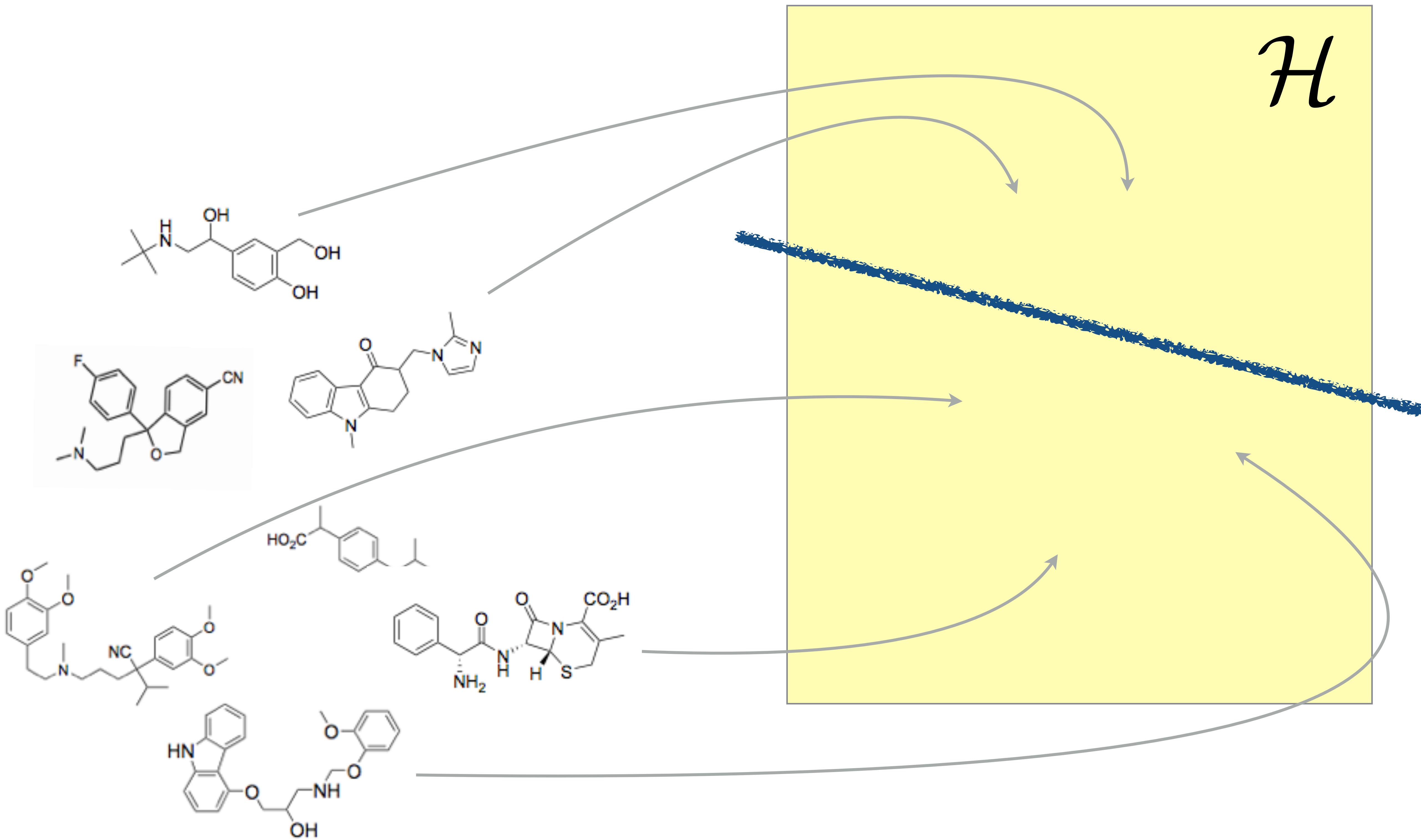
The multiscale Laplacian graph kernel

Risi Kondor and Horace Pan

The University of Chicago

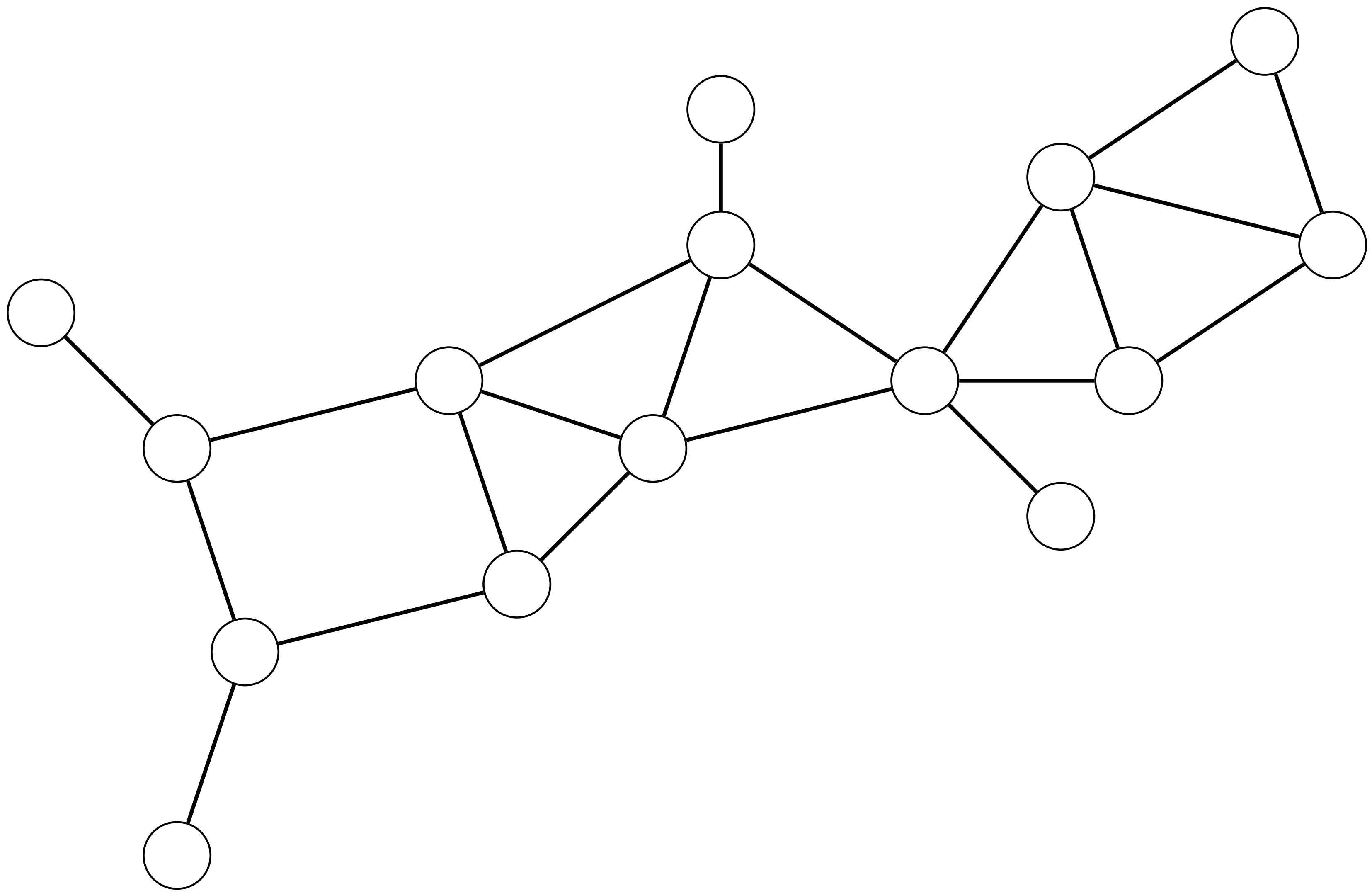
Graph kernel $k(\mathcal{G}_1, \mathcal{G}_2)$



Basic requirements:

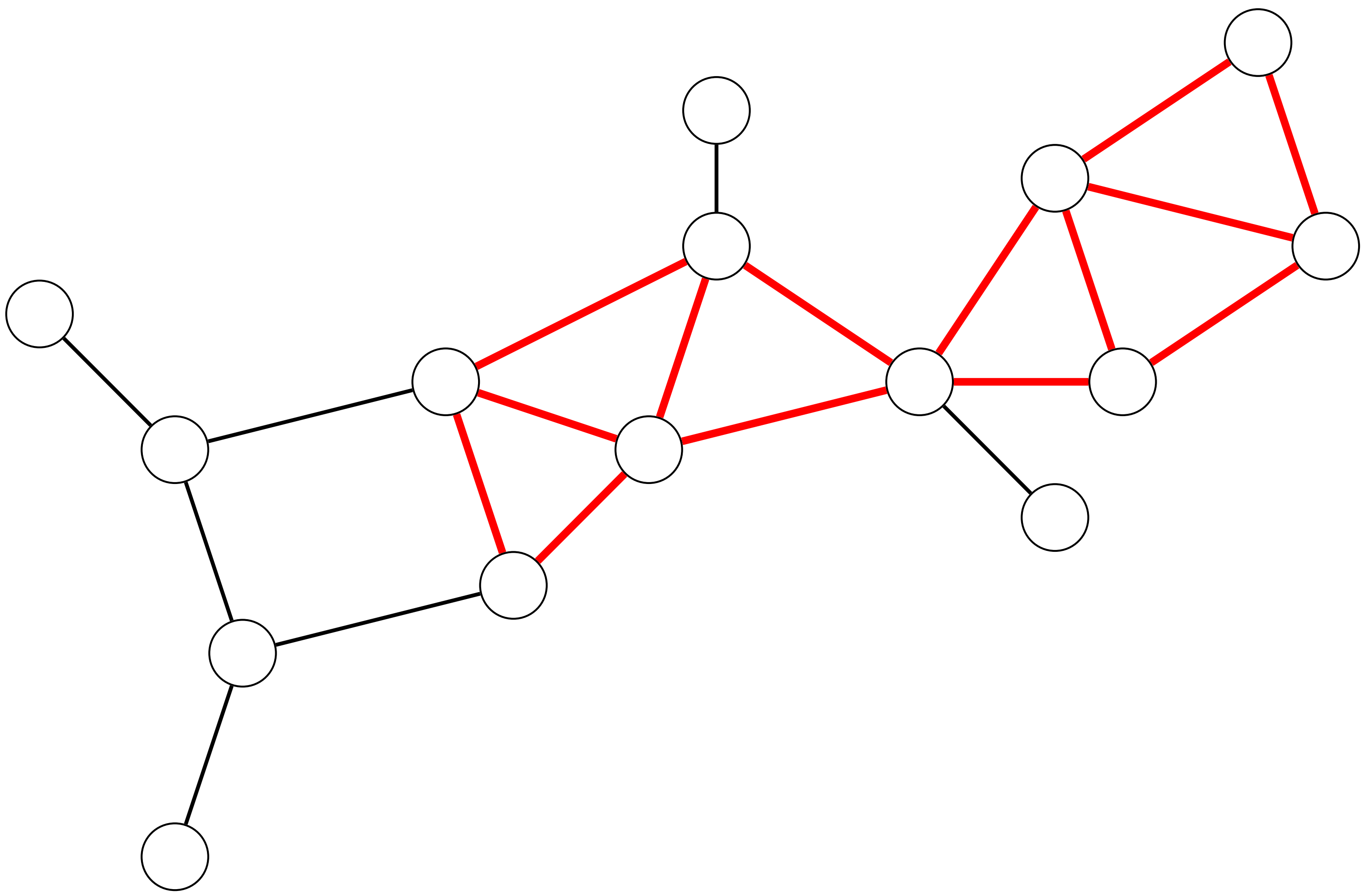
- Positive semi-definitess
- Invariance to permuting the vertices
- Should capture a sensible notion of similarity

1. Local graph kernels



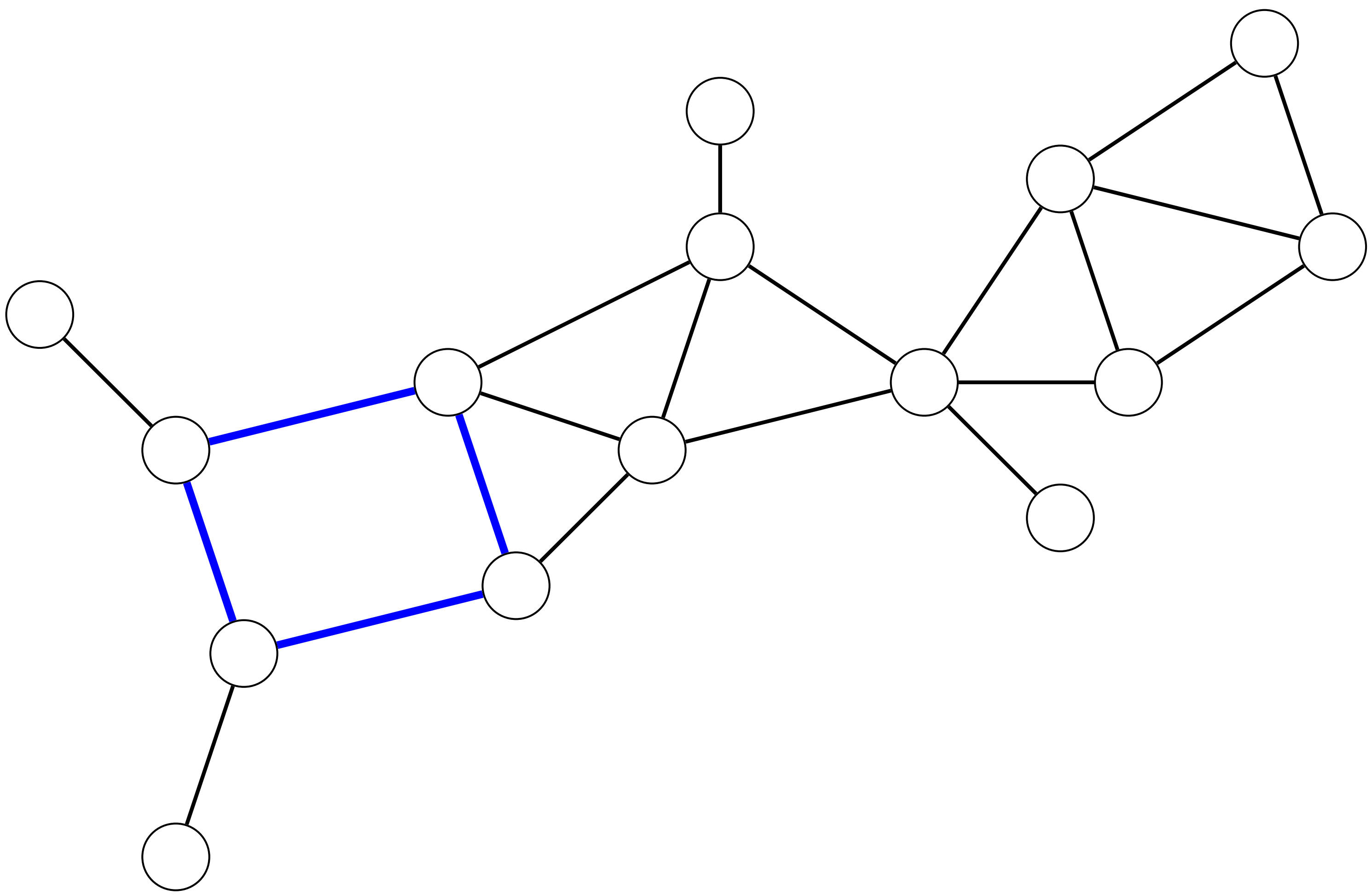
e.g., Graphlet kernels [Shervashidze et al., 2009]

1. Local graph kernels



e.g., Graphlet kernels [Shervashidze et al., 2009]

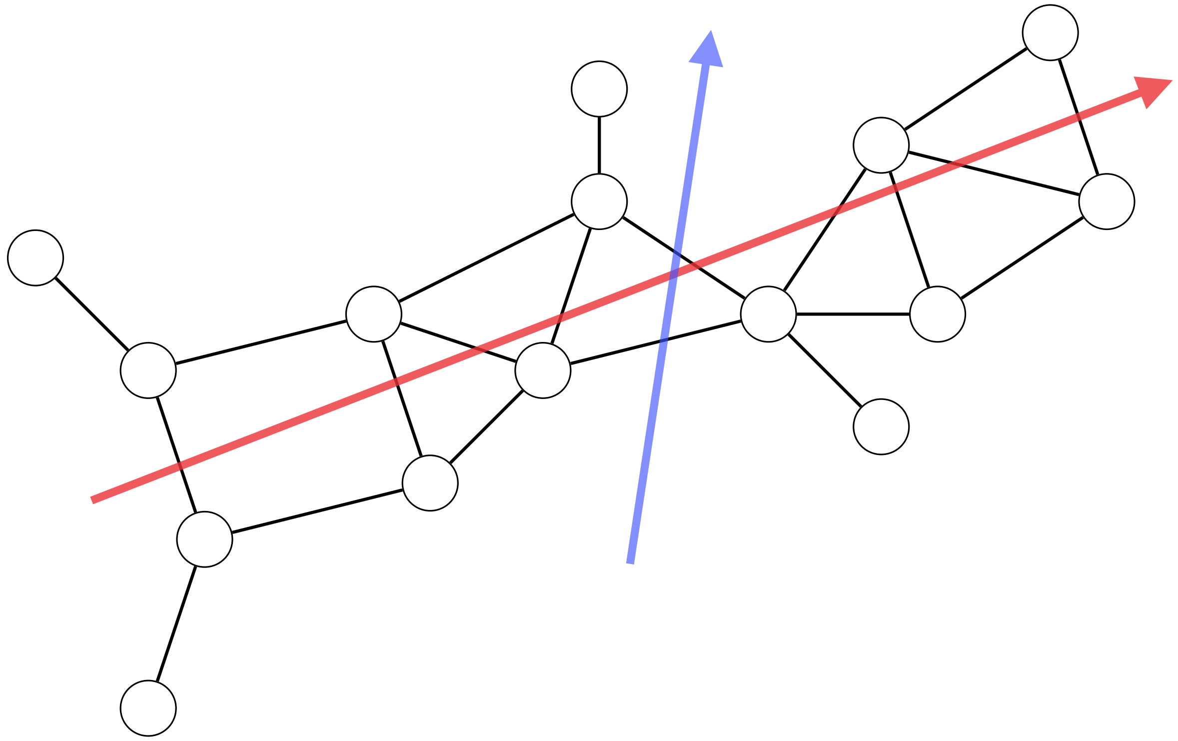
1. Local graph kernels



e.g., Graphlet kernels [Shervashidze et al., 2009]

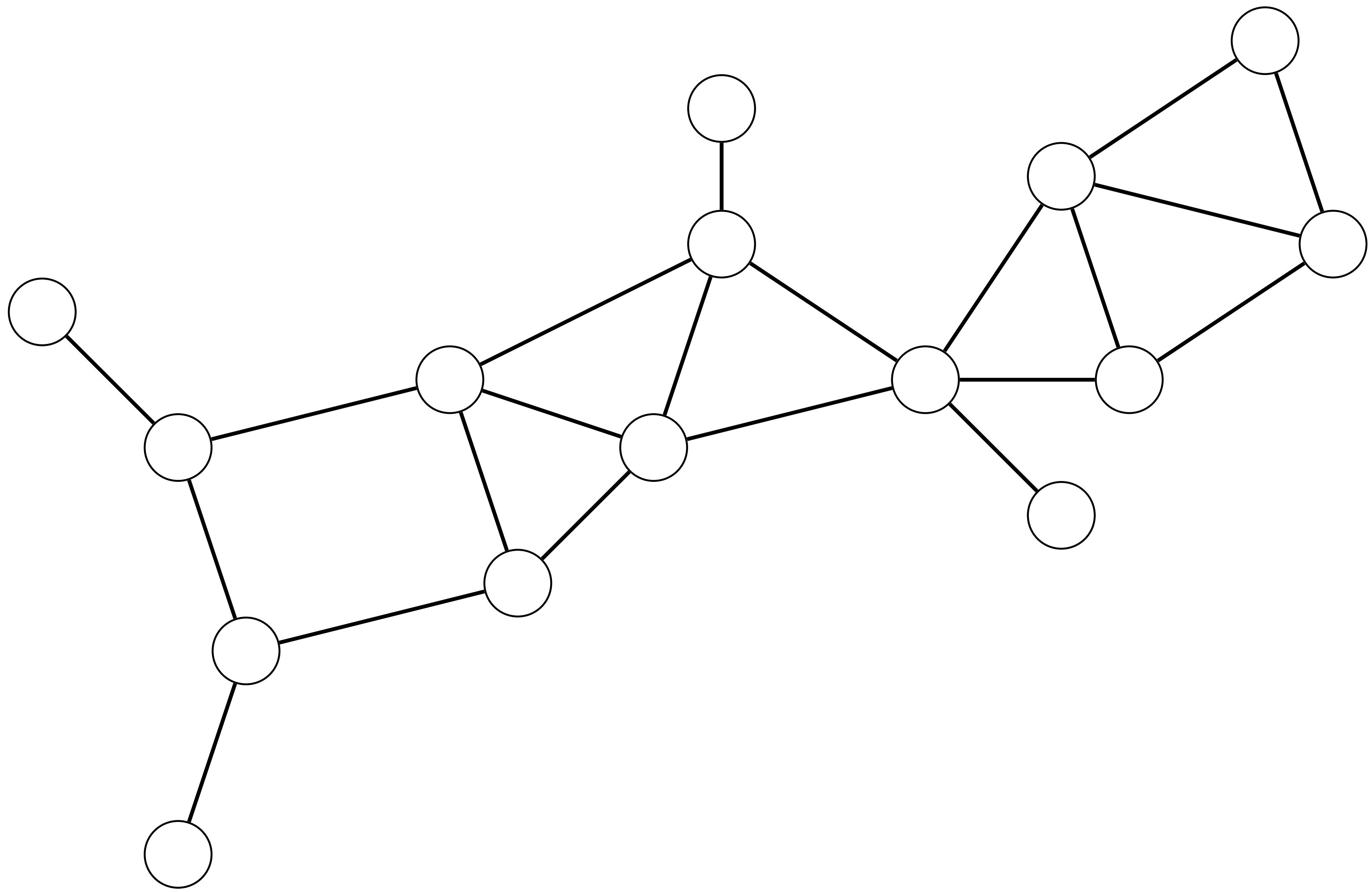
2. Spectral graph kernels

$$L = \begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 4 & -1 & -1 & -1 \\ & & -1 & 1 & & \\ & & -1 & & 2 & -1 \\ & & -1 & & -1 & 2 \end{pmatrix}$$



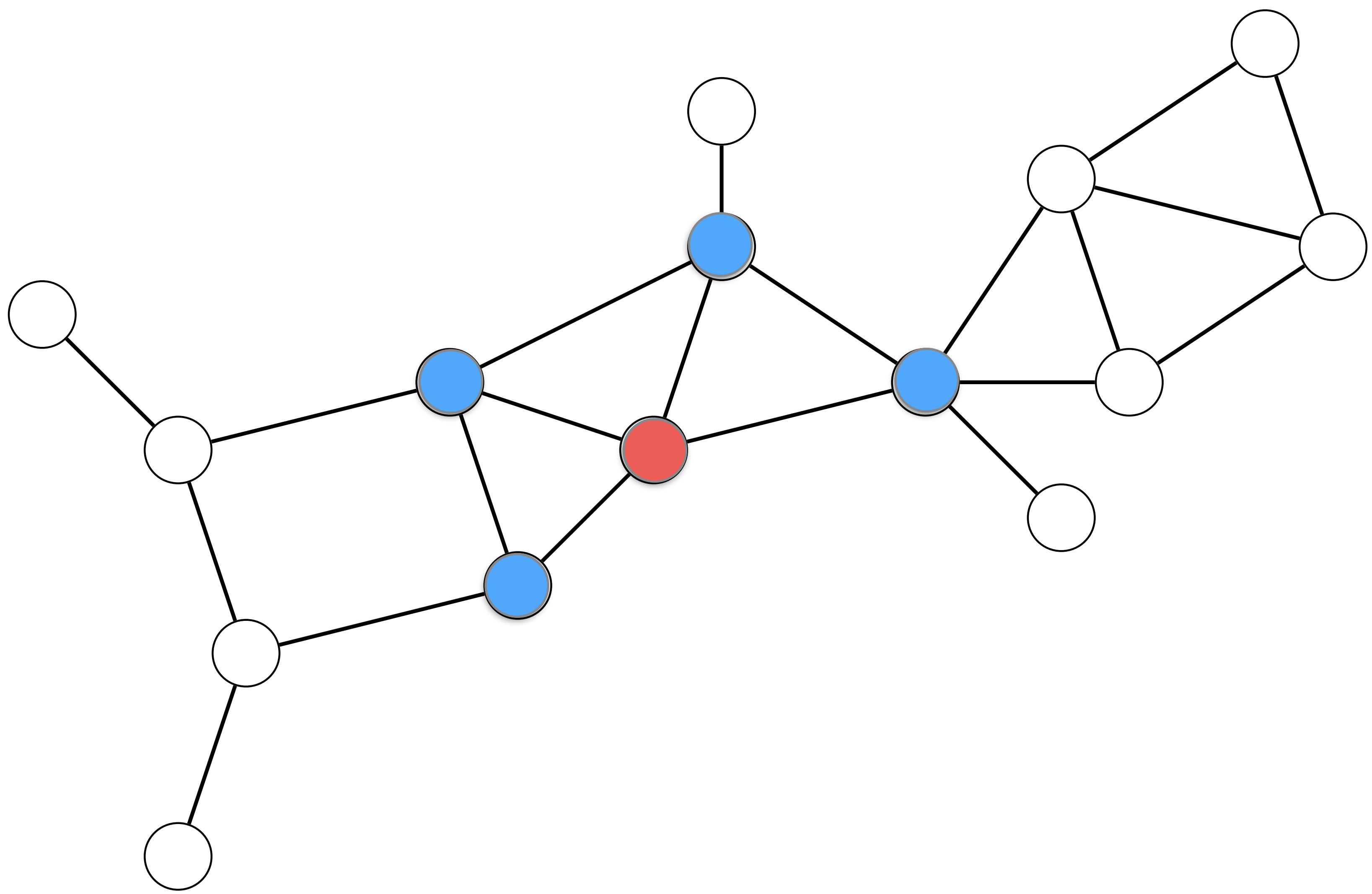
[Gärtner, 2002] [Vishwanathan et al, 2010]

3. Propagation kernels



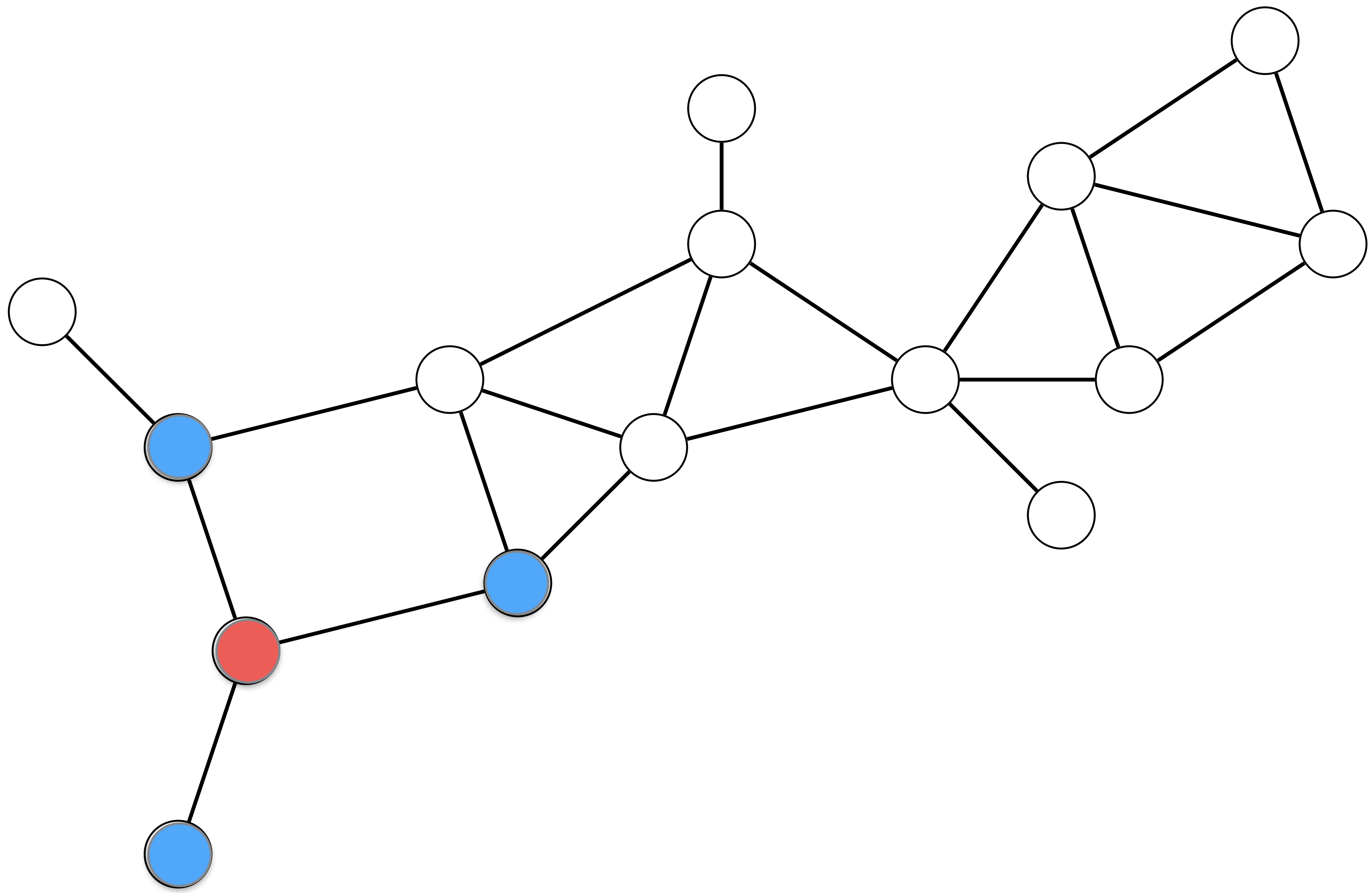
Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]

3. Propagation kernels



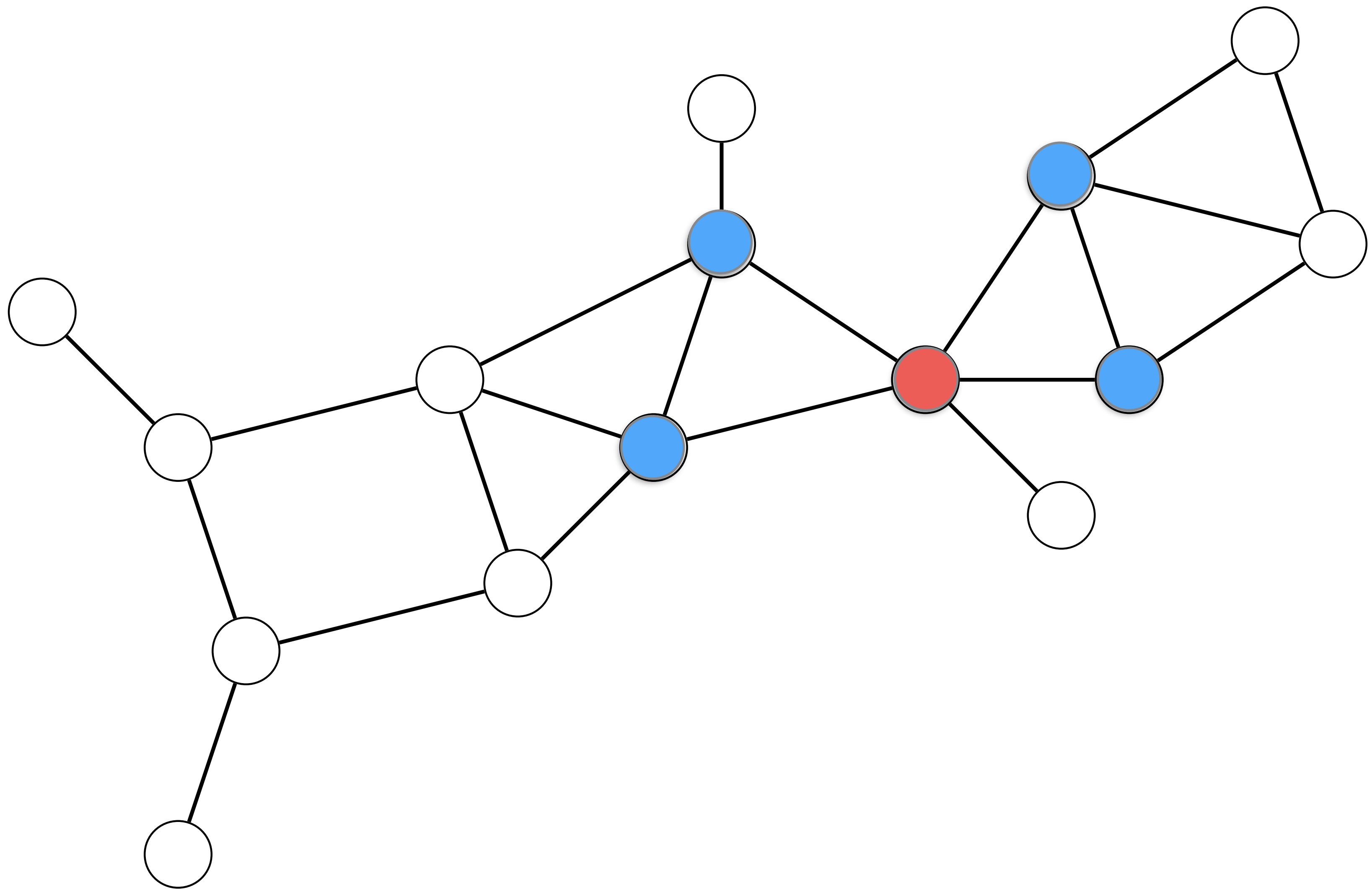
Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]

3. Propagation kernels



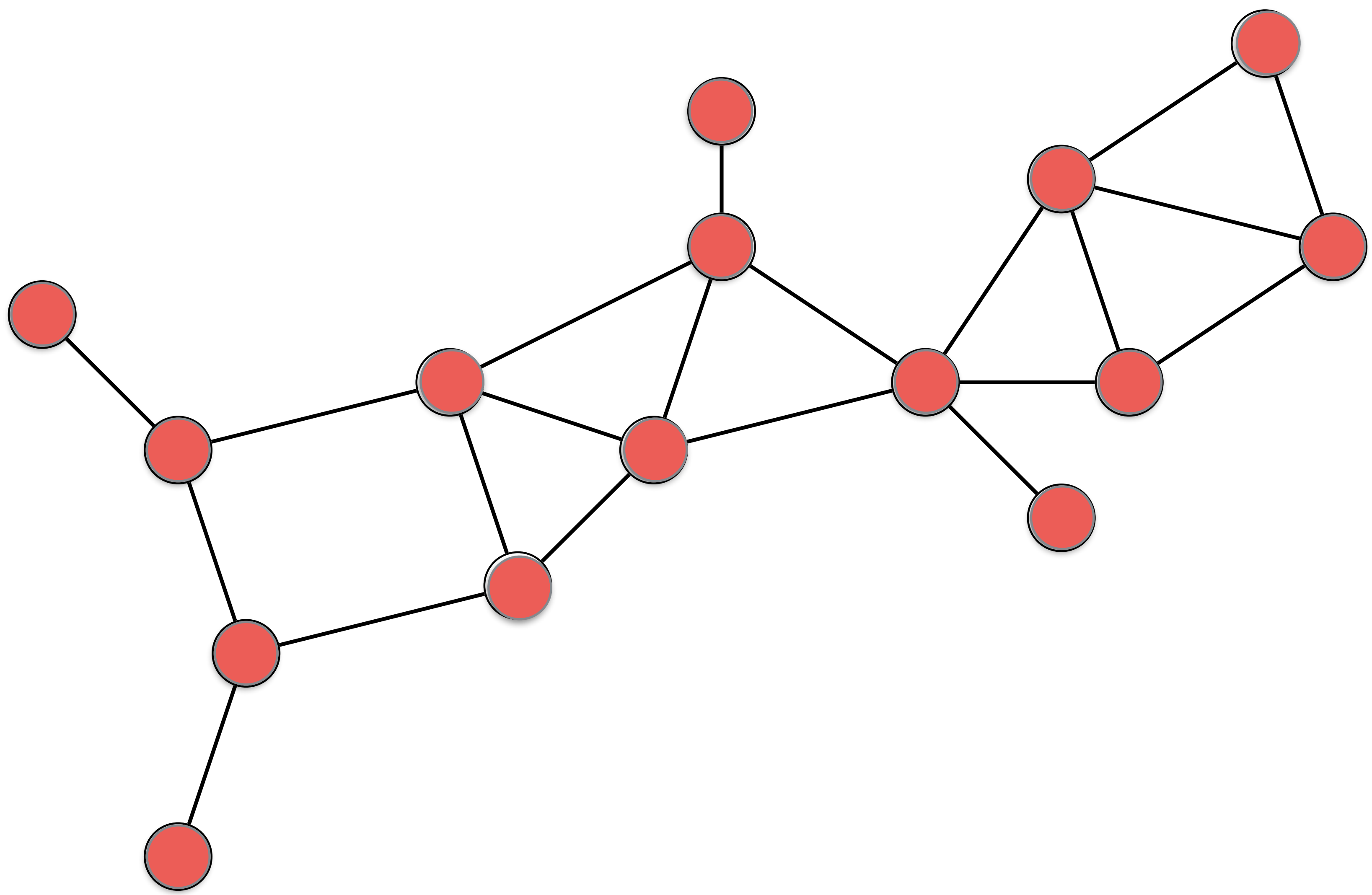
Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]

3. Propagation kernels



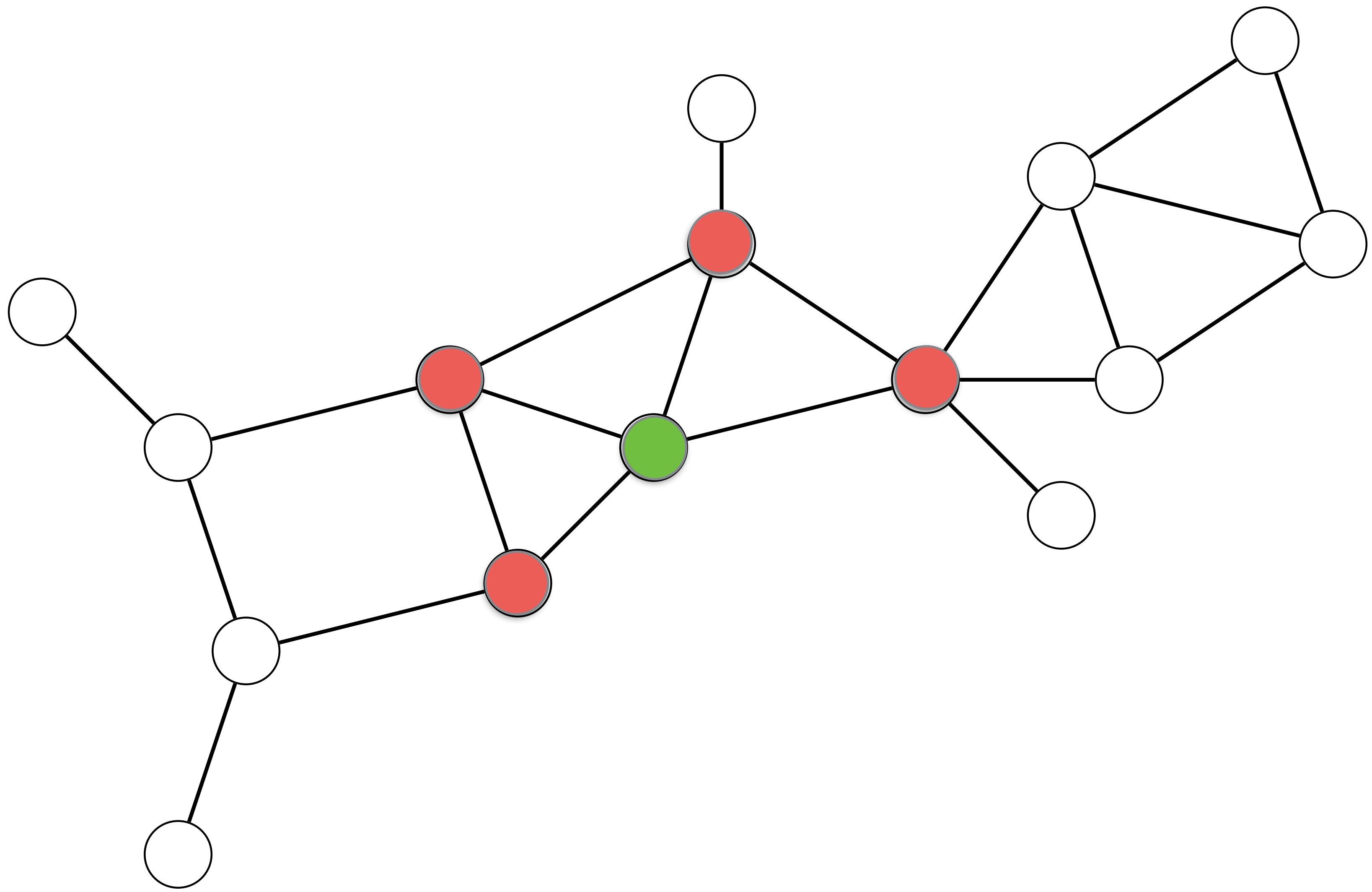
Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]

3. Propagation kernels



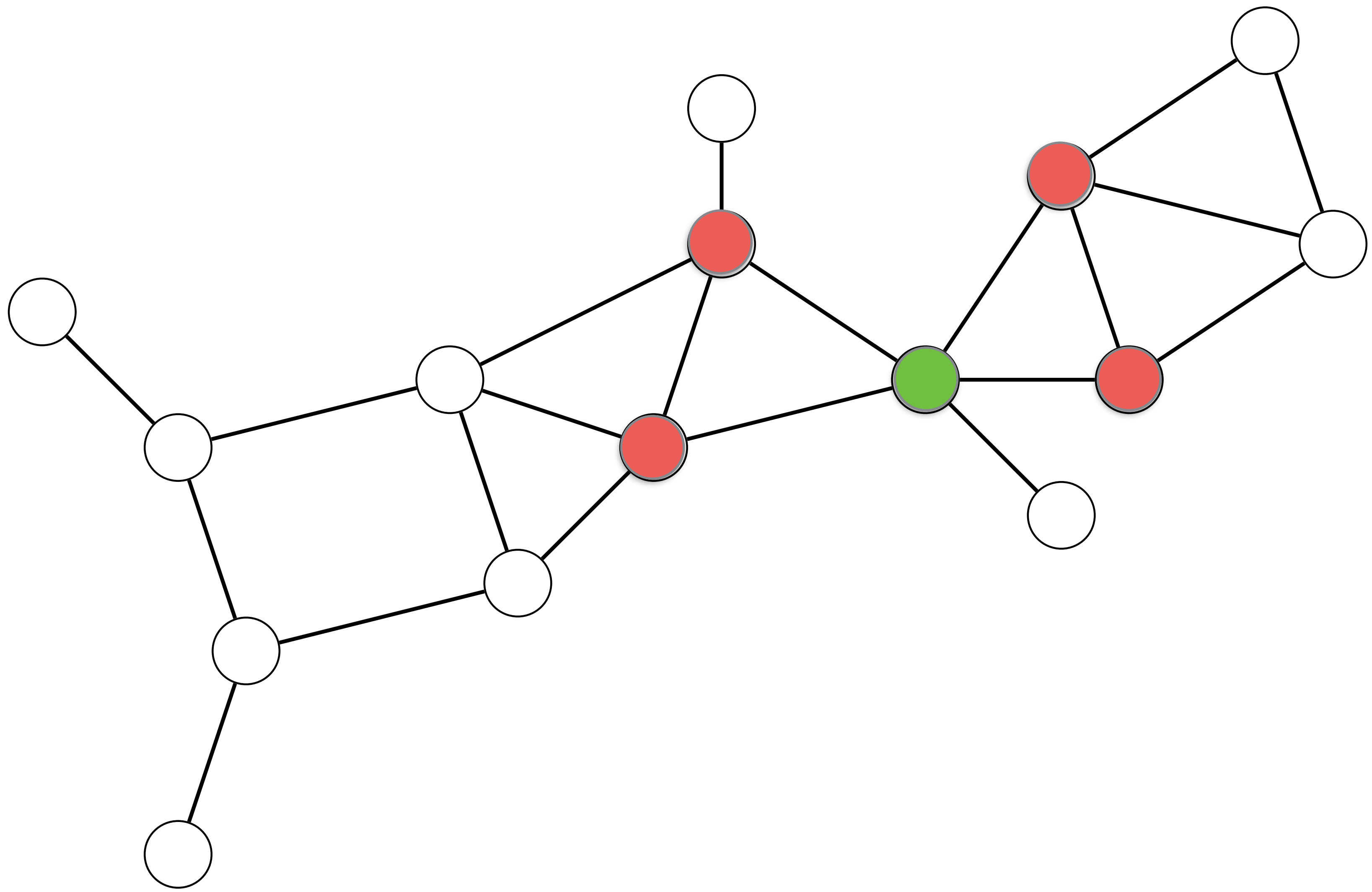
Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]

3. Propagation kernels



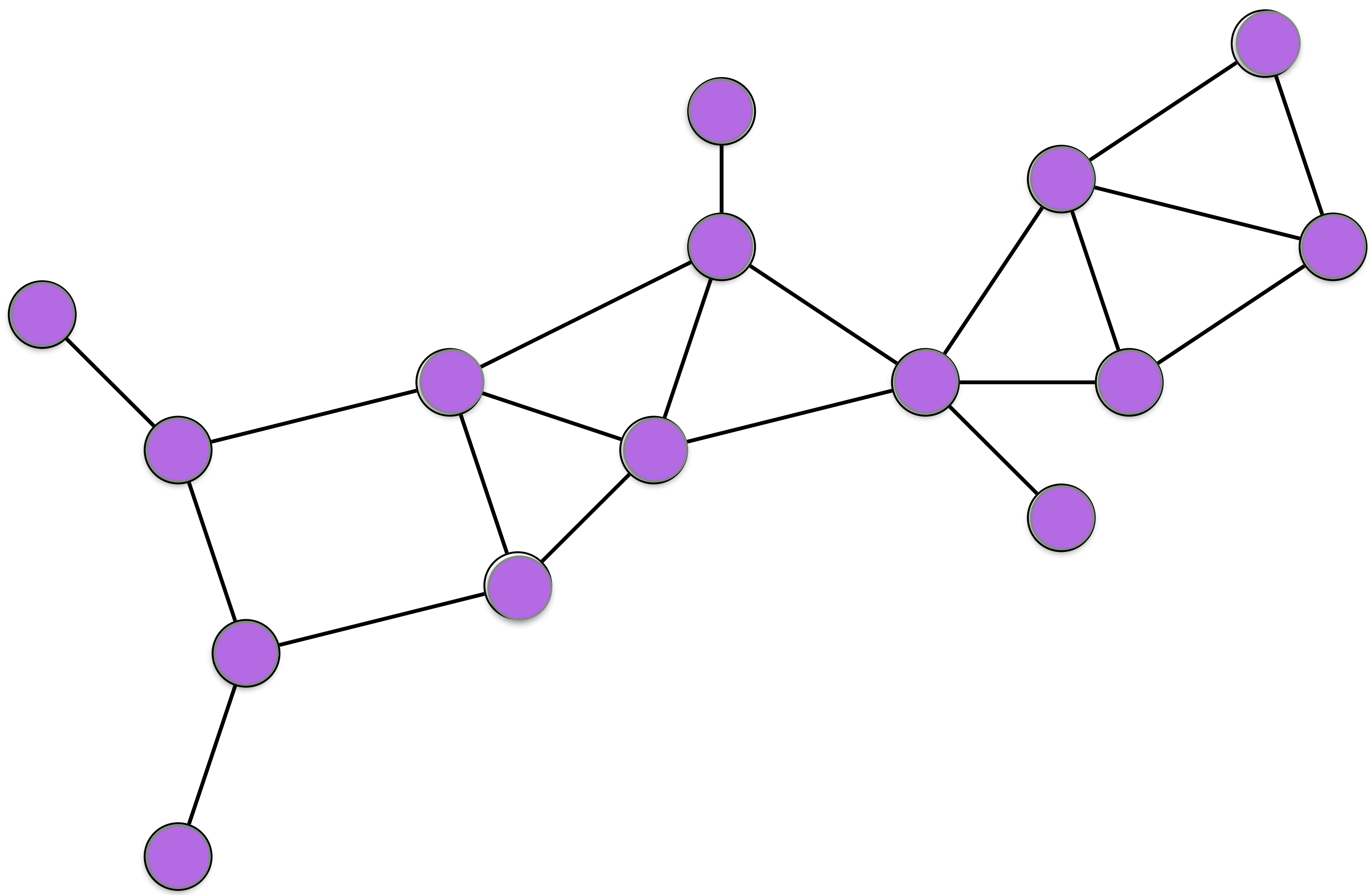
Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]

3. Propagation kernels



Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]

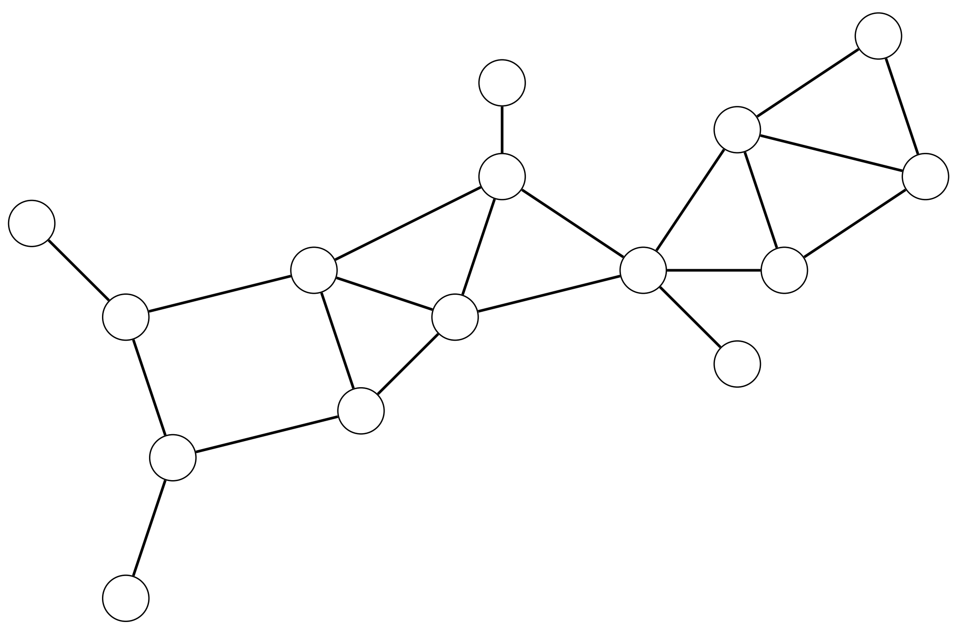
3. Propagation kernels

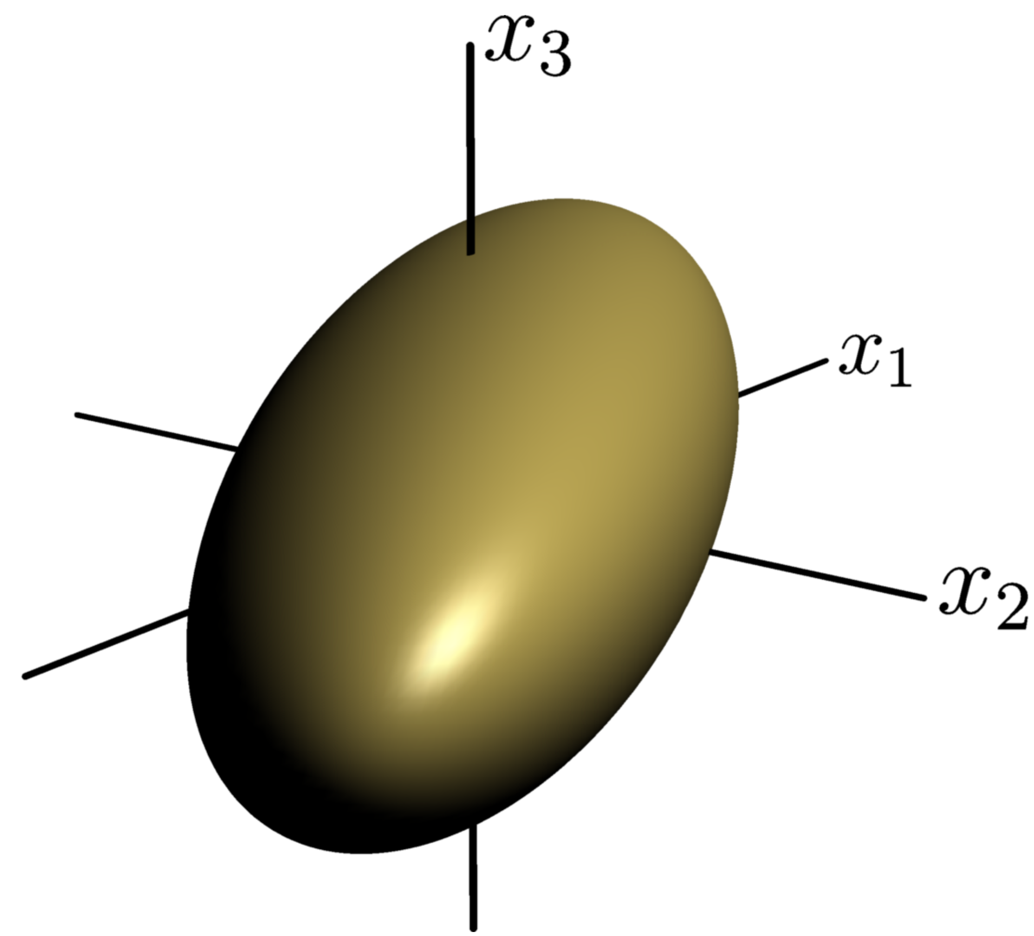


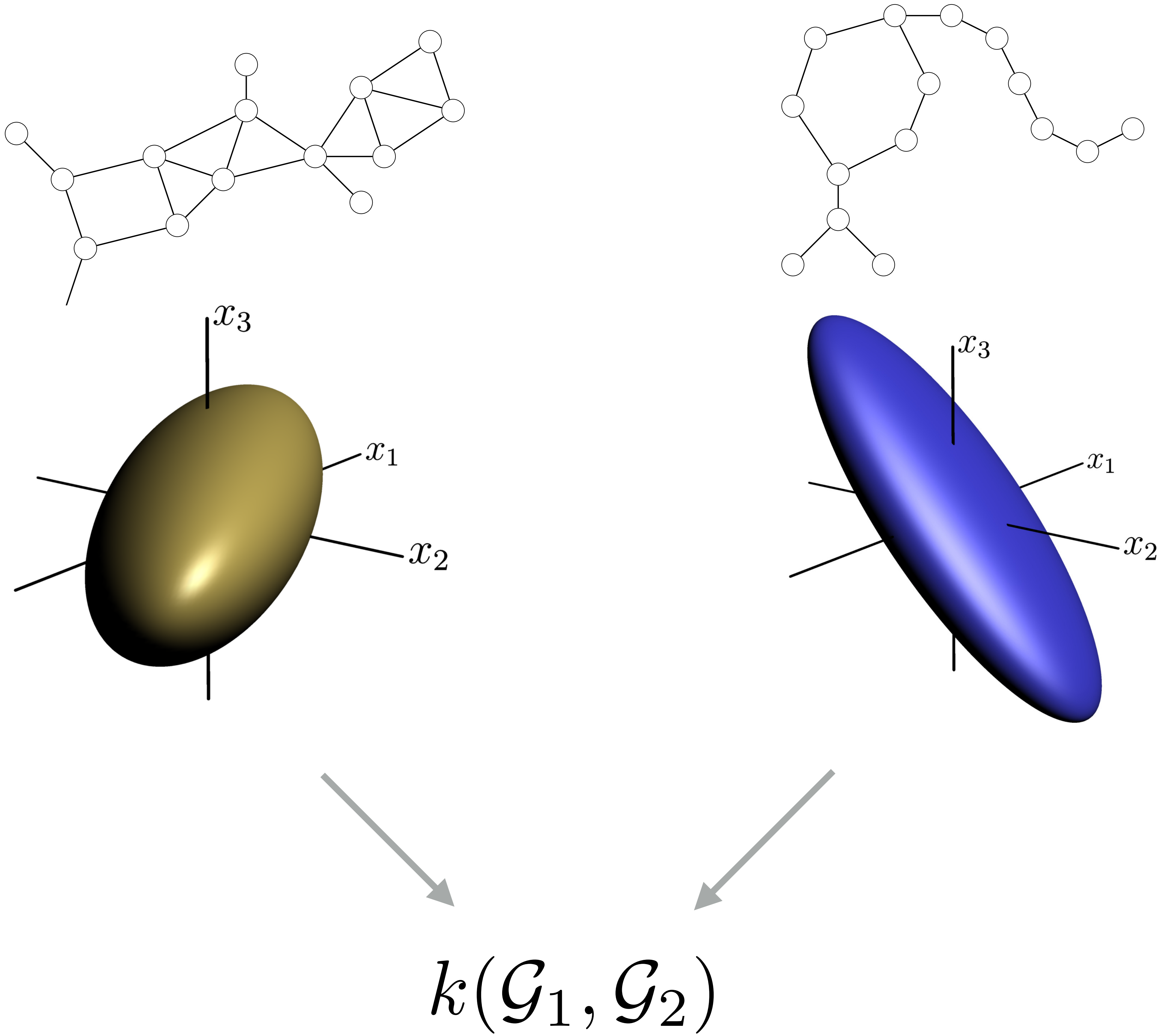
Weisfeiler—Lehmann kernels [Shervashidze et al., 2011]
Propagation kernels [Neumann et al., 2016]

The Laplacian Graph Kernel

The graph ellipsoid



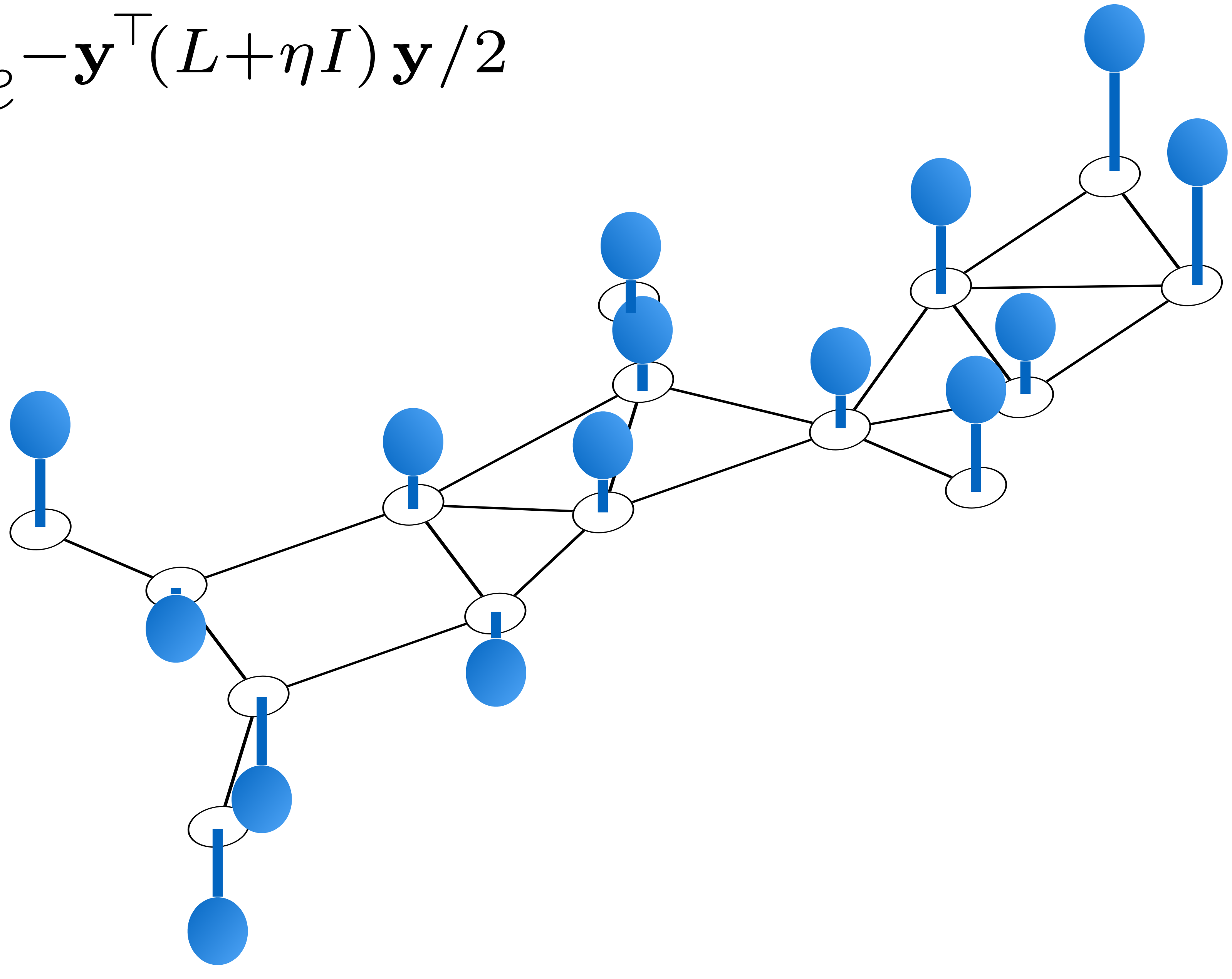




x_3
x_1
x_2
x_3
x_1
x_2
$k(\mathcal{G}_1, \mathcal{G}_2)$

$$k_{\mathrm{LG}}(\mathcal{G}_1, \mathcal{G}_2) = \frac{\left| \left(\frac{1}{2} L_1 + \frac{1}{2} L_2 \right)^{-1} \right|^{1/2}}{\left| L_1^{-1} \right|^{1/4} \left| L_2^{-1} \right|^{1/4}}$$

$$p(\mathbf{y}) \propto e^{-\mathbf{y}^\top (L+\eta I)\,\mathbf{y}/2}$$



Bhattacharyya kernel:

$$k(p_1, p_2) = \int \sqrt{p_1(x)}\sqrt{p_2(x)}\, dx,$$

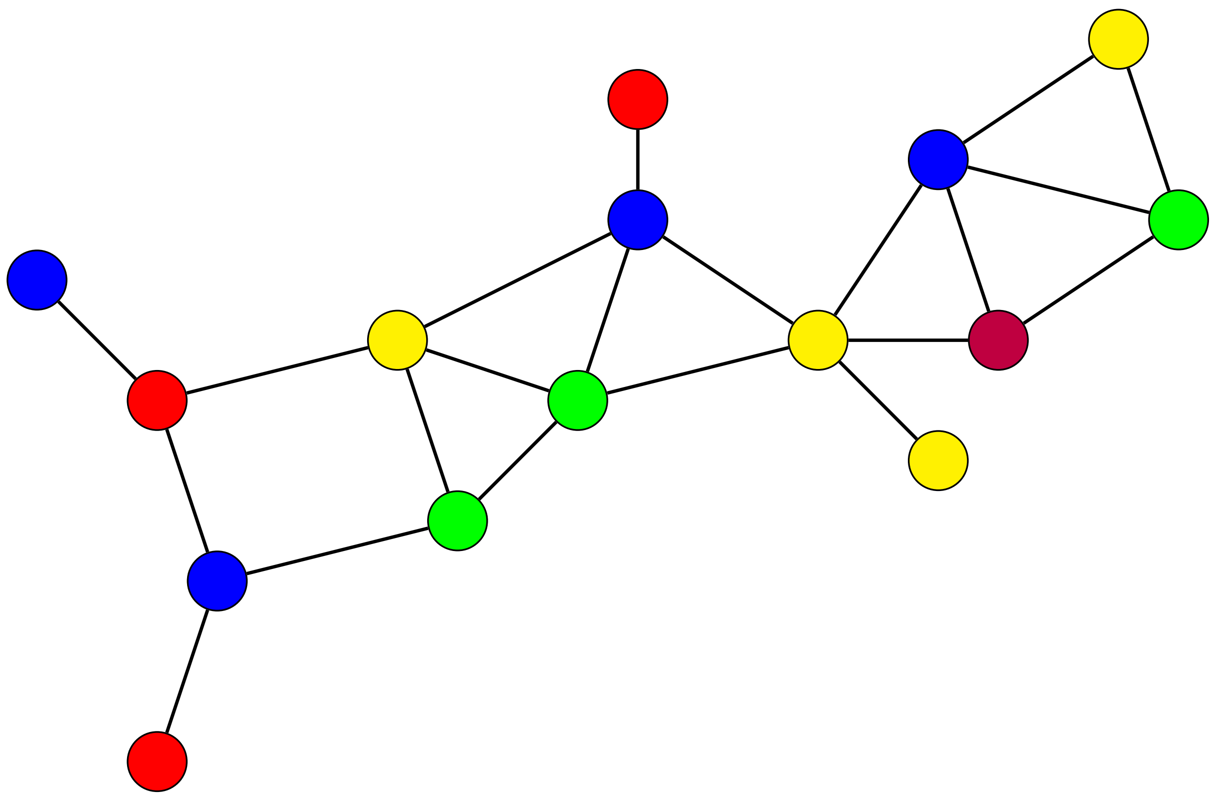
$$k_{\mathrm{LG}}(\mathcal{G}_1, \mathcal{G}_2) = k(p_1, p_2) = \frac{\left| \left(\frac{1}{2} S_1^{-1} + \frac{1}{2} S_2^{-1} \right)^{-1} \right|^{1/2}}{|S_1|^{1/4}\, |S_2|^{1/4}}$$

[K. and Jebara, 2003]

But the LG kernel is not relabeling invariant!

Transformation of variables:

$$\mathbf{z} := U\,\mathbf{y} \qquad\qquad \Sigma_{\mathbf{z}} = U\,\Sigma_{\mathbf{y}} U^{\top}$$

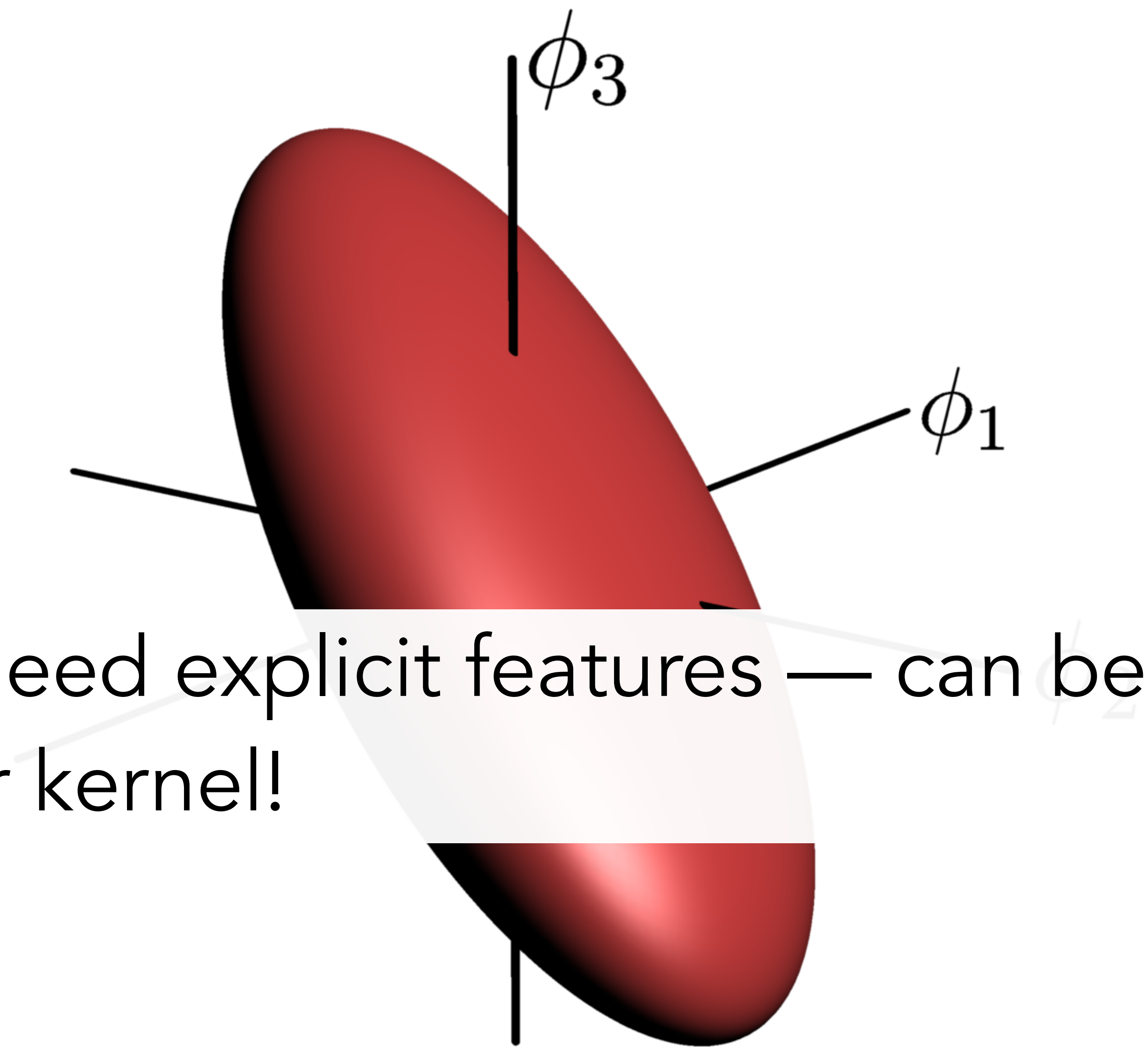


$U_{*,i}$ are the “features” of node i.

The **Feature space Laplacian graph kernel (FLG kernel)** is defined

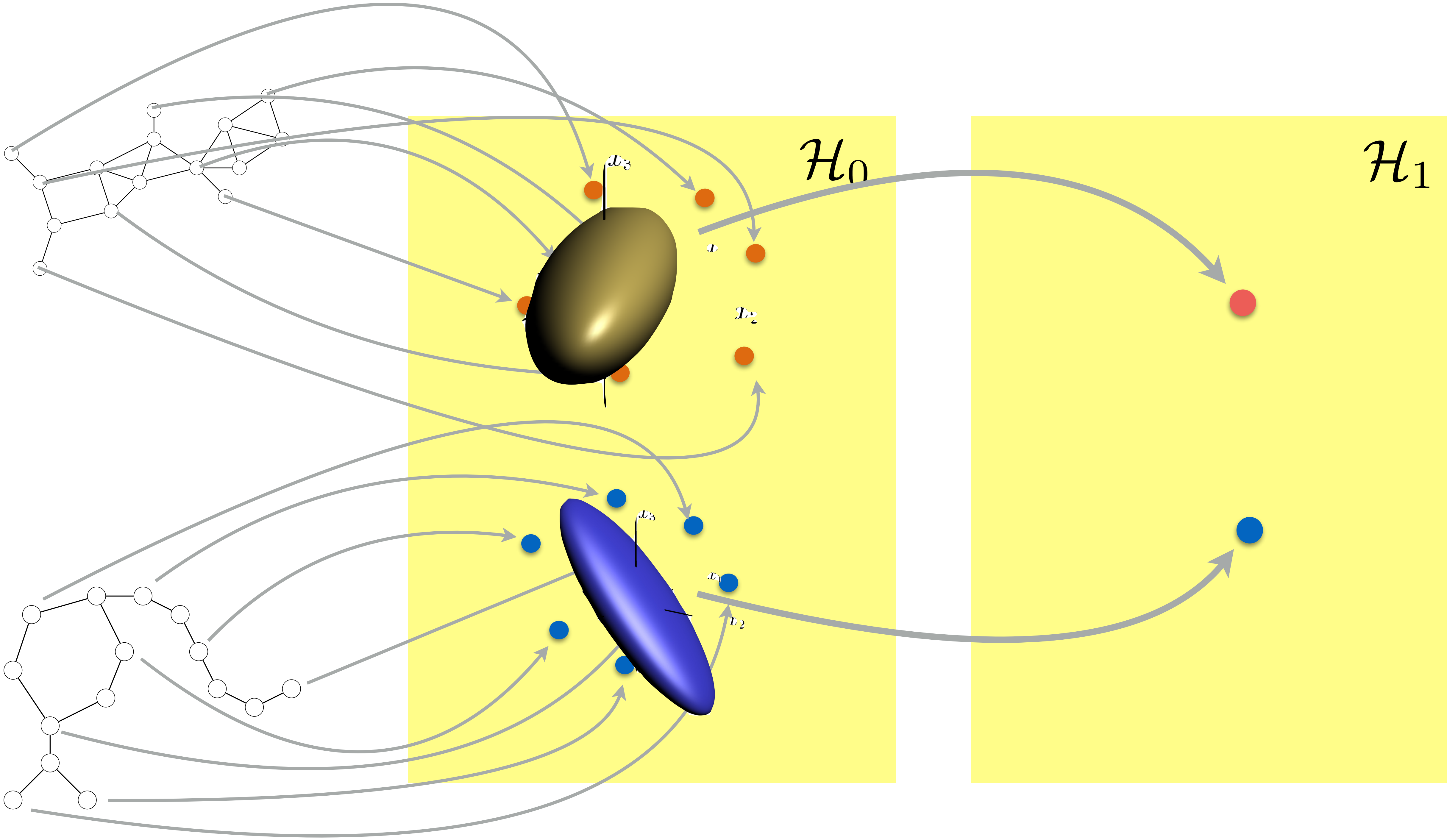
$$k_{\mathrm{FLG}}(\mathcal{G}_1, \mathcal{G}_2) = \frac{\left| \left(\frac{1}{2} S_1^{-1} + \frac{1}{2} S_2^{-1} \right)^{-1} \right|^{1/2}}{|S_1|^{1/4} \, |S_2|^{1/4}} \, ,$$

where $S_1 = U_1 L_1^{-1} U_1^{\top} + \gamma I$ and $S_2 = U_2 L_2^{-1} U_2^{\top} + \gamma I$.



Don't even need explicit features — can be induced from another kernel!

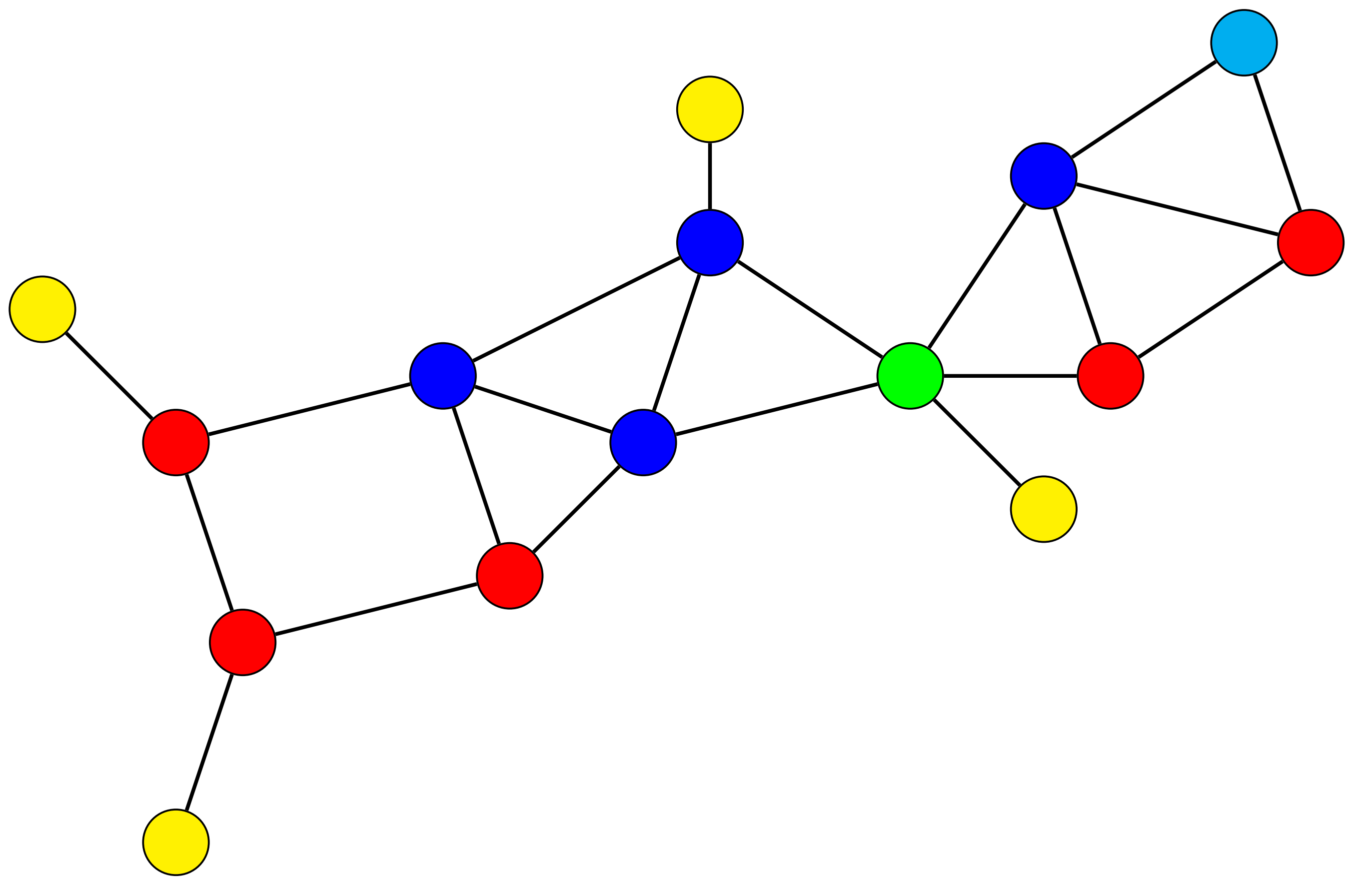
The ellipsoid is now in feature space and **combines** information about the graph structure with the features.



$\mathcal{H}_0$
$\mathcal{H}_1$

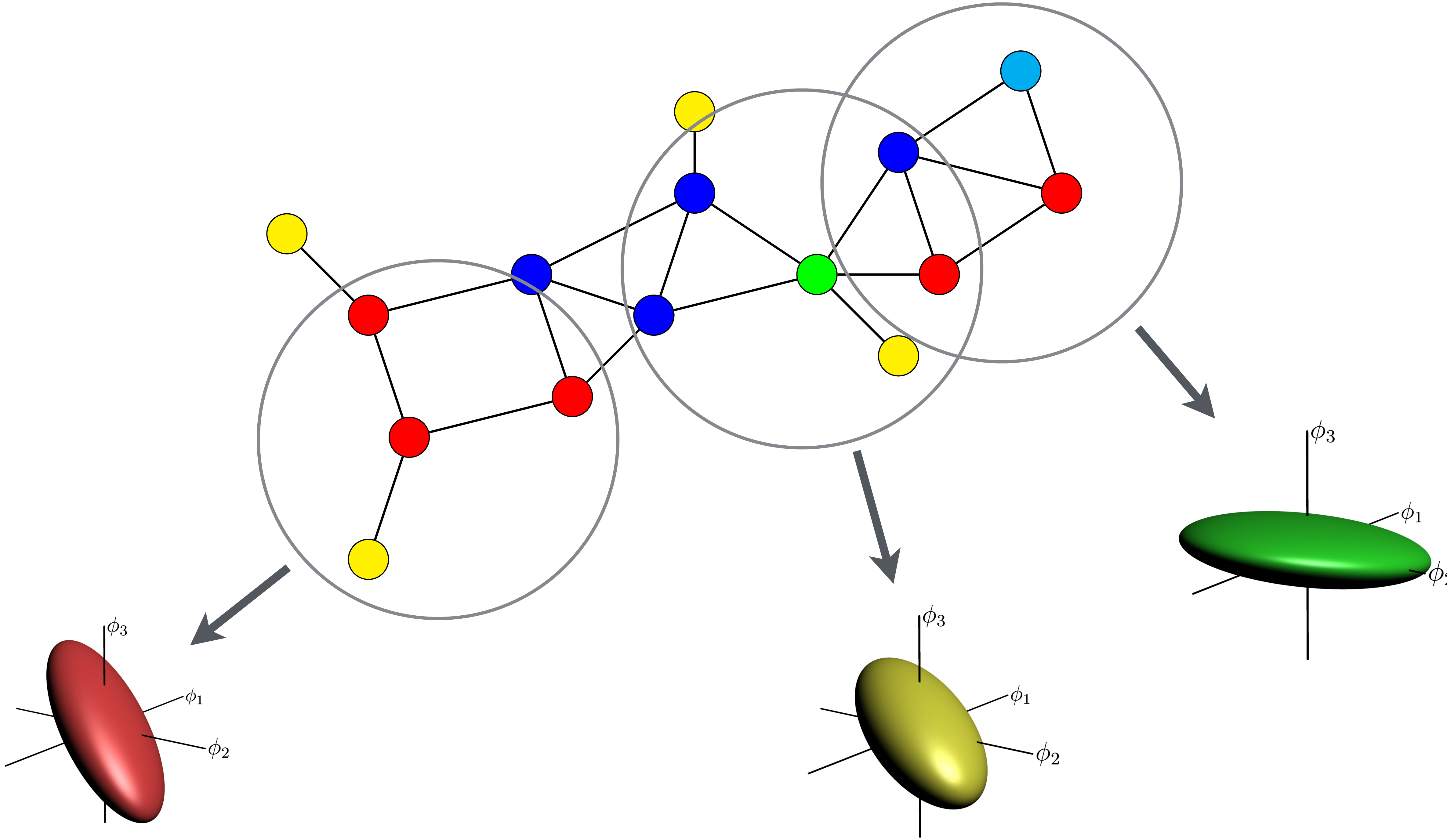
The **Multiscale** Laplacian Graph Kernel

1. Base features

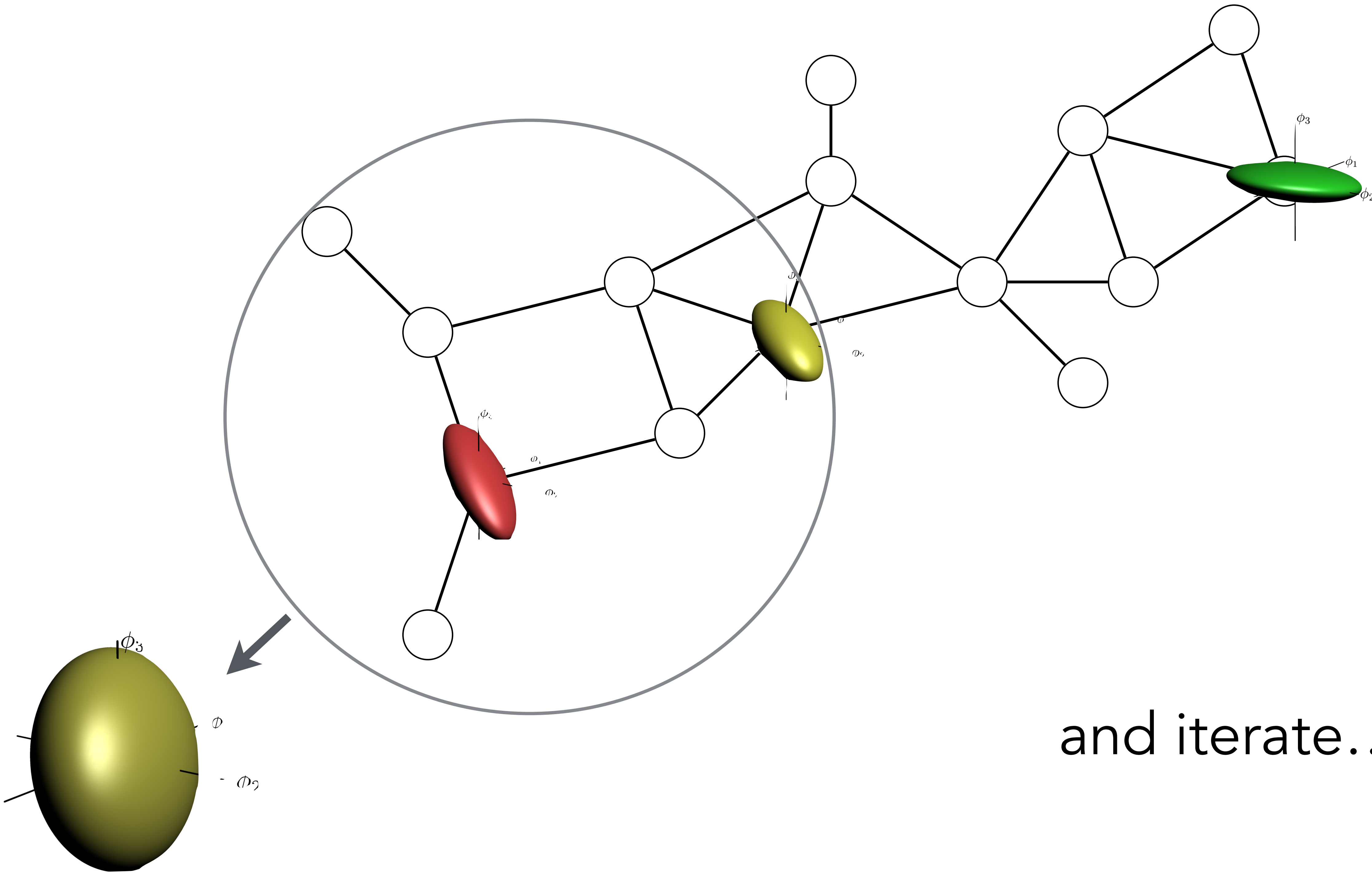


Start with something simple like node degrees.

2. Small subgraphs…

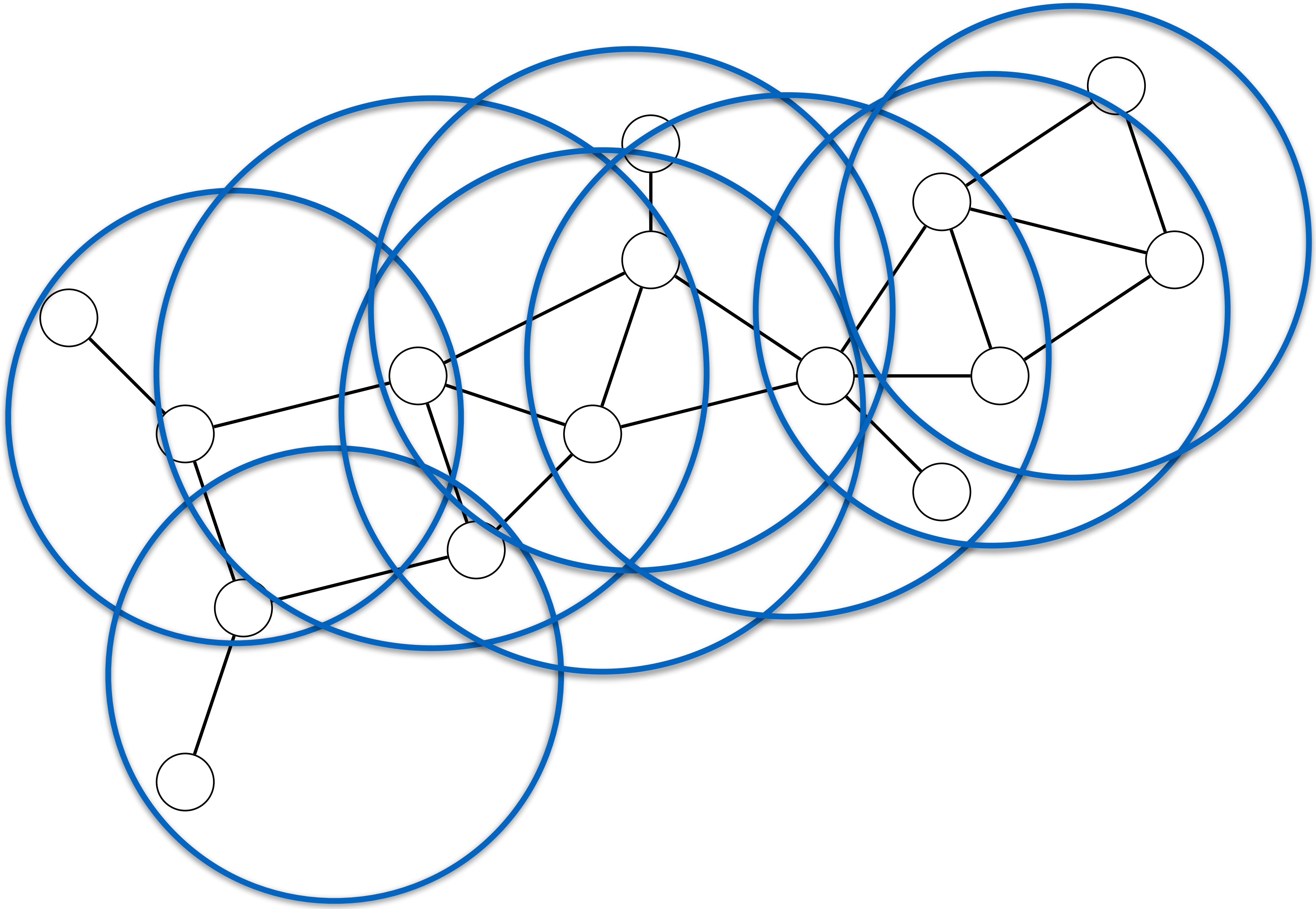


3. Larger subgraphs…

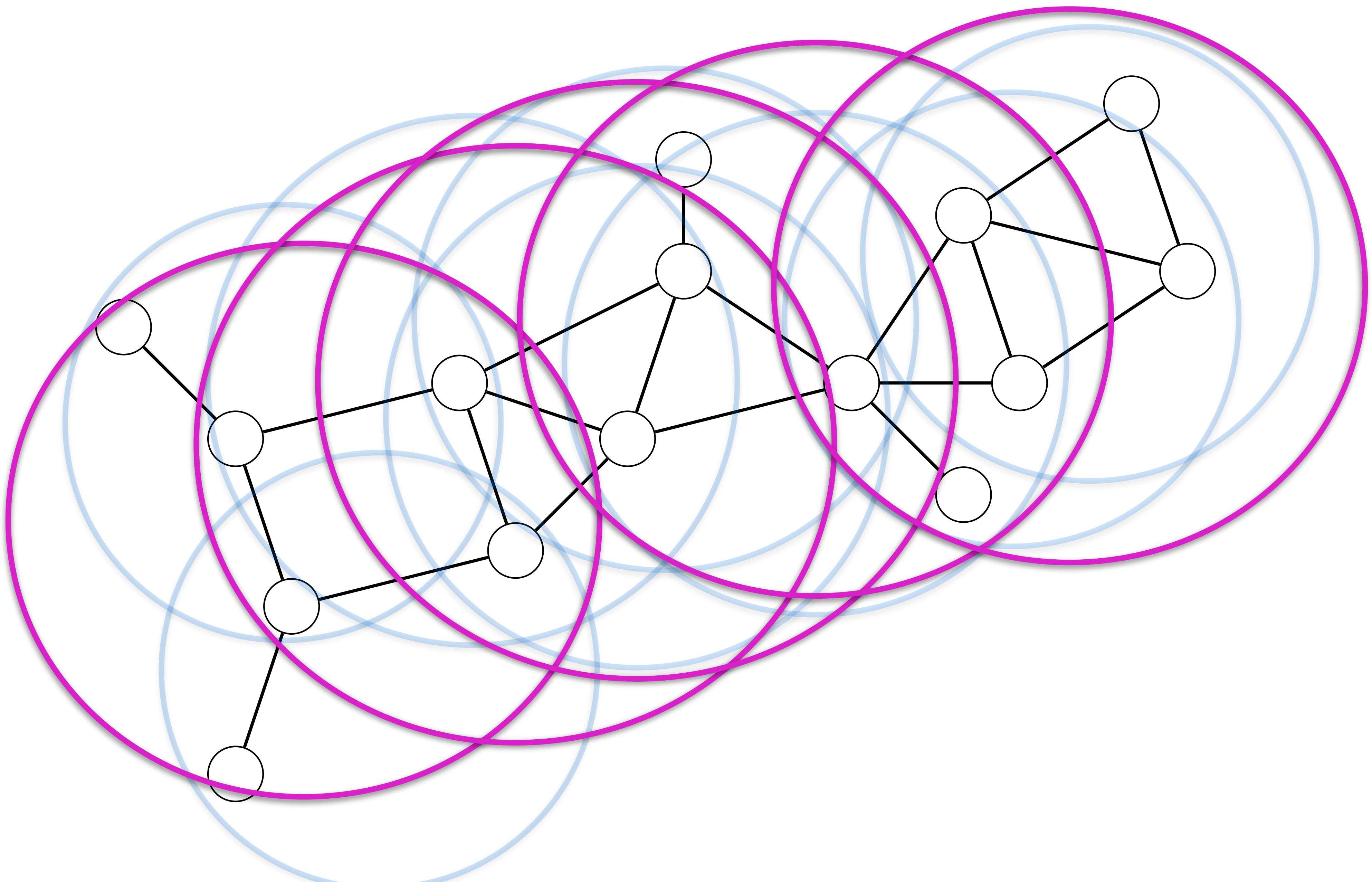


and iterate…

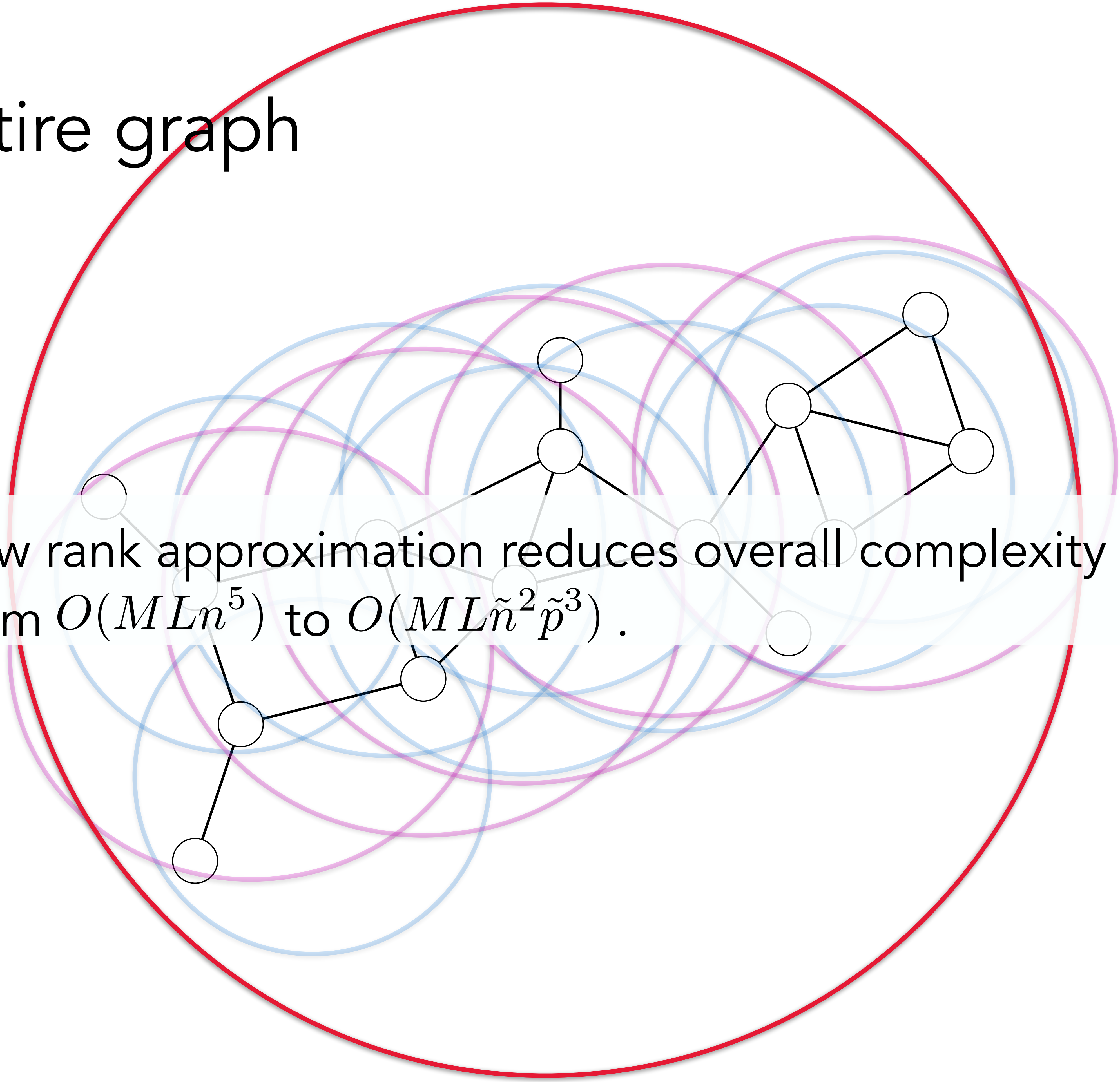
Level 1 subgraphs



Level 2 subgraphs



Entire graph



Low rank approximation reduces overall complexity from $O(MLn^5)$ to $O(ML\tilde{n}^2\tilde{p}^3)$.

Let $\mathcal{G}$ be a graph with vertex set V, and κ a positive semi-definite kernel on V. Assume that for each $v \in V$ we have a nested sequence of L neighborhoods

$$v \in N_1(v) \subseteq N_2(v) \subseteq \ldots \subseteq N_L(v) \subseteq V,$$

and for each $N_\ell(v)$, let $G_\ell(v)$ be the corresponding induced subgraph of $\mathcal{G}$. We define the **Multiscale Laplacian Subgraph Kernels (MLS kernels)**, $\mathfrak{K}_1, \ldots, \mathfrak{K}_L \colon V \times V \to \mathbb{R}$ as follows:

1. $\mathfrak{K}_1$ is just the FLG kernel $k_{\mathrm{FLG}}^{\kappa}$ induced from the base kernel κ between the lowest level subgraphs:

$$\mathfrak{K}_1(v, v') = k_{\mathrm{FLG}}^{\kappa}(G_1(v), G_1(v')).$$

2. For $\ell = 2, 3, \ldots, L$, $\mathfrak{K}_\ell$ is the FLG kernel induced from $\mathfrak{K}_{\ell-1}$ between $G_\ell(v)$ and $G_\ell(v')$:

$$\mathfrak{K}_\ell(v, v') = k_{\mathrm{FLG}}^{\mathfrak{K}_{\ell-1}}(G_\ell(v), G_\ell(v')).$$

We define the **Multiscale Laplacian Graph Kernel (MLG kernel)** between any two graphs $\mathcal{G}_1, \mathcal{G}_2 \in \mathfrak{G}$ as

$$\mathfrak{K}(\mathcal{G}_1, \mathcal{G}_2) = k_{\mathrm{FLG}}^{\mathfrak{K}_L}(\mathcal{G}_1, \mathcal{G}_2).$$

1. True multiscale/multiresolution graph kernel.
2. Combines information from subgraphs with relative position of subgraphs.
3. Invariant to relabeling.
4. Can compare graphs of different sizes.
5. “Smooth” w.r.t. perturbations.
6. Needs further tricks for efficient computation.

Method	MUTAG	PTC	ENZYMES	PROTEINS	NCI1	NCI109
WL	84.50(±2.16)	59.97(±1.60)	53.75(±1.37)	75.43(±1.95)	**84.76(±0.32)**	**85.12(±0.29)**
WL-Edge	82.94(±2.33)	60.18(±2.19)	52.00(±0.72)	73.63(±2.12)	**84.65(±0.25)**	**85.32(±0.34)**
SP	**85.50(±2.50)**	59.53(±1.71)	42.31(±1.37)	75.61(±0.45)	73.61(±0.36)	73.23(±0.26)
Graphlet	82.44(±1.29)	55.88(±0.31)	30.95(±0.73)	71.63(±0.33)	62.40(±0.27)	62.35(±0.28)
p–RW	80.33(±1.35)	59.85(±0.95)	28.17(±0.76)	71.67(±0.78)	TIMED OUT	TIMED OUT
MLG	84.21(±2.61)	**63.62(±4.69)**	**57.92(±5.39)**	**76.14(±1.95)**	80.83(±1.29)	81.30(±0.80)

Code at github.com/horacepan/MLGkernel

Conclusions

- Truly multiscale kernel: subgraphs compared by comparing their constituent sub-subgraphs
- Ideas extend beyond just kernels world, e.g., hierarchical deep learning architectures; `structure2vec` [Dai, Dai & Song, 2016].

Support: DARPA D16AP00112 YFA "Multiresolution Machine Learning for Molecular Modeling"