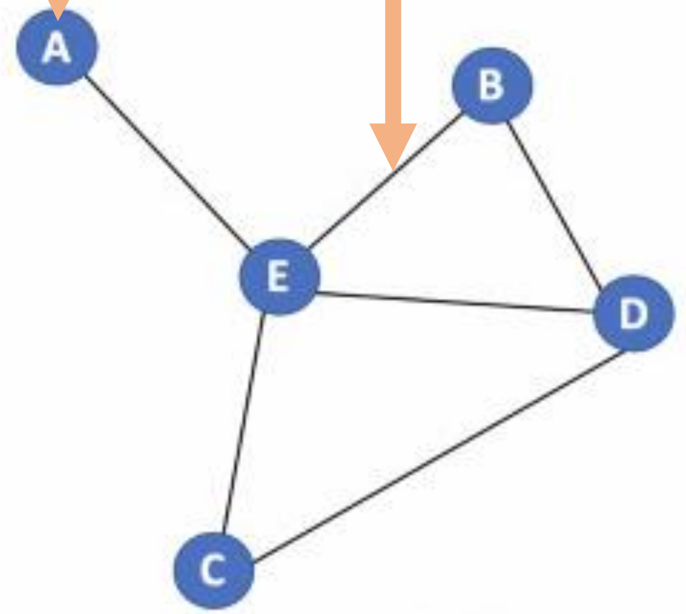
Graph Neural Network

What is Graph ?

Node

Edge



Graph G

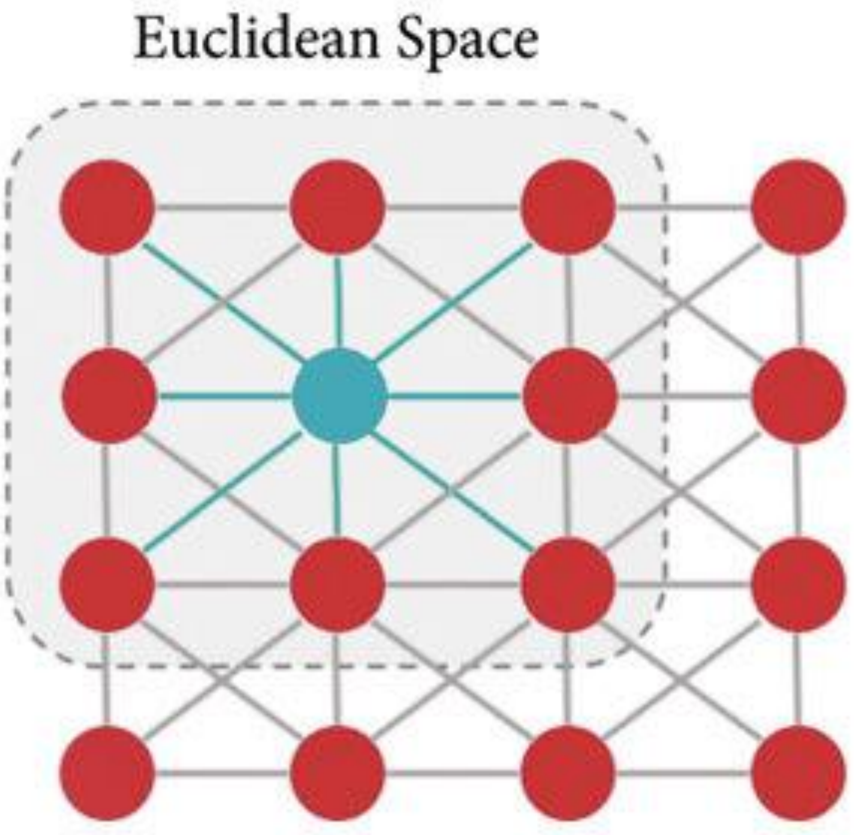
A	-1.1	3.2	4.2
B	0.4	5.1	-1.2
C	1.2	1.3	2.1
D	1.4	-1.2	2.5
E	1.4	2.5	4.5

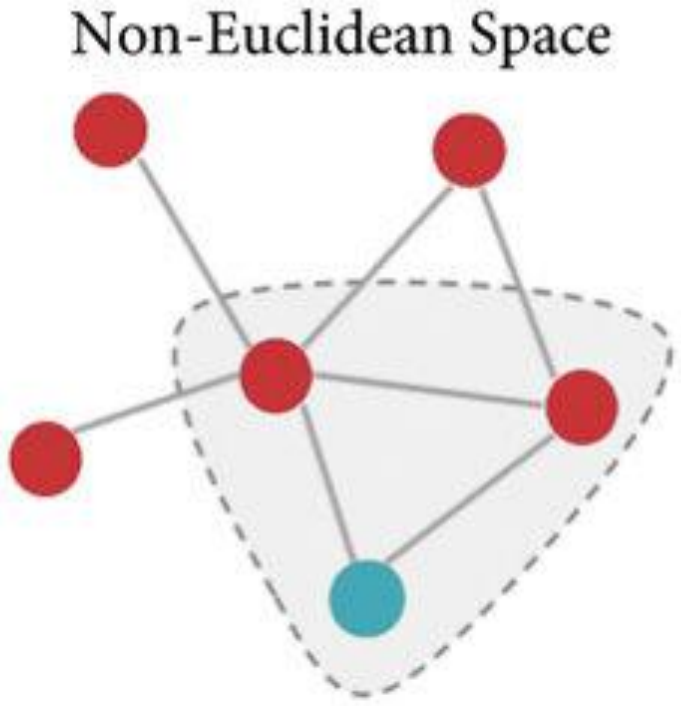
Node Feature

	A	B	C	D	E
A	0	0	0	0	1
B	0	0	0	1	1
C	0	0	0	1	1
D	0	1	1	0	1
E	1	1	1	1	0

Adjacency matrix A

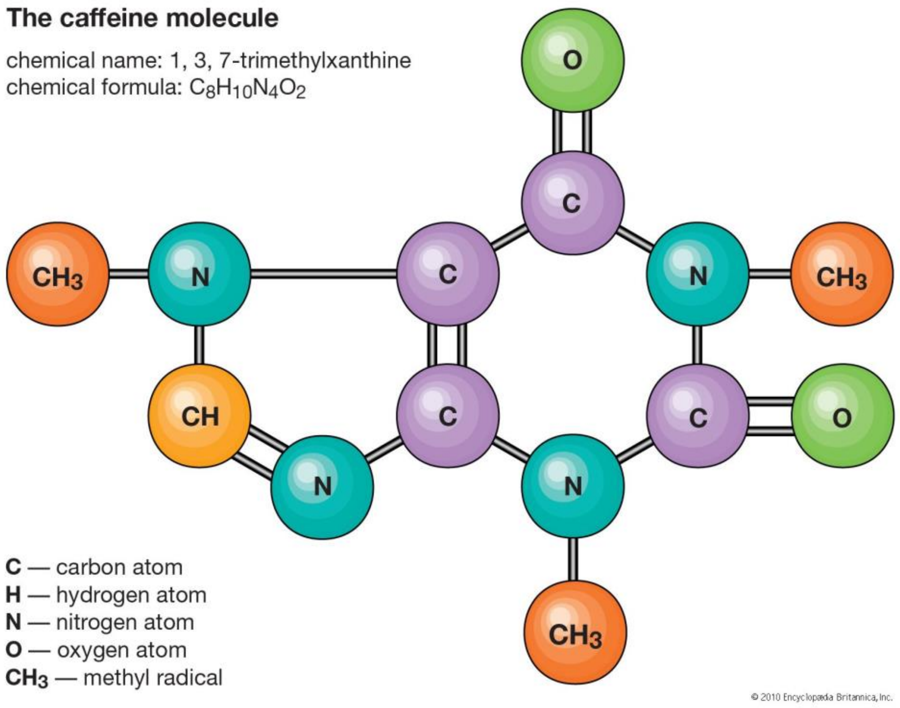
Why Graph is special?



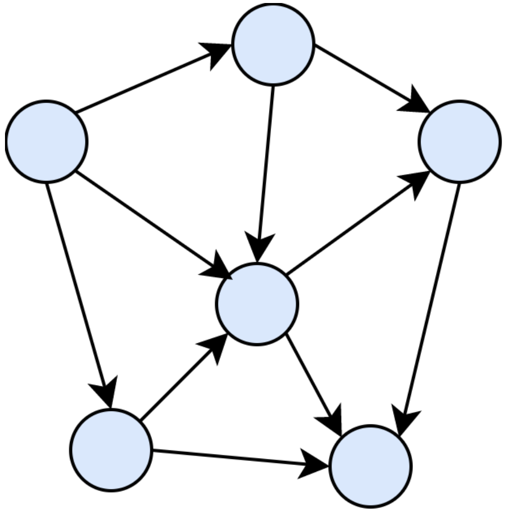


(Arbitrary structure)

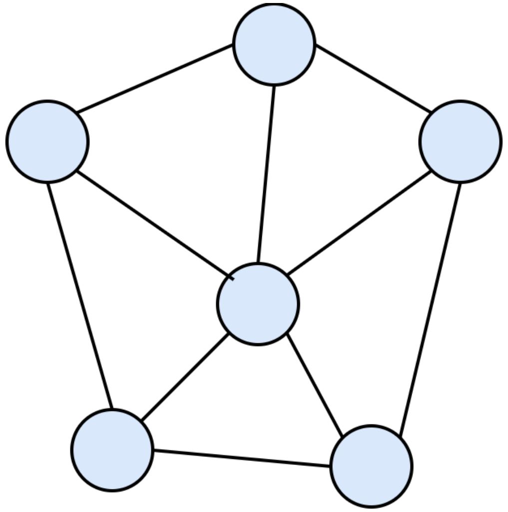
Example of graph



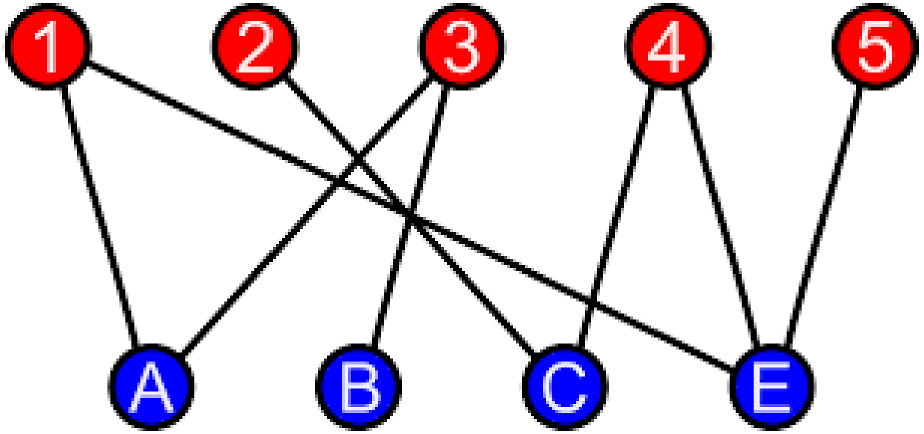
type of Graph



Directed Graph

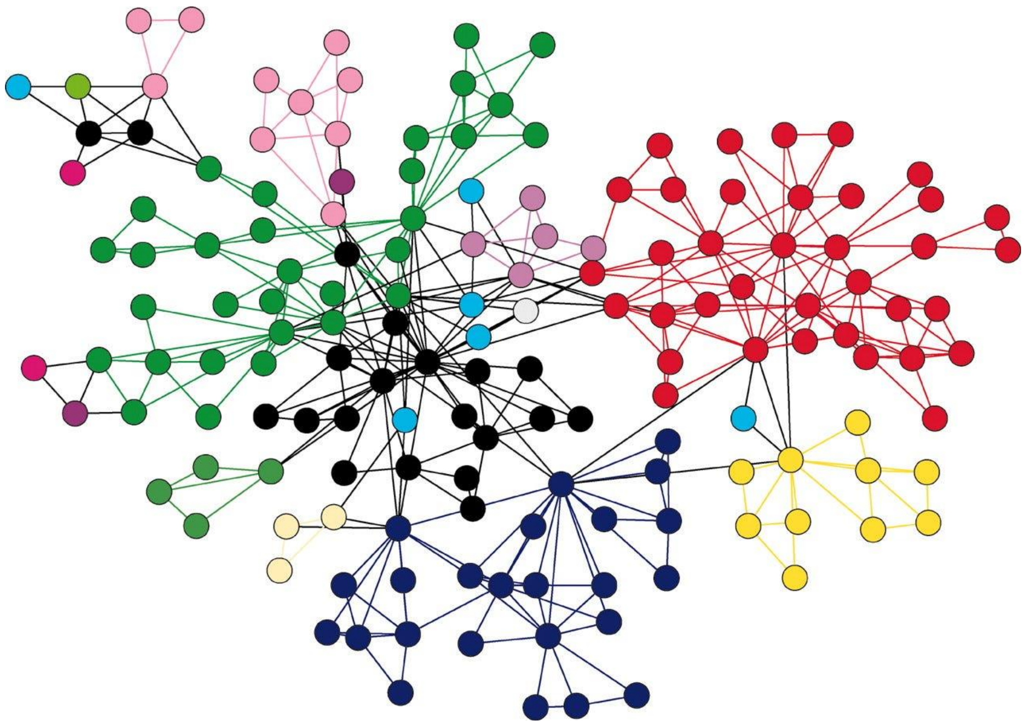


Undirected Graph

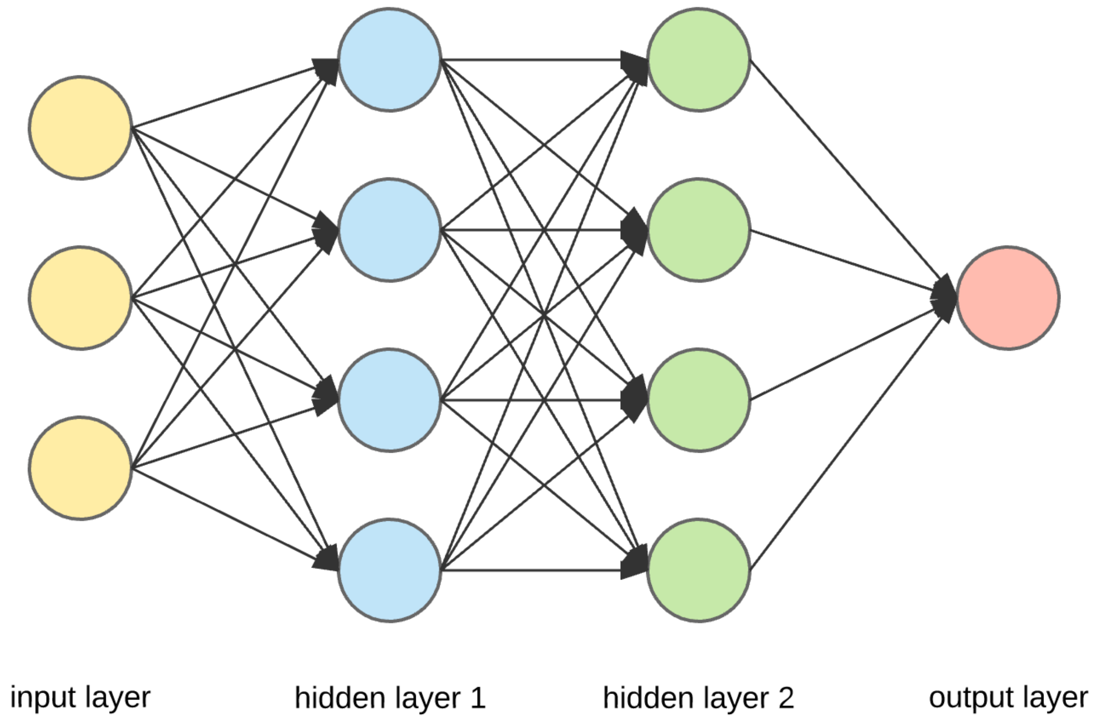


bipartite graph

Why must use graph neural network ?

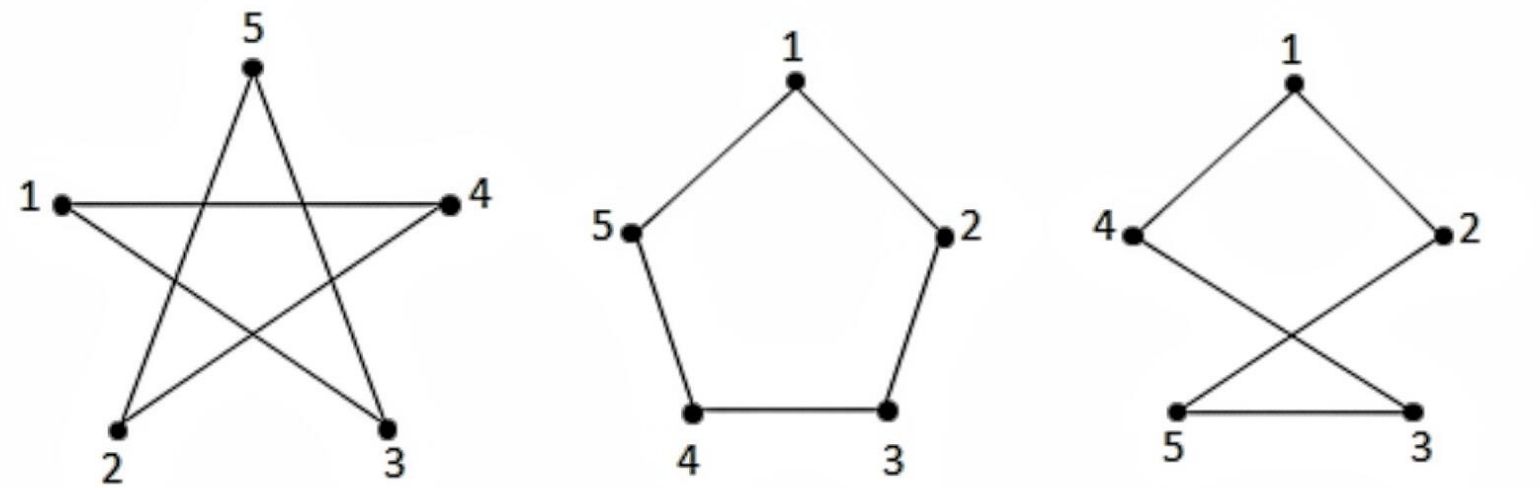


Graph is in complex domain



Graph structure not have a fixed size

Why must use graph neural network ?



	1	2	3	4	5
1			1	1	
2				1	1
3	1				1
4	1	1			
5		1	1		

(a)

	1	2	3	4	5
1		1			1
2	1		1		
3		1		1	
4			1		1
5	1			1	

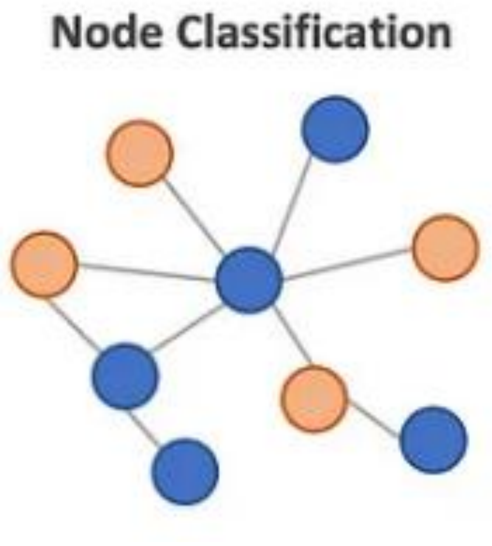
(b)

	1	2	3	4	5
1		1		1	
2	1				1
3				1	1
4	1		1		
5		1	1		

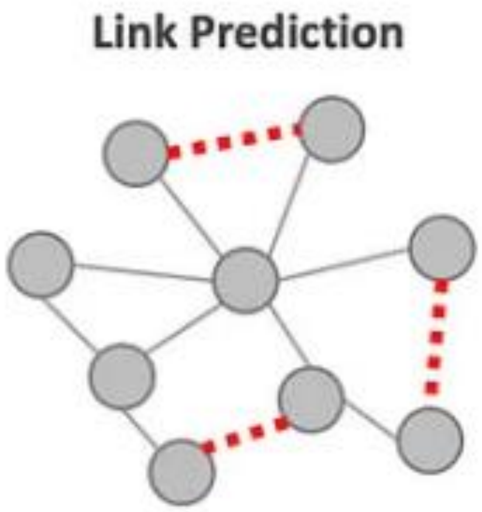
(c)

- Neural network is not permutation equivariant

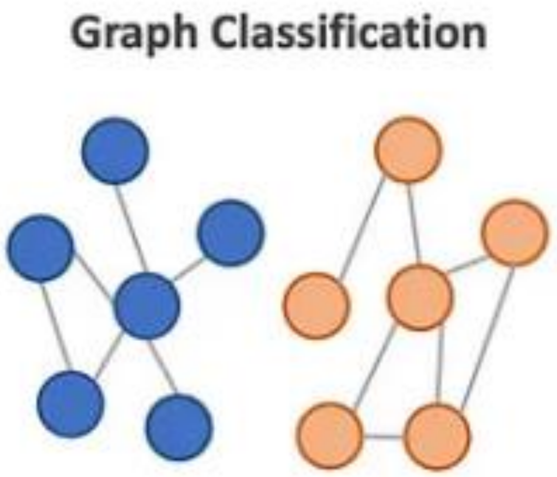
Type of Problem in graph neural network



EX. Scammer classification



EX. recommendation



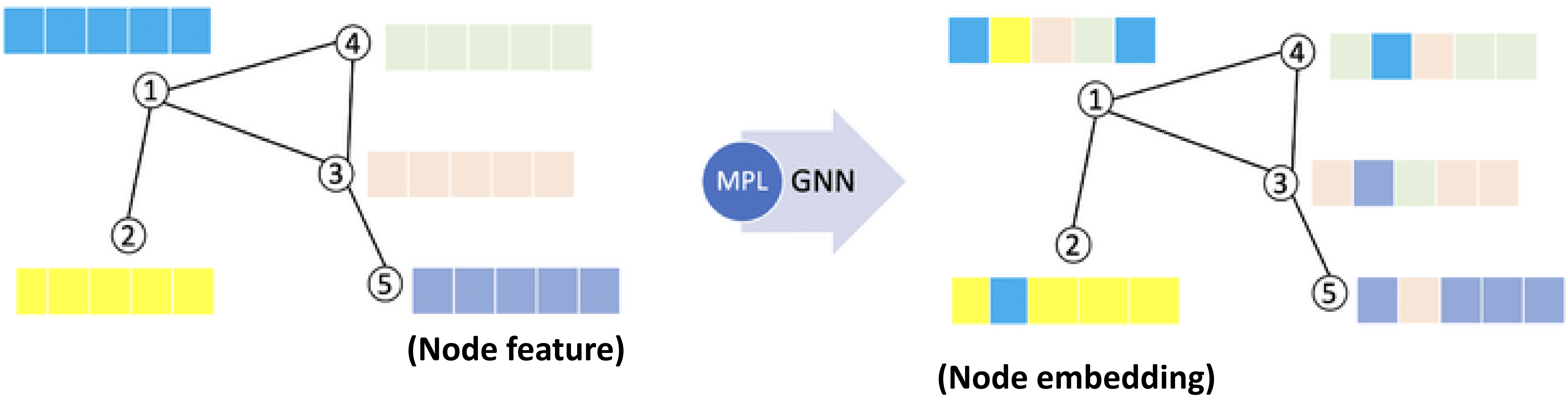
EX. Molecule classification

How graph neural network work

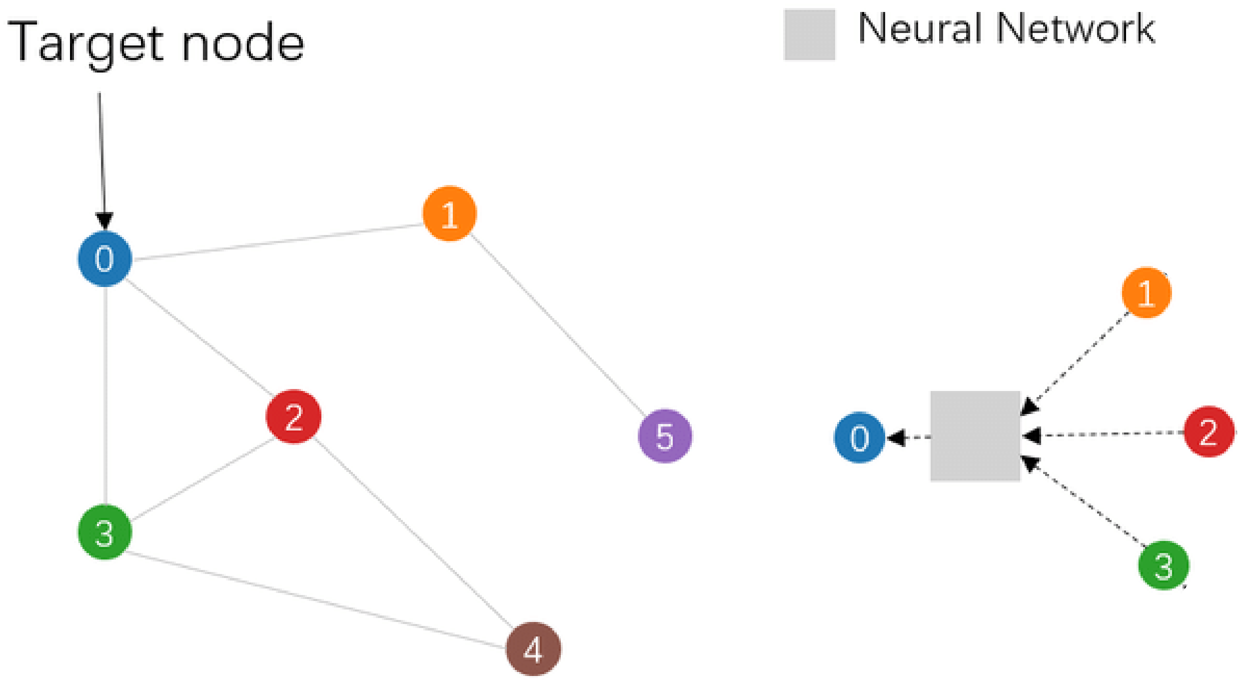
1. **Message passing layers**
2. **Graph pooling**

Message passing layers

Key idea : Update information of each node



Message passing layers



(a) Input graph

(b) Neighborhood aggregation

Message passing layers

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in N_u} \psi(\mathbf{x}_u, \mathbf{x}_v, \mathbf{e}_{uv})\right)$$

Neigborhood aggregation

$\mathbf{h}_u$	Node embedding	$\mathbf{x}_v$	Node feature neighborhood
$\mathbf{x}_u$	Node feature	$\mathbf{e}_{uv}$	edge feature

Message passing layers

Graph Convolutional Network

Key idea : use self node for message aggregation

aggregation

$$\mathbf{H} = \sigma\left(\underbrace{\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}}_{}\mathbf{\Theta}\right)$$

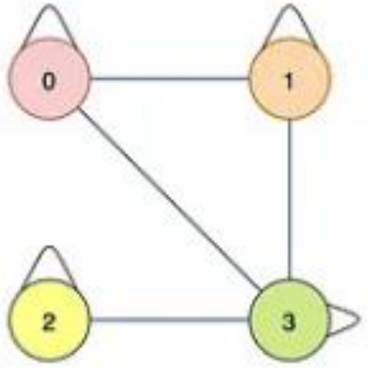
$$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$$

$\hat{A}$

	0	1	2	3
0	1	1	0	1
1	1	1	0	1
2	0	0	1	1
3	1	1	1	1

$\hat{D}$

	0	1	2	3
0	4	0	0	0
1	0	4	0	0
2	0	0	3	0
3	0	0	0	5



Graph Pooling

- Local pooling

Key idea : down-sample node in graph

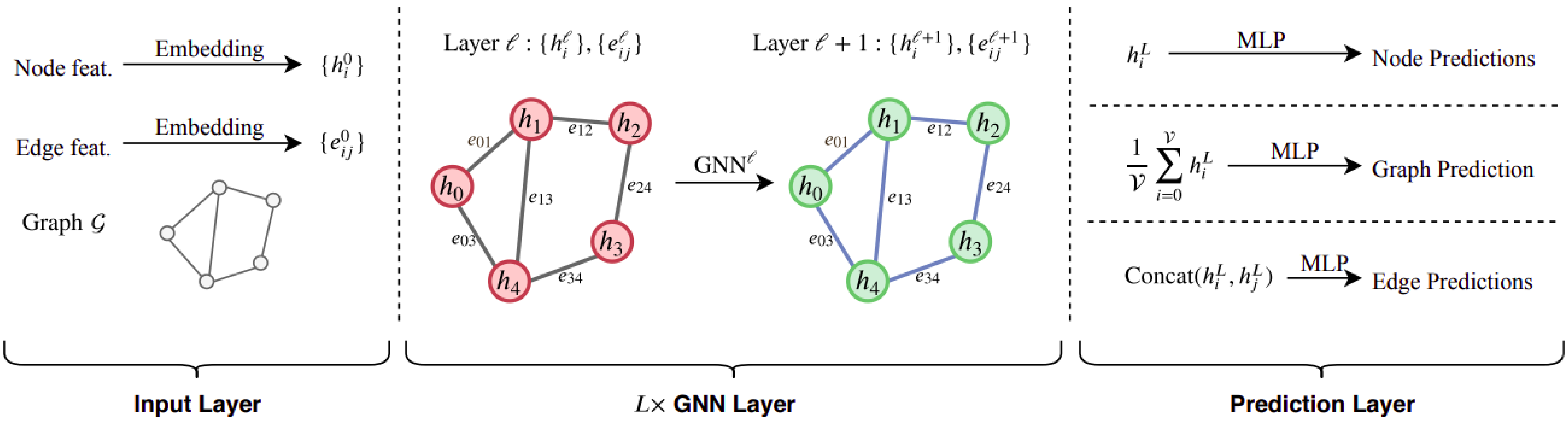
Method	Select	Reduce	Connect
DiffPool [55]	$\mathbf{S} = \mathtt{GNN}_1(\mathbf{A}, \mathbf{X})$ (w/ auxiliary loss)	$\mathbf{X}' = \mathbf{S}^\top \cdot \mathtt{GNN}_2(\mathbf{A}, \mathbf{X})$	$\mathbf{A}' = \mathbf{S}^\top \mathbf{A} \mathbf{S}$
MinCut [6]	$\mathbf{S} = \mathtt{MLP}(\mathbf{X})$ (w/ auxiliary loss)	$\mathbf{X}' = \mathbf{S}^\top \mathbf{X}$	$\mathbf{A}' = \mathbf{S}^\top \mathbf{A} \mathbf{S}$
NMF [3]	Factorize: $\mathbf{A} = \mathbf{W}\mathbf{H} \rightarrow \mathbf{S} = \mathbf{H}^\top$	$\mathbf{X}' = \mathbf{S}^\top \mathbf{X}$	$\mathbf{A}' = \mathbf{S}^\top \mathbf{A} \mathbf{S}$
LaPool [42]	$\begin{cases} \mathbf{V} = \lVert \mathbf{L}\mathbf{X} \rVert_d ; \\ \mathbf{i} = \{i \mid \forall j \in \mathcal{N}(i) : \mathbf{V}_i > \mathbf{V}_j\} \\ \mathbf{S} = \mathrm{SparseMax}\left(\beta \frac{\mathbf{X}\mathbf{X}_\mathbf{i}^\top}{\lVert \mathbf{X} \rVert \lVert \mathbf{X}_\mathbf{i} \rVert}\right) \end{cases}$	$\mathbf{X}' = \mathbf{S}^\top \mathbf{X}$	$\mathbf{A}' = \mathbf{S}^\top \mathbf{A} \mathbf{S}$
Graclus [16]	$\mathcal{S}_k = \left\{\mathbf{x}_i, \mathbf{x}_j \mid \arg\max_j \left(\frac{\mathbf{A}_{ij}}{\mathbf{D}_{ii}} + \frac{\mathbf{A}_{ij}}{\mathbf{D}_{jj}}\right)\right\}$	$\mathbf{X}' = \mathbf{S}^\top \mathbf{X}$	METIS [26]
NDP [7]	$\mathbf{i} = \{i \mid \mathbf{u}_{max,i} > 0\}$	$\mathbf{X}' = \mathbf{X}_\mathbf{i}$	Kron r. [18]
Top-K [24]	$\mathbf{y} = \frac{\mathbf{X}\mathbf{p}}{\lVert \mathbf{p} \rVert}$; $\mathbf{i} = \mathrm{top}_K(\mathbf{y})$	$\mathbf{X}' = (\mathbf{X} \odot \sigma(\mathbf{y}))_\mathbf{i}$;	$\mathbf{A}' = \mathbf{A}_{\mathbf{i},\mathbf{i}}$
SAGPool [30]	$\mathbf{y} = \mathtt{GNN}(\mathbf{A}, \mathbf{X})$; $\mathbf{i} = \mathrm{top}_K(\mathbf{y})$	$\mathbf{X}' = (\mathbf{X} \odot \sigma(\mathbf{y}))_\mathbf{i}$;	$\mathbf{A}' = \mathbf{A}_{\mathbf{i},\mathbf{i}}$

- Global pooling

Key idea : fixed-size representation of the whole graph

- Mean pooling
- Max pooling

Graph Neural Network Pipe-line



implementation

Graph regression

- ESOL: Water solubility data(log solubility in mols per litre) for common organic small molecules.

Name	#graphs	#nodes	#edges	#features	#classes
ESOL	1,128	~13.3	~27.4	9	1

apply in graph (link node graph)

How graph neural network work

Message passing

Local pooling

Global pooling

Imprementation

implementation

DiffPool

$$S^{(l)} = \mathrm{softmax}\left(\mathrm{GNN}_{l,\mathrm{pool}}(A^{(l)}, X^{(l)})\right),$$

$$X^{(l+1)} = {S^{(l)}}^T Z^{(l)} \in \mathbb{R}^{n_{l+1} \times d},$$

$$A^{(l+1)} = {S^{(l)}}^T A^{(l)} S^{(l)} \in \mathbb{R}^{n_{l+1} \times n_{l+1}}.$$