

# A Simple Embedding For Classifying Networks With A Few Graphlets

Luce le Gorrec, Philip A. Knight

Department of Mathematics and Statistics

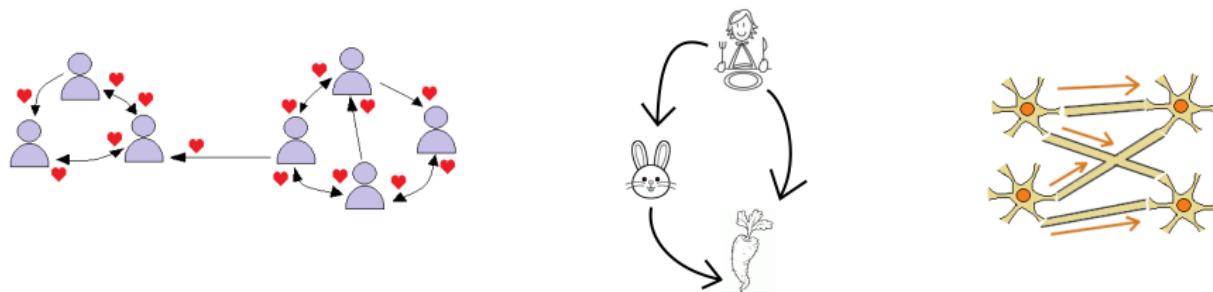
University of Strathclyde, Glasgow



Royal Academy  
of Engineering

# Graph Embedding - Principles

**Graphs** (networks): A powerful tool to represent interacting data.  
Social networks, biological networks, world wide web, etc.



**Embedding graphs** in some linear space to use Machine Learning.  
(Representing graphs as vectors).

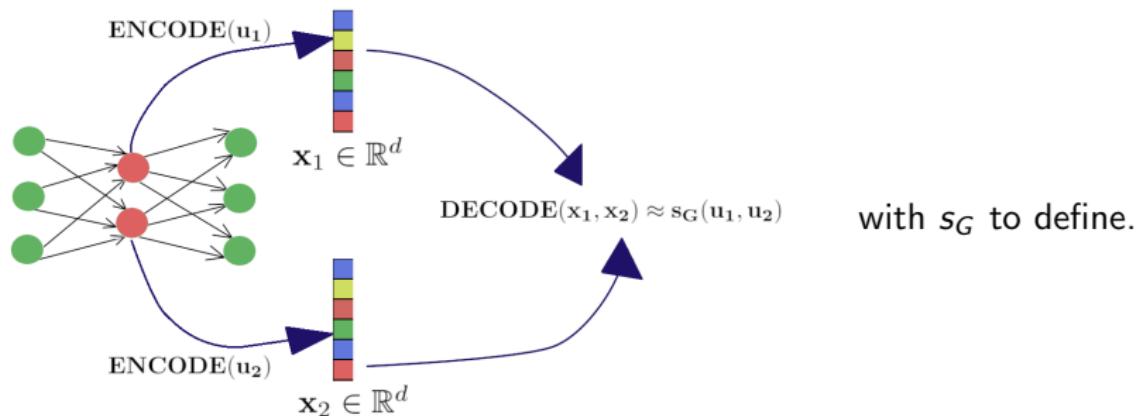
**What for:** detecting malware, predicting the function of a protein, the toxicity of chemical molecules, etc.

⇒ Classification/clustering purposes.

# Graph Embedding - Existing Methods

- Node embedding [1]
- Graph kernels [2]
- Neural networks [3]

✗ Suboptimal as not directly built for networks [7].



with  $s_G$  to define.

$$\implies \text{Emb}(G) = \sigma(x_1, \dots, x_n) \quad \text{e.g.} = \sum_{i=1}^n x_i$$



Royal Academy  
of Engineering

# Graph Embedding - Existing Methods

- Node embedding [1]
  - **Graph kernels [2]**
  - Neural networks [3]
- $\times$  Handcrafted features.  
 $\times$  Embedding comes as a byproduct.

$$k(G, G') = \text{similarity between graphs } G \text{ and } G'$$

$k$  may rely on random walk, Weisfeiler–Lehman algorithm, etc.

$\exists$  Hilbert space  $\mathcal{H}$  with hermitian product  $\langle \cdot, \cdot \rangle$ , a map  $\Phi : \mathcal{G} \mapsto \mathcal{H}$ :

$$\langle \Phi(G), \Phi(G') \rangle = k(G, G')$$

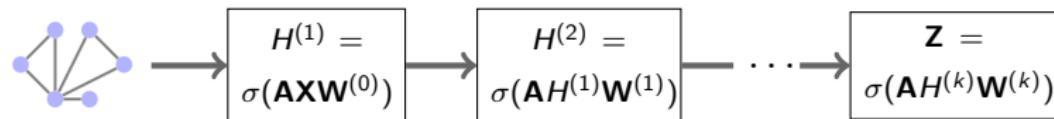
$$\implies \text{Emb}(G) = \Phi(G).$$



Royal Academy  
of Engineering

# Graph Embedding - Existing Methods

- Node embedding [1]
  - Graph kernels [2]
  - **Neural networks [3]**
- ✗ Requires detailed annotation.  
✗ Hard to handle.

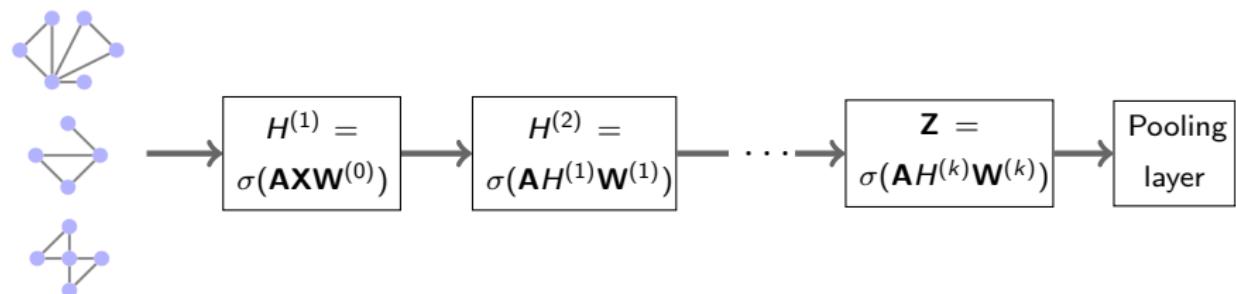


A adj. matrix  
X node features

# Graph Embedding - Existing Methods

- Node embedding [1]
- Graph kernels [2]
- **Neural networks [3]**

✗ Requires detailed annotation.  
✗ Hard to handle.



$$\mathbf{A} = \text{diag}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$$
$$\mathbf{X} = \text{diag}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$$

# Graph Embedding - Existing Methods

- Node embedding [1]
- Graph kernels [2]
- Neural networks [3]

⇒ We propose a simple approach, based on graphlet counting and PCA.



# Our Database

Our database: large, directed networks, divided into 4 classes:

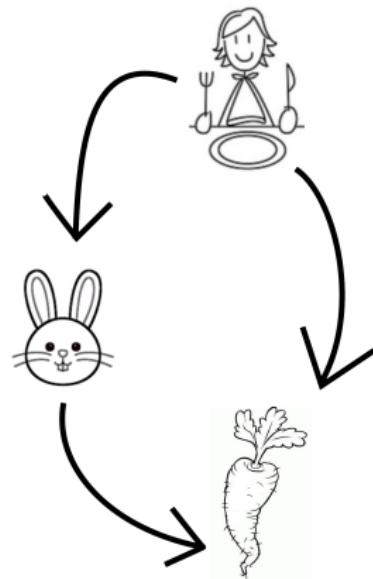
- Foodwebs • Electronic circuits • Discourse structures • Social networks



# Our Database

Our database: large, directed networks, divided into 4 classes:

- **Foodwebs**
- Electronic circuits
- Discourse structures
- Social networks



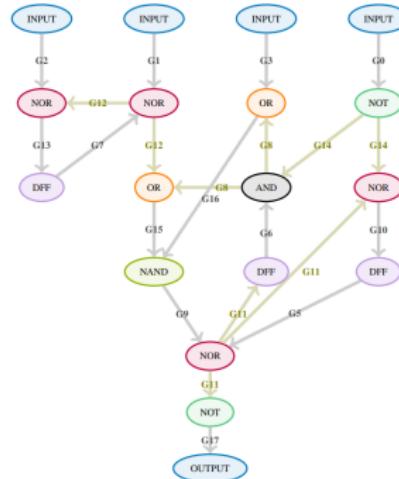
Royal Academy  
of Engineering

# Our Database

Our database: large, directed networks, divided into 4 classes:

- Foodwebs • **Electronic circuits** • Discourse structures • Social networks

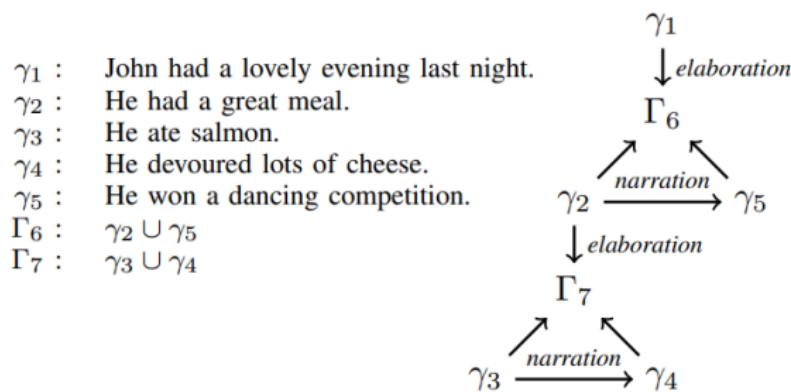
*I1:* # 4 inputs  
*I2:* # 1 outputs  
*I3:* # 3 D-type flip-flops  
*I4:* # 3 inverters  
*I5:* # 8 gates(1 ANDs +1 NANDs +2 ORs +4 NORs)  
*I6:* INPUT(G0)  
*I7:* INPUT(G1)  
*I8:* INPUT(G2)  
*I9:* INPUT(G3)  
*I10:* OUTPUT(G17)  
*I11:* G5 = DFF(G10)  
*I12:* G6 = DFF(G11)  
*I13:* G7 = DFF(G13)  
*I14:* G14 = NOT(G0)  
*I15:* G17 = NOT(G11)  
*I16:* G8 = AND(G14,G6)  
*I17:* G15 = OR(G12,G8)  
*I18:* G16 = OR(G3,G8)  
*I19:* G9 = NAND(G15,G16)  
*I20:* G10 = NOR(G14,G11)  
*I21:* G11 = NOR(G5,G9)  
*I22:* G12 = NOR(G1,G7)  
*I23:* G13 = NOR(G2,G12)



# Our Database

Our database: large, directed networks, divided into 4 classes:

- Foodwebs
- Electronic circuits
- **Discourse structures**
- Social networks



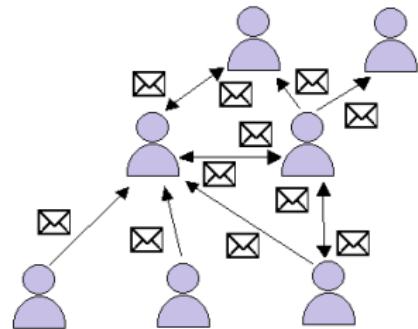
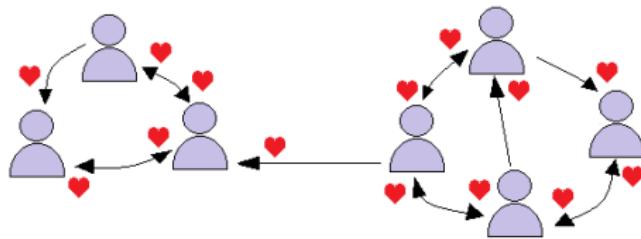
From the STAC dataset at  
[www.irit.fr/STAC/corpus.html](http://www.irit.fr/STAC/corpus.html)



# Our Database

Our database: large, directed networks, divided into 4 classes:

- Foodwebs
- Electronic circuits
- Discourse structures
- **Social networks**



Royal Academy  
of Engineering

# Our Database

Our database: large, directed networks, divided into 4 classes:

- Foodwebs
- Electronic circuits
- Discourse structures
- Social networks

**Table:** Number of nodes ( $n$ ) and edges ( $m$ ) of the networks.

	Food Webs 70 networks		Elec. Circ. 52 networks		Disc. Struct. 195 networks		Soc. Net. 81 networks	
	$n$	$m$	$n$	$m$	$n$	$m$	$n$	$m$
min	51	113	54	71	50	53	51	114
max	214	5,643	24097	52344	237	285	82168	870161
mean	104.4	1023	3358.9	6006.7	75.92	94.46	6430	48371

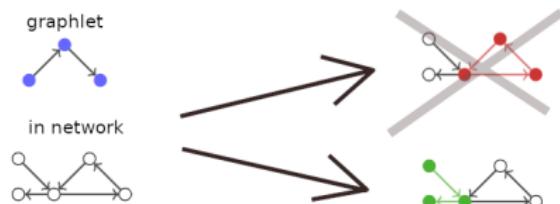
⇒ Curated database available at  
[www.github.com/luleg/DiscriminantMotifs](https://www.github.com/luleg/DiscriminantMotifs).



# Our Method - Global Ideas

Two key ideas behind our method:

- Use **3- and 4-node graphlets** to represent networks (212 connected 3/4-node graphlets).



# Our Method - Global Ideas

Two key ideas behind our method:

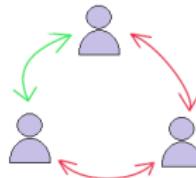
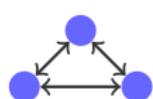
- Use **3- and 4-node graphlets** to represent networks (212 connected 3/4-node graphlets).

⇒ Capture structural topology of networks.

⇒ Networks from a same field share common motifs (statistical significant graphlets [4]).

3-clique

in friendship networks:



Royal Academy  
of Engineering

# Our Method - Global Ideas

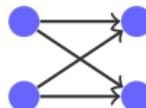
Two key ideas behind our method:

- Use **3- and 4-node graphlets** to represent networks (212 connected 3/4-node graphlets).

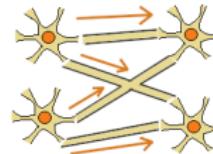
⇒ Capture structural topology of networks.

⇒ Networks from a same field share common motifs (statistical significant graphlets [4]).

bifan



in neuronal networks:

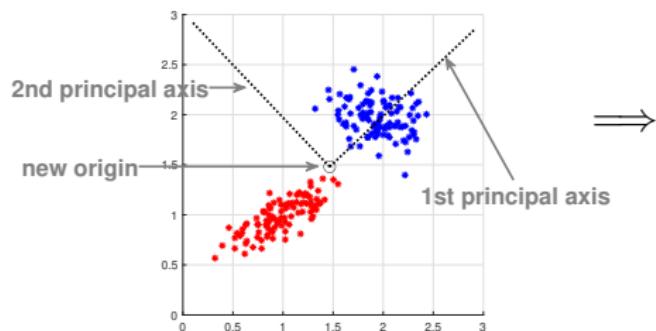


Royal Academy  
of Engineering

# Our Method - Global Ideas

Two key ideas behind our method:

- Use **3- and 4-node graphlets** to represent networks (212 connected 3/4-node graphlets).
- Use **PCA** to reduce dimensions and discriminate among networks.



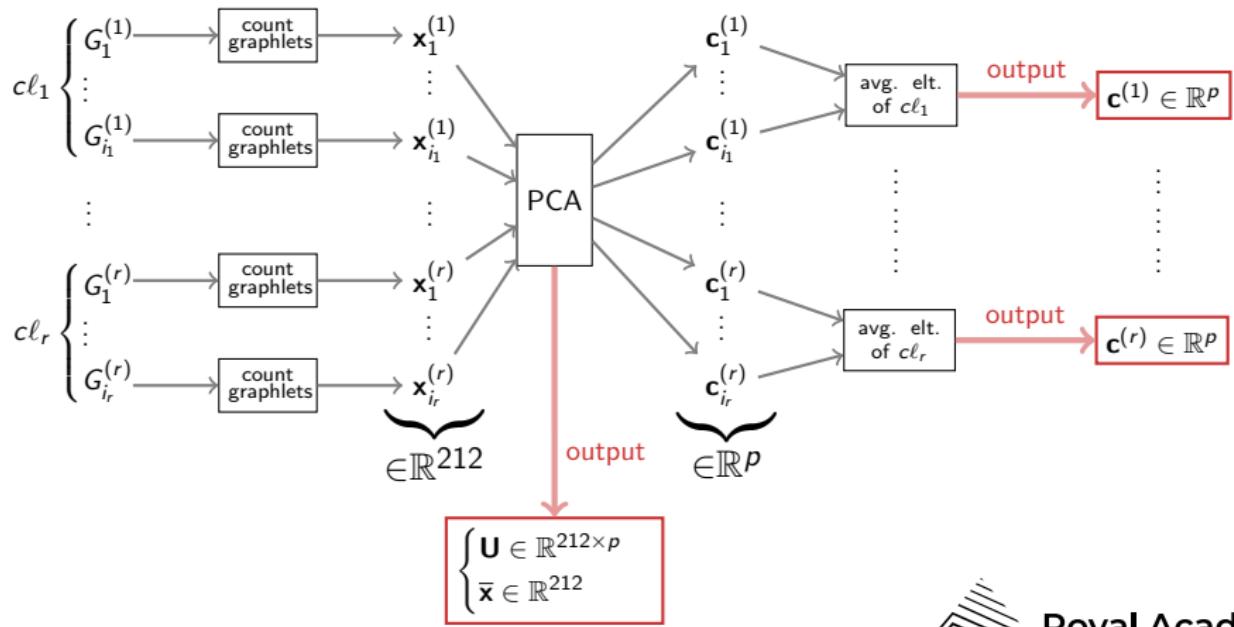
provides a new orthonormal basis, with axes sorted by order of maximal variance.



Royal Academy  
of Engineering

# Our Method - Totally Supervised Variant

Training stage: labels =  $\{cl_1, \dots, cl_r\}$ , components kept:  $p \ll 212$

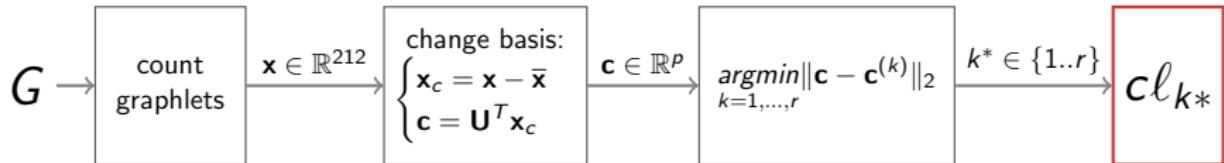


# Our Method - Totally Supervised Variant

Test stage: New graph  $G$ , which label among  $\{cl_1, \dots, cl_r\}$ ?

From the training stage:

$$\begin{cases} (\bar{\mathbf{x}} \in \mathbb{R}^{212}, \mathbf{U} \in \mathbb{R}^{212 \times p}) & \text{new reduced basis,} \\ \{(\mathbf{c}^{(1)}, cl_1), \dots, (\mathbf{c}^{(r)}, cl_r)\} & \text{class representatives.} \end{cases}$$



⇒ Methodology is similar to eigenfaces [5].

# Our Method - Totally Supervised Variant - Experiments

Comparison with two other embedding algorithms:

- gl2vec [6]
- graph2vec [7]

# Our Method - Totally Supervised Variant - Experiments

Comparison with two other embedding algorithms:

- gl2vec [6]
  - $N_{ob}(i)$ : Frequency of the  $i$ -th 3-node graphlet in the network.
  - $\overline{N_{rand}}(i)$ : Mean freq of the  $i$ -th 3-node graphlet in random graphs.
- $SRP_i = \frac{N_{ob}(i) - \overline{N_{rand}}(i)}{N_{ob}(i) + \overline{N_{rand}}(i) + \varepsilon}, \forall i.$
- graph2vec [7]

# Our Method - Totally Supervised Variant - Experiments

Comparison with two other embedding algorithms:

- gl2vec [6]
- graph2vec [7] (inspired by doc2vec). A set of graphs  $\mathcal{G}$ 
  - Nodes labelled from a common set of labels (in-/out-degrees  $\in \mathbb{N} \times \mathbb{N}$ )
  - Rooted subgraphs built by concatenating node labels with labels from their up-to- $d$ -hop neighbours.
  - Maximising the probability to get a subgraph in a graph.
  - Our setting: node labels ( $deg_{in}, deg_{out}$ ), depth of neighbourhood  $d = 1$ , embedding size  $emb = 64$ .



# Our Method - Totally Supervised Variant - Experiments

	(×10)	Precision		Recall		F1-score	
		mean	std	mean	std	mean	std
our method	<i>fw</i>	9.50	2.6	<b>10</b>	0.3	<b>9.74</b>	1.4
	<i>elec</i>	10.0	0.3	9.00	9.0	9.45	5.0
	<i>disc</i>	9.88	0.8	9.95	0.3	9.91	0.4
	<i>soc</i>	<b>10</b>	0.2	9.63	2.1	<b>9.81</b>	1.1
gl2vec	<i>fw</i>	<b>9.76</b>	2.5	8.65	5.3	9.16	3.3
	<i>elec</i>	9.26	8.5	6.84	11.2	7.79	8.2
	<i>disc</i>	9.89	0.7	<b>10</b>	0	9.94	0.4
	<i>soc</i>	8.98	3.2	<b>10</b>	0	9.46	1.8
graph 2vec	<i>fw</i>	9.51	3.6	9.73	3.1	9.61	2.6
	<i>elec</i>	<b>10</b>	0	<b>9.80</b>	3.6	<b>9.90</b>	1.9
	<i>disc</i>	<b>9.90</b>	0.7	<b>10</b>	0	<b>9.95</b>	0.3
	<i>soc</i>	9.80	2.3	9.30	3.9	9.54	2.4

Our method is:

- ⇒ Always better than gl2vec.
- ⇒ Better than graph2vec for social networks and food webs.

Note graph2vec: better when there are no bi-directed edges.



Royal Academy  
of Engineering

## Our Method - Less Supervised Variant

Two drawbacks from the previous method:

- Counting all 3/4-node graphlets is expensive.
- Matrix  $\mathbf{U}$  strongly depends on the training set.

⇒ Using the training set to select the most promising graphlets.

The expression level of the  $i$ th canonical axis in the  $p$  first principal axes is

$$\gamma(i) = \frac{1}{\sum_{s=1}^p \lambda_s} \sum_{k=1}^p \lambda_k \mathbf{u}_k(i)^2, \quad \lambda_k \text{ level of variance}$$

with  $\mathbf{u}_k(i)$  the  $i$ th coordinate of  $k$ th principal axis in the graphlet basis.

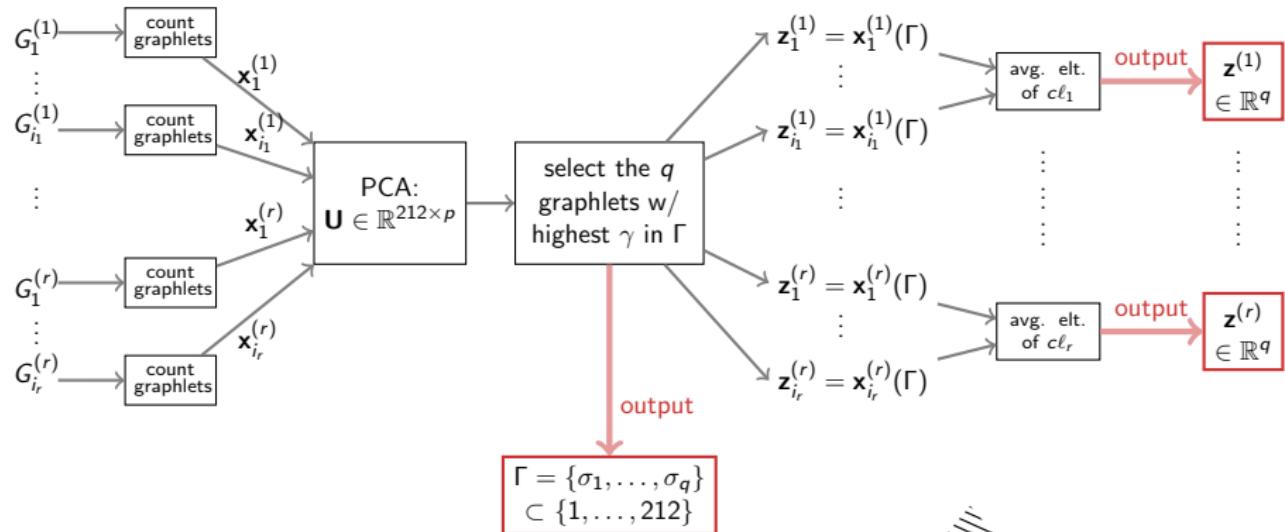
⇒  $\gamma$  gives a ranking of graphlets significance.



Royal Academy  
of Engineering

# Our Method - Less Supervised Variant

Training stage: Labels =  $\{cl_1, \dots, cl_r\}$ , components kept:  $p \ll 212$ , graphlets kept:  $q \ll 212$ .

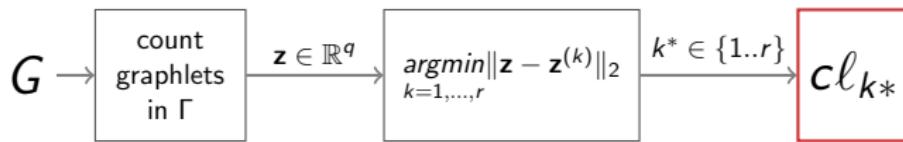


# Our Method - Less Supervised Variant

Test stage: New graph  $G$ , which label among  $\{cl_1, \dots, cl_r\}$ ?

From the training stage:

$$\begin{cases} \Gamma \subset \{1, \dots, 212\}, \quad |\Gamma| = q & \text{indices of the selected graphlets,} \\ \{(\mathbf{z}^{(1)}, cl_1), \dots, (\mathbf{z}^{(r)}, cl_r)\} & \text{class representatives, with } \mathbf{z}^{(k)} \in \mathbb{R}^q \end{cases}$$



# Our Method - Less Supervised Variant - Experiments

Comparisons with our totally supervised version.

	fw	Precision		Recall		F1-score	
		mean	std	mean	std	mean	std
our new method	fw	<b>9.50</b>	2.7	9.78	2.6	9.64	1.9
	elec	9.44	6.3	8.86	8.4	9.11	5.8
	disc	9.86	0.8	<b>9.95</b>	0.3	9.90	0.4
	soc	<b>10</b>	0	9.61	2.5	9.80	1.3
our old method	fw	<b>9.50</b>	2.6	<b>10</b>	0.3	<b>9.74</b>	1.4
	elec	<b>10.0</b>	0.3	<b>9.00</b>	9.0	<b>9.45</b>	5.0
	disc	<b>9.88</b>	0.8	<b>9.95</b>	0.3	<b>9.91</b>	0.4
	soc	<b>10</b>	0.2	<b>9.63</b>	2.1	<b>9.81</b>	1.1
gl2vec	fw	<b>9.76</b>	2.5	8.65	5.3	9.16	3.3
	elec	9.26	8.5	6.84	11.2	7.79	8.2
	disc	9.89	0.7	<b>10</b>	0	9.94	0.4
	soc	8.98	3.2	<b>10</b>	0	9.46	1.8
graph 2vec	fw	9.51	3.6	9.73	3.1	9.61	2.6
	elec	<b>10</b>	0	<b>9.80</b>	3.6	<b>9.90</b>	1.9
	disc	<b>9.90</b>	0.7	<b>10</b>	0	<b>9.95</b>	0.3
	soc	9.80	2.3	9.30	3.9	9.54	2.4
RF feat. select.	fw	9.17	4.0	8.74	6.7	8.93	4.0
	elec	4.30	12.3	7.77	18.2	5.46	12.8
	disc	9.70	2.2	9.77	3.7	9.73	2.4
	soc	8.16	9.5	6.08	13.0	6.91	11.0

- Comes at the cost of a slight decrease in quality.



# Our Method - Less Supervised Variant - Experiments

Comparisons with gl2vec and graph2vec.

	fw	Precision		Recall		F1-score	
		mean	std	mean	std	mean	std
our new method	fw	9.50	2.7	<b>9.78</b>	2.6	<b>9.64</b>	1.9
	elec	9.44	6.3	8.86	8.4	9.11	5.8
	disc	9.86	0.8	9.95	0.3	9.90	0.4
	soc	<b>10</b>	0	9.61	2.5	<b>9.80</b>	1.3
our old method	fw	9.50	2.6	10	0.3	9.74	1.4
	elec	10.0	0.3	9.00	9.0	9.45	5.0
	disc	9.88	0.8	9.95	0.3	9.91	0.4
	soc	10	0.2	9.63	2.1	9.81	1.1
gl2vec	fw	<b>9.76</b>	2.5	8.65	5.3	9.16	3.3
	elec	9.26	8.5	6.84	11.2	7.79	8.2
	disc	9.89	0.7	<b>10</b>	0	9.94	0.4
	soc	8.98	3.2	<b>10</b>	0	9.46	1.8
graph 2vec	fw	9.51	3.6	9.73	3.1	9.61	2.6
	elec	<b>10</b>	0	<b>9.80</b>	3.6	<b>9.90</b>	1.9
	disc	<b>9.90</b>	0.7	<b>10</b>	0	<b>9.95</b>	0.3
	soc	9.80	2.3	9.30	3.9	9.54	2.4
RF feat. select.	fw	9.17	4.0	8.74	6.7	8.93	4.0
	elec	4.30	12.3	7.77	18.2	5.46	12.8
	disc	9.70	2.2	9.77	3.7	9.73	2.4
	soc	8.16	9.5	6.08	13.0	6.91	11.0

- Comparison with graph2vec and gl2vec similar to before.

# Our Method - Less Supervised Variant - Experiments

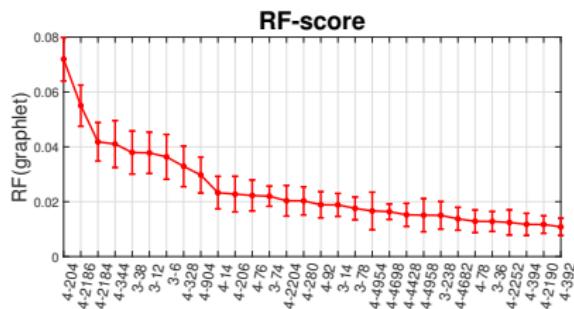
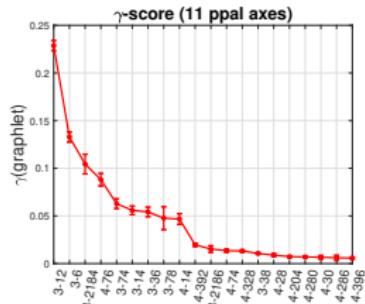
Comparisons with features selected via Gini scores from random forest [8].

	fw	Precision		Recall		F1-score	
		mean	std	mean	std	mean	std
our new method	fw	<b>9.50</b>	2.7	<b>9.78</b>	2.6	<b>9.64</b>	1.9
	elec	<b>9.44</b>	6.3	<b>8.86</b>	8.4	<b>9.11</b>	5.8
	disc	<b>9.86</b>	0.8	<b>9.95</b>	0.3	<b>9.90</b>	0.4
	soc	<b>10</b>	0	<b>9.61</b>	2.5	<b>9.80</b>	1.3
our old method	fw	9.50	2.6	10	0.3	9.74	1.4
	elec	10.0	0.3	9.00	9.0	9.45	5.0
	disc	9.88	0.8	9.95	0.3	9.91	0.4
	soc	10	0.2	9.63	2.1	9.81	1.1
gl2vec	fw	<b>9.76</b>	2.5	8.65	5.3	9.16	3.3
	elec	9.26	8.5	6.84	11.2	7.79	8.2
	disc	9.89	0.7	<b>10</b>	0	9.94	0.4
	soc	8.98	3.2	<b>10</b>	0	9.46	1.8
graph 2vec	fw	9.51	3.6	9.73	3.1	9.61	2.6
	elec	<b>10</b>	0	<b>9.80</b>	3.6	<b>9.90</b>	1.9
	disc	<b>9.90</b>	0.7	<b>10</b>	0	<b>9.95</b>	0.3
	soc	9.80	2.3	9.30	3.9	9.54	2.4
RF feat. select.	fw	9.17	4.0	<b>8.74</b>	6.7	<b>8.93</b>	4.0
	elec	4.30	12.3	7.77	18.2	5.46	12.8
	disc	9.70	2.2	9.77	3.7	9.73	2.4
	soc	8.16	9.5	6.08	13.0	6.91	11.0

- $\gamma$  measure provides much more discriminatory graphlets than Gini score.

# Our Method - Less Supervised Variant - Experiments

Comparisons with features selected via Gini scores from random forest [8].

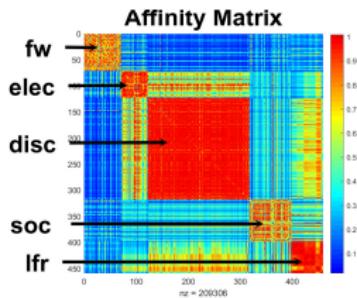


# Towards Unsupervised Methods

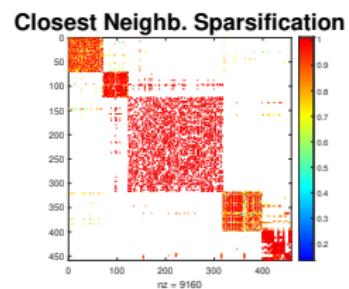
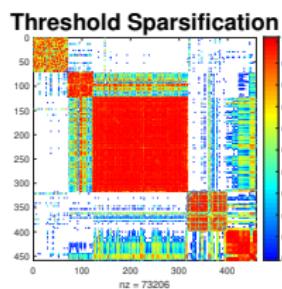
May these graphlets help to detect other kind of networks?

- Adding 60 LFR-generated networks,
- Using the algorithm from [9] to partition the dataset.

Affinity matrix



2 sparsifications



$$A(i,j) = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{2\sigma^2}\right),$$

with  $\mathbf{z}_k \in \mathbb{R}^q$  the embeddings.



Royal Academy  
of Engineering

# Towards Unsupervised Methods

	Results on thresh. sparsification					Results on closest neighbour sparsification											
	Uncovered Clusters					F1	Uncovered Clusters					F1					
	1	2	3	4	5	F1	1	2	3	4	5	6	7	8	9	10	F1
<i>fw</i>	70					9.86									70	9.86	
<i>elec</i>		6		46		9.39			4	2					46	9.39	
<i>disc</i>	1		192	2		9.77			52	26	56	59		1		1	9.80
<i>soc</i>	1	75		5		9.62	40	38					1	1		1	9.81
<i>lfr</i>			60			9.45							10	50			9.76

⇒ Extremely consistent results.

✗ But a “false” unsupervised method.

# Conclusion

- With a very simple **supervised method** based on **graphlets** and **PCA**, we are able to **accurately detect classes of networks**.
- Given a training set, we provide a **measure of graphlet significance** to find the **most discriminatory graphlets**.
- Graphlets with highest measure score seem to be able to **discriminate other kind of networks**.
- Our method can be used for **other applications**, on **undirected networks**, using **larger graphlets**.

MUTAG	our method		gl2vec		graph2vec	
mutagenic	yes	no	yes	no	yes	no
F1-score	6.49	8.88	7.23	9.21	5.49	8.70



Royal Academy  
of Engineering

# Thank you for your attention - Any questions?

Codes and dataset available on [github/luleg/DiscriminantMotifs](https://github.com/luleg/DiscriminantMotifs).

Counting graphlets: [www.ft.unicamp.br/docentes/meira/accmotifs/](http://www.ft.unicamp.br/docentes/meira/accmotifs/)

- [1] W.L. Hamilton, R. Ying, J. Leskovec, "Representation learning on graphs: Methods and applications", 2017.
- [2] S. Vishwanathan, N.N. Schraudolph, R. Kondor, K.M. Borgwardt, "Graph Kernels", 2010.
- [3] T.N. Kipf, M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", 2017.
- [4] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenstiel, M. Sheffer, U. Alon, "Superfamilies of Evolved and Designed Networks", 2004.
- [5] M. Turk, A. Pentland, "Eigenfaces for recognition", 1991
- [6] K. Tu, J. Li, D. Towsley, D. Braines, L.D. Turner, "GI2vec: Learning feature representation using graphlets for directed networks", 2019.
- [7] A. Narayanan, et al, "graph2vec: Learning distributed representations of graphs", 2017.
- [8] L. Breiman, "Random forests", 2001.
- [9] L. le Gorrec, S. Mouysset, I.S. Duff, P.A. Knight, D. Ruiz, "Uncovering hidden block structure for clustering", 2019.



Royal Academy  
of Engineering

