JLC metric + node feature similarity

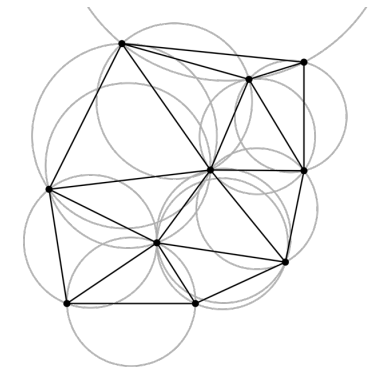
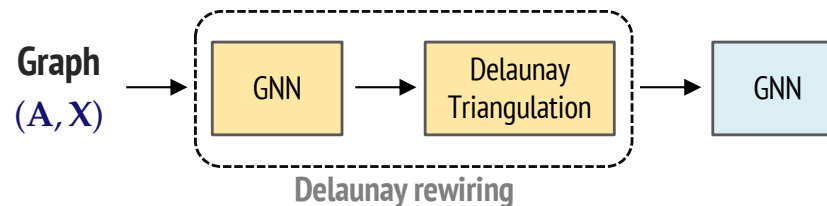
SJLR: Experimental Results

Method	Heterophilous						Homophilous			Overall
	Cornell	Texas	Wisconsin	Chameleon	Squirrel	Actor	Cora	Citeseer	Pubmed	
Baseline	<u>67.34</u> ± 1.50	58.05 ± 0.96	52.10 ± 0.95	40.35 ± 0.48	42.12 ± 0.29	28.62 ± 0.36	81.81 ± 0.26	68.35 ± 0.35	78.25 ± 0.37	<u>57.44</u>
RDC [32]	63.78 ± 1.68	59.47 ± 1.00	50.89 ± 1.00	40.33 ± 0.51	<u>41.98</u> ± 0.31	28.97 ± 0.33	81.54 ± 0.26	68.70 ± 0.35	78.42 ± 0.39	57.12
GDC [19]	64.18 ± 1.36	56.43 ± 1.15	49.61 ± 0.95	38.49 ± 0.51	33.20 ± 0.29	31.08 ± 0.27	82.63 ± 0.23	69.15 ± 0.30	79.04 ± 0.37	55.98
DE [42]	63.39 ± 1.29	57.41 ± 0.93	47.84 ± 0.86	<u>40.80</u> ± 0.55	41.68 ± 0.39	<u>29.99</u> ± 0.21	81.90 ± 0.24	68.99 ± 0.36	78.53 ± 0.26	56.73
PN [56]	64.44 ± 1.39	60.93 ± 1.15	51.78 ± 0.95	40.37 ± 0.59	40.92 ± 0.31	28.21 ± 0.21	78.89 ± 0.32	66.95 ± 0.40	76.60 ± 0.41	56.57
DGN [57]	65.19 ± 1.79	58.91 ± 0.93	50.76 ± 0.92	40.06 ± 0.60	41.30 ± 0.32	28.32 ± 0.36	81.34 ± 0.31	69.25 ± 0.35	78.06 ± 0.42	57.02
FA [1]	53.57 ± 0.00	59.26 ± 0.00	43.02 ± 0.49	27.76 ± 0.29	31.51 ± 0.00	26.69 ± 0.50	29.85 ± 0.00	23.23 ± 0.00	39.24 ± 0.00	37.13
SDRF [48]	63.88 ± 1.68	56.40 ± 0.89	40.99 ± 0.62	40.74 ± 0.45	41.44 ± 0.37	28.95 ± 0.33	81.42 ± 0.26	<u>69.37</u> ± 0.31	77.74 ± 0.42	55.66
FoSR [27]	56.65 ± 0.93	50.01 ± 1.37	<u>53.73</u> ± 1.08	40.26 ± 0.50	41.83 ± 0.28	28.80 ± 0.35	81.79 ± 0.26	67.99 ± 0.37	78.26 ± 0.39	55.48
SJLR (ours)	71.75 ± 1.50	<u>60.13</u> ± 0.89	55.16 ± 0.95	41.19 ± 0.46	41.86 ± 0.29	29.89 ± 0.20	<u>81.95</u> ± 0.25	69.50 ± 0.33	<u>78.60</u> ± 0.33	58.89

Classification results for the GCN model

Rewiring and the Role of Triangles

- Increasing the number of **triangles** can improve **over-squashing**
 - Improves curvature
- Delaunay **triangulation-based** rewiring [Attali et al., ICML '24]

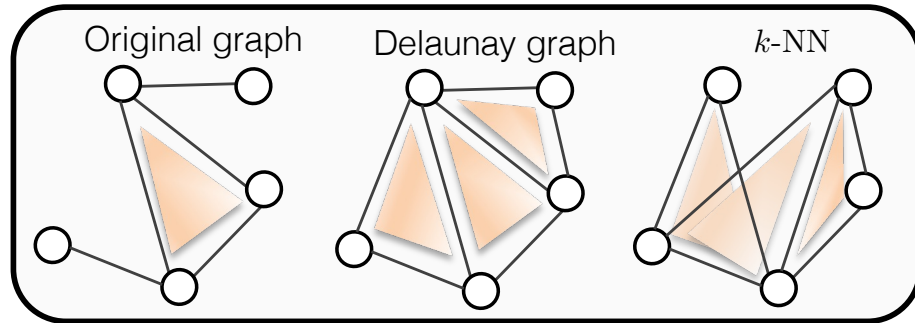


Source: Wikipedia

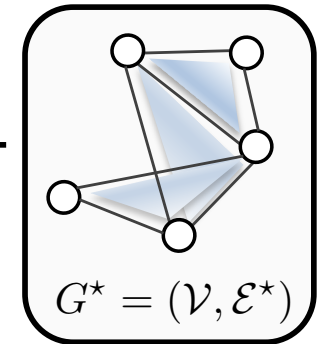
- Planar triangulation:** diameter grows proportionally to $\sqrt{n}$; also, small spectral gap
- Local triangles:** connects spatially proximal points
- Task-agnostic:** geometry-only, ignoring downstream loss

TRIGON: Triangle-based Graph Rewiring

① Input graphs

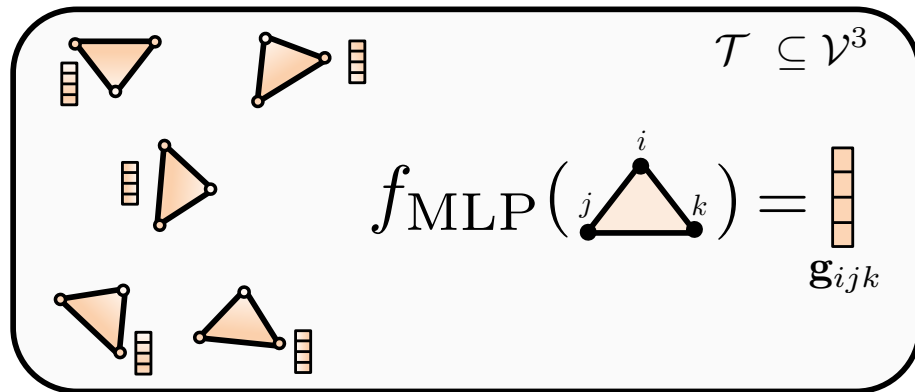


④ Rewired graph

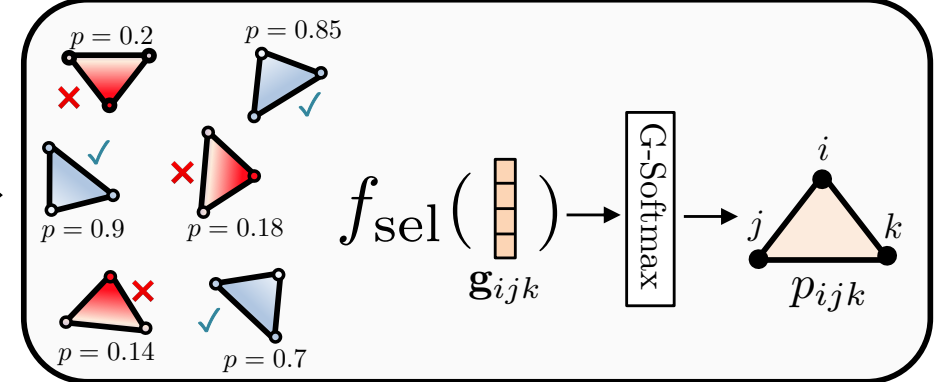


Triangle encoding

② Triangle encoding



③ Triangle selection



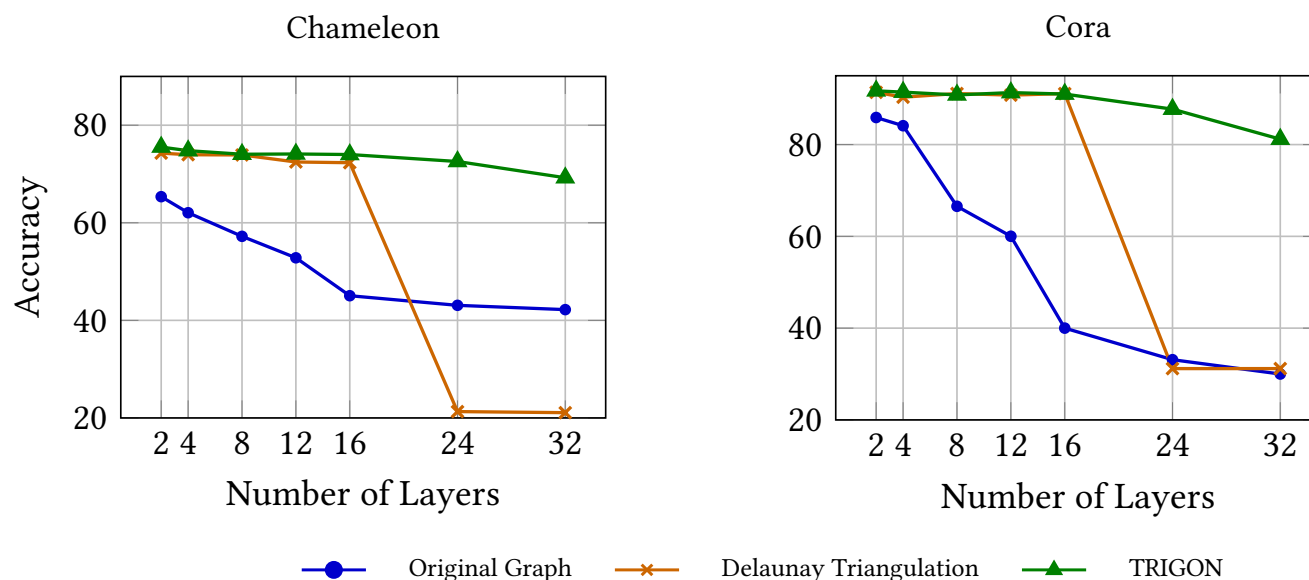
$\mathcal{L}_{\text{selector}}$
 $\mathcal{L}_{\text{GNN}}$

GNN

TRIGON: Experimental Results

	Base (GCN)	DIGL	FA	SRDF	FOSR	BORF	GTR	JDR	DR	TRIGON
Cham.	65.35±0.54	54.82±0.48	26.34±0.61	63.08±0.37	67.98±0.40	65.35±0.51	68.03±0.61	65.85±0.49	<u>74.28±0.48</u>	75.52±0.50
Squir.	51.30±0.38	40.53±0.29	22.88±0.42	49.11±0.28	52.63±0.30	≥24h	53.32±0.44	53.78 ±0.46	<u>65.25±0.26</u>	66.48±0.35
Actor	30.02±0.22	26.75±0.23	26.03±0.30	31.85±0.22	29.26±0.23	31.36±0.27	31.08±0.28	34.12 ±0.33	<u>41.36±0.20</u>	43.81±0.24
Texas	56.19±1.61	45.95±1.58	55.93±1.76	59.79±1.71	61.35±1.25	56.30±1.61	57.18±1.64	69.56 ±1.71	<u>70.46±1.61</u>	75.74±1.61
Wisc.	55.12±1.51	46.90±1.28	46.77±1.48	58.49±1.23	55.60±1.25	55.37±1.47	57.22±1.50	67.87 ±1.62	<u>70.98±1.50</u>	73.90±1.61
Cornell	44.78±1.45	44.46±1.37	45.33±1.55	47.73±1.51	45.11±1.47	46.81±1.56	47.57±1.52	57.31 ±1.60	<u>67.22±1.48</u>	69.11±1.53
R-Emp.	51.66±0.17	53.93±0.14	OOM	52.53±0.13	52.38±0.21	58.58±0.14	53.31±0.23	71.23 ±0.18	61.99±0.14	<u>66.52±0.13</u>
Cora	87.73±0.25	<u>88.31±0.29</u>	29.86±0.28	87.73±0.31	87.94±0.26	87.72±0.27	87.86±0.28	87.54±0.25	<u>91.39±0.24</u>	91.71±0.29
Citeseer	76.01±0.25	76.22±0.34	22.31±0.34	76.43±0.32	76.34±0.27	<u>76.49±0.28</u>	76.12±0.28	76.09 ±0.29	<u>81.14±0.34</u>	82.85±0.38
Pubmed	88.20±0.10	<u>88.51±0.10</u>	OOM	88.16±0.11	88.42±0.10	88.34±0.10	88.44±0.10	88.14±0.10	<u>88.69±0.10</u>	90.01±0.13

Classification results for the GCN model



Main Takeaways

- Several ongoing research efforts on **rewiring techniques**
 - Batch Ollivier-Ricci Flow (BORF) [Nguyen et al., ICML '23]
 - Spectral rewiring: FoSR [Karhadkar et al., ICLR '23]; GOKU [Liang et al., ICML '25]
 - Greedy Total Resistance (GTR) rewiring [Black et al., ICML '23]
- Going further: leverage the **internal functioning** of GNNs
 - Impact of width, depth, and topology on over-squashing [Di Giovanni et al., ICML '23]
- ☹ Highly-effective **deep GNNs**?
 - Not quite there yet

Clustering and Pooling for GNNs

w/ A. Duval

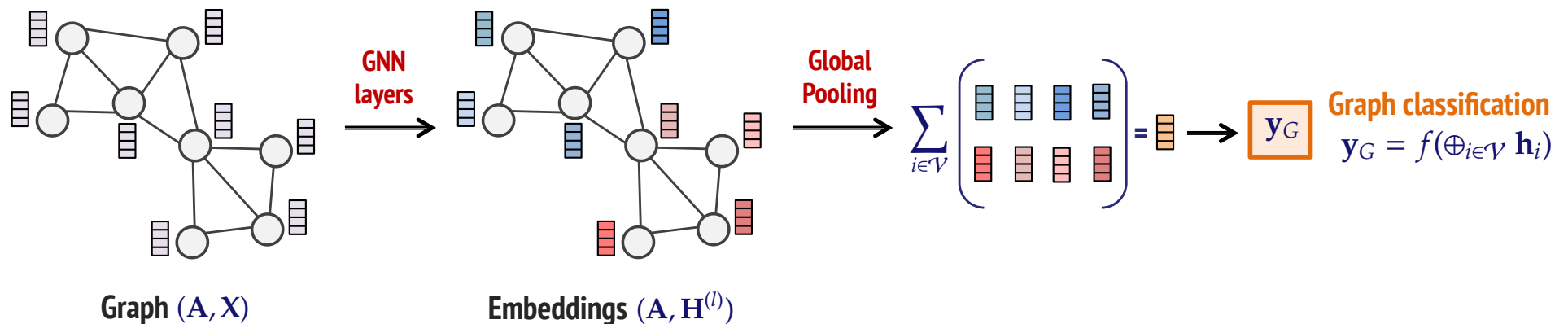
CIKM '22



Alexandre Duval

Why Structure-aware Graph Pooling?

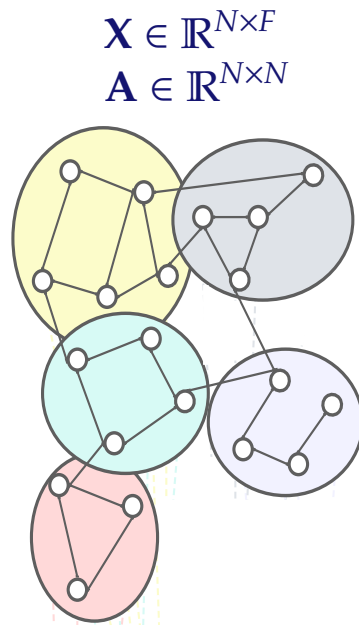
Graph-level tasks (e.g., graph classification)



Global Pooling

- 😊 **Fast** and **easy** to compute
- ☹ Discards information about the **graph (clustering) structure**

(Motif) Spectral Clustering with GNNs



- Clustering based on both **graph structure** $\mathbf{A} \in \mathbb{R}^{N \times N}$ and **node features** $\mathbf{X} \in \mathbb{R}^{N \times F}$

- Compute new node features using GNN layers

$$\bar{\mathbf{X}} = \text{GNN}(\mathbf{A}, \mathbf{X}; \Theta_{\text{GNN}})$$

- Learn a **cluster assignment** matrix using an MLP

$$\mathbf{S} = \text{FC}(\bar{\mathbf{X}}; \Theta) \in \mathbb{R}^{N \times K}$$

- Train GNN and MLP by optimizing a **clustering loss**

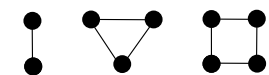
motif clustering loss $\longrightarrow \mathcal{L}_{mc} = -\frac{1}{K} \cdot \text{Tr} \left(\frac{\mathbf{S}^\top \mathbf{A}_M \mathbf{S}}{\mathbf{S}^\top \mathbf{D}_M \mathbf{S}} \right)$

motif adjacency matrix

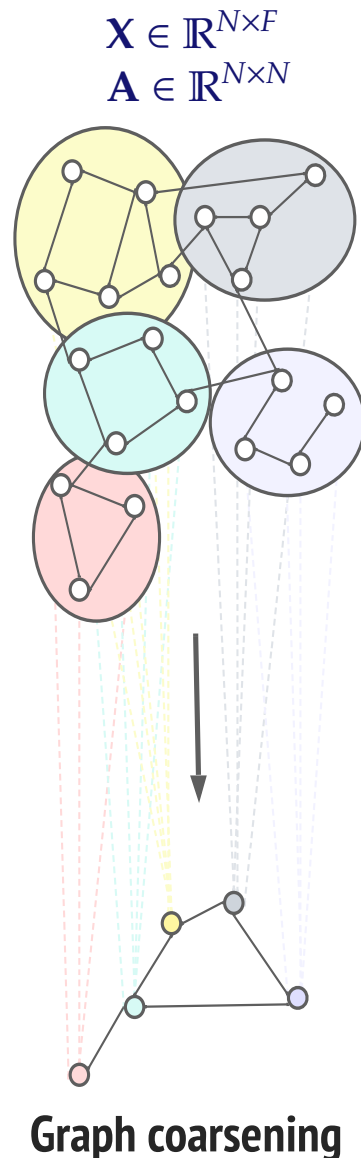
Types of motifs:

— We can allow combinations of motifs

— **Hosc** model: Higher-order spectral clustering



(Motif) Spectral Clustering with GNNs



- Clustering based on both **graph structure** $\mathbf{A} \in \mathbb{R}^{N \times N}$ and **node features** $\mathbf{X} \in \mathbb{R}^{N \times F}$

- Compute new node features using GNN layers

$$\bar{\mathbf{X}} = \text{GNN}(\mathbf{A}, \mathbf{X}; \Theta_{\text{GNN}})$$

- Learn a **cluster assignment** matrix using an MLP

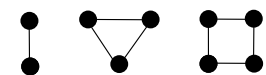
$$\mathbf{S} = \text{FC}(\bar{\mathbf{X}}; \Theta) \in \mathbb{R}^{N \times K}$$

- Train GNN and MLP by optimizing a **clustering loss**

motif clustering loss $\rightarrow \mathcal{L}_{mc} = -\frac{1}{K} \cdot \text{Tr} \left(\frac{\mathbf{S}^\top \mathbf{A}_M \mathbf{S}}{\mathbf{S}^\top \mathbf{D}_M \mathbf{S}} \right)$

motif adjacency matrix

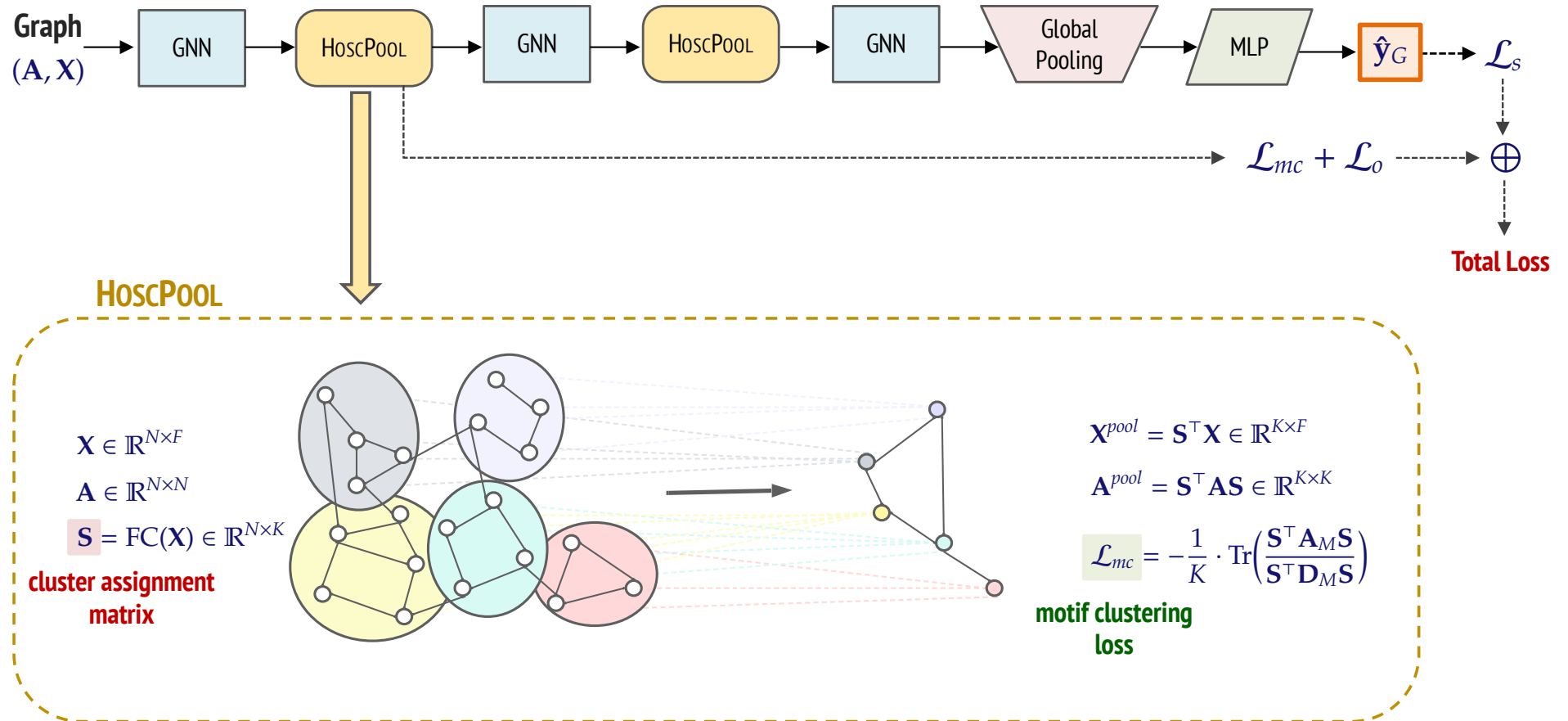
Types of motifs:



— We can allow combinations of motifs

— **Hosc** model: Higher-order spectral clustering

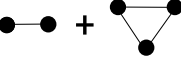
HoscPool: Hierarchical Clustering-based Pooling



HoscPool: Experiments on Graph Clustering

HoscPool as an end-to-end higher-order clustering algorithm

- Architecture:** message passing layer (GCN) + MLP

	spectral clustering	motif spectral clustering					
Dataset	SC	MSC	DiffPool	MinCutPool	HoscPool-1	HoscPool-2	HoscPool
<i>Cora</i>	0.150 $\pm$ 0.002	0.056 $\pm$ 0.014	0.308 $\pm$ 0.023	0.391 $\pm$ 0.028	0.435 $\pm$ 0.032	<u>0.464</u> $\pm$ 0.036	0.502 $\pm$ 0.029
<i>PubMed</i>	0.183 $\pm$ 0.002	0.002 $\pm$ 0.000	0.098 $\pm$ 0.006	0.214 $\pm$ 0.066	<u>0.230</u> $\pm$ 0.071	0.215 $\pm$ 0.073	0.260 $\pm$ 0.054
<i>Photo</i>	<u>0.592</u> $\pm$ 0.008	0.451 $\pm$ 0.011	0.171 $\pm$ 0.004	0.086 $\pm$ 0.014	0.495 $\pm$ 0.068	0.513 $\pm$ 0.083	0.598 $\pm$ 0.101
<i>PC</i>	0.464 $\pm$ 0.002	0.166 $\pm$ 0.009	0.043 $\pm$ 0.008	0.026 $\pm$ 0.006	0.497 $\pm$ 0.040	<u>0.499</u> $\pm$ 0.036	0.528 $\pm$ 0.041
<i>CS</i>	0.273 $\pm$ 0.006	0.011 $\pm$ 0.009	0.383 $\pm$ 0.048	0.431 $\pm$ 0.060	0.479 $\pm$ 0.022	<u>0.701</u> $\pm$ 0.029	0.731 $\pm$ 0.018
<i>DBLP</i>	0.027 $\pm$ 0.003	0.005 $\pm$ 0.006	0.186 $\pm$ 0.014	0.334 $\pm$ 0.026	<u>0.326</u> $\pm$ 0.027	0.284 $\pm$ 0.026	0.312 $\pm$ 0.027
<i>Polblogs</i>	0.017 $\pm$ 0.000	0.014 $\pm$ 0.001	0.317 $\pm$ 0.010	0.440 $\pm$ 0.390	<u>0.992</u> $\pm$ 0.003	0.994 $\pm$ 0.001	0.994 $\pm$ 0.005
<i>Email-eu</i>	0.485 $\pm$ 0.030	0.382 $\pm$ 0.019	0.096 $\pm$ 0.034	0.253 $\pm$ 0.028	0.317 $\pm$ 0.026	0.488 $\pm$ 0.025	<u>0.476</u> $\pm$ 0.021
<i>Syn1</i>	0.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.035 $\pm$ 0.000	0.043 $\pm$ 0.008	0.041 $\pm$ 0.006	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
<i>Syn2</i>	0.003 $\pm$ 0.000	0.050 $\pm$ 0.003	0.081 $\pm$ 0.008	0.902 $\pm$ 0.028	0.942 $\pm$ 0.028	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
<i>Syn3</i>	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.067 $\pm$ 0.001	0.052 $\pm$ 0.002	0.115 $\pm$ 0.006	<u>0.826</u> $\pm$ 0.005	1.000 $\pm$ 0.000

Clustering results (NMI) for the HoscPool model

HoscPool: Experiments on Graph Classification



Method	<i>Proteins</i>	<i>NCI1</i>	<i>Mutagen.</i>	<i>DD</i>	<i>Reddit-B</i>	<i>Cox2-MD</i>	<i>ER-MD</i>	<i>b-hard</i>
NoPOOL	71.6 $\pm$ 4.1	77.1 $\pm$ 1.9	78.1 $\pm$ 1.3	71.2 $\pm$ 2.2	80.1 $\pm$ 2.6	58.7 $\pm$ 3.2	72.2 $\pm$ 2.9	66.5 $\pm$ 0.5
RANDOM	75.7 $\pm$ 3.2	77.0 $\pm$ 1.7	79.2 $\pm$ 1.3	77.1 $\pm$ 1.5	89.3 $\pm$ 2.6	62.9 $\pm$ 3.6	73.0 $\pm$ 4.5	69.1 $\pm$ 2.1
GMT	75.0 $\pm$ 4.2	74.9 $\pm$ 4.3	79.4 $\pm$ 2.2	78.1 $\pm$ 3.2	86.7 $\pm$ 2.6	58.9 $\pm$ 3.6	74.3 $\pm$ 4.5	70.1 $\pm$ 3.4
MINCUTPOOL	75.9 $\pm$ 2.4	76.8 $\pm$ 1.6	78.6 $\pm$ 1.8	78.4 $\pm$ 2.8	89.0 $\pm$ 1.4	58.9 $\pm$ 5.1	75.5 $\pm$ 4.0	72.6 $\pm$ 1.5
DIFFPOOL	73.8 $\pm$ 3.7	76.7 $\pm$ 2.1	77.9 $\pm$ 2.3	76.3 $\pm$ 2.1	87.3 $\pm$ 2.4	57.1 $\pm$ 4.8	76.8 $\pm$ 4.8	70.7 $\pm$ 2.0
EIGPOOL	74.2 $\pm$ 3.1	75.0 $\pm$ 2.2	75.2 $\pm$ 2.7	75.1 $\pm$ 1.8	82.8 $\pm$ 2.1	59.8 $\pm$ 3.4	73.1 $\pm$ 3.8	69.1 $\pm$ 3.1
SAGPOOL	70.6 $\pm$ 3.5	74.1 $\pm$ 3.9	74.4 $\pm$ 2.7	71.5 $\pm$ 4.1	74.7 $\pm$ 4.5	56.9 $\pm$ 9.7	71.7 $\pm$ 8.2	39.6 $\pm$ 9.6
ASAP	74.4 $\pm$ 2.6	74.3 $\pm$ 1.6	76.8 $\pm$ 2.4	73.2 $\pm$ 2.5	84.1 $\pm$ 1.1	60.5 $\pm$ 5.5	74.5 $\pm$ 5.9	70.5 $\pm$ 1.7
HoscPOOL-1	76.7 $\pm$ 2.5	77.3 $\pm$ 1.6	79.8 $\pm$ 1.6	78.8 $\pm$ 2.0	91.2 $\pm$ 1.0	61.6 $\pm$ 3.5	76.2 $\pm$ 4.2	72.4 $\pm$ 0.8
HoscPOOL-2	77.0 $\pm$ 3.1	80.3 $\pm$ 2.0	92.8 $\pm$ 1.5	66.4 $\pm$ 4.6	92.8 $\pm$ 1.5	66.4 $\pm$ 4.6	77.9 $\pm$ 4.3	73.5 $\pm$ 0.8
HoscPOOL	77.5 $\pm$ 2.3	79.9 $\pm$ 1.7	82.3 $\pm$ 1.3	79.4 $\pm$ 1.8	93.6 $\pm$ 0.9	64.6 $\pm$ 3.9	78.2 $\pm$ 3.8	74.0 $\pm$ 0.4

Classification accuracy for the HoscPOOL model

Main Takeaways

- **End-to-end clustering** with GNNs
 - 😊 Leverages graph topology + node features
 - 😊 Avoids eigenvalue decomposition of the Laplacian matrix
 - 😊 Allows clustering of out-of-sample graphs
- **Higher-order** topological information
 - 😊 Flexible mechanism of **HoscPool**
- ☹ Performance of hierarchical **clustering-based pooling**
 - Graph classification benchmarks: small molecular graphs

Generalization of GNNs

w/ Y. Abbahaddou, J. Lutzeyer, A. Aboussalah, M. Vazirgiannis

ICML '25



Y. Abbahaddou

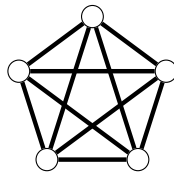
Generalization on GNNs and Challenges

- **Goal:** learn a predictor f_θ that performs well on **new graphs** $\mathcal{G} \sim \mathcal{D}_{\text{test}}$ different from those in the **training set** $\mathcal{D}_{\text{train}}$
- Why a **challenging** problem?
 - Topology shift
 - Size shift
 - Feature distribution shift

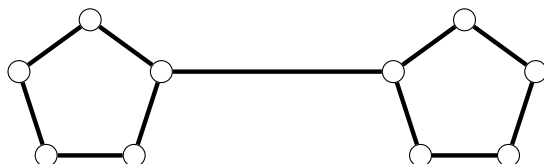
- Regularization
- Architecture refinements
- **Data augmentation**

Topology shift

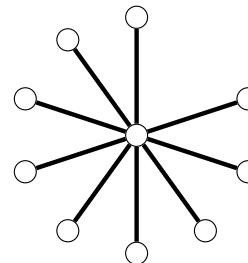
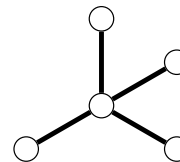
$\mathcal{D}_{\text{train}}$



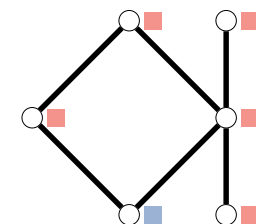
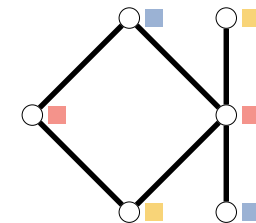
$\mathcal{D}_{\text{test}}$



Size shift



Feature distribution shift



A Theoretical Framework for Graph Data Augmentation

- Augmentation strategy:** For each training graph $(\mathcal{G}_n, y_n)$, the generator $\mathbf{A}_\lambda$ produces M samples

$$(\tilde{\mathcal{G}}_n^m, \tilde{y}_n^m) \sim \mathbf{A}_\lambda(\mathcal{G}_n, y_n), \quad m = 1, \dots, M$$

Vanilla training

Training set

$$\mathcal{D}_{\text{train}} = \{(\mathcal{G}_n, y_n)\}_{n=1}^N \longrightarrow \tilde{\mathcal{D}}_{\text{train}} = \mathcal{D}_{\text{train}} \cup \{(\tilde{\mathcal{G}}_n^m, \tilde{y}_n^m)\}_{n=1..N}^{m=1..M}$$

Empirical risk

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^N \ell(\mathcal{G}_n, \theta) \longrightarrow \mathcal{L}_{\text{aug}}(\theta) = \frac{1}{N} \sum_{n=1}^N \frac{1}{M} \sum_{m=1}^M \ell(\tilde{\mathcal{G}}_n^m, \theta)$$

cross-entropy

Optimal parameters

$$\theta_\star \simeq \hat{\theta} = \arg \min_{\theta} \mathcal{L}(\theta) \longrightarrow \hat{\theta}_{\text{aug}} = \arg \min_{\theta} \mathcal{L}_{\text{aug}}(\theta)$$

A Theoretical Framework for Graph Data Augmentation

Goals of **augmentation**: minimize the **generalization error**

Excess risk
(generalization error)

$$\mathbb{E}_{\mathcal{G} \sim \mathcal{D}} [\ell(\mathcal{G}, \hat{\theta}_{\text{aug}})] - \mathbb{E}_{\mathcal{G} \sim \mathcal{D}} [\ell(\mathcal{G}, \theta_{\star})]$$

true risk of augmented model true risk of model on data distribution

Theorem (informal)

Let $\ell(\cdot, \cdot) \in [0, 1]$ be a classification loss function. Then, with a probability at least $1 - \delta$ over the samples $\mathcal{D}_{\text{train}}$, we have

$$\mathbb{E}_{\mathcal{G} \sim \mathcal{D}} [\ell(\mathcal{G}, \hat{\theta}_{\text{aug}})] - \mathbb{E}_{\mathcal{G} \sim \mathcal{D}} [\ell(\mathcal{G}, \theta_{\star})] \leq 2\mathcal{R}(\ell_{\text{aug}}) + 5 \sqrt{\frac{2 \log(4/\delta)}{N}} + 2L_{\text{Lip}} \mathbb{E}_{\mathcal{G} \sim \mathcal{D}, \tilde{\mathcal{G}} \sim A_{\lambda}} [\|\tilde{\mathcal{G}} - \mathcal{G}\|]$$

generalization error Rademacher complexity
capacity of the GNN to fit random noise augmentation error
distance between the original graph and the augmented samples

$\|\mathbf{h}_{\tilde{\mathcal{G}}} - \mathbf{h}_{\mathcal{G}}\|$

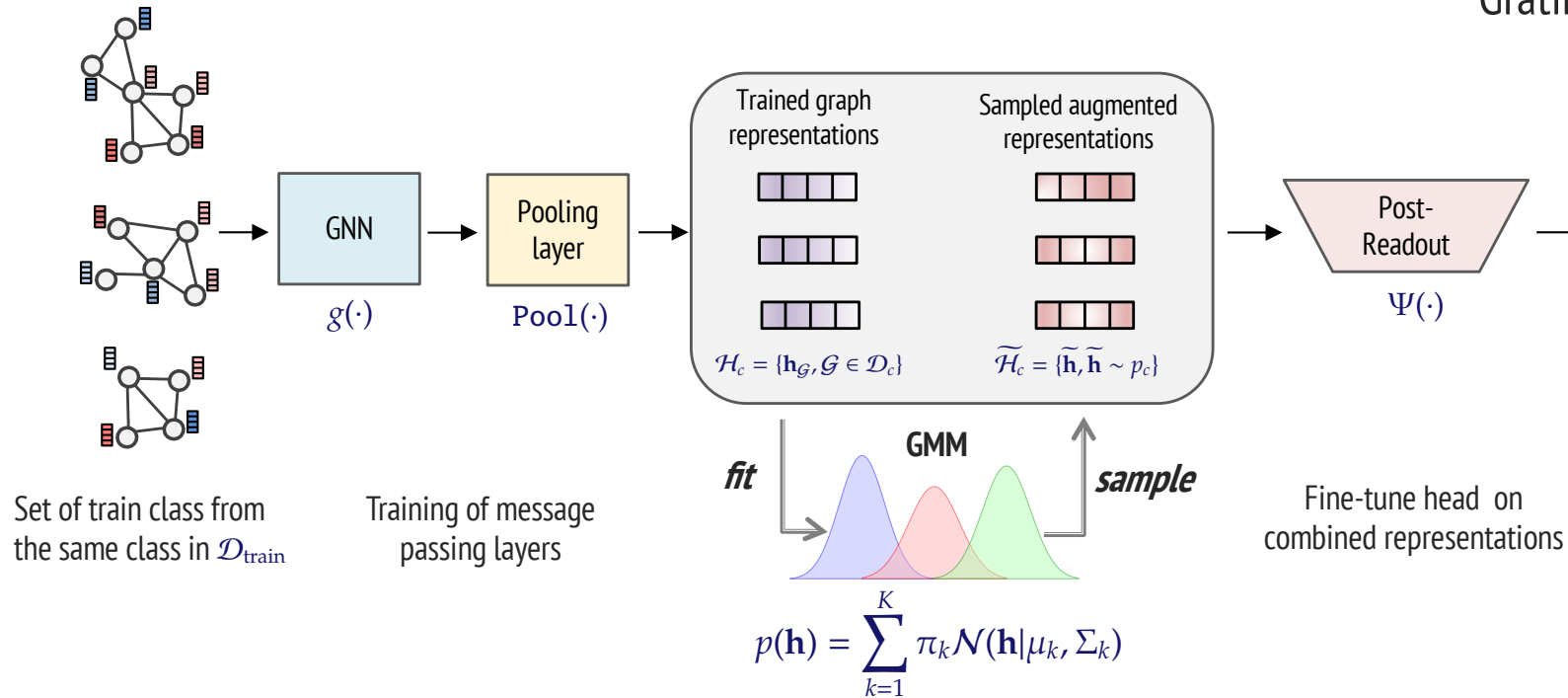
$$\mathcal{R}(\ell_{\text{aug}}) = \mathbb{E}_{\epsilon_n \sim P_{\epsilon}} \left[\sup_{\theta \in \Theta} \left| \frac{1}{N} \sum_{n=1}^N \epsilon_n \ell_{\text{aug}}(\mathcal{G}_n, \theta) \right| \right]$$

If augmented graphs are too far from originals, the bound becomes large (bias increases)

GRATIN: GMM-based Augmentation



Gratin dauphinois 🥔



1. Train GNN $f(\cdot, \theta) = \Psi \circ \text{Pool} \circ g$
2. Embed graphs $\mathcal{H} = \{\mathbf{h}_{\mathcal{G}}, \mathcal{G} \in \mathcal{D}_{\text{train}}\}$
3. Class-wise split $\mathcal{H} = \bigcup_{c=1}^C \mathcal{H}_c$, where $\mathcal{D}_c = \{\mathcal{G}_n \in \mathcal{D}_{\text{train}}, y_n = c\}$
4. Fit GMM and sample $p_c = \text{GMM}(\mathcal{H}_c)$
5. Fine-tune head: freeze $g(\cdot)$ and train $\Psi(\cdot)$ on $\mathcal{H} \cup \tilde{\mathcal{H}}$

GRATIN: Experimental Results

Model	IMDB-BIN	IMDB-MUL	MUTAG	PROTEINS	DD
No Aug.	73.00 $\pm$ 4.94	47.73 $\pm$ 2.64	73.92 $\pm$ 5.09	69.99 $\pm$ 5.35	69.69 $\pm$ 2.89
DropEdge	71.70 $\pm$ 5.42	45.67 $\pm$ 2.46	73.39 $\pm$ 8.86	70.07 $\pm$ 3.86	69.35 $\pm$ 3.37
DropNode	74.00$\pm$3.44	43.80 $\pm$ 3.54	73.89 $\pm$ 8.53	69.81 $\pm$ 4.61	69.01 $\pm$ 3.95
SubMix	<u>72.70$\pm$5.59</u>	46.00 $\pm$ 2.44	<u>77.13$\pm$9.69</u>	67.57 $\pm$ 4.56	70.11 $\pm$ 4.48
$\mathcal{G}$ -Mixup	72.10 $\pm$ 3.27	48.33 $\pm$ 3.06	88.77$\pm$5.71	65.68 $\pm$ 5.03	61.20 $\pm$ 3.88
GeoMix	69.69 $\pm$ 3.37	<u>49.80$\pm$4.71</u>	74.39 $\pm$ 7.37	69.63 $\pm$ 5.37	68.50 $\pm$ 3.74
GRATIN	71.00 $\pm$ 4.40	49.82$\pm$4.26	76.05 $\pm$ 6.74	70.97$\pm$5.07	71.90$\pm$2.81

Graph classification results for the **GRATIN** model on a GCN backbone

Main Takeaways

- “**Mixup**”-like techniques on graphs
 - ☺ Improve generalization through augmentations
 - $\mathcal{G}$ -Mixup [Han et al., ICML ‘22], GeoMix [Zhao et al. KDD ‘24]
- **GRATIN**: augmentations on the graph **embedding-space**
 - ☺ Combines structure + features
 - ☺ Avoid costly graph alignment
 - ☺ Scalability
- Augmentations with **Gaussian Mixture Models (GMMs)**
 - ☺ Expressive yet simple
 - ☺ A GMM is a universal approximator of densities

Outline of the Presentation

Part I. Brief introduction to Graph Neural Networks (GNN)

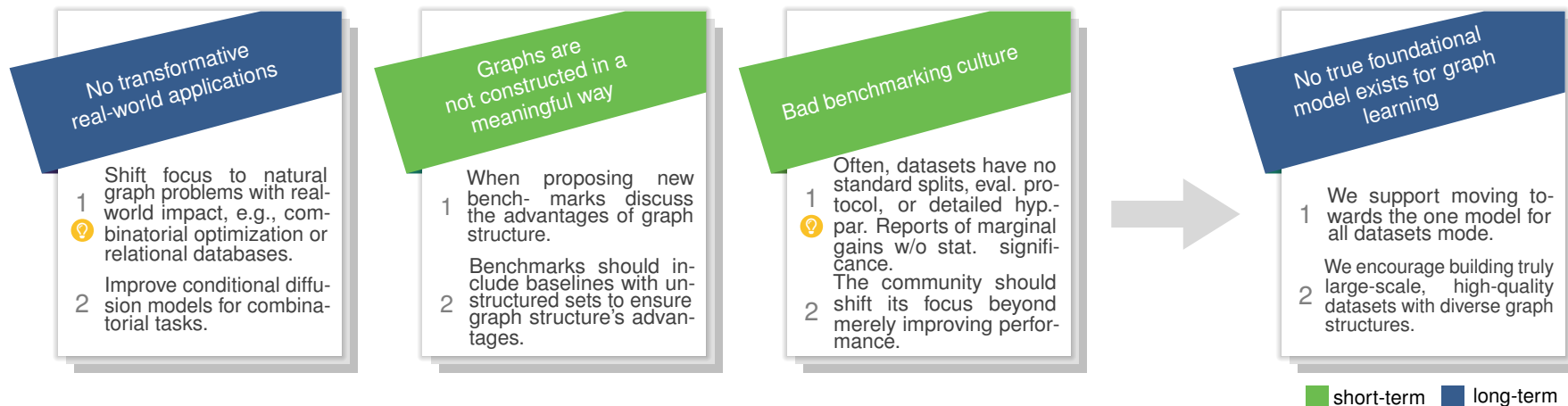
Part II. Topics in GNN model design

Part III. Perspectives and ongoing work

Perspectives

- Leverage **structural information** and beyond for GNNs
 - Rewiring, graph pooling, and generalization
- On **complex models**
 - GNNs, Hypergraph GNNs, Simplicial Complex Neural Networks, ...
- On proper **model evaluation**
 - Realistic datasets; proper experimental protocol; proper metrics
- On **problem modeling** and **practical applications**
 - Type of graph; node features; which learning problem

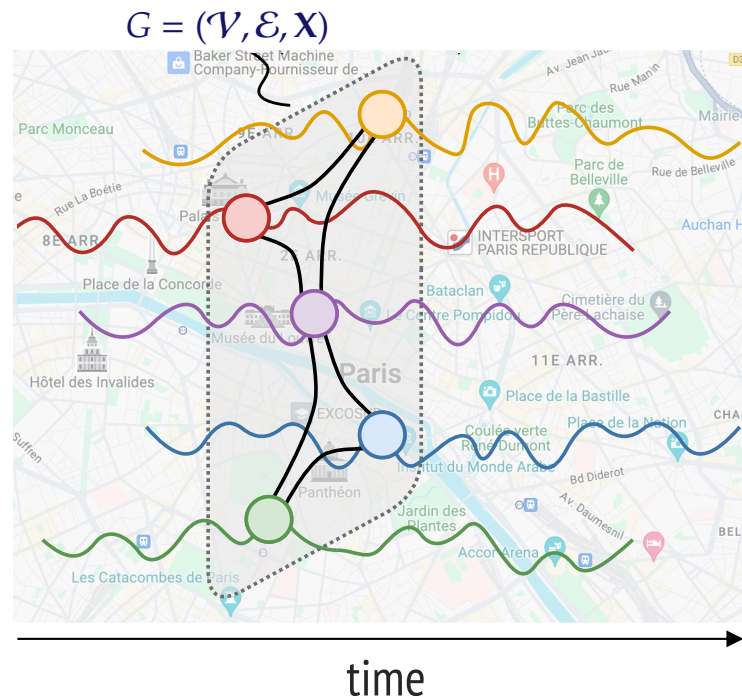
Beware of GNN Evaluation and Benchmarking



Position: Graph Learning Will Lose Relevance Due To Poor Benchmarks

Maya Bechler-Speicher^{*1,2} Ben Finkelshtein^{*3} Fabrizio Frasca^{*4} Luis Müller^{*5} Jan Tönshoff^{*5}
Antoine Siraudin⁵ Viktor Zaverkin⁶ Michael M. Bronstein³ Mathias Niepert⁷ Bryan Perozzi⁸
Mikhail Galkin⁸ Christopher Morris⁵

Spatiotemporal Graph Learning



- **Spatiotemporal prediction with GNNs**
 - Enhance predictions with **relational inductive biases**
- **Tasks**
 - Time series forecasting
 - Missing value completion (imputation)
 - Graph structure learning

Continuous Product Graph Neural Networks

Aref Einizade
LTCI, Télécom Paris
Institut Polytechnique de Paris
aref.einizade@telecom-paris.fr

Fragkiskos D. Malliaros
CentraleSupélec, Inria
Université Paris-Saclay
fragkiskos.malliaros@centralesupelec.fr

Jhony H. Giraldo
LTCI, Télécom Paris
Institut Polytechnique de Paris
jhony.giraldo@telecom-paris.fr

[Einizade et al., NeurIPS '24]

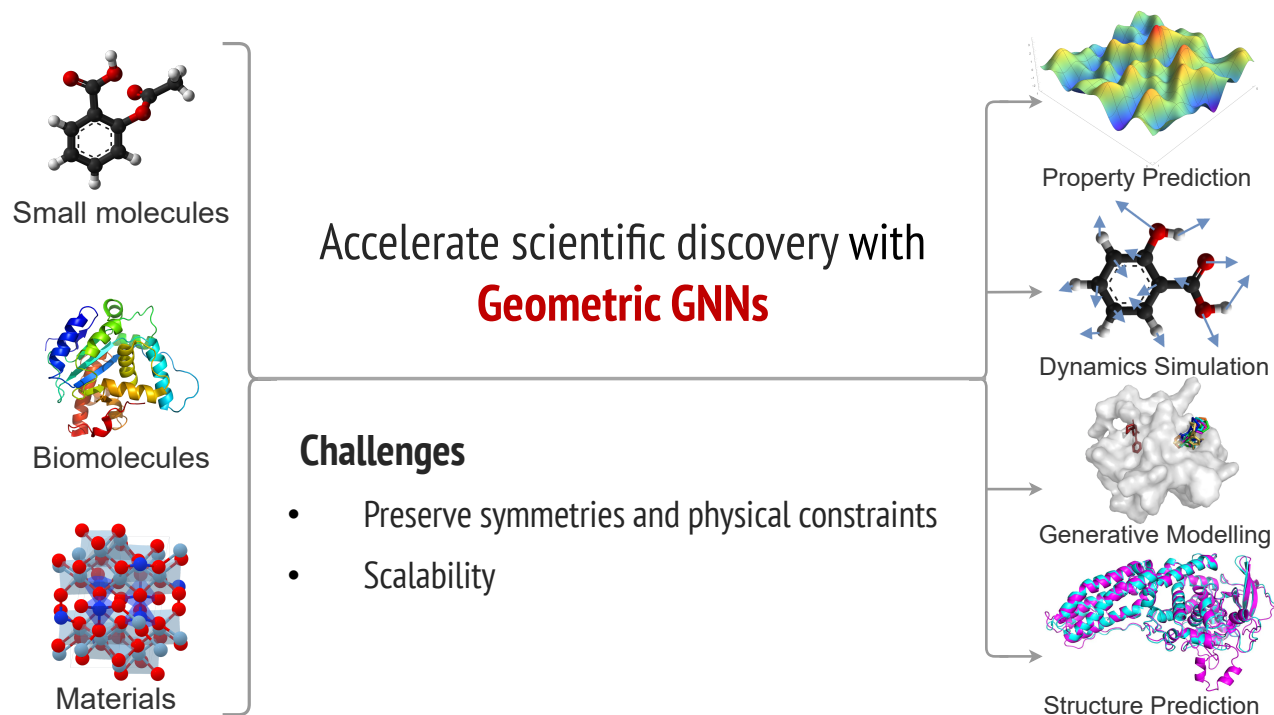
Gegenbauer Graph Neural Networks for Time-varying Signal Reconstruction

Jhon A. Castro-Correa, Jhony H. Giraldo, Mohsen Badiy, Frangkiskos D. Malliaros

[Castro-Correa et al., IEEE Trans. Neural Netw. Learn. Syst. '24]

Geometric Graph Neural Networks (GNNs)

for 3D atomic systems



FAENet: Frame Averaging Equivariant GNN for Materials Modeling

Alexandre Duval^{*1,2} Victor Schmidt^{*2} Alex Hernandez Garcia² Santiago Miret³ Fragkiskos D. Malliaros¹
Yoshua Bengio^{2,4} David Rolnick^{2,5}

[Duval et al., ICML '23]

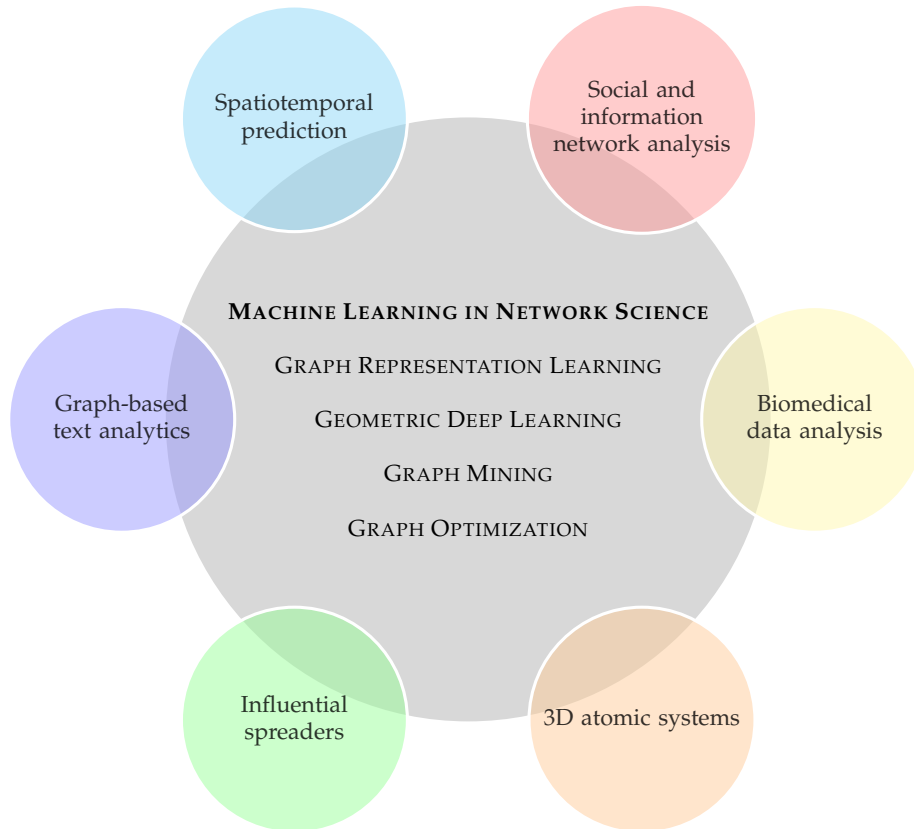
A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems

Alexandre Duval^{*,1,2} Simon V. Mathis^{*,3} Chaitanya K. Joshi^{*,3} Victor Schmidt^{*,1,4}
Santiago Miret⁵ Fragkiskos D. Malliaros² Taco Cohen⁶
Pietro Liò³ Yoshua Bengio^{1,4} Michael Bronstein⁷

¹Mila ²Université Paris-Saclay[†] ³University of Cambridge ⁴Université de Montréal
⁵Intel Labs ⁶Qualcomm AI Research[‡] ⁷University of Oxford

[Duval et al., arXiv '24]

Thank You!



Web: <http://fragkiskosm.github.io>

Email: fragkiskos.malliaros@centralesupelec.fr

**Supported
in part by:**

anr [®]
agence nationale
de la recherche

INSTITUT DATAiA
Science des données, Intelligence & Société

**NOKIA
BELL
LABS**

ifp *Energies
nouvelles*

AI4IDF [®]
Région
Île-de-France

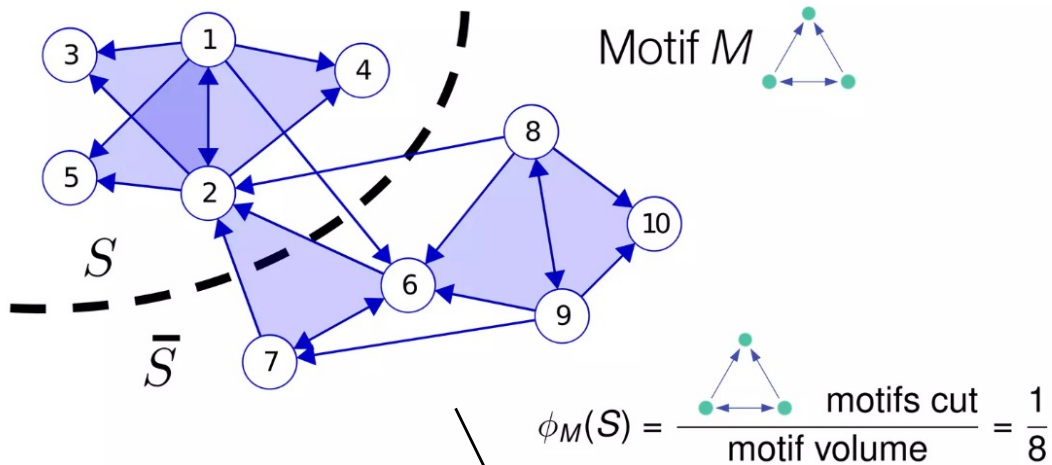
ENTALPIC

Labex
DigiCosme

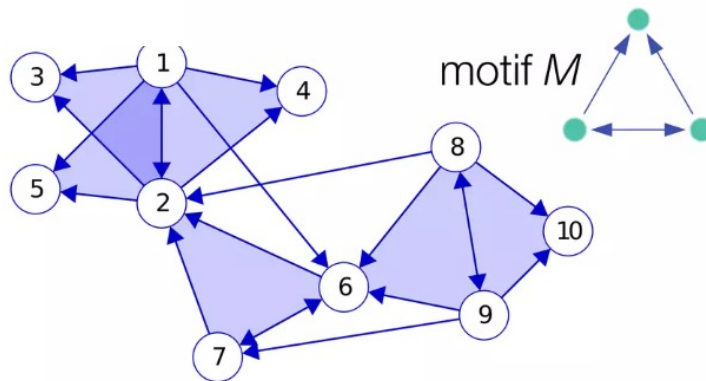
Backup Slides

Motif Spectral Clustering – Reformulation

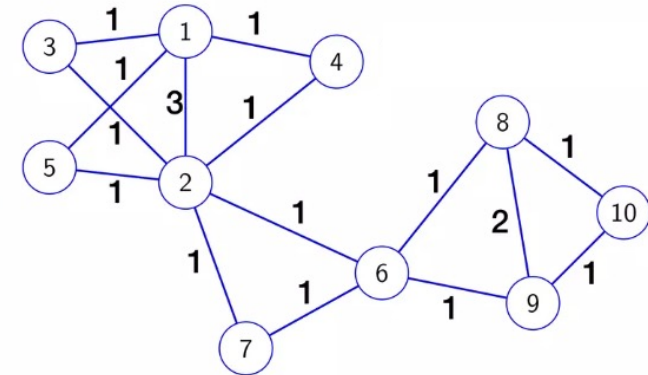
Motif-based conductance



Graph G



Weighted motif graph A_M



$$A_M(i, j) = \#\{\text{instances of motif } M \text{ that contain nodes } i \text{ and } j\}$$