Enhancing Graph Neural Networks with Topological Structures

Fragkiskos D. Malliaros

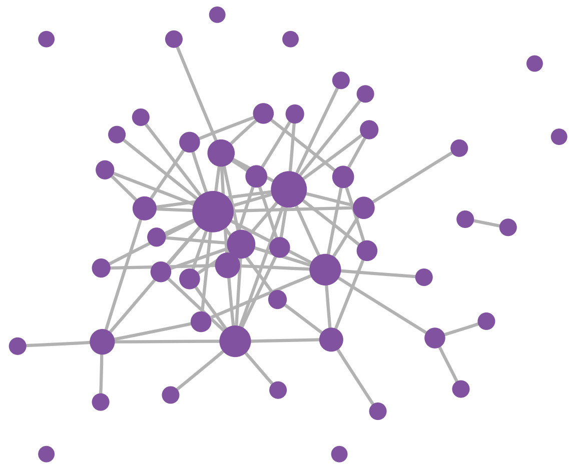
CentraleSupélec, Inria, Université Paris-Saclay

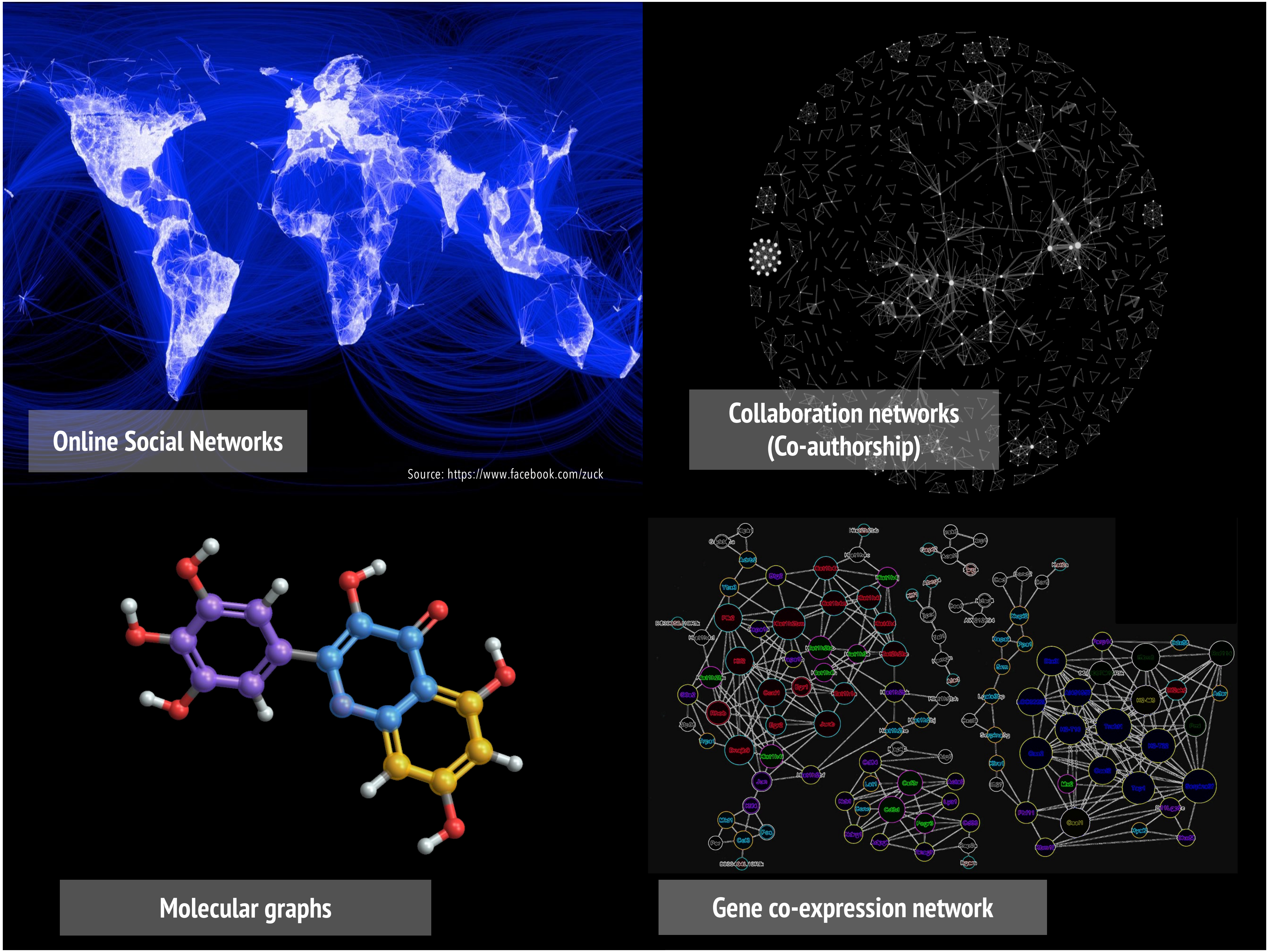
June 26, 2025

interactions + complexity

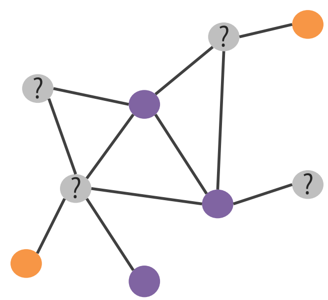
=

complex network

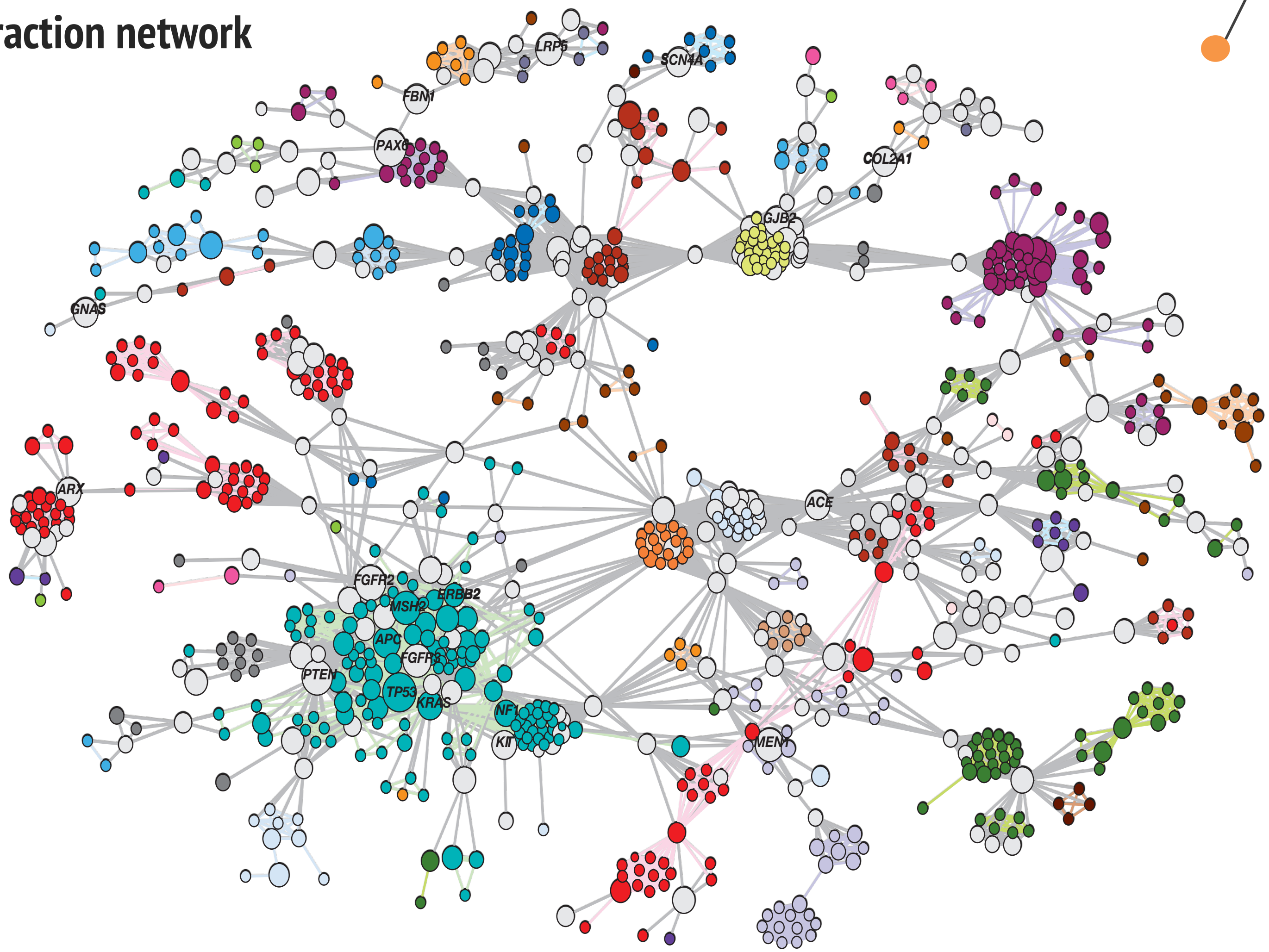



Online Social Networks
Source: https://www.facebook.com/zuck
Collaboration networks
(Co-authorship)
Molecular graphs
Gene co-expression network

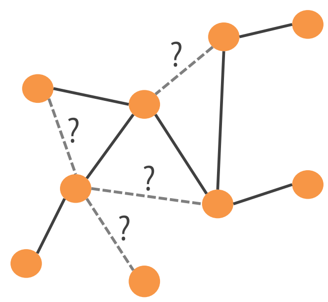
Node Classification



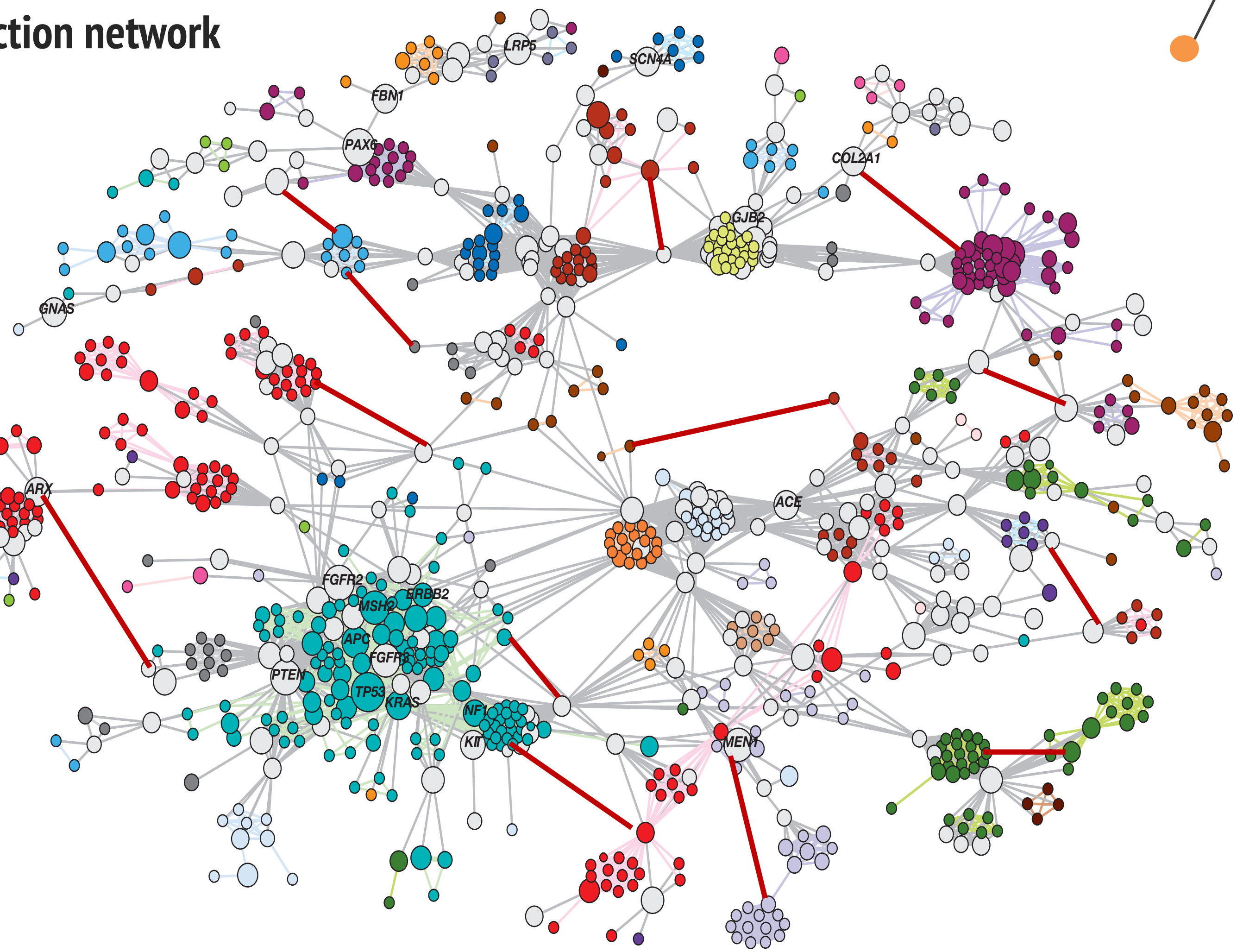
Gene interaction network



Link Prediction

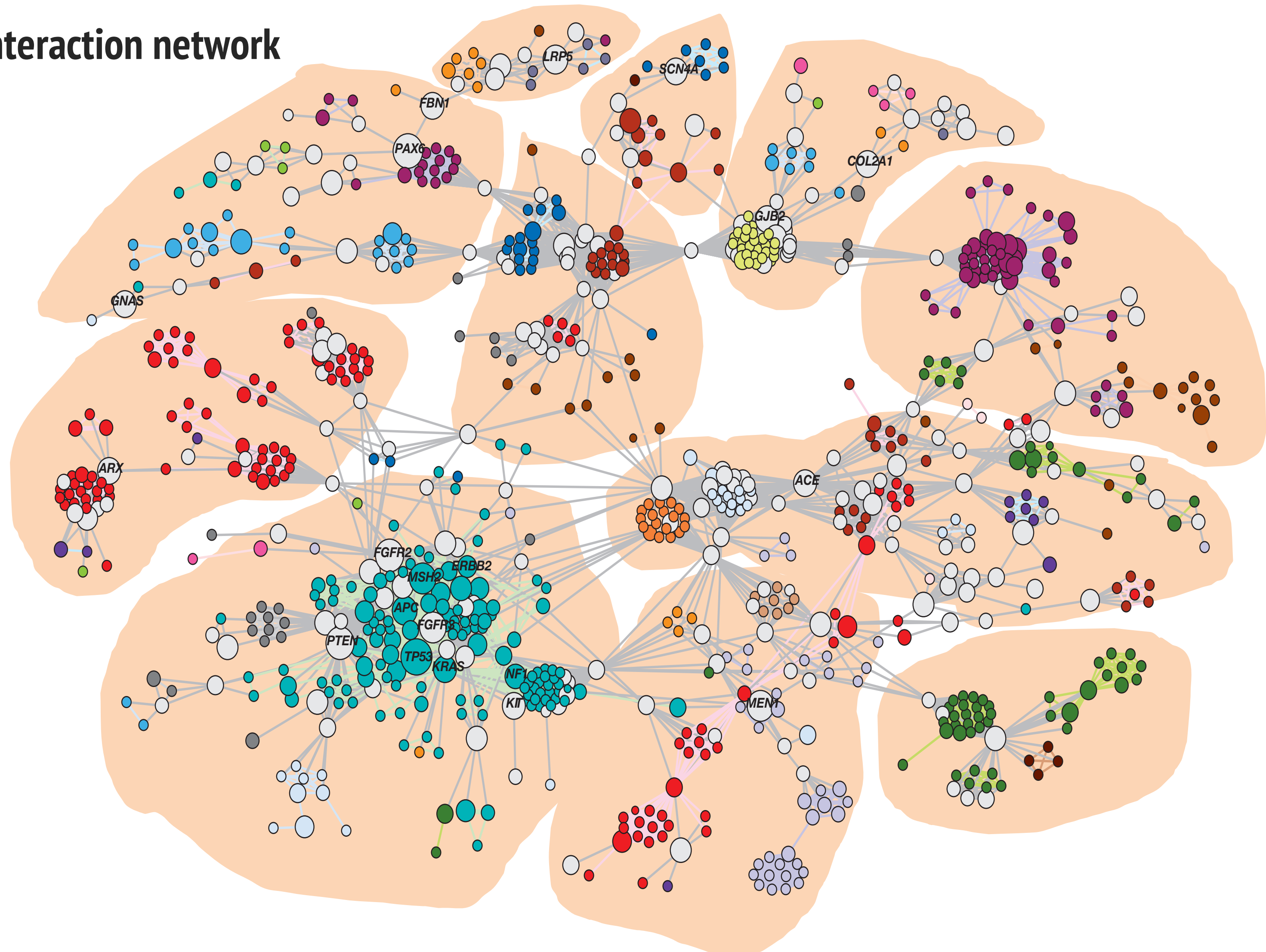


Gene interaction network



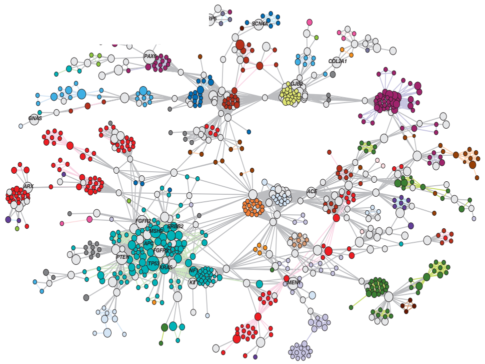
Community Detection (or Graph Clustering)

Gene interaction network



Graph Representation Learning

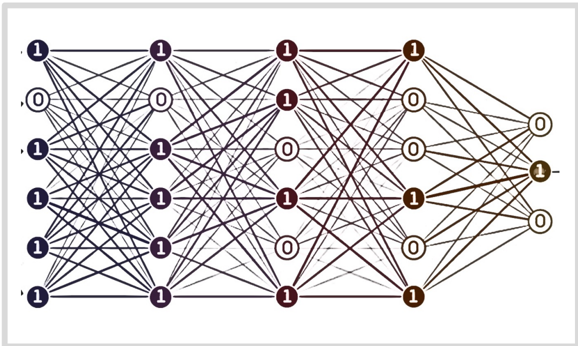
Learn features by transforming the graph into a low-dimensional latent representation



Input graph

$G = (\mathcal{V}, \mathcal{E})$

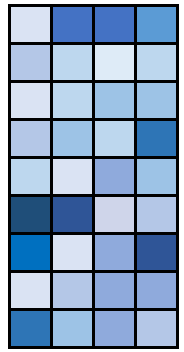
Graph Representation Learning



- Matrix factorization
- Random walk node embedding models
- **Graph Neural Networks**

$\Phi(\cdot)$

$[\Phi_1, \ldots, \Phi_d] \in \mathbb{R}^d,\ d \ll |\mathcal{V}|$



$|\mathcal{V}|$

Embeddings

- Link prediction
- Node classification
- Clustering
- Anomaly detection
- Influential maximization
- ...

Network analysis tasks

Challenges. Trustworthy models while dealing with the **complex** structure of **information-rich, large-scale** graphs

Outline of the Presentation

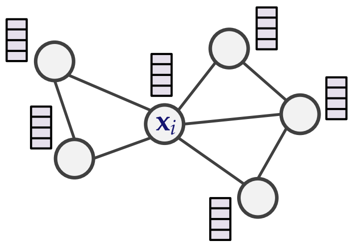
Part I. Brief introduction to Graph Neural Networks (GNN)

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

Part III. Perspectives and ongoing work

Graph Neural Networks (GNNs)

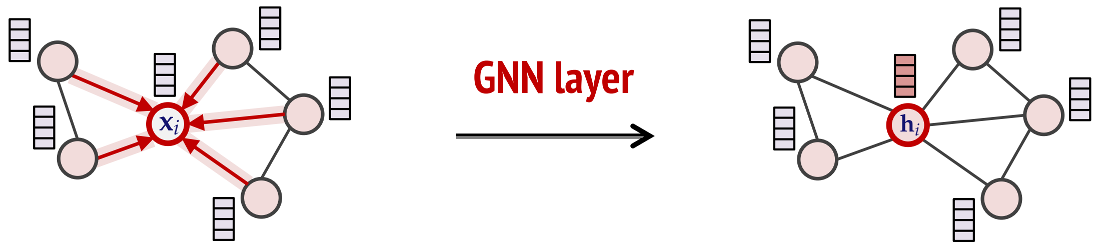
Graph $(\mathbf{A}, \mathbf{X})$



[Defferrard et al., NeurIPS '16], [Kipf and Welling, ICLR '17], [Gilmer et al., ICML '17], [Veličković et al., ICLR '18], [Bronstein et al., Geometric Deep Learning '21]

Graph Neural Networks (GNNs)

Graph $(\mathbf{A}, \mathbf{X})$ **Embeddings** $(\mathbf{A}, \mathbf{H})$

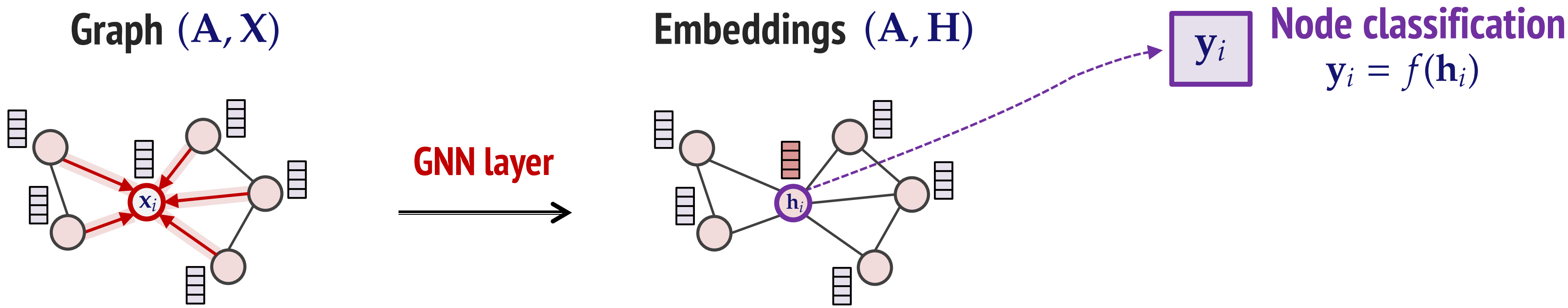


$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in \mathcal{N}_i} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

activation function (e.g., ReLU) **aggregation function** (e.g., sum) **message function**

[Defferrard et al., NeurIPS ‘16], [Kipf and Welling, ICLR ‘17], [Gilmer et al., ICML ‘17], [Veličković et al., ICLR ‘18], [Bronstein et al., Geometric Deep Learning ‘21]

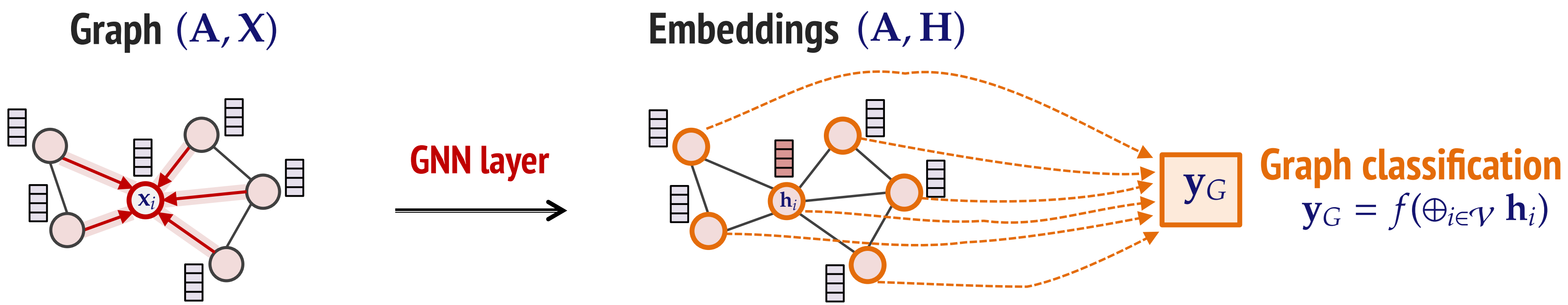
Graph Neural Networks (GNNs)



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

[Defferrard et al., NeurIPS '16], [Kipf and Welling, ICLR '17], [Gilmer et al., ICML '17], [Veličković et al., ICLR '18], [Bronstein et al., Geometric Deep Learning '21]

Graph Neural Networks (GNNs)

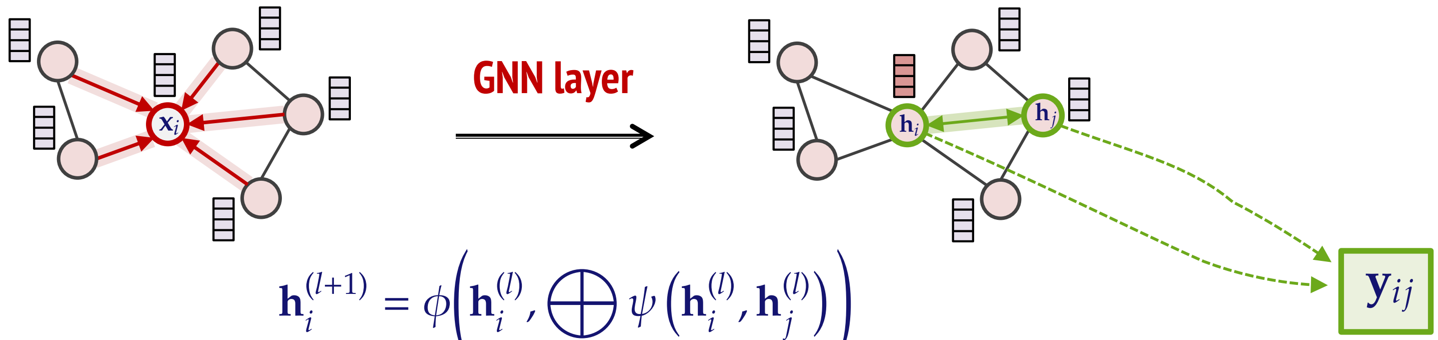


$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in \mathcal{N}_i} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

[Defferrard et al., NeurIPS ‘16], [Kipf and Welling, ICLR ‘17], [Gilmer et al., ICML ‘17], [Veličković et al., ICLR ‘18], [Bronstein et al., Geometric Deep Learning ‘21]

Graph Neural Networks (GNNs)

Graph $(\mathbf{A}, \mathbf{X})$ Embeddings $(\mathbf{A}, \mathbf{H})$



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

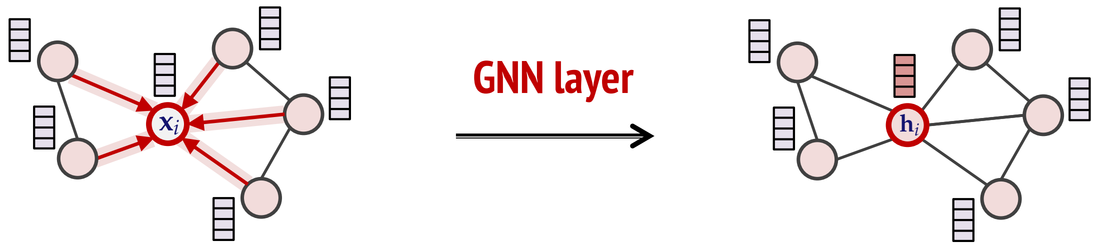
Link prediction

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

[Defferrard et al., NeurIPS '16], [Kipf and Welling, ICLR '17], [Gilmer et al., ICML '17], [Veličković et al., ICLR '18], [Bronstein et al., Geometric Deep Learning '21]

Graph Neural Networks (GNNs)

Graph $(\mathbf{A}, \mathbf{X})$ Embeddings $(\mathbf{A}, \mathbf{H})$



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

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

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

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

Different instances of GNN layers

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

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

GNNs with Attention (e.g., GAT)

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

[Defferrard et al., NeurIPS '16], [Kipf and Welling, ICLR '17], [Gilmer et al., ICML '17], [Veličković et al., ICLR '18], [Wu et al., ICML '19]

Challenges in GNN Model Design

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

Leverage structural (topological) information and beyond.

On the Trade-off between Over-smoothing and Over-squashing in GNNs

w/ J.H. Giraldo, K. Skianis, T. Bouwmans

CIKM '23

J.H. Giraldo

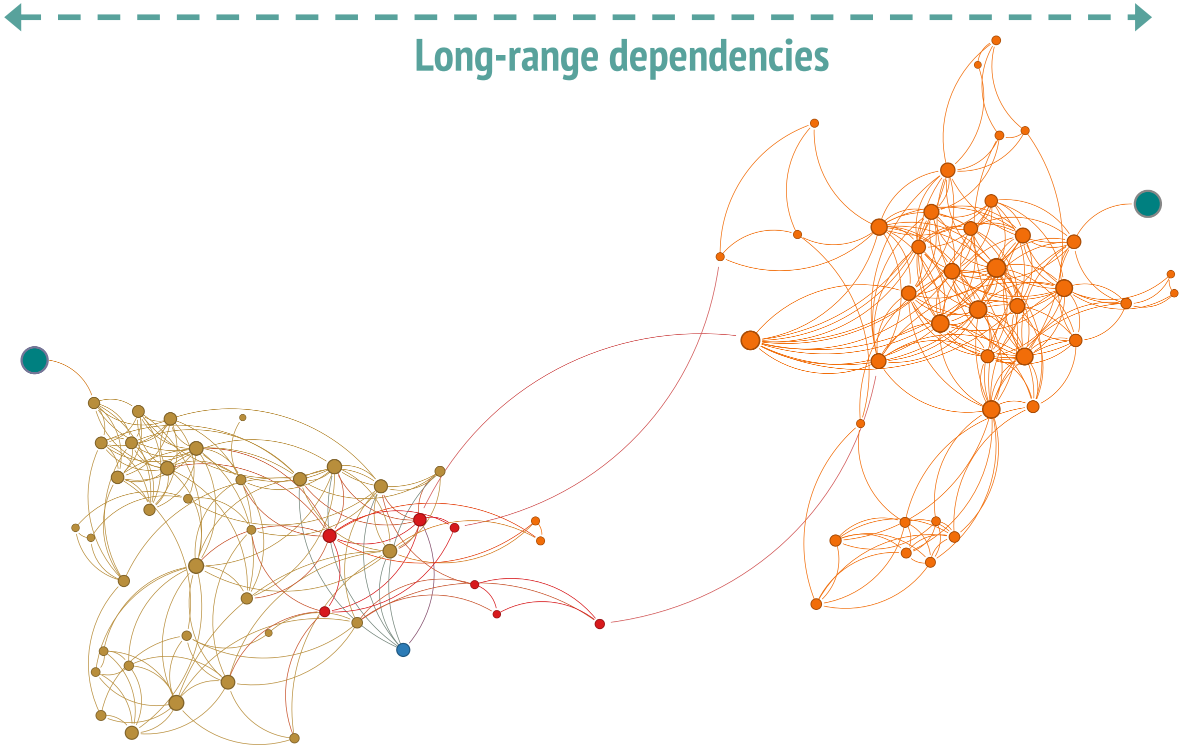
- Visiting PhD student
- Assistant Professor at Télécom Paris

TELECOM Paris
IP PARIS

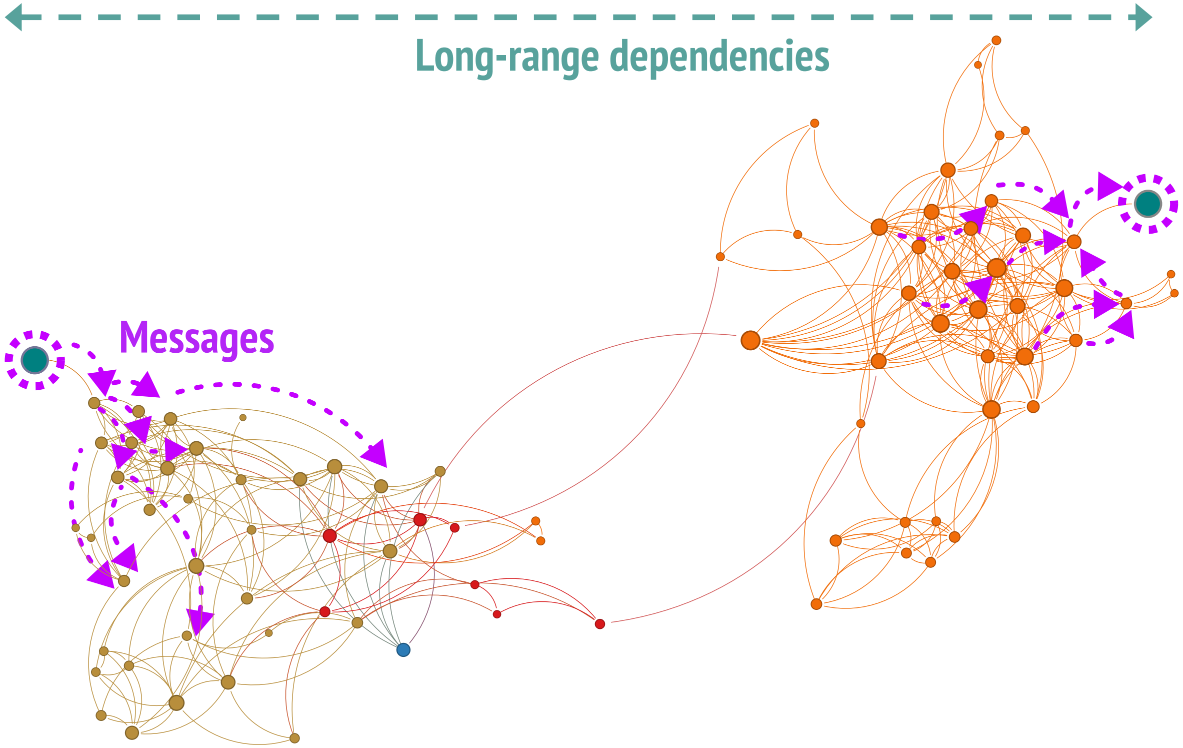
Univ. of Ioannina

La Rochelle Université

Long-range Dependencies and Deep GNNs

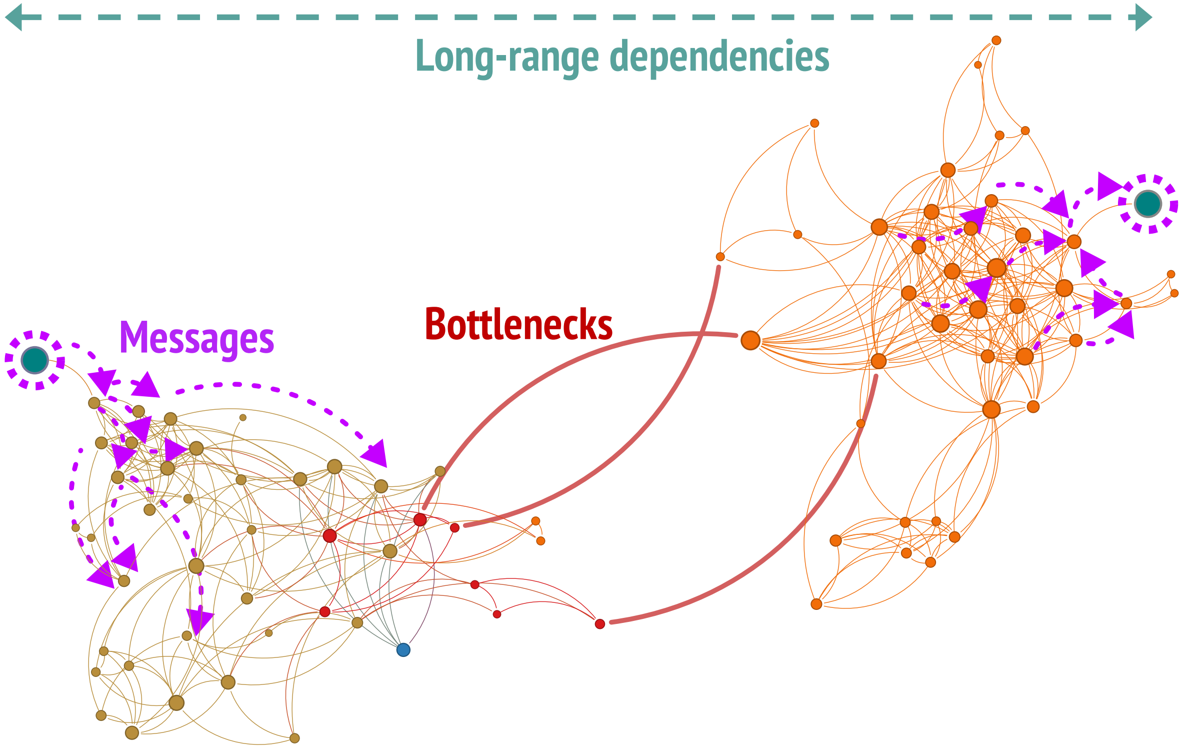


Long-range Dependencies and Deep GNNs



- **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

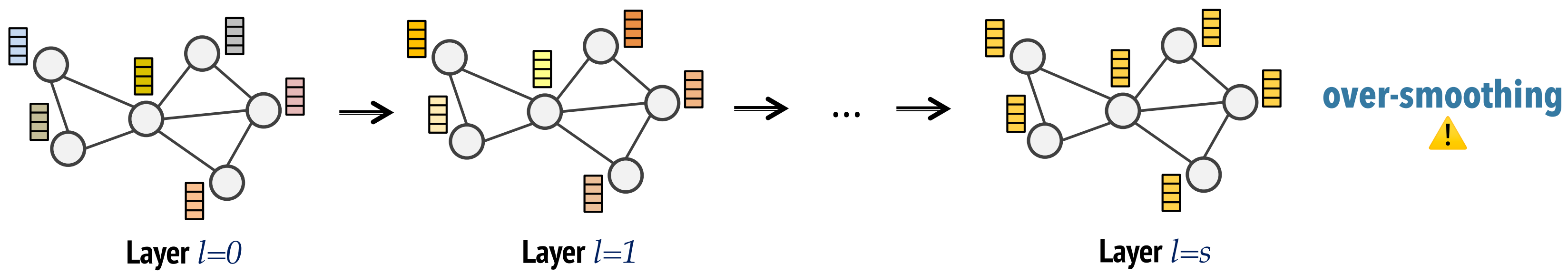
[Oono and Suzuki, ICLR '20], [Alon and Yahav, ICLR '21]

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 the **curvature-based** algorithm 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 $\mathbf{f} : \mathcal{V} \to \mathbb{R}$, we can compute s such that

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

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

  - GNNs converge **exponentially fast** to the **stationary distribution** π when stacking several layers $\longrightarrow$ **over-smoothing**
  - The convergence depends on the spectral gap λ_2

[Chung, Spectral Graph Theory, '92], [Oono and Suzuku, ICLR '20], [Wu et al. ICML '19]

The Over-smoothing – Over-squashing Trade-off

Cheeger constant and bottlenecks

- The **Cheeger constant** h_G of a graph

edge boundary
(# of edges crossing the cut)

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

sum of node
degrees in $\mathcal{S}$

  – 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**

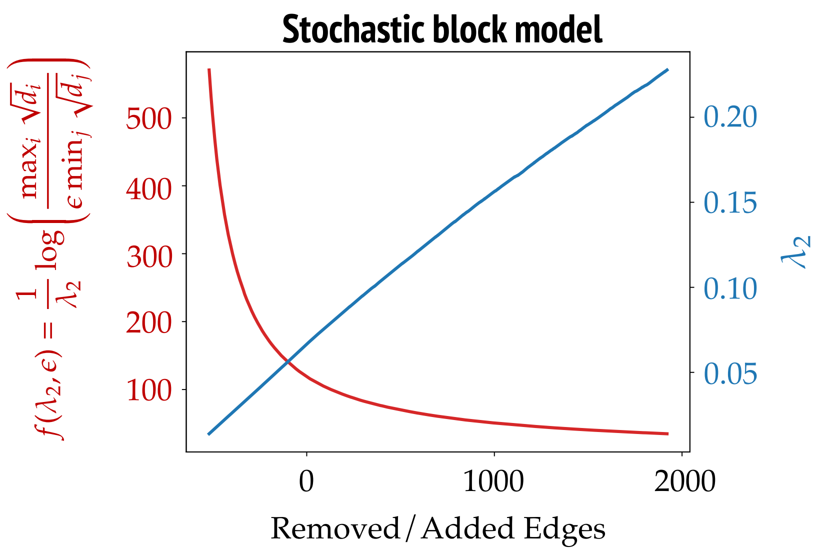
[Chung, Spectral Graph Theory, '92]

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

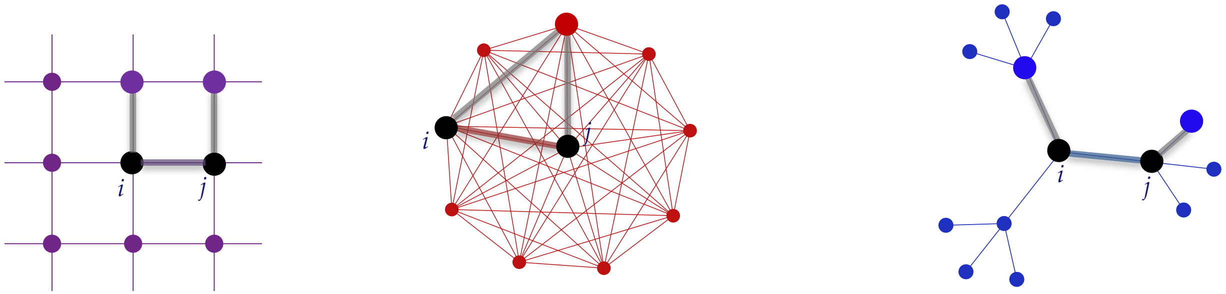


- We can increase mixing time by **removing** some edges
  - Alleviate over-smoothing
- We increase λ_2 by **adding** edges, improving h_G
  - Alleviate over-squashing

[Chung, Spectral Graph Theory, '92]

SJLR: 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



$\kappa(i,j) = 0$ $\kappa(i,j) > 0$ $\kappa(i,j) < 0$

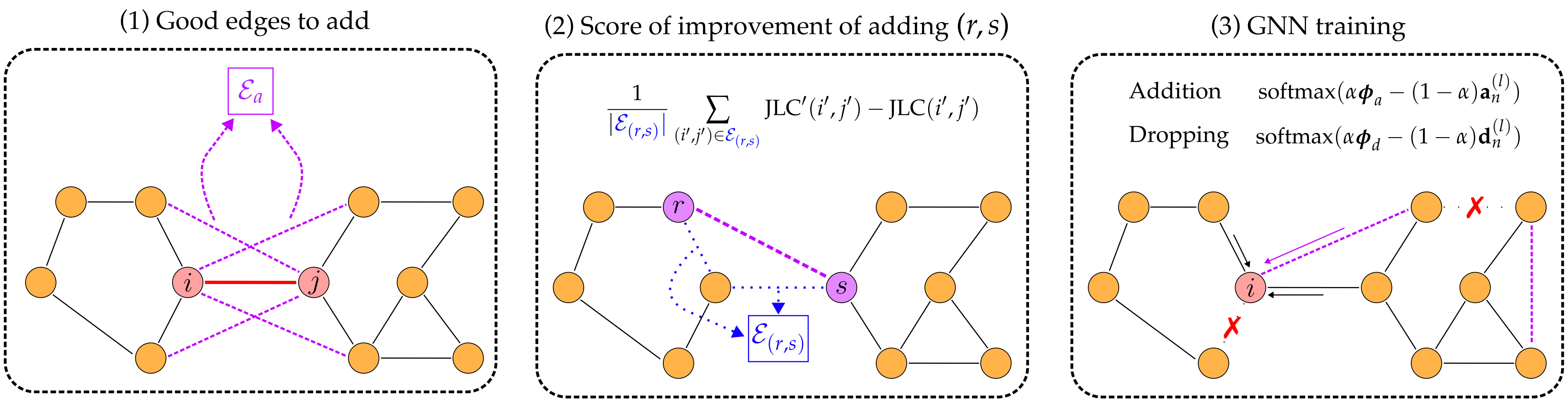
[Bronstein, Physics-inspired GNNs '23]

- **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**

[Topping et al., ICLR '21], [Ollivier, J. Funct. Anal. '09], [Jost and Liu, Discrete Comput. Geom. '14]

SJLR: The Algorithm

Stochastic Jost and Liu Curvature Rewiring



(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	Cornell	Texas	Wisconsin	Chameleon	Squirrel	Actor	Cora	Citeseer	Pubmed	*Overall*
Baseline	*67.34*±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	*57.44*
RDC [32]	63.78±1.68	59.47±1.00	50.89±1.00	40.33±0.51	*41.98*±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	*40.80*±0.55	41.68±0.39	*29.99*±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	*69.37*±0.31	77.74±0.42	55.66
FoSR [27]	56.65±0.93	50.01±1.37	*53.73*±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	*60.13*±0.89	**55.16**±0.95	**41.19**±0.46	41.86±0.29	29.89±0.20	*81.95*±0.25	**69.50**±0.33	*78.60*±0.33	**58.89**

Classification results for the GCN model

Method	Cornell	Texas	Wisconsin	Chameleon	Squirrel	Actor	Cora	Citeseer	Pubmed	*Overall*
Baseline	53.40±2.11	56.69±1.78	47.90±1.73	38.40±0.69	40.52±0.54	29.93±0.16	76.94±1.31	67.45±0.80	71.79±2.13	53.67
GDC [19]	58.65±1.43	57.42±0.74	45.93±1.05	38.13±0.55	36.63±0.31	**32.25**±0.17	76.02±1.70	66.22±1.13	71.91±2.30	53.68
DE [42]	*61.99*±1.04	*57.88*±0.81	*54.78*±0.89	40.38±0.47	41.28±0.32	30.62±0.17	80.59±0.80	**68.63**±0.51	74.47±1.65	*56.74*
PN [56]	53.11±1.36	50.47±1.04	48.72±1.65	**41.49**±0.68	39.72±0.33	22.58±0.29	75.55±0.42	64.16±0.41	73.81±0.52	52.18
DGN [57]	55.68±1.32	57.42±2.59	50.67±2.08	*40.99*±0.62	*41.72*±0.29	29.53±0.18	*80.65*±0.48	67.65±0.59	*74.95*±0.59	55.47
SDRF [48]	54.68±1.29	55.36±1.48	47.81±1.51	38.07±0.77	39.94±0.53	30.04±0.17	76.04±1.69	67.60±0.80	69.62±2.35	53.24
FoSR [27]	53.73±1.75	56.33±1.37	47.82±2.14	38.01±0.73	40.68±0.42	30.11±0.18	78.24±0.98	67.04±0.83	72.76±2.35	53.86
SJLR (ours)	**67.37**±1.64	**58.40**±1.48	**55.42**±0.92	40.17±0.49	**41.91**±0.34	*30.81*±0.18	**81.24**±0.77	*68.39*±0.69	**76.28**±0.96	**57.78**

Classification results for the SGC model

Main Takeaways

- Several ongoing research efforts on **rewiring techniques**
  - Batch Ollivier-Ricci Flow (BORF) [Nguyen et al., ICML '23]
  - First-order spectral rewiring (FoSR) [Karhadkar et al., ICLR '23]
  - Greedy Total Resistance (GTR) rewiring [Black et al., ICML '23]
  - Delaunay triangulation-based rewiring [Attali et al., ICML '24]
- 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

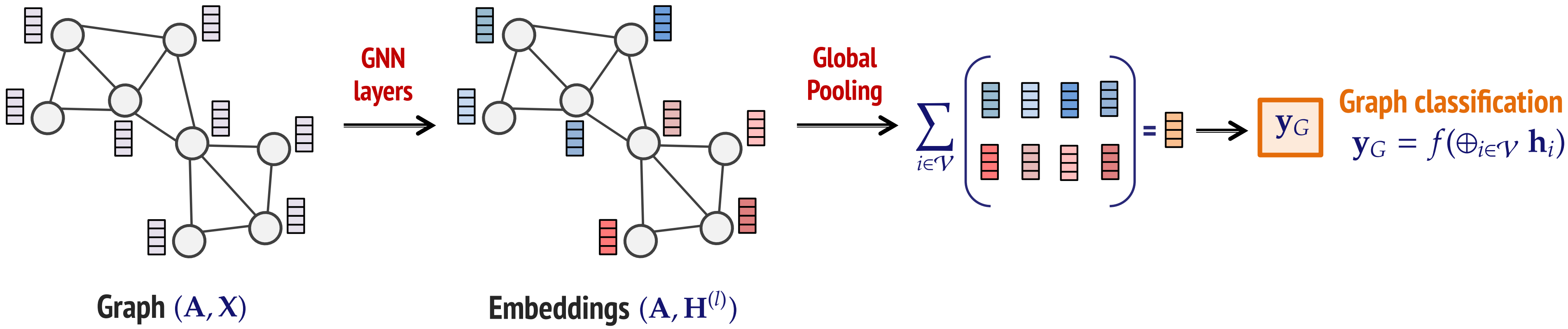
CIKM '22

- PhD '24
- Co-founder and CSO, Entalpic

Alexander Duval

Why Structure-aware Graph Pooling?

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



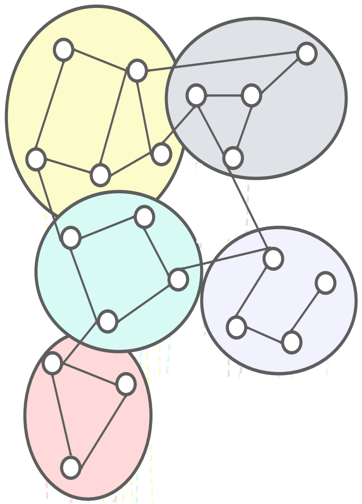
Global Pooling

☺ **Fast** and **easy** to compute

☹ Discards information about the **graph (clustering) structure**

(Motif) Spectral Clustering with GNNs

$\mathbf{X} \in \mathbb{R}^{N \times F}$
$\mathbf{A} \in \mathbb{R}^{N \times N}$



- 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

$$\bar{\mathbf{X}} = \mathrm{GNN}(\mathbf{A}, \mathbf{X}; \mathbf{\Theta}_{\mathrm{GNN}})$$

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

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

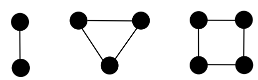
- Train GNN and MLP by optimizing a **clustering loss**

motif clustering loss

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

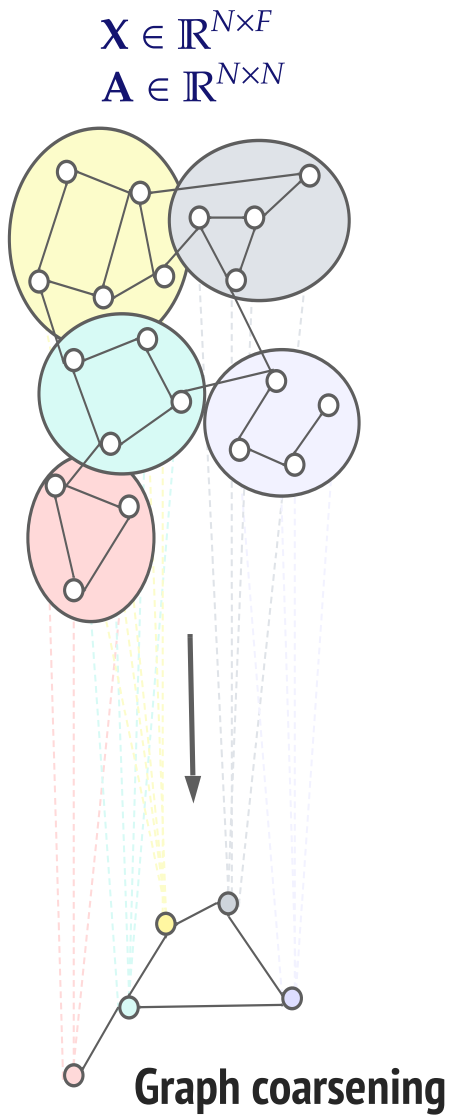
motif adjacency matrix

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

$$\bar{\mathbf{X}} = \mathrm{GNN}(\mathbf{A}, \mathbf{X}; \mathbf{\Theta}_{\mathrm{GNN}})$$

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

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

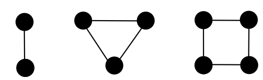
- Train GNN and MLP by optimizing a **clustering loss**

motif clustering loss

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

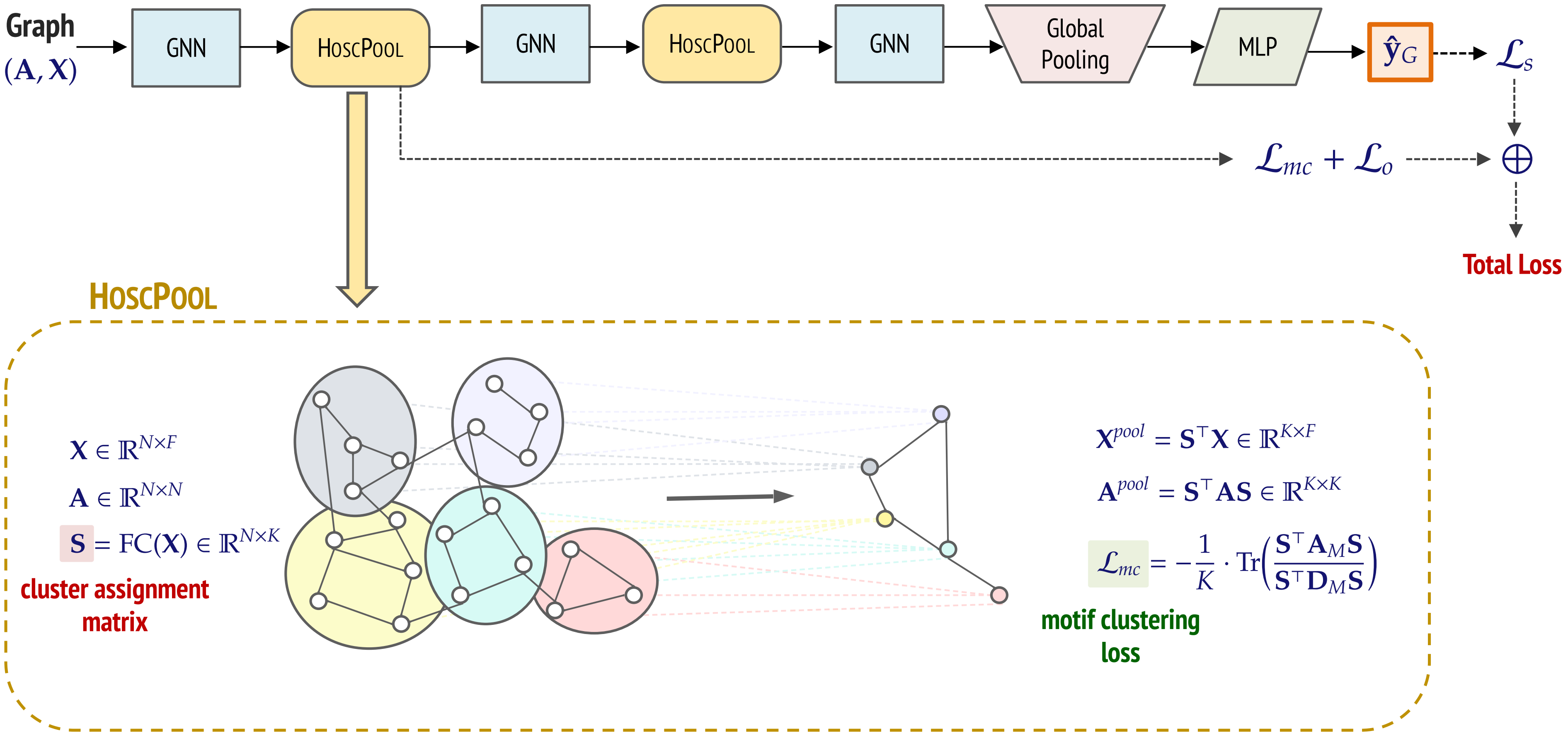
motif adjacency matrix

Types of motifs:



  - We can allow combinations of motifs
  - **HOSC** model: Higher-order spectral clustering

HoscPool: Hierachical Clustering-based Pooling

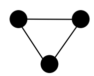
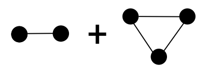


HoscPool: Experiments on Graph Clustering

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

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

spectral clustering (SC) — motif spectral clustering (MSC)

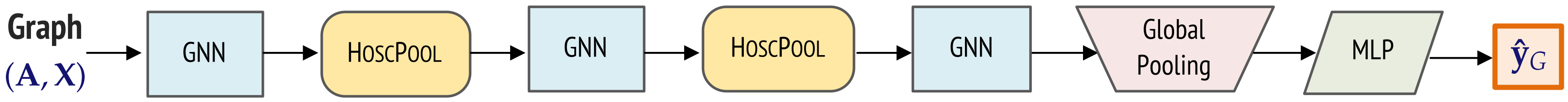




Dataset	SC	MSC	DiffPool	MinCutPool	HoscPool-1	HoscPool-2	HoscPool
Cora	0.150$\pm$0.002	0.056$\pm$0.014	0.308$\pm$0.023	0.391$\pm$0.028	0.435$\pm$0.032	0.464$\pm$0.036	**0.502**$\pm$0.029
PubMed	0.183$\pm$0.002	0.002$\pm$0.000	0.098$\pm$0.006	0.214$\pm$0.066	0.230$\pm$0.071	0.215$\pm$0.073	**0.260**$\pm$0.054
Photo	0.592$\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$0.101
PC	0.464$\pm$0.002	0.166$\pm$0.009	0.043$\pm$0.008	0.026$\pm$0.006	0.497$\pm$0.040	0.499$\pm$0.036	**0.528**$\pm$0.041
CS	0.273$\pm$0.006	0.011$\pm$0.009	0.383$\pm$0.048	0.431$\pm$0.060	0.479$\pm$0.022	0.701$\pm$0.029	**0.731**$\pm$0.018
DBLP	0.027$\pm$0.003	0.005$\pm$0.006	0.186$\pm$0.014	**0.334**$\pm$0.026	0.326$\pm$0.027	0.284$\pm$0.026	0.312$\pm$0.027
Polblogs	0.017$\pm$0.000	0.014$\pm$0.001	0.317$\pm$0.010	0.440$\pm$0.390	0.992$\pm$0.003	**0.994**$\pm$0.001	**0.994**$\pm$0.005
Email-eu	0.485$\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$0.025	0.476$\pm$0.021
Syn1	0.000$\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$0.000
Syn2	0.003$\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$0.000
Syn3	**1.000**$\pm$0.000	**1.000**$\pm$0.000	0.067$\pm$0.001	0.052$\pm$0.002	0.115$\pm$0.006	0.826$\pm$0.005	**1.000**$\pm$0.000

Clustering results (NMI) for the **HoscPool** model

[Benson et al, Science '16], [Ying et al., NeurIPS '18], [Bianchi et al., ICML '20]

HoscPool: Experiments on Graph Classification



Method	*Proteins*	*NCI1*	*Mutagen.*	*DD*	*Reddit-B*	*Cox2-MD*	*ER-MD*	*b-hard*
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	$76.7_{\pm 2.5}$	$77.3_{\pm 1.6}$	$79.8_{\pm 1.6}$	$\underline{78.8}_{\pm 2.0}$	$91.2_{\pm 1.0}$	$61.6_{\pm 3.5}$	$76.2_{\pm 4.2}$	$72.4_{\pm 0.8}$
HoscPool-2	$\underline{77.0}_{\pm 3.1}$	$\mathbf{80.3}_{\pm 2.0}$	$\underline{92.8}_{\pm 1.5}$	$\mathbf{66.4}_{\pm 4.6}$	$\mathbf{92.8}_{\pm 1.5}$	$\mathbf{66.4}_{\pm 4.6}$	$\underline{77.9}_{\pm 4.3}$	$\underline{73.5}_{\pm 0.8}$
HoscPool	$\mathbf{77.5}_{\pm 2.3}$	$\underline{79.9}_{\pm 1.7}$	$\mathbf{82.3}_{\pm 1.3}$	$\mathbf{79.4}_{\pm 1.8}$	$\mathbf{93.6}_{\pm 0.9}$	$\underline{64.6}_{\pm 3.9}$	$\mathbf{78.2}_{\pm 3.8}$	$\mathbf{74.0}_{\pm 0.4}$

Classification accuracy for the HoscPool model

[Baek et al., ICLR '21], [Bianchi et al., ICML '20], [Ying et al., NeurIPS '18], [Ma et al. KDD '19], [Lee et al., ICML '19], [Ranjan et al., AAAI '20]

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

- PhD student (graduating in July '25)
- Looking for a postdoc position.

Y. Abbahaddou

ÉCOLE POLYTECHNIQUE

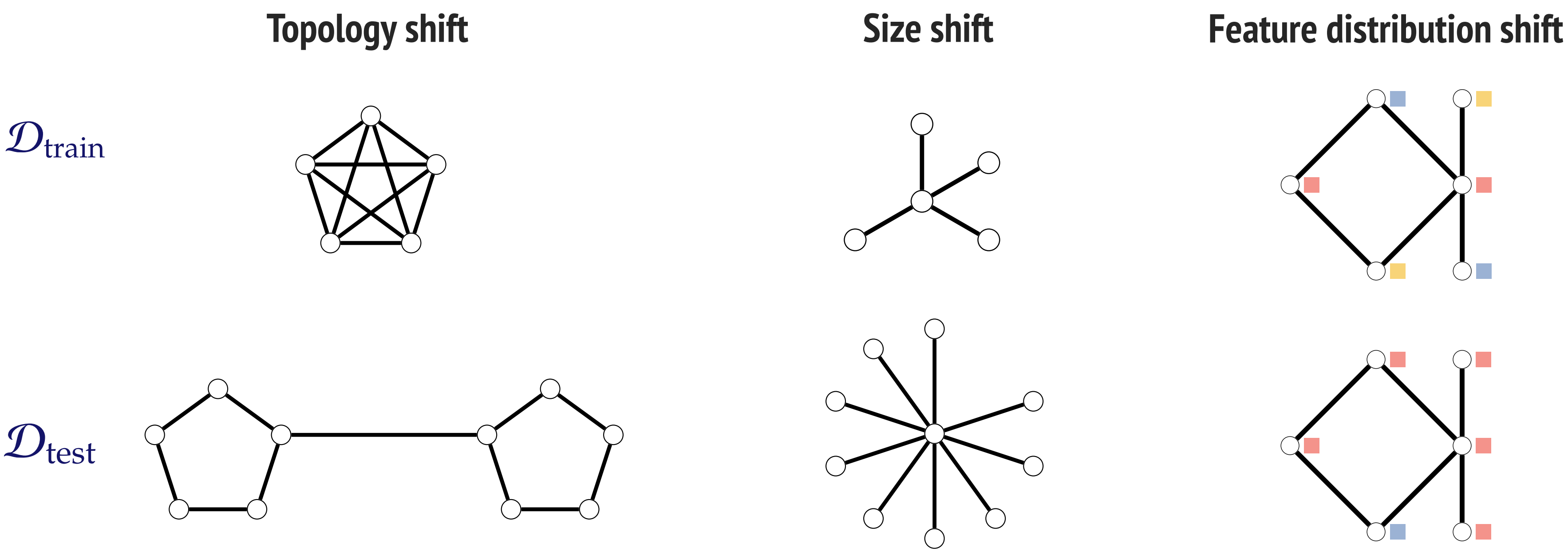
IP PARIS

NYU

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**



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

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

	Vanilla training		Training with augmentation
Training set	$\mathcal{D}_{\text{train}} = \{(\mathcal{G}_n, y_n)\}_{n=1}^{N}$	$\longrightarrow$	$\widetilde{\mathcal{D}}_{\text{train}} = \mathcal{D}_{\text{train}} \cup \{(\tilde{\mathcal{G}}_n^m, \tilde{y}_n^m)\}_{n=1..N}^{m=1..M}$
Empirical risk	$\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N} \underbrace{\ell\big(\mathcal{G}_n, \theta\big)}_{\text{cross-entropy}}$	$\longrightarrow$	$\mathcal{L}_{\text{aug}}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{M}\sum_{m=1}^{M}\ell\big(\tilde{\mathcal{G}}_n^m, \theta\big)$
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)

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

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

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

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

generalization error

Rademacher complexity
capacity of the GNN to fit random noise

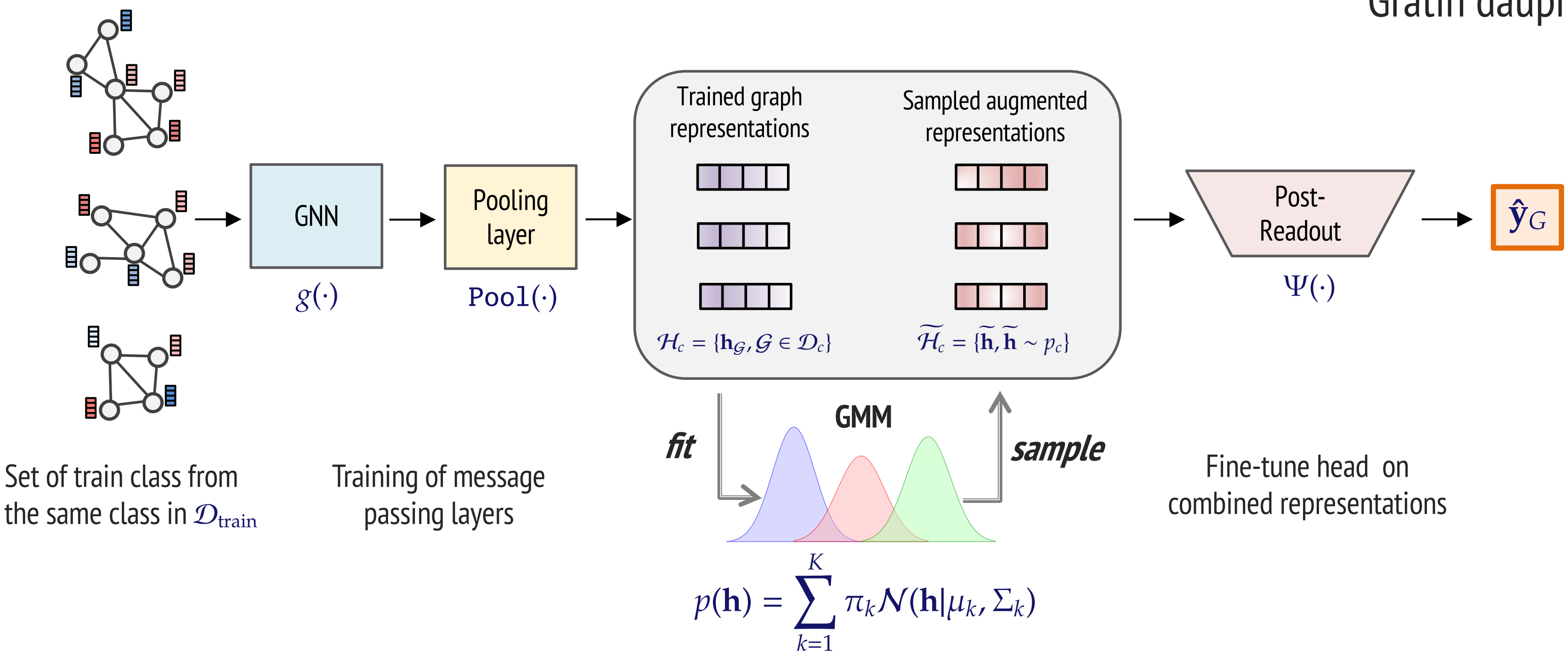
augmentation error
distance between the original graph and the augmented samples

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

If augmented graphs are too far from originals, the bound becomes large

[Shalev-Shwartz and Ben-David, Understanding Machine Learning '14]

GRATIN: GMM-based Augmentation



1. Train GNN $f(\cdot, \theta) = \Psi \circ \texttt{Pool} \circ g$
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 $\mathcal{H} \cup \widetilde{\mathcal{H}}$

GRATIN: Experimental Results

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

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

[Rong et al., ICLR ‘20], [Yoo et al., WWW ‘22], [[Han et al., ICML ‘22], [Zhao et al., KDD ‘24]

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

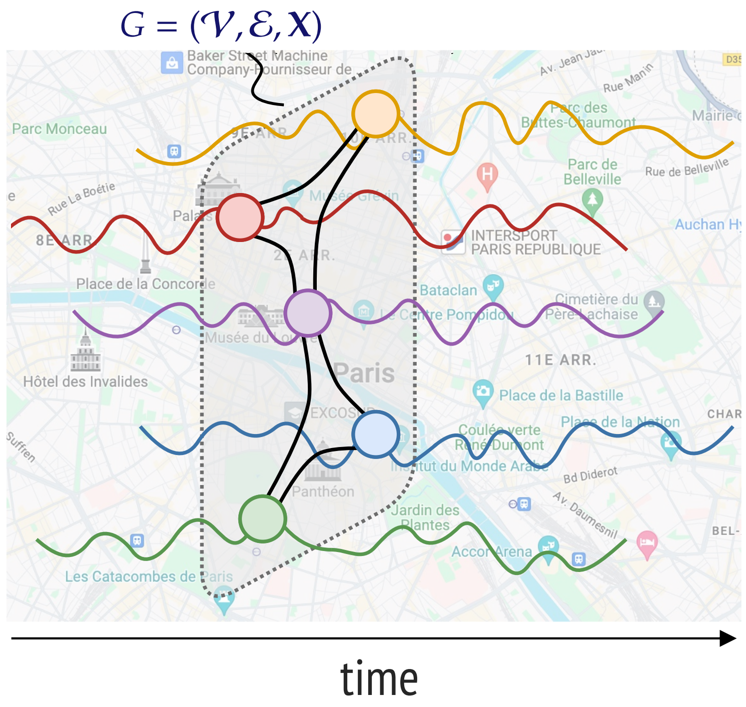
Beware of GNN Evaluation and Benchmarking

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 Siraudin [5] **Viktor Zaverkin** [6] **Michael M. Bronstein** [3] **Mathias Niepert** [7] **Bryan Perozzi** [8]
Mikhail Galkin [8] **Christopher Morris** [5]

[Bechler-Speicher et al., ICML '25]

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]

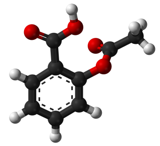
Gegenbauer Graph Neural Networks for Time-varying Signal Reconstruction

Jhon A. Castro-Correa, Jhony H. Giraldo, Mohsen Badiey, Fragkiskos D. Malliaros

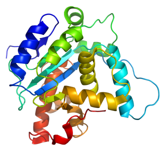
[Castro-Correa et al., IEEE Trans. Neural Netw. Learn. Syst. '24]

Geometric Graph Neural Networks (GNNs)

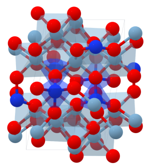
for 3D atomic systems



Small molecules



Biomolecules

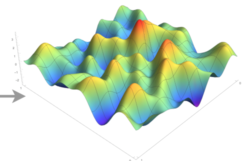


Materials

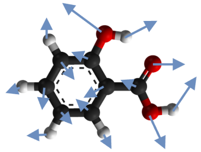
Accelerate scientific discovery with **Geometric GNNs**

Challenges

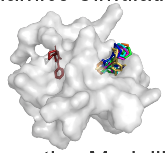
- Preserve symmetries and physical constraints
- Scalability



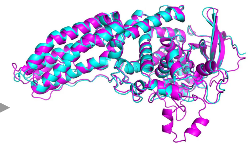
Property Prediction



Dynamics Simulation



Generative Modelling



Structure Prediction

FAENet: Frame Averaging Equivariant GNN for Materials Modeling

Alexandre Duval [*12] Victor Schmidt [*2] Alex Hernandez Garcia [2] Santiago Miret [3] Fragkiskos D. Malliaros [1] Yoshua Bengio [24] David Rolnick [25]

[Duval et al., ICML '23]

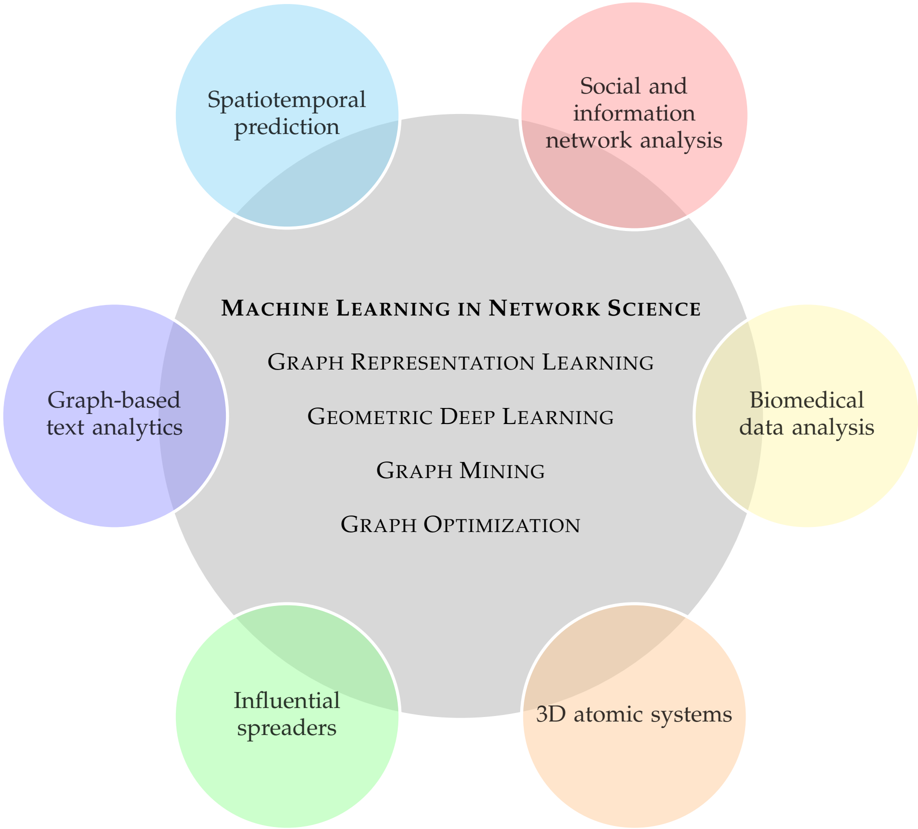
A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems

Alexandre Duval[*,1,2] Simon V. Mathis[*,3] Chaitanya K. Joshi[*,3] Victor Schmidt[*,1,4] Santiago Miret[5] Fragkiskos D. Malliaros[2] Taco Cohen[6] Pietro Liò[3] Yoshua Bengio[1,4] Michael Bronstein[7]

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

[Duval et al., arXiv '24]

Thank You!



Web: http://fragkiskos.me

Email: fragkiskos.malliaros@centralesupelec.fr

Supported in part by:

anr agence nationale de la recherche

ifp Energies nouvelles

Nokia Bell Labs

Labex DigiCosme

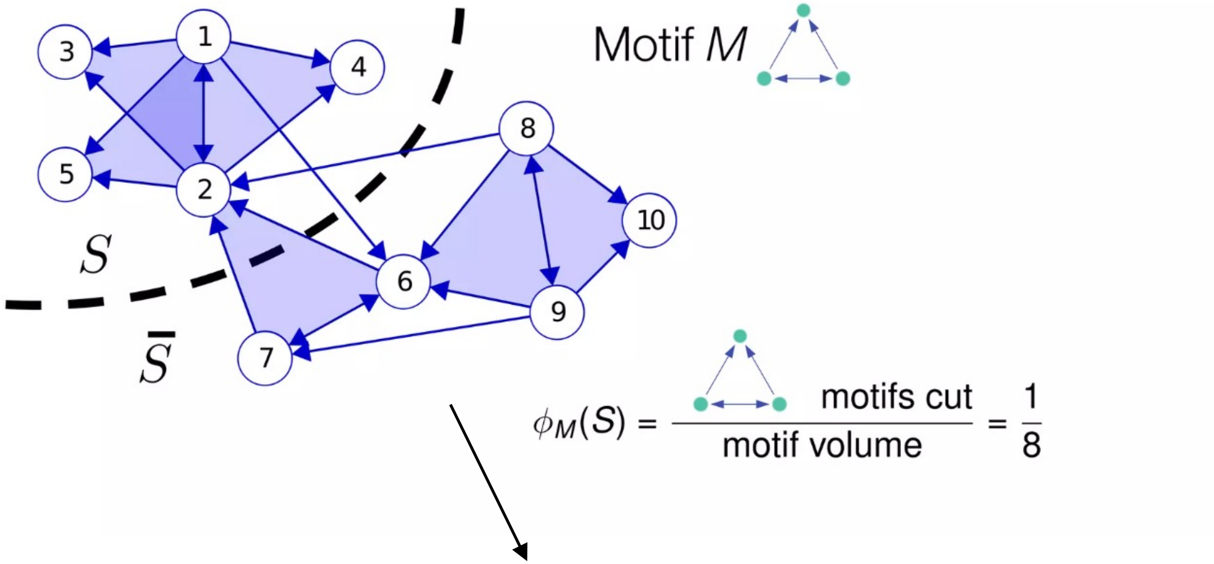
Institut DATAIA Science des données, Intelligence & Société

ENTALPIC

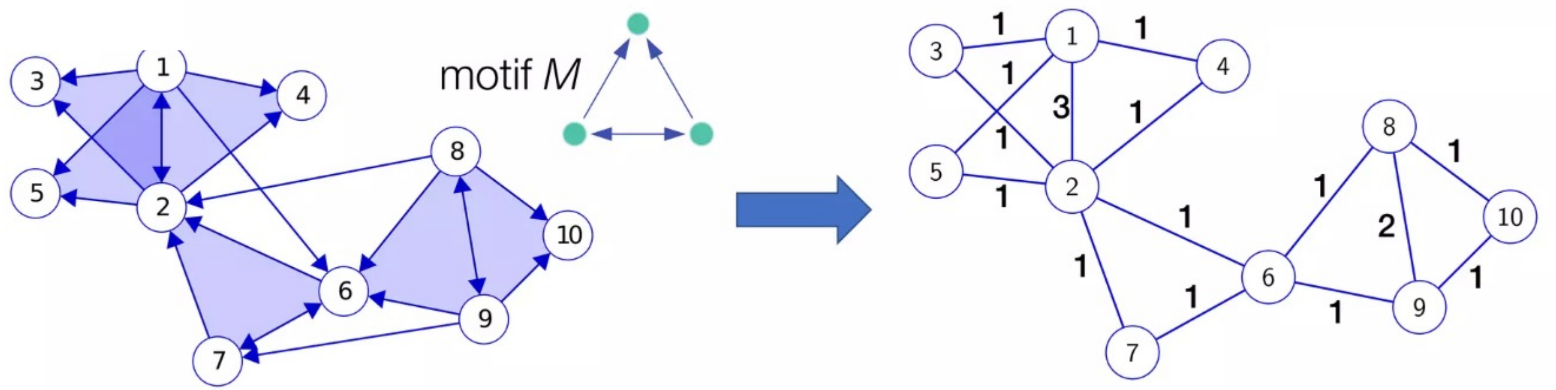
Backup Slides

Motif Spectral Clustering – Reformulation

Motif-based conductance



Graph G **Weighted motif graph A_M**



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

[Benson., AMS Spring Western Sectional '17]