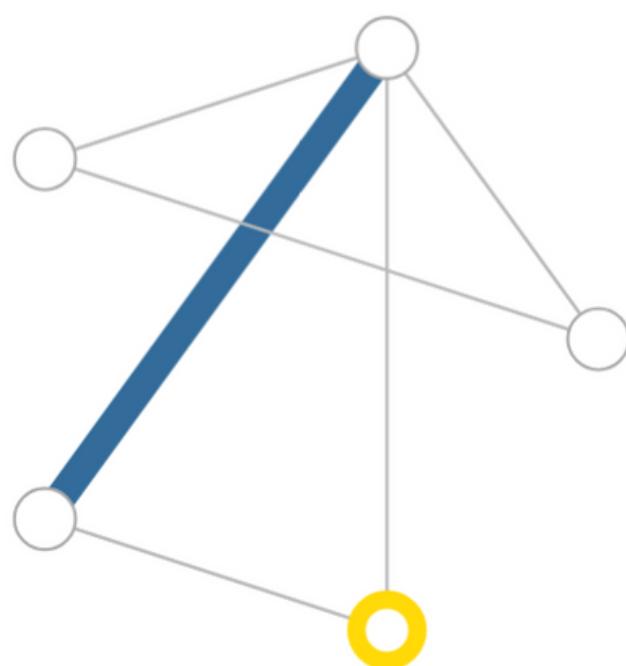


*Graph Neural Networks*

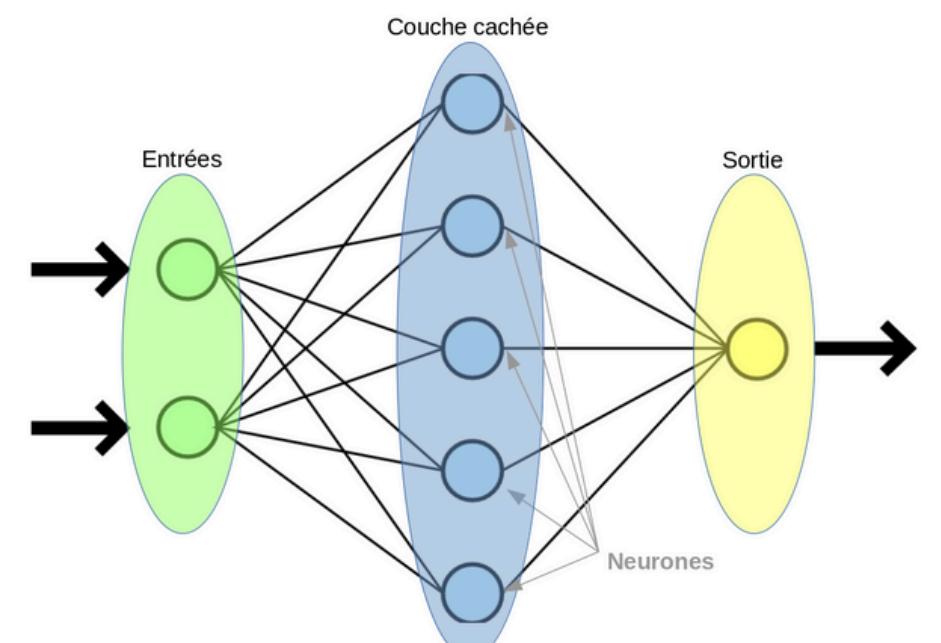
**GNN**

# Introduction : Pourquoi des graphes ?

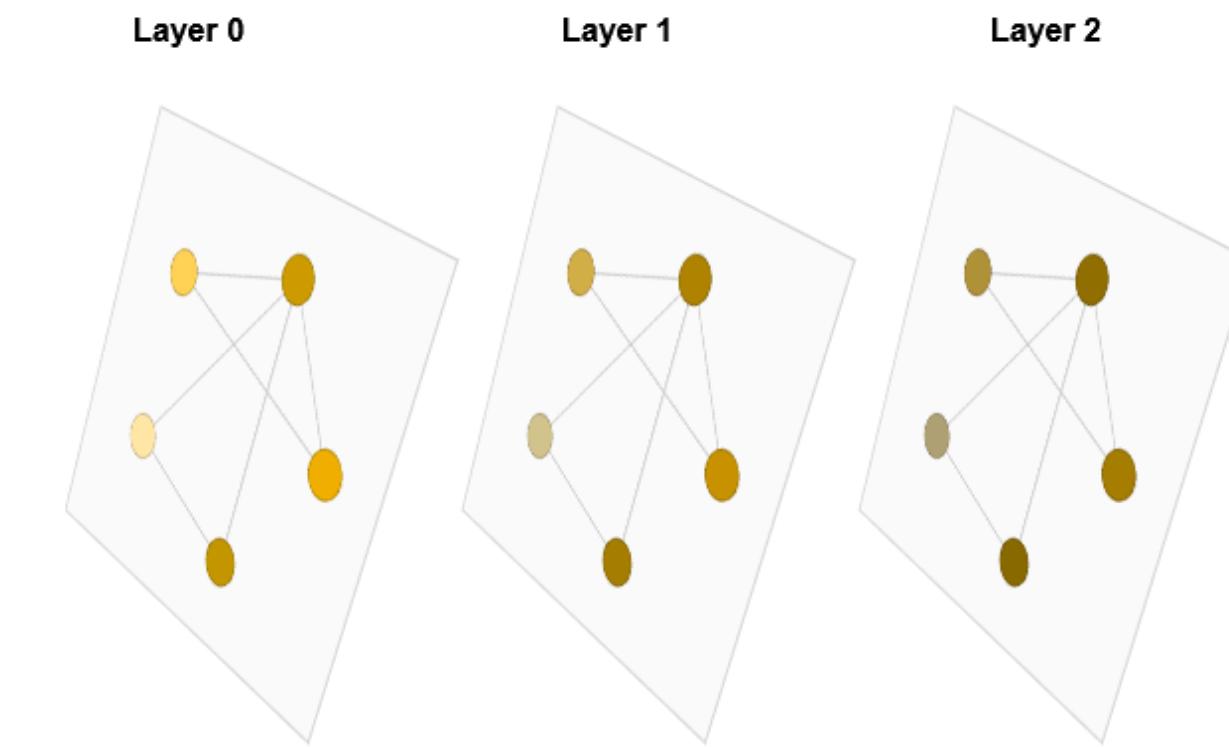
## GNN Graph Neural Networks



+



=

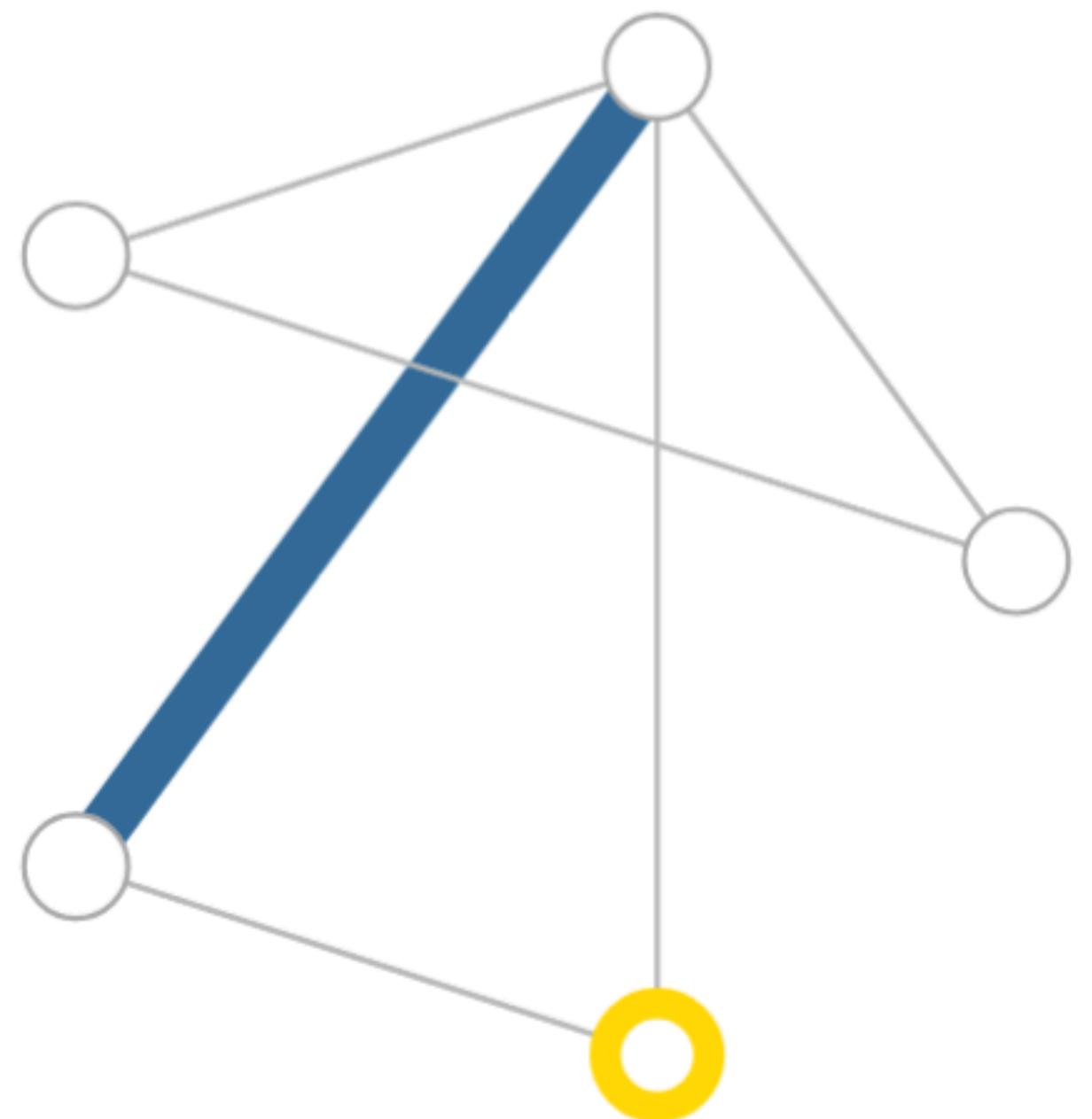


Graphes

Réseaux de neurones

GNN

# Rappel : définition d'un graphe

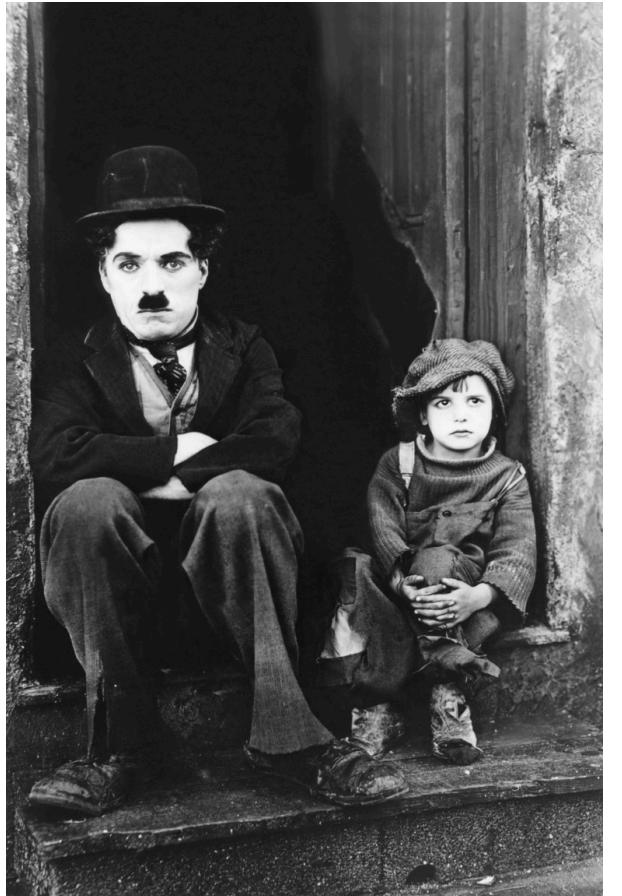


- **Nœuds (Node)** : type de nœud, nombre de voisins
- **Arêtes (Edge)** : type d'arête, poids
- **Global (Master node)** : nombre de nœuds, plus grande arête

# Introduction : Pourquoi des graphes ?

Données Euclidiennes

Images / Textes / Dataset

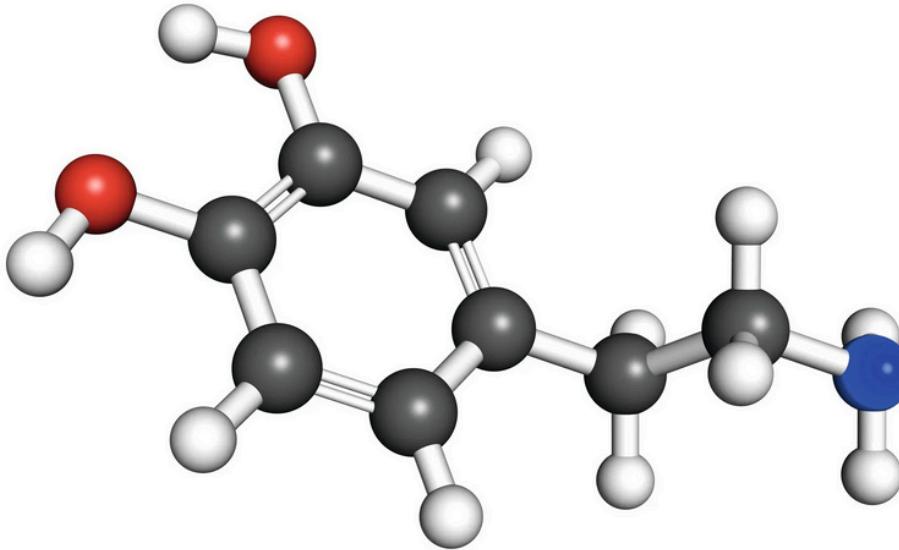


The Tragical Historie of  
HAMLET  
Prince of Denmark.

Enter two Sentinels. [Sc. i.]

1. S Tand: who is that?  
2. Tis I.  
1. O you come most carefully vpon your watch,  
2. And if you meeete Marcellus and Horatio,  
The partners of my watch, bid them make haste.  
1. I will: See who goes there.  
Enter Horatio and Marcellus.  
Hor. Friends to this ground.  
Mar. And leegemen to the Dane,  
O farewell honest souldier, who hath releued you?  
1. Barnardo hath my place, give you good night.  
Mar. Holla, Barnardo.  
2. Say, is Horatio there?  
Hor. A peece of him.  
2. Welcome Horatio, welcome good Marcellus.  
Mar. What hath this thing appear'd againe to night.  
2. I haue seene nothing.  
Mar. Horatio sayes tis but our fantasie,  
And wil not let beliefe take hold of him,  
Touching this dreaded sight twice seene by vs,  
Therefore I haue intreated him a long with vs  
To watch the minutes of this night,  
That if againe this apparition come,  
He may approoue our eyes, and speake to it.  
Hor. Tut, t'will not appear.  
2. Sit downe I pray, and let vs once againe  
Assaile your eares that are so fortified,  
What we haue two nights seene.

Données Non-Euclidiennes

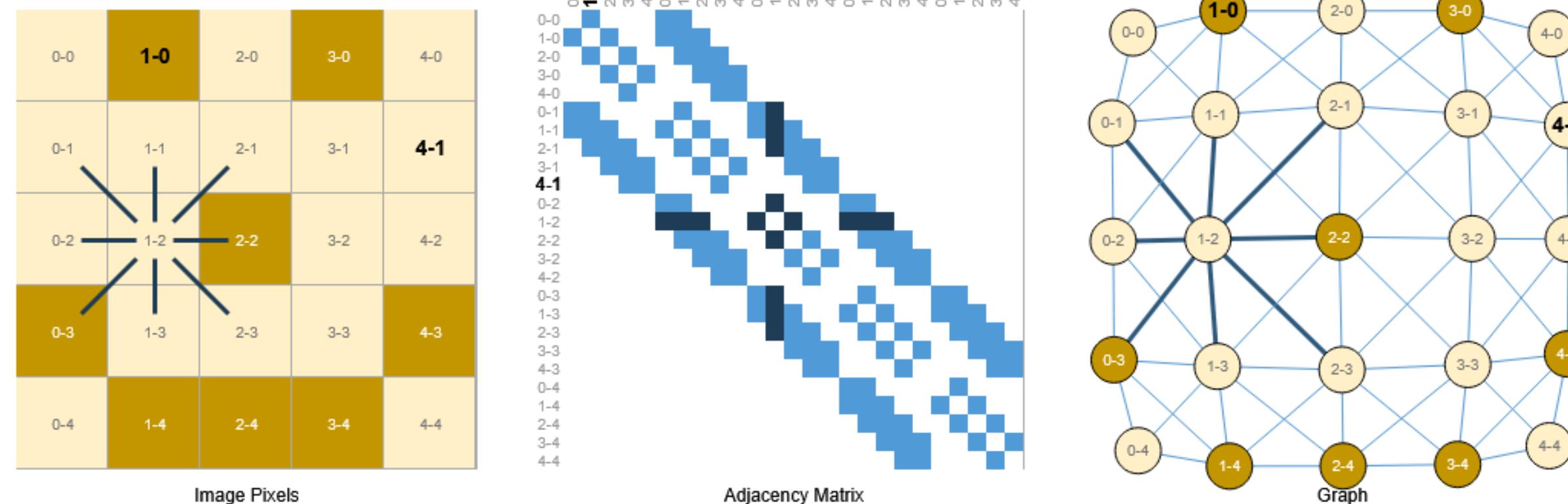


Molécules  
Réseaux Sociaux  
Réseaux de transport...



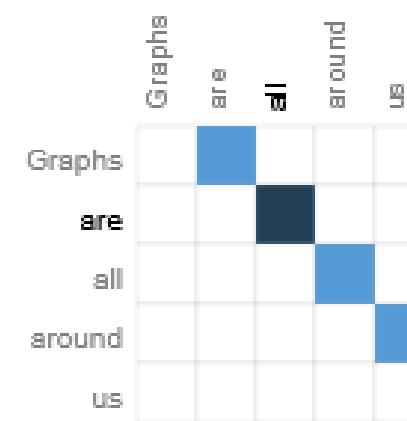
# Graphes comme mode de représentation

## Images



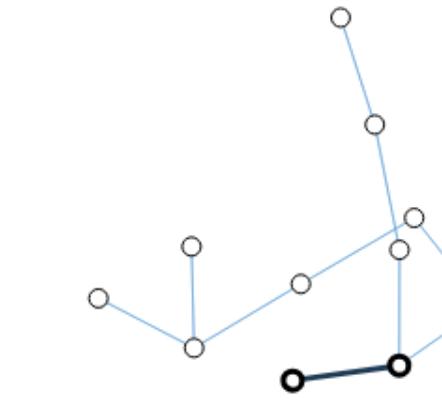
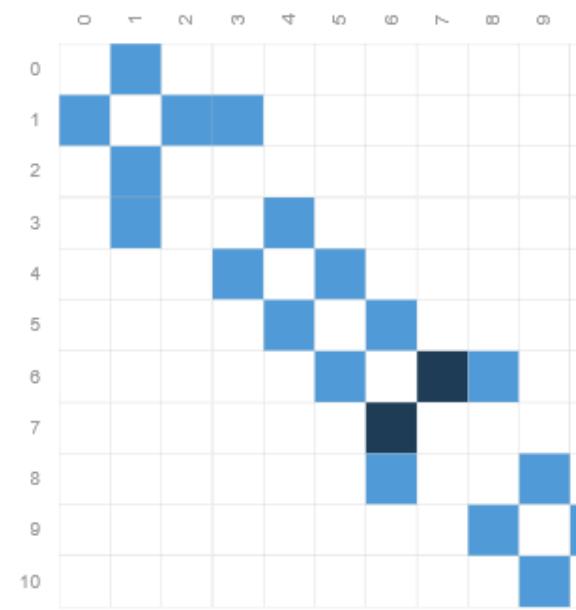
## Textes

Graphs → are → all → around → us

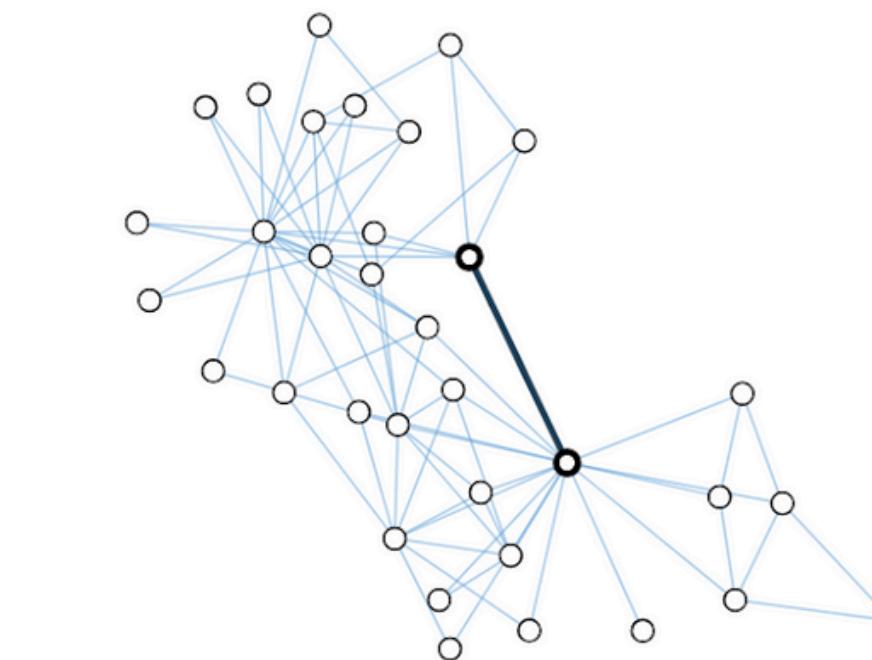
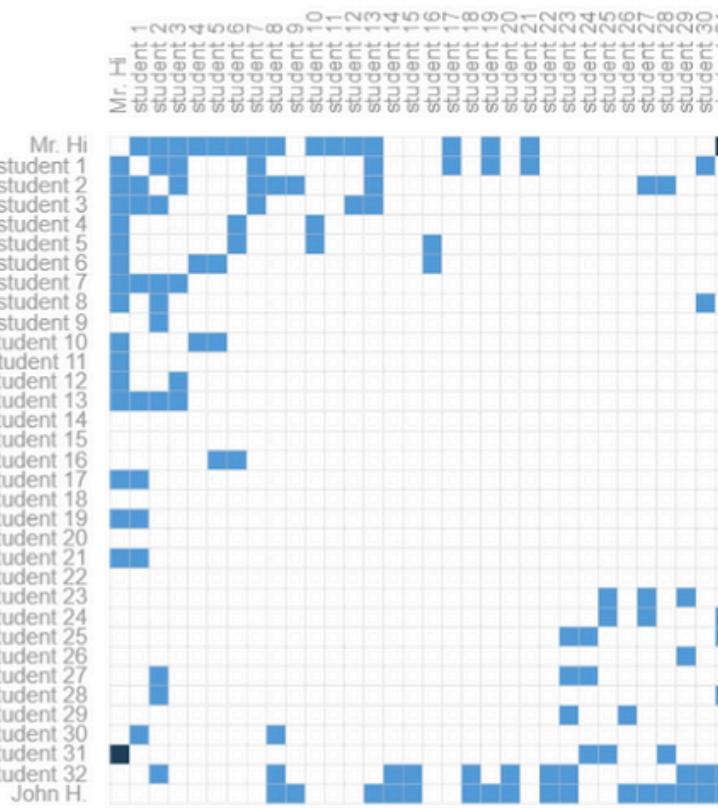


# Graphes comme mode de représentation

## Molécules



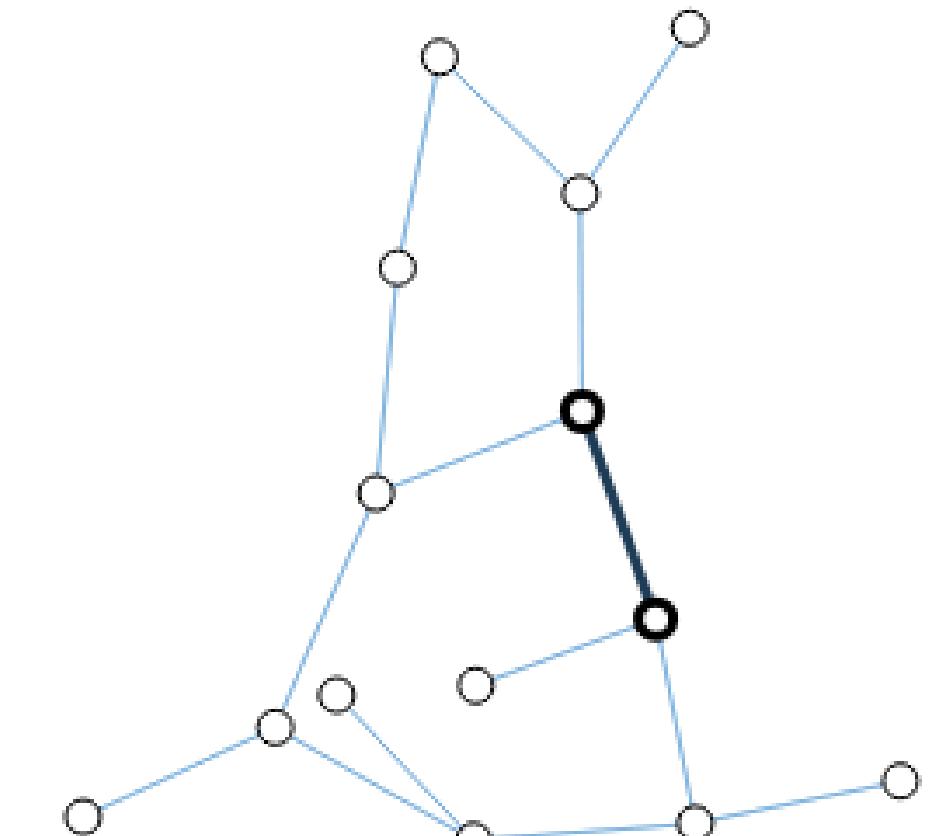
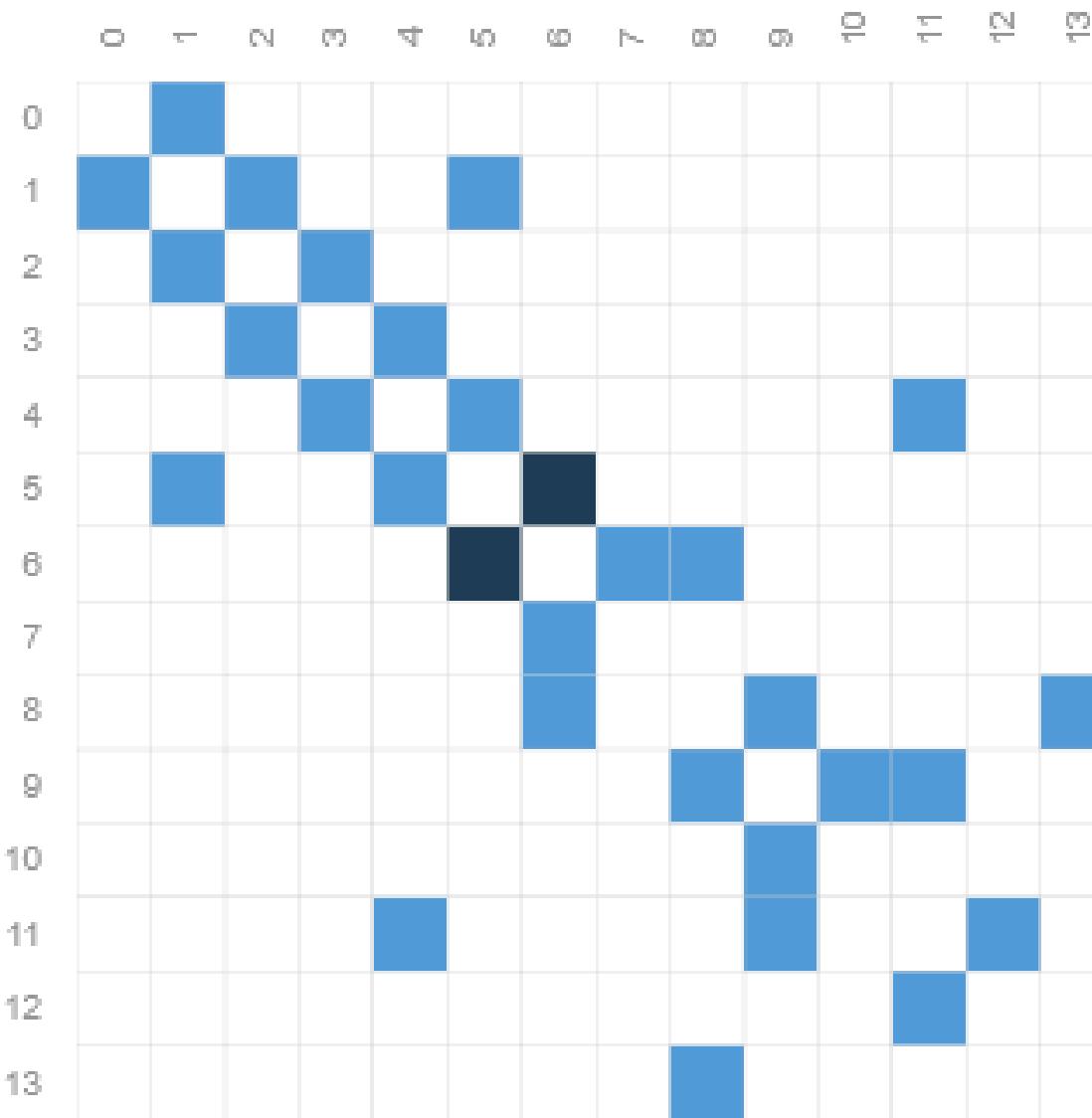
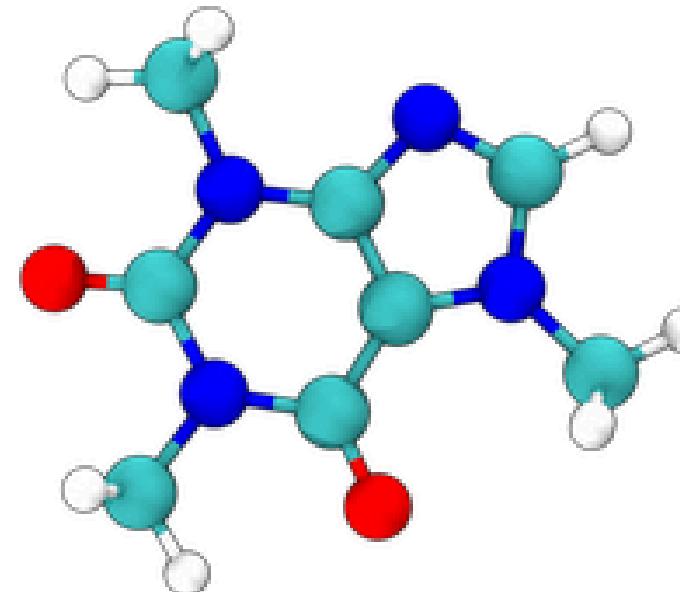
## Relations sociales



(Left) Image of karate tournament. (Center) Adjacency matrix of the interaction between people in a karate club. (Right) Graph representation of these interactions.

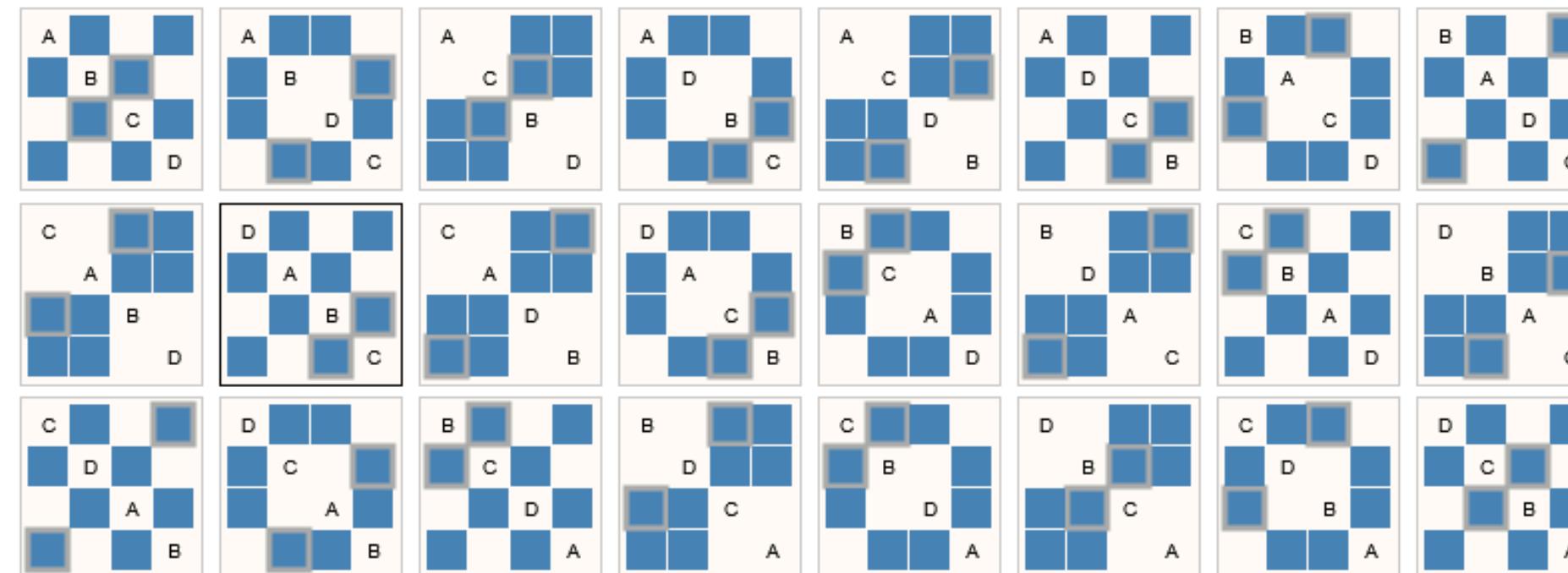
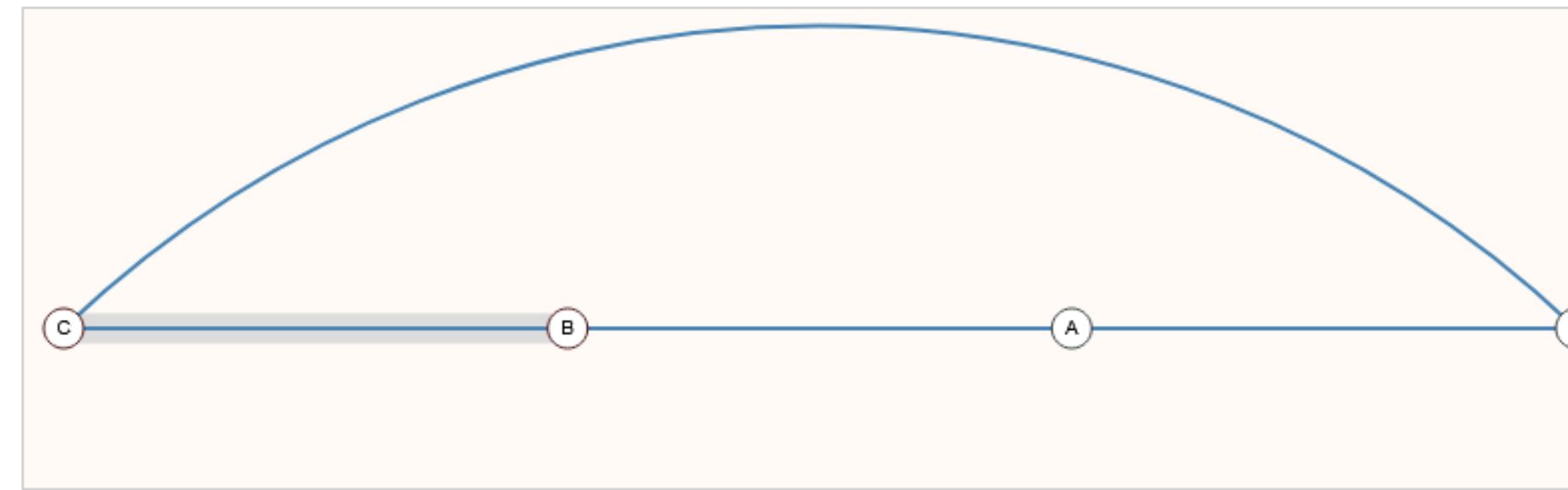
# Implémentation d'un graphe

Méthode usuelle : **Matrice adjacente**



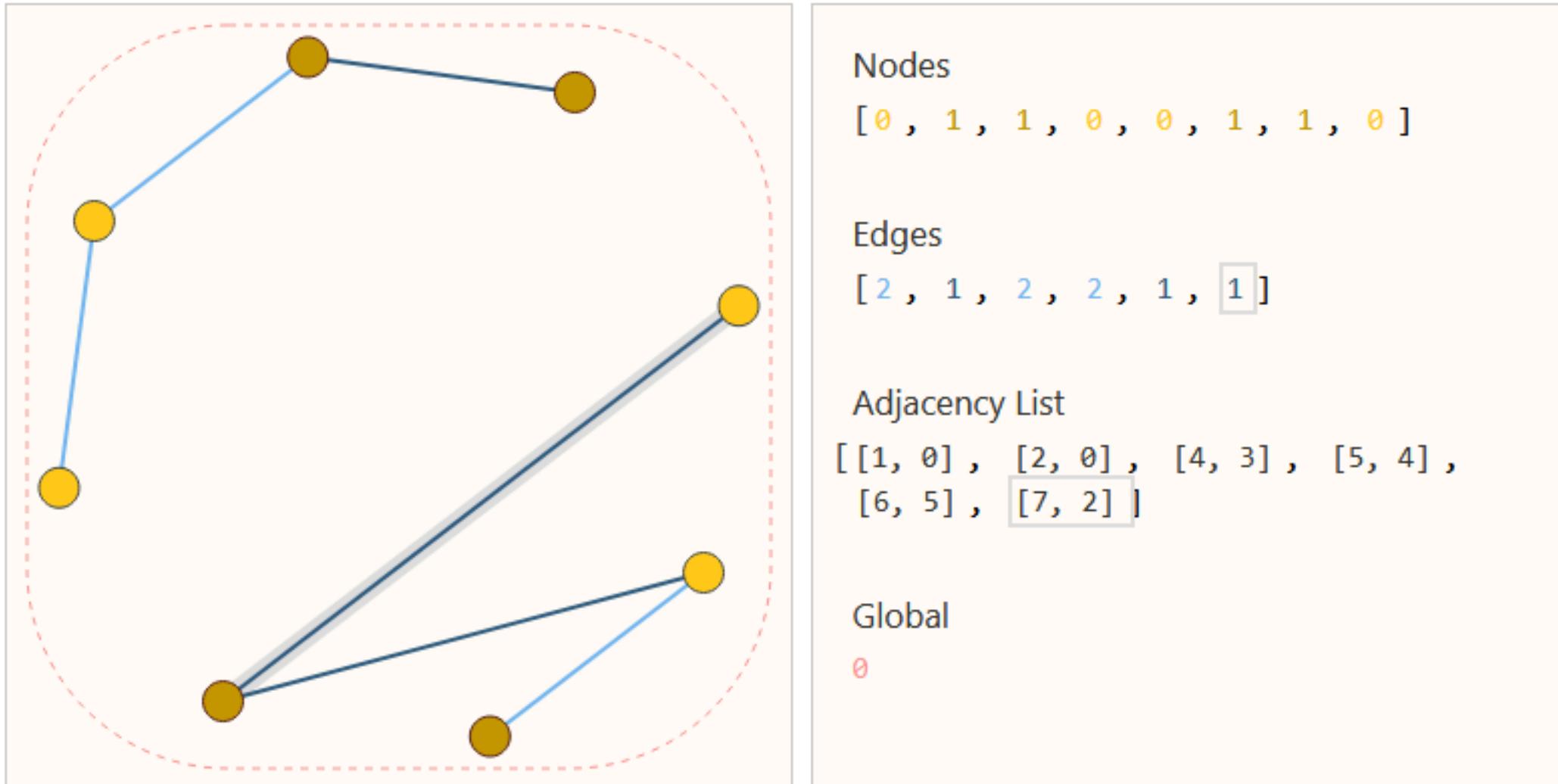
# Implémentation d'un graphe

**Problème :** Non compatible avec des réseaux de neurones  
(multiples représentations d'un même graphe)



# Implémentation d'un graphe

## Solution : Les listes adjacentes



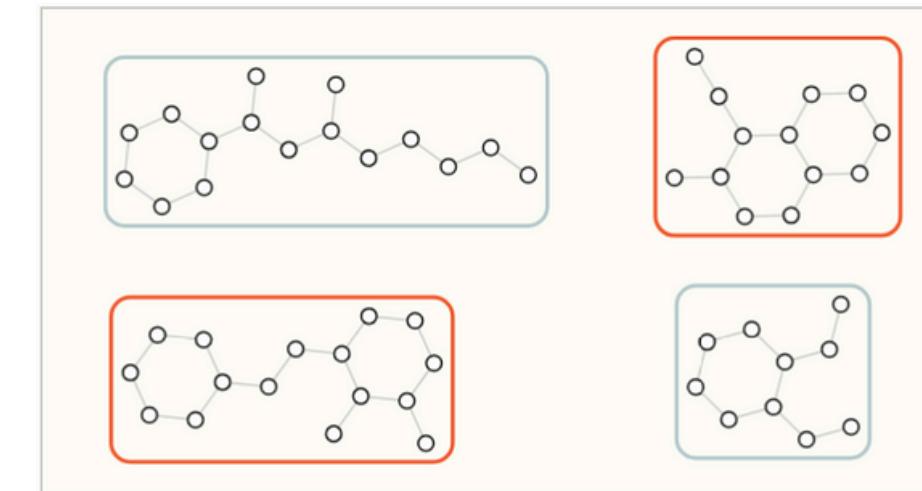
- **Nœuds** : liste indexée de chaque nœud avec poids
- **Arêtes** : liste indexée de chaque arête avec poids
- **Liste adjacente** : tuple  $[i, j]$  de l'arête entre  $i$  et  $j$
- **Global** : entier

# Classification des Problèmes GNN

## Graph-level task

Concept : Classification, différencier un graphe d'un autre

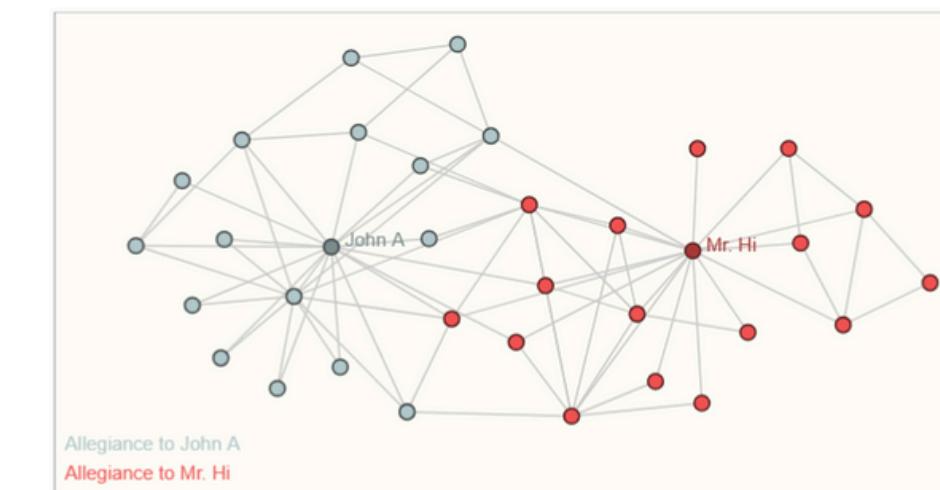
Application : Détection de propriété de molécules



## Node-level task

Concept : prévoir le rôle ou l'identité de chaque nœud

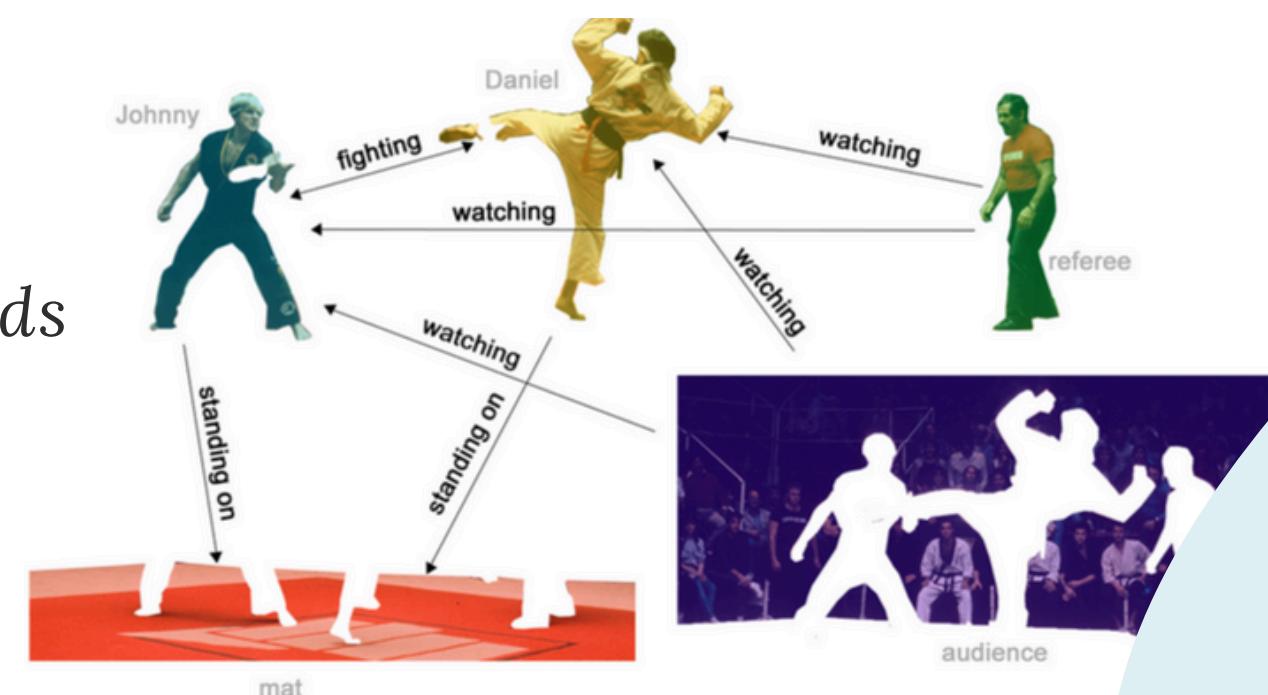
Application : Détection de fraude, détection d'alliances



## Edge-level task

Concept : prévoir le lien entre les différents nœuds

Application : Prédiction d'interactions sociales



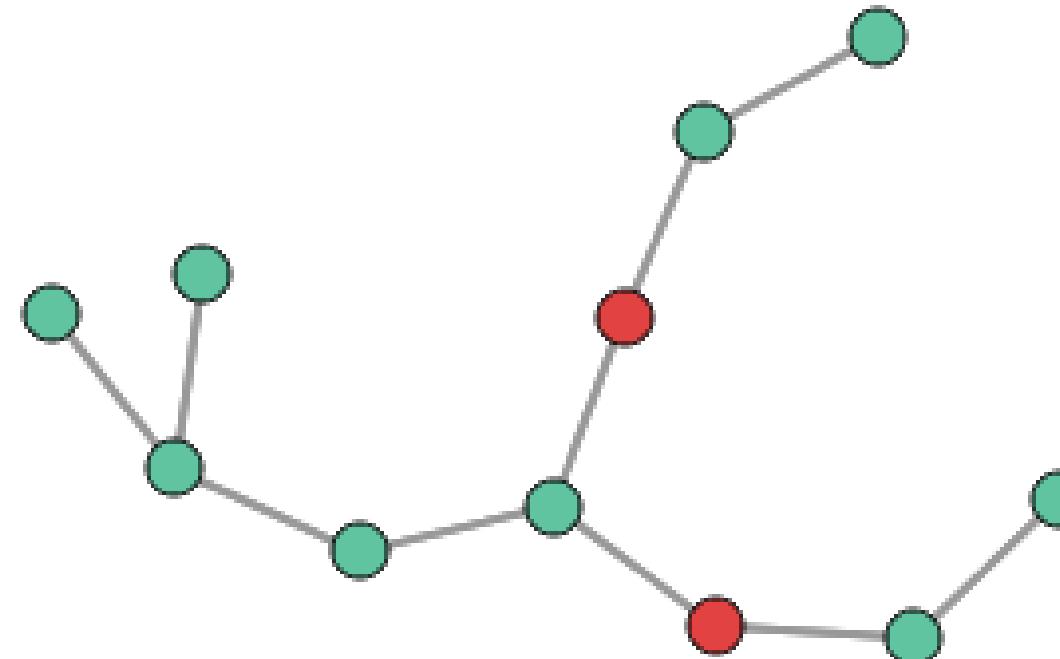
# Concept du GNN : Message Passing

**Définition :** une transformation optimisable sur tous les attributs du graphe (nœuds, arêtes, contexte global) qui préserve les symétries du graphe (invariances de permutation)

**Graph-in → Graph-out**

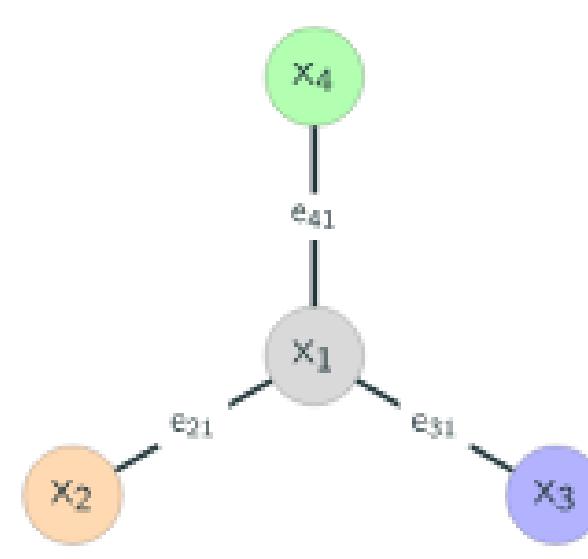
**Message Passing** : processus itératif où chaque nœud reçoit des informations de son voisinage pour mettre à jour son embedding (sa propre représentation vectorielle)

**Message Passing Neural Networks (MPNN)**



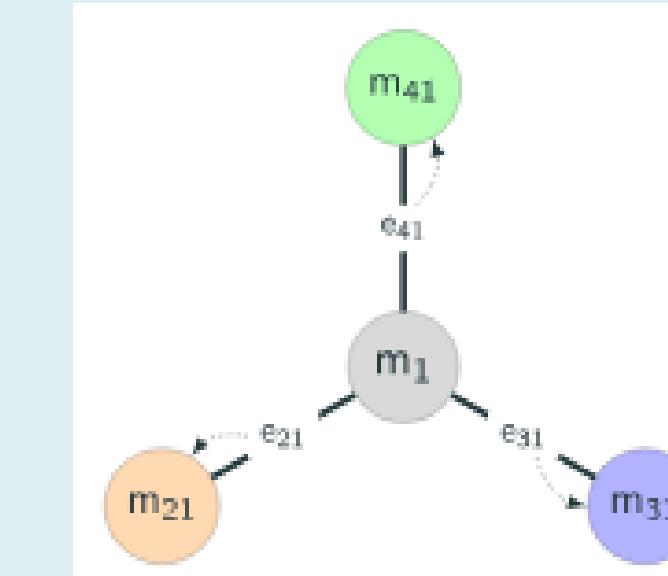
# Message Passing Neural Network

Décomposition d'une couche de convolution en 3 fonctions



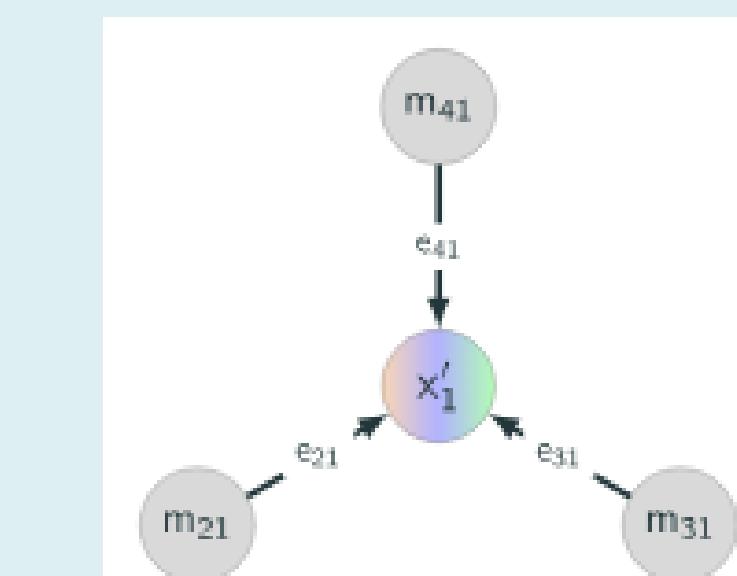
## Fonction de Message

Pour chaque arête reliant un nœud source  $u$  à un nœud cible  $v$ , on calcule un message. Le message dépend des caractéristiques du voisin ( $hu$ ), du nœud lui-même ( $hv$ ) et potentiellement des attributs de la liaison ( $euv$ , comme la distance ou le type de lien).



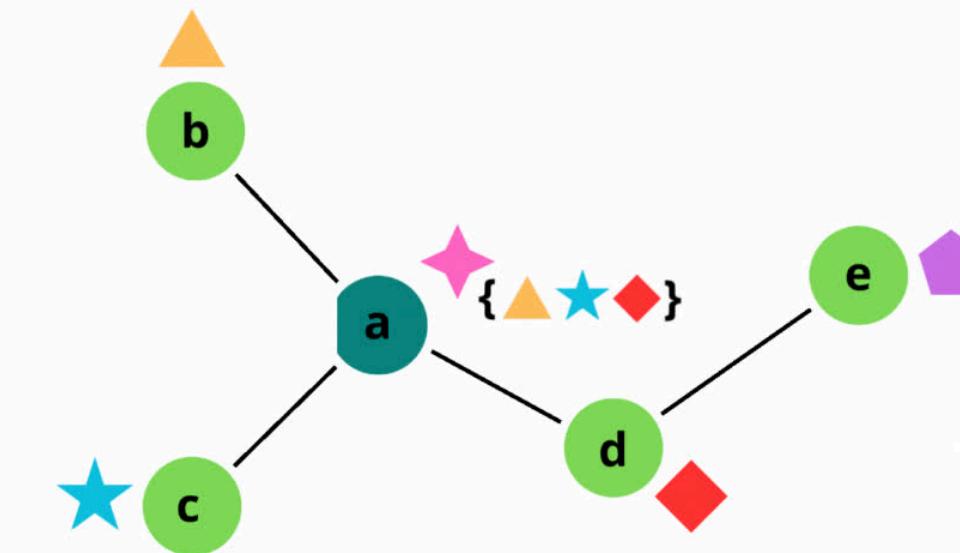
## Fonction d'Agrégation

Fonction qui compresse ces messages en un seul vecteur de taille fixe, peu importe le nombre de voisins  
Contrainte : invariante par permutation  
Opérateurs classiques : Somme, Moyenne, Maximum

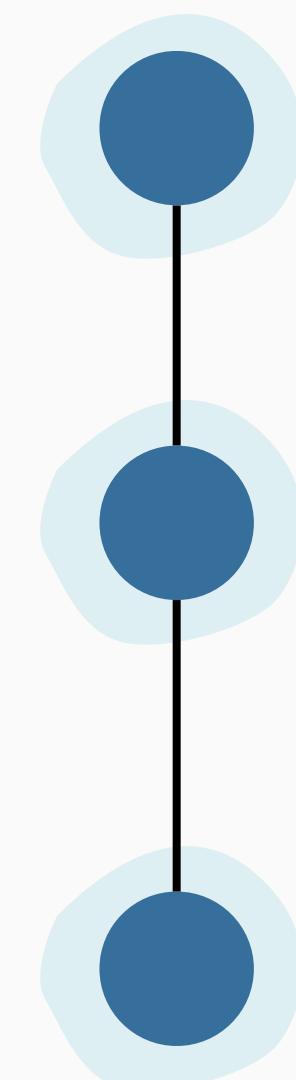


## Fonction d'Update

Le nœud  $v$  a reçu le résumé de son voisinage ( $Mv$ ), il doit mettre à jour son propre état. On combine l'ancien état du nœud avec le message agrégé, souvent via une opération non-linéaire (comme un MLP ou un ReLU).



# Profondeur



## 1 couche de MPNN

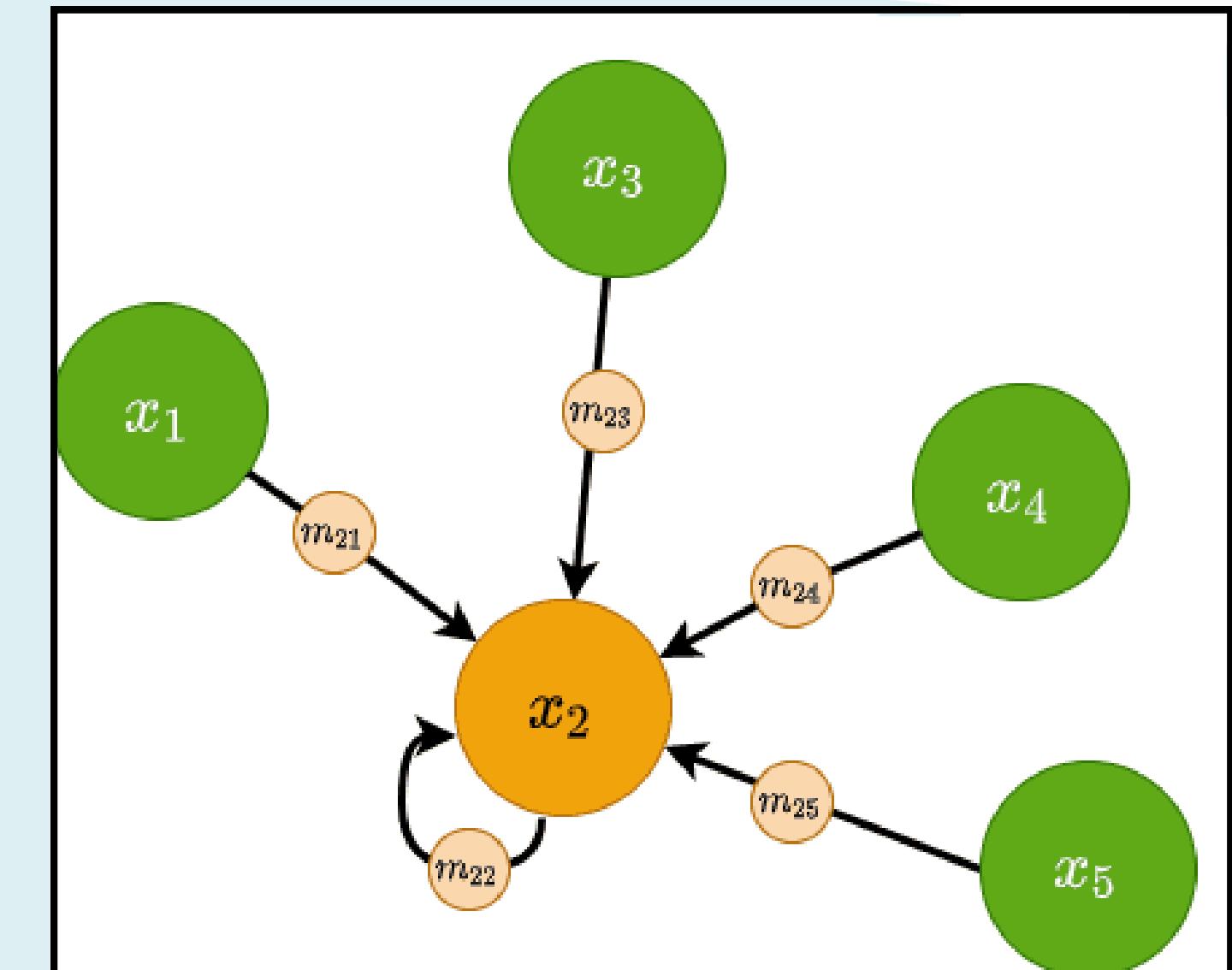
Un nœud connaît ses voisins directs (à 1 saut).

## 2 couches de MPNN

Un nœud reçoit des informations de ses voisins, qui eux-mêmes contiennent déjà des informations de leurs propres voisins. Le nœud a maintenant une vision à 2 sauts.

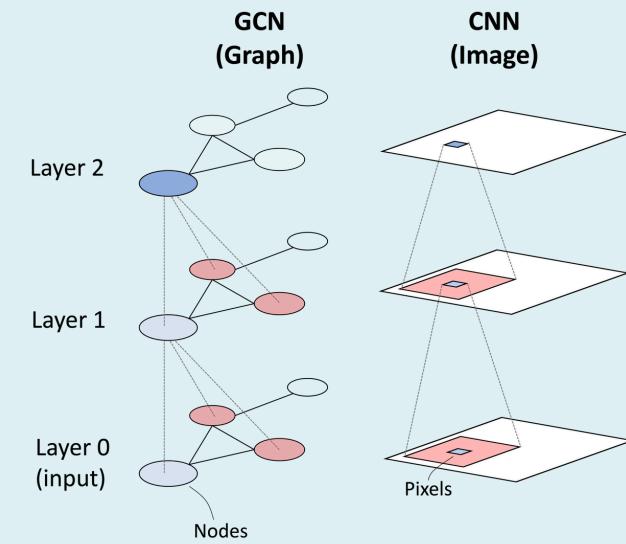
## K couches de MPNN

Le nœud possède une représentation qui encode la structure locale à une distance K.



# Architectures de référence

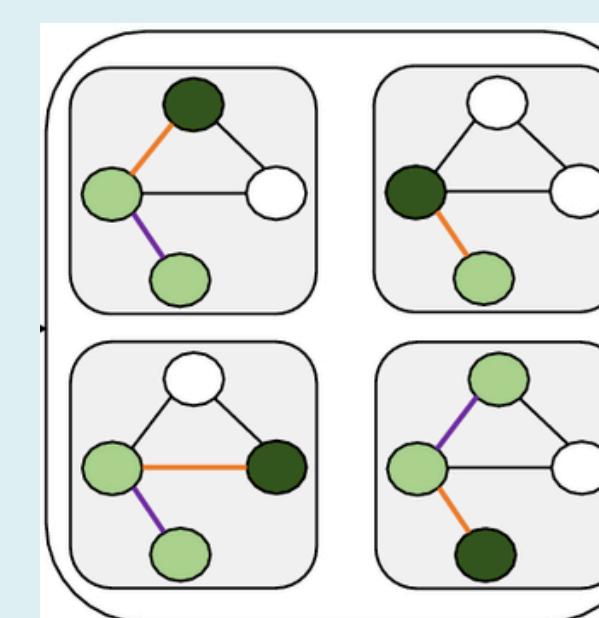
GCN, GraphSAGE & Attention



## GCN (Graph Convolutional Network)

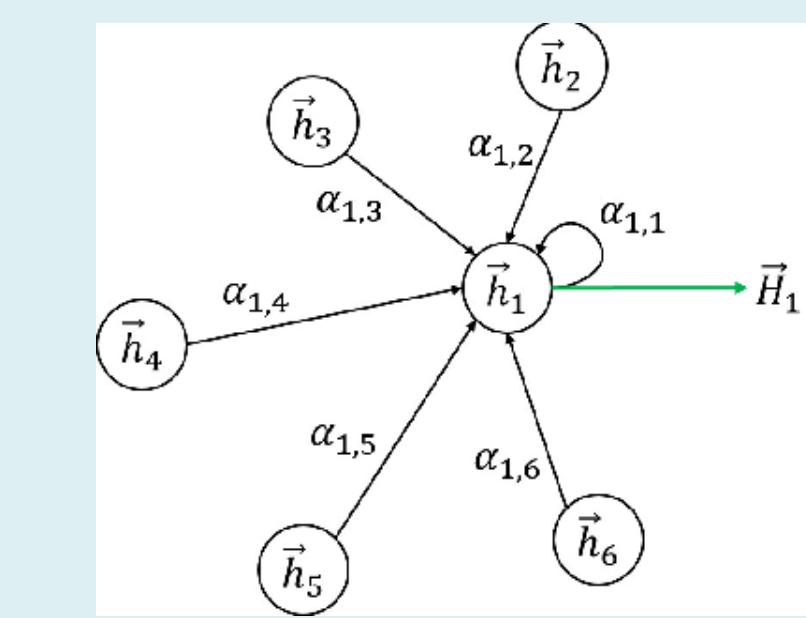
Utilise une approximation spectrale de premier ordre. C'est un filtre passe-bas sur le laplacien du graphe.

Le message dépend des caractéristiques du voisin ( $hu$ ), du nœud lui-même ( $hv$ ) et potentiellement des attributs de la liaison ( $euv$ , comme la distance ou le type de lien).



## GraphSAGE

Introduit l'échantillonnage de voisinage pour passer à l'échelle (inductive learning).



## GAT (Graph Attention Networks)

Introduit des coefficients d'attention anisotropes. Le modèle apprend dynamiquement l'importance relative de chaque voisin, ce qui est critique quand le graphe est bruité.

# Application : ZINC-250K

**Dataset** : ensemble de graphes moléculaires où

- Noeuds = atomes (C, N, O...) encodé par [numéro atomique, charge, aromaticité]
- Arêtes = liaisons chimiques (simple, double, triple, aromatique)
- Cible = valeur scalaire continue mesurant l'hydrophobie d'une molécule

**Pipeline en 3 étapes** :

1. **Embedding Layer** : projection des atomes dans un espace continu
2. **GCN (ou GIN)** : entre 4 et 8 couches pour ZINC. Chaque atome "apprend" son environnement chimique. Un carbone au sein d'un cycle benzénique doit avoir un embedding différent d'un carbone dans une chaîne linéaire.
3. **Readout (Global Pooling)** : fusion de tous les embeddings finaux en un seul vecteur représentant la molécule entière.

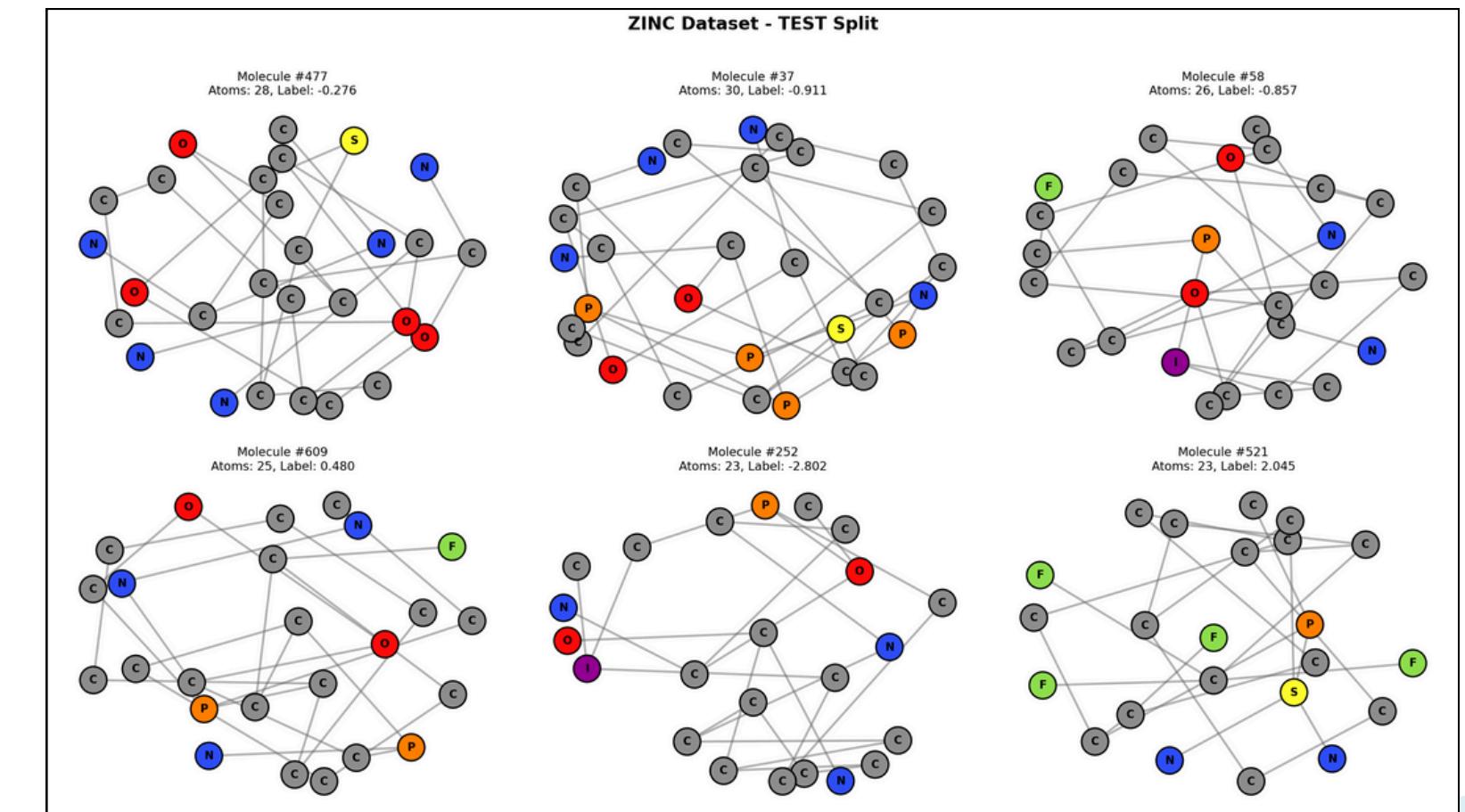
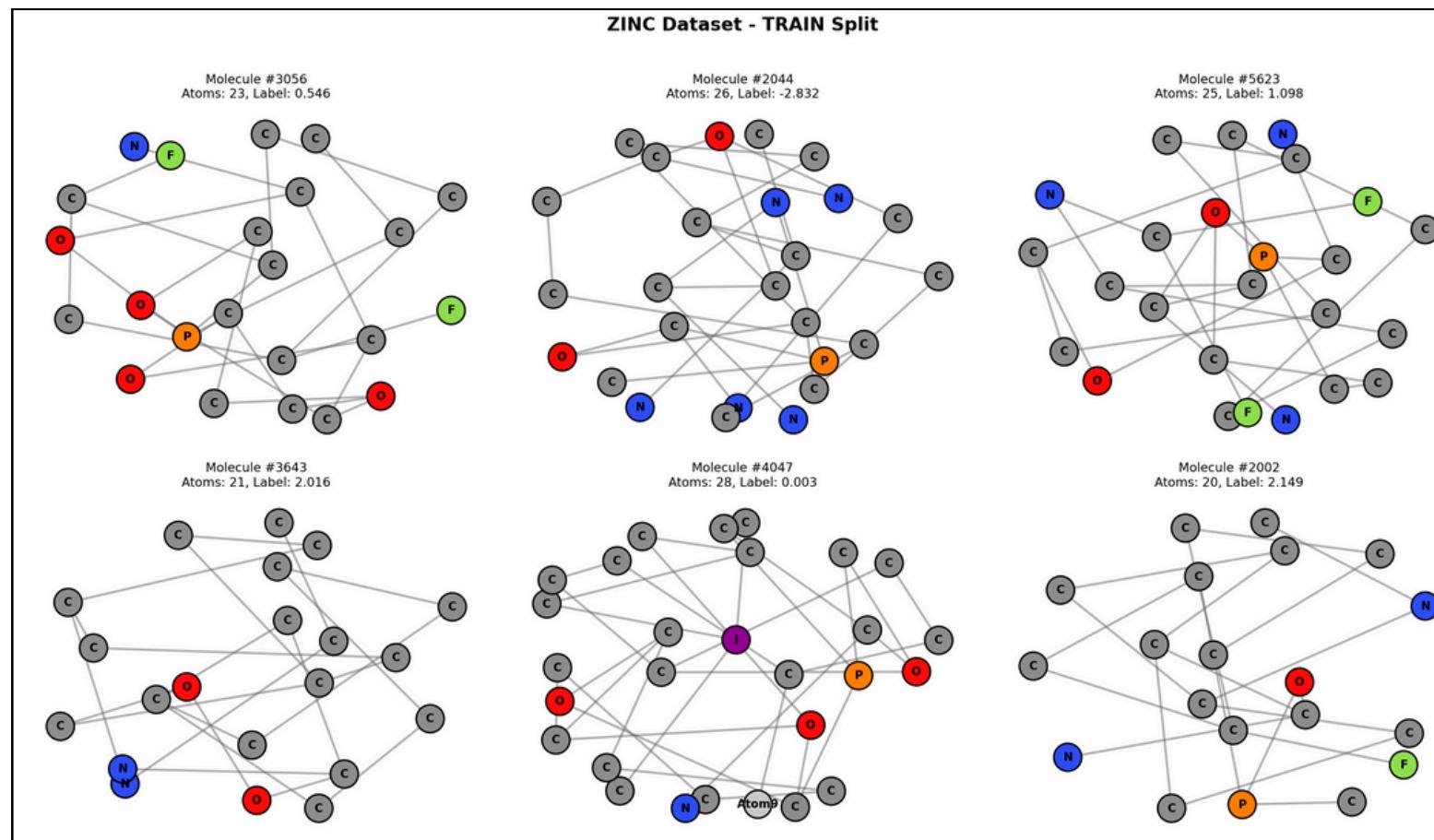
Puis passage dans un MLP (Multi-Layer Perceptron) pour avoir une valeur unique.

# Application : ZINC-250K

Résultats : Test MAE: 0.4057 / Train MAE: 0.3026 (Learning rate final : 1.56e-5)

Grande complexité de liaisons

L'agrégation de voisinage permet au réseau de 'peser' l'importance de chaque type d'atome et de sa position dans la structure pour prédire la solubilité



Le GNN parvient à différencier des molécules ayant un nombre d'atomes similaire mais des connectivités différentes (ex: Molécules #5623 et #609, toutes deux à 25 atomes).

# *Conclusion*