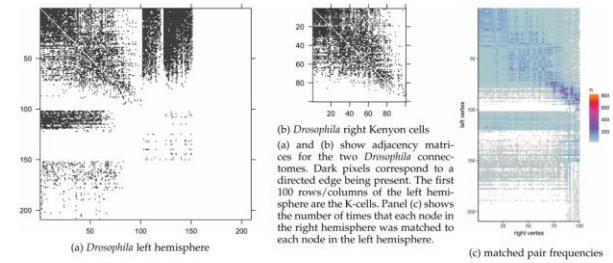
subgraph2vec

Article	Data	Data	Comments or Questions
Narayanan A , Chandramohan M , Chen L , et al. subgraph2vec: Learning Distributed Representations of Rooted Sub-graphs from Large Graphs[J]. 2016.	**Aim:** Learning latent representations of rooted subgraphs from large graphs. **Contributions:** ● Propose subgraph2vec. ● Develop a modified version of the skipgram language model. ● Discuss how subgraph2vec's representation learning technique would help to build the deep learning variant of WL kernel. **Applications:** ● graph classification and clustering tasks ● code clone detection ● malware detection	**Key results:** ● Benchmark Datasets Table 3: Average Accuracy (± std dev.) for subgraph2vec and state-of-the-art graph kernels on benchmark graph classification datasets ● Clone Detection Table 5: Clone Detection - Results ● Malware Detection Table 7: Malware Detection - Results **Background:** ● language model **Methods:** ● Generate rooted subgraphs: WL Relabeling Process ● Embeddings of those subgraphs: Radial Skip Gram Model ● Approximate the probability distribution: Negative Sampling ➢ $\textbf{for } \textit{each } sg_{cont} \in context_v^{(d)} \textbf{ do}$ $J(\Phi) = -\log \Pr\left(sg_{cont} | \Phi(sg_v^{(d)})\right)$	● When subgraph of degree 0 are considered, subgraph2vec provides node embeddings. ● How can we get the initial labels for each node? ☑ ● What is the Deep WL kernel? ☑

Dataset	MUTAG	PTC	PROTEINS	NCI1	NCI109
WL [6]	80.63 ± 3.07	56.91 ± 2.79	72.92 ± 0.56	80.01 ± 0.50	80.12 ± 0.34
Deep WL_{YV} [7]	82.95 ± 1.96	59.04 ± 1.09	73.30 ± 0.82	**80.31** ± 0.46	**80.32** ± 0.33
subgraph2vec	**87.17** ± 1.72	**60.11** ± 1.21	**73.38** ± 1.09	78.05 ± 1.15	78.39 ± 1.89

Kernel	WL [6]	Deep WL_{YV} [7]	subgraph2vec
Pre-training duration	-	421.7 s	409.28 s
ARI	0.67	0.71	**0.88**

Classifier	WL [6]	Deep WL_{YV} [7]	subgraph2vec
Pre-training duration	-	2631.17 s	2219.28 s
Accuracy	66.15	71.03	**74.48**

graph matching			
Article	Data	Data	Comments or Questions
Sussman Daniel, Park Youngser, Priebe Carey E, et al. Matched Filters for Noisy Induced Subgraph Detection.. 2019,	**Aim:** Finding the vertex correspondence between two noisy graphs with different number of vertices where the smaller graph is still large. **Contributions:** ● Proposed a number of padding methods to transform the noisy subgraph detection problem into a graph matching problem. **Applications:** ● determining whether a particular activity pattern is present in a social network ● detecting certain shapes in an image ● discovering motifs in brain networks	**Key results:** • Finding Kenyon Cells in a Drosophila Connectome  (a) *Drosophila* left hemisphere (b) *Drosophila* right Kenyon cells (a) and (b) show adjacency matrices for the two *Drosophila* connectomes. Dark pixels correspond to a directed edge being present. The first 100 rows/columns of the left hemisphere are the K-cells. Panel (c) shows the number of times that each node in the right hemisphere was matched to each node in the left hemisphere. (c) matched pair frequencies Fig. 5. Connectomes and matching analysis of the *Drosophila* connectomes. **Background:** • Subgraph Isomorphism • Graph Matching Algorithms • Statistical Models **Methods:** • Padding Approaches: Naive Padding, Centered Padding, Oracle Padding • Graph Matching Approaches: FAQ algorithm	• 三种padding方法没看明白。。