Feature Extraction Methods on Graphs without Learning (Graph-Level)

ACMS 80770: Deep Learning with Graphs

Instructor: Navid Shervani-Tabar

Department of Applied and Comp Math and Stats

UNIVERSITY OF NOTRE DAME

Kernel Methods

- In parametric regression and classification models, given a dataset $\{x^{(i)}, y^{(i)}\}_{i \leq n}$ of n data points $x^{(i)}$ and labels $y^{(i)}$, we learn a function $f_{\boldsymbol{w}}$ parametrized by $\boldsymbol{w}$, that maps $x^{(i)}$ to $y^{(i)}$.
- In essence, this approach compresses the training dataset $\{x^{(i)}, y^{(i)}\}_{i \leq n}$ into a parameter vector $\boldsymbol{w}$.
- For a test data x^*, we can predict its label by $y^* = f_{\boldsymbol{w}}(x^*)$.
- In a non-parametric approach, instead of learning a parametric function $f_{\boldsymbol{w}}$, we can predict $f(x^*)$ based on the similarity of x^* to the training points $\boldsymbol{X} = \{x^{(i)}\}_{i \leq n}$.
- Unlike the parametric approach, this approach requires access to the training data $\{x^{(i)}, y^{(i)}\}_{i \leq n}$ during the test time.

Kernel Methods

- For this approach, we need to find the similarity between every pair of data points x_i and x_j from the set of data points $\boldsymbol{X}$.
- This similarity is then plugged into a kernel machine to yield a prediction in the test time.
- For two data points x and x', a kernel function

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

is a map $k: \boldsymbol{X} \times \boldsymbol{X} \to \mathbb{R}^+$ that measures the similarity between x and x'.

- Here, ϕ defines a map from the space of data points $\boldsymbol{X}$ to an embedding space.

Kernel Methods

- $\phi(x)$ represents feature map of data point x.
- Two similar data points x_i and x_j in the Euclidean space should have similar representations $\phi(x_i)$ and $\phi(x_j)$ in the embedding space.
- Given kernel function $k(x_i, x_j)$ and data points $\{x_i\}_{i\leq n}$, we construct an $n\times n$ similarity matrix $\boldsymbol{K}$ named Gram matrix, such that

$$K_{ij} = k\left(x_i, x_j\right)$$

- For any set of data points $\{x_i\}_{i\leq n}$, Gram matrix is a positive semi-definite matrix.

Kernel Methods

- There are two approaches to construct kernel functions
  - Explicit kernels, where $k(x_i, x_j)$ is computed from the inner product $\langle\phi(x_i), \phi(x_j)\rangle$ given a mapping ϕ.
  - Implicit kernels, where $k(x_i, x_j)$ is directly defined as a function of x_i and x_j, corresponding to a scalar product in some unknown feature space.
- Implicit kernels are restricted in construction by that the resulting Gram matrix $\boldsymbol{K}$ should be positive semi-definite.
- Similar to Euclidean data, we can define kernels for graphs as well.

Isomorphism

- Given graphs $G = (V^G, E^G)$ and $H = (V^H, E^H)$, if we can define a bijection $\psi: V^G \to V^H$, such that

$$\forall v_i^G, v_j^G \in \mathrm{V}^G, \left(v_i^G, v_j^G\right) \in E^G \text{ iff } \left(\psi\left(v_i^G\right), \psi\left(v_j^G\right)\right) \in E^H$$

the graphs G and H are said to be isomorphic and denoted as

$$G \simeq H$$

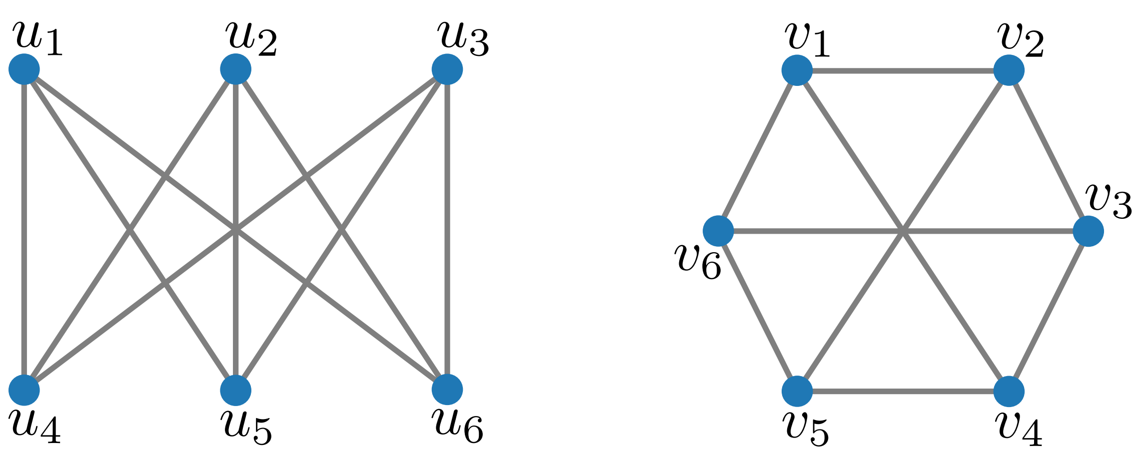
Isomorphism

- Given graphs $G = (V^G, E^G)$ and $H = (V^H, E^H)$, if we can define a bijection $\psi: V^G \to V^H$, such that

$$\forall v_i^G, v_j^G \in \mathrm{V}^G, \left(v_i^G, v_j^G\right) \in E^G \text{ iff } \left(\psi\left(v_i^G\right), \psi\left(v_j^G\right)\right) \in E^H$$

the graphs G and H are said to be isomorphic and denoted as

$$G \simeq H$$

- 

$$\psi : \begin{array}{ccc} V^G & & V^H \\ v_1 & \to & u_1 \\ v_2 & \to & u_4 \\ v_3 & \to & u_2 \\ v_4 & \to & u_5 \\ v_5 & \to & u_3 \\ v_6 & \to & u_6 \end{array}$$

Isomorphism

- Given graphs $G = (V^G, E^G)$ and $H = (V^H, E^H)$, if we can define a bijection $\psi: V^G \to V^H$, such that

$$\forall v_i^G, v_j^G \in \mathrm{V}^G, \left(v_i^G, v_j^G\right) \in E^G \text{ iff } \left(\psi\left(v_i^G\right), \psi\left(v_j^G\right)\right) \in E^H$$

  the graphs G and H are said to be isomorphic and denoted as

$$G \simeq H$$

- The mapping ψ saves adjacency.
- Isomorphism is not only hard to detect by eye, but also a challenging problem for computers as well.

Graph Kernels

- A graph kernel is a kernel k defined on a non-empty set of graphs $\mathcal{G} = \{G^{(i)}\}_{i \leq N}$ as

$$k : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}^{+}$$

- Let $\phi(G)$ be the feature map constructed from graph G.
- A graph kernel is defined as

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

- A graph kernel k quantifies the similarity of two graphs G and G'.
- A Gram matrix $\boldsymbol{K}$ for a set of n graphs $\{G^{(i)}\}_{i \leq n}$ is an $n \times n$ matrix with elements K_{ij} defined as

$$K_{ij} = k(G^{(i)}, G^{(j)})$$

Graph Kernel Designs

- Here, we only look at the explicit graph kernels and study methods to construct feature vectors $\phi(G)$ from a graph G.

- Graphs are permutation invariant, i.e., they are invariant to the ordering of the nodes.

- Therefore, the feature map ϕ should also be invariant to the permutations of the input graph.

Graph Kernel Designs

- There are several graph kernels, but they generally fall into the following categories:
  - Substructure-based kernels
  - Path-based kernels
  - Neighborhood aggregation kernels
  - Assignment-based kernels
  - Neural Network-based kernel

- We look at graph kernel designs for undirected graphs, with no self-loops.

Substructure-Based Kernels

- Substructure-Based Kernels compare two graphs in terms of the frequency of different patterns in the subgraphs of the graph.

- These methods stem from bag-of-nodes approach in natural language processing.

- Two types of substructure-based graph kernels are
  - Bag of nodes kernel
  - Graphlet kernel

Bag of Node Kernel

- **Bag of nodes graph kernel:** Compares graph-level features generated by aggregating node-level features.
- The feature descriptor for each graph is defined as the histogram of the node features in the graph.
- Bag of nodes kernel is defined as

$$k_{BN}(G, H) = \langle \phi_{BN}(G), \phi_{BN}(H) \rangle$$

- Feature map ϕ_{BN} is defined as

$$\phi_{BN}(G) = (|\sigma_1(G)|, \ldots, |\sigma_N(G)|)$$

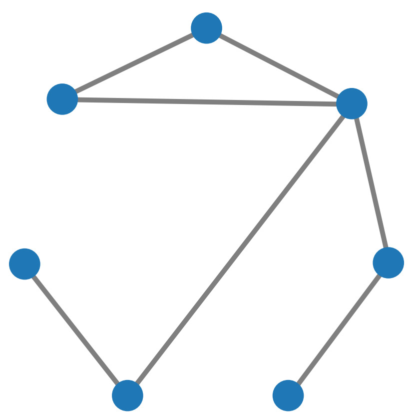
where $\sigma_n(G)$ indicates the set of nodes v_i of G with a given statistics.

Bag of Node Kernel

- **Bag of node degrees:** Characterizes a graph G based on frequency of different degree values in that graph.
- For bag of degrees, the set of nodes σ_n is defined as

$$\sigma_n(G) = \{v_i | v_i \in V^G, d_i = n\}$$

- $\phi_{BN}(G)$

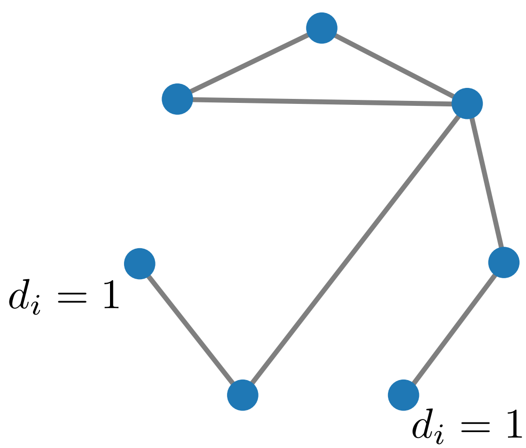


Bag of Node Kernel

- **Bag of node degrees:** Characterizes a graph G based on frequency of different degree values in that graph.
- For bag of degrees, the set of nodes σ_n is defined as

$$\sigma_n(G) = \{v_i | v_i \in V^G, d_i = n\}$$

- $\phi_{BN}(G)$



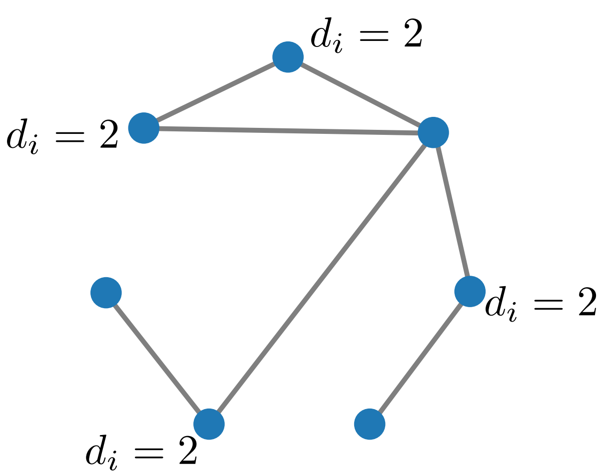
$d = 1$	$d = 2$	$d = 3$	$d = 4$
2			

Bag of Node Kernel

- **Bag of node degrees:** Characterizes a graph G based on frequency of different degree values in that graph.
- For bag of degrees, the set of nodes σ_n is defined as

$$\sigma_n(G) = \{v_i | v_i \in V^G, d_i = n\}$$

- $\phi_{BN}(G)$



$d = 1$	$d = 2$	$d = 3$	$d = 4$
2	4		

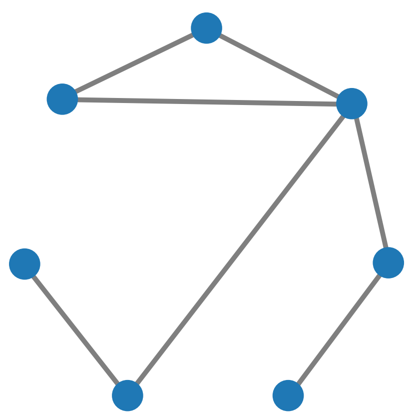
Bag of Node Kernel

- **Bag of node degrees:** Characterizes a graph G based on frequency of different degree values in that graph.

❖ For bag of degrees, the set of nodes σ_n is defined as

$$\sigma_n(G) = \{v_i | v_i \in V^G, d_i = n\}$$

- $\phi_{BN}(G)$



$d = 1$	$d = 2$	$d = 3$	$d = 4$
2	4	0	

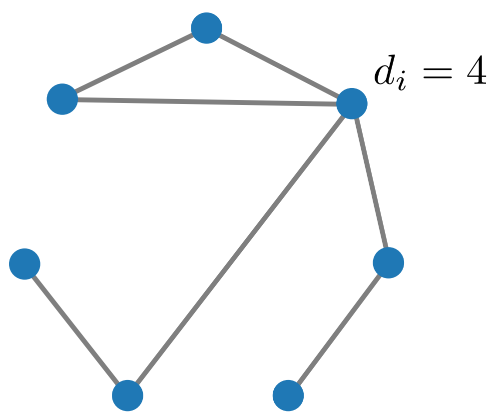
Bag of Node Kernel

- **Bag of node degrees:** Characterizes a graph G based on frequency of different degree values in that graph.

- For bag of degrees, the set of nodes σ_n is defined as

$$\sigma_n(G) = \{v_i | v_i \in V^G, d_i = n\}$$

- $\phi_{BN}(G)$



$d = 1$	$d = 2$	$d = 3$	$d = 4$
2	4	0	1

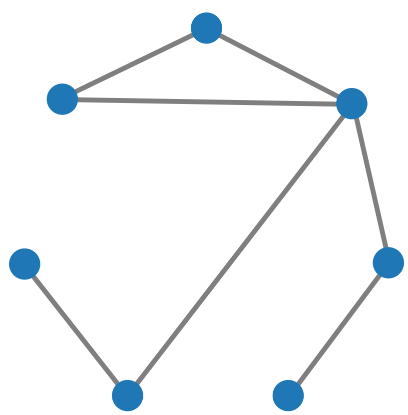
Bag of Node Kernel

- **Bag of node degrees:** Characterizes a graph G based on frequency of different degree values in that graph.

❖ For bag of degrees, the set of nodes σ_n is defined as

$$\sigma_n(G) = \{v_i | v_i \in V^G, d_i = n\}$$

- $\phi_{BN}(G) = (2, 4, 0, 1)$



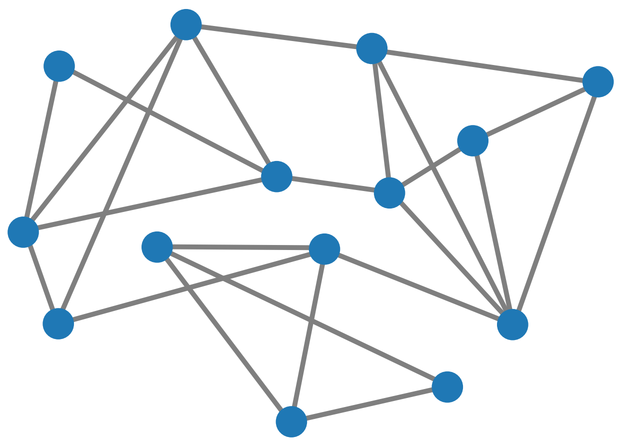
$d = 1$	$d = 2$	$d = 3$	$d = 4$
2	4	0	1

❖ Does not reflect the global structure of the graph.

Induced Subgraph

- An induced subgraph is a subgraph $G' = (V', E')$ consist of a subset of the nodes in the graph $G = (V, E)$ and all edges that connect them

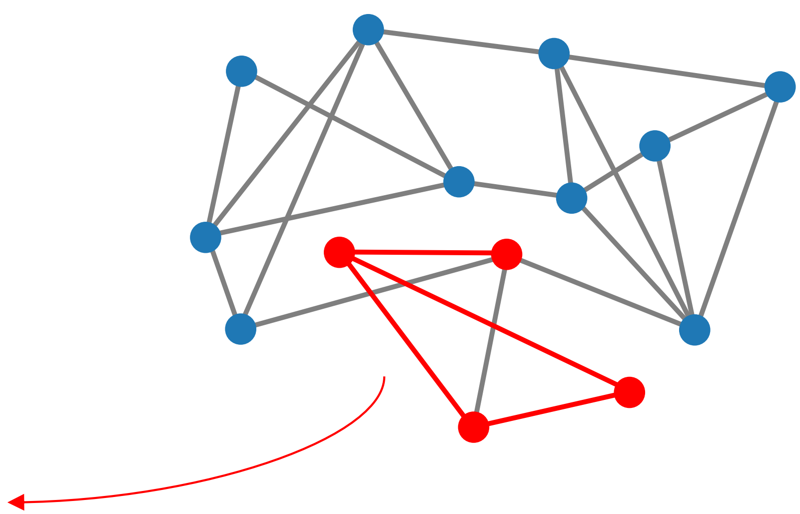
$$E' = \{(v_i, v_j) \mid v_i, v_j \in V', (v_i, v_j) \in E\}$$



Induced Subgraph

- An induced subgraph is a subgraph $G' = (V', E')$ consist of a subset of the nodes in the graph $G = (V, E)$ and all edges that connect them

$$E' = \{(v_i, v_j) \mid v_i, v_j \in V', (v_i, v_j) \in E\}$$

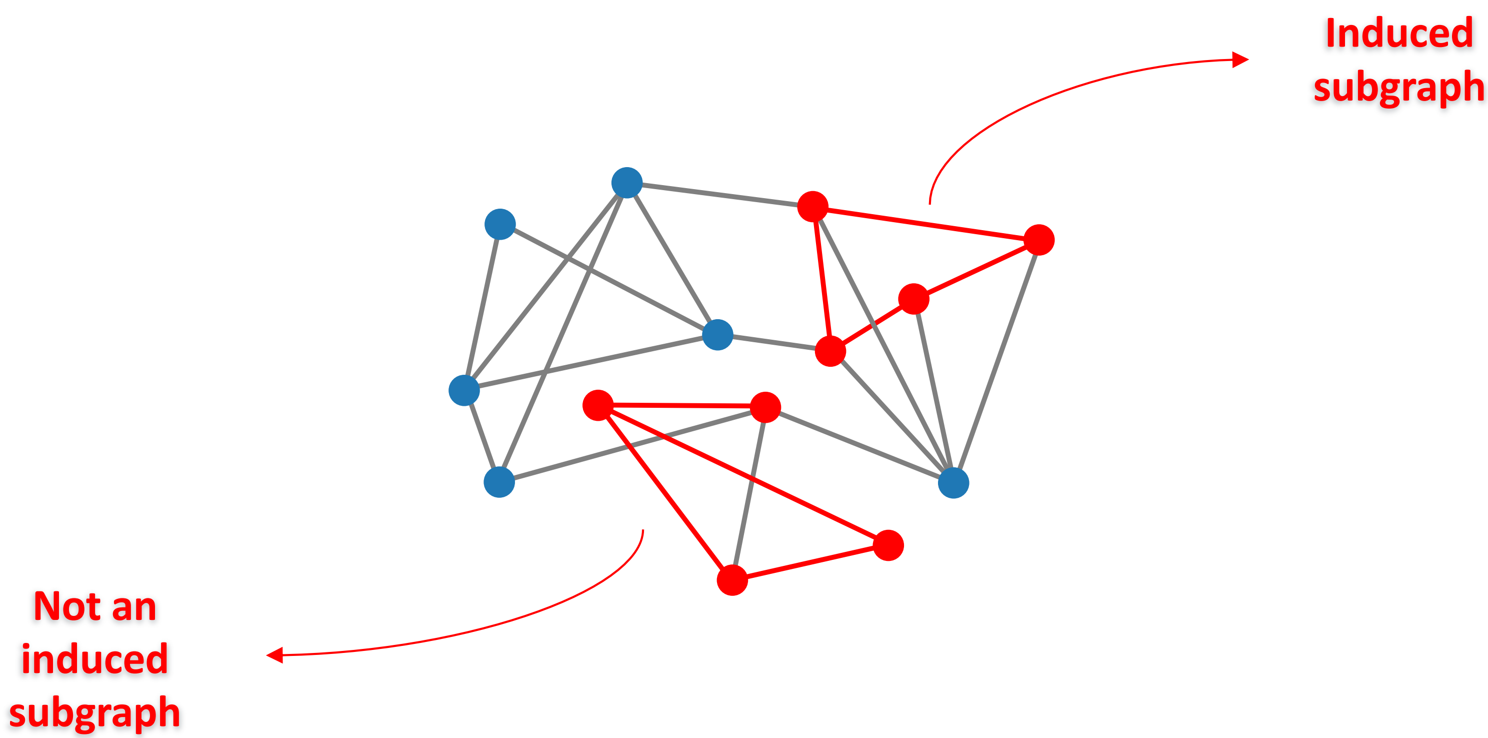


Not an induced subgraph

Induced Subgraph

- An induced subgraph is a subgraph $G' = (V', E')$ consist of a subset of the nodes in the graph $G = (V, E)$ and all edges that connect them

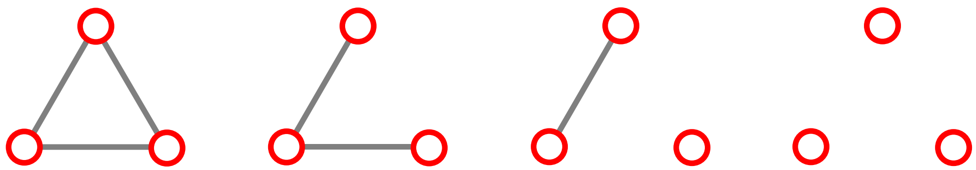
$$E' = \{(v_i, v_j) \mid v_i, v_j \in V', (v_i, v_j) \in E\}$$



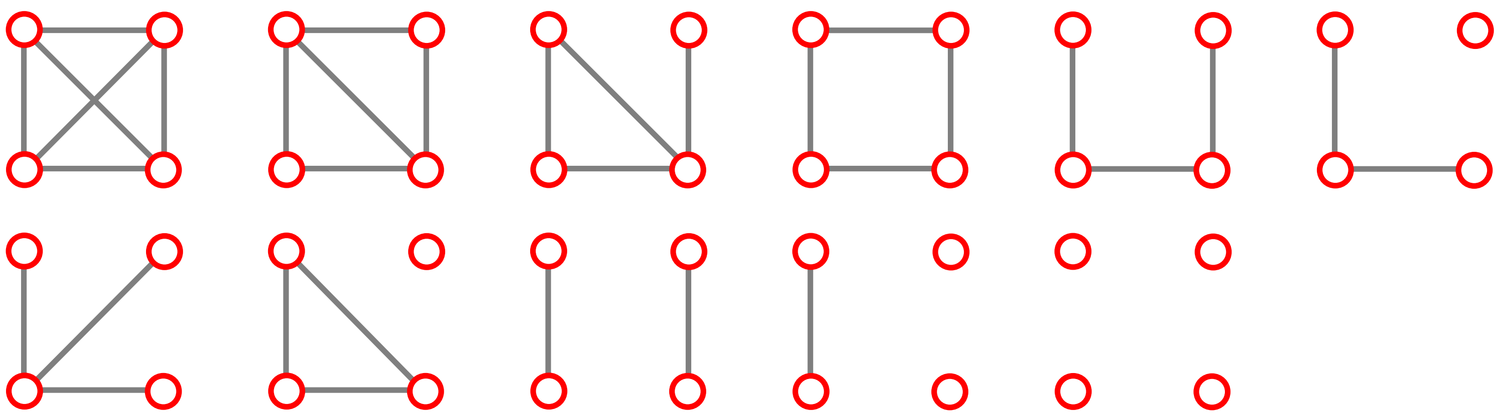
Graphlets

- Graphlets are non-isomorphic induced subgraph patterns of a particular size k with all different combination of edge connections.

- $k = 3$ graphlets



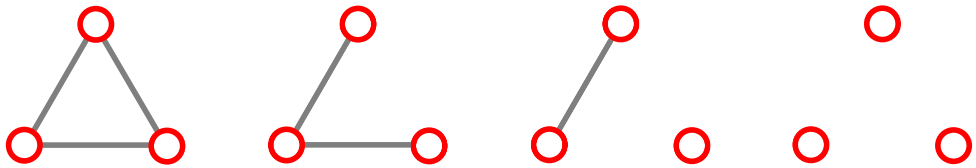
- $k = 4$ graphlets



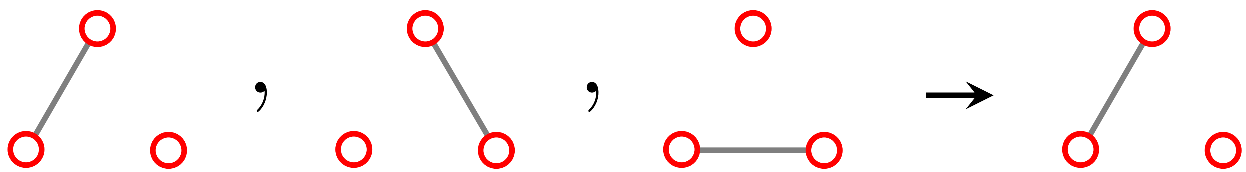
Graphlets

- Graphlets are non-isomorphic induced subgraph patterns of a particular size k with all different combination of edge connections.

- $k = 3$ graphlets



- Graphlets is one instance of an isomorphism type



Graphlet Kernel

- Graphlet kernel counts k-minor graphlets on induced subgraphs of size k in a graph G of with $|\mathrm{V}| = n$.
- They look at $3-$, $4-$, and $5-$minor subgraphs.
- The graphlet kernel is defined as

$$k_{GR}(G, H) = \langle \phi_{GR}(G), \phi_{GR}(H) \rangle$$

where

$$\phi_{GR}(G) = \frac{1}{\sum_n |\sigma_n(G)|} \left(|\sigma_1(G)|, \ldots, |\sigma_N(G)| \right)$$

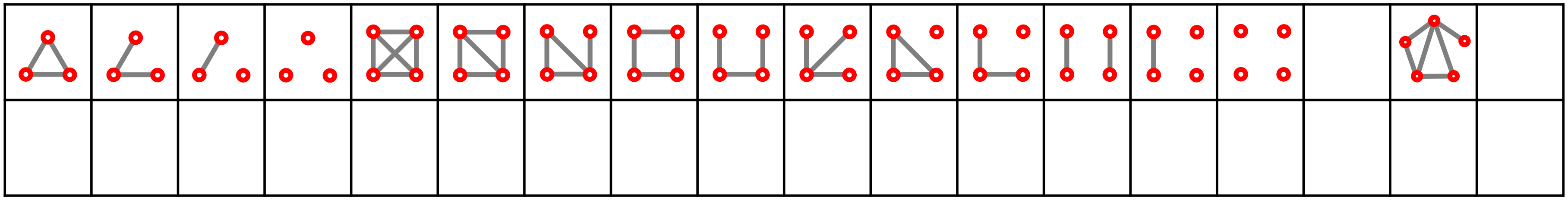
with $\sigma_n(G)$ the set of induced subgraphs of isomorphic type σ_n in G.

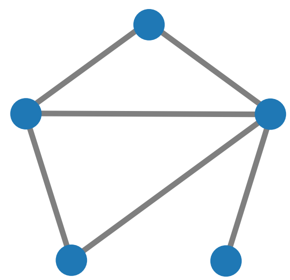
Graphlet Kernel

- $\phi_{GR}(G)$

σ_n

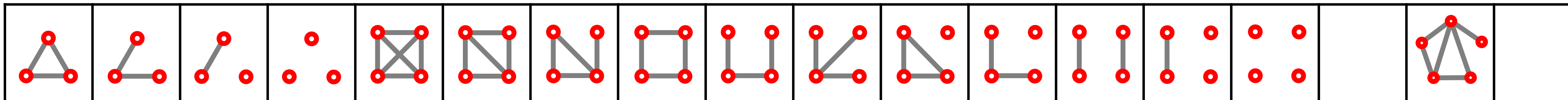
$|\sigma_n|$



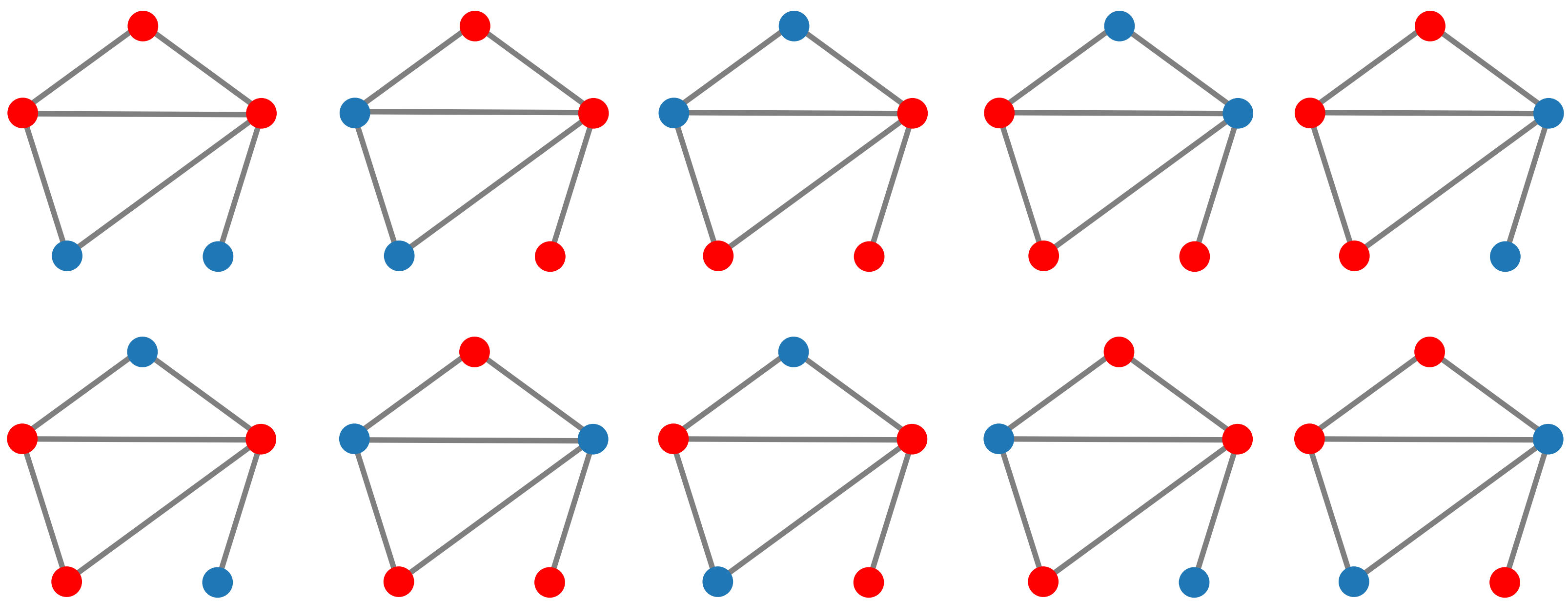


Graphlet Kernel

- $\phi_{GR}(G)$

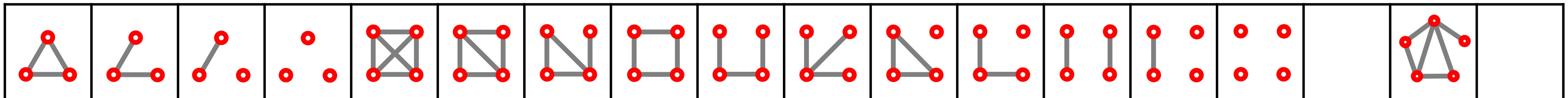
σ_n																		
$\vert\sigma_n\vert$																		

- k=3 minors $\binom{5}{3} = 10$



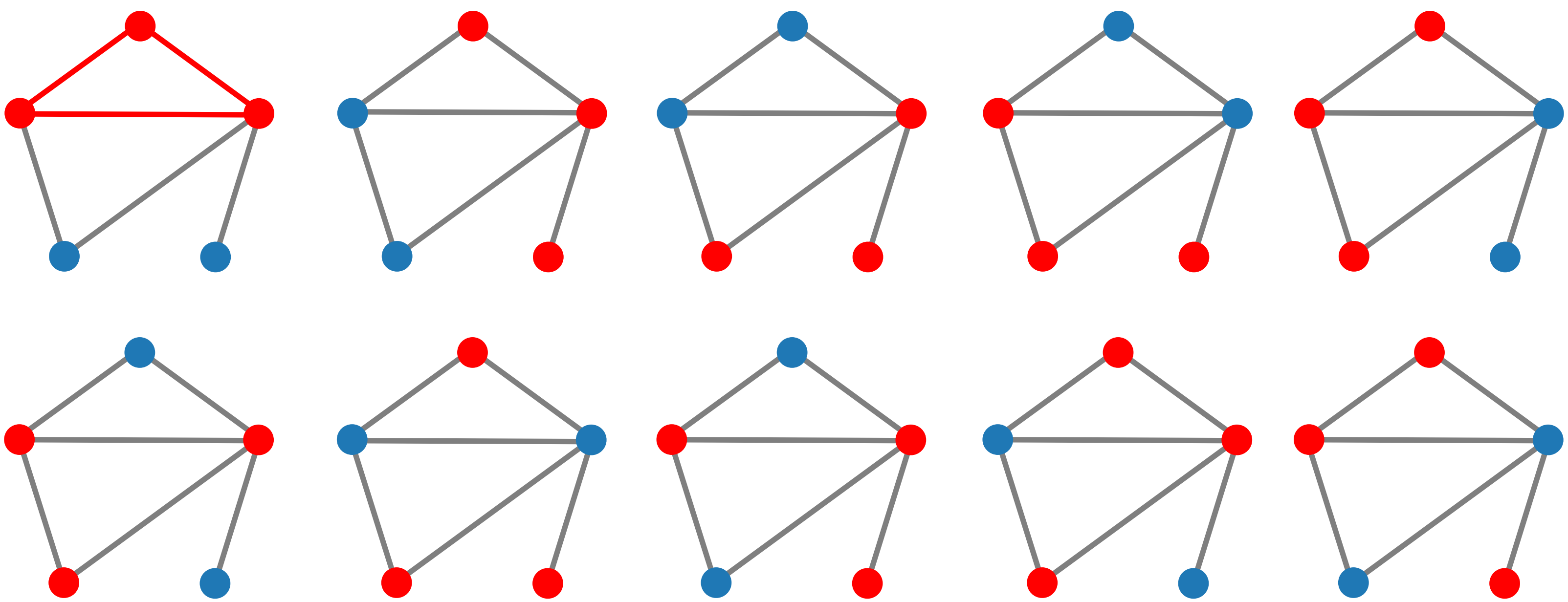
Graphlet Kernel

- $\phi_{GR}(G)$



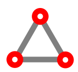
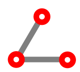
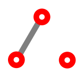
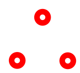
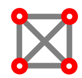
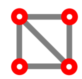
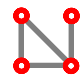
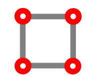
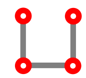
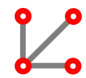
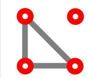
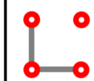
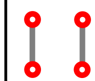
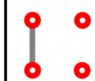
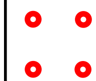
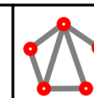
σ_n																		
$\lvert\sigma_n\rvert$																		

- k=3 minors

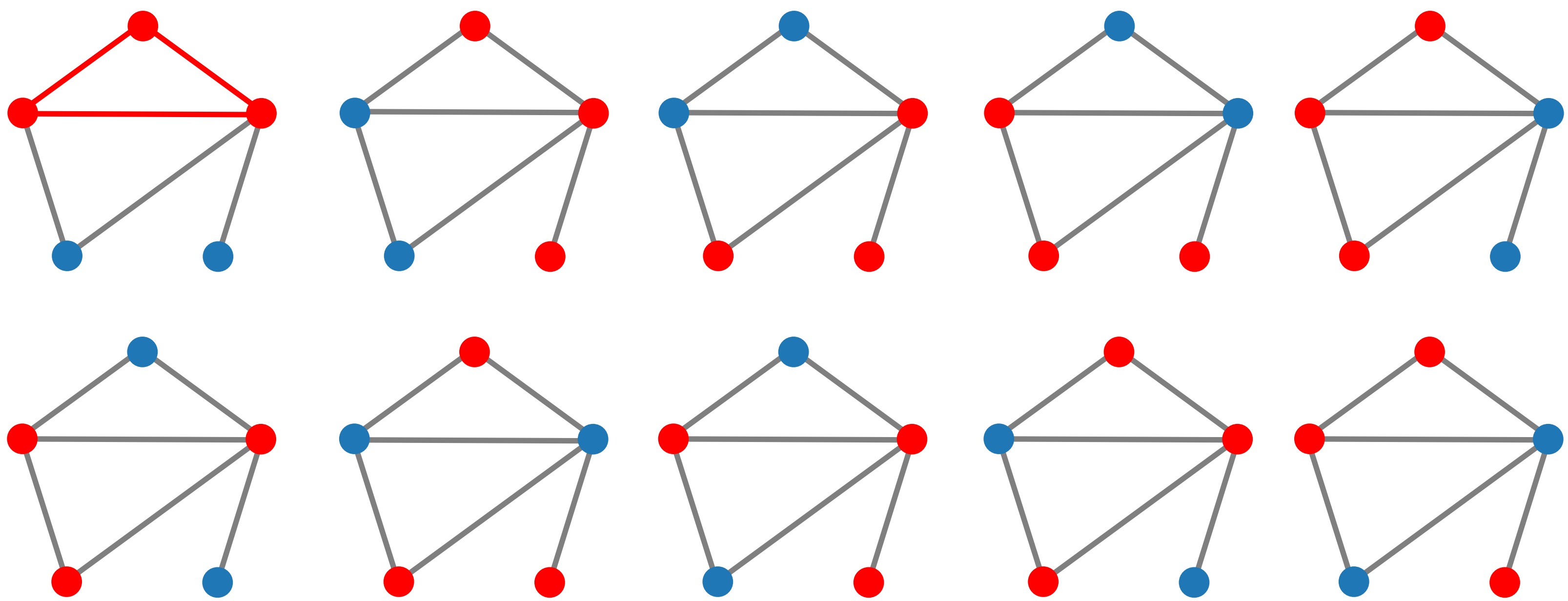


Graphlet Kernel

- $\phi_{GR}(G)$

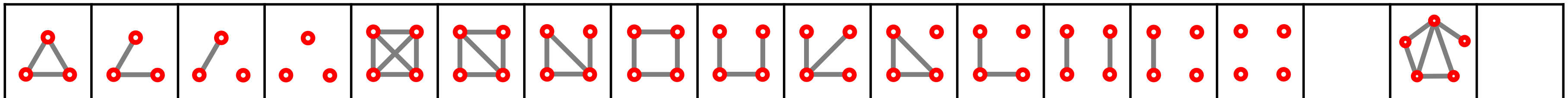
σ_n																		
$\|\sigma_n\|$	1																	

- k=3 minors



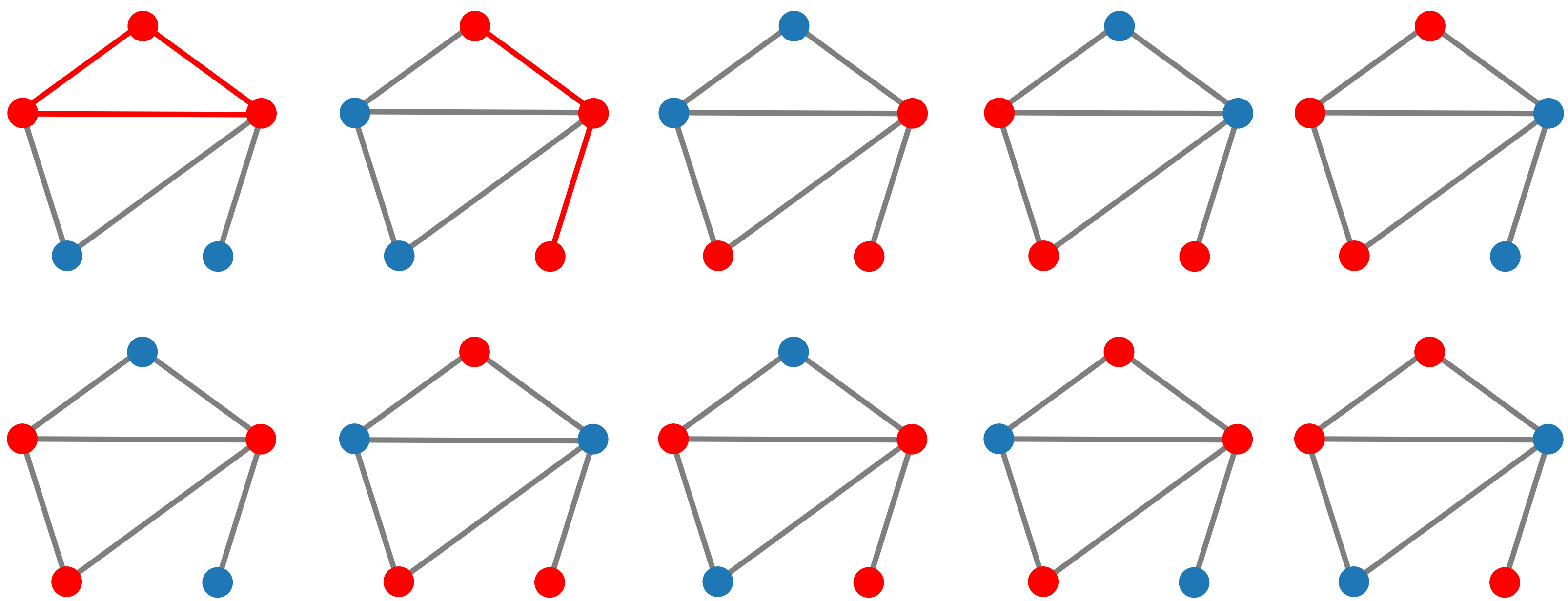
Graphlet Kernel

- $\phi_{GR}(G)$



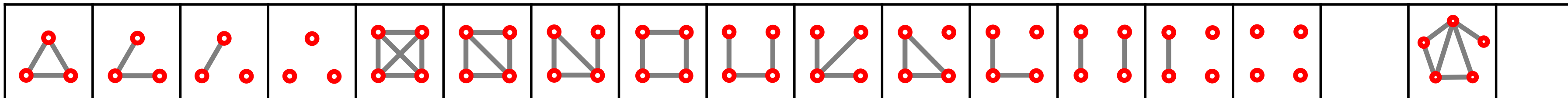
σ_n																		
$\|\sigma_n\|$	1	1																

- k=3 minors



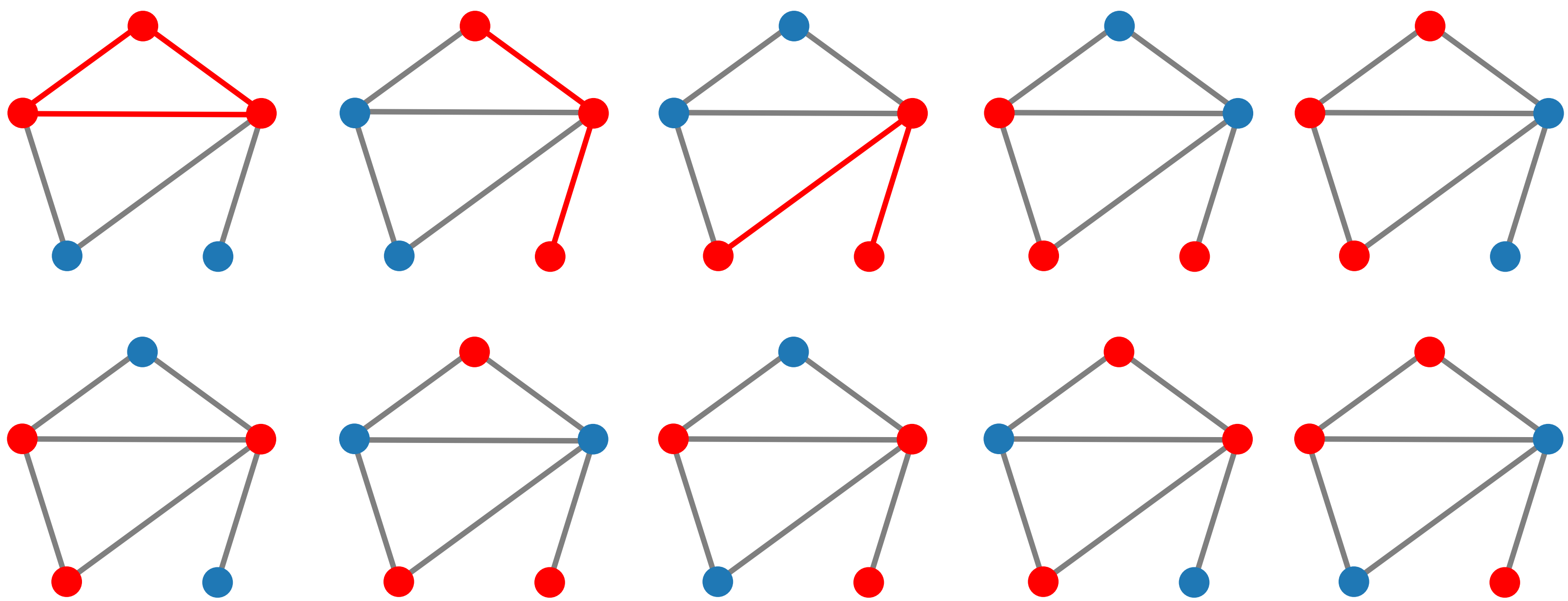
Graphlet Kernel

- $\phi_{GR}(G)$



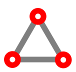
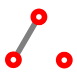
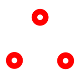
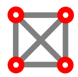
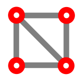
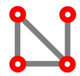
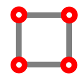
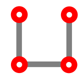
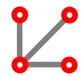
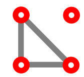
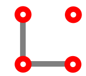
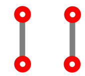
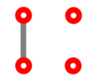
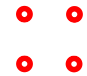
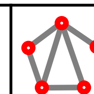
σ_n																		
$\lvert\sigma_n\rvert$	1	2																

- k=3 minors

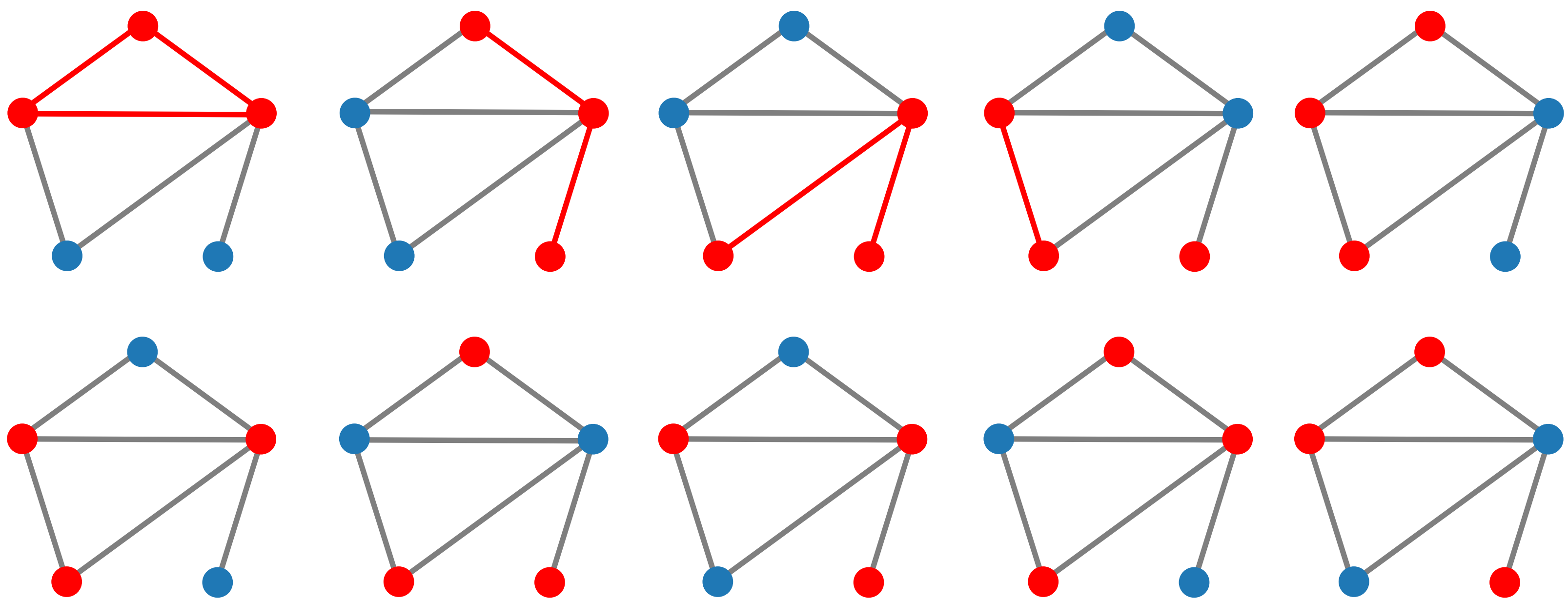


Graphlet Kernel

- $\phi_{GR}(G)$

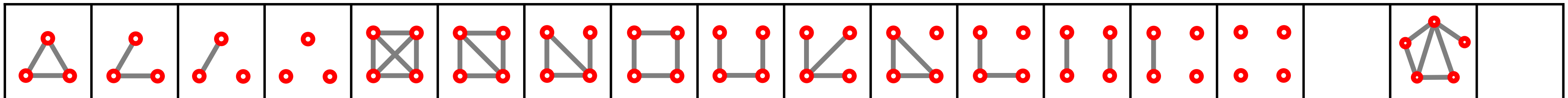
σ_n																		
$\|\sigma_n\|$	1	2	1															

- k=3 minors

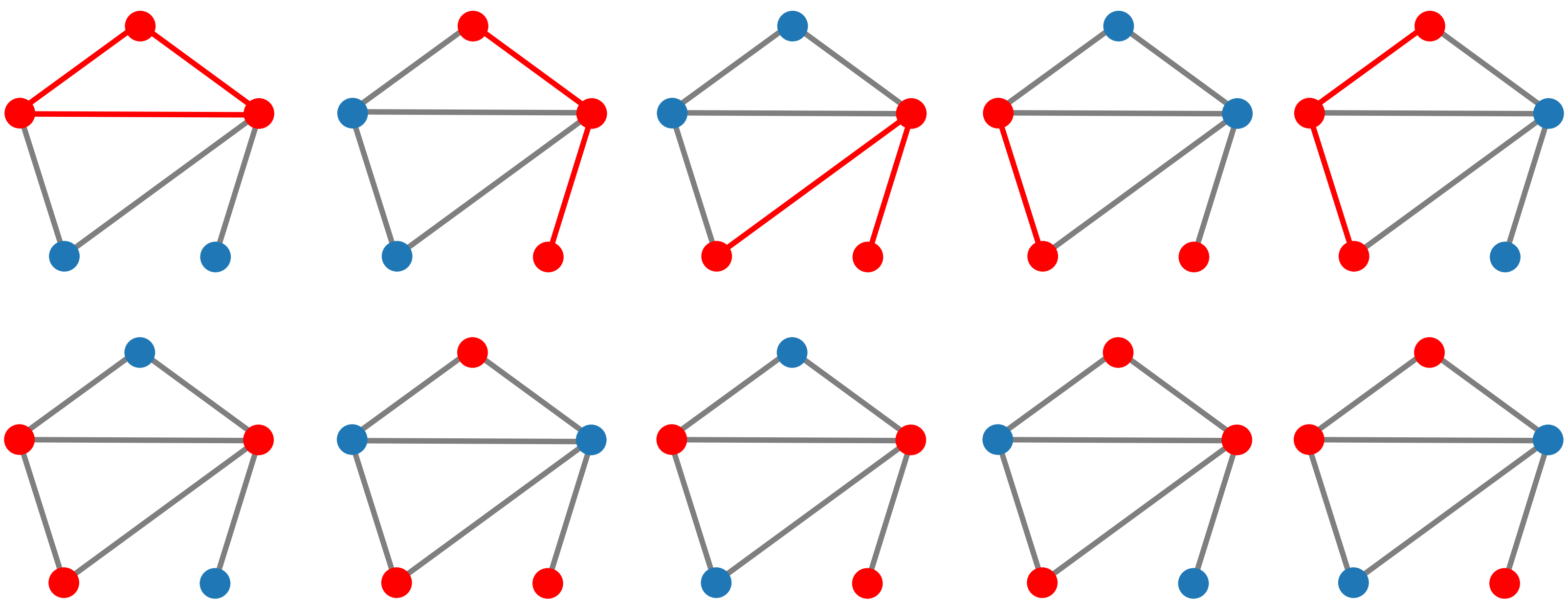


Graphlet Kernel

- $\phi_{GR}(G)$

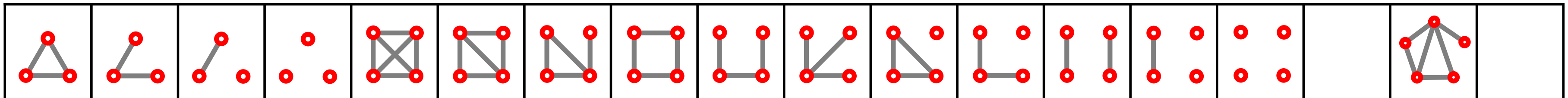
σ_n																		
$\lvert\sigma_n\rvert$	1	3	1															

- k=3 minors



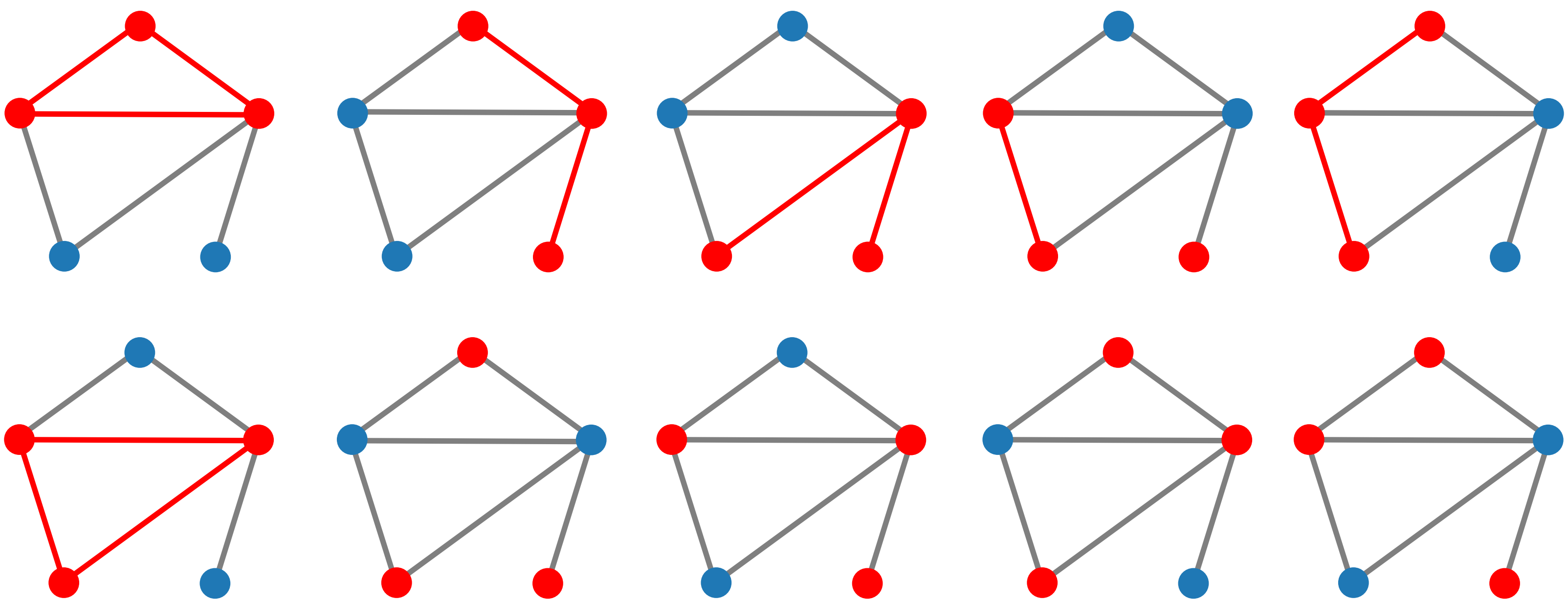
Graphlet Kernel

- $\phi_{GR}(G)$



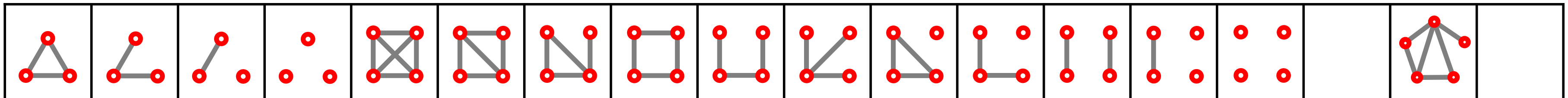
σ_n																		
$\lvert\sigma_n\rvert$	2	3	1															

- k=3 minors



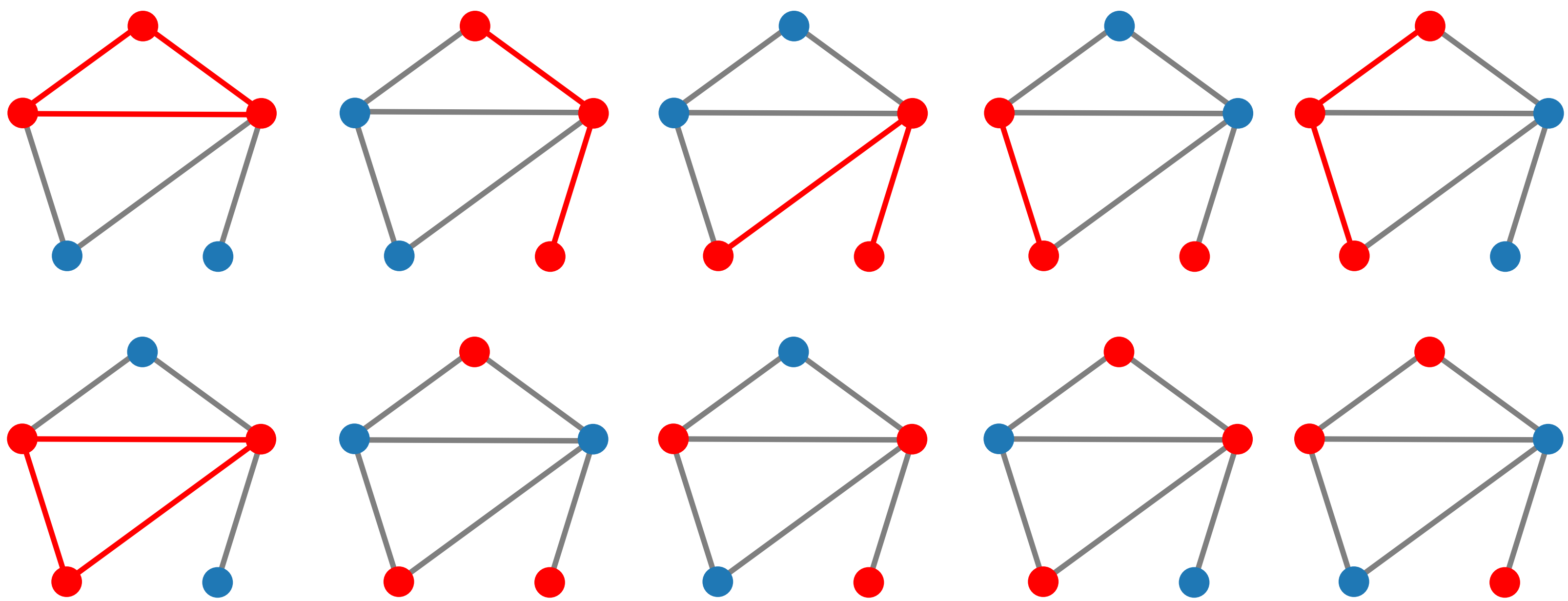
Graphlet Kernel

➢ $\phi_{GR}(G)$



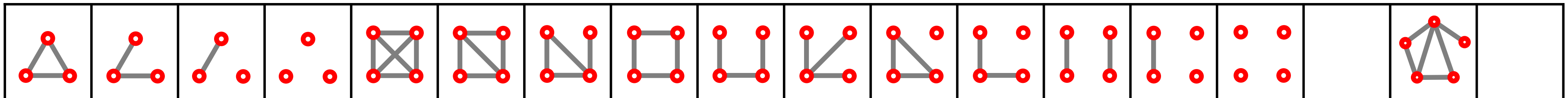
σ_n																		
$\lvert\sigma_n\rvert$	2	3	1	1														

- k=3 minors

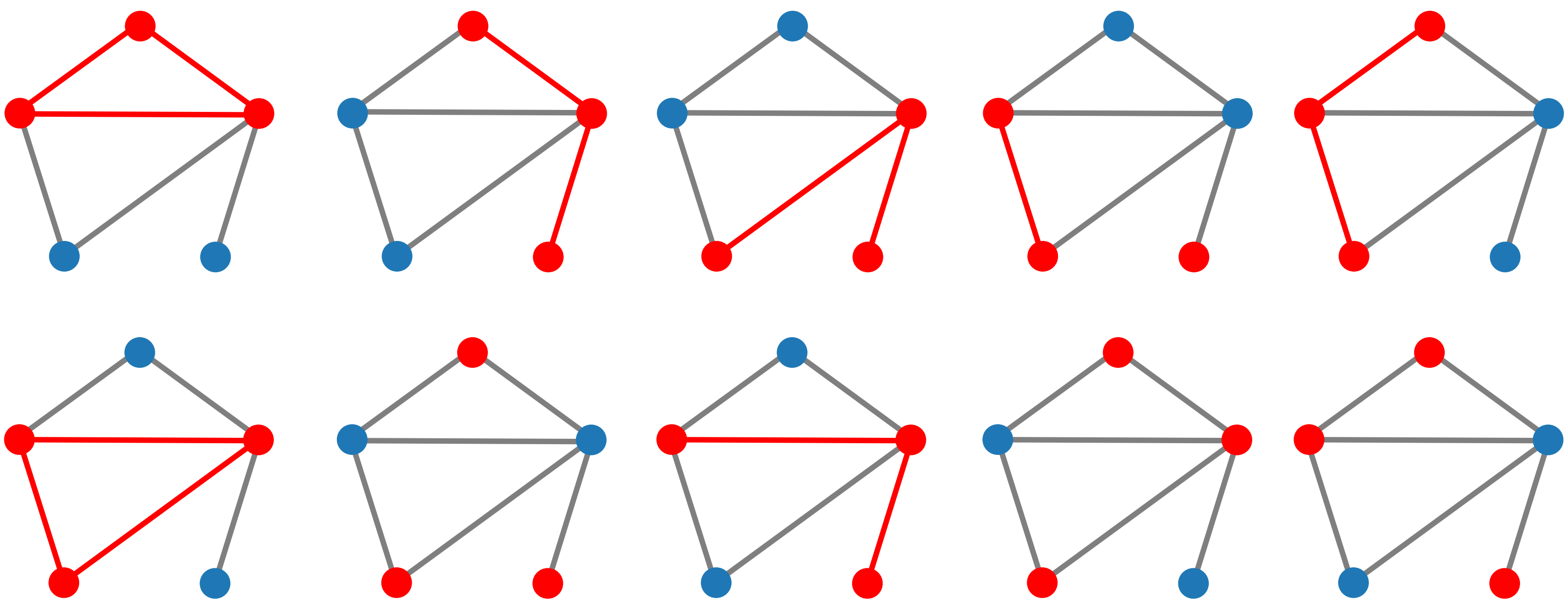


Graphlet Kernel

- $\phi_{GR}(G)$

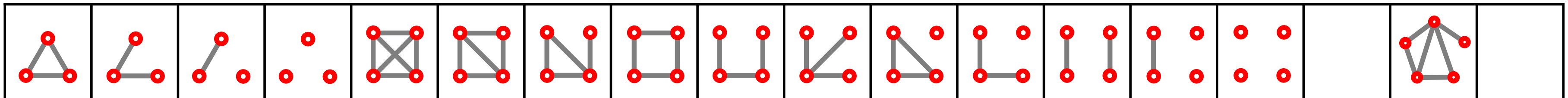
σ_n																		
$\lvert\sigma_n\rvert$	2	3	1	1														

- k=3 minors



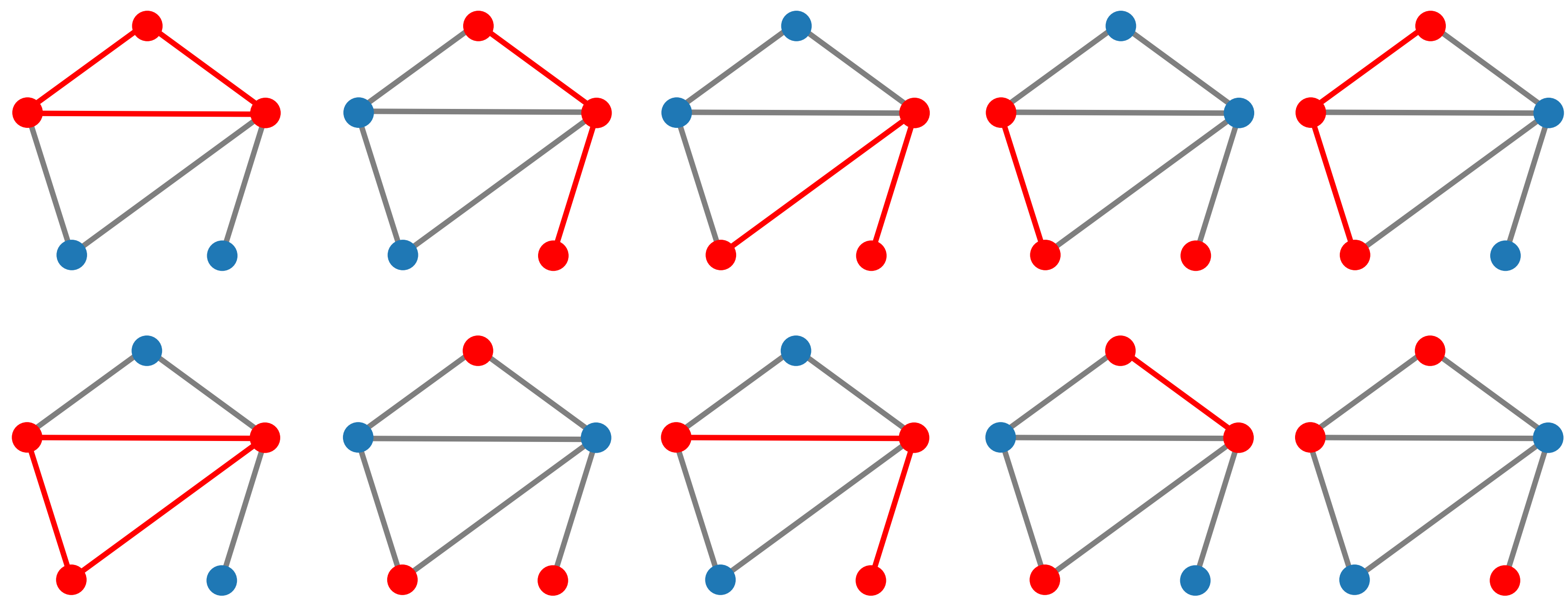
Graphlet Kernel

➢ $\phi_{GR}(G)$



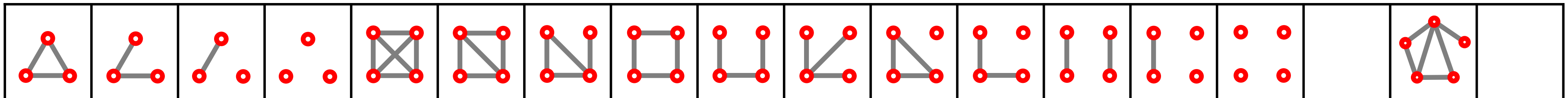
σ_n																		
$\lvert\sigma_n\rvert$	2	3	2	1														

- k=3 minors



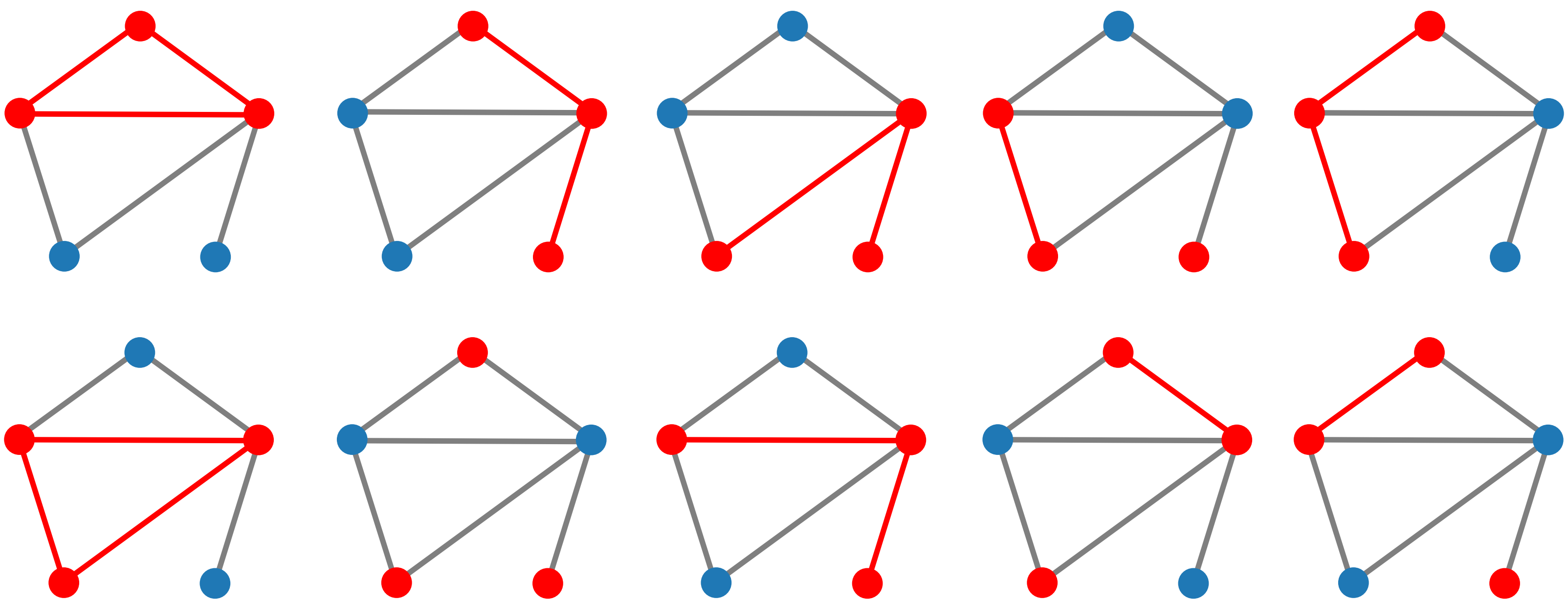
Graphlet Kernel

➢ $\phi_{GR}(G)$



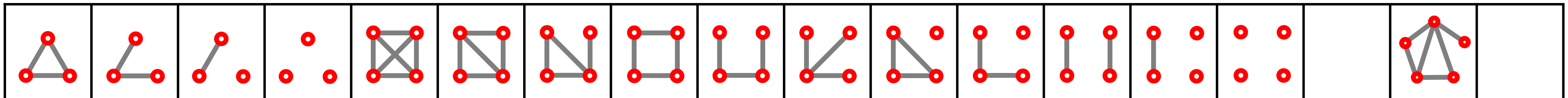
σ_n																		
$\lvert\sigma_n\rvert$	2	3	3	1														

- k=3 minors



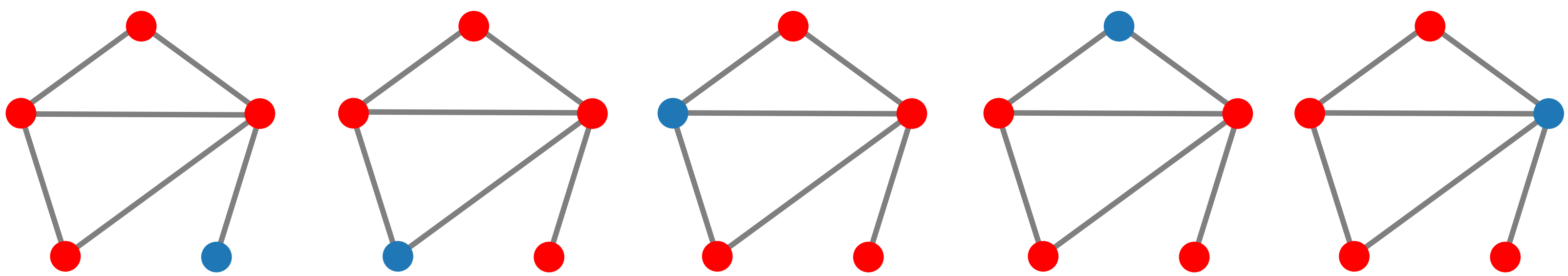
Graphlet Kernel

➢ $\phi_{GR}(G)$



σ_n																		
$\lvert\sigma_n\rvert$	2	3	3	1														

- k=4 minors $\quad \binom{5}{4} = 5$

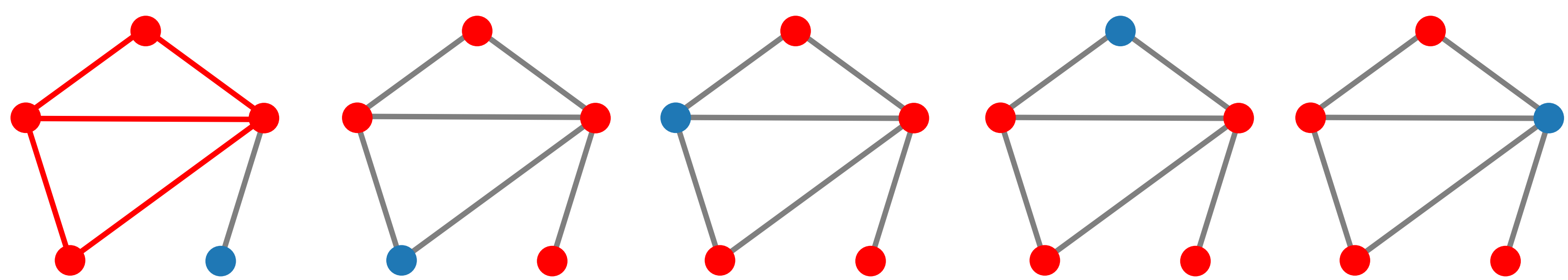


Graphlet Kernel

- $\phi_{GR}(G)$

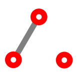
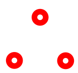
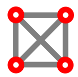
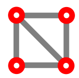
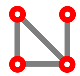
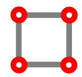
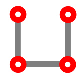
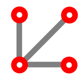
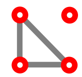
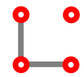
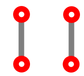
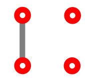
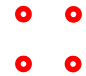
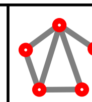
σ_n																		
$\lvert\sigma_n\rvert$	2	3	3	1		1												

- k=4 minors

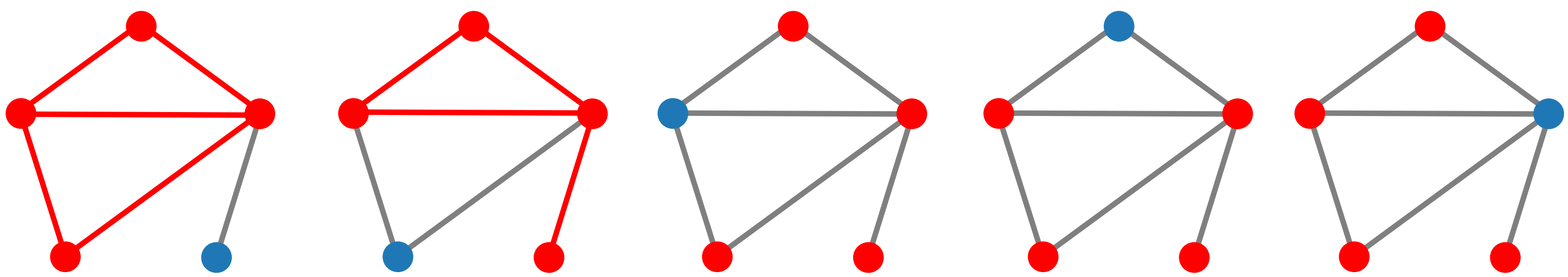


Graphlet Kernel

➤ $\phi_{GR}(G)$

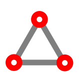
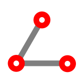
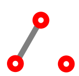
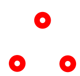
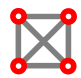
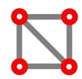
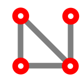
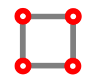
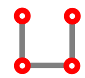
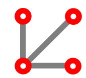
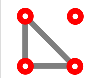
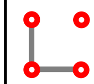
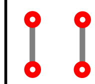
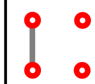
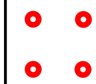
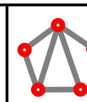
σ_n																		
$\|\sigma_n\|$	2	3	3	1		1	1											

- k=4 minors

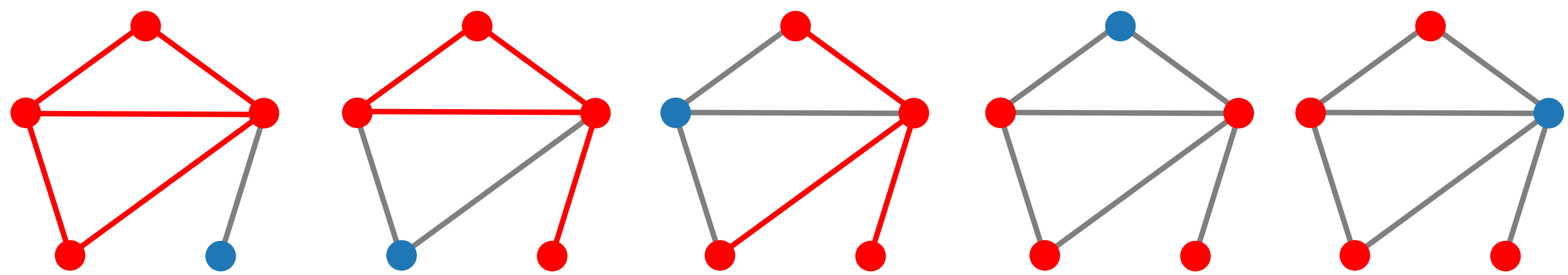


Graphlet Kernel

- $\phi_{GR}(G)$

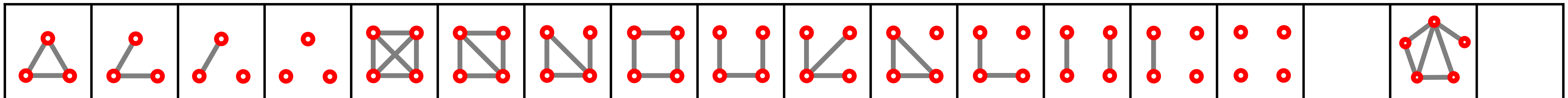
σ_n																		
$\lvert\sigma_n\rvert$	2	3	3	1		1	1			1								

- k=4 minors



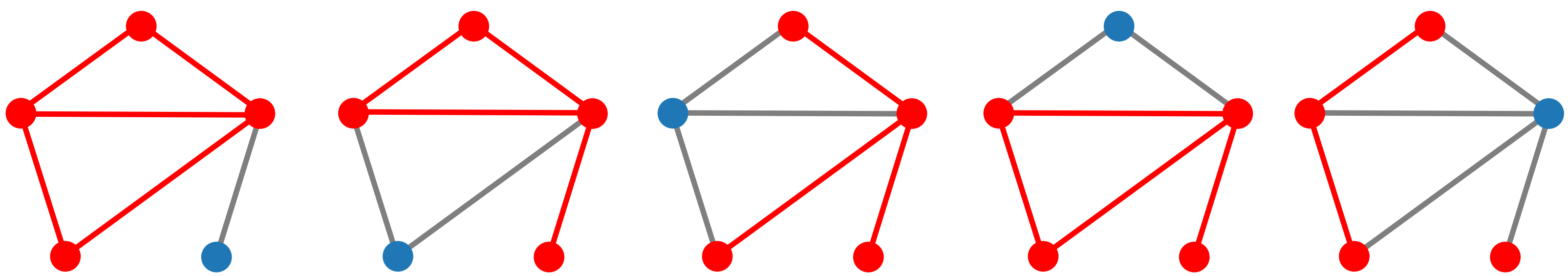
Graphlet Kernel

- $\phi_{GR}(G)$



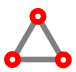
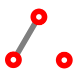
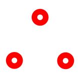
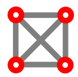
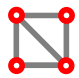
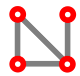
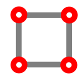
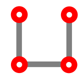
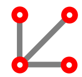
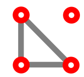
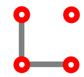
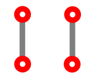
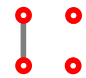
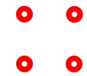
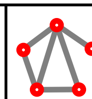
σ_n																		
$\lvert\sigma_n\rvert$	2	3	3	1		1	2			1		1						

- k=4 minors



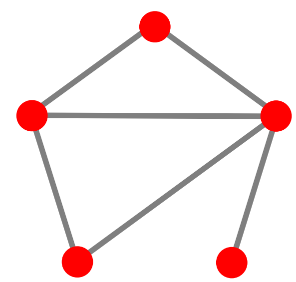
Graphlet Kernel

- $\phi_{GR}(G)$

σ_n																		
$\lvert\sigma_n\rvert$	2	3	3	1		1	2			1		1						

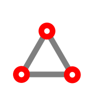
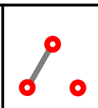
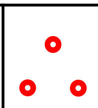
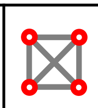
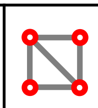
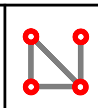
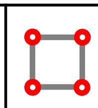
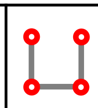
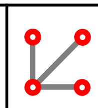
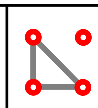
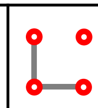
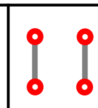
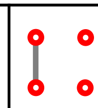
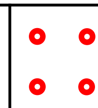
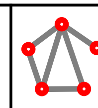
- k=5 minors

$$\binom{5}{5} = 1$$

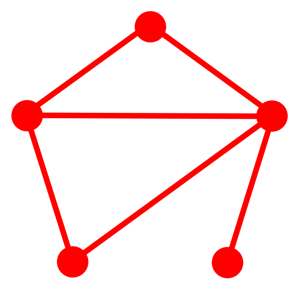


Graphlet Kernel

- $\phi_{GR}(G)$

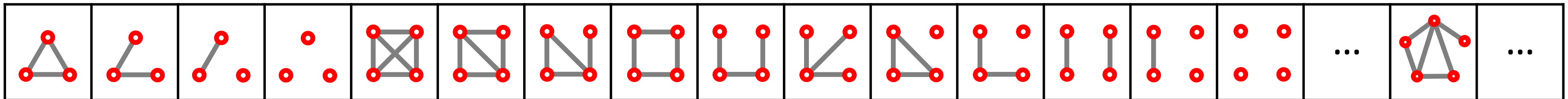
σ_n																		
$\|\sigma_n\|$	2	3	3	1		1	2			1		1					1	

- k=5 minors

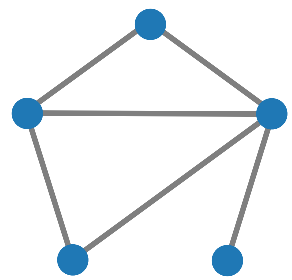


Graphlet Kernel

➢ $\phi_{GR}(G) = (\frac{2}{15}, \frac{3}{15}, \frac{3}{15}, \frac{1}{15}, 0, \frac{1}{15}, \frac{2}{15}, 0, 0, \frac{1}{15}, 0, \frac{1}{15}, 0, 0, 0, \ldots, \frac{1}{15}, \ldots)$



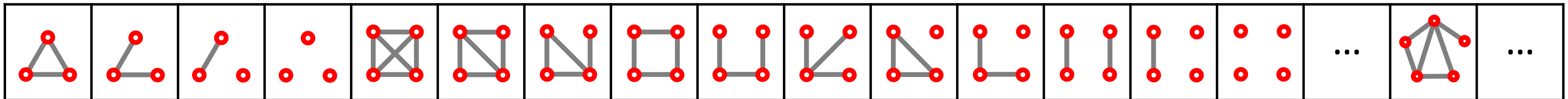
σ_n															...		...
$\lvert\sigma_n\rvert$	2	3	3	1	0	1	2	0	0	1	0	1	0	0	0	...	1	...



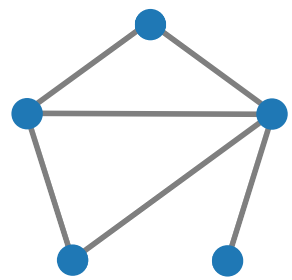
Graphlet Kernel

➢ $\phi_{GR}(G) = (\frac{2}{15}, \frac{3}{15}, \frac{3}{15}, \frac{1}{15}, 0, \frac{1}{15}, \frac{2}{15}, 0, 0, \frac{1}{15}, 0, \frac{1}{15}, 0, 0, 0, \ldots, \frac{1}{15}, \ldots)$

σ_n



$\|\sigma_n\|$	2	3	3	1	0	1	2	0	0	1	0	1	0	0	0	…	1	…



❖ Computationally expensive, so only small graphlets.

$$\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} \leq n^k$$

❖ Small graphlets only look at the local structure.

Neighborhood Aggregation-based Kernels

- Compare two graphs based on their local structure.
- The underlying concept is to iteratively assign summary of the neighborhood attributes as the new attribute for each node.
- While each iteration aggregates local attributes, after a few iterations, each node attribute summarizes the extended neighborhood.
- One such method, namely Weisfeilier-Leman algorithm, was introduced to study isomorphism between two graphs.

Weisfeilier-Leman Kernel

- Is a type of neighborhood aggregation kernels.
- Based on the WL test of isomorphism
- Label function $l: V^G \cup V^H \rightarrow \Sigma$.
  - Set initial label $l_i^{(0)} = 1$ for each node $v_i \in V^G \cup V^H$.
  - At each iteration, aggregate labels of neighboring nodes and construct a tuple of the current label $l_i^{(t-1)}$ and the multiset of the aggregated neighborhood labels.

$$l_i^{(t-1)}, \{\{l_j^{(t-1)} | v_j \in N(v_i)\}\}$$

  - Relabel each unique tuple as $\sigma_k^{(t)} \in \Sigma^{(t)}$.
  - Set new labels $l_i^{(t)}$ as the corresponding $\sigma_k^{(t)}$.

Weisfeilier-Leman Kernel

- The feature vectors $\phi^{(t)}(G)$ at each iteration can be used to compare the two graphs.
- The Weisfeilier-Leman graph kernel is defined as

$$k_{WL}(G, H) = \langle \phi_{WL}(G), \phi_{WL}(H) \rangle$$

where

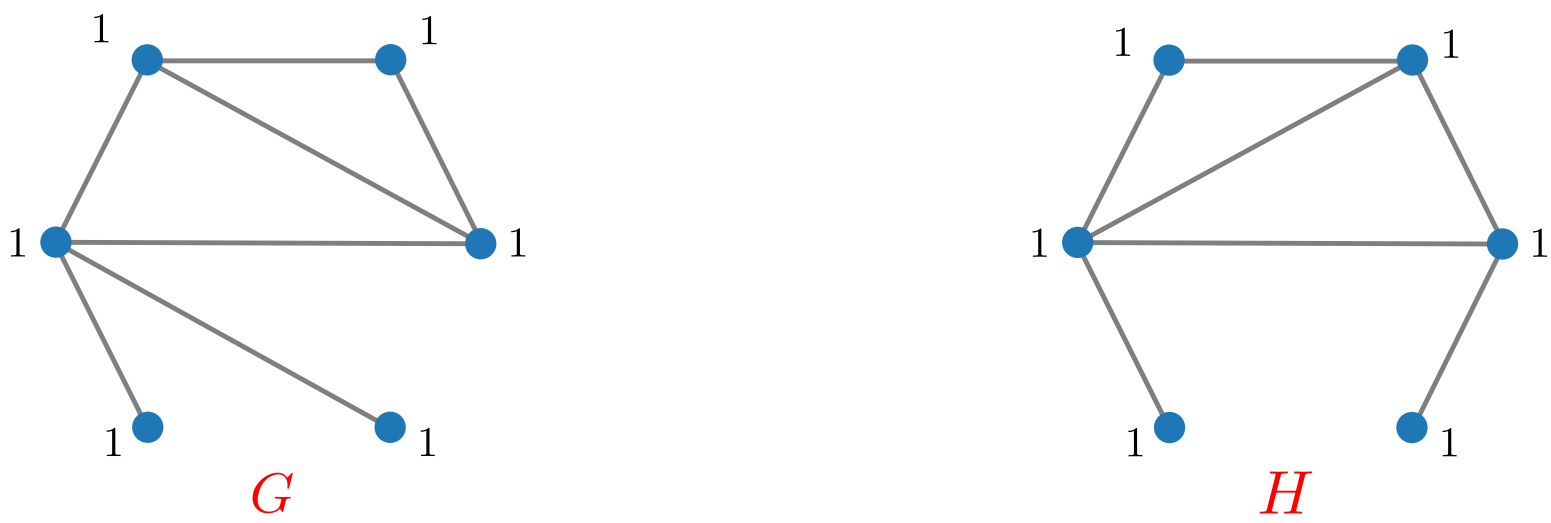
$$\phi_{WL}(G) = \left(|\sigma_1^{(0)}(G)|, \ldots, |\sigma_{|\Sigma^{(T)}|}^{(T)}(G)| \right)$$

with $\Sigma^{(t)}$ the set of labels at iteration t and $\sigma_i^{(t)}$ as

$$\sigma_n^{(t)}(G) = \left\{ v_i \mid v_i \in V^G, l_i^{(t)} = \Sigma_n^{(t)} \right\}$$

Weisfeilier-Leman Kernel

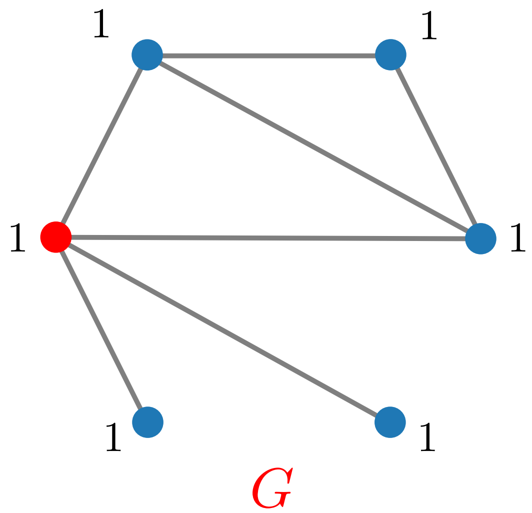
- $t = 0$

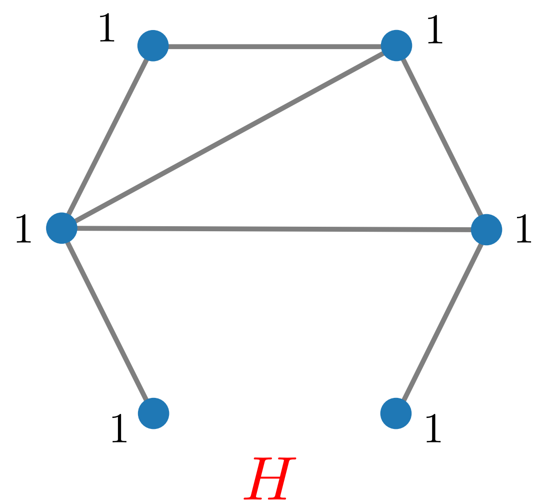


$$\Sigma^{(0)} = \{1\}$$

Weisfeilier-Leman Kernel

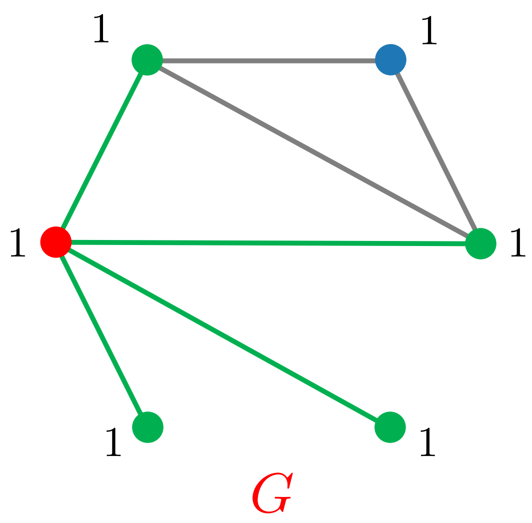
- $t = 1$

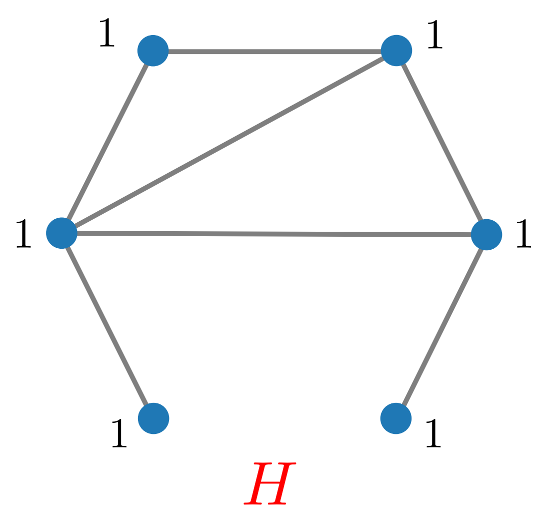




Weisfeilier-Leman Kernel

- $t = 1$

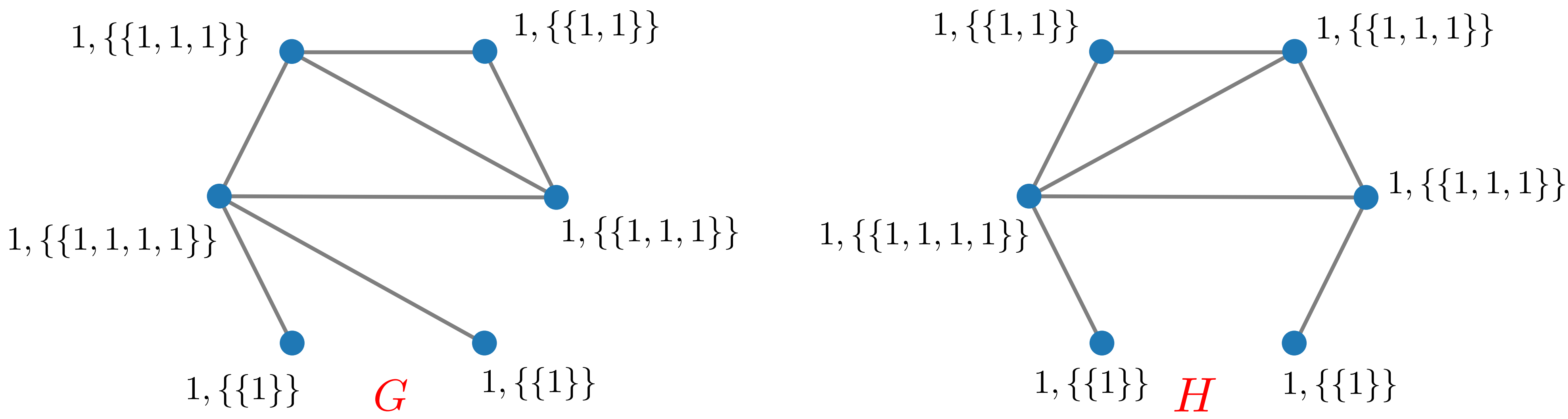




$$1, \{\{1, 1, 1, 1\}\}$$

Weisfeilier-Leman Kernel

- $t = 1$



$2 \leftarrow 1, \{\{1\}\}$

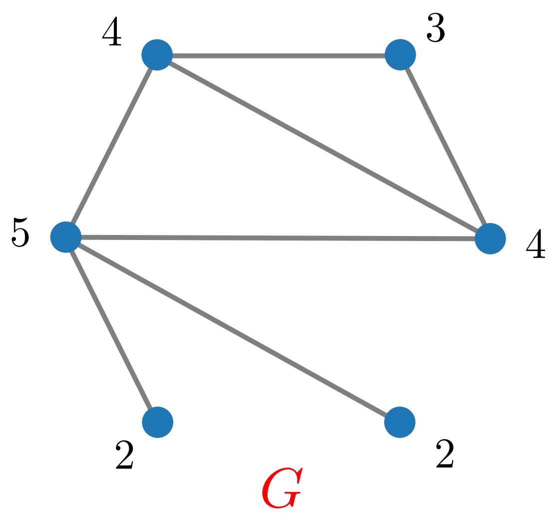
$3 \leftarrow 1, \{\{1, 1\}\}$

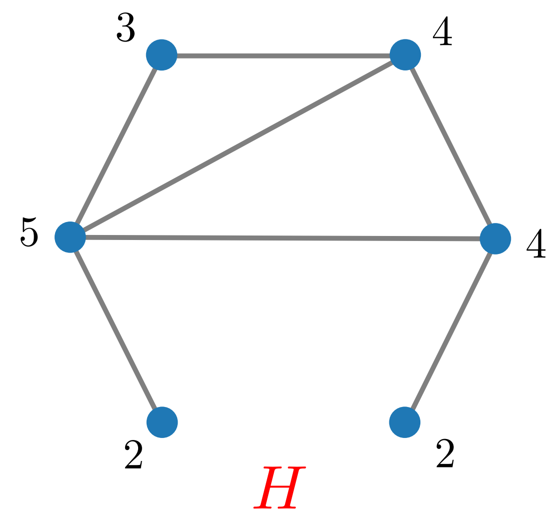
$4 \leftarrow 1, \{\{1, 1, 1\}\}$

$5 \leftarrow 1, \{\{1, 1, 1, 1\}\}$

Weisfeilier-Leman Kernel

- $t = 1$



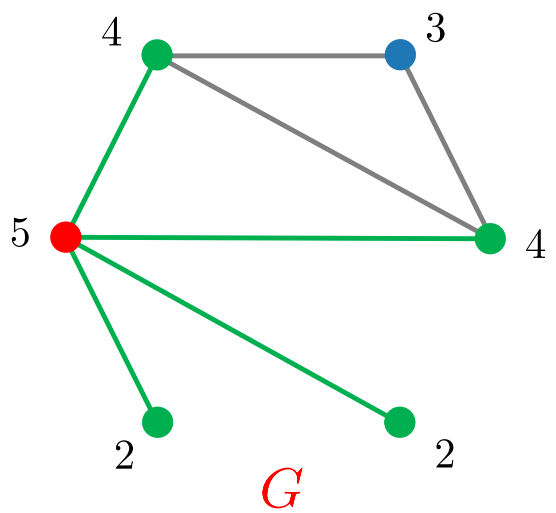


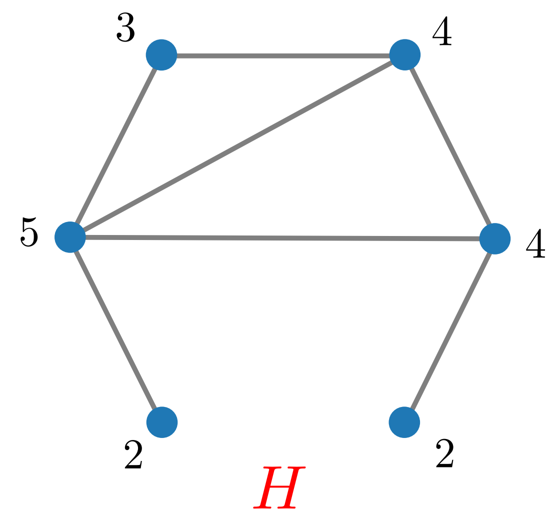
$$\Sigma^{(1)} = \{2, 3, 4, 5\}$$

$$2 \leftarrow 1, \{\{1\}\}$$
$$3 \leftarrow 1, \{\{1, 1\}\}$$
$$4 \leftarrow 1, \{\{1, 1, 1\}\}$$
$$5 \leftarrow 1, \{\{1, 1, 1, 1\}\}$$

Weisfeilier-Leman Kernel

- $t = 2$

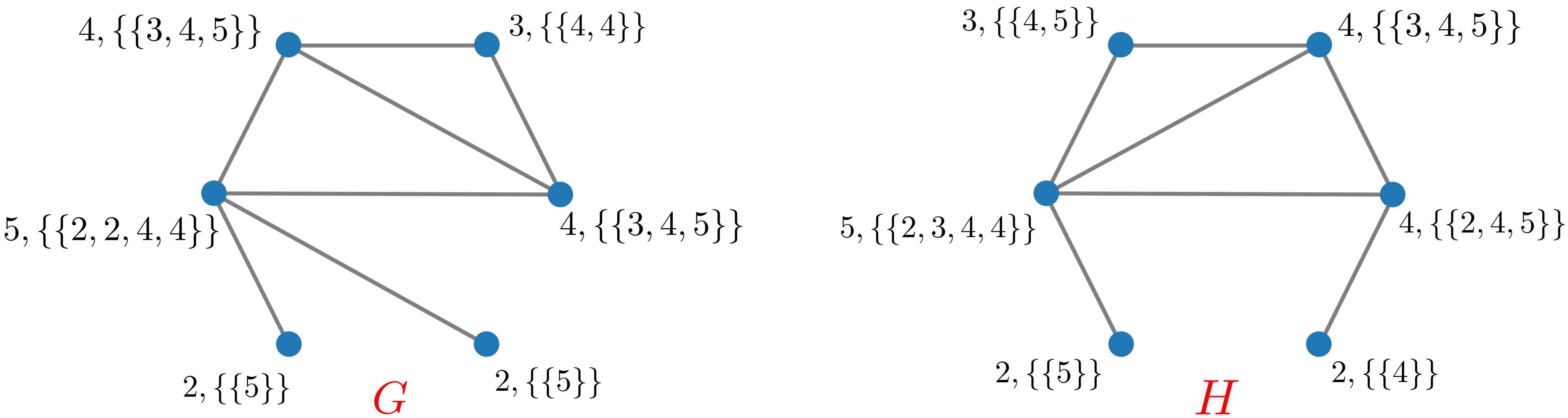




$$5, \{\{2, 2, 4, 4\}\}$$

Weisfeilier-Leman Kernel

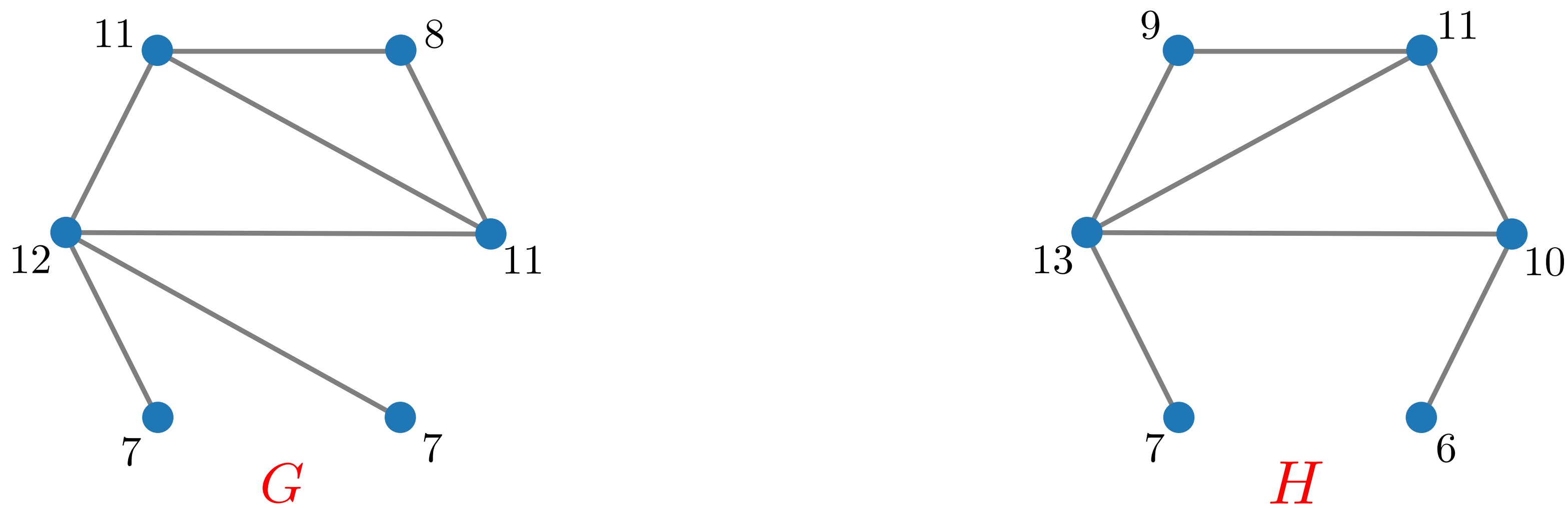
- $t = 2$



$6 \leftarrow 2, \{\{4\}\}$	$10 \leftarrow 4, \{\{2, 4, 5\}\}$
$7 \leftarrow 2, \{\{5\}\}$	$11 \leftarrow 4, \{\{3, 4, 5\}\}$
$8 \leftarrow 3, \{\{4, 4\}\}$	$12 \leftarrow 5, \{\{2, 2, 4, 4\}\}$
$9 \leftarrow 3, \{\{4, 5\}\}$	$13 \leftarrow 5, \{\{2, 3, 4, 4\}\}$

Weisfeilier-Leman Kernel

- $t = 2$



$$\Sigma^{(2)} = \{6, 7, 8, 9, 10, 11, 12, 13\}$$

$$6 \leftarrow 2, \{\{4\}\} \qquad 10 \leftarrow 4, \{\{2, 4, 5\}\}$$
$$7 \leftarrow 2, \{\{5\}\} \qquad 11 \leftarrow 4, \{\{3, 4, 5\}\}$$
$$8 \leftarrow 3, \{\{4, 4\}\} \qquad 12 \leftarrow 5, \{\{2, 2, 4, 4\}\}$$
$$9 \leftarrow 3, \{\{4, 5\}\} \qquad 13 \leftarrow 5, \{\{2, 3, 4, 4\}\}$$

Weisfeilier-Leman Kernel

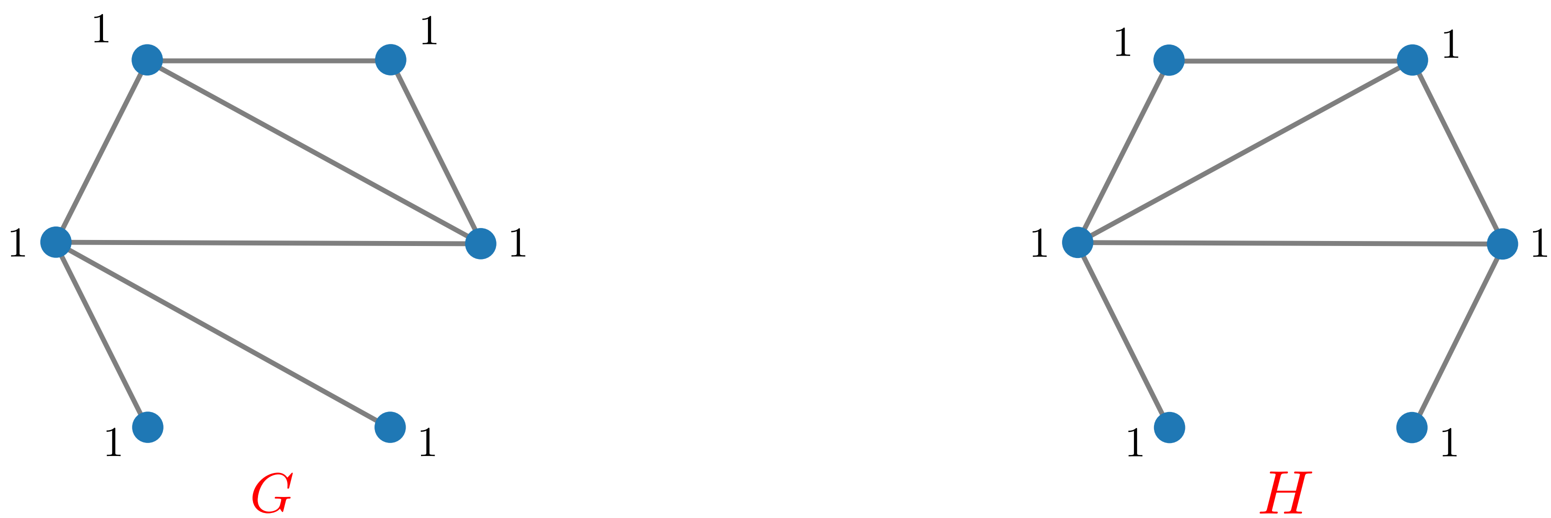
➤

$\Sigma^{(0)}$ $\Sigma^{(1)}$ $\Sigma^{(2)}$

	1	2	3	4	5	6	7	8	9	10	11	12	13
$\phi_{WL}(G)$													
$\phi_{WL}(H)$													

Weisfeilier-Leman Kernel

➤

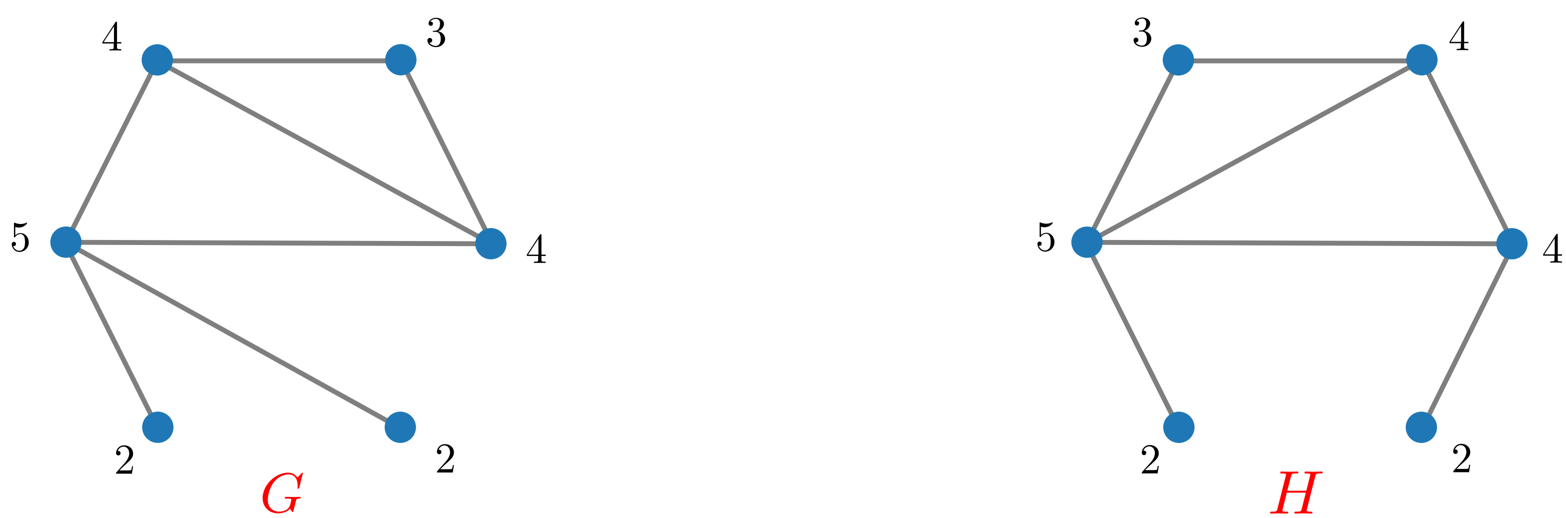


$\Sigma^{(0)}$ $\Sigma^{(1)}$ $\Sigma^{(2)}$

	1	2	3	4	5	6	7	8	9	10	11	12	13
$\phi_{WL}(G)$	6												
$\phi_{WL}(H)$	6												

Weisfeiler-Leman Kernel

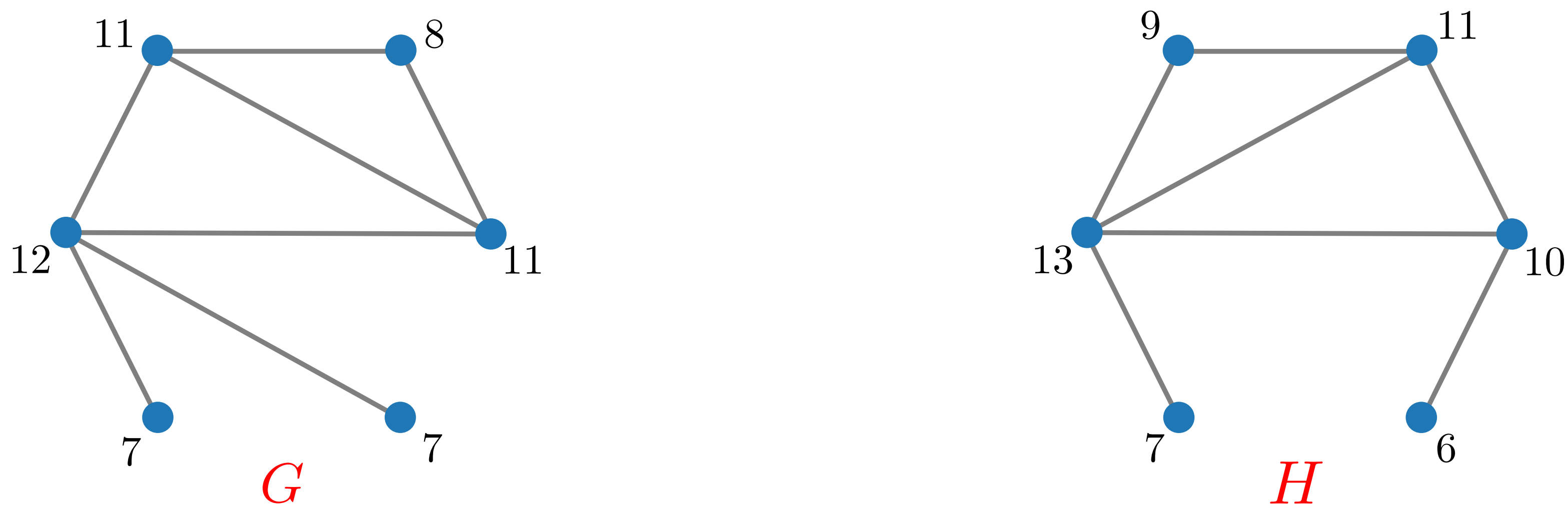
➤



$\Sigma^{(0)}$ $\Sigma^{(1)}$ $\Sigma^{(2)}$

	1	2	3	4	5	6	7	8	9	10	11	12	13
$\phi_{WL}(G)$	6	2	1	2	1								
$\phi_{WL}(H)$	6	2	1	2	1								

Weisfeiler-Leman Kernel



$\Sigma^{(0)}$ $\Sigma^{(1)}$ $\Sigma^{(2)}$

	1	2	3	4	5	6	7	8	9	10	11	12	13
$\phi_{WL}(G)$	6	2	1	2	1	0	2	1	0	0	2	1	0
$\phi_{WL}(H)$	6	2	1	2	1	1	1	0	1	1	1	0	1

Summary

- Kernel Methods
- Isomorphism
- Graph kernel
- Graph kernel design
- Substructure-Based Kernels
- Bag of nodes kernel
- Bag of node degrees
- Induced subgraph
- Graphlet
- Graphlet kernel
- Neighborhood Aggregation-based Kernels
- Weisfeilier-Leman Kernel